

2/2013

# | atp | journal |

plus



***Control of transport  
and industrial processes***

*reviewed slovak professional  
magazine for scientific  
and engineering issues*

***Riadenie dopravných  
a priemyselných procesov***

*recenzované periodikum  
vedeckých a inžinierskych  
publikácií*

ISSN 1336-5010

**Odborný garant**

prof. Ing. Karol Rástočný, PhD.  
Žilinská univerzita v Žiline, Elektrotechnická fakulta

**Technical guarantee**

prof. Ing. Karol Rástočný, PhD.  
University of Žilina, Faculty of Electrical Engineering

**Vydavateľ**

HMH s.r.o.  
Tavariškova 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**

## Redakčná rada

### Draft committee

prof. Ing. Alexík Mikuláš, PhD., FRI ŽU, Žilina  
doc. Ing. Dvoran Ján, CSc., FCHPT STU, Bratislava  
prof. Ing. Fikar Miroslav, DrSc., FCHPT STU, Bratislava  
doc. Ing. Hantuch Igor, PhD., KAR FEI STU, Bratislava  
doc. Ing. Hrádčoký Ladislav, PhD., SJF TU, Košice  
prof. Ing. Hulkó Gabriel, DrSc., SJF STU, Bratislava  
prof. Ing. Jurišica Ladislav, PhD., FEI STU, Bratislava  
doc. Ing. Kachaňák Anton, CSc., SJF STU, Bratislava  
prof. Ing. Krokavec Dušan, CSc., KKUI FEI TU Košice  
prof. Ing. Madarász Ladislav, PhD., FEI TU, Košice  
prof. Ing. Malindžák Dušan, CSc., BERG TU, Košice  
prof. Ing. Mészáros Alojz, CSc., FCHPT STU, Bratislava  
prof. Ing. Mikleš Ján, DrSc., FCHPT STU, Bratislava  
prof. Dr. Ing. Moravčík Oliver, MTF STU, Trnava  
prof. Ing. Murgaš Ján, PhD., FEI STU, Bratislava  
prof. Ing. Rástočný Karol, PhD., KRIS ŽU, Žilina  
prof. Ing. Schreiber Peter, CSc., MTF STU, Trnava  
prof. Ing. Skyva Ladislav, DrSc., FRI ŽU, Žilina  
prof. Ing. Smieško Viktor, PhD., FEI STU, Bratislava  
doc. Ing. Šturcel Ján, PhD., FEI STU, Bratislava  
prof. Ing. Taufer Ivan, DrSc., Univerzita Pardubice  
prof. Ing. Veselý Vojtech, DrSc., FEI STU, Bratislava  
prof. Ing. Žalman Milan, PhD., FEI STU, Bratislava

**Ing. Bartošovič Štefan,**

generálny riaditeľ ProCS, s.r.o.

**Ing. Boďo Vladimír, CSc.,**

riaditeľ AXESS, spol. s r.o.

**Ing. Csölle Attila,**

riaditeľ Emerson Process Management, s.r.o.

**Ing. Horváth Tomáš,**

riaditeľ HMH, s.r.o.

**Ing. Hrica Marián,**

riaditeľ divízie A & D, Siemens, s.r.o.

**Jiří Kroupa,**

riaditeľ kancelárie pre SK, DEHN + SÖHNE

**Ing. Murančan Ladislav,**

PPA Controll a.s., Bratislava

**Ing. Petergáč Štefan,**

predseda predstavenstva Datalan, a.s.

**Ing. Pilňan Branislav,**

sales leader HPS, HONEYWELL s.r.o.

**Marcel van der Hoek,**

generálny riaditeľ ABB, s.r.o.

## Redakcia

### Editors office

**ATP Journal**

**Galvaniho 7/D**

**821 04 Bratislava 2**

tel.: 02/5026 1752 – 5

fax: 02/5026 1757

e-mail: vydavatelstvo@hmh.sk

www.atpjournal.sk

**Ing. Anton Gérer**

šéfredaktor – editor in chief

gerer@hmh.sk

**Ing. Martin Karbovanec**

vedúci vydavateľstva – editorial office manager

karbovanec@hmh.sk

**Ing. Branislav Bložon**

odborný redaktor – editor

blozon@hmh.sk

**Peter Kanda**

DTP grafik – DTP graphic designer

ntp@hmh.sk

**Dagmar Votavová**

obchod a marketing – sales and Marketing

atp\_podklady@hmh.sk

**Mgr. Bronislava Chocholová**

jazyková redaktorka – text corrector

Datum vydania: júl 2013

Autorské práva HMH, s.r.o., Všetky práva vyhradené

Copyright HMH, Ltd., 2011, All Rights Reserved

ISSN 1336-5010

**ATP Journal PLUS 2/2013**

## Predhovor

60 rokov automatizácie na Katedre riadiacich a informačných systémov .....	6
J. Spalek	

## Riadenie dopravných procesov

Tvorba 3D modelu a klasifikácia objektov pozemnej komunikácie s využitím odrazivosti laserového lúča .....	11
J. Halgaš, M. Hruboš, A. Janota	
Globálny a mentálni modely dopravy v mestských aglomeráciách.....	17
P. Příbyl, O. Příbyl	
Jednoduchý nízkonákladový informačný systém pre vodičov.....	23
P. Vestenický, M. Vestenický	
Fuzzy model doby reakcie osôb v cestnom tuneli pri vzniku mimoriadnej udalosti.....	28
P. Matis, J. Spalek	
Integrácia bezpečného riešenia LDWS systému s použitím VANET siete v inteligentných dopravných systémoch .....	34
E. Bubeníková, M. Franeková, J. Beňuš	
Aplikácie v dopravnej telematike .....	41
R. Pirník	
Núdzové ovládanie riadiacich systémov v železničnej doprave .....	45
P. Nagy, J. Ždánsky	

## Riadenie priemyselných procesov

Použitie kontextových blokov v generickom programovaní s JIT kompiláciou .....	51
M. Gregor, J. Spalek	
Adaptívne riadenie parametrov v generickom programovaní založené na hodnotení funkcie fitness .....	57
M. Gregor, J. Spalek	
Tvorba modelu reálneho akčného člena za účelom návrhu regulátora .....	62
J. Hrbček, V. Šimák, L. Pekár	
Lineárne riadenie prirodzene nestabilného systému s použitím PID regulátora .....	68
V. Šimák, D. Nemeč, J. Hrbček, E. Bubeníková	
Virtualizácia MES systémov integrovaných v distribuovaných systémoch riadenia.....	74
P. Peniak, M. Franeková	
Numerický model systému s vysokou dostupnosťou pomocou virtualizácie serverom.....	78
P. Peniak	
Pohotovosť a bezpečnosť typických architektúr SRCS so safety PLC.....	82
K. Rástočný, J. Ždánsky	
Vplyv diagnostiky na integritu bezpečnosti SRCS so safety PLC.....	87
J. Ždánsky, K. Rástočný	
Špecifické požiadavky na softvér SRCS so safety PLC.....	92
J. Ždánsky, K. Rástočný	
Optimalizácia plánu broadcast vysielania v bezdrôtových senzorových sieťach .....	96
P. Holečko	

## Inteligentné budovy

Je budúcnosť inteligentných domov v automatizácii rutinných ovládacích úloh? .....	100
T. Miklušček, A. Janota	
Návrh inteligentného používateľského rozhrania pre inteligentný dom .....	106
D. Michalovič, T. Miklušček, A. Janota	
Virtuálny 3D model inteligentnej budovy .....	113
M. Kostka, T. Miklušček, A. Janota	
Bezpečnosť siete ZigBee a jej použitie v inteligentnom dome.....	120
J. Ďurech, M. Franeková	

# Content

## Preface

60 years of Control Engineering at the Department of Control and Information System .....	6
J. Spalek	

## Control of transport processes

The 3D model design and the objects classification of a road communication utilizing an echo value of laser beam .....	11
J. Halgaš, M. Hruboš, A. Janota	
Global and mental models of traffic in urban agglomerations .....	17
P. Příbyl, O. Příbyl	
Simple and low cost information system for car drivers .....	23
P. Vestenický, M. Vestenický	
Fuzzy model used to estimation reaction time of people during incident in road tunnel .....	28
P. Matis, J. Spalek	
Integration of secure solutions LDWS using VANET networks in intelligent transport systems .....	34
E. Bubeníková, M. Franecková, J. Beňuš	
Transport telematics applications .....	41
R. Pirník	
Emergency operation of the control systems in railway transport .....	45
P. Nagy, J. Ždánsky	

## Control of industrial processes

Using context blocks in genetic programming with JIT compilation .....	51
M. Gregor, J. Spalek	
Fitness-based Adaptive Control of Parameters in Genetic Programming .....	57
M. Gregor, J. Spalek	
Creating the model of real actuator for the purpose of design the controller .....	62
J. Hrbček, V. Šimák, L. Pekár	
Linear control of naturally unstable system using PID regulator .....	68
V. Šimák, D. Nemeč, J. Hrbček, E. Bubeníková	
Virtualization for manufacturing execution systems integrated with distributed control systems .....	74
P. Peniak, M. Franecková	
Numerical model of high availability system with server virtualization .....	78
P. Peniak	
Availability and safety of typical SRCS architectures with safety PLC .....	82
K. Rástočný, J. Ždánsky	
A effect of diagnostics on the safety integrity of SRCS with safety PLC .....	87
J. Ždánsky, K. Rástočný	
Specific software requirements of SRCS with safety PLC .....	92
J. Ždánsky, K. Rástočný	
Optimisation of broadcast scheduling in wireless sensor networks .....	96
P. Holečko	

## Intelligent buildings

Is automation of routine control tasks a future for smart houses? .....	100
T. Miklušček, A. Janota	
Design of the smart user interface for the smart house .....	106
D. Michalovič, T. Miklušček, A. Janota	
The virtual 3D model of the intelligent building .....	113
M. Kostka, T. Miklušček, A. Janota	
Safety of ZigBee network and its using in intelligent house .....	120
J. Ďurech, M. Franecková	

# 60 rokov automatizácie na Katedre riadiacich a informačných systémov

Juraj Spalek

## Abstrakt

Príspevok je venovaný 60 výročiu vzniku katedry, ktorej pedagogickým i vedecko-výskumným portfóliom je automatizácia. Článok opisuje stručnú históriu katedry, jej pozíciu na materskej univerzite a fakulte, charakteristiku súčasných priorít v oblasti vedy a výskumu, Okrem toho sa venuje formám spolupráce s akademickými a komerčnými inštitúciami doma i v zahraničí a tiež poskytuje stručný prehľad najvýznamnejších projektov, riešených jej členmi v ostatnom období.

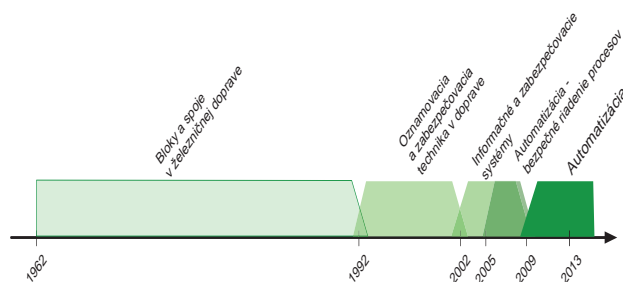
**Kľúčové slová:** Katedra riadiacich a informačných systémov, profil, automatizácia, veda a výskum, bezpečnostne kritický proces, Elektrotechnická fakulta, Žilinská univerzita

## Úvod

V súčasnosti, kedy je na trhu práce v oblasti automatizácie dopravných a výrobných technologických procesov najviac žiadaný „analytik - automatizér, ovládajúci informačné technológie a zameraný na konkrétne riadenie procesov. Odborník, ktorý ovláda fyzikálnu podstatu teórie riadenia a vie ju aplikovať v reálnom živote“ (prof. Ing. Vojtech Veselý, DrSc., URPI FRI STU), je záujem nielen komerčnej sféry, ale aj maturantov zameraných na moderné, dynamicky sa rozvíjajúce študijné odbory, ktorých obsah je na hranici medzi aplikovanou informatikou, mikroprocesorovou riadiacou technikou a niečím, čo zaujme budúceho zamestnávateľa a zaručí absolventovi – zamestnancovi atraktívnu pracovnú pozíciu. V prípade Katedry informačných a riadiacich systémov sa za jedinečnú pridanú hodnotu dlhodobo považuje schopnosť absolventov navrhovať riadiace systémy pre bezpečnostne kritické procesy, kde zlyhanie riadiaceho a informačného systému môže spôsobiť významné škody na zdraví ľudí, majetku, životnom prostredí a pod. Preto sa od riadiacich systémov vyžaduje vysoká vopred preukázaná funkčná bezpečnosť, bezpečnosť akvizície, prenosu, uschovania a spracovania informácií. Ide o novú vednú disciplínu, ktorá má širokú podporu nielen v EÚ, ale aj vo svete.

## 1. Stručne o histórii katedry

Vznik Katedry riadiacich a informačných systémov (KRIS) je spätý so vznikom Vysokej školy železničnej v Prahe (1953), po presťahovaní do Žiliny (1959 – 1962) s históriou Vysokej školy dopravnej, kde bola s pôvodným názvom Katedra blokov a spojov (KBS) súčasťou Strojníckej a elektrotechnickej fakulty. Katedra mala vynikajúci odborný potenciál, uznávaný doma i v zahraničí. V tomto ranom období katedry zohrali významnú úlohu viacerí odborníci v oblasti železničnej a zabezpečovacej techniky, najmä prof. V. Chudáček, prof. V. Kroutl, prof. K. Ludvík, prof. O. Poupě. Výrazným vývojom prešiel aj obsah a názov študijného odboru inžinierskeho štúdia, ktorý vtedy katedra zabezpečovala (obr. 1).



Obr. 1 Genéza študijného odboru Automatizácia na katedre

Fig. 1 Genesis of the Automation study branch

## 2. Súčasnosť katedry

Katedra je jednou zo ôsmich katedier Elektrotechnickej fakulty ŽU. Aj keď je jej súčasná kvalifikačná štruktúra pomerne nevyvážená (4 profesori, 1 hosťujúci profesor, len 1 docent), priemerný vek 42 rokov je dobrou východiskovou pozíciou pre ďalší rozvoj. Dominantné oblasti vedeckovýskumnej práce pracovníkov katedry sú v tab. 1. Aktivity katedry sú integrované v rámci národnej a medzinárodnej spolupráce s akademickou a priemyselnou sférou a nadobúdajú najrôznejšie podoby - od výskumných projektov až po výmenu študentov a odborníkov.

Pozornosť si zasluhujú špeciálne laboratória katedry, založené v spolupráci s domácimi i zahraničnými partnerskými inštitúciami. Ako príklady možno uviesť Spoločné vývojovo-výskumné laboratórium výskumu nových sofistikovaných riadiacich technológií pre železničnú a cestnú dopravu so spoločnosťou Betamont, s.r.o., Spoločné laboratórium tunelových systémov s Dopravnou fakultou ČVUT Praha, Laboratórium bezpečnostne kritických riadiacich systémov so spoločnosťami Siemens, s.r.o. a B+R automatizace, s.r.o. Nemenej významnou je dlhodobá spolupráca s firmou Scheidt & Bachmann Slovensko, s.r.o. nielen v projektovej oblasti, ale aj pri budovaní spoločného laboratória bezpečnostne kritických riadiacich systémov.

Podrobnejšie údaje sú vo výročných správach [2].

T. Brončeková, Ing. PhD.	Algoritmizácia úloh, programovanie, informačné systémy v doprave a v medicíne
E. Bubeníková, Ing.	Teória informácií. Číslkové spracovanie signálov a obrazov. E-learning.
M. Franeková, prof.	Riadenie bezpečnostne kritických procesov. Bezpečnostne relevantný komunikačný systém v aplikáciách riadenia dopravných a priemyselných procesov. Metódy hodnotenia bezpečnosti komunikácie v uzatvorených a otvorených prenosových systémoch. Bezpečnostné kódy a kryptografické techniky.
P. Holečko, Ing. PhD.	Informačné a komunikačné systémy a ich bezpečnosť. Webové technológie. Bezdrôtové senzorové siete. Modelovanie a simulácia.
J. Hrbček, Ing. PhD.	Teória automatického riadenia. Prediktívne riadenie procesov. Programovateľné logické automaty.
A. Janota, prof.	Dopravná telematika. Umelá inteligencia. Senzory.
P. Nagy, Ing.	Bezpečné riadenie dopravy. Železničné zabezpečovacie systémy. Systémy na ochranu osôb a majetku. Informačné systémy.
P. Peniak, Ing. PhD.	Informačné systémy v procesnom riadení. Virtualizácia. Počítačové siete MAN, WAN.
R. Pirník, Ing. PhD.	Komunikačné, informačné a riadiace systémy pre dopravné telematické aplikácie. Legislatívna a normalizačná činnosť v oblasti IDS.
P. Příbyl, prof.	Teória dopravy. Spracovanie dát. Analýza rizík. Inteligentné dopravné systémy.
K. Rástočný, prof.	Analýza rizík a hodnotenie bezpečnosti riadiacich systémov súvisiacich s bezpečnosťou (SRCS) s aplikáciou najmä v železničnej doprave, prípadne v priemysle. Modelovanie funkčných a bezpečnostných vlastností SRCS.
V. Šimák, Ing. PhD.	Jednočipové mikropočítače. Robotika, autonómne roboty. Spracovanie obrazu.
P. Vestenický, doc.	Rádiofrekvenčná identifikácia. Dopravná telematika so zameraním na prepravu nebezpečného tovaru. Počítačové siete. Mikropáskové filtre.
J. Ždánsky, Ing. PhD.	Riadiace systémy na báze PLC a safety PLC s definovanou SIL (Safety Integrity Level) a/alebo s definovanými spoľahlivostnými vlastnosťami.

Tab. 1 Prioritné oblasti vedy a výskumu členov katedry

#### 4. Vedeckovýskumné priority katedry

Vedecká a výskumná orientácia katedry spočíva na troch pilieroch podľa tab. 1. Z uvedených pilierov má katedra najvýraznejšiu tradíciu v oblasti analýzy a syntézy riadiacich a informačných systémov od teoretických modelov až po riešenie aktuálnych projektov praxe, vrátane ich implementácie. V mnohých oblastiach vedy a techniky má KRIS výhradné postavenie v SR; najmä v expertíznej činnosti zameranej na analýzu a syntézu železničných zabezpečovacích systémov.

Oblasť bezpečného prenosu a spracovania informácií pri riadení vybraných bezpečnostne kritických procesov, či už ide o bezpečnostné riadiace systémy v priemysle, zabezpe-

čovacie systémy v doprave, zložité priemyselné technológie alebo bezpečnostné systémy na ochranu osôb a majetku, dáva dostatočný priestor pre výskumné a vývojové aktivity celého kolektívu katedry. Realizácia informačných služieb pre operatívne riadenie s podporou automatizácie a výpočtovej techniky dáva možnosť uplatnenia sa v rozhodujúcich odvetviach národného hospodárstva.

Výber z množstva vedeckovýskumných projektov, ktoré boli v týchto oblastiach výskumu riešené na katedre za ostatných 5 rokov, je uvedený v tab. 3. Výstupy viacerých z nich vyústili do praktických aplikácií a spolu s teoretickými výsledkami boli publikované na mnohých domácich i zahraničných konferenciách a časopisoch. Ich prehľad z priestorových dôvodov nie je uvedený, záujemca ich nájde na stránke EF ŽU [2].

Prioritné smery vedeckovýskumnej činnosti katedry	Cieľový segment národného hospodárstva		
	Železničná doprava	Cestná doprava	Priemysel
<b>Prioritný smer 1<sup>1)</sup></b> Analýza rizika a modelovanie bezpečnostných vlastností riadiacich a prenosových systémov.	Metódy na výpočet ukazovateľov bezpečnosti a objektívne stanovenie zostatkového rizika moderných zabezpečovacích systémov.	Výskum metód na komparatívne i absolútne hodnotenie bezpečnosti telematických objektov dopravnej infraštruktúry (radičov, koncentrátorov, technológií tunelov).	Metódy a postupy na výpočet ukazovateľov bezpečnosti riadiacich, informačných a prenosových systémov pre výrobné procesy.
<b>Prioritný smer 2<sup>2)</sup></b> Formalizácia a modelovanie funkčných vlastností riadiacich systémov.	Metódy a nástroje formálnej špecifikácie a verifikácie funkčných vlastností SRCS. Uplatnenie formálnych i poloformálnych metód (matematické modelovanie, formálna logika a grafický zápis) na analýzu modelov a následnú syntézu riadiaceho systému s definovanými bezpečnostnými a spoľahlivostnými vlastnosťami.		
<b>Prioritný smer 3<sup>3****)</sup></b> Výskum a vývoj v oblasti inteligentných dopravných systémov (IDS)	Posudzovanie a technológie bezpečnejších európskych železničných riadiacich systémov, identifikácii lepších a inteligentnejších riešení. Analýza a syntéza protokolov komunikačného systému s osobitným sortimentom služieb.	Použitie nástrojov Soft-Computing na kvantifikáciu rizika cestných tunelov. Implementácia eSafety do IDS a zvyšovanie kvality služieb inteligentnej dopravy na báze systémov integrovaných dopravných.	Implementácia umelej inteligencie do oblasti riadenia kritických procesov. Výskum metód prediktívneho riadenia vybraných výrobných procesov.
*) <i>Prioritný smer 1</i> Prechod systémov riadenia bezpečnostne kritických procesov na nový technologický			

stupeň si vyžaduje aplikáciu súboru modelov umožňujúci exaktný výpočet rizika pri riadení kritických procesov a výpočet ukazovateľov bezpečnosti riadiacich, informačných a prenosových systémov. Úloha riadenia s pevným koncom je postavená tak, aby integrálnym kritériom optimality bola hodnota akceptovateľného rizika pri fixovaných hodnotách efektívnosti. Výstupy modelovania sú porovnateľné s výsledkami doterajších metód, založených na skúsenostnom princípe. Takéto modely možno použiť pri úlohách analýzy aj syntézy. Prioritný smer garantujú prof. Karol Rástočný, prof. Mária Franeková.

**\*\*\*) Prioritný smer 2** Použitie nových technológií je sprevádzané zvyšovaním zložitosti navrhovaných hardvérových a softvérových riešení. Zložitosť systémov a požiadavka funkčnej bezpečnosti vyvolávajú potrebu nových prístupov k špecifikácii funkčných požiadaviek. V tejto oblasti nachádzajú uplatnenie formálne a poliformálne metódy, ktoré sú založené na matematickom modelovaní, formálnej logike a grafických zápisoch. Jedným z perspektívnych a výrazne sa presadzujúcich grafických jazykov je objektovo orientovaný jazyk UML, ktorý poskytuje široký rozsah prostriedkov nielen na formálnu špecifikáciu, ale aj na formálnu verifikáciu funkčných vlastností riadiacich systémov. Prioritný smer garantujú prof. Rástočný, prof. Janota.

**\*\*\*\*) Prioritný smer 3** Úsilie zamerané na zvyšovanie bezpečnosti, spoľahlivosti a kvality dopravných služieb vyvoláva v oblasti výskumu a vývoja mimoriadny záujem o realizáciu IDS, ktoré môžu riešiť požiadavky spoločnosti na zvyšovanie prepravných výkonov osobnej i nákladnej dopravy, komfortu cestujúcich, znižovanie negatívnych vplyvov na životné prostredie i zvyšovanie bezpečnosti účastníkov dopravy. Základný i aplikovaný výskum v tejto oblasti je zameraný na implementáciu princípov eSafety do IDS. Prioritný smer garantujú prof. Spalek, prof. Janota, prof. Příbyl.

**Tab. 2** Prioritné smery vedeckovýskumnej činnosti katedry

VEGA – 1/0040/08	2008-2010	<i>Matematicko-grafické modelovanie bezpečnostných vlastností bezpečnostne kritických riadiacich systémov</i>	prof. Ing. Karol Rástočný, PhD.
VEGA – 1/0023/08	2008-2010	<i>Teoretický aparát na analýzu a hodnotenie rizík telematických systémov v doprave</i>	prof. Ing. Juraj Spalek, PhD.
APVV VMSP-P-0142-09	2009-2011	<i>Vývoj, výskum a implementácia doplnkovej cestnej signalizácie pre zabezpečenie bezpečnosti na železničných priecestiach a priechodoch pre chodcov</i>	ARDOS, a.s. / prof. Ing. Juraj Spalek, PhD.
ITMS: 26220120028	2009-2011	<i>Centrum excelentnosti pre systémy a služby inteligentnej dopravy I.</i>	prof. Ing. K. Matiaško, PhD. (FRI) / prof. Ing. Juraj Spalek, PhD.
ITMS 26220220021	2009-2012	<i>Centrum translačnej medicíny</i>	UK Bratislava / prof. Ing. Mária Franeková, PhD.
APVV SK-PL-0015-09	2010-2012	<i>Posudzovanie bezpečnosti železničných zabezpečovacích systémov</i>	prof. Ing. Karol Rástočný, PhD.
ITMS: 26220220089	2010-2014	<i>Nové metódy merania fyzikálnych dynamických parametrov a interakcií motorových vozidiel dopravného prúdu a vozovky</i>	prof. Ing. Aleš Janota, PhD.
A-0930-RT-GC	2010-2012	<i>Helicopter fuselage crack monitoring and prognosis through on-board sensor (HECTOR)</i>	doc. Ing. Róbert Hudec, PhD. (KTaM) / Dr. Ing. Peter Vestenický
ITMS: 26220120050	2010-2013	<i>Centrum excelentnosti pre systémy a služby inteligentnej dopravy II.</i>	prof. Ing. K. Matiaško, PhD. (FRI) / prof. Ing. Juraj Spalek, PhD.
VEGA – 1/0392/10	2010-2012	<i>Výskum lokalizácie mobilných uzlov v bezdrôtových senzorických sieťach</i>	Ing. Peter Brída, PhD. (KTaM) / Dr. Ing. Peter Vestenický
ITMS 26110230004	2010-2013	<i>Systematizácia transferu pokrokových technológií a poznatkov medzi priemyselnou sférou a univerzitným prostredím</i>	Ing. Emília Bubeniková
ITMS 26220220089	2010-2014	<i>Nové metódy merania fyzikálnych dynamických parametrov a interakcií motorových vozidiel, dopravného prúdu a vozovky</i>	Betamont, s.r.o./prof. Ing. Aleš Janota, PhD.
VEGA 1/0453/12	2012-2014	<i>Štúdium interakcií motorového vozidla, dopravného prúdu a vozovky</i>	prof. Ing. Aleš Janota, PhD.
VEGA 1/0388/12	2012-2014	<i>Kvantitatívne hodnotenie integrity bezpečnosti riadiacich systémov pre dráhové aplikácie</i>	prof. Ing. Karol Rástočný, PhD.
KEGA 024 ŽU-4/2012	2012-2014	<i>Modernizácia technológií a metód vzdelávania so zameraním na oblasť kryptografie pre bezpečnostne kritické aplikácie</i>	prof. Ing. M. Franeková, PhD.
ITMS 26220220169	2012-2013	<i>Výskumné centrum systémov dopravnej telematiky</i>	Betamont, s.r.o./Ing. Rastislav Pírník, PhD.

**Tab. 3** Významné projekty katedry za ostatných 5 rokov – výber



### 3. Vedeckovýskumná spolupráca s firmami

Významné, akademické i komerčné inštitúcie, ako aj orgány regionálnej, či oblastnej samosprávy, centrálné orgány štátnej správy SR, no najmä centrálné orgány EÚ si uvedomujú, že v čase pretrvávajúcej svetovej hospodárskej a finančne krízy je potrebné investovať milióny eur do rozvoja vzdelanosti a vývoja takých sofistikovaných riešení, ktorých pridaná hodnota sa prejaví v blízkej budúcnosti. Tento tlak sa

prejavuje aj v oblastiach vedeckého a výskumného záujmu katedry a jej komerčných aktivít (tab. 4). Kladie mimoriadne nároky jednak na hľadanie spôsobov vyššej efektívnosti využitia pracovnej kapacity jej vedeckých i technických zamestnancov, ako aj širšieho zapájania nadaných študentov do riešiteľských kolektívov. Často je však nevyhnutné pristúpiť aj k selekcii projektov a zadaní podľa aktuálnych kapacitných možností katedry.

Výber z významných projektov spolupráce s firmami sú v tab. 5.

<b>Aktivita 1</b>	<b>Výskum a vývoj</b>
Oblasti	Analyza a syntéza informačných systémov, moderných zabezpečovacích systémov, inteligentných dopravných systémov i systémov na ochranu osôb a majetku
	Analyza a syntéza protokolov komunikačných systémov s osobitným sortimentom služieb, analýza a syntéza systémov s definovanou úrovňou bezpečnosti
<b>Aktivita 2</b>	<b>Kurzy a vzdelávanie</b>
Oblasti	Železničné zabezpečovacie systémy na báze mikroprocesorovej techniky
	Informačné technológie zamerané na oblasť širokopásmových a multimediálnych sietí, SDH hierarchiu prenosových systémov, bezpečnej komunikácie
	Koncepcia, analýza, syntéza, projektovanie informačných systémov pre dopravné a priemyselné technologické procesy, riešenie konkrétnych problémov železničnej prevádzky so zameraním na železničné zabezpečovacie zariadenia
	Školenia bezpečnostných technikov v oblasti technických prostriedkov ochrany majetku, a hodnotenia bezpečnosti moderných elektronických SRCS
<b>Aktivita 3</b>	<b>Normotvorná činnosť</b>
Oblasti	Systémy na ochranu osôb a majetku (elektrická zabezpečovacia a požiarne signalizácia), sieťové aplikácie, technológie železničnej a cestnej dopravy, železničné zabezpečovacie systémy
<b>Aktivita 4</b>	<b>Projektovanie</b>
Oblasti	Riadiace, informačné a moderné riadiace systémy cestnej dopravy a systémov na ochranu osôb a majetku

Tab. 4 Oblasti komerčných aktivít katedry

Objednávateľ	Obdobie riešenia	Názov projektu	Zodp. riešiteľ
AŽD Praha, s.r.o.	2008	<i>P-103-0002/08 Verifikácia algoritmov IRI.</i>	prof. Ing. Karol Rástočný, PhD.
Kybernetika, s.r.o., Košice	2008	<i>P-103-0009/08 Programovanie PLC automatov so zameraním na safety technológie.</i>	prof. Ing. Karol Rástočný, PhD.
První Signální, a. s., Ostrava,	2008	<i>P-103-0021/08 Posúdenie bezpečnosti systému MODEST-APPOLO</i>	prof. Ing. Karol Rástočný, PhD.
AŽD Praha, s.r.o.	2009	<i>P-103-0010/09 Verifikácia algoritmov IRI – úprava funkcií</i>	prof. Ing. Karol Rástočný, PhD.
PLOTEX, Martin	2009-2010	<i>P-103-0017/09 Posúdenie bezpečnosti počítača osí BO 2.3.</i>	prof. Ing. Karol Rástočný, PhD.
IDSS a.s.	2009-2010	<i>Inteligentné dopravné systémy súčasť vybavenia pozemných komunikácií mesta Martin - Technická štúdia. Žilina.</i>	Ing. P. Londa / Ing. Rastislav Pirník, PhD.
IDSS a.s.	2009-2010	<i>Inteligentné dopravné systémy súčasť vybavenia pozemných komunikácií mesta Ružomberok - Technická štúdia.</i>	Ing. P. Londa / Ing. Rastislav Pirník, PhD.
Scheidt & Bachmann Slovensko s.r.o.	2009-2010	<i>P-103-0021/09 Posúdenie bezpečnosti zariadenie BUES 2000</i>	prof. Ing. Karol Rástočný, PhD.
Mesto Prešov	2009-2010	<i>Inteligentné dopravné systémy - súčasť vybavenia pozemných komunikácií mesta Prešov</i>	prof. Ing. Ján Čelko, PhD. / Ing. Rastislav Pirník, PhD.
VÚD, a.s.	2009-2011	<i>325/1/2009 EASYWAY (Improving Safety and Mobility by Intelligent Network Operations and Traveller Services on the European Road Network)</i>	prof. Ing. Milan Dado, PhD. (KTaM) / prof. Ing. Juraj Spalek, PhD
SIEMENS AG, Östereich	2010	<i>P-103-0010/09 Rozšírenie existujúceho celkového posudku pre elektronické stavidlá ...</i>	prof. Ing. Karol Rástočný, PhD.
SignalBau, a.s., Přerov	2010-2011	<i>Zmluva č. 1/2010 Hodnotenie bezpečnosti traťového zabezpečovacieho zariadenia EAH-08</i>	prof. Ing. Karol Rástočný, PhD.
VÚD, a.s., Žilina	2010-2011	<i>350/2/2011 Analýza metodických a technických podmienok (TP) a návrh nových TP súvisiacich s projektom NSDI.</i>	Ing. Rastislav Pirník, PhD.
VÚD, a.s., Žilina	2010-	<i>2/2010 Interoperabilita Komplexnej služby elektro-</i>	prof. Ing. Juraj Spalek, PhD.

	2011	<i>nického výberu mýta v SR s inými systémami výberu mýta</i>	
AŽD Praha, s.r.o.	2010-2012	<i>P-103-0019/10, P-103-0006/11, P-103-0002/12 Modelovanie funkčného správanía sa RBCC</i>	prof. Ing. Karol Rástočný, PhD.
SIEMENS AG, Östereich	2010-2012	<i>P-103-0015/10, P-103-0012/11, P-103-0016/12 Rozšírenie existujúceho celkového posudku pre elektronické stavadlá na Slovensku</i>	prof. Ing. Karol Rástočný, PhD.
První Signalní, a.s. Ostrava	2012-2013	<i>2/2012 Aktualizácia a rozšírenie produktového portfólia modulov systému NEXUS</i>	prof. Ing. Karol Rástočný, PhD.
VÚD, a.s., Žilina	2013	<i>343/100/2013 Výskum využítia údajov z mýtného systému</i>	prof. Ing. Aleš Janota, PhD.

**Tab. 5 Projekty financované súkromným sektorom za ostatných 5 rokov – výber**

## Záver

Ukazuje sa, že šesťdesiatročná tradícia, odborná erudovanosť i skúsenosti súčasných zamestnancov katedry poskytujú dostatočné záruky na dlhodobé dosahovanie konkurencieschopných výsledkov vo vede a výskume v medzinárodnom meradle, a tiež výchovu kvalitných vysokoškolsky vzdelaných odborníkov, ktorí sa úspešne uplatnia vo všetkých konvenčných oblastiach odboru automatizácia a vďaka vysokej adaptabilite a kreatívnosti aj v príbuzných odboroch.

Podrobnejšie informácie sú na <http://kris.uniza.sk/>.

## Pod'akovanie

*This contribution/publication is the result of the project implementation: Centre of excellence for systems and services of intelligent transport, ITMS 26220120050 supported by the Research & Development Operational Programme funded by the ERDF.*



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

## Literatúra

[1] KRIS-50. Pamätnica k 50. výročiu vzniku katedry, Žilina 2003

[2] Výročné správy Elektrotechnickej fakulty ŽU:  
[http://fel.uniza.sk/index.php?option=com\\_k2&view=item&layout=item&id=86&Itemid=219](http://fel.uniza.sk/index.php?option=com_k2&view=item&layout=item&id=86&Itemid=219)

## Abstract

The paper is dedicated to the 60 anniversary of the Department, whose teaching and scientific research portfolio is oriented to automation. The paper provides a brief history of the Department, its position within the parent University and Faculty, description of three priority areas of research activities, forms of cooperation with academic and commercial institutions both at home and abroad, and an overview of the most important R&D projects solved by members of the Department during recent 5 years.

## Juraj Spalek, prof. Ing. PhD.

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacích a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina

E-mail: [juraj.spalek@fel.uniza.sk](mailto:juraj.spalek@fel.uniza.sk)

# Tvorba 3D modelu a klasifikácia objektov pozemnej komunikácie s využitím odrazivosti laserového lúča

Ján Halgaš, Marián Hruboš, Aleš Janota

## Abstrakt

V článku je prezentovaný návrh systému použitého pre tvorbu 3D modelu pozemnej komunikácie. Meranie prostredia je vykonávané pomocou mobilnej meracej platformy pozostávajúcej z 2D laserového skenera, veľmi presnej pozičnej navigácie (kombinácia GPS a INS), 7 kamier a súboru ďalších senzorov. Laserový skener meria vzdialenosť bodov nachádzajúcich sa v prostredí a navigácia vyhodnocuje aktuálnu pozíciu. V článku sú opísané rovnice potrebné na prepočet nameranej vzdialenosti a polohy do 3D súradníc. Množstvo prepočítaných súradníc vytvára mračno bodov, kde je každý bod určený pomocou trojrozmerných súradníc a hodnoty odrazivosti, ktorú skener umožňuje merať. Body z mračna bodov sú následne klasifikované do jednej zo skupiny objektov nachádzajúcich sa v okolí pozemnej komunikácie (povrch vozovky, fasády budov, stromy, zvislé dopravné značenie a iné). Prezentovaný prístup klasifikácie nameraných bodov predpokladá využitie informácie o priestorovom umiestnení spolu s využitím hodnoty odrazivosti. Výsledok klasifikácie je porovnaný so spôsobom klasifikácie využívajúcim len priestorové umiestnenie skenovaných bodov.

**Kľúčové slová:** 3D model, laserový skener, odrazivosť, klasifikácia

## Úvod

Výskumný projekt, ktorého čiastkové výsledky sú prezentované v tomto článku, nesie názov „Štúdium interakcií motorového vozidla, dopravného toku a vozovky“. Praktické experimenty boli realizované na technickom vybavení získanom v rámci projektu ITMS-26220220089 „Nové metódy merania fyzikálnych dynamických parametrov a interakcií motorových vozidiel, dopravného prúdu a vozovky“, financovaného z európskych štrukturálnych fondov. Pozornosť autorov sa sústreďuje na vytvorenie 3D modelu pozemnej komunikácie, ktorému predchádza rad určitých krokov. Prvotným z nich je zber informácií z prostredia, ktoré chceme previesť do 3D zobrazenia.

Zvolené zariadenia vychádzajú z požiadavky skonštruovania meracej platformy použiteľnej na zber informácií o okolí, samotnej pozemnej komunikácii a dopravnom prúde. Meracia platforma (obr.1) okrem iného pozostáva z laserového 2D skenera s rozsahom 70° určeného na snímanie povrchu vozovky (s menším dosahom, ale väčšou presnosťou merania) a skenera s rozsahom 360° umožňujúceho predovšetkým meranie okolitého priestoru. Sedem kamier vytvára obrazový záznam skenovaného prostredia. Pomocou časovej pečiatky priradenej zaznamenaným dátam je možné skombinovať vizuálny obraz s nameranými dátami. V blízkej budúcnosti sa predpokladá umiestnenie častí obrazových záznamov priamo do 3D modelu.

Základnými jednotkami určenými na zber informácií o okolí sú 2D laserový skener s rozsahom 360°, ktorý meria vzdialenosť medzi hlavou skenera a skenovaným bodom, a presný navigačný systém, ktorý určuje aktuálnu polohu. V článku je predstavený postup použitý pri zbere dát a ich

spracovaní za účelom vytvorenia 3D modelu pozemnej komunikácie s využitím pre inteligentné dopravné systémy.



Obr. 1 Koncept meracieho vozíka, viditeľné sú kamery a obidva skenery

Fig. 1 The concept of the mobile proprietary measurement platform; cameras and both scanners are visible

## 1. Tvorba mračna bodov

Minimálne uhlové rozlíšenie použitého skenera je  $0.125^\circ$  a maximálny dosah 200 metrov. Aby bolo možné vytvoriť 3D obraz prostredia, je poloha tohto skenera určená pomocou navigácie (SPAN - Synchronised Position Attitude Navigation), kombinujúcej globálny pozičný systém (GPS) a inerciálnu navigáciu (INS).

Pre výpočet súradníc jedného bodu sme navrhli vlastnú výpočtovú metódu. Metóda je založená na cyklickom zaznamenávaní meraného rezu priestorom, pričom súradnice jednotlivých meraní sú definované GPS súradnicami. Po zmeraní viacerých rezov je možné vytvoriť výsledný trojrozmerný objekt, zložený z nameraných rezov. Matematické rovnice pre našu výpočtovú metódu vychádzajú z parametrických rovníc kružnice. Naše rovnice sú rozšírené aj o použitie GPS súradníc. Laserový skener poskytuje údaje o meraných bodoch pomocou počiatočného uhla, počtu zmeraných bodov, uhlovej diferencie medzi jednotlivými bodmi a vzdialenosťami medzi laserovým skenerom a meranými bodmi. Uhol, ktorý zvierá meraný bod s osou  $x$  je daný súčtom konštánty, vyjadrujúcej mechanické pootočenie laserového skenera, počiatočného uhla a násobku poradového čísla meraného bodu v danom reze a uhlovej diferencie. Na základe počtu zmeraných bodov je nastavená hodnota opakovania výpočtového cyklu. Výpočet sa opakuje toľko krát, koľko je zmeraných bodov. Poradové číslo meraného bodu je v každom výpočtovom cykle jedného meraného rezu inkrementované [1].

Matematické rovnice pre našu výpočtovú metódu:

$$\begin{aligned} x_{in} &= \left\{ \cos[(\alpha_{0n} + i * \Delta\alpha_n + 90) * \pi * 2 / 360] * d_{in} \right\} \\ y_{in} &= \left\{ -\sin[(\alpha_{0n} + i * \Delta\alpha_n + 90) * \pi * 2 / 360] * d_{in} \right\} \quad (1) \\ z_{in} &= d \end{aligned}$$

kde premenné parametrických rovníc majú nasledujúci význam:

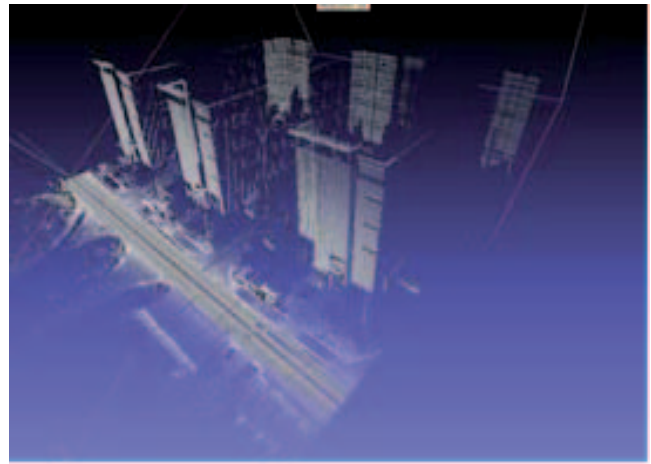
- $x_{in}$ ,  $y_{in}$  a  $z_{in}$  sú súradnice  $i$ -tého meraného bodu v  $n$ -tom reze
- $\alpha_{0n}$  je hodnota počiatočného uhla
- $i$  je poradové číslo aktuálne vypočítavaného bodu
- $\Delta\alpha_n$  je hodnota uhlovej diferencie pre daný rez
- $d_{in}$  je hodnota vzdialenosti  $i$ -tého bodu v  $n$ -tom reze od laserového skenera
- $n$  je poradové číslo meraného rezu, pričom hodnota  $n$  je inkrementovaná pri meraní nového rezu.

Poloha získaná zo SPAN je poloha GPS korigovaná INS. Zaznamenávané dáta sú vo formáte GGA, z ktorého sa extrahuje zemepisná šírka (*lat*) a dĺžka (*long*). Tieto parametre sú pomocou známeho vzorca použité na prepočet posunu  $d$  oproti predchádzajúcemu meraniu.  $R$  je polomer Zeme.

$$\begin{aligned} a &= \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2\left(\frac{\Delta long}{2}\right) \\ c &= 2 \cdot \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right) \quad (3) \end{aligned}$$

$$d = R \cdot c$$

Navrhnutú metódu sme použili pri prepočte dát z reálnych meraní, ktoré je možné ďalej spracovať a vizualizovať (obr. 2).



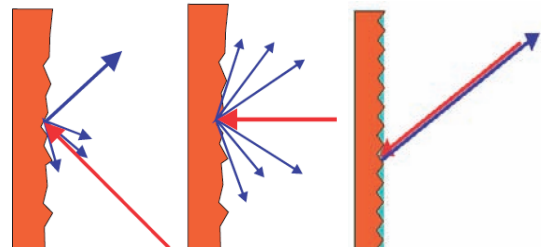
Obr. 2 Vizualizované mračno bodov znázorňujúce budovy a cestu v kampuse univerzity

Fig. 2 A visualized cloud of points representing buildings and the road in the university campus

## 2. Analýza odrazivosti

Použitý 2D skener okrem hodnôt o nameranej vzdialenosti umožňuje merať aj hodnotu intenzity zachyteného lúča (odrazivosť). Signál prijatý od perfektne odrážajúceho bieleho povrchu zodpovedá definícii 100 % spätnej remisie (odrazu alebo rozptylu). Výsledkom tejto definície je, že odrazivosť od svetlo odrážajúcich povrchov (povrchy zrkadiel, odrazníkov) je viac ako 100 %. Väčšina povrchov odráža laserové lúče rozptylom do všetkých smerov (obr. 3).

Závislosť meniaceho sa odrazu laserového lúča je funkciou štruktúry a farby povrchu. Svetlé povrchy odrážajú laserové lúče lepšie ako tmavé povrchy a možno ich detegovať na väčšie vzdialenosti. Na veľmi hrubých (drsnych) povrchoch sa časť energie stráca v dôsledku rozptylu lúčov. To má za následok pokles dosahu skenovania.



Obr. 3 (a) Rozptýlenie lúča na nerovnosti povrchu

(b) rozptyl lúča dopadajúceho kolmo na nerovný povrch

(c) lúč dopadajúci pod uhlom vzhľadom na povrch a kolmo vzhľadom na štruktúru povrchu

Fig. 3 (a) Distraction of laser beam on roughness surface

(b) Distraction of a laser beam falling onto an uneven surface at a right angle

(c) The laser beam falling onto a surface at a certain angle to the surface and at a right angle to the surface structure

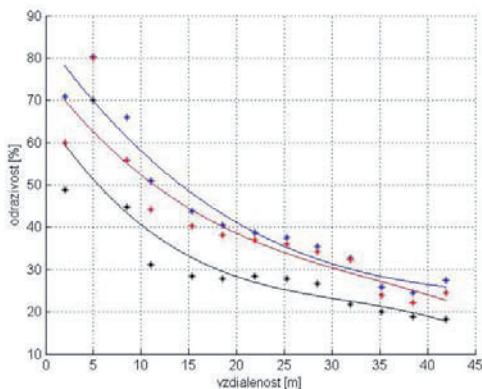
Ak laserový lúč dopadá kolmo na povrch, dochádza k optimálnemu odrazu energie. Ak lúč dopadá pod určitým uhlom, spôsobí to zodpovedajúcu stratu energie a zníženie dosahu skenovania. Objekty menšie ako priemer laserového lúča nemôžu odrážať všetku energiu laserového svetla. Energia tej časti laserového svetla, ktorú nemožno odraziť, je stratená, čo zníži dosah skenovania [2].

## 2.1. Faktory ovplyvňujúce odrazivosť

V nasledujúcej časti je opísaná analýza faktorov, ktoré podľa našich meraní najviac vplyvajú na hodnotu intenzity odrazeného lúča. Medzi uvedené faktory radíme vzdialenosť, uhol dopadu lúča na povrch, štruktúru povrchu a jeho farbu. Pri bežných meraniach každý zo spomenutých faktorov ovplyvňuje konečnú odrazivosť súčasne. Merania, ktoré sme vykonali, boli zamerané na samostatnú analýzu jednotlivých faktorov s čo najväčším potlačením vplyvu ostatných faktorov.

### Vzdialenosť

Pre vyjadrenie vplyvu vzdialenosti na hodnotu odrazivosti sme postupne zvyšovali vzdialenosť objektu, povrch, farba a uhol dopadu boli počas merania konštantné. Z výsledkov merania (obr. 4) vidieť, že aj pre iné farebné povrchy ostala zmena odrazivosti v závislosti od vzdialenosti približne rovnaká. Priemerný pokles odrazivosti, pri ideálnych podmienkach (izbová teplota a úplná viditeľnosť) bol priemerne 0.75% na meter. Tento fakt je možné zapracovať pri vyhodnotení merania a neskoršej klasifikácii. Pokles odrazivosti by bol vyšší pri znížených podmienkach viditeľnosti (hmla, dážď).



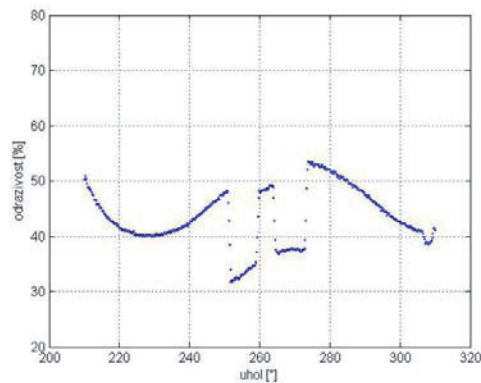
Obr. 4. Hodnota odrazivosti v závislosti od vzdialenosti objektu, modrá – svetlý lesklý povrch, červená – svetlý matný povrch, čierna – tmavý povrch

Fig. 4 Reflection dependency on an object distance; blue – polish bright surface; red – mat bright surface; black - dark surface

### Farba a povrch objektu

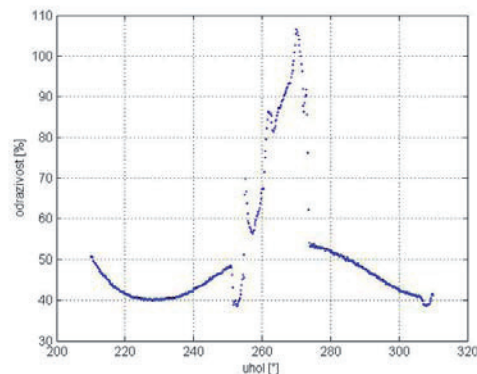
Je všeobecne známe, že rôzne farby odrážajú rôzne množstvo žiarenia. Túto skutočnosť sme overovali meraním, kde sme sledovali vplyv hraničných farebných odtieňov (biela, čierna) na hodnotu odrazivosti. Pri rovnakej štruktúre povrchu sa potvrdil predpoklad, že materiál svetlých odtieňov odrazí väčšie množstvo žiarenia. Pre použitý typ skenera s vlnovou dĺžkou 905 nm predstavuje rozdiel odrazivosti medzi tmavým a svetlým matným povrchom približne 20% (obr. 5). Značný rozdiel v odrazivosti sa prejavil pri lesklom nerezovom plechu (obr. 6), kde pri kolmom dopade (hlava skenera v polohe 270°) hodnota odrazivosti prekročila 100%. Odrazivosť lesklého materiálu pod uhlom dopadu vzdiaľujúcim sa pravému uhlu prudko klesá.

Keďže použitý 2D skener meria pomocou úzkych lúčov, odrazivosť je značne ovplyvnená uhlom, pod ktorým lúč dopadá na povrch a samotnou štruktúrou povrchu



Obr. 5 Príklad nameranej odrazivosti, na bielej stene čierna podložka s bielym pásom

Fig. 5 An example of the reflection on a white wall; a black base with a white stripe



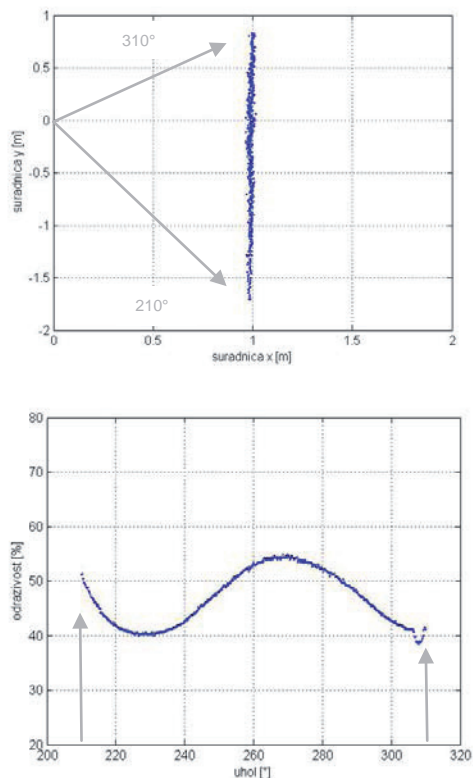
Obr. 6 Príklad nameranej odrazivosti, na bielej stene umiestnený lesklý nerezový plech

Fig. 6 An example of the reflection on a white wall; a polished stainless steel

### Uhol dopadu

Ako bolo naznačené v predchádzajúcej časti, štruktúra povrchu objektu, na ktorý laserový lúč dopadá, značne ovplyvňuje namerané hodnoty odrazivosti. Pre rovný povrch je žiarenie z laserového lúča najslabšie približne pod uhlom 45°. Tento poznatok sa mení v závislosti od nerovnosti povrchu. Keď sú nerovnosti na povrchu väčšie ako priemer dopadajúceho laserového lúča, dochádza k odrazu priamo na nich a tým sa mení uvažovaný uhol dopadu. Pokiaľ sú štruktúra povrchu a nerovnosti na nej porovnateľné s priemerom dopadajúceho lúča, nie je možné určiť uhol, pod ktorým lúč dopadne na povrch, dochádza vtedy k náhodnému rozptylu. Z hodnôt odrazivosti je ale možné určiť stálosť štruktúry povrchu objektu.

Hodnota odrazivosti pre jeden nameraný bod nepodáva postačujúcu informáciu pre stanovenie akéhokoľvek predpokladu. Hodnoty odrazivosti sa menia vzhľadom k uhlu dopadu s určitou závislosťou. Pokiaľ sú štruktúra a tvar povrchu konštantné, s menšími nerovnosťami ako priemer lúča, túto závislosť je možné využiť. Vzťah medzi hodnotami odrazivosti vzhľadom na polohu hlavy skenera sa mení v závislosti od typu povrchu. Pre uvedený príklad (obr. 6) je najvyššia hodnota pri kolmom dopade lúča (hlava skenera v polohe 270°). Spodné hodnoty odrazivosti sú pre tento povrch približne pod uhlom dopadu 60° (hlava skenera v polohe 210°).



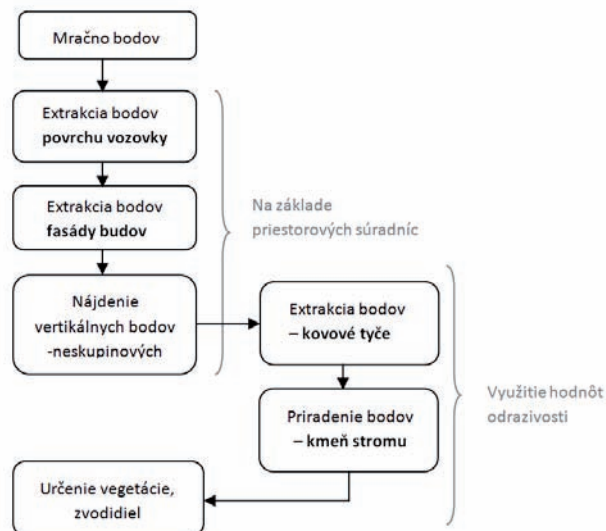
**Obr. 7** 2D Grafická reprezentácia súradníc povrchu skenovanej steny (vyššie), Grafické vyjadrenie odrazivosti, prislúchajúca k nameraným súradniciam, v závislosti na polohe otočnej hlavy skenera (nižšie)

**Fig. 7** The 2D coordinates of a scanned wall (above); reflectivity associated to the measured coordinates, depending on the position of rotary scanner head (below)

### 3. Klasifikácia objektov

Využíva sa niekoľko techník klasifikácie [3], [4], [5], v závislosti od požiadaviek a účelu, pre ktorý bude 3D model použitý. Vytvorené mračno bodov obsahuje informácie o polohách skenovaných bodov, ale nenesie informáciu o tom, ktoré body spoločne vytvárajú objekty. Vytváraná aplikácia bude slúžiť pre potreby inteligentných dopravných systémov. Medzi objekty, o klasifikácii ktorých uvažujeme, patria povrch pozemnej komunikácie, dopravné značenie (vertikálne, horizontálne), zvodidlá, fasády budov a vegetácia. V súčasnosti existuje niekoľko prístupov ku klasifikácii objektov z mračna bodov. Štandardne sa využívajú informácie o polohe objektu spracované z priestorových súradníc. Tento prístup je spoľahlivý na určenie objektu, ktorého poloha vzhľadom k polohe skenera pri snímaní bude zachovaná. Takto na základe informácie o polohe skenovaných bodov vieme jednoznačne určiť povrch pozemnej komunikácie a čiastočne fasády budov.

Klasifikácia mračna bodov prezentovaná v tomto článku využíva ako doplnkovú informáciu aj hodnotu intenzity odrazeného lúča. Klasifikovanie objektov prebieha postupne (obr. 8) od objektov, ktoré majú najväčšie bodové zastúpenie a sú najjednoduchšie. Už zaradené body nie sú v ďalšom procese klasifikácie použité, čo znižuje výpočtovú náročnosť. Uplatnenie doplnkovej informácie o odrazivosti sa preukázalo pri odlišení kmeňa stromu od tyče vertikálneho dopravného značenia a pri odlišení zvodidla od ostatných objektov v blízkom okolí vozovky.

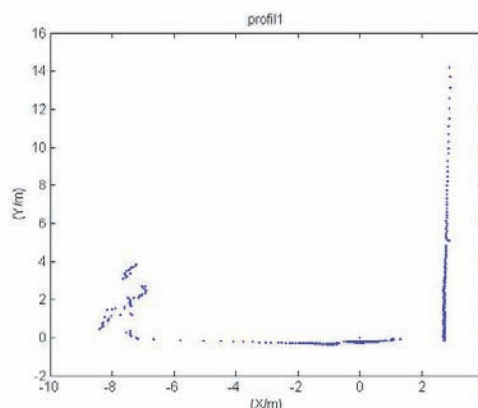


**Obr.8** Postup pri klasifikácii objektov  
**Fig. 8** The procedure for objects classification

#### 3.1. Horizontálne objekty - povrch vozovky

Povrch vozovky má spomedzi objektov v scéne zásadne horizontálny charakter (obr. 9). Tento fakt môžeme aplikovať na histogram skenovaného profilu, kde je vyjadrená početnosť bodov na osi Y. Hrot histogramu odpovedá množstvu bodov ležiacich v jednej rovine, teda povrchu vozovky. Takto je možné klasifikovať všetky body, ktoré susedia so špičkou histogramu. Body predstavujúce povrch sú extrahované do osobitného súboru a už nie sú použité pri ďalšej klasifikácii.

Meraním zameraným na preukázanie vplyvu farby na hodnotu obrazivosti bolo preukázané, že rozdielne farby majú rozdielnu odrazivosť. Kontrast medzi tmavým povrchom vozovky a bielou (reflexnou) farbou použitou na horizontálne dopravné značenie, bude použitý na rozlíšenie horizontálneho dopravného značenia v extrahovanom mračne bodov.



**Obr. 9** Skenovaný profil s bodmi patriacimi (zľava): vegetácii, vozovke, fasáde budovy  
**Fig. 9** Points in a scanned profile represent (left to right): vegetation, roads, building's facade

#### 3.2. Vertikálne objekty

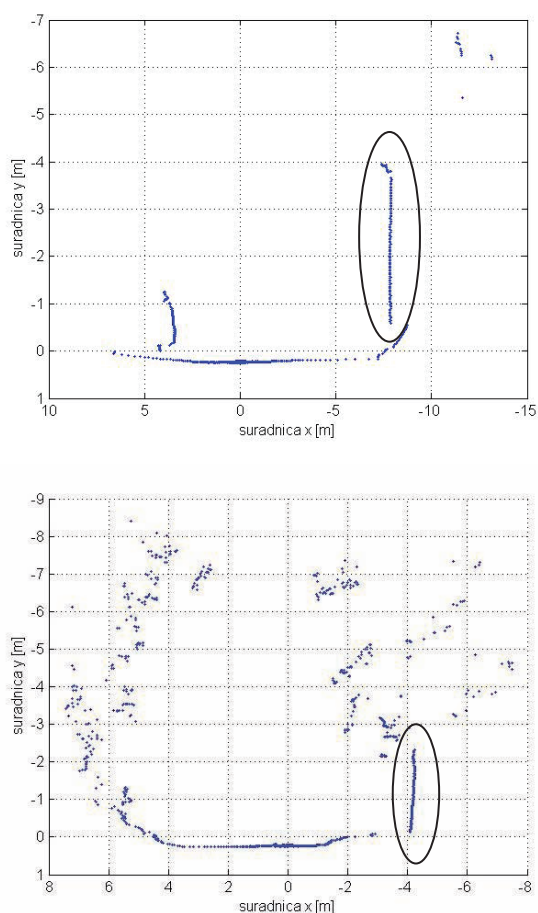
Medzi vertikálne objekty uvažované v okolí pozemnej komunikácie radíme fasády budov, stožiare lúč, vertikálne dopravné značenie, ale aj kmene stromov. Sú to objekty, pri ktorých predpokladáme, že sú tvorené bodmi, ktoré majú medzi sebou vertikálny charakter.

## Fasády budov

Podobný postup je aplikovaný na ostávajúce body v profile, podľa hodnôt vzdialeností na osi  $x$  sa vytvorí histogram bodov, ktorý umožní klasifikovať ostatné objekty v scéne (obr. 9). Pre fasády budov vo všeobecnosti platí vertikálny charakter s vysokými hodnotami výskytu bodov v malom intervale. Ak sa táto skutočnosť opakuje v rovnakej  $x$ -ovej súradnici aj pre niekoľko susedných skenovaných profilov, hovorí to o pokračovaní vertikálnej roviny, ktorej body môžeme klasifikovať ako fasádu budovy.

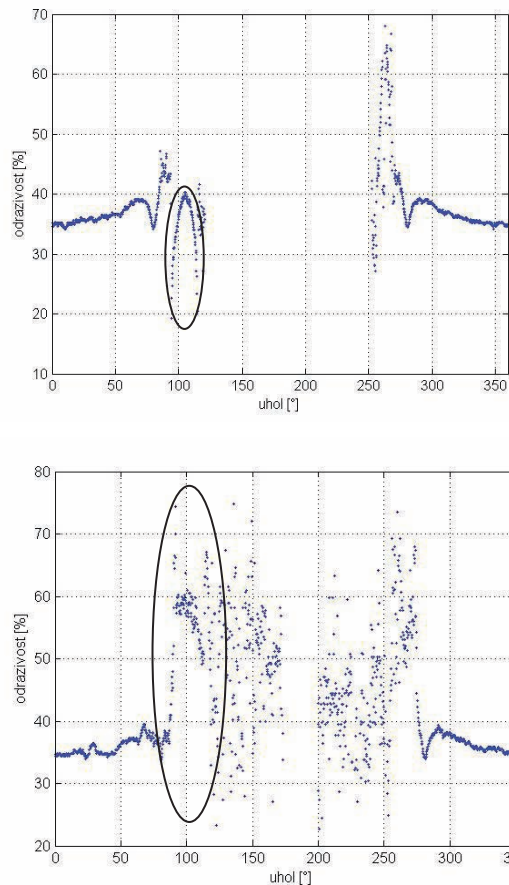
## Horizontálne dopravné značenie

Na ilustráciu porovnávame skenované profily (obr. 10). Vyznačené sú body odpovedajúce kmeňu stromu a stožiaru dopravného značenia. Pri presnosti metódy laserového skenovania *Time of flight* rádovo centimetre je ťažké určiť na základe algoritmu založenom len na spracovaní priestorových súradníc, o ktorý z uvažovaných vertikálnych objektov ide. Spracovaním informácie o odrazivosti v závislosti od uhla nie je tiež možné presne stanoviť, o aký typ povrchu ide, ale na základe poznatkov uvedených v časti 2.1 vieme povedať, že ide o objekt s rovnakými povrchovými vlastnosťami (obr. 11). Skombinovaním s informáciou o vertikálnom charaktere vieme zaradiť skupinu bodov odpovedajúcu vyznačenej parabole medzi vertikálne dopravné značenie.



**Obr. 10** Skenovaný profil s vyznačeným dopravným značením (vyššie), kmeňom stromu (nižšie)

**Fig. 10** Scanned profile with marked road signs (above); a marked tree trunk (below)



**Obr. 11** Odrazivosť bodov v závislosti od polohy hlavy skenera, vyznačené body odpovedajúce tyči dopravného značenia (vyššie), kmeňu stromu (nižšie)

**Fig. 11** Reflectivity of scanned points depends on the position of the scanner head; marked reflectivity of road signs (above), marked reflectivity of a tree trunk (below)

Klasifikácia na základe odrazivosti je automatizovane vykonávaná prechádzaním skenovaných profilov. Body sú aproximované a následne je aproximovaná krivka vyjadrená polynómom 2. stupňa:

$$y = p_1x^n + p_2x^{n-1} + \dots + p_nx + p_{n+1} \quad (3)$$

Na základe výsledku polynómu pre  $x=1$  je objekt priradený do skupiny vertikálneho dopravného značenia.

## 3.3. Vegetácia

Pod vegetáciou rozumieme koruny stromov a kríky v okolí vozovky. Vegetácia je klasifikovaná v poslednom stupni, kde z pôvodného mračna bodov zostali len body, ktoré nespĺnili podmienky do zaradenia ani do jednej z klasifikovaných skupín. Pri klasifikácii využívame rozptýlenosť hodnôt odrazivosti, ktorá je spôsobená rôznym postavením listov.

V rámci klasifikácie vegetácie je možné určiť aj body, ktoré pripadajú zvodidlu. Tu sa znovu využíva informácia o odrazivosti, ktorá má pre skupinu bodov v profile odpovedajúcu zvodidlám so stálym povrchom a rovnakým tvarom známy charakter a je možné ju vyhľadať. Pre zníženie výpočtovej náročnosti sa vyhľadáva len v priestoroch, kde sa zvodidla môžu reálne nachádzať, t. j. 0.2÷1 m od povrchu vozovky.

Postupy prezentované v tomto príspevku naznačujú náš prístup k tvorbe 3D modelu na základe meraných dát zo zariadení primárne určených na iné použitie. Využitie 2D laserového skenera a pozičnej navigácie si vyžadovalo tvorbu vlastnej metódy, na základe ktorej sme boli schopní namerané dáta prepočítať do súradníc v trojrozmernom priestore. Vytvorený 3D model chceme využiť pre potreby inteligentných dopravných systémov, kde ako prvý krok je potrebné automatizovane klasifikovať mračno bodov. Na rozdiel od bežne používaných algoritmov klasifikácie bola použitá aj hodnota odrazivosti, ktorá pomohla určiť typ objektu, v prípadoch, kde len informácia o priestorovom umiestnení nie je postačujúca. Klasifikované body sú rozdelené do súborov podľa príslušnosti k určovanému objektu. Takéto súbory môžu byť ďalej spracované s využitím informácií napríklad o umiestnení vertikálneho dopravného značenia. Po porovnaní s vytvorenou databázou by program vedel určiť, v ktorých častiach 3D modelu dopravná značka nebola nájdená. Predpokladá sa využitie záznamov z kamier inštalovaných na meracej platforme, na ktorých by bolo možné problematické objekty dodatočne skontrolovať na základe časovej pečiatky.

### PodĎakovanie

Článok bol vypracovaný s podporou Slovenskej grantovej agentúry VEGA, v rámci projektu č. 1/0453/12 „Štúdium interakcií motorového vozidla, dopravného prúdu a vozovky“.

### Literatúra

- [1] HRUBOŠ, M.: Nástroj na zistenie stavu degradácie vozovky v čase (diplomová práca). KRIS EF ŽU v Žiline, 78 s., 2012
- [2] LD-OEM1000 to 5100 Laser Measurement System. Operating Instructions. SICK AG - Division Auto Ident - Germany, dokument číslo 8011504/2009-01-15, 98 s.
- [3] ABUHADROUS, I., AMMOUN, S., NASHASHIBI, F., GOULETTE, F., LAURGEAU, C.: Digitizing and 3D modeling of urban environments and roads using vehicle-borne laser scanner system. Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, September 28- October 2, 2004, s. 76-81
- [4] HUIJING, Z., YUZHONG, Ch., RYOSUKE, S.: An efficient extrinsic calibration of a multiple laser scanners and cameras' sensor system on a mobile platform. Proc. of the 2007 IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, June 13-15, 2007, s. 422-427
- [5] YOKOHAMA, H., DATE, H., KANAI, S., TAKEDA, H.: Pole-like objects recognition from mobile laser scanning data using smoothing and principal component analysis. Int. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vo. XXXVIII-5/W12, 2011, ISPRS Calgary 2011 Workshop, Calgary, Canada, August 29-31, 2011, s. 115-120

### Abstract

This paper presents an approach adopted by the authors to design a system used for creating of a 3D model of road communication. Measurements of the environment have been realised using the mobile proprietary measurement platform that consists of two laser measurement systems (2D scanners), high accuracy position navigation (SPAN - Synchronised Position Attitude Navigation), 7 cameras and in future also a set of different sensors. The laser scanner captures raw data of the environment and SPAN estimates an actual position. The paper describes formulas designed to process a measured distance and position into the 3D coordinate system. Processed data creates a cloud of points where each scanned point is represented by three space coordinates and echo value as well. Scanned points are consequently classified to one of groups of objects found in the road environment (walls, road surface, trees, vertical road signs etc.) using the described method. The presented approach supposes utilization of the point placement information and the echo value information to classify measured points. Obtained results are compared with results obtained by a classification method working without reflexion considerations.

### Ing. Ján Halgaš

Žilinská univerzita  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
Tel.: +421 41 513 3321  
Email: jan.halgas@fel.uniza.sk

### Ing. Marián Hruboš

Žilinská univerzita  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
Tel.: +421 41 513 3307  
Email: marian.hrubos@fel.uniza.sk

### prof. Ing. Aleš Janota, PhD., Eurlng.

Žilinská univerzita  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
Tel.: +421 41 513 3356  
Fax: +421-41 513 1515  
Email: ales.janota@fel.uniza.sk



# Globální a mentální modely dopravy v městských aglomeracích

Pavel Příbyl, Ondřej Příbyl

## Abstrakt

Přestože se dopravní parametry měří na mnoha místech dopravní sítě, není lehkou úlohou interpretovat tato data do srozumitelné podoby. Data využívaná pro lokální řízení jsou ukládána v databázích, ale jejich další využití je omezeno. Dispečerů dopravy spíše rozhodují na základě CCTV kamer, než by využívali dopravní data. Předmětem tohoto příspěvku je návrh nástroje pro lepší porozumění dopravním datům a to i nad rozsáhlými dopravními sítěmi. K tomu účelu je zaveden pojem globální model. Nejdříve je ale představena hierarchická architektura městského řídicího systému, který je schopen řešit stacionární i nestacionární dopravní stavy a to i využitím umělé inteligence. Dále je popsán způsob vytvoření globálního dopravního modelu nad dopravní sítí, jehož podstatou je redukce rozměru dat z hlediska topologie sítě i z hlediska jejich dimenze a následná interpretace ve srozumitelné podobě pro dispečery.

**Klíčová slova:** dopravní data, řízení dopravy, umělá inteligence, dimenze dat, redukce rozsahu dat

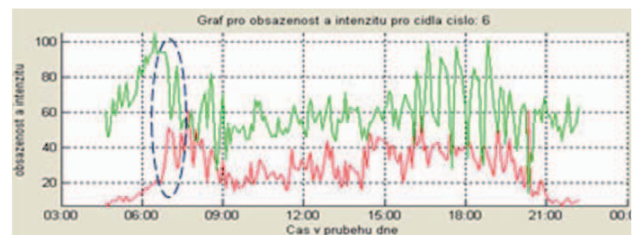
## Motto úvah

Každé větší město je v rámci rozvoje dopravních systémů postupně vybavováno detektory situovanými na rozhodujících místech dopravní sítě a detektory na křižovatkách. Zpracování dopravních dat se pro potřeby lokálního řízení většinou odehrává v místě, například na úrovni jednoho řadiče. Měřená data z detektorů jsou zároveň přenášena na nadřazenou (hlavní) dopravní ústřednu, která může mít různé formy, ale obvykle je obsluhována lidmi. Ti mají za úkol sledovat dopravu, resp. její kvalitu a případně činit potřebné zásahy.

Data z detektorů jsou ve formě časových řad, viz obr. 1, a lze je velmi těžko přímo interpretovat na ordinální stupnici vyjadřující kvalitu dopravy, například v zavedené stupnici 1 až 5 (readability problem). Kromě toho se obvykle jedná o desítky měření a lidská obsluha není schopna nejenom všechny hodnoty interpretovat a už vůbec ne odhalit skryté vazby mezi nimi. Vzniká tak známý problém rozsahu (hugeness problem). Při pohledu na obrázek je zřejmé, že nemusí být jednoznačné, jak může tato data interpretovat lidská obsluha, dokonce lze mít za to, že různí dispečerů „vidí“ kvalitu dopravy vyjádřenou uvedenými časovými řadami různě a obtížně by se shodli třeba i na definici kongesce, což by pak vedlo k rozličným řídicím zásahům.

Lze konstatovat, že systémy pro vyhodnocování dopravních dat jsou v rozsáhlých telematických systémech velmi často nedostatečné, a rozsahu sítě neadekvátní. Chybí modely zprostředkovávající globální pohled na dopravní situaci ve městě, které jsou příbuzné lidskému myšlení a lidskému chápání dopravy.

Pro zobrazení stavu dopravy na strategických místech dopravní sítě se často používají jednoduché stavové automaty vyhodnocující jen jeden dopravní parametr - rychlost dopravy, a z ní odvozující její kvalitu v hrubé stupnici odpovídající verbálním výrazům: dobrá/na mezi stability/kongesce. Podobný systém je aplikován i v Praze, který interpretuje kvalitu dopravy ve formě tříbarevné stupnice založené pouze na měření průměrné rychlosti, což je zjednodušení, až na samé hranici přijatelnosti.



Obr. 1 Časová řada intenzity dopravy a obsazenosti

Fig. 1 Time series of traffic flow and occupancy

## 1. Architektura moderního systému řízení města

Základem jakéhokoli řízení dopravy je nutnost dokonale poznat procesy, které se mají řídit. Vzhledem k tomu, že není reálné zkoumat řízení přímo v ulicích je nutno zpracovat kvalitní modely dopravy [1]. Modelů existuje několik typů, každý z nich je ale do značné míry abstraktní představou o zkoumaném procesu/systému. Rozlišujeme makroskopické modely, které pracují s agregovanými dopravními veličinami a mohou například optimalizovat řízení křižovatek v celé oblasti (TRANSYT). Stále lepší a kvalitnější modelování přináší mikroskopické modely, které pracují s každým vozidlem jednotlivě a jsou schopny modelovat i individuální chování řidičů. V této kategorii modelů je rozšířen AIMSUN či VISSIM. Trend dnešní doby je v kombinaci výhod obou výše popsaných typů v takzvaných mesoskopických modelech. Modely těchto kategorií nejsou předmětem tohoto článku, ale musí být brány do úvahy při jakémkoli návrhu managementu města. Ve městech je doporučena tříúrovňová hierarchie řídicího systému: (1) dopravní řadiče a senzory jsou na úrovni infrastruktury komunikací a lokálně řídí daný uzel, (2) oblastní řídicí ústředny pracující v zásadě automaticky ovládají řadiče v dopravně vymezené oblasti a na (3) nejvyšší úrovni dopravního útvaru pracuje řídicí ústředna integrující všechny subsystémy a řešící mimořádné situace za účasti expertů. Obr. 2 vyjadřuje nejenom zá-

kladní strukturu moderního řídicího systému orientovaného na budoucnost a stále vyšší dopravní nároky, ale popisuje i některé procesy. V obrázku jsou naznačeny dva zásadně odlišné dopravní stavy, které je nutné řešit i jinými technickými prostředky:

1. **Stacionární dopravní stavy (znázorněny modřevlevo):** v tomto případě jsou měřené dopravní parametry (obvykle intenzita a obsazenost či rychlost) v zásadě ve shodě s dopravním modelem, na který jsou nastaveny řídicí algoritmy dopravní ústředny.
2. **Nestacionární dopravní stavy (znázorněny růžově-vpravo):** nazývané také mimořádnými dopravními situacemi. Typicky se jedná o nehody, mimořádné kongesce, dílčí uzavření tunelu apod. Pro řešení situací, detekovaných detektory lze využít databázi předpřipravených scénářů.

### 1.1 Stacionární dopravní stavy na úrovni oblasti

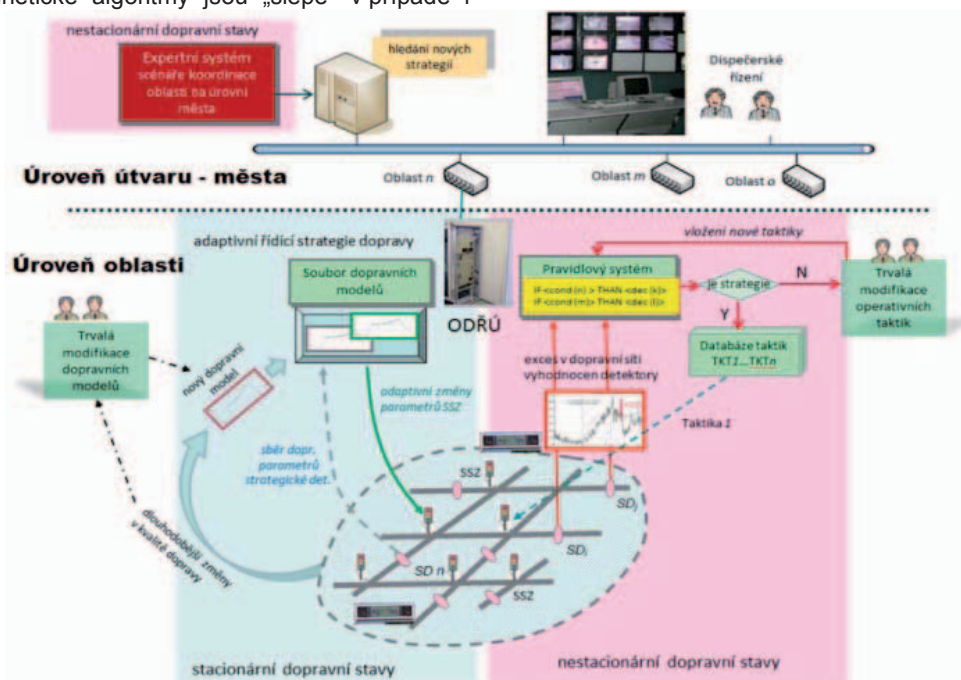
V tomto hierarchickém systému na nejnižší úrovni pracují **světelná signalizační zařízení (SSZ)**. Z rozboru literatury vyplývá, že v případě zatížených městských sítí nejsou řešené využívající principy umělé inteligence, jako jsou fuzzy technologie, či neuronové sítě a genetické algoritmy velkým přínosem a to speciálně pro křižovatky v dopravní síti s vazbami na sousední křižovatky nebo pro křižovatky s významnými dopravními zátěžemi ze všech směrů [2], [3], [4], [5]. Fuzzy řízení, v porovnání s dynamickým, trpí na tom, že má jisté zpoždění, neboť pracuje s integrálními veličinami (délka kolon, predikce intenzit apod.) [6]. Neuronové sítě a genetické algoritmy jsou „slepé“ v případě i

menších změn v dopravě a je vždy nutné je přeučovat na nové podmínky.

Jedinou výjimkou, kdy lze využít výše uvedených principů řízení, které nejsou dosud používány v praxi, jsou SSZ s nerovnoměrnou zátěží, kdy lze dílče přerozdělovat signál Volno na základě délek kolon tzv. split optimalizací. Optimální dopravní řadič v zatížené dopravní síti by měl:

1. Mít implementováno to nejlepší **dopravně závislé řízení** nalažené na místní podmínky, včetně vazeb na MHD a chodce. Toto řízení umožňuje v sekundovém rastru řídit dopravní proudy podle binární logiky.
2. Umožňovat „**split**“ **optimalizaci**, neboli mít možnosti případně přerozdělovat lokálně délky zelených v rámci jednoho či několika cyklů tak, aby se vyprazdňovaly kolony na úkor vjezdů, kde časy zdržení nejsou významné.
3. Přijímat a **implementovat okrajové podmínky pro řízení:** dobu cyklu  $T_c$  a ofset  $T_{off}$  a tak adaptivně měnit vnitřní řídicí proměnné uzlu, dle podmínek v celé síti.

Na druhé hierarchické úrovni pracuje v automatickém režimu počítač ovládající skupinu řadičů. Oblast musí být vybírána na základě dopravně-inženýrských výpočtů induktivně, či deduktivně s využitím expertních odhadů. Má tvořit co nejkompaktnější celek se slabými vazbami na okolí. Počítače na této úrovni tvoří oblastní dopravní ústředny. V tomto místě je soustředěna strategie řízení stacionárních dopravních stavů, které byly připraveny off-line a podle dopravní situace se z nich vybírají strategie vhodné pro danou situaci.

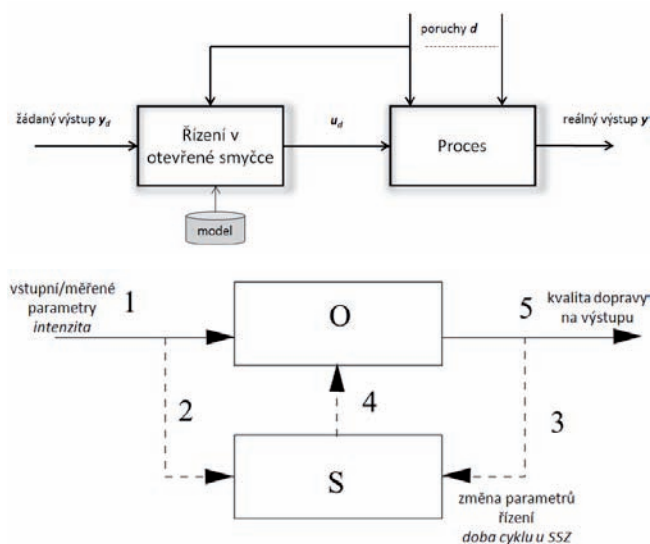


Obr. 2 Obecná struktura městského řídicího systému

Fig. 2 Generic structure of urban traffic control system

V zásadě se může jednat o dva typy řízení [7]:

1. **Řízení výběrem připravených programů:** jedná se o obdobu dopravně závislého řízení, kdy je připravena sestava programů a z nich se, dle měřených parametrů vybere program, který je dané situaci nejbližší. Jedná se tedy o diskretní změnu okrajových podmínek implementovanou dopravními řadiči.
2. **Adaptivní řízení:** v tomto případě oblastní ústředna, na základě měřených dopravních dat, inkrementálně mění okrajové podmínky dopravních řadičů. V tomto případě se pracuje s jedním globálním programem, jehož parametry se mění dle zatížení sítě.



Obr. 3 Řízení dle modelu (nahore) a řízení adaptivní (dole)

Fig. 3 Model based control (abowe) and adaptive control (below)

Na tomto místě není zcela jednoznačné, dát některému ze systémů přednost. Výhodou prvního je, že je dobře „čitelný“, což znamená, že se v každém okamžiku rozliší, jaký program je implementován. V případě, že je předem připraven dostatečný soubor programů, může být i pokrytí většiny dopravních situací dostatečně jemné. Takovýto systém však nereaguje na dopravní stavy, které nejsou předem připraveny v modelu, viz obr. 3, neboť nemá žádnou zpětnou vazbu.

Teoreticky je výhodnější, a z hlediska budoucnosti určitě, adaptivní řízení, kdy se mění řídicí parametry v bloku S dle dopravního zatížení. Jednou z nevýhod může být „nečitelnost“ vnitřních optimalizačních programů a také to, že se obvykle ladí delší dobu. Další nevýhodou je, že řídicí systém pracuje v jistých okrajových podmínkách a pokud jsou překročeny, nemůže řídicí systém S reagovat.

### 1.2 Stacionární dopravní stavy – mimo rozsah modelu

Ve výše uvedených případech byl základní předpoklad v tom, že je znám dopravní model vycházející z off-line měření dopravních parametrů. V reálu jsou spolehlivě měřeny dopravní parametry, které se porovnávají s parametry modelu, a systém vybírá vhodné programy, či mění řídicí parametry podle podobnosti reálu a modelu.

Doprava je ovšem dynamická, jsou měněny přepravní vztahy, staví se nové objekty apod. V těchto případech dochází k dlouhodobější změně vztahů v dopravní síti a doprava se pohybuje mimo okrajové podmínky, které byly základem pro primární návrh. Přitom se stále jedná o stacionární stavy s dlouhodobějším horizontem změn, takzvané sezónní změny. Tyto situace jsou znázorněny v Obr. 2 šipkou ukazující, že je nutné doplnit řízení oblasti o nové dopravní situace. Tím by se mělo zabývat dopravně inženýrské pracoviště znázorněné v obrázku jako skupina osob vytvářející dopravní modely pro postupně se měnící dopravní stavy.

### 1.3 Nestacionární dopravní stavy na úrovni oblasti – mimořádné události

V dopravní síti vznikají také nepravidelně nestacionární děje. Těmi jsou myšleny především dopravní nehody, či náhlé tvorby kolon v různých místech a z různých příčin. Tyto stavy, které vyvolávají nestacionaritu v dopravě nelze dobře predikovat. Pokud se však, víceméně náhodně, vyskytují ve stejných místech, lze připravit dopravní scénáře, které jejich negativní působení na dopravu omezí. Pro dopravní excesy opakující se na „známých“ místech, které lze identifikovat dopravními detektory lze připravit algoritmo-

vané řešení vycházející z poznatků, jak by tuto situaci řešil nejlepší expert.

Dobrý a moderní řídicí systém by měl postupně eliminovat ruční zásahy operátorů do řízení a stále více se spoléhat na automatické reakce systému na oblastní úrovni dle předem daných a promyšlených scénářů. V praxi to vystihuje schéma v pravé polovině citovaného obrázku. Dále popsaný přístup lze využít, pokud se zná problém (nehoda na Jirákově mostě, ve směru ...), dopravní senzory jsou schopny ho identifikovat a je reálně najít scénář, který pomůže danou situaci řešit.

Pro automatizaci těchto procesů se nejlépe hodí pravidlový expertní systém, jehož cílem je nahradit výpočetní technikou rozhodování expertní skupiny [8], [9]. Takový systém je vybaven souborem scénářů pro automatizované ovládání dopravy, které předem připravili experti. Ti je mohou připravit dvěma přístupy:

- deduktivním: kdy jsou sesbírány zkušenosti, jak řešit konkrétní dopravní problém, skupinou expertů. Ti pak stanoví vstupní/měřené parametry a pravidla jak má reagovat řídicí systém. Protože se jedná o soubor logických podmínek, lze poměrně dobře hledat a diskutovat případné chyby;
- induktivním: řešení analytické, které by bylo příliš nákladné, se nahrazuje mikrosimulacemi, které poměrně přesně vyjadřují chování systému. Pravidla vytváří opět experti na základě simulací. Nákladnější je naprogramovat i expertní pravidla jako nadstavbu nad simulačními programy, což umožňuje populární AIMSUN i VISSIM, a sledovat přínos expertního řízení.

Pravidla jsou tedy jednoznačně deklarovaná a tím i kontrolovatelná ve formě logických rovnic:

*POKUD <podmínka (n1) & <podmínka (n8) > POTOM <rozhodnutí (k0)>*

*POKUD <podmínka (m1) & <podmínka (n1) & <podmínka (n8) > POTOM <rozhodnutí (k6)>*

Pro jednu situaci se může uplatnit i více logických rovnic, pro přehlednost uspořádaných v tabulkách. Tabulky mohou obsahovat až do cca 30 pravidel, aby byla pravidla ještě kontrolovatelná. S každou novou mimořádnou událostí, u které je pravděpodobné, že se bude v budoucnosti opakovat se stejnými příznaky a ve stejném místě je vhodné/nutné najít novou řídicí strategii, kterou opět vytváří experti znalí dopravní problematiky a znalí možností řídicích systémů. Nový scénář se vkládá do databáze (DB pravidlového systému) a příště na vznik této situace reaguje systém automaticky. Velmi důležitou součástí expertního řízení je jejich testování a následná verifikace na dostatečně velkých souborech dat.

### 1.4 Úroveň útvaru – celoměstský systém

V případě tří-úrovňové hierarchie řízení je na nejvyšší úrovni centrální řídicí systém, kde již hraje i významnou roli člověk - dispečer či operátor. Toto uspořádání je ve shodě s teorií ovládání komplexních technologických systémů.

Tato nejvyšší vrstva monitoruje dopravu nad všemi oblastmi a integruje další subsystemy. Jejím hlavním smyslem je dohlížet nad správnou činností oblastí a reagovat na mimořádné situace, které přesahují možnosti oblastního řízení. Znamená to, že i na této úrovni je expertní systém, který za pomoci scénářů koordinuje dopravní oblasti, pro něž je to jediná možná komunikace. Typicky lze například za pomoci předem známých scénářů ovládat dopravu koordinovat různé oblasti v případě uzavírání tunelů, ať již z důvodů nehod či pravidelné údržby. Dalším příkladem může být závažný exces v jedné oblasti, který má dopad na další oblasti.

Na této nejvyšší úrovni, jako na jediné, operuje i člověk. Hlavním smyslem jeho činnosti by mělo být monitorování dopravy a „porozumění“ jejímu vývoji a změnám a to proto, aby byl dispečer schopen reagovat odpovídajícím způsobem na mimořádnou událost, která ještě není v databázi taktik.

Vždy by se mělo jednat o dílčí zásahy, které neovlivní dopravu v širším kontextu. Je nutné vycházet ze zásady, že lidské vnímání situace nemusí vždy zohledňovat současný a budoucí stav a lidská reakce může vnést do celého systému nestabilitu.

Jak ale bylo zdůrazněno v úvodu, interpretace informací z množiny detektorů, které jsou poskytovány v dopravní ústředně, člověkem-dispečerem, není prakticky možná. I významné zjednodušení, kdy dispečer sledují pouze jeden parametr, jako je rychlost, nevede k tomu, aby se obraz dopravy posoudil nad danou oblastí jako celek a již vůbec se není možné bavit o predikování vývoje dopravní situace v časovém horizontu několika desítek minut. V následující kapitole je uvedena myšlenka tvorby globálního modelu, který „pokrývá“ danou oblast a poskytuje informace o globální úrovni kvality dopravy. Je zde i zmíněn mentální model, který by měl souviset s vnímáním dopravy tak, jak jí vnímá člověk. Tento nový pohled není v současné době rozpracován do většího detailu, ale poskytuje výchozí bod pro další výzkum.

## 2. Podstata tvorby globálního modelu dopravy ve městě

Dopravní data v libovolné topologii dopravní sítě vytváří vícedimenzionální časo-prostorově proměnný vektor, který není člověk schopen v jeho komplexnosti interpretovat ve formě hodnocení kvality dopravy. Proto je nutné využít matematických metod redukce dimenze prostoru, ve kterém jsou měřena dopravní data a v dalším kroku i transformaci měřených hodnot dopravních parametrů, obvykle intenzity dopravy a rychlosti, na ordinální hodnoty kvality dopravy v podobě, kterou je pozorovatel schopen interpretovat jako číselnou stupnici 1-5. Tyto hodnoty kvality dopravy jsou také srozumitelné pro příjemce informací, tedy pro řidiče.

Jakýkoli automaticky pracující klasifikátor hodnotí dopravu a její kvalitu v jednom místě. Takovýchto míst jsou v městské síti desítky, někdy až stovky. Jedním z důležitých sdělení, které obsahuje tato kapitola je, že je nutné se na dopravu ve městě, či ve vymezené dopravní síti dívat jako na celek. Základní myšlenkou je vytvoření **globálního modelu** komplexní městské sítě využívajícího okamžité hodnocení dopravy v každém z měřicích míst ke stanovení obrazu dopravy nad celou sítí. Tento obraz stavu dopravy je ale časově proměnný, neboť doprava je dynamický systém a měřené hodnoty se získávají v ekvidistančních časových krocích. Obraz dopravy, resp. její kvalita nabývá určitého konečného množství stavů. V tomto smyslu lze definovat stavový prostor dopravy jako prostor, kde je kvalita dopravy vyjádřena vektorem. Tento vektor se ve stavovém prostoru pohybuje v závislosti na vstupních proměnných, stavových proměnných a vnitřních vazbách. V současné době se stavová teorie používá hlavně v teorii řízení.

### 2.1 Principy tvorby globálního modelu

Vzhledem k novosti navrženého postupu, u kterého nejsou známy praktické aplikace, je vhodné postupovat po krocích a vytvářet globální model vždy pro jisté dopravní sub-oblasti, které mohou souhlasit s rozdělením působnosti oblastní řídicí ústředny. Globální model oblasti dává dispečerovi přehled o obrazu chování dopravy v takovéto oblasti. V tomto logickém kroku by byla tedy celá městská oblast dekomponována na sub-oblasti. Globální model je vytvářen ve třech či čtyřech krocích:

1. Redukce korelovaných detektorů do virtuálního detektoru – tento krok snižuje množství měřených dat a nahrazuje velké skupiny detektorů několika virtuálními detektory, bez podstatné ztráty informací o dopravě.
2. Redukce dvou měřených parametrů do jedné hodnoty – např. intenzita a obsazenost je transformována do hodnot ukazatele kvality dopravy (LOS – Level Of Service); tím se dva těžko interpretovatelné časové parametry transformují na lidsky srozumitelnější nominální hodnotu a dané stupnici.
3. Redukce množiny  $i$ -LOS dané oblasti do jedné výsledné LOS hodnotící obraz dopravy nad oblastí; dílčí hodnocení mohou být/a nemusí nahrazena sumárním hodnocením.
4. Vizualizace obrazu dopravy jako mentální mapa.

**1. krok řešení:** dopravu lze ve stacionárním/řádném stavu charakterizovat jako homogenní a obvykle lze najít skupiny detektorů, pro které jsou měřená data nějakým způsobem korelována. Proto lze obvykle větší či menší skupinu detektorů nahradit jedním nebo několika detektory virtuálními, jejichž informační obsah dostatečně vypovídá o obrazu dopravy v oblasti.

Matematické metody redukce dat jsou novým moderním přístupem v analýze velkých objemů dat poskytovaných dopravními senzory. Matematicky to znamená provést proces transformace množiny  $S$  detektorů do jedno či málo prvkové množiny  $J$ , kdy platí  $S \gg J$ . Doprava je stále charakterizována časovými řadami intenzity a obsazenosti, ale snížil se významně počet zpracovávaných hodnot, platí

$$\begin{aligned} 1.krok \quad q^S(t) &\xrightarrow{\otimes} q^J(t) \\ \kappa^S(t) &\xrightarrow{\otimes} \kappa^J(t) \end{aligned} \quad (1)$$

Pro redukci počtu detektorů je výhodné metody shlukové analýzy, které mají výhodu v tom, že je jejich interpretace názorná a lze najít i jejich fyzikální, resp. dopravně-inženýrské zdůvodnění. Aplikace shlukování pro redukci dopravních dat je uvedena například v [10]. Na obr. 4 je znázorněna redukce počtu  $s$ - detektorů v jisté oblasti na  $i$ -významných detektorů. Počet a umístění shlukovaných detektorů by se měl shodovat pro všechny dále zpracovávané parametry, tedy v případě hodnocení kvality dopravy alespoň pro intenzitu a obsazenost.

Na výstupu první etapy je tedy jedna nebo několik časových řad alespoň dvou dopravních parametrů – intenzity a obsazenosti. Tyto časové řady mají pseudo-stochastický průběh se silnou „stejnoseměnou“ složkou, kterou představují denní variace dopravy, na niž je superponována stochastická komponenta. Tato komponenta v principu závisí na distribuci vozidel v síti dané statistickými modely, ale ovlivňují ji vnější podmínky (technické parametry vozidla a komunikační, světelná signalizace apod.).

**2. krok řešení:** model vychází z toho, že přímo z časových řad není možné jednoduše odhadovat kvalitu dopravy, a proto je nutné provést další transformaci. Tentokrát budou dva dopravní parametry  $q$  (intenzita) a  $\kappa$  (obsazenost) nad  $J$  virtuálními detektory transformovány do jednoho rozměru  $\Delta$  s danou metrikou  $L$  dle vzorce

$$2.krok \quad \{q^J(t), \kappa^J(t)\} \xrightarrow{x} \{\Delta_L(t)\} \quad (2)$$

Jak bylo řečeno,  $\Delta$  charakterizuje kvalitu dopravy LOS (Level of Service) na celočíselné stupnici  $L \in \{1, 5\}$ . Klasifikaci dopravy zajišťují automatizovaně pracující klasifikátory. Výstupem tohoto kroku je ohodnocená úroveň dopravy  $\Delta$  na  $J$  virtuálních detektorech ( $J$ -rozměrný vektor).

**3. krok řešení:** v tomto kroku se dílčí hodnocení kvality dopravy prováděné nad reálnými, či virtuálními detektory transformuje do obrazu dopravy nad analyzovanou oblastí  $\Psi^E$ , přičemž úroveň dopravy nabývá libovolných hodnot od 1 do 5. Tato transformace se nejlépe realizuje expertním systémem popsáným dále.

V případě této transformace každá klasifikace dopravy  $\Delta_L^N$  nad každým reálným či virtuálním detektorem  $N$  vytváří obraz dopravy nad sledovanou městskou sítí  $\Psi^E$ . Matematicky zapsáno se jedná o následující přiřazení:

$$3.krok \quad \left| \Delta_L^N \right| \xrightarrow{\oplus} \Psi^E \quad (3)$$

Pro získání této výsledné hodnoty charakterizující kvalitu dopravy je vhodné použít vhodnou formu expertního systému, který byl popsán výše. V tomto specifickém případě je však navrhováno využití expertního systému s uvažováním principů neurčitosti. Takovýto systém je založen na tom, že heuristické znalosti jsou vyjádřeny ve zjednodušeném tvaru produkčních pravidel ve tvaru  $E \Rightarrow H$  s váhou  $V$ :

POKUD  $\{ \text{předpoklad } E \}$  POTOM  $\{ \text{závěr } H \}$  S  $\{ \text{váha } V \}$ ,

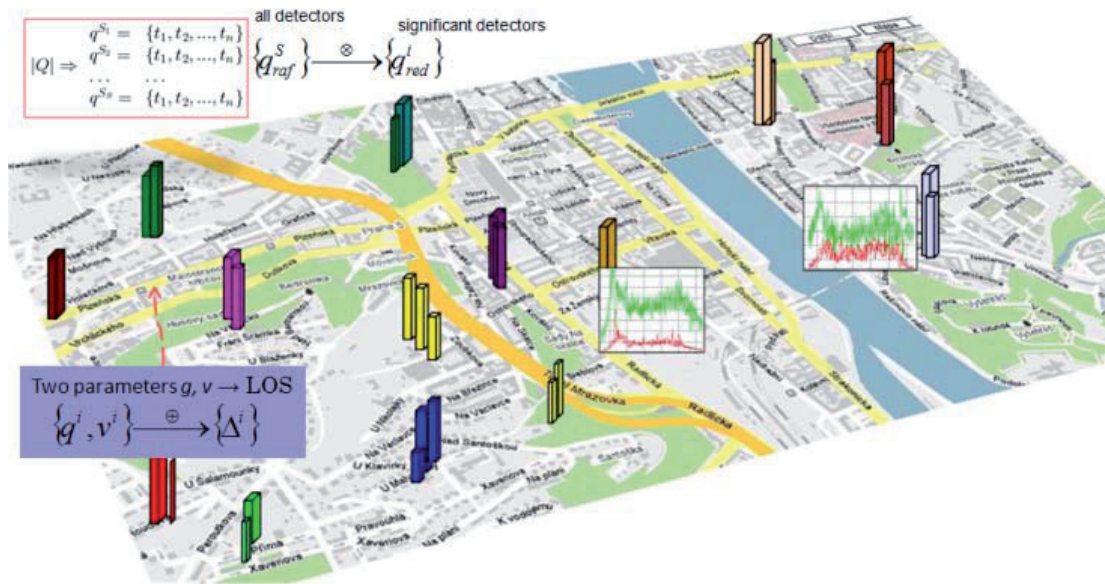
kde  $E$  ... je (evidence),  
 $H$  ... (hypothesis) jsou elementární tvrzení  
 $V$  ... váha  $V$  je subjektivní mírou důvěry experta v platnost pravidla.

Ve výsledku je pak hodnocení globální úrovně dopravy nad komplexní dopravní sítí vyjádřeno v souboru pravidel (příklad):

1.  $IF \Delta_L^1 = 3 \text{ OR } \Delta_L^5 = 4 \quad \text{THAN } \Psi^E = 3$
2.  $IF \Delta_L^3 = 3 \text{ OR } \Delta_L^4 = 4 \quad \text{THAN } \Psi^E = 4,5$
3.  $IF \Delta_L^1 = 1 \text{ OR } \Delta_L^5 = 2 \quad \text{THAN } \Psi^E = 3,8$

V tomto případě hodnocení LOS nad jednotlivými detektory odhaduje výslednou kvalitu.

Tak, jak je u expertních systémů obvyklé pravidla se ladí za účasti expertů. Výsledkem dobře pracujícího expertního systému je globální hodnocení kvality dopravy na úrovni, na jaké by je mohl provádět špičkový expert.

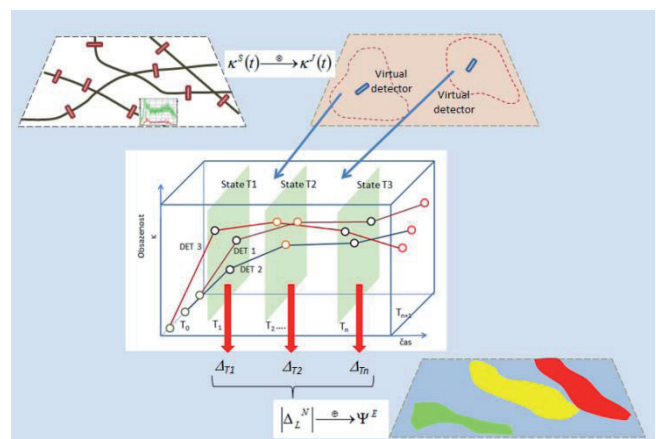


Obr. 4 Dopravní model města s vyznačenou redukcí v prostoru a v dimenzi proměnných  
 Fig. 4 Traffic model of a city with a marked reduction in space and dimension of variables

### Závěr a směr dalšího výzkumu

Globální model městské oblasti spočívající v redukcí počtu proměnných a jejich transformaci do hodnocení LOS nejprve nad virtuálními detektory a následně ve vytvoření obrazu dopravy  $\Psi^E$  nad celou oblastí je vytvářen po jednotlivých časových krocích. Tím je vlastně vytvořen stavový prostor, který je charakterizován pozicí hodnotícího vektoru LOS v diskretních časech. S výjimkou mimořádných stavů lze tento vývoj odhadovat. Tím, že jsou známy trajektorie dílčích hodnocení lze, prohledáváním stavového prostoru najít budoucí hodnoty  $\Psi^E$ .

Pro velmi názornou interpretaci obrazu dopravy nad danou oblastí se hodí tzv. mentální mapa. Mentální proto, že představuje hodnocení dopravy tak, jak by ho mentálně hodnotili nejlepší experti. Tato **mentální mapa** tedy vhodným grafickým s barevným rozlišením interpretuje vypočítané hodnoty  $\Psi^E$  a to včetně jejich změn.



Obr. 5 Procesy vedoucí od měření dopravních dat k tvorbě globálního modelu a mentální mapě

Fig. 5 Processes streaming from traffic data measurement to a global and mental model

Takovéto zobrazení kvality dopravy a jejího vývoje je velmi názorné i pro dispečery.

Mentální mapou lze také rozumět grafické (schematické) vyjádření představ člověka o geografickém prostoru a to nejčastěji jeho kvalitě nebo uspořádání. Přitom nejde o zachycení reálného prostoru, ale hlavně o jeho subjektivní odraz ve vědomí člověka. V případě globálního modelu je zkoumaná aglomerace nahlížena z hlediska kvality dopravy. Takovýto model umožní sledovat desítky či stovky senzorů s časově proměnnými parametry v globálním pohledu a oceňovat dopravu dle dohodnutých stupňů. Využití mentálních map ve velkých dispečincích je předmětem dalšího výzkumu.

## Poděkování

*This contribution is the result of the project implementation: **Research centre of transport telematics, ITMS 26220220169 supported by the Research & Development Operational Programme funded by the ERDF.***



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

## Literatura

- [1] JEANNOTTE, K., CHANDRA, A., ALEXIADIS, V., SKABARDONIS, A.: Traffic Analysis Toolbox Volume II : Decision Support Methodology for Selecting Traffic Analysis Tools. Report No. FHWA-HRT-04-039 (Vol. II), 2004.
- [2] LIU, H.-H., HSU, P.-L.: Design and Simulation of Adaptive Fuzzy Control on the Traffic Network. SICE-ICASE, International Joint Conference. pp. 4961-4966, 2006.
- [3] YAN, L., HUIMIN, FAN, X: Design of an adaptive fuzzy logic controller for traffic signals. Intelligent Control and Automation, WCICA 2008. 7th World Congress on Topic(s): Computing & Processing (Hardware/Software) ; Robotics & Control Systems. pp. 730-735, 2008.
- [4] HU, Y., CHIOU, A., HAN, Q.: Hierarchical Fuzzy Logic Control for Multiphase Traffic Intersection Using Evolutionary Algorithms. 2009. Industrial Technology, ICIT 2009. IEEE International Conference on Topic(s): Computing & Processing (Hardware/Software); Power, Energy, & Industry Applications ; Robotics & Control Systems, pp. 1-6, 2009.
- [5] Kaige, W., YANG, W., QU, S.: A stochastic adaptive traffic signal control model based on fuzzy reinforcement learning. Computer and Automation Engineering (ICCAE), The 2nd International Conference on Vol. 5, pp. 467-471, 2010.
- [6] KRČÁL, J.: Metody výpočtu a druhy řízení dopravního uzlu. Diplomová práce FD ČVUT, Praha, 2003.
- [7] PAPAGEORGIOU, M.: „Automatic Control Methods in Traffic and Transportation. Operations Research and Decision Aid Methodologies in Traffic and Transportation Management“, Proceeding of the NATO ASI on Operations Research and Decision Aid, Hungary, 1997

[8] DAVARYNEJAD, M., VRACKEN, J.: A Survey of Fuzzy Set Theory in Intelligent Transportation: State of the art and future trends. Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics, San Antonio, TX, USA, pp. 3952–3958, 2009.

[9] HADJIMICHAEL, M.: „A fuzzy expert system for aviation risk assessment“, Expert Systems with Applications, 36(3), pp. 6512–6519, 2009.

[10] PŘIBYL, O., PŘIBYL, P.: Mining data from induct loops. ITSC 2005, 8th International IEEE Conference on ITS, Vienna, Austria, 2005.

## Abstract

Even though there are typically many traffic sensors and a lot of measured traffic data in every city, it is not an easy task to interpret such data and to understand what is happening on the streets. Such traffic data are used by local controllers and stored in huge data bases but their usage is limited. Dispatchers typically decide about control measures based on additional information from CCTV cameras and not based on such data.

The objective of this paper is to point out the importance of processing and understanding traffic data. To do this, usage of so called **global model** is proposed. First, an overview of existing control strategies and control levels, together with some real world implications is presented. Next, the basic overview of a global model and its major steps is provided as well. These steps cover the process of data pre-processing, data reduction, and transformation into level of service. Important in such approach is to keep in mind time- and space- dependencies. Also careful handling of typical behaviour with a lot of expert knowledge and unexpected event, which require quick reaction is an important and not so easy task dealt with in this paper.

**prof. Ing. Pavel Příbyl, CSc.**

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacích a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
Tel.: +421 415133301  
E-mail: [pavel.pribyl@fel.uniza.sk](mailto:pavel.pribyl@fel.uniza.sk)  
[pribylp@fd.cvut.cz](mailto:pribylp@fd.cvut.cz)

**doc. Ing. Ondřej Příbyl, Ph.D.**

České vysoké učení technické  
Fakulta dopravní  
Ústav aplikované matematiky  
Na Florenci 25,  
Praha 1, 11000  
Czech Republic  
Tel.: +420 731225297  
E-mail: [pribylo@fd.cvut.cz](mailto:pribylo@fd.cvut.cz)

# Simple and Low Cost Information System for Car Drivers

Peter Vestenický, Martin Vestenický

## Abstract

A simple on-board unit (OBU) was developed to inform the drivers of motor vehicles about traffic signs in the selected critical road sectors. Information about the traffic signs is transmitted into vehicle by radio beacons which are placed at selected points. The system of radio beacons arrangement and simple communication protocol between the OBU and the radio beacons has been proposed. The signalling of the traffic signs to the driver is performed by playing of voice records. In addition to this basic function the OBU informs about rapid vehicle braking by blinking of yellow light. Moreover the OBU is equipped by simple alcohol tester and triaxial accelerometer. The data from the accelerometer are cyclically recorded into non-volatile memory for purpose of road traffic accident analysis. A simple mathematical algorithm for such analysis has been created.

**Keywords:** On-board unit, voice signalization, traffic sign, alcohol tester, road traffic accident analysis

## Introduction

In year 2003 the European Commission agreed the Third European Action Programme for Road Safety (RSAP) which set the ambitious target to halve the number of road deaths by the year 2010. This action programme started positive trend in the road safety area which the European Commission wants to keep also in this decade by setting similar target of halving the overall number of road deaths in the European Union by 2020 starting from 2010 [1].

A simple on-board unit (OBU) for active driver support has been developed to decrease the number of road traffic accidents on roads of Slovakia. The OBU is determined to be installed into any type of motor vehicle. The installation of OBU does not require any intervention into vehicle network installation except for the connecting of power plug into cigarette lighter connector. Mechanical fixation in vehicle is solved by a standard GPS/PDA car holder at present state of design (Fig. 1).



Fig. 1 On-board unit

Basic functions of the on-board unit have been specified to warn the vehicle driver against the most frequently reasons of road traffic accidents which are analysed in [2].

## 1. On-board unit and its features

The main function of the OBU is voice signalization of selected traffic signs for example at critical road sectors. Information about traffic signs is transmitted by stationary radio beacons which are an integral part of system solution (Fig. 2). The main (compulsory) information can be supplemented by additional information whose receiving is optional.



Fig. 2 Radio beacon

The next functions of on-board unit enable to signal rapid deceleration of vehicle by blinking of supplemental yellow light if it is installed in a rear part of the vehicle, to test the presence of alcohol in driver's breath, to record vehicle acceleration in three axes into rewriteable memory for purposes of motion analysis in the case of road accident. The on-board unit can be equipped with the Bluetooth module which will enable communication with an external GPS receiver or external GSM phone for emergency calls [3].

The on-board unit is controlled by powerful 16 bit microcontroller (CPU) PIC24FJ64GA004 working at 32 MHz clock frequency which provides computing power of 16 MIPS. Peripheral circuits are connected to I/O ports of CPU and these circuits perform required operations in cooperation with firmware of the OBU (Fig. 3, Fig. 4).

Main function – voice signalization of traffic signs – is performed by the receiver module working at frequency 433.92 MHz in cooperation with CPU. The voice notifications are prepared as voice records on the Secure Digital memory card and they are being played by digital – analogue (D/A) converter and speaker. Two blocks of triaxial accelerometers have ranges of  $\pm 2g$  and  $\pm 100g$  and provide information about vehicle braking or road traffic accident.



Fig. 3 The OBU prototype

The data from the accelerometers are being cyclically recorded into the FRAM (Ferroelectric RAM) memory. The blocks of accelerometers and the memory are being diagnosed and tested at every boot of the OBU. In case of the accident detection the cyclic recording is stopped and data from accelerometers together with real time information can be copied into personal computer via USB interface for the analysing of vehicle motion before, during and after the road traffic accident.

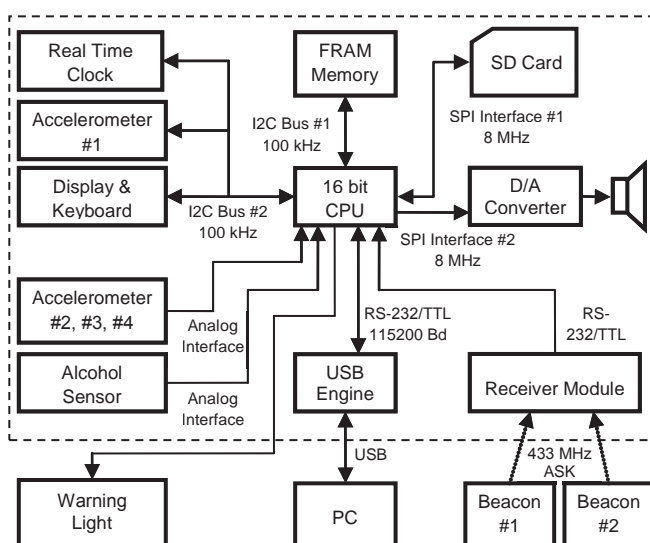


Fig. 4 Block diagram of the on-board unit

## 2. Radio beacons

The core of radio beacon is created by 8 bit microcontroller which performs all required functions in cooperation with software (Fig. 5). Radio beacon is also equipped with temperature sensor which detects worsened meteorological conditions. Configuration data i. e. content of transmitted datagram are written into radio beacon by personal computer with configuration software via standard serial port (RS-232) or via USB/RS-232 convertor in case of serial port absence.

Powering of radio beacon is solved by combination of a solar panel and accumulator. The solar panel supplies energy for whole radio beacon and for accumulator charging in case of sufficient solar radiation. The secondary role of the solar panel is to detect decreased visibility and distinction between day and night. The accumulator is powering the radio beacon when incident solar radiation is insufficient for correct operation of the radio beacon.

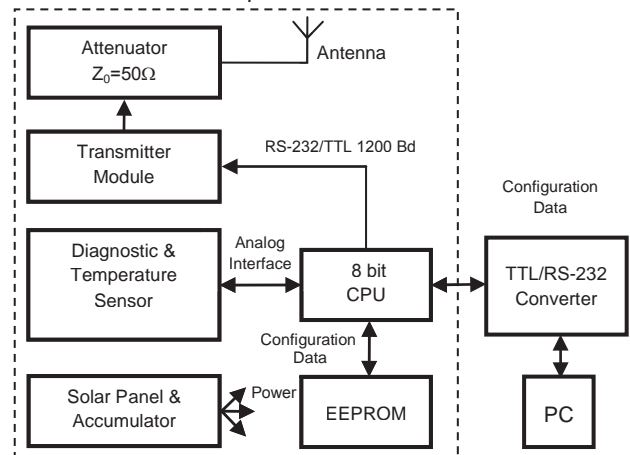


Fig. 5 Block diagram of radio beacon

The transmitter module works at frequency 433.92 MHz with 100 % amplitude modulation (ASK). The module has output power approximately 1 mW and enables the transmission of datagrams which are generated by the CPU. The additional circuitry contains attenuator and antenna. The attenuator can decrease the radiated power from antenna to reduce the area of radio beacon signal coverage.

A simple communication protocol for information transfer from radio beacon into the OBU has been created. This protocol is based on transfer of the frame whose transmission is continuously repeated in endless loop. The frame format is defined in the Table 1. All frame items are transferred as ASCII characters to increase error detection efficiency.

STX	ID	IDN	I1H	I1L	I1DH	I1DL	CRCH	CRCL	ETX
-----	----	-----	-----	-----	------	------	------	------	-----

Table 1 Frame format of radio beacon

Individual items denote:

- STX Start of Text, start of frame delimiter,
- ID identification number of radio beacon can take value from 1 to 15,
- IDN identification number of next radio beacon can take value from 1 to 15,
- I1H identification of main information – high byte,
- I1L identification of main information – low byte,
- I1DH identification of supplementary information – high byte,
- I1DL identification of supplementary information – low byte,
- CRCH checksum – high byte,



CRCL checksum – low byte,  
 ETX End of Text, end of frame delimiter.

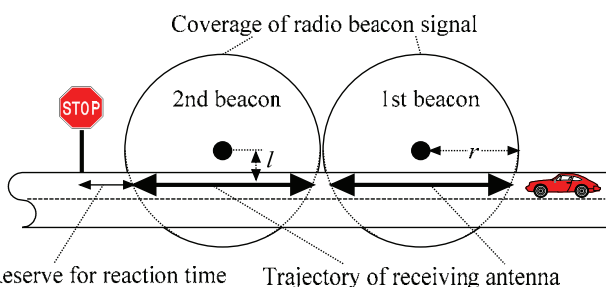
A cyclic code CRC 16 – CCITT with generator polynomial  $G(x)=x^{16}+x^{12}+x^5+1$  is used for error detection during frame transmission. The CRC checksum is computed only from marked frame items (Table 1). Information is transmitted in standard format of the asynchronous serial interface: 1 start bit, 8 data bits, 1 stop bit at modulation speed 1200 Bd. Complete frame duration can be computed from the formula (1):

$$t_f = \frac{n_f \cdot N_b}{v_{mod}} \quad (1)$$

where

$n_f$  number of frame items (=10),  
 $N_b$  length of one item (=10 bits),  
 $v_{mod}$  modulation speed (=1200 Bd).

After substitution of given values  $t_f = 83.3$  ms.



**Fig. 6 Arrangement of the radio beacons**

Philosophy of radio beacon arrangement beside road is shown in the Fig. 6. The radio beacons are placed in pairs to specify the direction and they must be arranged in sufficient distance from each other so that the individual coverage areas have not been overlaid. Also sufficient reserve for driver reaction time before marked traffic sign must be created.

Radius of signal coverage is approximately  $r = 25$  m. Assuming that the radio beacon is placed outside of traffic lane axis at distance  $l = 6$  m the trajectory of receiving antenna in radio beacon signal coverage is  $s = 2\sqrt{r^2 - l^2} = 48.5$  m. Vehicle with velocity  $v = 130$  km/h (36.11 m/s) passes through this distance at the time  $t = 1.3$  s. Information from radio beacon is transmitted in frame whose duration is 83.3 ms (1). It means that the on-board unit receives frame from one radio beacon more than 15 times. This is sufficient reserve for reliable receiving of at least one error-free frame.

The process of information receiving from the radio beacons (part of the OBU firmware) is waiting on the message from the receiver module. After receiving the message the CRC check is performed and if the CRC is correct the software procedure (Fig. 7) decides if the received message was sent by the first or the second beacon of the pair. In the first case the procedure waits again until the message from the second beacon is received. After the correct receiving of both messages the procedure identifies pair of beacon and main program loop of the OBU firmware plays the appropriate audio warning.

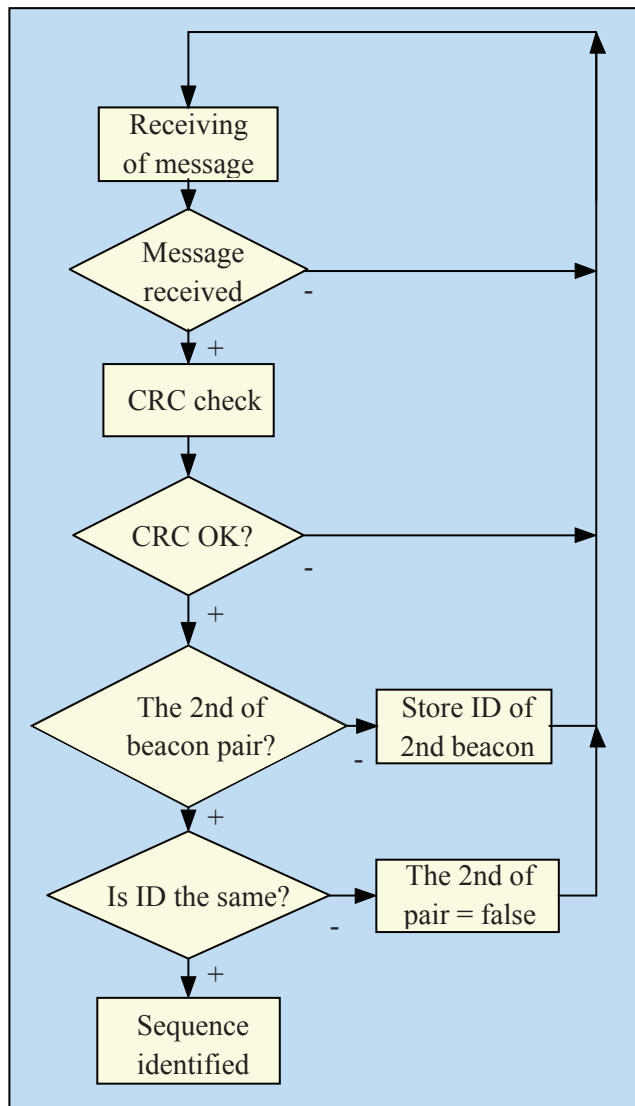
Because the communication will be performed between stationary radio beacon and moving vehicle the influence of Doppler's shift  $f_D$  which is given by formula (2) must be taken into account:

$$f_D = \frac{v \cdot f_C}{c} \quad (2)$$

where

$v$  vehicle velocity,  
 $f_C$  carrier frequency,  
 $c$  speed of light.

After substitution of given values i. e.  $f_C = 433.92$  MHz,  $v = 36.11$  m/s (130 km/h) and  $c = 3 \cdot 10^8$  m/s into formula (2) the Doppler's shift value is  $f_D = 52.23$  Hz. Taking the bandwidth of receiver module (600 kHz) [4] and used modulation speed (1200 Bd) into account it is negligible.



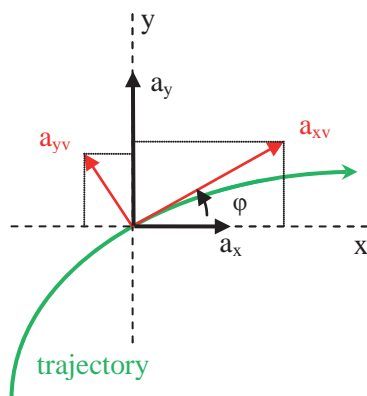
**Fig. 7 Processing of information received from radio beacon pair**

### 3. Simplified reconstruction of vehicle motion in the case of road traffic accident

The on-board unit includes two blocks of triaxial accelerometers with ranges of  $\pm 2g$  and  $\pm 100g$  (Fig. 4). The data from the accelerometers are continuously written into memory. Applying physical theory and mathematical computations the trajectory and velocity of vehicle in every point can be reconstructed [5]. It may be useful for analysing of road traffic accidents. The time range of record in memory is enough to store data approximately 30 s before and 30 s after the accident detection. The detection is based

on local extreme monitoring in series of acceleration samples provided by blocks of accelerometers.

For correct motion reconstruction the block of accelerometers would be placed on gyro stabilized platform to keep its position constant compared to fixed coordinate system of Earth or the angles between vehicle coordinate system and fixed coordinate system would be measured by gyroscopes. In the real construction of on-board unit the gyroscopes have not been used for price reason. Therefore the acceleration samples  $a_{xv}$  and  $a_{yv}$  (Fig. 8) which are provided by accelerometers in the moving vehicle coordinate system must be recalculated to acceleration samples  $a_x$  and  $a_y$  which correspond with samples measured in fixed coordinate system  $xy$ . Both coordinate systems form an angle  $\varphi$  which is dynamically changed during vehicle motion. Note that (for simplicity and testing) only two dimensional calculations were taken into account i. e. the vehicle motion on horizontal plane only was assumed.



**Fig. 8 Geometrical relations of two coordinate system**

The recalculation of acceleration values from vehicle into fixed inertial coordinate system issues from geometrical relations in the Fig. 8 and can be done by applying difference equations (3) and (4). Actual values of velocity and trajectory must be computed by equations (5), (6) and (7), (8), respectively. Then next value of angle  $\varphi$  is given by equation (9).

$$a_x(n) = a_{xv}(n) \cdot \cos \varphi(n-1) - a_{yv}(n) \cdot \sin \varphi(n-1) \quad (3)$$

$$a_y(n) = a_{xv}(n) \cdot \sin \varphi(n-1) + a_{yv}(n) \cdot \cos \varphi(n-1) \quad (4)$$

$$v_x(n) = v_x(n-1) + a_x(n-1) \cdot \Delta t \quad (5)$$

$$v_y(n) = v_y(n-1) + a_y(n-1) \cdot \Delta t \quad (6)$$

$$s_x(n) = s_x(n-1) + v_x(n-1) \cdot \Delta t + \frac{1}{2} a_x(n-1) \cdot \Delta t^2 \quad (7)$$

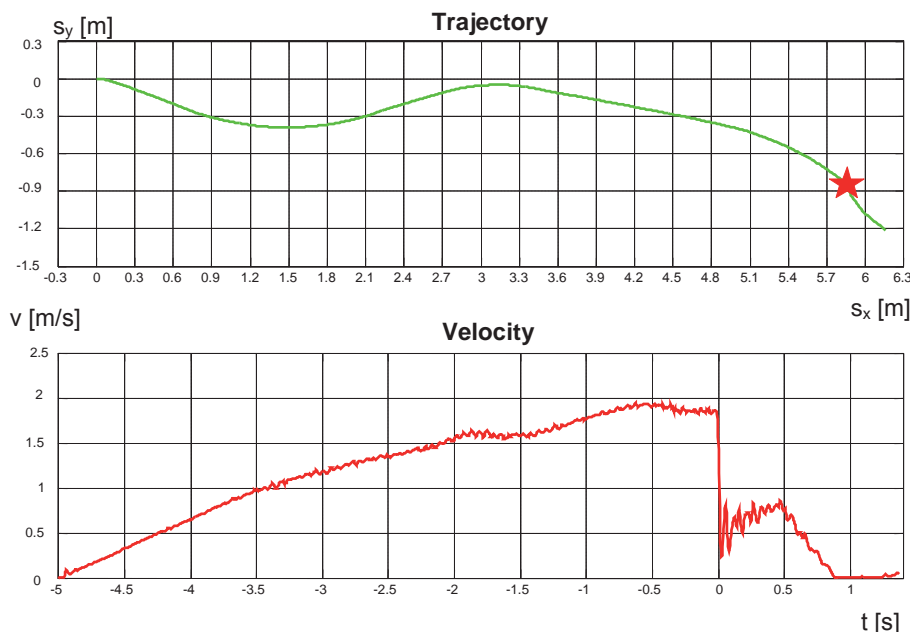
$$s_y(n) = s_y(n-1) + v_y(n-1) \cdot \Delta t + \frac{1}{2} a_y(n-1) \cdot \Delta t^2 \quad (8)$$

$$\varphi(n) = \arctg \frac{s_y(n) - s_y(n-1)}{s_x(n) - s_x(n-1)} \quad (9)$$

The accelerations are sampled by frequency 250 Hz in tested prototypes of the on-board unit therefore  $\Delta t = 4$  ms. Software for vehicle motion reconstruction has been created in MATLAB environment. Test data have been obtained during "crash" tests by using a remote controlled model of car. The trajectory and velocity reconstructed from the testing data are shown in the Fig. 9.

## Conclusion

Tests of the OBU have been performed by staff of the Transport Research Institute, Inc. in Žilina. Results of the tests proved full functionality of manufactured OBU prototypes and radio beacons. During next development of the OBU the radio communication subsystem between the OBU and the radio beacons will be re-evaluated. This subsystem is now based on simple radio modules working in non-licensed frequency band 434 MHz. For example, more appropriate radio modules are produced by Kapsch AG company and they are focused on intelligent transport system applications. They work in frequency band 5.855 – 5.925 GHz in conformance with standard IEEE 802.11p



**Fig. 9 Motion reconstruction from recorded acceleration values**

WAVE. For improvement of motion reconstruction accuracy the implementation of precision three axes combined gyroscope and accelerometer ADIS16400 is planned. This gyroscope is manufactured by Analog Devices, Inc. and is being used in navigation systems of unmanned aerial vehicles [6]. Disadvantage of these future plans is high price of more appropriate components.

### Acknowledgements

*This work has been supported by the Slovak Research and Development Agency, project No. VMSP-P-0030-07 "Simple On-board Unit for Active Support of Driver".*

*The paper was elaborated with support of the Slovak grant agency VEGA, grant No. 1/0453/12 "Study of interactions of a motor vehicle, traffic flow and road".*

### References

- [1] COM/2010/0389 final: Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and The Committee of the Regions. Towards a European road safety area: policy orientations on road safety 2011-2020. [online]. URL <<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2010:0389:FIN:EN:HTML>>
- [2] MIKULA, J., KUČERKOVÁ, V., VAŠEK, K., HRUDKAY, K., KRÍŽAN, J., KUCHARÍK, D.: Simple On-board Unit for Active Driver Support (PAJED). Annual research report of project APVV no. VMSP-P-0030-07. Transport Research Institute, Inc., Žilina, 2008 (in Slovak)
- [3] VACULÍK, M., VESTENICKÝ, P., VESTENICKÝ, M.: Vehicle On-board Unit. Proceeding of International conference ITS Bratislava '07, September 11th – 12th 2007, Bratislava. ISBN 978-80-254-0207-8 (in Slovak)
- [4] Aurel Wireless, Inc.: RX-4M50RR30SF 434 MHz High Performance Low Cost Receiver. Datasheet, [online]. URL <[http://www.aurelwireless.com/wireless/Short\\_Form/650200650\\_sf.pdf](http://www.aurelwireless.com/wireless/Short_Form/650200650_sf.pdf)>
- [5] GREWAL, M., S., WEILL, L., R., ANDREWS, A., P.: Global Positioning Systems, Inertial Navigation, and Integration. John Wiley & Sons, Inc., Hoboken, New Jersey, USA, 2007. ISBN 978-0-470-04190-1
- [6] Analog Devices, Inc.: ADIS16400/ADIS16405 Triaxial Inertial Sensor with Magnetometer. Datasheet, [online]. URL <[http://www.analog.com/static/imported-files/data\\_sheets/ADIS16400\\_16405.pdf](http://www.analog.com/static/imported-files/data_sheets/ADIS16400_16405.pdf)>

### Assoc. prof. Dr. Ing. Peter Vestenický

University of Žilina  
Faculty of Electrical Engineering  
Department of Control and Information Systems  
Univerzitná 8215/1  
010 26 Žilina, Slovakia  
Phone: +421415133345  
Fax: +421415131515  
E-mail: peter.vestenicky@fel.uniza.sk

### Ing. Martin Vestenický, Ph.D.

University of Žilina  
Faculty of Electrical Engineering  
Department of Telecommunications and Multimedia  
Univerzitná 8215/1  
010 26 Žilina, Slovakia  
Phone: +421415132255  
Fax: +421415131520  
E-mail: martin.vestenicky@fel.uniza.sk

# Fuzzy model času reakcie osôb v cestnom tuneli pri vzniku mimoriadnej udalosti

Peter Matis, Juraj Spalek

## Abstrakt

Článok sa zaoberá procesom evakuácie osôb po vzniku mimoriadnej udalosti v cestnom tuneli. Čas potrebný na evakuáciu sme rozdelili na jednotlivé časové intervaly podľa správania osôb. Na odhad času evakuácie je použitý fuzzy expertný systém. Podrobne sa venujeme nastaveniu fuzzy inferenčného systému na odhad času potrebného na detekciu mimoriadnej udalosti.

**Kľúčové slová:** cestný tunel, mimoriadna udalosť, evakuácia, samozáchrana, detekcia, fuzzy lingvistický inferenčný systém, expertný systém, term, univerzum

## Úvod

Cestný tunel predstavuje špecifickú časť cestnej infraštruktúry. Cestná doprava je spojená s rôznymi rizikami ako sú napríklad vznik porúch vozidiel, vznik dopravných nehôd, riziká spojené s prepravou nebezpečných nákladov alebo obmedzenie plynulosti dopravy s čím sú spojené predĺženie času prepravy a zvyšovanie nákladov na prepravu. Ak nastane mimoriadna situácia práve v cestnom tuneli, zvyšujú sa následky pre všetky zúčastnené osoby (účastníci cestnej premávky, členovia zložiek integrovaného záchranného systému (IZS), správca komunikácie a pod.). Ľudské úsilie je zameraná na obmedzenie počtu katastrof v cestných tuneloch a na zmiernenie ich následkov. Aj napriek tomu nie je možné úplne zabrániť vznikom katastrof a zvlášť spojených s požiarom. Tie predstavujú najnebezpečnejšiu situáciu v tuneli. Požiar môže vzniknúť ako následok dopravnej nehody alebo z dôvodu technickej poruchy. Úmyselné spôsobenie požiaru sa väčšinou neuvažuje. Tunel predstavuje uzavreté prostredie, preto pri vzniku požiaru nemôžu splodiny horenia voľne unikať. Tak isto osoby, ktoré sa v tuneli nachádzajú ho môžu opustiť len cez únikové východy. Vo všeobecnosti, dobre postavená stavba umožňuje voľný pohyb ľudí a ich rýchlu evakuáciu v prípade nebezpečenstva. Normatívy pre stavby nariaďujú, že v prípade nebezpečenstva musí byť každý, kto je prítomný, schopný dostatočne rýchlo opustiť rizikový priestor. V oblasti cestných tunelov sú to [1], [2], [3], [4]. Rozhodujúce sú prvé minúty od vzniku požiaru a rýchlosť začatia procesu evakuácie. Tento článok popisuje metódy použité na odhad času úniku ohrozených osôb. Tieto metódy berú do úvahy špecifické technologické vybavenie cestného tunela [5]. Schopnosť rýchlo identifikovať požiar a efektívne informovať osoby v tuneli tak, aby začali evakuáciu, znižuje reakčný čas a urýchľuje evakuáciu. Čas potrebný na detekciu a čas potrebný na informovanie osôb môže byť až niekoľko minút a významne predlžuje celkový čas evakuácie [6].

V súčasnosti nie je publikovaných mnoho štúdií správania sa ľudí pri požiaroch v tuneli [7]. Parametre správania sa ľudí významne ovplyvňujúce čas evakuácie sú:

- Voľba východu a únikovej dráhy – osoby sa rozhodujú na základe vlastností prostredia, teda vzdialenosti k východu, viditeľnosti a pod. Ľudia si pri požiaroch zvyčajne volia najbližší východ, ale niekedy sa radšej rozhodnú

pre únikovú cestu, ktorú poznajú (napr. portál tunela), pretože únikové východy z tunela sú pre nich rovnako odstrašujúce a neznáme.

- Stádovitosť – interakcia medzi ľuďmi je kľúčovým faktorom modelovania evakuácie. Ľudia sú silne ovplyvnení chovaním ostatných, platí to pri rozhodnutí opustiť automobil aj pri voľbe únikovej dráhy.
- Pohyb proti prevládajúcemu prúdu pohybu – Simulácia evakuácie pri požiaroch v tuneli nepotrebuje zohľadňovať takéto chovanie, pretože osoby nachádzajúce sa v rovnakej časti tunela budú mať tendenciu unikať rovnakým smerom.

Účinky požiaru na chovanie ľudí – Dym ovplyvňuje rýchlosť pohybu osôb (štúdie rozloženia rýchlosti pod vplyvom dymu), voľbu východu, zhoršuje zdravotnú spôsobilosť unikajúcich a pod.

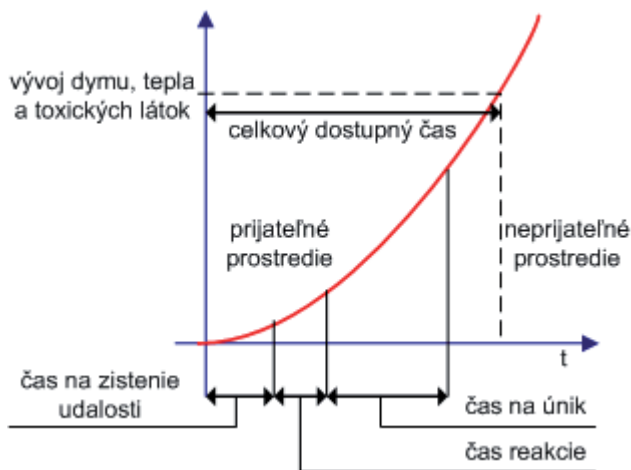
## 1. Proces evakuácie

Pri vzniku mimoriadnej udalosti v tuneli sú rozhodujúce prvé minúty od vzniku požiaru a rýchlosť začatia procesu samozáchranu. Súbor aktivít pozostáva od zastavenia vozidla až po opustenie ohrozeného priestoru únikovým východom.

V príspevku je rozbor scenára samozáchranu v prípade, že požiar je zistený osobou v tuneli a táto oznámi vznik mimoriadnej udalosti na tiesňovú linku. Pred miestom vzniku požiaru zastavujú vozidlá. Od vzniku požiaru po jeho spozorovanie ubehol čas, ktorý budeme označovať ako čas potrebný na detekciu  $t_d$ . Osoby, ktoré priamo vidia požiar začínajú odhadovať nebezpečenstvo. Čas, počas ktorého si osoby uvedomia, čo sa deje a rozhodujú sa ako na situáciu zareagujú, označujeme ako čas reakcie  $t_r$ . Tento čas závisí aj od toho, či je požiar viditeľný, ako ďaleko stojí vozidlo od miesta požiaru, od osvetlenia a pod. Dôležitá je tiež reakcia osoby na vzniknuté nebezpečenstvo. Niektorí zavolá na tiesňovú linku, niektorí nechce opustiť vozidlo a zdržiava proces evakuácie a niektorí môžu napr. použiť SOS búdku na ohlásenie udalosti. Po prijatí rozhodnutia o začatí evakuácie osoby začnú vykonávať činnosti, ktoré predchádzajú samotnému procesu evakuácie. Tieto činnosti môžu byť racionálne, ako napr. poskytnutie pomoci deťom, starším osobám pri vystupovaní z vozidla alebo iracionálne ako napr. hľada-

nie dokladov a mobilného telefónu, vybranie batožiny a pod. V našom modelovom prípade osoba ktorá spozorovala požiar informuje dispečera tiesňovej linky a ten následne dispečera tunela. Ten začne po overení udalosti vykonávať činnosti vedúce k zmierneniu následkov požiaru: uvedenie do činnosti ventilácie, uzavretie tunela a uvedenie do činnosti informačného systému. Informačný systém, ktorý tvoria napríklad subsystemy ozvučenia a informačné displeje, upozorní účastníkov na vzniknuté nebezpečenstvo, urýchľuje a organizuje ich záchranu. Tieto prvé minúty od vzniku požiaru sú najdôležitejšie na záchranu osôb. Účastníci premávky sa začínajú pohybovať smerom k únikovým východom. Čas chôdze potrebný na evakuáciu označujeme ako  $t_u$ . Po príchode zložiek integrovaného záchranného systému (IZS), tieto preberajú organizáciu záchranu osôb, následne likvidujú požiar. Požiadavky na zmenu režimu ventilácie alebo iného technologického vybavenia realizujú prostredníctvom dispečera tunela alebo sami. V našom prípade sa budeme zaoberať iba procesom samozáchranu, t. j. časom od vzniku požiaru po únik osoby z ohrozenej oblasti [7].

Proces evakuácie sa dá definovať ako čas, ktorý uplynie od momentu, kedy ľudia udalosť identifikujú do momentu, kedy sa presunú na bezpečné miesto. Na obr. 1 je znázornené prijateľné prostredie ako prostredie v ktorom je únik osôb možný. V prípade požiaru je to do času, kým dym ostáva oddelený pod stropom tunela vo výške viac ako 2 metre a kým sa ešte neprejavuje priame pôsobenie tepelného výkonu požiaru.



Obr. 1 Vývoj požiaru a celkový čas na záchranu

Fig. 1 Growth the fire and available safe egress time

Evakuačný proces pozostáva z niekoľko časových úsekov, ktoré možno analyzovať samostatne, pretože sú z hľadiska správania ľudí a činností špecifické:

- čas na zistenie udalosti,
- čas reakcie,
- čas na únik.

Čas na zistenie udalosti sa dá charakterizovať ako čas, kedy si osoba uvedomí, že sa niečo deje. Tento čas súvisí s charakterom udalosti. Ľudia, ktorí priamo vidia plamene a dym, si túto situáciu uvedomia skoro okamžite. Platí, že čím je vizuálny následok udalosti väčší, tým je tento čas kratší. A môže sa značne znížiť, ak je tunel vybavený systémami na automatickú identifikáciu požiarov a informovanie osôb. Taktiež dôležitou je výchova a informovanosť vodičov. Čas na zistenie udalosti je individuálny a závisí od druhu udalosti, jej prejavu, vzdialenosti od miesta a od všeobecnej informovanosti. Vhodný systém na detekciu v kombinácii so systémom na informovanie osôb v tuneli tento čas skracujú a zjednocujú pre všetky osoby v tuneli.

Čas reakcie je čas od okamihu zistenia udalosti do momentu začatia evakuácie. Osoba v tuneli si počas tohto času uvedomuje situáciu a hľadá vhodné riešenie. Až v 80 %

prípadoch zopakuje správanie ostatných osôb, ktoré sa rozhodli už skôr. Pretože je tento čas individuálny pre každého, sú tu iné faktory, ktoré ho ovplyvňujú. Ide o správanie jednotlivca vo svojom okolí, prostredie tunela, viditeľnosť únikových východov, viditeľnosť alebo možnosť počutia verbálnych inštrukcií a pod. Všetky tieto faktory ovplyvňujú čas potrebný na prijatie rozhodnutia akú evakuačnú možnosť si má jedinec zvoliť. Aj tu platí, že vhodný systém na informovanie osôb tento čas skracuje a zjednocuje pre všetky osoby.

Čas na únik je časový interval, potrebný na pohyb chôdzou do miesta mimo ohrozenej oblasti. Na výpočet tohto časového intervalu slúžia evakuačné modely. Pri výpočtoch sa berie do úvahy rýchlosť pohybu osôb, hustota unikajúcich osôb, viditeľnosť, odstavené vozidlá a iné prekážky, šírka únikových ciest, vzdialenosť únikových východov, vplyv starších a hendikepovaných osôb a pod.

Modely dynamiky chodcov delíme do dvoch kategórií, makroskopické a mikroskopické. Makroskopické prístupy využívajú rôzne analógie dynamiky tekutín a plynov na simuláciu tokov v doprave a pohybu chodcov, napr. Evacnet. Medzi mikroskopické modely patria modely silového chovania, celulárne automaty, napr. Exodus a Egress, modely založené na pravidlách, agentove modely, napr. Simulex, PedGo a Pathfinder a hybridné modely.

V prvej fáze evakuácie osoby opúšťajú vozidlo. Rozdielna situácia môže byť pri opúšťaní osobného vozidla a pri úniku z plne obsadeného autobusu. V ďalšej fáze unikajúce osoby smerujú k únikovým východom a to rôznou rýchlosťou. Strednú hodnotu unikajúcich osôb určuje PIARC z roku 1999 na  $0,5 - 1,5 \text{ m.s}^{-1}$ , pričom sa myslí rýchlosť osôb, ktoré sa pohybujú voľne a nie sú ovplyvnené zástupom ľudí. Čas úniku je potrebné zahrnúť do výpočtu času samozáchranu a tento čas môže byť rôzny v závislosti od stavebného usporiadania tunela a jeho vybavenia osvetlením, značením únikových východov a pod. Vzhľadom k tomu, že existuje veľké množstvo modelov na simuláciu evakuácie osôb, nebudeme sa času úniku podrobne v tomto článku venovať.

Prognóza pohybu osôb v priebehu evakuácie je základným aspektom na posúdenie ich bezpečnosti. Evakuáciu možno považovať za bezpečnú, ak medzi časom RSET (required safe egress time) potrebným na evakuáciu a časom ASET (available safe egress time) dostupným na evakuáciu platí:

$$RSET \leq ASET \quad (1)$$

Čas RSET je zložený z čiastkových intervalov:

$$RSET = t_d + t_v + t_z + t_u \text{ (min)}, \quad (2)$$

kde:

$t_z = t_r + t_z \text{ (min)}$ ,  $t_d$  je čas od vzniku do detekcie požiaru (min),

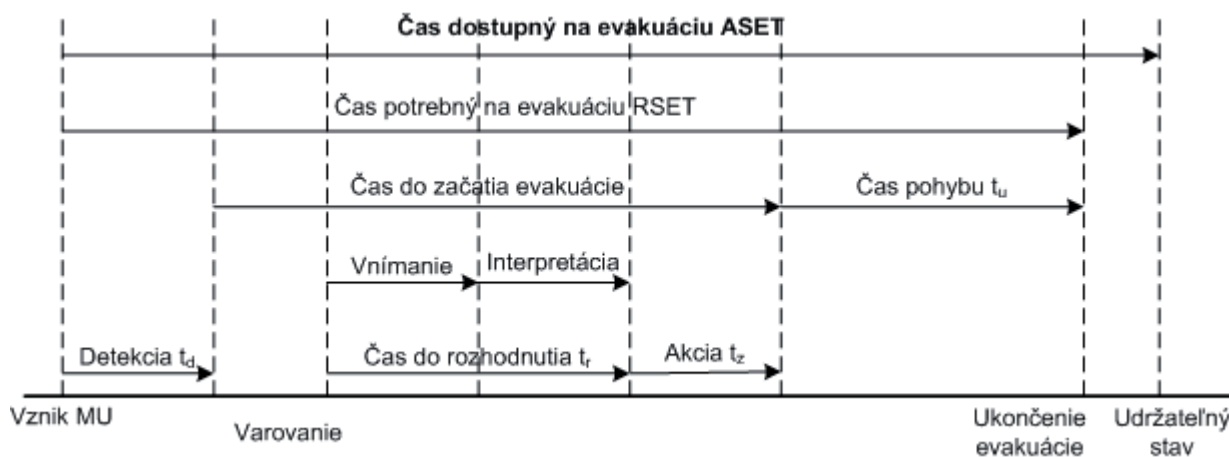
$t_v$  – čas od detekcie požiaru do vyhlásenia evakuácie (min),

$t_r$  – čas od vyhlásenia evakuácie, do prijatia rozhodnutia k jej uskutočneniu (min),

$t_z$  – čas od prijatia rozhodnutia o začatí evakuácie do jej reálneho začatia, (min),

$t_u$  – čas úniku osôb (min).

Dostupný čas pre evakuáciu osôb ASET je na obr. 2. Čas  $t_d$  závisí od vybavenia objektu požiarovými bezpečnostnými zariadeniami. Čas  $t_v$  závisí od technického vyhotovenia a umiestnenia požiarneho bezpečnostného zariadenia, bezpečnostného manažmentu a od reakcie osôb. Významné časové zdržanie v procese evakuácie predstavuje čas  $t_z$ . Tento čas zahŕňa trvanie rozhodovacieho procesu  $t_r$  a čas do uskutočnenia evakuácie  $t_z$ . Vnímáním je časový úsek, v ktorom osoby spozorujú varovný signál. Interpretácia varovania je časový interval, kedy osoby posudzujú závažnosť varovania a dochádza k rozhodnutiu. V tomto časovom úseku si môžu osoby zisťovať ďalšie informácie, ktoré sú pre rozhodnutie zásadné. Čas do



Obr. 2 Dostupný čas na evakuáciu osôb

Fig. 2 Available safe egress time

uskutočnenia evakuácie je časový interval, v ktorom osoby realizujú sústavu opatrení, ktoré považujú pred uskutočnením evakuácie za nutné, napr. zhromaždenie detí, zhromaždenie cenností, dokladov a pod. Medzi najvýznamnejšie skutočnosti ovplyvňujúce čas od vyhlásenia evakuácie po jej začatie môžeme zaradiť:

- **spôsob vyhlásenia požiarneho poplachu** – spravidla vyhlasovaný technickým zariadením. Za najefektívnejší spôsob sa považuje vyhlásenie požiarneho poplachu dobre počuteľnou varovnou správou vysielanou ozvučiacim zariadením alebo pomocou rádiového vysielача. Varovné akustické signály sirén sú menej efektívne, lebo vyžadujú ďalší čas, kedy osoby zhromažďujú informácie o vzniknutej udalosti,
- **vizuálny prístup** – rozhodovací proces je do značnej miery ovplyvnený chovaním ostatných osôb v tuneli ako aj viditeľnosťou ohňa alebo dymu,
- **výcvik, znalosti** – výcvik osôb zameraný na reakciu na varovný signál do značnej miery ovplyvňuje rozhodovací proces,
- **spoločenské vzťahy** – v prípade vzniku mimoriadnej udalosti sa osoby budú v prvom rade snažiť zhromaždiť rodinných príslušníkov, resp. osoby ku ktorým majú blízky vzťah. Táto činnosť zaberá dosť času, zvlášť ak nie sú blízke osoby pri vzniku incidentu pohromade.

Predpokladaný čas od vyhlásenia evakuácie po jej začatie sa odhaduje na menej ako 1 minútu v prípade, že sa informácia o vzniku mimoriadnej udalosti dostane k osobám prostredníctvom vysielanej verbálnej varovania. V prípade akustického signálu môže byť tento čas viac ako 4 minúty.

Predpokladaný čas evakuácie  $t_u$  predstavuje časový úsek, kedy dochádza k pohybu osôb objektom na voľné priestranstvo alebo do iného bezpečného priestoru [8], [9].

## 2. Návrh modelu na odhad času reakcie osôb

Po vzniku mimoriadnej udalosti je potrebné začať proces evakuácie v najkratšom možnom čase. Parametre, ktoré majú vplyv na to, koľko osôb sa zachráni sú množstvo osôb, veľkosť požiaru, šírenie dymu, čas úniku osôb, čas zisťovania nebezpečenstva a čas reakcie.

Po vzniku mimoriadnej udalosti sa začína pred týmto miestom vytvárať kolóna vozidiel, ktorých závisí od toho, či ide o mestský alebo diaľničný tunel. Pre výpočet sa uvažuje vždy rovnaká konfigurácia vozidiel pre mesto a diaľnicu. Tento pomer vozidiel je možné individuálne upraviť.

Veľkosť požiaru ovplyvňuje mortalitu priamo pôsobením tepla, pričom sa neuvažuje pôsobenie v okruhu niekoľko desiatok metrov a hlavne množstvom vyprodukovaného dymu. Šírenie dymu sa uvažuje pre určité fyzikálne podmienky (rýchlosť pohybu vzdušiny, teplota vzduchu a pod.) a daný profil tunela. Uvažuje sa časopriestorová závislosť hladiny dymu vo výške 2 metre. Pre zjednodušenie je možné uvažovať, že pre požiar s tepelným výkonom 5 MW sa uvažuje dosiahnutiu hladiny dymu vo výške 2 metre po 6 minútach. Teplý dym sa hromadí pod stropom tunela a s časom a vzdialenosťou klesá vplyvom ochladzovania. Po dosiahnutí hladiny 2 metre nad povrchom vozovky ohrozuje unikajúce osoby. Na stanovenie mortality je potrebné porovnať vzdialenosť, kam sa až dostanú unikajúce osoby a vzdialenosť v ktorej sa dym nachádza nižšie ako 2 metre nad vozovkou.

Čas úniku osôb udáva čistý čas od začiatku chôdze až po dosiahnutie únikového východu. Závisí od kvality únikových ciest, vzdialenosti od únikových východov a rýchlosti pohybu osoby. Modelovanie chôdze pre štandardizovaný tunel a pre maximálnu vzdialenosť medzi únikovými východmi sa vytvára na základe údajov z iných simulačných nástrojov. Pre zloženie cestujúcich sa uvažuje pomer 50 % muži, 40 % ženy a 10 % detí. Dôležité sú aj iné parametre, ako osvetlenie, kvalita značenia a pod.

Čas potrebný na zistenie nebezpečenstva udáva čas potrebný na uvedomenie si, že sa deje niečo mimoriadne. Tento čas je úmerný vzdialenosti od požiaru. Výskumy ukázali, že priemerný čas uvedomenia si nebezpečenstva je 120 sekúnd. Pri rýchlosti pohybu  $0,5 - 1,5 \text{ m}\cdot\text{s}^{-1}$  by osoba prešla vzdialenosť 60 – 180 metrov. Tunel vybavený identifikačným a informačným systémom čas uvedomenia si nebezpečenstva výrazne skracuje.

Čas reakcie je časový interval, počas ktorej sa osoba po zistení nebezpečenstva rozhoduje, čo má urobiť. Výskumy ukázali, že priemerný čas reakcie je 90 sekúnd, čo predstavuje možnosť prejdania 45 – 135 metrov. Tento čas je možné skrátiť podaním rýchlej a zrozumiteľnej informácie.

Výsledkom a cieľom vytváraného modelu je ohodnotenie úspešnosti procesu samoevakuácie. Výsledok bude závisieť od použitých technologických zariadení a subsystémov tunela. Týmto sú komplexný systém na identifikáciu udalosti, zväčša tvorený niekoľkými zariadeniami. Pokiaľ je udalosť rýchlo identifikovaná, je potrebné informáciu a prípadné pokyny k evakuácii poskytnúť postihnutým osobám pomocou systému na informovanie osôb, ktorý môžu tvoriť viaceré subsystémy. Skorá informácia taktiež zastavuje prúd vozidiel vchádzajúcich do tunela a spúšťa pripravený režim ventilácie. Po zistení nebezpečenstva a prijatí rozhodnutia

na opustenie nebezpečného priestoru sa ľudia vyberú smerom k najbližšiemu núdzovému východu (resp. smerom od miesta požiaru). Tento čas je principiálne určený vzdialenosťou únikových východov, zvolenou priemernou rýchlosťou a ďalšími parametrami ovplyvňujúcimi pohyb osôb. Tento čas sa nedá dodatočne upravovať bezpečnostnými opatreniami, je daný základnými stavebnými parametrami tunela.

Kvalitu výslednej cieľovej funkcie, ktorá má časový rozmer, pri všetkých skúmaných subsystémoch určuje viac komponentov. Na identifikáciu požiaru sa používa napríklad líniový hlásič teploty, senzor opacity alebo videodetekcia dymu. Každý z týchto senzorov využíva iný princíp a reaguje na rôzne fyzikálne veličiny. Účinnosť senzorov ovplyvňuje aj ich umiestnenie v tuneli, spôsob zapojenia do systému a ďalšie okrajové podmienky. Podobné je to aj so systémami na šírenie informácií pre osoby v tuneli a pri systémoch na urýchlenie evakuácie. Ide teda o klasifikáciu viacerých systémov a subsystémov s rôznymi okrajovými podmienkami, teda o hodnotenie heterogénneho systému. Na hodnotenie takýchto systémov sa s úspechom používajú expertné systémy. Prehľadnou metódou je fuzzy lingvistický, inferenčný systém. Základným princípom je inferencia na základe jazykového popisu, čo predstavuje množina podmienených výrokov v tvare:

**if  $X_1$  is  $A_1$  and ... and  $X_n$  is  $A_n$  then  $Y$  is  $B$ .**

$X_1, \dots, X_n$  sú fuzzy premenné ktoré môžu dosahovať fuzzy hodnoty reprezentujúce jazykové výrazy,

$A_1, \dots, A_n, B$  sú jazykové výrazy, napr. Malý, Stredný, Veľký.

Postup riešenia úlohy je založený na predpokladoch:

- riziko straty života pre osoby predstavuje požiar (dym),
- na záchranu sú dôležité prvé minúty od vypuknutia požiaru,
- je nevyhnutné využiť prvé minúty od vzniku požiaru na proces samozáchranu,
- samozáchrana je proces zložený z fáz: fáza detekcie a informovania, fáza reakcie a fáza vlastného úniku,
- vzhľadom k neprítomnosti zložiek IZS v prvých minútach od vzniku požiaru, je nutné, aby tunelový systém pomohol skrátiť časy jednotlivých fáz,
- dĺžku času úniku ovplyvňuje hlavne vzdialenosť únikových východov a kvalita únikových ciest, nedá sa vo väčšej miere dodatočne upravovať,
- skrátenie času evakuácie je možné na úkor skrátenia času detekcie, času informovania a času reakcie,
- šírenie dymu v smere unikajúcich osôb je dané fyzikálnym 2D modelom.

Výsledne riešenie relatívne objektívne, vďaka expertne formulovaným rovniciam fuzzy systému, hodnotí subsystémy tunela z hľadiska ich cieľových funkcií: rýchlosť identifikácie požiaru (30-180 s.), schopnosti informovania (pokrytie 0-100%) a dĺžku úniku (1-6 min.) [4], [6].

### 3. Tvorba fuzzy expertného systému

Hodnotiť prínos technologického vybavenia k bezpečnosti účastníkov v cestnom tuneli je vo všeobecnosti veľmi zložitý problém, ktorý sa nedá riešiť bežnými matematickými postupmi. To z toho dôvodu, že nie sú k dispozícii dáta a štatistické údaje, ktoré by umožňovali posúdiť ako sa budú meniť schopnosti záchranu osôb v závislosti od zmien parametrov technologického vybavenia. Jedna z možností, ako tento problém vyriešiť je použitie metód umelej inteligencie, konkrétne fuzzy lingvistický model systému. Čím viac sa na jeho tvorbe sa podieľajú experti tým je model presnejší. Značnou výhodou je, že je dobre kontrolovateľný a vďaka

tomu sa znižuje stupeň neurčitosti. Na strane vstupných informácií sú základné parametre súvisiace s technologickým vybavením. Použitím fuzzy lingvistickej aproximácie sú získané výstupné dáta. Výstupom algoritmu je číselné vyjadrenie prírastkov časov jednotlivých fáz procesu samozáchranu. Tunely s maximálnym technologickým vybavením poskytujú užívateľom vysokú bezpečnosť a časy potrebné na detekciu a reakciu osôb sú nízke. So zhoršením vybavenia tento čas narastá a klesá bezpečnosť tunela. Výstupná hodnota môže slúžiť na analýzu nákladov a prínosov. Vzhľadom k tomu, že pre potreby nášho modelu nie je vo väčšine prípadov možné opísať výsledok jednotlivých pravidiel v podobe lineárnej alebo inej kombinácie vstupných parametrov, je vhodné použiť fuzzy systém typu Mamdani. V takom prípade je výstupná premenná  $y$  definovaná na univerze  $Y$  a vstupné premenné  $x_i$  na univerzoch  $X_i$ . Nech univerzum  $Y$  je pokryté súborom fuzzy množiny  $B^i$  a univerzá  $X_i$  súbormi množín  $A^i_j$ . Potom sa dá nelineárna funkcia  $f$  aproximovať fuzzy systémom so súborom  $r$  pravidiel nasledujúceho typu:

If ( $x_1 = A_1^{jk}$ ) and ...and ( $x_n = A_n^{jk}$ ) then ( $y = B^{jk}$ ),  $k = 1, 2, \dots, r$

Tento typ fuzzy systému je označovaný ako fuzzy systém so závermi (v závere každého pravidla je výrok).

### 3.1 Nastavenie lingvistických premenných

Expertný systém na odhad času reakcie osôb je tvorený tromi oddelenými fuzzy inferenčnými systémami:

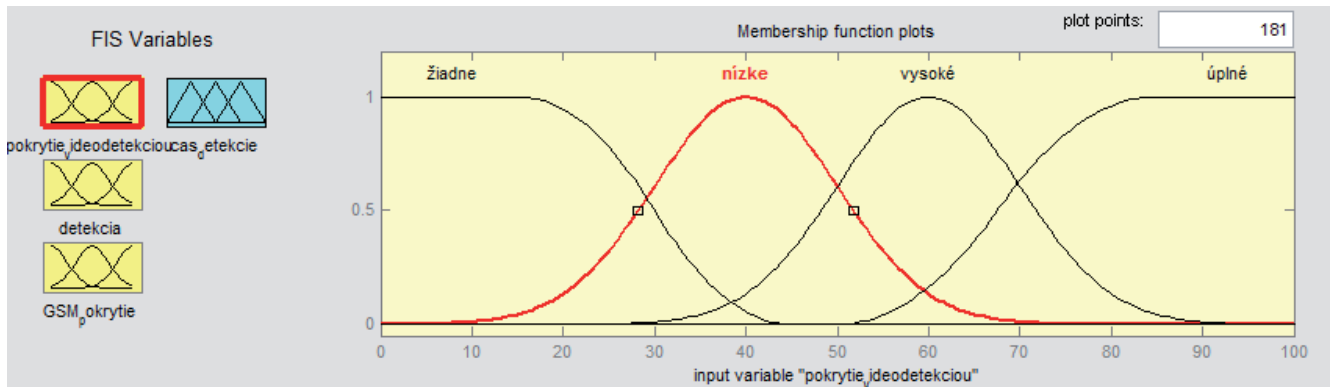
- na odhad času detekcie,
- na odhad času potrebného na informovanie osôb,
- na odhad času potrebného na začatie evakuácie.

V tomto článku bližšie opíšeme nastavenia jednotlivých lingvistických premenných v systéme na odhad času detekcie mimoriadnej udalosti. Na strane vstupov sú premenné „Pokrytie systémom videodetekcie“, „Rýchlosť detekcie lineárneho hlásiča požiaru“ (záleží od výšky umiestnenia, možnosti hodnotiť teplotný gradient a teplotnú diferenciu) a „GSM pokrytie“. Na strane výstupu sa nachádza premenná „Čas detekcie“.

Vstupná premenná „pokrytie systémom videodetekcie“ (obr.3) je percentuálnym vyjadrením pokrytia tunela systémom videodetekcie. Definícia lingvistickej premennej je nasledujúca: Množina termov lingvistickej premennej je: {žiadne, nízke, vysoké, úplné}. Termy sú definované na univerze 0 – 100 %. Priebeh jednotlivých funkcií príslušnosti je zastúpený klasickou Gaussovou distribučnou krivkou s maximami v hodnotách 40 % a 60 %, Z krivkou s maximom v intervale 0 – 15 % a S krivkou s maximom na hodnote 85 %.

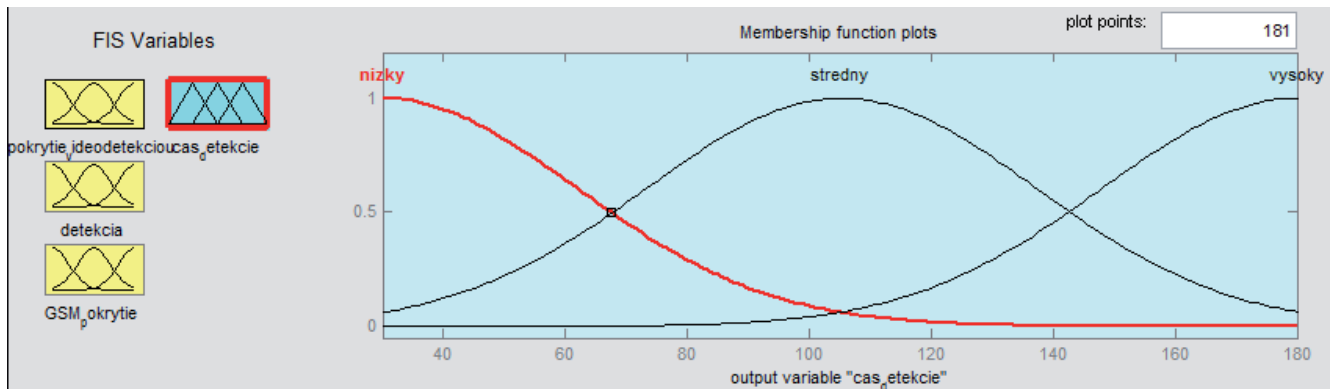
Vstupná premenná „Rýchlosť detekcie lineárneho hlásiča“ reprezentuje rýchlosť detekcie lineárneho hlásiča, čo je závislé od výšky jeho umiestnenia, nastavenia teploty poplachu, možnosti hodnotenia teplotného gradientu a možnosti hodnotenia teplotnej diferencie. Množina termov lingvistickej premennej je: {veľmi rýchla, rýchla, pomalá, veľmi pomalá}. Termy sú definované na univerze 30 – 180 s. Priebeh jednotlivých funkcií príslušnosti je zastúpený klasickou Gaussovou distribučnou krivkou s maximami v hodnotách 90 s. a 120 s., Z krivkou s maximom 0 – 52,5 s. a S krivkou s maximom na hodnote 157,5 s. Vstupná premenná „GSM pokrytie“ udáva percentuálne

pokrytie tunelovej stavby signálom technológie GSM. Priebeh jednotlivých funkcií príslušnosti sú zastúpené klasickými Gaussovými distribučnými krivkami s maximami v hodnotách 40 % a 60 %, Z krivkou s maximom na intervale 0 – 15 % a S krivkou s maximom v hodnote 85 % [10].



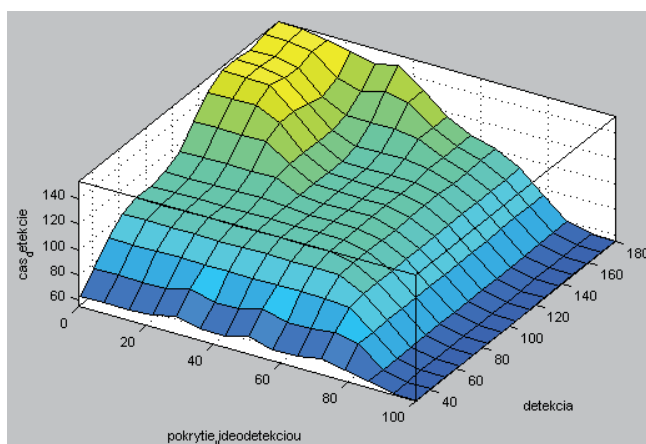
Obr. 3 Vstupná premenná „Pokrytie tunela systémom videodetekcie“

Fig. 3 Function shapes for IF-part of variable „Coverage of video detection system“



Obr. 4 Výstupná premenná „Čas detekcie“

Fig. 4 Function shapes for THEN-part of variable „Detection time“



Obr. 5 Závislosť času detekcie od rýchlosti detekcie lineárneho hlásiča a pokrytia systémom videodetekcie

Fig. 5 Detection time dependence on detection time of linear detector and coverage of videodetection system

Výstupná premenná „Čas detekcie“ udáva hodnotu času potrebného na detekciu požiaru. Množina termov lingvistickej premennej je: {nízky, stredný, vysoký}. Termy sú definované na univerze 30– 180 s. Priebehy jednotlivých funkcií Závislosť výstupnej premennej „Čas detekcie“ od vstupných premenných „Pokrytie systémom videodetekcie“ a „Rýchlosť detekcie lineárneho hlásiča“ je znázornená na obr. 5. Vstupnú premennú „GSM pokrytie“ sme nastavili ako konštantu na hodnotu 50 %, čo pre lingvistickú premennú predstavuje hodnotu „stredné“ pokrytie. Analogicky boli nastavené aj vstupné a výstupné premenné ostatných dvoch subsystémov. príslušnosti sú zastúpené klasickými Gaussovými distribučnými krivkami s vrcholmi v hodnotách 30, 110 a 180

s. (obr. 4). Inferenčný mechanizmus bol nastavený typickým expertným spôsobom. Jednotlivé pravidlá sú kombináciou nastavenia opísaných vstupných lingvistických premenných, definované na univerze 30– 180 s. Priebehy jednotlivých funkcií Závislosť výstupnej premennej „Čas detekcie“ od vstupných premenných „Pokrytie systémom videodetekcie“ a „Rýchlosť detekcie

pokrytie sys. video.	rých. detek. LHP	GSM pokrytie	čas detekcie
žiadne 0%	veľmi rýchla	nízke 0-30%	nízky
žiadne 0%	veľmi rýchla	stredné 30-60%	nízky
žiadne 0%	veľmi rýchla	vysoké 60-100%	nízky
žiadne 0%	rýchla	nízke 0-30%	stredný
žiadne 0%	rýchla	stredné 30-60%	stredný
žiadne 0%	rýchla	vysoké 60-100%	stredný
žiadne 0%	pomalá	nízke 0-30%	stredný
žiadne 0%	pomalá	stredné 30-60%	vysoký
žiadne 0%	pomalá	vysoké 60-100%	vysoký
žiadne 0%	veľmi pomalá	nízke 0-30%	stredný
žiadne 0%	veľmi pomalá	stredné 30-60%	vysoký
žiadne 0%	veľmi pomalá	vysoké 60-100%	vysoký
nízke 0-50%	veľmi rýchla	nízke 0-30%	nízky
nízke 0-50%	veľmi rýchla	stredné 30-60%	nízky
nízke 0-50%	veľmi rýchla	vysoké 60-100%	nízky
nízke 0-50%	rýchla	nízke 0-30%	stredný

Tab. 1 Inferenčné pravidlá systému na odhad času detekcie

lineárneho hlásiča“ je znázornená na obr. 5. Vstupnú premennú „GSM pokrytie“ sme nastavili ako konštantu na hodnotu 50 %, čo pre lingvistickú premennú predstavuje hodnotu „stredné“ pokrytie. Analogicky boli nastavené aj



vstupné a výstupné premenné ostatných dvoch subsystémov. príslušnosti sú zastúpené klasickými Gaussovými distribučnými krivkami s vrcholmi v hodnotách 30, 110 a 180 s (obr. 4). Inferenčný mechanizmus bol nastavený typickým expertným spôsobom. Jednotlivé pravidlá sú kombináciou nastavenia opísaných vstupných lingvistických premenných. Tab. 1 zobrazuje niekoľko inferenčných pravidiel systému

## Záver

Navrhnutý deterministický prístup umožňuje analyzovať rôzne scenáre a tunely s rôznymi parametrami. Vychádza zo základného predpokladu, že o prežití postihnutých osôb sa rozhoduje v prvých minútach po vypuknutí požiaru. Osobám musí byť poskytnutá šanca na samozáchranu. Okrem vzdialenosti k únikovému východu rozhoduje o skrátení času na únik aj skoré a relevantné informovanie o nutnosti evakuácie.

Navrhnutá metóda, pomocou metód soft computingu, hodnotí kvalitu technologického vybavenia tunela vo forme skrátenia času potrebného na zistenie nebezpečnej udalosti a času na reakciu. Všeobecne sa tento čas odhaduje na 3,5 minút. Pri kvalitnom a rýchlom informovaní sa tento čas dá skrátiť a použiť na únik osôb.

## PodĎakovanie

*Táto 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.**



**Agentúra**  
Ministerstva školstva, vedy, výskumu a športu SR  
pre štrukturálne fondy EÚ

*„Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ“*

## Literatúra

- [1] DIRECTIVE 2004/54/EC of European parliament and of the Council: Minimum Safety Requirements for Tunnels in the Trans-European Road Network, Brusels, 2004
- [2] Nariadenie vlády SR č. 344/2006 Z. z. o minimálnych bezpečnostných požiadavkách na tunely v cestnej sieti, 2006
- [3] Technické podmienky TP 02/2011, Analýza rizík pre slovenské cestné tunely, 2011
- [4] PŘIBYL, P., JANOTA, A., SPALEK, J.: Analýza a řízení rizik v dopravě, Praha, BEN-technická literatúra, 2008, ISBN 978-80-7300-214-5

na odhad času detekcie. Celkovo bolo definovaných 48 inferenčných pravidiel. Vzhľadom na praktické použitie fuzzy modelu sa všeobecne odporúča urobiť nastavenie pravidiel súčinnosťou viacerých nezávislých expertov [11], [12].

[5] Technické podmínky MD ČR TP98, Technologické vybavení tunelů pozemních komunikací, 2003

[6] Technické podmínky MD ČR TP229, Bezpečnost v tunelech pozemních komunikací, 2010

[7] SEBESTYÉNOVÁ J.: Simulácia evakuácie osôb pri požari, Posterus, 2012, ISSN 1338-0087

[8] PONIŠT, S.: Význam inteligentných dopravných systémov v doprave, ISSN 1336-5851

[9] KACHMAN, D.: Konceptia inteligentných dopravných systémov, Novus Scientia, 2007

[10] NOVÁK, V. : Fuzzy množiny a jejich aplikace, SNTL, Praha 1986

[11] SPALEK, J., JANOTA A., BALAZOVIČOVÁ M., PŘIBYL P.: Rozhodovanie a riadenie s podporou umelej inteligencie, Žilina, EDIS, 2005, ISBN 80-8070-354-X

[12] GULLEY N., ROSER JANG J. S.: Fuzzy Logic TOOL-BOX For Use with MATLAB

[13] DADO, M., ZAHRADNÍK, J.: Technológie a služby inteligentnej dopravy. Žilina, EDIS, 2007, ISBN 978-80-8070-691-3

[14] KŘOVÁK J., ZAMRAZILOVÁ E.: Expertní odhady, Praha, SNTL, 1989, ISBN 04-306-89

[15] ONISAWA T., KACPRZYK J.: Reliability and Safety Analyses under Fuzziness, Physica-Verlag A Springer-Verlag Company, 1995, ISBN 3-7908-0837-7

## Abstract

The paper deals with the process of evacuation after an incident in a road tunnel. The time required for evacuation we were divide to time intervals according activities of people. We use fuzzy expert system to calculate evacuation time. We focused on setting fuzzy inference system to estimate detection time.

**Ing. Peter Matis, prof. Ing. Juraj Spalek, PhD.**

Žilinská univerzita  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 1  
010 01 Žilina  
peter.matis@fel.uniza.sk, juraj.spalek@fel.uniza.sk

# Integrácia bezpečného riešenia LDWS s použitím VANET siete v inteligentných dopravných systémoch

Emília Bubeníková, Mária Franeková, Ján Beňuš

## Abstrakt

Príspevok sa zaoberá prepojením asistenčnej služby pre detekciu vybočenia z jazdného pruhu (LDWS systém) na C2C jednotku, ktorá aktívne monitoruje stav pohybových vlastností vozidla s použitím siete VANET. Popisuje možné riešenia hľadania vodorovného dopravného značenia na ceste pomocou segmentačnej metódy založenej na Houghovej transformácii. Realizovaný algoritmus v programovom prostredí Matlab deteguje vybočenie vozidla z jazdného pruhu, vytvára výsledný alarm, ktorý je v podobe autorizovanej správy prenášaný medzi vozidlami prostredníctvom komunikácie C2C.

**Kľúčové slová:** inteligentné dopravné systémy, aplikácie zamerané na bezpečnosť, cestná doprava, LDWS systémy, VANET siete, pseudoanonymná autorizácia zdroja správ, digitálny podpis, Houghova transformácia, certifikačná autorita, SW simulácia

## Úvod

Aktuálny vývoj v oblasti cestných telematických a asistenčných systémov vytvára nové možnosti v oblasti zvýšenia efektivity a bezpečnosti cestnej premávky. Použitie týchto technológií je veľmi široké, niektoré typy systémov sú už na trhu, iné vo fáze vývoja. O stave vývoja a nasadzovania aplikácií informujú vývojové tímy na medzinárodných kongresoch IDS (inteligentných dopravných systémov), napr. [1]. Hlavným cieľom je vybudovanie európskej stratégie na zrýchlenie výskumu a vývoja a použitia inteligentných integrovaných bezpečnostných systémov IISS (Intelligent Integrated Safety Systems,) vrátane pokročilých systémov asistencie vodiča ADAS (Advanced Driver Assistance Systems) z dôvodu zvýšenia bezpečnosti na cestách v Európe. K dôvodom patrí zvyšujúci sa počet nehôd a skutočnosť, že príspevky k bezpečnosti dosiahnuteľné štandardnými bezpečnostnými opatreniami, už dosiahli svoje hranice. Od vzniku automobilov sa zvyšovala aj ich bezpečnosť. Mnoho automobiliek vyvíja vlastné systémy ochrany pasažierov ale i automobilov ako majetku. Mnohé bezpečnostné prvky síce redukovali zranenia a následky po dopravnej nehode ale i napriek tomu štatistiky zranených a usmrtených osôb pri nehodách neboli optimistické. Výrobcovia automobilov sa preto snažili nielen redukovať následky nehôd ale najšť spôsob ako im predchádzať. Protikolízny systém je neoddeliteľnou súčasťou autonómneho správania sa automobilu a jeho história siaha až do roku 1933, keď Norman Bel Geddes predstavil na svetovej výstave vozidlo poháňané elektrickými obvodmi vsadenými do vozovky, ktoré bolo riadené rádiovými. Protikolízne systémy patria medzi systémy ADAS a môžu byť rozdelené do štyroch kategórií:

- Systémy pre aktívnu bezpečnosť - adaptívne predné osvetlenie, rozpoznávanie značiek pri ceste, varovanie pred čelnou kolíziou a zmiernovanie následkov nehôd.
- Systémy na podporu vodiča - upozornenie vodiča na prekročenie rýchlosti, varovanie pred veľkou rýchlosťou v zatáčke, varovanie pri vybočení z jazdného pruhu.

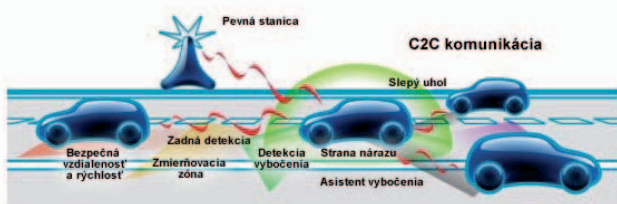
- Systémy zabezpečujúce informácie pre vodiča - informácia o rýchlosti, varovanie pred prekážkou v mŕtvom uhle.
- Systémy zabezpečujúce energetickú účinnosť - autonómny tempomat, údaje o účinnosti hnacieho ústrojenstva, poradca spotreby paliva, ekologická jazda.

Tieto systémy snímajú okolie automobilu a snažia sa identifikovať vznikajúce nebezpečné situácie a reagovať na ne prostredníctvom VC komunikácie (Vehicular Communications). Častokrát používajú ako vstupné dáta obrazové údaje získané z kamerových systémov umiestnených na vhodných miestach v automobile, ako sú spätné zrkadlá alebo čelné sklo.

Systém LDWS (Lane Departure Warning System) je systém upozornenia pred neúmyselným vybočením z jazdného pruhu na diaľniciach a cestách im podobných [2]. Faktor únavy a nepozornosti pri riadení vozidla hrá veľkú úlohu v počte dopravných nehôd. Podľa prieskumov a vypracovaných štúdií Svetovej zdravotníckej organizácie (WHO) sa až 80% dopravných nehôd v USA a Európe pripisuje krátkodobej nepozornosti vodiča v dobe 3 sekundy pred nehodou [29]. Štúdia Daimler Benz ukázala, že iba 0,5 sekundy času naviac pred zrážkou by odvrátilo 60% nehôd spôsobených nárazom zozadu a 1,35 sekundy by odvrátilo 90 % z nich [3]. V súčasnosti je pre IDS definovaných 32 služieb, ktoré sú kategorizované do 8 oblastí [3]. Medzi používateľské služby, ktoré súvisia priamo s vozidlom možno zaradiť zlepšovanie rozhľadu, automatické riadenie vozidla, vyhýbanie sa pozdĺžnym kolíziám počas jazdy, vyhýbanie sa priečnym kolíziám počas jazdy, pohotovostná bezpečnosť a obmedzovanie prednehodových stavov.

V súčasnosti je aktuálna problematika integrácie ponúkaných služieb IDS do čo najväčšieho celku. Pri realizácii týchto cieľov je nutné vytvoriť komunikačné spojenia medzi inteligentnými vozidlami s použitím štandardov pre bezdrôtovú komunikáciu, tzv. VANET (Vehicular Ad Hoc Networks) siete, kde je ale potrebné zabezpečiť overenie pravosti poskytovaných služieb. Na obr. 1 je naznačený princíp komunikácie medzi jednotlivými vozidlami - C2C (Car-to-Car)

komunikácia alebo medzi vozidlom a pevnou stanicou - C2I (Car-to-Infrastructure) komunikácia. Pre takúto komunikáciu je najvhodnejšia komunikácia krátkeho dosahu DSRC (Dedicated Short Range Communications).



Obr.1 Princíp komunikácie medzi vozidlami prostredníctvom VANET siete

Fig.1 Principle of communication between vehicles via VANET

Na takomto druhu projektov sa dnes podieľa viacero výskumných skupín po celom svete, napr. CARLINK, SeVe-Com, CAR 2 CAR a iné.

Postupná implementácia aplikácií riadenia cestnej premávky s použitím VANET sietí dokáže zefektívniť takmer všetky oblasti cestnej dopravy.

Pre komunikáciu medzi vozidlami sa v súčasnosti vyvíjajú technológie WAVE (Wireless Access in Vehicular Environments), CALM (Communications Architecture for Land Mobile Environment), C2C-CC (Car-to-Car Communication Consortium), pričom pre aplikácie zamerané na bezpečnosť má najväčšiu podporu práve technológia C2C [4].

Autori sa v príspevku zaoberajú len integráciou asistenčnej služby pre detekciu vybočenia z jazdného pruhu (LDWS), ktorú možno zaradiť medzi aplikácie so zameraním na bezpečnosť ale i s prepojením na reakciu a zaslanie autorizovanej správy ostatným vozidlám prostredníctvom siete VANET. Nosná časť príspevku je sústredená na overenie návrhu SW aplikácie detekcie jazdného pruhu založenej na Houghovej transformácii v SW nástroji Matlab a na analýzu možnosti prenosu autorizovanej správy s použitím kryptografických schém digitálneho podpisu.

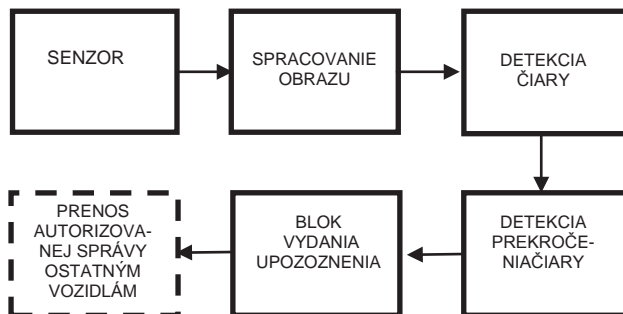
Navrhovaný proces spolupráce LDWS s komunikáciou prostredníctvom siete VANET predpokladá tieto neoddeliteľné súčasti:

- Proces detekcie – aplikácia LDW je autonómna. Autori príspevku predpokladajú, že vozidlo je schopné samostatne rozoznať určité nebezpečenstvo. Navrhovaná aplikácia detekcie jazdného pruhu pracuje na báze Houghovej transformácie v SW nástroji MATLAB. Možné ďalšie spôsoby detekcie prekročenia jazdného pruhu sú riešené v prácach [5], [6], [7].
- Proces šírenia správy – po vzniku nebezpečenstva sú vytvorené dva typy správ. Najskôr je vytvorená varovná správa (warning message). Vzniká kedykoľvek, keď vozidlo deteguje nebezpečenstvo prekročenia jazdného pruhu a upozornenie je šírené do ostatných vozidiel. Po odznení rizika musí byť vytvorená odvolacia správa (revocation message).
- Rozhodovací proces – vozidlo musí dokázať vyhodnotiť, ktoré správy sú a ktoré nie sú pre neho dôležité pred vstupom do potenciálne nebezpečnej oblasti.
- Proces riešenia bezpečnosti poskytovanej služby - spôsoby ako zabrániť útokom alebo manipulácii so správami. Autori riešia problém s použitím prvkov kryptografie, so zameraním len na autentifikáciu a integritu správy, tzn. ochranu správy pred jej pozmenením, s možnosťou identifikácie jej odosielateľa.

## 1. Návrh systému na detekciu čiar v obraze s prepojením na C2C komunikáciu

Prístupmi vo vývoji LDWS systémov sa možno stretnúť napr. v [8], [9], [10]. Princípy vychádzajú z použitia nasledujúcich spôsobov číslícového spracovania obrazu: stereovízie, multidimenzionálnej Fourierovej transformácie, Kalmanových filtrov, morfológických operácií a pod.

Navrhovaná softvérová aplikácia systému, ktorý integruje podsystem detegujúci prekročenie čiar do systému C2C prostredníctvom siete VANET pozostáva z nasledovných blokov (obr. 2).



Obr.2 Návrh systému na vyhodnocovanie prekročenia čiar a jeho prepojenie na C2C komunikáciu

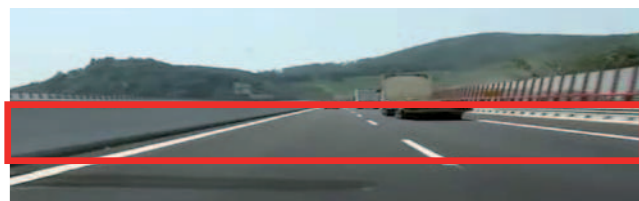
Fig.2 Design of system for evaluation of lane crossing and its connection to C2C communications

### 1.1 Senzor

Prostredníctvom senzora v podobe digitálnej kamery umiestnenej vo vozidle sa zosníma obrazová informácia do vhodného zvoleného obrazového formátu (v našej aplikácii bol použitý formát videosúboru .AVI).

### 1.2 Blok spracovania obrazu

V bloku spracovania obrazu dôjde k predspracovaniu obrazovej informácie aplikáciou nástrojov na odstránenie šumov v obraze. Predspracovaním sledujeme dva primárne ciele: skrátenie času analýzy a zvýšenie jej kvality z pohľadu danej aplikácie. Čas potrebný na analýzu celej sekvencie bol redukovaný odstránením nepotrebných nasnímanej plochy (priestorová redukcia – na každej snímke bola zvolená len definovaná oblasť záujmu ROI (Region of Interest), v ktorej sa s najväčšou pravdepodobnosťou predpokladá prítomnosť detegovaného objektu a bude použitá na ďalšie spracovanie (pozri obr. 3). Tým, že sa spracováva len časť obrázku nároky na spracovanie sa znižujú.



Obr.3 Vyčlenenie zóny záujmu

Fig.3 Allocation of zone interest

### 1.3 Blok detekcie čiar

V bloku detekcie čiar sa vyhľadáva jazdný pruh segmentnými technikami vhodnými pre detekciu čiar [11]. Informácia o polohe jazdných pruhov je získaná na základe hranovej detekcie vodiacich čiar. Na detekciu čiar je použitá metóda Houghova transformácia (podrobnejší opis realizovaného algoritmu je uvedený v časti 3.1).

## 1.4 Blok na vydanie upozornenia

Následne sú nájdené čiary zlúčené do jednej a prostredníctvom algoritmu detekcie prekročenia čiary, ktorý je založený na hľadaní prípadných priesečníkov nájdených čiar vodorovného dopravného značenia s modelom auta, ktorý predstavuje vodorovnú čiaru s definovanými rozmermi je zobrazené vizuálne varovanie pre vodiča, tzv. alert.

## 1.5 Blok prenosu autorizovanej správy

Vytvorené varovanie vyšle vozidlo ďalej autorizovaným spôsobom v podobe bezpečnostne relevantnej správy najbližšiemu vozidlu (správa typu beacon). Pre vytvorenie takéhoto druhu výstrahy musí každý paket typu beacon obsahovať informácie o stave odosielateľa ako sú pozícia vozidla, rýchlosť a smer. Tieto správy sú digitálne podpísané spolu s pripojeným certifikátom odosielateľa. Bezpečný beacon a bezpečná výmena správ dovoľujú vozidlám odhľadať stav svojho okolia a svojich fyzických susedov.

## 2. Možnosti autorizácie správ prenášaných medzi vozidlami

Z pohľadu komunikačnej bezpečnosti platia pre VANET siete podobné princípy ako pre iné bezdrôtové siete. Medzi hlavné požiadavky na bezpečnú komunikáciu medzi vozidlami patrí autentizácia správy a jej integrita, nepopierateľnosť správy, aktuálnosť správy, kontrola prístupu a dôvernosť správy. Tieto služby sa realizujú vo väčšine prípadov na báze kryptografických techník. Autori sa zameriavajú len na službu nepopierateľnosti správy a zabezpečenia aktuálnosti správy, ktorá používa techniky kryptografických digitálnych podpisov v kombinácii s časovými pečiatkami.

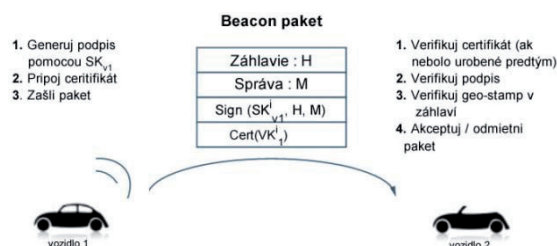
V rámci autorizovanej komunikácie medzi vozidlami sa presadila technika digitálnych podpisov na báze asymetrickej kryptografie s použitím PKI (Public Key Infrastructure), ktorá vychádza z predpokladu existencie CA (certifikačných autorít). Každá CA je zodpovedná za svoj priradený región v cestnej infraštruktúre (má prevažne národný charakter) a zabezpečuje identifikáciu všetkých pohyblivých uzlov (vozidiel), ktoré sú u nej registrované, ako aj správu krátkodobých kľúčov (K) a pridelenie certifikátov (Cert). Pre povolenie komunikácie medzi uzlami z rozdielnych regiónov musí CA poskytovať certifikáty aj pre ostatné CA (cross certification). Certifikáty vydáva CA po registrácii uzla v režime „offline“, na základe jednoznačného identifikátora ID vozidla. Pridelený certifikát po počiatočnej registrácii má dlhodobý charakter. V certifikáte sú uvedené atribúty uzla a trvanie certifikátu. Z pohľadu zabezpečenia služby autorizácie správ si treba uvedomiť, že ide o autorizáciu pohybujúcej sa skupiny uzlov (vozidiel) a bezpečnostne relevantná správa môže byť zasielaná skupine vozidiel, to znamená, že ide o viacnásobnú autentizáciu.

Podľa zvoleného typu schémy digitálneho podpisu je každému uzlu pridelený výrobcom pár kryptografických kľúčov, tzv. verejný a súkromný kľúč vozidla  $K=\{VK_V, SK_V\}$ . Ide o dlhodobé kľúče. Na verejný kľúč vozidla  $VK_V$  udeľuje CA dlhodobý certifikát. Súkromné kľúče sú uložené vo vozidle, v tzv. bezpečnostnom module HSM (Hardware Security Module), ktorý zároveň poskytuje bezpečnú časovú základňu pri generovaní časových pečiatok pre digitálny podpis. Modul HSM má za úlohu aj správu všetkých kryptografických operácií s kľúčmi. V prípade ohrozenia týchto citlivých informácií by mali byť tieto informácie z HSM vymazané. V schémach digitálneho podpisu sa na rozdiel od bežnej komerčnej sféry, napr. v bankovníctve pri autorizácii správ v rámci VC sietí využíva pseudoanonymná identifikácia, aby sa zaistila bezpečnosť a zároveň aj anonymita majiteľa vozidla. Pre získanie pseudonymu sa vo vozidle generuje sada kľúčových párov a verejné kľúče sú zaslané do prí-

slušnej CA cez zabezpečený komunikačný kanál. Certifikačná autorita následne podpíše každý z verejných kľúčov a vygeneruje sadu pseudonymov pre každé vozidlo. Každý pseudonym obsahuje identifikátor CA, životnosť pseudonymu, verejný kľúč a podpis CA a teda žiadnu informáciu o identite vozidla.

Frekvencia zmeny pseudonymov závisí od stupňa ochrany vozidla, vstupných parametrov (pozícia, rýchlosť) a nastavenia systému. Pre zabezpečenie ďalších pseudonymov sa používajú takzvané sady pseudonymov. Tieto pseudonymy sa periodicky dopĺňajú z CA. V okamihu, keď prejde uzol zo sady pseudonymov 1 na sadu pseudonymov 2 už ďalej nemôže používať žiadny pseudonym zo sady 1.

Pred zaslaním bezpečnostne relevantnej správy (v našej aplikácii – beacon packet) sa v bezpečnostnej jednotke vozidla  $V_1$  pomocou jeho súkromného kľúča  $SK_{V_1}$  vygeneruje digitálny podpis, ktorý je funkciou správy  $M$  ako aj záhlavia správy  $H$ , tak ako to je principiálne naznačené na obr. 4.



Obr.4 Prenos autorizovanej správy medzi vozidlami  
Fig.4 Transmission of authorised message between vehicles

Takto vytvorené kryptografické číslo sa pripojí ku správe, zároveň s certifikátom  $Cert$ , ktorý je previazaný na  $i$ -ty anonymný verejný kľúč odosielateľa  $VK_{V_1}^i$ , ktorý certifikuje príslušná CA. Na strane vozidla  $V_2$  sa najprv validuje prijatý certifikát (ak to nebolo urobené predtým) a verifikuje sa prijatý digitálny podpis pomocou  $i$ -teho verejného kľúča vozidla  $VK_{V_1}^i$ , ktorý si v pravidelných časových intervaloch sťahuje vozidlo  $V_2$  (resp. ostatné vozidlá). Zároveň sa z hlavičky  $H$  verifikuje informácia typu *geo stamp* a po absolvovaní týchto procedúr je bezpečnostne-relevantná správa akceptovaná alebo nie.

Proces generovania digitálneho podpisu vozidlom  $V_1$  možno matematicky vyjadriť:

$$V_1 \rightarrow * : M, H, \text{Sign}_{SK_{V_1}^i}[(M, H) | T], \text{Cert}_{VK_{V_1}^i}, \quad (1)$$

kde  $M$  predstavuje zasielanú bezpečnostne relevantnú správu,  $H$  predstavuje záhľavie správy,  $SK_{V_1}^i$  je krátkodobý súkromný kľúč vozidla  $V_1$  v  $i$ -tom okamihu,  $VK_{V_1}^i$  je krátkodobý verejný kľúč vozidla  $V_1$ , v  $i$ -tom okamihu  $T$  je časová pečiatka,  $Cert$  je krátkodobý certifikát vozidla  $V_1$  (pre anonymný verejný kľúč  $VK_{V_1}^i$ ),  $*$  predstavuje počet prijímačov (v prípade, ak by správa bola zasielaná viacerým vozidlám).

Aktuálny certifikát vozidla  $V_1$  platný v  $i$ -tom časovom okamihu pre anonymný verejný kľúč vozidla  $V_1$  ( $VK_{V_1}^i$ ) obsahuje:

$$\text{Cert}_{V_1}^i[VK_{V_1}^i] = VK_{V_1}^i | \text{Sign}_{SK-CA}[VK_{V_1}^i | ID_{CA}] \quad (2)$$

kde  $\text{Sign}_{SK-CA}$  predstavuje podpis certifikátu príslušnou certifikačnou autoritou na základe jej súkromného kľúča  $SK-CA$ ,  $ID_{CA}$  predstavuje jednoznačné identifikačné číslo príslušnej certifikačnej autority.

### 3. Praktická realizácia s overením návrhu systému

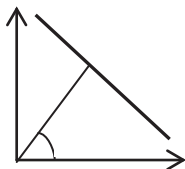
SW realizácia detekcie pohyblivého objektu (vozidla) na vozovke, so zameraním na detekciu jeho vybočenia z jazdného pruhu, bola realizovaná v SW nástroji MatlabR2010a v toolboxe Signal Processing. Autori zvolili segmentačnú metódu na báze Houghovej transformácie, ktorá je štandardnou funkciou softvérového nástroja. V prípade vybočenia vozidla je v rámci vozidlovej komunikácie autorizovaným spôsobom (ktorý je opísaný v časti 2) vysielaná bezpečnostne relevantná správa typu beacon.

#### 3.1 Použitie Houghovej transformácie v SW aplikácii detekcie čiar

Houghova transformácia (HT) je segmentačná technika často používaná v prípadoch, keď je potrebné detegovať objekty so známym tvarom hranice (čiara, kružnica a pod.) [12]. Táto metóda hľadá požadované tvary v triede tvarov pomocou procedúry hlasovania. Procedúra sa realizuje v príslušnom parametrickom priestore, z ktorého sú následne získané kandidátske objekty ako lokálne maximá v akumuláčnom priestore. Využíva princíp, že jedným bodom môže prechádzať nekonečne veľa priamok, každá v inom smere. Transformácia sa snaží identifikovať jednu z týchto priamok a to tú, ktorá najlepšie vystihuje údaje v obraze tým, že prechádza najväčším počtom bodov v obraze. Pri jej použití pri hľadaní priamok pomocou HT sa vychádza z normálovej rovnice priamky (3)

$$y = \frac{x \cdot \cos \Theta}{\sin \Theta} + \frac{r}{\sin \Theta}, \quad (3)$$

kde  $r$  predstavuje najmenšiu vzdialenosť priamky (v bode  $X[x, y]$ ) od počiatku súradnicovej sústavy,  $\Theta$  je veľkosť orientovaného uhla od kladnej x-ovej poloosi po polpriamku vedenú z počiatku súradnicovej sústavy kolmo na hľadanú priamku (obr.5). Použitie normáloveho tvaru rovnice priamky má tú výhodu, že je nezávislý na orientácii osí.



Obr.5 Hľadaná priamka s premennými z normálovej rovnice priamky

#### Fig.5 Seeking a line with variables from the normal line equation

Ako vstupné dáta sú použité súradnice pixelov  $x, y$ , ktoré tvoria priamku. Neznáme sú hodnoty  $r$  a  $\Theta$ . Do rovnice sú postupne vkladané súradnice pixelov, množina riešení  $r$  a  $\Theta$  vytvorí v Houghovom priestore spojitú krivku. Výsledky sa zapisujú do tzv. akumulátora. Každý bod priamky je transformovaný do diskretizovanej krivky  $(r, \Theta)$ , pričom hodnota akumulátorových buniek je pozdĺž tejto krivky inkrementovaná. Toto sa opakuje pre všetky pixely na priamke. Po vykreslení všetkých kriviek všetkých pixelov sa všetky pretnú v jednom bode  $B_{\max}(r_{\max}, \Theta_{\max})$ , čo sú vlastne parametre hľadanej priamky a v akumulátore sú potom súradnice maxima.

Vo väčšine prípadov sa pred samotnou transformáciou používa metóda predspracovania, ktorá umožní zvýrazniť hrany objektov. Tu možno použiť známe hranové detektory (napr. Cannyho, Sobelov) pre získanie binárneho obrazu, ktorý obsahuje body reprezentujúce hranice objektov.

Vstupným parametrom pre hľadanie čiar pomocou Houghovej transformácie je farebný výrez snímaného obrazu. Tento farebný obrazový výrez sa prevedie na šedotónový obraz definovaný podľa vzťahu (4):

$$Y = 0.299R + 0.587G + 0.144B, \quad (4)$$

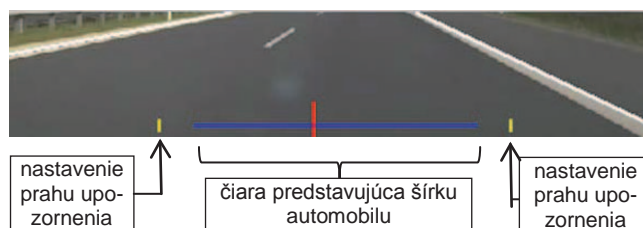
$$U = -0.147R - 0.289G + 0.436B,$$

$$V = 0.615R - 0.515G - 0.100B.$$

Ďalej obraz podľa vypočítaného prahu spracováva hranový operátor. Na hľadanie hrán bola v príspevku zvolená Sobelova metóda, pretože dosahuje uspokojivé výsledky za pomerne krátky čas.

#### 3.2 Realizácia detekcie prekročenia čiar

Detekcia prekročenia čiar v softvérovej realizácii funguje na princípe hľadania prieniku úsečiek. Cyklicky sa porovnáva poloha nájdených čiar (pomocou HT) vodorovného dopravného značenia na ceste s úsečkou, ktorá má presne stanovené hodnoty a reprezentuje automobil v obraze [13].



Obr.6 Princíp detekcie prekročenia čiar

#### Fig.6 Principle of line crossing detection

Prienik dvoch úsečiek v rovine je možné vo všeobecnej rovine zistiť pomocou nasledujúcej matematickej analýzy:

$$d(A, B, C) = x_A(y_B - y_C) + y_A(x_C - x_B) + x_B y_C - y_B x_C, \quad (5)$$

kde  $A, B, C$  sú body priamok sú súradnicami  $A[x_A, y_A]$ ,  $B[x_B, y_B]$ ,  $C[x_C, y_C]$ .

Funkcia  $d(A, B, C)$  nadobúda rôzne znamienko podľa toho, či bod leží v jednej alebo druhej polrovine určenej priamkou  $BC$ . Ak teda  $d(A, P, Q) \cdot d(B, P, Q) < 0$  priamka  $PQ$  oddeľuje body  $A, B$ .

Ak vyšetríme hodnoty:

$$s_A = d(A, P, Q), \quad s_P = d(P, A, B), \quad (6)$$

$$s_B = d(B, P, Q), \quad s_Q = d(Q, A, B).$$

Ak úsečky  $AB, PQ$  nemajú spoločný bod, tak platí:

$$s_A \cdot s_B > 0 \text{ alebo } s_P \cdot s_Q > 0. \quad (7)$$

To znamená, že sa nepretínajú a nedošlo k prekročeniu čiar.

### 4. Výsledky SW realizácie

Algoritmus, ktorý deteguje prekročenie čiar z nasnímaného farebného videozáznamu bol realizovaný v programovom prostredí Matlab v toolboxe Image Processing. Obrazové dáta boli nasnímané vo formáte .avi pomocou kamery umiestnenej do oblasti vnútorného spätného zrkadla motorového vozidla v rozlíšení 720x576 pixelov.

Vstupnými dátami pre testovanie algoritmu boli dopredu nasnímané a na harddisk počítača uložené videosekvencie priestoru pred automobilom. Algoritmus postupne načítava jednotlivé snímky videa. Keďže zábery videa zachytili aj pre vyhodnocovanie prekročenia čiar nepodstatné údaje (hori-

zont, vonkajší priestor zosnímanej kapoty vozidla), bolo potrebné každú snímku orezať. Oblasť záujmu bola vyčlenená ako úsek cesty nachádzajúci sa pred autom a bola stanovená empiricky na veľkosť cca 580x100 pixelov. Za predpokladu, že snímacia kamera bola pevne pripevnená na snímacom mieste, sú rozmer a pozícia výrezu vzhľadom na celkový záber videa tiež pevné [14], [15], [13]. Samotný farebný výrez sa prevedie na šedotónový (podľa vzťahu (4)) a podľa automaticky určeného prahu ho spracováva Sobelov hranový operátor. Výstupom po aplikovaní hranového operátora je binárny obraz, ako je možné vidieť na obr.7.



**Obr.7 Pôvodný obraz a binárny obraz po aplikovaní hranového operátora**

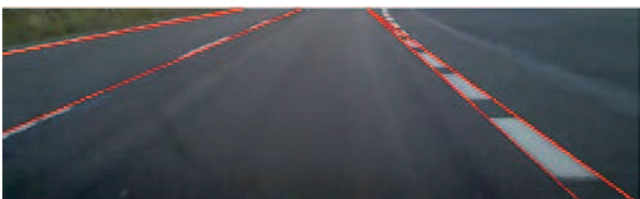
**Fig.7 The original image and the binary image after applying of edge operator**

Na binárny obraz sa aplikujú funkcie HT. Výstupom je štruktúra dát obsahujúca polohy začiatkových a koncových bodov priamok, vzdialenosť priamok od počiatku súradnicovej sústavy a veľkosť orientovaného uhla od kladnej x-ovej polosi po polpriamku vedenú z počiatku súradnicovej sústavy kolmo na hľadanú priamku. V situácii, keď sa na ceste nachádzajú prerušované čiary, tak je nájdených množstvo čiar. Na obr.8 sú červenou farbou označené všetky nájdené čiary po aplikácii HT. Ich parametre vstupujú do algoritmu spájania podobných čiar. Porovnanie nájde niekoľko favoritov, ktorých orientácia uhla zodpovedá najväčšiemu množstvu nájdených čiar v určitej stanovenej tolerancii, (predvolená je hodnota 5 stupňov) a určí jednu čiaru obr. 9.



**Obr.8 Čiary nájdené pomocou HT (pre ilustráciu sú zakreslené v pôvodnom obrázku)**

**Fig.8 Lines found using HT (for illustration are plotted in the original image)**



**Obr.9 Výsledné súvislé čiary po spájaní nájdených čiar po aplikácii hlasovača**

**Fig.9 Resulting continuous lines after connecting found lines with poller applied**

Výstupom sú nové parametre čiar, ktoré vstupujú do tzv. hlasovača. Hlasovač má za úlohu vyhodnotiť z posledných 10 snímok platnosť nájdenej čiar. Hodnotu pomeru hlasovača ( $n$  z 10) je možné meniť pred spustením programu.

Tieto dáta potom postupujú do ďalšieho spracovania, v ktorom sa prostredníctvom algoritmu, ktorého princíp bol spomenutý vyššie, deteguje prekročenie čiar a táto skutočnosť sa indikuje optickou informáciou v podobe písmena „A“ (alert) v obraze (obr. 10).



**Obr.10 Zobrazenie upozornenia v obraze**

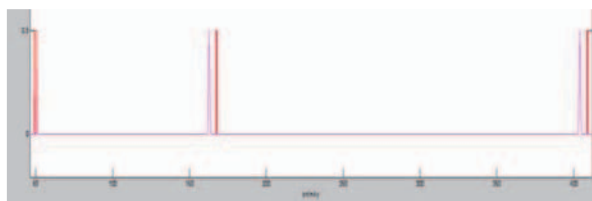
**Fig.10 Display of alarm in image**

Pre nájdenie optimálnych hodnôt parametrov niektorých vstupných premenných ako sú: citlivosť zlučovania čiar, pomer hlasov v hlasovači, nastavenie prahu upozornenia, bolo testovaných množstvo kombinácií nastavení vstupných premenných. Správnosť detekcie bola stanovená na základe porovnania vopred určených čísel rámcov, u ktorých bola subjektívne určená hodnota prekročenia čiar v skúmanom .AVI zázname. Tieto údaje boli následne porovnané s hodnotami, ktoré boli určené na tom istom zázname pomocou vytvoreného algoritmu. Vo zvolenom testovanom zázname došlo v reálnych podmienkach k trojnásobnému prekročeniu čiar. Pri niektorých hodnotách testovaných rozsahov vstupných parametrov vôbec nedošlo k detekcii prekročenia čiar alebo bola detekcia oneskorená až o niekoľko desiatok snímok.

Pri praktickom použití algoritmu je vhodné, aby došlo k detekcii prekročenia čiar a následnému vydaniu alertu vo forme obrazovej signalizácie ešte pred fyzickým prekročením čiar. Na základe tejto skutočnosti boli pre správnu detekciu hodnoty vstupných parametrov určené nasledovne:

- citlivosť zlučovania čiar (rozsah hodnôt 3-5);
- pomer hlasov v hlasovači (rozsah hodnôt 4-6);
- nastavenie prahu upozornenia (rozsahy hodnôt 10, 20).

Tieto hodnoty boli získané ako optimálne na základe porovnania subjektívneho prekročenia čiar v rámcoch s hodnotami 50, 168, 409 a vytvoreným algoritmom. Na obr. 11 je zobrazené porovnanie subjektívne určených hodnôt obrazových rámcov (červená farba) a fialových hodnôt rámcov určených algoritmom. Ako je z obrázku vidieť, algoritmom pre daný videozáznam boli určené čísla rámcov, v ktorých došlo k prekročeniu čiar 49, 165, 407, čo predstavuje úspešné a včasné detegovanie prekročenia čiar.



**Obr.11 Porovnanie subjektívne určených frame (červené čiary) a hodnôt rámcov, keď bol vykonaný alert (ružové čiary)**

**Fig.11 Comparison of subjectively determined frames (red lines) and frame values when an alarm has been triggered (pink lines)**

Po detekcii prekročenia jazdného pruhu, ak vodič má v riadiacej jednotke vozidla implementované bezpečnostné mechanizmy v rámci komunikácie, vyšle vozidlo varovanie v podobe bezpečnostne relevantnej správy najbližšiemu vozidlu. Ako už bolo zdôraznené základným nástrojom na zaistenie bezpečnej komunikácie v rámci vyvíjaných aplikácií IDS sú digitálne podpisy.

V súčasnosti sa v komerčnej sfére používajú nasledujúce schémy digitálneho podpisu, ktoré vychádzajú z asymetrickej kryptografie [16]:

- Schéma digitálneho podpisu s RSA algoritmom.
- Schéma DSA (Digital Signature Algorithm) s modifikovaným El Gamalovým algoritmom.

- Schéma ECDSA (Elliptic Curve Digital Signature Algorithms) s algoritmom eliptických kriviek.

Treba si uvedomiť, že proces realizácie fáz digitálneho podpisu na báze asymetrickej kryptografie je pre požadovanú bezpečnosť výpočtovo náročný problém, čo pri autorizovanom prenose správ medzi viacerými uzlami v rámci cestnej infraštruktúry môže spôsobovať časové oneskorenie v prijatí autorizovanej správy a následne v reakcii na vzniknutú udalosť. Preto je potrebné viazať sa síce na výpočtovo bezpečné schémy digitálnych podpisov, ale zároveň spĺňajúce požiadavky na rýchlosť výpočtu generovania a verifikácie podpisov [17].

Podľa výstupov uvádzaných v rámci riešenia projektov pre oblasť VANET sú v medzivozidlovej komunikácii odporúčané algoritmy eliptických kriviek, ktoré sa dajú použiť na šifrovanie/dešifrovanie prenášaných správ (algoritmus ECIES -Elliptic Curve Encryption Scheme) ako aj na tvorbu digitálnych podpisov. Preto odporúčame v integrovanom bezpečnom riešení LDWS systému použiť digitálny podpis ECDSA, ktorý dnes potrebuje podstatne menšiu dĺžku kľúča na dosiahnutie rovnakej úrovne bezpečnosti v porovnaní so schémou digitálneho podpisu RSA. Kryptografické systémy založené na eliptických krivkách so 160 bitovým kľúčom poskytujú rovnakú bezpečnosť ako RSA systémy s 1024 bitovým kľúčom a zároveň je u nich proces generovania podpisu omnoho rýchlejší v porovnaní s RSA schémami. Autori odporúčajú pre sledovanú aplikáciu použiť eliptické krivky nad konečným telesom  $F_p$ , kde  $p$  je nepárne prvočíslo, ktoré sú viac vhodné pre SW realizáciu.

Príklad postupu pre vytvorenie digitálneho podpisu vo vozidle pre správu dĺžky 200B (čo je podľa [18] používaná dĺžka paketu beacon) realizovaný v SW nástroji Cryptool [19] s algoritmom ECDSA s parametrami: 'EC-prime256v1' (dĺžka kľúča 256 bitov), hašovací funkcia SHA-1 (otlačok 160 bitov) je nasledovný:

#### Generovanie kľúča odosielateľa správy:

Doménové parametre eliptickej krivky  $E_p(a, b)$  z konečného telesa  $F_p$  ( $p=256$ ) s definovanou rovnicou  $y^2 = x^3 + ax + b$ .

$a =$   
1157920892103562487626974469494075735300861434152903141955  
33631308867097853948

$b =$   
4105836372515214212932612978004726840911444101599372555483  
5256314039467401291

Súradnice bodu  $G$  na eliptickej krivke [ $G_x, G_y$ ]:

$G_x =$   
4843956129390645175905258525279791420276294952604174799584  
4080717082404635286

$G_y =$   
3613425095674979579858512791958788195661110667298501507187  
7198253568414405109

Náhodné číslo  $k$  z intervalu  $(1, r-1)$ ;  $r$  je rád bodu  $G$ .  $k = 1$

$r =$   
1157920892103562487626974469494075735299969552241357603424  
22259061068512044369

#### Generovanie súkromného kľúča $SK = \{s\}$ :

$s =$   
5360461347100802301472767289333233164587335713400826903728  
2775009260129099968

Výpočet hašovacieho kódu zo správy  $M$ :

$f = 261352032232054377235764067545527908991888747218$

Tvorba verejného kľúča  $VK = \{u, v\}$ :

$u =$   
2964304138027375849776943500083924317112371168396960298692  
5689917739695548172

Súradnice bodu  $V$  na eliptickej krivke [ $V_x, V_y$ ]:

$V_x =$   
1565813079353998767197529237946007087365592172606061245429  
2820100262076517759

$V_y =$   
1011554232685778471146365626238695619503329032912132022726  
64533221061125756902

Konvertuj element konečného poľa  $V_x$  na číslo  $i$ :

$i =$   
1565813079353998767197529237946007087365592172606061245429  
2820100262076517759

Vypočítaj prvú časť podpisu:  $c = i \text{ mod } r$ ,

Vypočítaj druhú časť podpisu:  $d = u^{-1} \cdot (f + s \cdot c) \text{ mod } r$ .

Digitálny podpis tvorí dvojica:  $\{c, d\}$ :

$c =$   
1565813079353998767197529237946007087365592172606061245429  
2820100262076517759

$d =$   
7916348686409560006306567085585801249772925266595752958865  
0926725104868444034

**Podpísaná správa:**  $\{M, c, d\}$ .

Dĺžka podpisu je dvojnásobkom dĺžky kľúča.

Pozn.: Hodnoty uvádzané v príklade sú v hexadecimálnom tvare.

Po prenose správy sa v prijatých vozidlách realizuje verifikácia prijatej správy s digitálnym podpisom  $\{M, c, d\}$  pomocou nasledovného postupu:

Najprv sa overí, či  $c, d$  sú celé čísla z intervalu  $(1, r-1)$ .

Vypočíta sa hašovacie kódy  $f$  a hodnota  $w = (d')^{-1} \text{ mod } r$ .

Vypočíta sa  $u_1, u_2$ :  $u_1 = (f \cdot w) \text{ mod } r$ ;  $u_2 = c \cdot w \text{ mod } r$ .

Vypočíta sa:  $u_1 \cdot G + u_2 \cdot V = (x_0, y_0)$ ;  $v = x_0 \text{ mod } r$ .

Ak platí  $v = c$ , potom je podpis platný a správa bola zaslaná z autorizovaného zdroja.

## Záver

Cieľom tohto príspevku bol opis integrácie LDWS systému do vozidlovej komunikácie C2C. Nosná časť mapuje použitie segmentačných metód a algoritmov, ktoré dokážu z obrazovej informácie detegovať jazdné pruhy a ďalšími algoritmi vyhodnotiť, kedy vozidlo opúšťa svoj jazdný pruh. Vzniknutá situácia vedie k vygenerovaniu a zaslaniu autorizovanej varovnej správy okoloidúcim vozidlám, podpísanej digitálnym podpisom v rámci VC (vehicular networks). Uvedené metódy a algoritmy boli implementované v programovom prostredí MATLAB, Image Processing Toolbox. Príklad tvorby digitálneho podpisu v schéme ECDSA bol realizovaný v SW nástroji Cryptool.

Realizovaný algoritmus detekcie hrán využívajúci Houghovu transformáciu a Sobelov operátor hľadania hrán dokázali úspešne detegovať plné čiary, a to aj v prípade ich zlej kvality alebo nekvalitného obrazu. Úspešnosť metódy závisela od kvality čiar na cestách, vlastností snímaného obrazu a zvolených vstupných nastavení parametrov výpočtu. Metóda využívajúca Houghovu transformáciu dosiahla výborné výsledky aj pri nekvalitnom značení na ceste, problém jej ale robili krátke prerušované čiary.

Ďalej sa v realizovanej SW aplikácii predpokladá doplnenie algoritmu detekcie čiar o detekciu oranžových čiar, ktoré sa používajú pri dočasnom značení dopravnej komunikácie a majú prednosť pre bielym vodorovným značením.

Výber schémy digitálneho podpisu so zameraním na algoritmus eliptických kriviek ECDSA bol zvolený na základe porovnania efektívnosti podpisovaných schém, uvedených napr. v [19] vzhľadom k opisovanej aplikácii, kde je okrem bezpečnosti dôležitý parameter výkonnosť použitej digitálnej podpisovej schémy.

## Podakovanie

Táto publikácia vznikla s podporou Edukačnej grantovej agentúry Slovenskej republiky (KEGA) číslo: 024ŽU-4/2012: Modernizácia technológií a metód vzdelávania so zamera-

ním na oblasť kryptografie pre bezpečnostne kritické aplikácie (50 %).

Táto publikácia vznikla s priamou podporou projektu ITMS 26220120050 s názvom "Centre of excellence for systems and services of intelligent transport II" riešeného v rámci Agentúry Ministerstva školstva, vedy, výskumu a športu SR pre štrukturálne fondy EÚ (50 %).



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

## Literatúra

[1] 19. Intelligent Transportation Congress, Vienna, Austria, 22.–26. October, 2012. Dostupné na internete: <http://2012.itsworldcongress.com>

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

[3] SCHLOSSER, T.: Inteligentné dopravné systémy, Jaga group, ISBN 80-88905-64-8

[4] Car2Car Communication Consortium. <http://www.car-2-car.org/>

[5] ZOLOTOVÁ, I., KARCH, P.: Contribution to modification of Graph cut method and its implementation in the image segmentation, In.: International Journal of Circuits, Systems and Signal Processing, Volume 6, Issue 1, pp. 49-52, ISSN 19984464, 2012.

[6] YEH, H., CHEN, Y. H.: Development of vision-based lane and vehicle detecting system on PXA255 embedded system, International Journal of Circuits, Systems and Signal Processing, vol. 1, no. 2, pp. 177-181, 2007.

[7] YINGHUA, H., HONG, W., BO, Z: Color-Based Road Detection in Urban Traffic Scenes, In: IEEE Transactions on intelligent transportation systems, Vol.5, No 4, december 2004.

[8] DICKMANN, E.D., MYSLIWETZ, B. D.: Recursive 3-D road and relative ego-state recognition, Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 199-213, 1992.

[9] LeBLANC, D.J. et al: CAPC: A Road-Departure Prevention System," Control Systems Magazine, IEEE, vol. 16, no. 6, pp. 61-71, 1996.

[10] KREUCHER, C., LAKSHMANAN, L.: LAN a lane extraction algorithm that uses frequency domain features. Robotics and Automation, vol.15, no. 2, pp. 343-350, 1999.

[11] HLAVÁČ, V., SEDLÁČEK, M.: Zpracování signálů a obrazů, ČVUT Praha, ISBN 80-01-02114-9, 2002.

[12] BUBENÍKOVÁ, E., MUZIKÁŘOVÁ, L., HALGAŠ, J.: Application of Image Processing in Intelligent Transport Systems. In: 11 th IFAC/IEEE International Conference on Programmable Devices and Embedded Systems, ISBN: 978-3-902823-21-2, 2012.

[13] SLOTKA, A.: Obrazové segmentačné techniky pre dopravné aplikácie. Diplomová práca, Žilinská univerzita v Žiline, 2012.

[14] BUBENÍKOVÁ, E., SLOTKA, A., MUZIKÁŘOVÁ, L.: Využitie obrazových segmentačných techník pri detekcii

jazdného pruhu. In: Technolog: časopis pre teóriu a prax mechanických technológií – Nekonenčné technológie 2012, ISSN 1337-8996, roč.4, č.2, s. 99-104, 2012.

[15] BUBENÍKOVÁ, E.: Spracovanie obrazovej informácie v dopravných aplikáciách. Písomná časť dizertačnej skúšky, ŽU v Žiline, 2012.

[16] FIPS 186-2: Digital Signature Standard. NIST, 2000.

[17] FRANEKOVÁ, M.: Mathematical apparatus for safety evaluation of cryptography and safety codes used in safety related communication system. In: Modern transport telematics: 11th international conference on transport systems telematics, TST 2011, Katowice-Ustrón, Poland, October 2011, selected papers: Springer-Verlag, Berlin Heidelberg CCIS 104 - Communications in Computer and Information Science. ISBN 978-3-642-24659-3, pp. 126-135, 2011.

[18] <http://www.cryptool.com>

[19] DENG, J., CHENG, X., GUI, Q.: Design of Hyper Elliptic Curve Digital Signature. 2009. Dostupné na internete: [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=5190178](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5190178)

## Abstract

The paper deals with interconnection of assistance service to detect lane departure (LDWS) on C2C unit, which actively monitors the status of movement characteristics of the vehicle using VANET network. Describes possible solutions to searching road markings on the road using segmentation method based on the Hough transform. Algorithm implemented in Matlab detects crossing traffic lane, creates the final alarm, which is transmitted in the form of an authorized management between vehicles through C2C communication.

## Ing. Emília Bubeníková

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 1  
010 26 Žilina  
Tel.: 0421425133344  
[emilia.bubenikova@fel.uniza.sk](mailto:emilia.bubenikova@fel.uniza.sk)

## prof. Ing. Mária Franeková, PhD.

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 1  
010 26 Žilina  
Tel.: 0421425133346  
[maria.franekova@fel.uniza.sk](mailto:maria.franekova@fel.uniza.sk)

## Ing. Ján Beňuš

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 1  
010 26 Žilina  
Tel.: 0421425133321  
[jan.benus@fel.uniza.sk](mailto:jan.benus@fel.uniza.sk)



# Aplikácie v dopravnej telematike

Rastislav Pirník

## Abstrakt

Úlohou tohto príspevku je analyzovať, definovať a sumarizovať požiadavky na siete pripájajúce telematické technológie využívané v inteligentných dopravných systémoch (IDS). Článok na základe dostupných a platných štandardov opisuje koncepciu spracovania obrazovej informácie v hybridnom rozhraní technologických a otvorených sietí. Príspevok je zameraný spracovanie obrazovej informácie a jej tunelovanie cez rozšírenú metropolitnú prenosovú technológiu xDSL. Ďalej príspevok opisuje navrhnutý kompresný algoritmus, a jeho optimalizáciu a dosiahnuté výsledky ktoré boli vhodné pre potreby rozhraní medzi otvorenými a technologickými typmi sietí dopravnno-telematických systémov.

**Kľúčové slová:** IDS, videodetekcia, xDSL, dispečerské systémy

## Úvod

Pevný širokopásmový prístup – Broadband Fixed Access (BFA) je technológia slúžiaca na zabezpečenie vysokorýchlostného prístupu do dátových sietí prostredníctvom inštalovaných vedení. V prístupovej sieti sa najčastejšie využívajú symetrické metalické vedenia tvorené krúteným párom, ktoré boli pôvodne určené na prenos analógových signálov na analógovej účastníckej telefónnej prípojke. Aj napriek tomu, že prenosové parametre týchto vedení sú „nevhodné“ pre potreby širokopásmového prístupu, používajú sa aj pre tento účel, pretože ich úplné nahradenie modernejšími médiami je v niektorých oblastiach extrémne finančne náročné.

K technológiám, ktoré sa výraznejšie presadzujú na komunikačnom trhu a vyhovujú požiadavkám na pripojenie do siete internetu, hlavne pre malé a stredné podniky, patria popri bezdrôtových alebo optických, predovšetkým prípojky digitálnej účastníckej slučky - Digital Subscriber Loop (ADSL alebo VDSL).

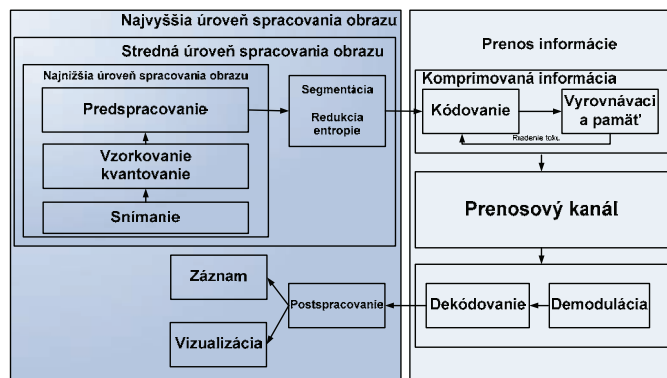
Medzi progresívne dohľadové metódy slúžiace na zisťovanie aktuálneho stavu procesu (pre oblasť IDS ide najmä o cestnú dopravu) patrí videodohľad a videodetekcia pohybujúcich sa objektov (vozidiel, chodcov prípadne iných objektov na/popri dopravnej ceste).

## 1. Spracovanie obrazovej informácie

Pri spracovávaní obrazu (nielen pre aplikácie IDS) je možné rozlíšiť niekoľko úrovní [1]. Najvyššia úroveň predstavuje vlastne porozumenie obrazu, vyhodnotenie zistených faktov a konkrétny výstup zo systému v závislosti od požiadaviek. Na to, aby bolo možné vôbec porozumieť obrazu, je potrebné ho spracovať najprv na nižších úrovniach, kde sa spracováva na základe jasu, farby a ich zmien, alebo iných jednoduchých, či zložitejších vlastností bodov a oblastí, ktoré pomôžu odlíšiť objekty od pozadia, obmedziť vplyv šumu a podobne.

Vo všeobecnosti je úlohou spracovania obrazu vhodným spôsobom obraz reality v čase: zaznamenať, spracovať a následne vyhodnotiť sledované parametre.

Delíme ich na kategórie tak ako sú opísané na obr. 1.



Obr. 1 Schéma spracovania digitálneho obrazu

Fig. 1 Digital image processing scheme

### 1.1. Najnižšia úroveň spracovania

Na najnižšej úrovni sa zaoberáme činnosťami, pri ktorých vykonávaní nie je potrebné žiadne špeciálne rozhodovanie zo strany systému. Jednotlivé funkcie pracujú automaticky podľa vopred zadefinovaného postupu. Za základné kroky sú považované:

- Získanie obrazu;
- Predspracovanie.

### 1.2 Stredná úroveň spracovania

Na strednej úrovni sa riešia úlohy extrahovania a charakterizovania častí – oblastí obrazu. Na rozdiel od nízkej úrovne sa pri analýze vyžaduje oveľa väčšia flexibilita. Charakteristické sú nasledujúce kroky:

- Segmentácia;
- Reprezentácia a popis.

### 1.3 Najvyššia úroveň spracovania

Posledná a najvyššia úroveň zahŕňa kroky rozpoznania a interpretácie obsahu, ktoré sa týkajú najmä aplikácií vyžadujúcich automatizovanú analýzu digitálneho obrazu. Analýza obrazu tu predstavuje objavenie, identifikovanie a pochopenie významu vzorov získaných zo spracovávaného obrazu, o ktorých sa všeobecne vopred nič nepredpokladá. Charakteristické sú nasledujúce kroky:

- Rozpoznanie a interpretácia – porozumenie je založené na znalostiach, cieľoch, tvorbe plánov k ich dosiahnutiu i využití spätných väzieb medzi rôznymi úrovňami spracovania. Základnou úlohou týchto krokov je vybaviť stroj/počítač aspoň čiastočne schopnosťami, aké majú ľudské bytosti v spojitosti s riešením problému videnia a vnímania. Na úspešne rozpoznanie a interpretáciu sa využívajú poznatky z oblasti geometrie, štatistiky, neurových sietí, umelej inteligencie či fuzzy množín.

Podrobnejšie o jednotlivých úrovniach v [1].

## 2. Metódy spracovania obrazu

Metódy spracovania obrazu nachádzajú dnes uplatnenie i vo veľmi rozsiahlej oblasti inteligentných systémov riadenia dopravy. Na každý Video Image Processing systém (VIP), ktorý má za úlohu pozorovať a získavať parametre procesov v doprave, sú kladené nasledujúce všeobecné požiadavky:

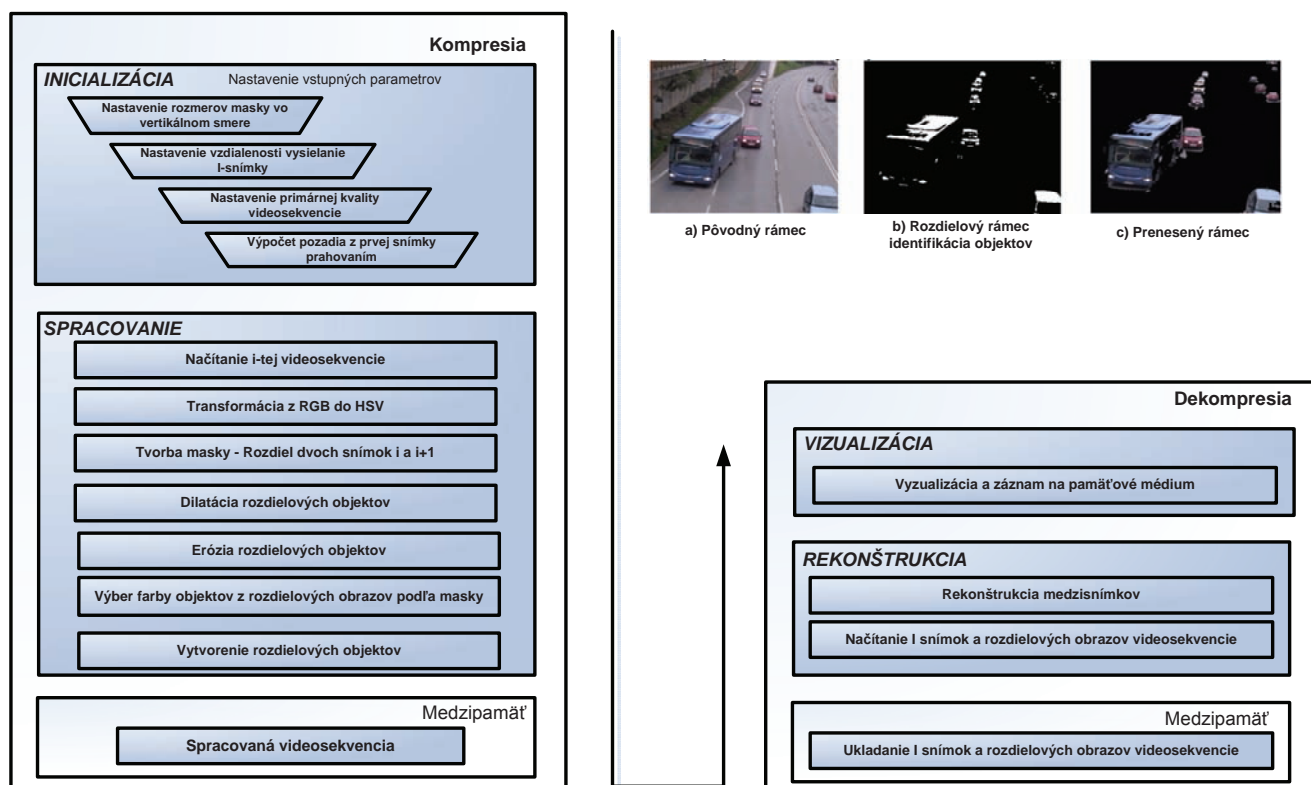
- Automatické rozpoznanie každého objektu od ostatných a od pozadia tak, aby sa podchytili všetky objekty záujmu na scéne.
- Správne rozpoznanie typov objektov – vozidlá nákladné a osobné, cyklisti, chodci, rušne, odvesy atď.
- Zabezpečenie funkčnosti s ohľadom na široké spektrum možných situácií, ktoré sa vyskytujú v spojitosti s dopravným procesom.
- Nezávislosť od svetelných alebo v prípade termovízie, teplotných podmienok snímanej scény – deň a noc, zima a leto a pod.
- Vykonávanie v reálnom čase.

To, do akej miery sú splnené uvedené požiadavky, závisí od konkrétnej aplikácie. So systémami, ktoré spracovávajú obrazové informácie sa stretávame najčastejšie v cestnej doprave. Princípy, na ktorých spočívajú, nám však nebránia aby sme ich aplikovali i v iných druhoch dopravy.

Komunikačné siete budované pre oblasť IDS sú vo všeobecnosti definované ako uzavreté (dôveryhodné) prenosové systémy v správe jedného právneho subjektu. V prípade Slovenskej Republiky (SR) ide o Slovenskú správu ciest (SSC) a Národnú diaľničnú spoločnosť (NDS). Tieto organizácie spravujú cesty prvej triedy (SSC) a diaľničnú sieť (NDS). Budovanie komunikačných sietí v réžii týchto spoločností je v dnešnej dobe upravené v Technických podmienkach (TP) [2] a [3]. Tieto TP definujú použitie optických prenosových systémov a na jednotlivých úrovniach OSI.

Nároky na telematické komunikačné siete musia korešpondovať so základnými charakteristikami prenosov a musia zabezpečiť požiadavky na ich dostupnosť, bezpečnosť a spoľahlivosť:

- Dostupnosť - vyjadruje čas, počas ktorého je služba (aplikácia) poskytovaná zákazníčkovi a komunikačná sieť je v prevádzke. Hodnota môže byť stanovená v čase (napr. ako doba, po ktorú systém môže byť mimo prevádzky) a dá sa vyjadriť matematicky aj ekonomicky.
- Bezpečnosť - zaistenie ochrany informácie ako takej. V prípade pevných sietí možno bezpečnosť prenosu informácie riešiť programovým vybavením centier a jednotlivých aplikácií, v prípade bezdrôtových prostriedkov sa rieši architektúrou vlastnej telekomunikačnej siete a protokolom prenášanej správy.
- Spoľahlivosť - spoľahlivosť prenosu vo vzťahu k telekomunikačnému prostrediu je potrebné vnímať v rámci spoľahlivej funkcie jednotlivých technológií vedúcej ku včasnému doručeniu informácie ako takej. Má priamy vzťah k pojmu dostupnosti, preto ju možno sledovať vo všetkých segmentoch prenosového reťazca. Vyjadruje spoľahlivosť zariadení, fyzickej vrstvy, kvality vyhotovenia, ale aj odolnosť zvolených technických prostriedkov proti rušeniu (Electromagnetic Compatibility, EMC).



Obr. 2 Navrhnutý algoritmus spracovania obrazovej informácie

Fig. 2 Designed image information processing algorithm

### 3. Kompresný algoritmus

Problematika spracovania video-tokov pre potreby hybridnej siete na aplikačnej vrstve riadiacich telematických dopravných systémov prináša množstvo úloh, ktoré bolo nutné aj v rámci [1], [4] vyriešiť. Išlo predovšetkým o tieto úlohy:

- Navrhnuť algoritmus na predspracovanie, detekciu a ohraničenie pohyblivých objektov a navrhnuté riešenie otestovať.
- Vytvoriť simulačný model (program) spracovania testovacích video-tokov pre rozhranie hybridnej siete na aplikačnej vrstve v prostredí MATLAB.
- Za pomoci vytvoreného simulačného modelu stanoviť optimálnu konfiguráciu zmenených parametrov na testovacej video-sekvencii.
- Vykonať experimenty na testovacom video-toku za účelom stanovenia optimálneho pomeru prenesených dát diferenciálnych objektov ku kompresnému pomeru s dôrazom na zachovanie definovanej kvality výsledného rekonštruovaného video-toku v prostredí MATLAB.
- Porovnať rekonštruované video-toky s pôvodnými vstupnými video-tokmi objektívnou metódou SSIM.
- Navrhnuť metódu subjektívneho posúdenia kvality rekonštruovaného videotoku dispečerom IDS systémov.

#### 3.1 Optimalizácia kompresného algoritmu a dosiahnuté výsledky

Na overenie činnosti navrhnutého bloku VIDEO processing a následnej optimalizácie jeho parametrov (pre potreby snímania dopravných procesov), sa za prioritný parameter simulácií zvolil parameter threshold, ktorý slúži na primárne oddelenie rozdielových oblastí od pozadia snímky.

Výstupom simulácie bola hodnota priemernej bitovej rýchlosti spracovaného výstupného video-toku v závislosti od zmeny ďalších parametrov (video\_keyframe a BG\_adaptation) pri nastavovanom kroku parametra Prah (napríklad simulačnom kroku 0,01).

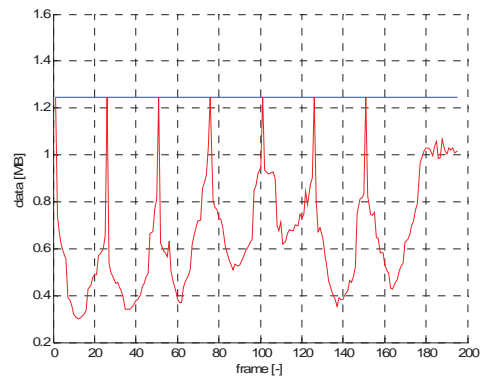
Ako zdrojová bola použitá video-sekvencia s názvom „768x576\_traffic.avi“ (768x576\_skuska.avi), z ktorej bol v programovom prostredí MATLAB za pomoci funkcie IDS\_main vytvorený avi-objekt1, ktorého výsledná veľkosť tvorila hodnotu 156,72 MB (195 snímok s rozlíšením 768x576 o veľkosti 1,2442 MB).

Následne sa postupom opísaným na hlavnom obrázku 2. vytvorila snímka pozadia a selektovali sa rozdielové objekty ako rozdiel pozadia po sebe idúcich snímok avi-objektu1. Súbor avi objekt2 predstavoval video-sekvenciu, ktorá obsahuje rekonštruované snímky vytvorené spojením pozadia a príslušných rozdielových objektov príslušných snímok (obr. 2c).

Na nasledujúcich obrázkoch sú zobrazené simulácie pre nastavené parametre: Odsimulované hodnoty pre parameter  $F=Prah$  z množiny  $\{0,01 \text{ až } 0,1\}$  (krok simulácie 0,01) pre nastavené parametre  $A=$ „768x576\_traffic.avi“  $B=25$ ,  $C=100$ ,  $D=0.5$ ,  $E=25$ ,  $G=1$ ,  $H=576$ , kde  $A=input\_video$ ,  $B=video\_fps$ ,  $C=video\_quality$ ,  $D=video\_keyframe$ ,  $E=BG\_adaptation$ ,  $F=Threshold$ ,  $G=top\_boundary$ ,  $H=bottom\_boundary$

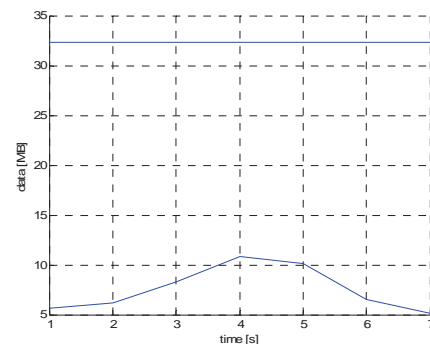
Na obr. 3 je zobrazený modrou farbou pôvodný nekomprimovaný dátový tok s priemernou hodnotou 1.2442 MB/snímka a červenou farbou dátový tok prenášaných rozdielových objektov pre jednotlivé snímky s priemerným dátovým tokom diferenciálnych objektov 0.6567 MB/snímku. Každá 25 snímka je následne prenesená ako plná. Na obr. 4 je zachytená závislosť zmeny dátového toku rozdielových objektov v závislosti od času. Z grafu je viditeľné, že medzi 3 - 5 sekundou záznamu rastie dátový tok. V snímanom

videozázname sa objavil veľký objekt v podobe autobusu (obr. 2c), čo ovplyvnilo veľkosť prenášaných dát v rozdielových snímkach.



Obr. 3 Porovnanie dátových tokov na snímku zdrojového (modrá krivka) a komprimovaného videotoku (červená krivka).

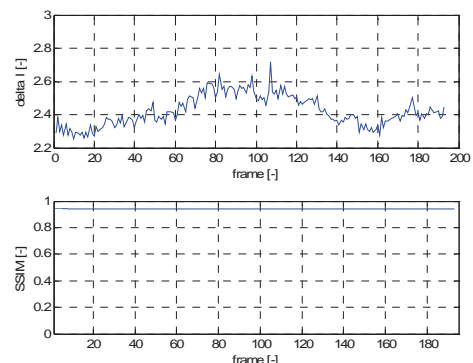
Fig. 3 Comparison of data flows per frame of source and compressed video-flow.



Obr. 4 Veľkosť dátového toku rozdielových objektov v čase.

Fig. 4 Data flow volume of differential objects in time

Obr. 5 opisuje v hornej časti dynamickú zmenu kvality rekonštruovaného obrazu, získanú metódou porovnania dvoch rámcov metódou SSIM. V dolnej časti sú graficky zobrazené odsimulované hodnoty SSIM. Zhoda pôvodných pixelov s rekonštruovanými vyjadrená bezozmerným číslom vyšla blízka hodnote 93,99%. Z toho je možné usúdiť, že iba 6% pixelov na výslednej snímke bolo pozmenených.



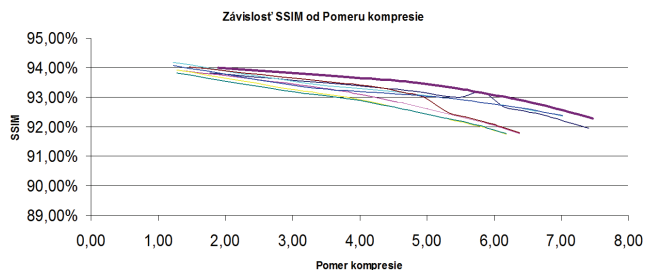
Obr. 5 Miera zhody pixelov pôvodného a rekonštruovaného obrazu vyjadreného parametrom SSIM.

Fig. 5 Pixels-match ratio of the original and reconstructed image expressed by the SSIM parameter.

Na zistenie priebehu hodnôt parametra SSIM bol porovnávaný zdrojový súbor „768x576\_traffic.avi“ s prenesenými a

rekonštruovaným súborom na strane dispečera: „768x576\_skuska\_reconstructed.avi“. Porovnávanie sa vykonávalo v programovom prostredí MATLAB pomocou navrhutej simulácie, ktorá sa vykonávala na základe funkcie [SSIM]=Video\_Quality\_Measurement\_main. Výstupom takto realizovanej funkcie je bezrozmerný číselný údaj a výsledné hodnoty sú prevedené a vyjadrené percentuálne

Na obr. 6 sú graficky zobrazené a sumarizované odsimulované priebehy hodnôt parametra SSIM v závislosti od zmeny parametra Pomer kompresie K (vzorec 2). Ide o pomer objemu vstupného prúdu dát ( $V_{SD}$ ) k objemu výstupného prúdu dát ( $V_{yD}$ ).



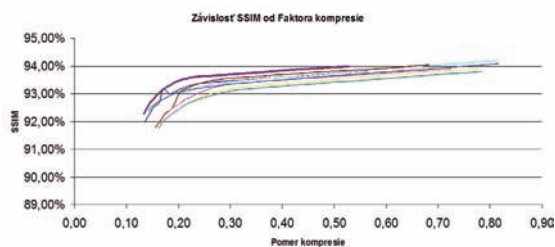
Obr. 6 Závislosť SSIM od zmeny parametru Pomer kompresie (K) – pomer objemu vstupného prúdu dát ( $V_{SD}$ ) k objemu výstupného prúdu dát ( $V_{yD}$ ).

Fig. 6 Dependency of SSIM on Compression ratio (K) parameter modification.

$$K = \frac{V_{SD}}{V_{yD}} \quad (1)$$

$$C = \frac{V_{yD}}{V_{SD}} \quad (2)$$

Na obr.7 sú graficky zobrazené a sumarizované odsimulované priebehy hodnoty SSIM v závislosti od zmeny parametra Faktor kompresie C (vzorec 2). Ide o pomer objemu výstupného prúdu dát ( $V_{yD}$ ) k objemu vstupného prúdu dát ( $V_{SD}$ ).



Obr. 7 Závislosť SSIM od zmeny parametru Faktor kompresie C.

Fig. 7 Dependency of SSIM on Compression factor parameter modification.

## Záver

Snaha o integráciu siete internetu ako univerzálneho komunikačného prostredia prináša tlak, aby bolo možné bezpečne a s vopred definovanou mierou zabezpečenia pristupovať cez verejnú otvorenú dátovú sieť k technologickým procesom bežiacim vo vnútri technologickej siete za účelom zberu dát, ich správy a vzdialenej diagnostiky. Z hľadiska prenosu dát je nutné a na základe uvedených skutočností vyplýva, že je potrebné, aby sa v dnešnej dobe pri budovaní komunikačnej infraštruktúry mestských (krajské mestá

a mestá s vysokou intenzitou dopravy) dopravnotelematických systémov kládli vysoké požiadavky na kvalitu prenosu a dostatočnú prenosovú kapacitu siete. To je možné v dnešnom čase aj pri použití prístupových systémov DSL využívajúcu obmedzenú prenosovú kapacitu metalických sietí za pomoci nových techník spracovania obrazových a iných vstupných informácií.

## Podakovanie

Táto publikácia vznikla s priamou podporou projektu ITMS 26220120050 s názvom "Centre of excellence for systems and services of intelligent transport II" riešeného v rámci Agentúry Ministerstva školstva, vedy, výskumu a športu SR pre štrukturálne fondy EÚ.



"Podporujeme výskumné aktivity na Slovensku/Projekt je spolu-financovaný zo zdrojov EÚ"

## Literatúra

[1] PIRNÍK, R.: Model prenosu technologických dát v prostredí xDSL. 113 p., dizertačná práca. Žilinská univerzita v Žiline, Žilina, 2009.

[2] TP 09/2008. Zariadenia, infraštruktúra a systémy technologického vybavenia pozemných komunikácií. [online]. Bratislava: SSC. 2008. Dostupné na internete: <[http://www.ssc.sk/files/documents/technicke-predpisy/tp2008/tp\\_09\\_2008.pdf](http://www.ssc.sk/files/documents/technicke-predpisy/tp2008/tp_09_2008.pdf)>.

[3] TP 10/2008 Inteligentné dopravné systémy a dopravné technologické zariadenia. [online]. Bratislava : SSC. 2008. Dostupné na internete: < [http://www.ssc.sk/files/documents/technicke-predpisy/tp2008/tp\\_10\\_2008.pdf](http://www.ssc.sk/files/documents/technicke-predpisy/tp2008/tp_10_2008.pdf)>.

[4] HOLEČKO, P., FRANEKOVÁ, M.: Simulačné platformy pre bezdrôtové senzorové siete. In: Journal of Information, Control and Management Systems. - ISSN 1336-1716. - Ročník. 9, Číslo 3. 2011, s. 259-266.

## Abstract

The goal of this paper is to analyse, define, and summarise the requirements on networks connecting telematic technologies based on Intelligent Transport Systems (ITS). The paper describes the methodics of setting and a general concept of technological networks interface based on available and applicable standards. The paper focuses on image information processing and its tunnelling through the widespread metropolitan transmission xDSL technology. Further a designed compression algorithm and its optimization are presented. The results achieved were suitable for the needs of interfaces between open and technological network types of transport-telematic systems.

Rastislav Pirník, Ing. PhD.

Katedra riadiacich a informačných systémov  
Elektrotechnická fakulta  
Žilinská univerzita v Žiline  
Univerzitná 8215/1, 010 26 Žilina  
tel. č.: 041/513 3351  
e-mail: [rastislav.pirnik@fel.uniza](mailto:rastislav.pirnik@fel.uniza)

# Núdzové ovládanie radiacích systémov v železničnej doprave

Peter Nagy, Juraj Ždánsky

## Abstrakt

Príspevok sa zaoberá problematikou núdzového ovládania železničných zabezpečovacích systémov používaných na riadenie železničnej dopravnej prevádzky. Zameriava sa predovšetkým na oblasť staničných zabezpečovacích systémov, kde je použitie núdzového ovládania najčastejšie. Konvenčné zabezpečovacie systémy založené na pevnej logike sú koncipované tak, že pri poruche zabezpečovacieho systému, ktorá vedie k obmedzeniu železničnej dopravy v riadenej oblasti, použitie núdzového ovládania umožní zachovanie dopravnej prevádzky. Cenou za to je však zníženie úrovne bezpečnosti riadenia. Moderné elektronické systémy s programovateľnou logikou majú aplikované mechanizmy, ktoré dovoľia použitie núdzového ovládania pri zachovaní dostatočne vysokej úrovne integrity bezpečnosti (SIL - safety integrity level).

**Kľúčové slová :** riadenie železničnej dopravnej prevádzky, železničný zabezpečovací systém, staničný zabezpečovací systém, núdzové ovládanie, úroveň integrity bezpečnosti, stavový model zabezpečovacieho systému

## Úvod

Riadiace systémy používané v železničnej doprave sa podľa miesta použitia odlišujú svojim účelom, sortimentom poskytovaných funkcií a samozrejme konštrukciou. Na rôznych úrovniach riadenia sa používajú rôzne architektúry radiacích systémov. V podmienkach Železníc Slovenskej republiky (ŽSR) sa prakticky stretávame s trojúrovňovým riadením dopravnej prevádzky, pričom najnižšia (procesná) úroveň riadenia je realizovaná na úrovni železničnej stanice, resp. medzistaničného úseku, vyššia – tzv. dispečerská úroveň riadenia na úrovni traťového úseku dĺžky približne 100 km a najvyššia úroveň riadenia je centrálna úroveň hlavného, resp. ústredného dispečera celej siete ŽSR.

Riadiace systémy aplikované na procesnej úrovni riadenia sa bezprostredne podieľajú na riadení pohybu vlakov v sieti ŽSR. Pretože realizáciou svojich radiacích funkcií priamo vplývajú na bezpečnosť dopravnej prevádzky, považujú sa za riadiace systémy súvisiace s bezpečnosťou a z toho vyplývajú následné požiadavky na konštrukciu a funkčné vlastnosti týchto radiacích systémov. Vzhľadom na bezpečnostné funkcie týchto radiacích systémov sa pre ne historicky zaužívalo označenie železničné zabezpečovacie zariadenie, resp. železničné zabezpečovacie systémy.

Dôležitou súčasťou každého riadiaceho systému je jeho rozhranie umožňujúce operátorovi systému získavať stavové informácie o riadenom procese a prostredníctvom vhodných ovládacích prvkov dávať príkazy na zmenu stavu riadeného systému. Je dôležité, aby získané stavové informácie zodpovedali aktuálnemu stavu riadeného procesu, pretože operátor musí na prípadnú zmenu stavu procesu reagovať v reálnom čase. To kladie značné nároky na systémy zberu, spracovania a prezentácií stavových informácií, ktoré musia pracovať v reálnom čase rovnako, ako samotný radiaci systém.

## 1. Ovládanie železničných zabezpečovacích systémov

Vzhľadom na historický vývoj železničných zabezpečovacích systémov (ďalej len zabezpečovacie systémy) je obsluha jednotlivých typov systémov sústredená predovšetkým do železničných staníc, v ktorých je pohyb koľajových vozidiel riadený prostredníctvom staničných zabezpečovacích systémov. Zo železničných staníc sú spravidla ovládané aj traťové alebo priecestné zabezpečovacie systémy v traťových úsekoch priľahlých k železničnej stanici. Na moderných koridorových tratiach sa ovládanie viacerých staníc sústreďuje do centier riadenia dopravy, z ktorých je diaľkovo ovládaná doprava v dlhšom traťovom úseku. Typická dĺžka riadeného úseku v sieti ŽSR je približne 100 km. Z centrálného dispečerského pracoviska možno ovládať aj niekoľko desiatok podriadených staníc.

Vývoj ovládacích pracovísk na riadenie železničnej dopravy sledoval vývoj samotných zabezpečovacích systémov. Pôvodné zabezpečovacie zariadenia používali špecializované ovládacie prvky vyvinuté pre ich potreby. V mechanických zabezpečovacích zariadeniach to boli napríklad mechanické stavadlové páky rôznych konštrukcií, špeciálne konštruované viacpolohové radiče, indikácie boli odvodené od polohy týchto ovládacích prvkov, prípadne sa používali špecializované elektromechanické indikačné zariadenia – napríklad elektromechanické relé s indikačným terčom, elektromechanické tzv. Frischenovo hradlo, kontrolné žiarovky alebo elektromechanické zvončeky.

Reléové systémy rovnako používajú špecializované ovládacie pracoviská – v podmienkach ŽSR sú to typicky vodorovné alebo šikmé ovládacie stoly alebo zvislé kontrolné a ovládacie skrine. Sú na nich umiestnené potrebné ovládacie prvky v podobe tlačidiel alebo radičov špeciálnej konštrukcie typicky používané len v týchto systémoch. Potrebné stavové informácie operátor získava pomocou vhodne usporiada-

ných indikačných žiaroviek rôznych farieb, prípadne doplnených akustickými návěstami realizovanými elektromechanickými zvončekmi. Hoci indikačné prvky sú spravidla univerzálne, ich usporiadanie a konštrukcia celého indikačného zariadenia je opäť typická len pre zabezpečovacie systémy.

Použitie moderných elektronických systémov založených na koncepcii univerzálnych, prípadne špecializovaných počítačov, mení do istej miery prístup k ovládaniu a kontrole zabezpečovacích systémov. Ako indikačné a ovládacie prvky sa používajú štandardné, bežne dostupné komerčné prvky bez špeciálnych úprav pre riadenie kritických procesov v doprave, založené spravidla na báze štandardných osobných počítačov (PC). Napriek použitiu komerčne dostupných prvkov, sa však zabezpečovacie systémy odlišujú od bežných riadiacich systémov určených pre priemyslové aplikácie ako v konštrukcii, riadiacich algoritmoch a postu-  
poch, tak aj v spôsobe vizualizácie riadených procesov.

### 1.1 Bezpečnosť ovládania zabezpečovacích systémov

Zabezpečovacie systémy sú technologickým nástrojom riadenia železničnej dopravy – sú riadiacim systémom, ktorý kontroluje alebo nahradzuje človeka (operátora riadiaceho systému, rušňovodiča) pri vykonávaní jednotlivých činností súvisiacich s riadením pohybu koľajových vozidiel po železničnej dopravnej ceste. Účelom kontroly, prípadne náhrady človeka, je predovšetkým prispievať k zvýšeniu bezpečnosti železničnej dopravy tým, že zabezpečovací systém eliminuje omyly človeka. Táto základná funkcia je realizovaná jednotlivými kategóriami zabezpečovacích systémov, ktoré okrem algoritmov funkčnej bezpečnosti (realizujúcich tzv. dopravno-bezpečnostné algoritmy) majú implementované aj potrebné technológie na zaistenie technickej bezpečnosti samotného riadiaceho systému.

Zabezpečovací systém možno považovať podľa normy EN 50129 [1] za systém s nepretržitou prevádzkou. Pokiaľ je systém funkčný, tak jeho bezpečnosť (primárna bezpečnosť) je zaisťovaná najmä pomocou technických opatrení. Organizačné opatrenia sú používané len v minimálnej miere a to len vtedy, keď použitie technických opatrení je nemožné (obmedzené technické možnosti, veľké finančné nároky). Ak je zabezpečovací systém čiastočne alebo úplne nefunkčný, riadený dopravný proces musí pokračovať a jeho bezpečnosť (sekundárna bezpečnosť) je zaisťovaná pomocou organizačných opatrení [2].

Pri bezporuchovej činnosti zabezpečovacích systému sú chybné príkazy operátora eliminované správnou činnosťou príslušného zabezpečovacieho systému. Úroveň eliminácie závisí predovšetkým od podielu, akým sa na riadení dopravného procesu podieľa človek a akým zabezpečovací systém. Je preto snahou v maximálnej miere obmedzovať podiel človeka na realizácii bezpečnostne kritických operácií. Typickým spôsobom realizácie tohto obmedzenia je realizácia tzv. dopravno-bezpečnostných algoritmov (DBA) staničným, traťovým alebo priescestným zabezpečovacím systémom s maximálnou mierou automatizácie ich činnosti a s minimálnym podielom činnosti človeka [3]. Pri tvorbe DBA musia byť použité také metódy a postupy, ktoré garantujú integritu bezpečnosti proti systematickým poruchám na úrovni SIL4 [1], [4]. Automatizácia činnosti a eliminovanie ľudského faktora je možné predovšetkým v systémoch s jednoduchšími DBA – v traťových a priescestných zabezpečovacích systémoch.

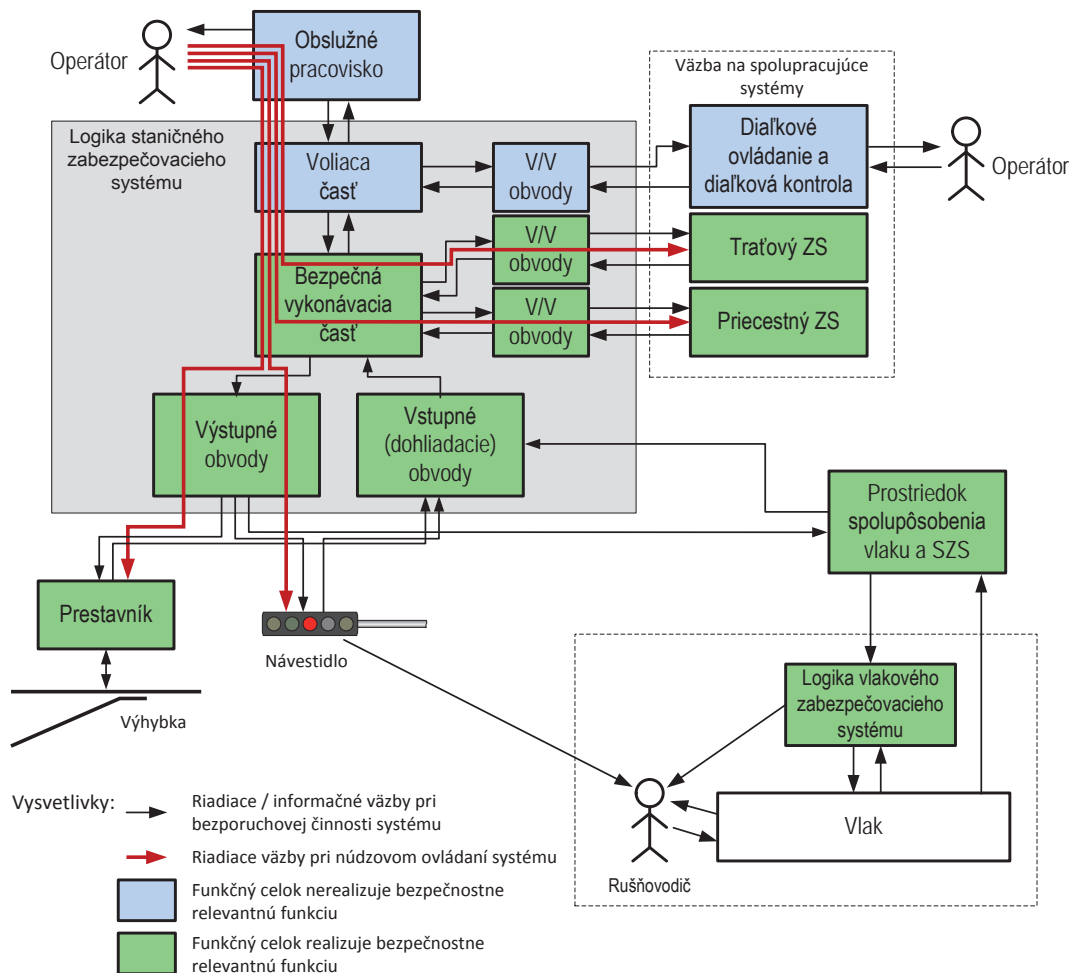
### 1.2 Staničné zabezpečovacie systémy

V prípade staničných zabezpečovacích systémov je úloha človeka zatiaľ ťažko zastupiteľná najmä v časti rozhodovania o postupnosti krokov pri riadení jazdy koľajových vozidiel v obvode stanice. Do určitej miery možno automatizovať

činnosti súvisiace s riadením jazdy vlakov cez stanicu. V prípade štandardnej dopravnej situácie bez mimoriadností v doprave, možno použiť automatické stavenie vlakových ciest podľa plánovaného grafikonu vlakovkej dopravy na základe detekcie prítomnosti vlaku vo vopred definovaných miestach železničnej dopravnej cesty. Rovnako možno automaticky rušiť vlakové cesty po ich prejení príslušným vlakom. Automatické stavenie posunových ciest a vôbec automatizácia posunu je však veľmi ťažko realizovateľná a zatiaľ nie je predmetom automatickej činnosti staničných zabezpečovacích systémov. Riadenie posunu si najmä vo väčších staniaciach vyžaduje značné skúsenosti operátora, ktorý veľmi často len odhaduje správanie riadeného dopravného procesu a nemôže ovplyvniť napríklad čas, za ktorý sa posunujúci diel uvedie do pohybu po vydaní súhlasu na jazdu, nemôže ovplyvniť čas manipulácie s posunujúcim dielom, len minimálne môže ovplyvniť rýchlosť jazdy koľajových vozidiel, atď. Okrem toho riadenie posunu (najmä vo veľkých staniaciach) dovoľuje extrémne veľké množstvo rôznych variantov jász posunujúcich dielov. Všetky tieto faktory komplikujú realizáciu automatizácie posunu v staniaciach. To je hlavným dôvodom, prečo je nevyhnutná prítomnosť človeka pri riadení pohybu koľajových vozidiel v staniaciach.

Pri riadení pohybu koľajových vozidiel v železničnej stanici sa staničný zabezpečovací systém podieľa na plnení a kontrole splnenia podmienok pre postavenie jazdnej cesty pred jazdou koľajového vozidla (vlaku alebo posunujúceho dielu) a kontrole splnenia podmienok pre zrušenie jazdnej cesty po jazde koľajového vozidla a následnom zrušení jazdnej cesty. Staničný zabezpečovací systém možno chápať ako riadiaci systém súvisiaci s bezpečnosťou, ktorý prijíma príkazy od operátora, prijíma (sníma) informácie o stave riadeného procesu – stave riadených prvkov umiestnených v koľajisku a polohe koľajových vozidiel, spracováva ich, na základe tohto spracovania posúdi oprávnenosť a správnosť príkazu operátora a na základe tohto posúdenia tieto príkazy transformuje na privedenie energie k výkonným prvkom umiestneným v koľajisku (prestavník, návěstidlo) alebo prenesie informáciu do spolupracujúceho zabezpečovacieho systému (väzba na vlak, trať, priescestie a pod.). Obr. 1 znázorňuje riadiace a informačné väzby staničného zabezpečovacieho systému pri jeho bezporuchovej činnosti a pri núdzovom ovládaní. Pri bezporuchovej činnosti systému príkazy vydávané lokálnym alebo vzdialeným operátorom prijíma a spracováva voliacu časť systému. Hlavnou funkciou voliackej časti je skontrolovať správnosť zadaného príkazu a príkaz transformovať na riadiace príkazy pre konkrétne prvky umiestnené v koľajisku. Voliacu časť však nekontroluje bezpečnosť vydávaných riadiacich príkazov. Riadiace príkazy spracované voliacou časťou systému sú následne spracované bezpečnou vykonávacou časťou zabezpečovacieho systému, ktorá realizuje dopravno-bezpečnostné algoritmy – kontroluje splnenie funkčných podmienok pre ovládanie prvkov v koľajisku, pričom tieto funkčné algoritmy realizuje bezpečne. V tomto funkčnom celku je aplikovaná technika zaisťujúca funkčnú aj technickú bezpečnosť systému. Bezpečné výstupy z vykonávacej časti sú následne cez bezpečné výstupné obvody prenášané k ovládaným prvkom.

V prípade, že nastane porucha v realizácii DBA spôsobená chybou vonkajšieho prvku, poruchou vstupného alebo výstupného obvodu alebo poruchou bezpečnej vykonávacej časti, zabezpečovací systém musí v stanovenom čase poruchu identifikovať a následne reagovať vopred stanoveným bezpečným stavom. Tento bezpečný stav spravidla zne-  
možňuje ovládanie niektorých prvkov v koľajisku – napríklad prestavovanie pohyblivých častí výhybiek, ovládanie niektorých návěstidiel a v konečnom dôsledku tak obmedzuje pohyb koľajových vozidiel v obvode stanice. Z hľadiska používateľa zabezpečovacieho systému je však takéto



**Obr.1 Riadiace a informačné väzby staničného zabezpečovacieho systému pri normálnom a núdzovom ovládaní**  
**Fig. 1 Control and information connections of interlocking system during standard and emergency operation**

obmedzenie negatívnym javom, pretože sa obmedzuje riadený technologický proces. Preto používatelia zabezpečovacích systémov požadujú možnosť núdzového riadenia pohybu koľajových vozidiel v prípade poruchy riadiaceho systému.

Z toho dôvodu je do zabezpečovacích systémov implementovaná možnosť núdzového ovládania vydaním príkazov, ktoré možno súhrnne označiť ako núdzové príkazy. Pre núdzové príkazy je charakteristické, že sa používajú v prípade neštandardných stavov systému. Vo všeobecnosti ide o príkazy používané v prípade, keď v dôsledku poruchy vonkajšieho prvku, vstupnej alebo výstupnej jednotky alebo z dôvodu nespĺnenia niektorej definovanej podmienky DBA nedovoľujú vykonanie štandardného príkazu na postavenie alebo zrušenie jazdnej cesty, prípadne ovládanie zvoleného prvku v koľajisku. Pretože riadenie dopravného procesu si vyžaduje umožnenie činnosti systému aj v takomto prípade, zabezpečovací systém je štandardne vybavený možnosťou použitia núdzového ovládania umožňujúceho núdzové riadenie dopravného procesu a prvkov v koľajisku. Pre núdzové ovládanie je charakteristické priame ovládanie vonkajších objektov (prestavníkov, návestidiel, ...) bez kontroly podmienok požadovaných pri štandardnom ovládaní. V prípade konvenčných zabezpečovacích systémov s pevnou logikou sa núdzové ovládanie napríklad realizuje tak, že zvláštnym ovládacím prvkom je ovládaný priamo výstupný obvod príslušného ovládaného prvku v koľajisku. V schéme na obr. 1 je toto ovládanie znázornené červenou čiarou. Takéto ovládanie môže za určitých okolností spôsobiť priame ohrozenie bezpečnosti a preto musí byť podmienené vykonaním organizačných a administratívnych opatrení stanovených obvykle prevádzkovateľom dráhy. V prípade konvenčných zabezpečovacích systémov so špecializova-

nými ovládacími prvkami sú ovládacie prvky pre núdzové ovládanie spravidla vybavené bezpečnostným záverom, ktorý má zabrániť náhodnému použitiu ovládacieho prvku a súčasne upozorniť operátora na vykonávanie núdzového príkazu pri obsluhu príslušného ovládacieho prvku. Obsluha ovládacích prvkov núdzových príkazov je okrem toho spravidla evidovaná buď použitím počítača, alebo iného významového prvku.

Pretože pri vykonávaní núdzových príkazov (niektorí výrobcovia ich označujú aj ako tzv. rizikové príkazy) spravidla nie sú všetky potrebné bezpečnostné podmienky kontrolované zabezpečovacím systémom, úroveň bezpečnosti riadenia závisí od dodržania predpísaných organizačných opatrení operátorom zabezpečovacieho systému. Organizačné opatrenia spravidla od operátora vyžadujú vizuálnu kontrolu stavu riadených a kontrolovaných prvkov v koľajisku – v závislosti od konkrétneho núdzového príkazu napríklad kontrolu voľnosti určeného koľajového úseku (určených koľajových úsekov), kontrolu koncovej polohy pohyblivej časti jednej alebo viacerých výhybiek alebo výkoľajok, kontrolu prejazdu koľajového vozidla určeným miestom a podobne. V prípade správnej činnosti zobrazenia stavov riadenej technológie na zobrazovacích jednotkách zabezpečovacieho systému je pritom dovolené niektoré kontroly odvodiť od indikácií zabezpečovacieho systému – napríklad voľnosť koľajového úseku smie byť odvodená od indikácie voľnosti príslušného úseku na obslužnom pracovisku systému. To si však vyžaduje zaistenie dostatočnej bezpečnosti zobrazovania.

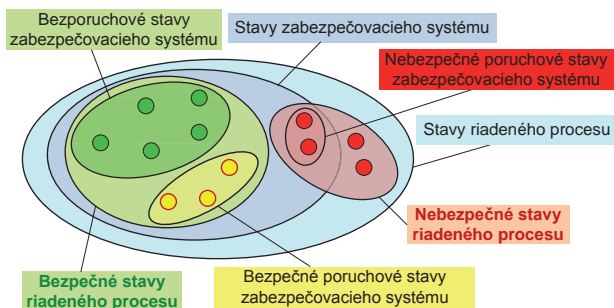
Vykonanie núdzového príkazu spravidla obchádza kontrolu splnenia niektorých dôležitých podmienok na zaistenie bezpečnosti – napríklad núdzové dovolenie jazdy vlaku pomo-

cou tzv. privolávacej návesti umožňuje dať na príslušnom hlavnom návěstidle súhlas na jazdu vlaku aj bez technickej kontroly splnenia akejkoľvek podmienky pre bezpečnú jazdu vlaku (voľnosť koľajových úsekov, po ktorých sa bude uskutočňovať jazda vlaku, správna poloha pohyblivých častí všetkých výhybiek, po ktorých sa bude pohybovať vlak, atď.). Kontrola bezpečnostne relevantných podmienok jazdy vlaku sa v takomto prípade obmedzuje len na administratívnu formu a plná váha bezpečnosti zostáva na operátorovi zabezpečovacieho systému, ktorý je povinný buď sám, alebo v spolupráci s ďalšími pracovníkmi zúčastnenými na riadení dopravného procesu skontrolovať a zaistiť administratívne podmienky pre bezpečnú jazdu vlaku.

Hlavným nedostatkom týchto bezpečnostných postupov je účasť človeka na riadení železničnej dopravy bez jeho kontroly zabezpečovacím systémom. V prípade korektnej činnosti zabezpečovacieho systému je operátor pri vykonávaní bezpečnostne rizikových operácií kontrolovaný zabezpečovacím systémom. V prípade, že nastane taký stav zabezpečovacieho systému, ktorý znemožní jeho štandardné funkcie, operátor sa dostáva do psychického tlaku, pretože zabezpečovací systém je čiastočne alebo úplne nefunkčný, pritom je ale nevyhnutné ďalej zaisťovať riadený dopravný proces. V prípade kumulácie stresových faktorov, akými sú dlhšie trvanie mimoriadneho stavu obmedzenia funkčnosti zabezpečovacieho systému, zložitá dopravná situácia, únava operátora a ďalšie faktory, môže nastať stav, keď človek urobí chybu – neoprávnené použije niektorý núdzový príkaz, núdzový príkaz aplikuje na nesprávny prvok v koľajisku, neskontroluje všetky administratívne opatrenia podmieňujúce realizáciu daného núdzového príkazu, alebo urobí inú chybu a pri určitej dopravnej situácii spôsobí vznik nehody. Inou príčinou vzniku situácie vedúcej k nehode môže byť dlhodobé používanie niektorých núdzových príkazov napríklad počas stavebných úprav koľajiska stanice, pri modernizácii zabezpečovacích systémov a pri podobných situáciách, keď nemožno plne využívať dopravnobezpečnostné algoritmy zabezpečovacích systémov. Dlhodobé a opakované používanie núdzových príkazov (napríklad privolávacej návesti) vedie k rutinnej činnosti a to má často za dôsledok opomenutie alebo zanedbanie niektorých administratívnych úkonov a opäť môže viesť k vzniku nehody.

## 2. Stavový model zabezpečovacieho systému

Rôzne prevádzkové situácie uvedené v predchádzajúcej časti možno znázorniť pomocou stavového modelu zabezpečovacieho systému (obr. 2).



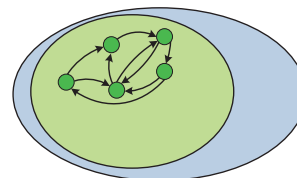
Obr.2 Stavy riadeného procesu a zabezpečovacieho systému

Fig.2 States of controlled process and signalling system

Stavový priestor riadeného procesu je vo všeobecnosti tvorený množinou bezpečných a množinou nebezpečných stavov. Stavový priestor zabezpečovacieho systému možno podobne rozdeliť na množinu bezporuchových a množinu

poruchových stavov zabezpečovacieho systému. Ak sa zabezpečovací systém nachádza v niektorom z bezporuchových stavov, riadený proces sa musí nachádzať v niektorom z bezpečných stavov. Podľa vplyvu na riadený proces možno množinu poruchových stavov zabezpečovacieho systému rozdeliť na množinu bezpečných poruchových stavov, pri ktorých sa riadený dopravný proces bude nachádzať v bezpečnom stave a množinu nebezpečných poruchových stavov zabezpečovacieho systému, pri ktorých sa riadený dopravný proces dostane do nebezpečného stavu.

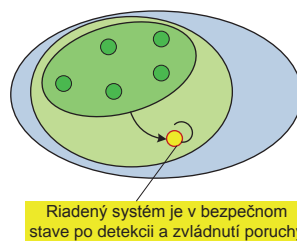
Zabezpečovací systém a riadený dopravný proces môžu v priebehu realizácie riadiacej funkcie prechádzať medzi jednotlivými stavmi.



Obr.3 Prechody medzi bezpečnými stavmi zabezpečovacieho systému

Fig.3 Transitions between the safety states of the signalling system

V prípade bezporuchovej činnosti zabezpečovacieho systému riadený dopravný proces nadobúda stavy z množiny bezpečných stavov (obr. 3), pričom prechody medzi jednotlivými stavmi sa dejú na základe stanoveného riadiaceho algoritmu zabezpečovacieho systému. V prípade, že sa zabezpečovací systém dostane do poruchového stavu, riadený dopravný proces môže prejsť do bezpečného alebo nebezpečného stavu. Zabezpečovací systém ako riadiaci systém súvisiaci s bezpečnosťou má byť konštruovaný tak, aby dokázal v stanovenom čase detegovať vznik poruchového stavu a vzniknutú poruchu zvládol tak, že riadený dopravný proces sa dostane do vopred definovaného bezpečného stavu [5]. V prípade, že zabezpečovací systém nedostane ďalší príkaz, riadený proces má v takomto prípade zotrvať v bezpečnom stave až do odstránenia poruchy (obr.4).

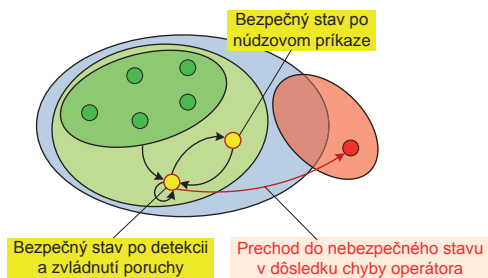


Obr.4 Prechod do bezpečného stavu pri bezpečnej poruche zabezpečovacieho systému

Fig.4 Transition to safe state on safe failure of the signalling system

Ako bolo uvedené v predchádzajúcom texte, pri riadení dopravného procesu je takýto bezpečný stav spravidla obmedzujúci. Preto sa od zabezpečovacieho systému požaduje možnosť vydania núdzových príkazov. V prípade, že operátor splní všetky administratívne a organizačné opatrenia stanovené pre daný núdzový príkaz, riadený dopravný proces sa dostane do iného bezpečného stavu, v ktorom je možná jazda koľajových vozidiel. V prípade, že operátor sa pri realizácii stanovených administratívnych alebo organizačných opatrení dopustí chyby, vzniká ohrozenie, že riadený dopravný proces sa v dôsledku chyby operátora dostane do nebezpečného stavu, v ktorom za istých okolností môže vzniknúť nehoda. Príslušné prechody medzi stavmi zobrazuje obr. 5.

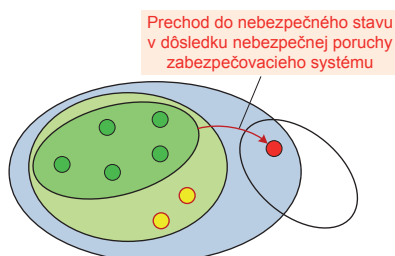




**Obr.5** Prechod do bezpečného alebo nebezpečného stavu pri núdzovom príkaze počas bezpečnej poruchy zabezpečovacieho systému

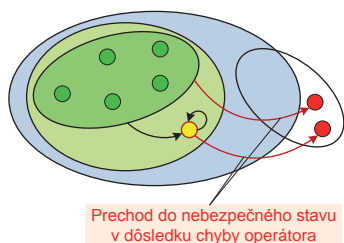
**Fig.5** Transition to safety state or dangerous state on emergency command during safe failure of the signalling system

Do nebezpečného stavu sa riadený dopravný proces môže dostať aj v dôsledku nebezpečnej poruchy zabezpečovacieho systému (obr. 6). Konštrukcia zabezpečovacieho systému však má byť taká, aby pravdepodobnosť takéhoto prechodu bola veľmi malá. Pre zabezpečovacie systémy, ktorých bezpečnostné funkcie sú realizované so SIL4, tolerovateľná intenzita zlyhania (THR – tolerable hazard rate) musí byť z intervalu  $<10^{-9} \text{ h}^{-1}$ ,  $10^{-8} \text{ h}^{-1}$ ) [1].



**Obr.6** Prechod do nebezpečného stavu pri nebezpečnej poruche zabezpečovacieho systému

**Fig.6** Transition to dangerous state on dangerous failure of the signalling system



**Obr.7** Prechod do nebezpečného stavu pri núdzovom príkaze a chybe operátora

**Fig.7** Transition to dangerous state on emergency command and mistake of the operator

Riadený proces sa do nebezpečného stavu môže dostať pri chybe alebo omyle operátora aj v prípade bezporuchového stavu zabezpečovacieho systému (obr. 7). Takáto situácia môže nastať napríklad v prípade, kedy operátor zabezpečovacieho systému nesprávne vyhodnotí prevádzkový stav, v ktorom mu zabezpečovací systém oprávnené nedovoľuje vykonať požadovanú operáciu, pretože nie sú splnené podmienky pre jej vykonanie. Operátor sa však mylne domnieva, že podmienky splnené sú a že zabezpečovací systém sa nachádza v poruche. V takom prípade môže použiť núdzový príkaz neoprávnené a bez opodstatnenia a v dôsledku vlastného omylu priviesť tak riadený proces do nebezpečného stavu a následne spôsobiť nehodu. Takáto situácia môže nastať predovšetkým v situáciách, keď sa niektoré subsystémy zabezpečovacieho systému nachádzajú v poruchovom stave a pre určité funkcie je preto nevyhnutné používať núdzové príkazy. Ak takáto situácia trvá

príliš dlho, operátor správnu ale obmedzujúcu činnosť zabezpečovacieho systému môže chybné vyhodnotiť ako poruchu ďalšieho subsystému zabezpečovacieho systému.

### 3. Bezpečnosť použitia núdzových príkazov

Aby aj v prípade núdzovej ovládania zabezpečovacieho systému bola zachovaná požadovaná úroveň integrity bezpečnosti bezpečnostných funkcií, pri použití moderných elektronických zabezpečovacích systémov výrobcovia aj pri použití núdzových príkazov aplikujú doplnkové techniky na zvýšenie bezpečnosti.

#### 3.1. Bezpečnosť zobrazovania

Prvou podmienkou bezpečnosti núdzového ovládania je zaistenie správnych informácií, na základe ktorých sa operátor rozhoduje pre použitie núdzového príkazu a na základe ktorých aj osobne kontroluje podmienky pre bezpečné vykonanie požadovanej operácie. Napríklad pred obslužením ovládacieho prvku, ktorým sa dovoľí jazda vlaku na privolávaciu návesť, je operátor povinný pohľadom alebo pochádzkou skontrolovať voľnosť všetkých prechádzaných koľajových úsekov ležiacich v zamýšľanej ceste vlaku, skontrolovať správnu polohu všetkých pohyblivých častí všetkých prechádzaných a odvratných výhybiek a výkoľajok a tieto v správnej polohe aj zaistiť. Pri kontrole voľnosti aj správnej polohy pohyblivých častí výhybiek sa pritom smie spoľahnúť na indikácie poskytované riadiacim systémom. To však vyžaduje, aby príslušné informácie boli zobrazované bezpečne. Bezpečné zobrazovanie prevádzkovej situácie sa však nepožaduje pri vykonávaní bezpečnostných funkcií riadiaceho systému pri jeho bezporuchovej činnosti. Vtedy totiž bezpečná časť zabezpečovacieho systému kontroluje všetky príkazy operátora a prípadné chybné príkazy, ktoré by mohli viesť k nebezpečnému stavu riadeného procesu sú ignorované.

V prípade núdzových príkazov preto výrobcovia používajú doplňujúce techniky, ktorými zaisťujú požadovanú úroveň bezpečnosti zobrazenia informácií pri núdzových príkazoch. Medzi tieto techniky patria napríklad:

- indikácia správnosti zobrazenia všetkých základných farieb na zobrazovacej jednotke zabezpečovacieho systému;
- informácie pre núdzové príkazy sa zobrazujú nielen v grafickej ale aj v textovej podobe, pričom spravidla sa vypisujú informácie, ktoré bezpečnostné podmienky pre požadovanú operáciu sú a ktoré nie sú splnené;
- indikácia dynamickej činnosti zabezpečovacieho systému – realizuje sa napríklad tzv. indikátorom aktivity – pri viackanálových zabezpečovacích systémoch sa zobrazenie stavovej informácie realizuje každým kanálom samostatne, pričom informácie od jednotlivých kanálov sa na zobrazovacej jednotke periodicky striedajú. Informácie z jednotlivých kanálov sa odlišujú stanoveným zobrazeným detailom (napríklad farbou orámovania alebo podfarbenia informácie, zobrazením zvláštneho symbolu v zobrazovanej informácii, pričom rôzne kanály zobrazujú iný tvar symbolu a operátor smie zobrazenie považovať za správne len v prípade dynamického striedania všetkých symbolov);
- dynamickejšou kontrolou zobrazovanej informácie bezpečnou časťou zabezpečovacieho systému – informáciu zobrazenú na zobrazovacej jednotke zadávajúca časť systému, ktorá nerealizuje bezpečnostné funkcie vyčítá z videopamäte a posiela na kontrolu každému kanálu viackanálovej bezpečnostne relevantnej časti; pri tejto kontrole sa alternatívne posiela buď celá zobrazovaná informácia, alebo sa z obsahu videopamäte vytvorí len digitálny odtlačok, ktorý bezpečnostne relevantná časť

porovná s digitálnym odtlačkom informácie odosielanej na zobrazovanie. V prípade, že sa oba odtlačky zhodujú, zobrazenie sa považuje za bezpečné a následný núdzový príkaz sa vykoná, v opačnom prípade sa príkaz ignoruje.

### 3.2. Bezpečnosť ovládania

Napriek tomu, že núdzové príkazy sú určené na ovládanie riadeného dopravného procesu v prípade bezpečnej poruchy zabezpečovacieho systému, aj pri realizácii núdzového ovládania sa predpokladá vykonanie núdzových príkazov samotným zabezpečovacím systémom. To znamená, že operátor zadá núdzový príkaz nie bezpečnej voliacej časti systému, ktorá v bezpečnej vykonávacej časti zabezpečovacieho systému aktivuje príslušnú bezpečnostnú funkciu na spracovanie núdzového príkazu.

Na rozdiel od konvenčných zabezpečovacích systémov systémy s programovateľnou logikou majú aj pri realizácii núdzových príkazov integrované bezpečnostné funkcie, ktoré kontrolujú splnenie stanovených podmienok. Splnenie podmienok núdzových príkazov však kontroluje operátor a nie bezpečnostne relevantná časť zabezpečovacieho systému a operátor len potvrdzuje splnenie týchto podmienok. Väčšina elektronických staničných zabezpečovacích systémov preto vykonanie núdzového príkazu podmieňuje zvláštnym potvrdením príkazu operátorom – potvrdenie sa spravidla realizuje zadaním stanovenej kombinácie znakov z klávesnice systému (napríklad kombinácia znakov „asdf.“ v prípade elektronického zabezpečovacieho systému ESA výrobcu AŽD Praha, alebo kombinácia znakov „ano.“ v prípade elektronického stavadla EBILOCK 950 výrobcu Bombardier) alebo zvláštnou potvrdzovacou sekvenciou obslúh myšou (napríklad kliknutie na tlačidlo s textom „OK“ resp. „Áno“, ktoré sa zobrazí spolu so žiadosťou o potvrdenie núdzového príkazu v samostatnom okne vygenerovanom na zobrazovacej jednotke po zadaní núdzového príkazu v prípade elektronického stavadla SIMIS W výrobcu Siemens).

V prípade, že operátor v stanovenom čase požadovaným spôsobom nepotvrdí vykonanie požadovaného núdzového príkazu, núdzový príkaz sa nevykoná.

Všetky elektronické systémy majú okrem toho implementovanú automatickú registráciu všetkých núdzových príkazov, ktorá na médiu nedostupnom operátorovi systému zaznamenáva všetky núdzové príkazy operátora súčasne s identifikáciou operátora a časom vydania príkazu.

### Záver

Hoci moderné elektronické zabezpečovacie systémy používané na riadenie pohybu koľajových vozidiel v železničnej stanici majú aplikované techniky na dosiahnutie SIL 4, na základe požiadaviek používateľov musia umožňovať aj núdzové ovládanie riadeného dopravného procesu. Použitie núdzových príkazov však môže znížiť celkovú bezpečnosť zabezpečovacieho systému vzhľadom k požiadavke používateľov, že núdzové príkazy musia byť dostupné v každom prevádzkovom stave systému. V prípade, že operátor systému nesprávne vyhodnotí prevádzkovú situáciu a núdzový príkaz využije neoprávnené alebo ho aplikuje na nesprávny prvok, môže riadený dopravný proces priviesť do nebezpečného stavu. Na minimalizovanie rizika vzniku takejto situácie výrobcovia systému zavádzajú dodatočné bezpečnostné opatrenia spočívajúce vo zvýšení bezpečnosti zobrazovania pri použití núdzových príkazov a v zavedení dodatočného potvrdzovania a archivovania všetkých núdzových príkazov.

Napriek všetkým uvedeným opatreniam však nemožno vylúčiť riziko takého zlyhania operátora, že neoprávnené použije núdzový príkaz v bezporuchovom stave zabezpečovacieho systému a riadený dopravný proces privedie do nebezpečného stavu, ktorý vyvrcholí vznikom dopravnej nehody. Je preto na zváženie, či by v prípade bezporuchového stavu riadiaceho systému, nemala byť obmedzená možnosť použitia núdzových príkazov. To by si však vyžadovalo zmenu prístupu používateľov zabezpečovacích systémov a s tým súvisiacu zmenu platnej legislatívy.

### PodĎakovanie

*Tento článok vznikol s podporou vedeckej grantovej agentúry SR VEGA v rámci riešenia grantu č. VEGA-1/0388/12 „Kvantitatívne hodnotenie integrity bezpečnosti riadiacich systémov pre dráhové aplikácie“.*

### Literatúra

- [1] STN EN 50 129: Dráhové aplikácie. Komunikačné a signalizačné systémy a systémy na spracovanie údajov. Elektronické signalizačné systémy súvisiace s bezpečnosťou. 2004.
- [2] ZAHRADNÍK, J., RÁSTOČNÝ, K., KUNHART, M.: Bezpečnosť železničných zabezpečovacích systémov. EDIS – vydavateľstvo ŽU, ISBN 80-8070-296-9, 2004.
- [3] ZAHRADNÍK, J., RÁSTOČNÝ, K.: Aplikácie zabezpečovacích systémov. EDIS – vydavateľstvo ŽU, ISBN 80-8070-546-1, 2006.
- [4] RÁSTOČNÝ, K., RÁSTOČNÝ, K. Jr.: UML – a Part of an Interlocking System Development Process. In: Communications in Computer and Information Science, Volume 329, Telematics in the Transport Environment, Berlin: Springer-Verlag, s. 293-300, ISSN 1865-0929, ISBN 978-3-642-34049-9, 2012.
- [5] RÁSTOČNÝ, K., ŽDANSKÝ, J., NAGY, P.: Some Specific Activities at the Railway Signalling System Development. In: Communications in Computer and Information Science, Volume 329, Telematics in the Transport Environment, Berlin: Springer-Verlag, s. 349-355, ISSN 1865-0929, ISBN 978-3-642-34049-9, 2012.

### Abstract

The paper deals with emergency operation of the control systems used to control of railway traffic. It focuses primarily on interlocking devices, where the use of emergency operation is the most frequently. Conventional interlocking based on solid logic is designed so that the failure of the interlocking leads to a reduction of the railway traffic in controlled area. In this case the use of emergency operations allows traffic remaining. The cost for this is reducing of the controlling safety level. Modern electronic systems with programmable logic have applied to the mechanisms that allow the use of emergency operations while maintaining of a sufficiently high of safety integrity level.

### Ing. Peter Nagy

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
Tel.: ++421 41 513 3357  
E-mail [peter.nagy@fel.uniza.sk](mailto:peter.nagy@fel.uniza.sk)

# Using Context Blocks in Genetic Programming with JIT Compilation

Michal Gregor, Juraj Spalek

## Abstract

The paper concerns work relating to the use of contexts and context blocks in genetic programming. The paradigm provides an organised, abstract way of sharing information among multiple nodes in the syntax tree produced by genetic programming. The paper presents how the existing approach can be integrated with a system for parsing, source code generation, machine code generation: either by interpretation, or – more interestingly – by JIT compilation. The paper also explores how some of the existing application based on context blocks, such as context block constants and node-attached modules with ancestry tracking (NAMAT), will be able to transition to the new system.

**Keywords:** genetic programming, context blocks, evolution, modularity

## Introduction

The effectiveness of evolutionary methods and of genetic programming (GP) in particular has in the past years been made manifest by a number of researchers. The use of abstract syntax trees (ASTs) presents a well-understood abstraction commonly used in many existing conventional tools such as interpreters, compilers, static analysis tools and other. As shown by the advocates of genetic programming, ASTs also represents a viable representation for programs that are to be evolved as they allow for a relatively straight-forward implementation of genetic operators, such as mutation and crossover.

Even though genetic programming has now found its application in a wide range of domains, some issues still remain open for investigation, such as developing a robust approach to modularity – so that if parts of the solution can be reused and if the problem has hierarchical structure, GP can take full advantage of this. There is also a long-standing problem of evolving constants – this being the more embarrassing because the theory of genetic algorithms underlying GP has been developed exactly with that in mind.

To this end have the authors proposed the high-level concept of contexts and context blocks, which are to facilitate solution of multiple problems, including the two aforementioned ones.

Our current implementation is built in C++ and uses a straight-forward object-oriented abstract model of the AST, which is executed directly by recursively calling an overridden virtual method provided by each distinct node class. With the ongoing development of high-level tools for implementation of interpreters, compilers and code generators, such as the LLVM framework, it is becoming increasingly simple to create interpreters and even just-in-time compilers for simple domain-specific languages.

Due to these developments we now seek to extend our implementation to include said concepts and especially to provide integration between contexts and context blocks on one side and LLVM-based code generation on the other – all this with the view of making the framework more flexible,

and in order to make use of LLVM's optimization capabilities. Performance gains should also be achieved – especially when re-executing the same piece of code several times, which is often the case in GP.

This paper is supposed to outline the steps necessary to implement such solution. Furthermore, it explores issues and design decisions that are to be faced in course of extending contexts and context blocks with support for such features as required in order to integrate them with LLVM-based code generation and execution.

## 1. LLVM

In recent years programmers are continually being provided with valuable tools – be these on the side of integrated development environments, code management systems and various collaboration tools; or on the side of actual languages, program libraries and frameworks facilitating development.

Of the latter class is the LLVM compiler infrastructure – a collection of modular and reusable compiler and toolchain technologies [1]. Available as a C++ library, it provides groundwork upon which several compilers, interpreters and libraries are built. Most notably perhaps there is Clang – a well-known C/C++ compiler.

LLVM presents a tool that can – with a reasonable amount of effort – be used to provide interpretation, just-in-time compilation, optimization and more for a simple domain-specific language. In any case, this is the use that we have in mind specifically in this paper.

## 2. Flex, Bison

To complete the list of required tools let us also mention the well-known pair: the tool for building tokenizers called Flex [2] and the tool for writing parsers called Bison [3].

Let it suffice, for the sake of brevity, to state that Flex takes in what is basically a set of regular expressions and generates a tokenizer in C or C++, which can be used to

parse source code in the corresponding language into distinct tokens.

Bison provides similar functionality for generating parsers: it takes in an annotated context-free grammar and produces a corresponding C or C++ parser, which will typically be used to parse tokenized source code into an abstract syntax tree.

### 3. Genetic Programming

Genetic programming (GP) is a technique introduced by John Koza (see [4]). It utilizes the previously outlined concepts to evolve computer programs. The main idea of Genetic Programming revolves around the way in which the individuals are represented, that is to say around the syntactic trees (also known as parse trees). The problem will be analysed more specifically in the following sections.

#### 3.1 Representation

Naturally, we could use source-code directly to represent solution. However, such representation would hardly be suited to perform any reasonably complex sort of processing on, let alone the implementation of well-performing genetic operators. The main idea of genetic programming then is to use abstract syntax trees (ASTs) as genotype. ASTs had already been in widespread use in interpreters, compilers, and static analysis tools before they were introduced as a possible choice for representation of individuals in genetic programming. They have since also shown themselves an appropriate representation for implementation of genetic operators.

An example of a simple syntax tree is given in Fig. 1. It encodes the expression  $x \cdot y + \ln x$ .

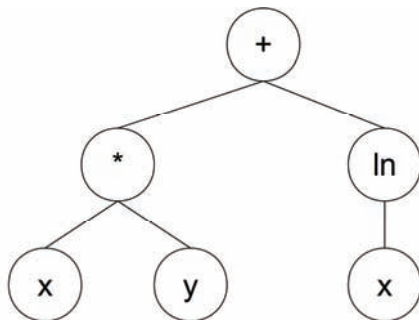


Fig. 1 A simple example of an AST

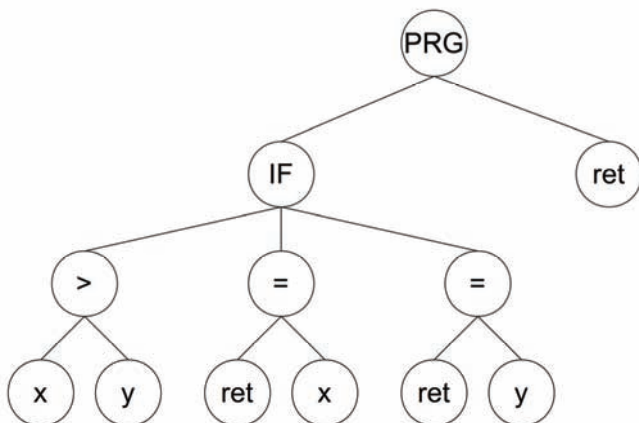


Fig. 2 A more complex AST

A more advanced example including mechanisms such as conditional execution, assignment and return is shown in Fig. 2. The root node – PRG is a node with one of more child nodes. The value produced by the last of these is returned as the return value of the program.

The program shown in Fig. 2 can also be expressed by C++ code given in Listing 1.

#### Listing 1 Code expressed by Fig. 2

```

1. if(x > y) ret = x;
2. else ret = y;
3. return ret;
  
```

#### 3.2 The Closure Property

The representation as proposed by Koza has one important property, known as the *closure property*, which requires that any valid tree generated from a set of terminals:

$$T = \{t_1, t_2, \dots, t_n\}, \quad (1)$$

and a set of non-terminals:

$$NT = \{t_1, t_2, \dots, t_m\}, \quad (2)$$

represents a valid program, which states that any non-terminal should be able to handle as an argument any data type and value returned from a terminal of non-terminal.

In contrast to this approach, several researches focus on the so-called *strongly typed genetic programming* [5], where nodes are allowed to have different incompatible return and argument types. In this case, type constraints have to be enforced, which introduces several fundamental differences. The most notable aspect is that when generating, crossing or mutating a tree care has to be taken to ensure that the return type of the node used as an input is compatible with the data type of the input itself.

The closure property can still be enforced in strongly typed genetic programming using dynamic typing. Non-terminals can be built so that they accept an argument of any type, but throw an exception if type id of the argument is not as expected.

#### 3.3 Crossover and Mutation

As mentioned the AST-based representation has been chosen expressly for the purpose of making implementation of crossover and mutation more straight-forward.

As it is, crossover can simply be implemented by swapping two randomly selected subtrees of two parent individuals, while mutation can be implemented in several ways, e.g. by replacing a subtree by a newly-generated one.

### 4. Parsing and Code Generation

As mentioned, the paper considers how best to integrate the existing context and context block approach with code generation based on LLVM and hopefully also with parsing based on Flex/Bison.

Our existing framework only works with a single representation of the solution: the AST. While the AST can be converted into plain text, this is by no means a full-fledged alternative representation, and cannot be parsed back into an AST, or processed in any way.

What we now seek is to provide support for two distinct modes of code generation:

- Generation of source code from the AST (to be made parsable back into an AST using Flex/Bison);
- Generation of machine code using LLVM.

These features will require some adjustments to our existing framework, especially to our AST representation. Most of these, however, are implementation details, and as such are

not of much immediate interest to the reader. We will therefore focus our attention on those aspects of the solution, which directly concern integration of contexts and context blocks with these features.

## 5. Concerning the AST Representation

Let us now make a few observations concerning our existing representation of ASTs and the modifications and additions that will now be required.

### 5.1 The Abstract Interface

An AST is, in our implementation represented by class `Tree`, which does several things – most notably it holds the root node, it takes in arguments of prescribed types. When running the tree, these arguments are packed into an argument list so that they can be presented to every node upon its execution.

Nodes themselves have a minimum interface: they declare the number of arguments they take and specify their types. Child nodes with the specified types are presented to node's `process()` method as they are (as opposed to executing them first and presenting the node with their outputs). This is to allow for branching, such as IF nodes, loops and similar.

It is obvious that such representation is rather abstract. On one hand this allows the programmer to provide nodes with arbitrary behaviour, while assuming very little – no even the IF node is presumed. On the other hand, such level of abstraction may have its downsides. To these we now turn.

### 5.2 Making the Representation More Explicit

Although the AST interface outlined above has proved itself to be sufficiently expressive for use in GP, its highly abstract nature makes further operations on the AST cumbersome, and often even trivial optimizations such as collapsing constants are very difficult to write.

Thus, if the AST representation is to be more useful, and especially in order to allow for a reasonably straight-forward implementation of code generation, we mean to make the representation more explicit.

This can be achieved by modelling certain features of the nodes explicitly: like their being constants, variables, expressions, etc.

## 6. On Use of Contexts and Context Blocks

Let us now with all brevity turn to the concept of contexts and context blocks as presented in [6] (for further details consult the same). We will explain the motivation behind developing and using the concept first and then move on to consider what course to take when integrating it with LLVM-based code generation.

### 6.1 Motivation for Introduction of the Context Concept

When building a library that implements genetic programming, one is faced with a number of issues. This section lists some of these and shows how the *context* concept can help to alleviate them.

Regarding this, it is of the essence to note that nodes may and often will be moved into another tree. An instance of this is crossover, which, as mentioned, swaps selected subtrees of two ASTs.

To illustrate the various issues that this introduces, let us consider that some nodes may be in need of sharing data among themselves. Such data may in a certain way be attached:

- To the structure of the tree itself;
- To a particular execution of the tree;
- To some other feature of the system.

To provide an instance of data attached to the structure of the tree, we may mention functions and modules, which are a feature of the program, that is, they are tree-specific and callable from multiple points in the tree (usually from any point). As for data attached to a particular execution of a tree, variables may be supplied in the way of an example.

The solution we describe in [6] is to introduce the so-called *contexts*, with which nodes are presented so that they may store their data within them and afterwards share such data where appropriate. Furthermore, as different types of data need to be attached to different aspects of the system, there should be several contexts: at the very least a *tree context* (for tree-attached data) and a *process context* (created for a particular execution of a tree).

### 6.2 Variables as Execution-attached Data

Context block mechanics are best explained using the concept of variables. By a variable, we mean an object that can:

- Be assigned a value of a given type;
- Be converted to a value of that type;
- Hold its value until the execution of the tree stops (or, optionally, even on to the next execution);
- Be accessed from multiple points in the same tree.

Now, variables can – in our AST representation – be implemented using a terminal node which returns a special kind of object that represents the variable – other nodes can take this as an argument and operate on it (e.g. a non-terminal may be added that provides assignment, etc.). An example is shown in Fig. 3.

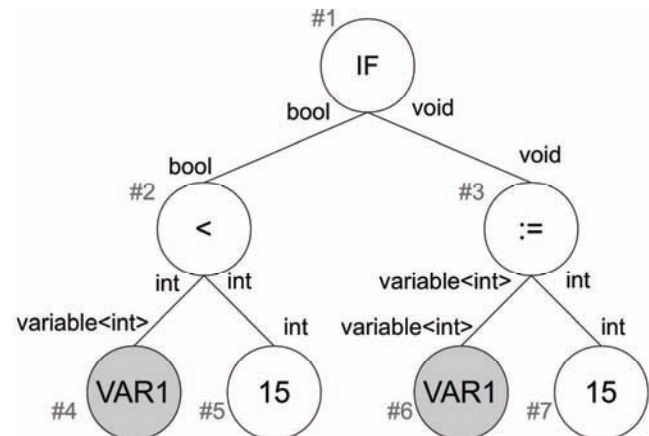


Fig. 3 AST with a variable

The tree shown in Fig. 3 can be translated into the following piece of pseudo-code (Listing 2):

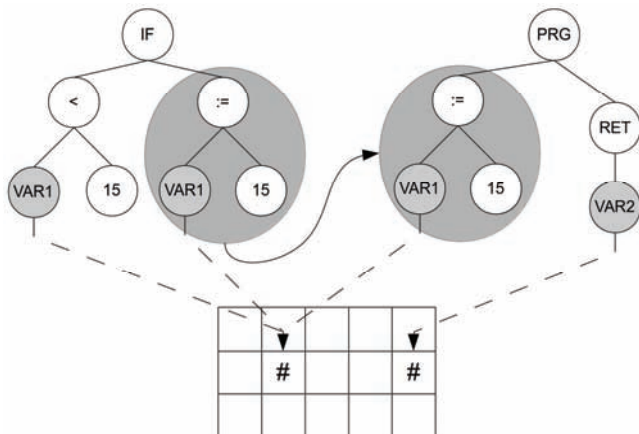
#### Listing 2 Code expressed by Fig. 3

```
1. if (var1 < 15) var1 = 15;
```

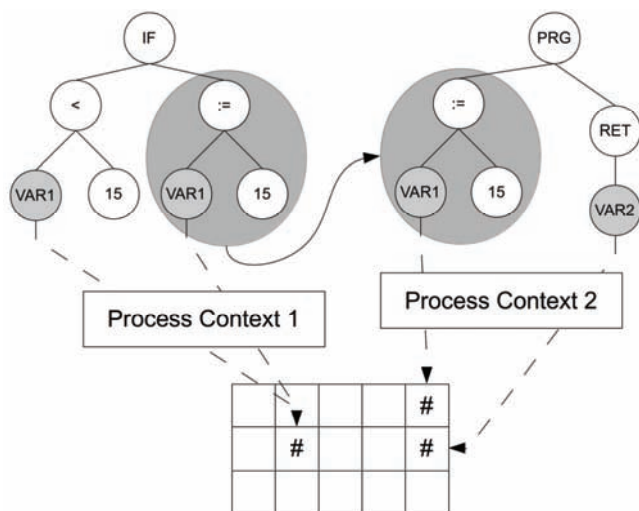
This example also illustrates that type `variable<int>` is convertible to `int`.

That being said, it should be obvious to the reader, that variable objects returned by node #4 and node #6 should (if their return value is to represent the same variable) point to the same piece of data, in this case, to the same integer. Although it seems that this problem can easily be accounted

for – e.g. by providing both objects with the same pointer to the data – the issue is, in fact, more complex than that.



**Fig. 4 Variables from two trees sharing the same data after crossover**



**Fig. 5 Variables from two trees storing data in process contexts**

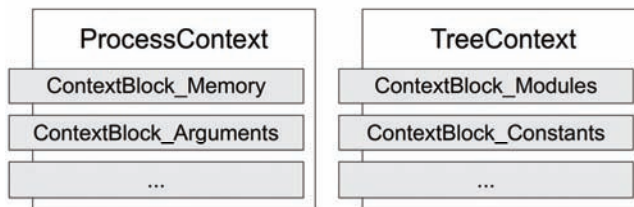
More specifically, it is complicated by the fact that it must be possible to move nodes #4 and #6 to another tree (in order to implement crossover), in which case all the trees would share the same piece of data. An example is shown in Fig. 4. Also, this does not work very well if the tree itself is required to be thread-safe (allowing for several parallel executions).

To alleviate such issues, a separate process context may be created for every execution and every distinct variable provided with a unique key under which it will store its value in the context. The principle is illustrated in Fig. 5.

### 6.3 Context Blocks and Their Most Notable Instances

The concept presented so far can itself be made more generic by introduction of context blocks, that is, arbitrary types that can be stored in and retrieved from contexts. One type of such context block has been presented already – the *memory context block* of the process context.

There are other types of context blocks, some of which will be listed and briefly described in the following sections. An overview is given in Fig. 6.



**Fig. 6 An overview of context and context block types**

There is a feature inherent to this approach that should receive some further attention: the context blocks can easily and transparently be accessed from other portions of code as long as these know about existence of that particular context block. It is thus possible to implement nodes that store and share some of their data using a context block and let other components of the program to operate on these data – for an instance to store and share modules in the tree context and implement a crossover operator that performs crossover on them – all this in a clear and modular fashion.

### 6.4 Constant Context Block

The context paradigm has been used by the authors to implement a system for evolving constants (which is a long-standing problem in GP). The approach is based on storing constants centrally on a per-tree basis, which – in combination with support for arbitrary custom genetic operators allows for embedding an entire genetic algorithm within the main GP process. For further information concerning the particulars of this solution as well as the results achieved, you may see [6].

### 6.5 Node-attached Modules with Ancestry Tracking

The context paradigm has also been used by the authors to implement node-attached modules with ancestry tracking (NAMAT). This mechanism is implemented as a distinct context block in the tree context. Paper [7] presents the method in detail.

In this place a few brief remarks will suffice – and these too will only be presented because the implementation of NAMAT modules has considerable bearing on our object. NAMAT modules are entities containing the following:

- Their code in the form of a subtree;
- Their function signature (by which an object storing the return type and types of the arguments is understood), as our system is strongly typed;
- A unique module ID;
- A list (of predefined length) storing module IDs of ancestor modules;
- A default return value.

Nodes calling modules refer to the same using the so-called *module tokens*. These are simply storage tokens, which identify a storage unit within the module context block, wherein the module resides. When provided with both – a module token and the tree context containing the module context block, a node is able to call the module.

Being contained in the individual, modules can easily be made subject to evolution by adding module-aware genetic operators.

Furthermore, for every NAMAT module there is a unique module ID, which changes every time the module is modified. Furthermore, every module ID is stored in a list prior to its being updated: thus, every module manages a list of its ancestor IDs, and thus can tell which modules are related to it. The list of ancestors is a circular buffer of predefined size.

The default return value is provided in case we allow modules to call other modules. In such case a recursion limit has to be specified. Once it is exceeded the module in question returns its default return value instead of executing its subtree. This is done in order to prevent infinite loops.

IDs themselves are implemented as char pointers: the unique identifier is the address of the char. The IDs can be serialized in the same way that pointers generally are: naturally if two copies of the same ID are to retain the same identity, they must be serialized and deserialized together.

When moving module-calling nodes from one tree to another, especially in the course of crossover, the module is moved with the module-caller. This is what we mean by saying that modules are *node-attached*. The rationale behind this is that module-callers should be able to rely on modules to serve essentially the same tasks even after they have been moved to another individual – otherwise highly fit code might perform very poorly in the new individual (this issue is termed *the problem of remapping* in our work).

However, should the individual into which the module-caller is being moved already contain a *related module* that is likely to serve a similar purpose, the module-caller should – instead of copying its module into this individual – be remapped to such existing module. The notion of modules being related to each other is implemented using the ancestry list already mentioned. We suppose that if a module has been modified less than a predefined number of times (given by the size of the list of ancestors) it is still likely to more or less serve a similar purpose.

## 7. Code Generation and Contexts

As mentioned hereinbefore, we currently make use of two main contexts: the *process context* and the *tree context*. Let us then treat of each of these separately.

### 7.1 Process Context

It should be evident that most functions now performed by the process context will – in the case of LLVM-based code generations be transferred to LLVM, which will for the most part be made responsible for managing memory for every running instance of the program.

Naturally, when generating source code, the process context will likewise not be necessary.

### 7.2 Tree Context

The same, however, is not true of the tree context. Most notably, constants and modules still have to be managed somehow. When generating machine code using LLVM, this, happily, is not of so much concern: from this point on, the identity of constants and modules ceases to matter. On the other hand, when generating source code, these must, if possible, be preserved, and that in such way as to be parsable back into ASTs.

This may be difficult in some cases: e.g. in that of ancestry tracking. While we may be quite able to serialize and deserialize lists of ancestor IDs and storage tokens, generating code will of a necessity be more difficult if it is to remain readable. Also, as mentioned regarding serialization of module IDs – if several copies of the same ID are to retain the same identity, they have to be serialized together.

Similarly, if code for several ASTs is generated and parsed separately, with no common context, module IDs, even if they are correctly exported and parsed will no longer be identical.

It seems that the easiest way to account for this case is to provide the source code generator and the parser with a common *translation context* that will allow us to circumvent the issue.

Furthermore, while simply working with the ASTs in the course of GP evolution, tree context should still be able to perform most of its functions without further changes.

## 8. Concerning the Translation Context

The translation context should – in the same way the other contexts do – accommodate an arbitrary number of context blocks, which can be accessed by any piece of code, which knows of their existence, and has access to the context itself.

Thus, a context block can easily be provided, which will create and maintain a lexicon (e.g. a bidirectional map, or a bidirectional hashmap), mapping from module IDs to strings, or numbers, and back.

Naturally, it is of the essence that the translation context be properly used. If, for example, source code is generated from an AST with module *MID1*, and *MID1* is translated into textual representation of “*module1*”, and code for another AST with module *MID25* is later generated, with *another context*, this other context may again map module id *MID25* into “*module1*”. It is obvious then, that under such circumstances it would not be advisable to use a single translation context when parsing these sources back into ASTs later, as the modules would then share the same ID, even though – far from being identical – they would not even be related.

## Conclusion

We have here presented some background concerning the concept of contexts and context blocks, the rationale behind it, its uses and existing applications such as context block constants and node-attached modules with ancestry tracking (NAMAT).

With the increasing availability of powerful tools such as the LLVM library, our framework is to be extended to cover parsing and code generation: for source code and machine code alike, the latter being accomplished using LLVM.

This step requires some additional work concerning the integration of said context and context block paradigm. Thus the main aim of this paper was to provide the necessary groundwork for such integration and for the transition of existing application based on context blocks.

Various issues concerning this have been discussed. Most notably – it has been argued that in order to effectively generate source code that can be parsed back into its AST equivalent, a new context is required: the translation context. It has been shown how such context can be utilized to implement code generation for NAMAT modules – specifically for their module IDs and their ancestry lists.

All vital aspects of the solution should thus now be provided for. It now only remains to complete the implementation. Evaluation of the added flexibility and of the performance gains resulting from the transition should form part of our future work.

It also remains to be investigated what further enhancements will be possible once some aspects of the new system are in place. This is especially true of the additional expressiveness gained by making the AST representation more explicit.

## Acknowledgement

This contribution/publication is the result of the project implementation:

**Centre of excellence for systems and services of intelligent transport, ITMS 26220120050** supported by the Research & Development Operational Programme funded by the ERDF.



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

## References

- [1] The LLVM Compiler Infrastructure [online]. [cit. 18-03-2013]. URL: <<http://llvm.org/>>.
- [2] Flex: The Fast Lexical Analyzer [online]. [cit. 18-03-2013]. URL: <<http://flex.sourceforge.net/>>.
- [3] Bison – GNU Parser Generator [online]. [cit. 18-03-2013]. URL: <<http://www.gnu.org/software/bison/>>.
- [4] KOZA, J.: On the Programming of Computers by Means of Natural Selection. Cambridge, Massachusetts: MIT Press, 1992. ISBN 0-262-11170-5.
- [5] MONTANA, D. J.: Strongly Typed Genetic Programming. Evolutionary Computation: 3, 1995. URL: <<http://vishnu.bbn.com/papers/stgp.pdf>>.
- [6] GREGOR, M., SPALEK, J., CAPÁK, J.: Use of Context Blocks in Genetic Programming for Evolution of Robot Morphology. In: *Proceedings of 9th International Conference, ELEKTRO 2012*, 2012. ISBN 978-146731179-3.
- [7] GREGOR, M., SPALEK, J.: On Use of Node-attached Modules with Ancestry Tracking in Genetic Programming. In: *Proceedings of the 11th IFAC/IEEE International Conference on Programmable Devices and Embedded Systems*, 2012. ISBN 978-3-902823-21-2.

## Abstrakt

Článok je venovaný práci týkajúcej sa použitia kontextov a kontextových blokov v genetickom programovaní. Táto paradigma predstavuje organizovaný a abstraktný spôsob zdieľania informácií medzi viacerými uzlami v syntaktickom strome vytvorenom genetickým programovaním. Článok prezentuje ako možno existujúce prístupy integrovať so systémom na parsing a generovanie zdrojového kódu a strojového kódu: jednak formou interpretácie a jednak – čo je zaujímavejšie – na základe JIT kompilácie. Článok tiež skúma, ako sa docieli prenesenie existujúcich aplikácií založených na kontextových blokoch – napríklad kontext blokových konštánt a modulov so sledovaním pôvodu viazaných k uzlom – do nového systému.

## Ing. Michal Gregor

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
E-mail: [michal.gregor@fel.uniza.sk](mailto:michal.gregor@fel.uniza.sk)

## prof. Ing. Juraj Spalek, PhD.

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
E-mail: [juraj.spalek@fel.uniza.sk](mailto:juraj.spalek@fel.uniza.sk)



# Fitness-based Adaptive Control of Parameters in Genetic Programming

Michal Gregor, Juraj Spalek

## Abstract

This paper concerns applications of genetic algorithms and genetic programming to tasks for which it is difficult to find a representation that does not result in a highly complex and discontinuous fitness landscape. In such cases the algorithm is prone to getting trapped in local extremes. The paper proposes several adaptive mechanisms that are useful in preventing the search from getting trapped.

**Keywords:** genetic algorithms; genetic programming; adaptive parameter setting

## Introduction

Genetic algorithms and genetic programming represent a well-known optimization method. Among their strengths is the flexibility of representation, which allows for their application to a wide variety of tasks. Even though the representation of the solution is not forced to take a single prescribed form, it is still required that it follow certain guidelines so that the algorithm is able to search the resulting fitness landscape with reasonable efficiency. In certain tasks, such as evolving algorithms with memory it may be very difficult to find a representation that does not result in a highly complex and discontinuous fitness landscape, which makes the process of search prone to getting trapped in local extremes. This paper presents several adaptive mechanisms that aim to improve properties of the standard genetic programming with respect to this problem. These are all based on the observation that it is often possible to help the algorithm escape from local maxima by introducing new genetic material to the process. There are multiple ways to achieve this, some of which will be presented in this paper. Following sections present respectively – the model problem to which the adaptive mechanisms were applied; an overview of existing approaches to parameter control; and description of the proposed adaptive mechanisms as well as the results.

### 1. THE MODEL PROBLEM

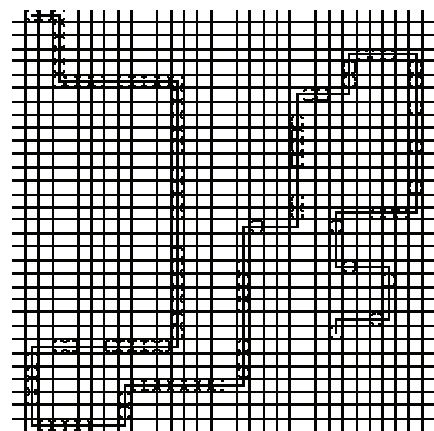
Let us first briefly describe the model problem on which the results achieved by various versions of the proposed adaptive mechanisms are to be presented – the artificial ant problem. Application of genetic programming to this task has been described in detail by John Koza [1].

The artificial ant problem is essentially a trail-following task. The actor – an artificial ant is to navigate in an environment following an irregular path consisting of pieces of food which it is supposed to collect. The ant has severely limited sensing capabilities – it only sees a single tile that is right in front of it.

The solution is represented by a simple syntactic tree. Automatically defined functions, recursion and other advanced concepts are not utilized.

In contrast to Koza's original application, where the syntactic tree represents a controller (i.e. the program runs, orders the agent to perform a certain action, waits for its completion and then continues to run the same way until the end of the program is reached at which point the whole syntactic tree is re-executed), the mode of execution has been modified in our implementation – the whole syntactic tree is now run in every step and it is used to determine which action the agent should take. This means that some explicit memory model, such as indexed memory, is now required to make account of previous inputs and actions.

This approach has several advantages, but it also makes the solution considerably more difficult to evolve. Some of the underlying issues have been discussed by Astro Teller in [2]. We shall, at this point, confine ourselves to concluding that standard genetic programming has shown itself to be unable to solve the model problem in the prescribed mode (that is, none of the solutions has achieved maximum fitness). The search easily becomes trapped in local extremes of the fitness landscape.



**Figure 1** The Santa Fe trail problem “Santa Fe trail” problem:

[https://www.google.sk/search?q=santa+fe+trail+problem&tbm=isch&tbo=u&source=univ&sa=X&ei=2Y60UbbeEsHNhAestoC4Cw&ved=0CF0QsAQ&biw=1018&bih=520#facrc=\\_&imgsrc=TUcyZ\\_\\_6CzmK4M%3A%3BzWXt988hDB3irM%3Bhttp%253A%252F%252Fupload.wikimedia.o](https://www.google.sk/search?q=santa+fe+trail+problem&tbm=isch&tbo=u&source=univ&sa=X&ei=2Y60UbbeEsHNhAestoC4Cw&ved=0CF0QsAQ&biw=1018&bih=520#facrc=_&imgsrc=TUcyZ__6CzmK4M%3A%3BzWXt988hDB3irM%3Bhttp%253A%252F%252Fupload.wikimedia.o)

The parameters of the algorithm are as follows: the maximum of 150 generations; 500 individuals; the maximum depth of the tree set to 10; the Santa Fe trail (Fig. 1).

## 2. EXISTING APPROACHES TO PARAMETER CONTROL

As mentioned, in some applications based on the theory of genetic algorithms and genetic programming, the fitness landscape can be so complex that additional techniques may be required in order to find the global optimum.

Among the approaches that aim to enhance various properties of the standard algorithm, such as convergence speed and resilience to getting trapped in local extremes, are different parameter setting schemes. There are many different approaches most of which would fall into one of the following categories [3], [4]:

- static parameter control,
- dynamic parameter control,
- adaptive parameter control,
- self-adaptive parameter control.

### Static Parameter Control

The common feature of static parameter control approaches is that the setting they provide remains constant for the entire duration of the evolutionary process. There are many works analysing the problem of finding optimum settings for parameters like mutation rate and crossover rate. Some of these are listed in [5], e.g. the work of Mühlenbein, which proposes the following formula for the mutation rate:

$$p_m = 1/L \quad (1)$$

where  $L$  is the length of the bit string by which the individual is represented.

### Dynamic Parameter Control

As stated in 5, dynamic parameter approaches typically prescribe a deterministically decreasing schedule over a number of generations and provides a formula for mutation rate derived by Fogarty:

$$p_m(t) = \frac{1}{240} + \frac{0.11375}{2^t} \quad (2)$$

where  $t$  is the generation counter.

Articles 5, 5 also both refer to a more general expression derived by Hesser and Männer:

$$p_m(t) = \sqrt{\frac{\alpha}{\beta}} \times \frac{\exp\left(\frac{-\lambda t}{2}\right)}{\lambda \sqrt{L}} \quad (3)$$

where  $\alpha$ ,  $\beta$ ,  $\gamma$  are constants,  $\lambda$  is the population size and  $t$  is the generation counter and  $L$  is again the length of the bit string.

### Adaptive Parameter Control

Adaptive parameter control techniques monitor the search process itself and provide feedback. Some examples can be found in 5. The authors propose the following formulas for crossover and mutation probability respectively:

$$p_c = \begin{cases} k_1 \frac{f_{max} - f'}{f_{max} - \bar{f}} & f' > \bar{f} \\ k_3 & f' \leq \bar{f} \end{cases} \quad (4)$$

$$p_m = \begin{cases} k_2 \frac{f_{max} - f}{f_{max} - \bar{f}} & f > \bar{f} \\ k_4 & f \leq \bar{f} \end{cases} \quad (5)$$

where  $f$  is the fitness value of the individual to be mutated,  $f'$  is the larger of the fitness values of the individuals to be crossed and  $k_3$  and  $k_4$  are constants. It is required that  $k_1$  and  $k_2$  be less than 1,0 in order to constrain  $p_c$  and  $p_m$  to the range of (0,1). The  $p_c = k_3$   $f' \leq \bar{f}$  and  $p_m = k_4$   $f \leq \bar{f}$  expressions are to prevent crossover and mutation probabilities from exceeding 1,0 for suboptimal solutions.

The authors call this approach AGA (Adaptive Genetic Algorithm). Paper 5 also observes that  $p_c$  and  $p_m$  are zero for the solution with maximum fitness and that  $p_c = k_1$  for  $f' = \bar{f}$ , while  $p_m = k_2$  for  $f = \bar{f}$ . For further details and for information concerning setting the values of the constants refer to [5].

### Self-adaptive Parameter Control

When using the self-adaptive parameter control approach, parameters such as mutation rate and crossover probability of each individuals are part of its genome and are evolved with it. As stated in [5], the idea behind this is that a good parameter value will provide an evolutionary advantage to the individual. For further reference see [5].

## 3. ADAPTIVE VALUE-SWITCHING OF MUTATION RATE

Most of the existing parameter setting mechanisms, as presented in the previous section, either focus on setting GA-specific parameters such as length of the bit string (e.g. rule (1)), or are not adaptive (e.g. (1), (2) and (3)). The AGA mechanism described in [5] behaves adaptively, but its purpose is to speed up convergence, which (as shown later) makes the problem worse as AGA does not discern between local and global optima and thus the probability of getting trapped increases.

Furthermore – as mentioned hereinbefore – equations (4) and (5) assign the best individual zero crossover and mutation probabilities, while assigning high probabilities to less fit individuals. The reasoning behind this is that the less fit individuals can safely be disrupted by high mutation rates and recombined by crossover, thus employing the solutions with subaverage fitness to search the space of solutions 5, while highly fit individuals are preserved.

However, such approach has a very obvious downside which does not seem to be addressed – the highly fit individuals obviously contain the most excellent genetic material available and by disallowing mutation and crossover for these individuals the genetic code they carry effectively becomes isolated and is not used to generate new solutions.

### Description of the AVSMR Mechanism

The idea that the most fit solutions should survive crossover and mutation unmodified is obviously valid, yet that feature can be enforced by using elitism (i.e. the best individual is

copied to the next generation unmodified). Keeping that in mind we propose an adaptation scheme called AVSMR: *Adaptive Value-switching of Mutation Rate*. The main idea is that the mutation rate should be increased to a high value when the search has become trapped in an extreme so as to provide the search process with new genetic material some of which may previously have been unavailable. To determine whether the search has become trapped the adaptive mechanism observes the change of average fitness in time.

To describe the solution in more detail – the algorithm works with 2 values of mutation probability – the normal value and the high value. The algorithm switches from the normal value to the high value once the trigger criterion activates.

The trigger criterion itself is based on a measure that we will herein term a *delta sum*:

$$\Delta S_i = \alpha \cdot \Delta S_{i-1} + \frac{\bar{f}_i - \bar{f}_{i-1}}{\bar{f}_i} \quad (6)$$

where  $\Delta S_i$  is the delta sum in generation  $i$  and  $\bar{f}_i$  is the average fitness in generation  $i$  and  $\alpha$  is the feedback coefficient (the experiments have been carried out for  $\alpha = 0,4$ ).

If the delta sum is lower than a preset value for a predefined number of generations, that is to say the increase of average fitness in the last few generations is low, indicating that the search has become trapped – the mutation rate is set to its high value so as to provide the search with new genetic material. As mentioned before, when used in conjunction with elitism it is guaranteed that the best solution is not destroyed by the high mutation probability.

The mutation probability is reset back to its normal value when at least one of the following conditions is true:

- the average fitness increases enough to produce a sufficiently large delta sum;
- the maximum fitness increases;
- mutation has been set to its high value for at least  $n$  generations.

The  $n$ -generation limit is to ensure that the activation does not go on indefinitely (with the high mutation probability it is not very likely that the average fitness will increase sufficiently to satisfy the first condition and maximum fitness may not increase as well).

It has been observed that average fitness typically decreases when the criterion activates because the search process is to a large extent disrupted by the high mutation rate. However after the  $n$ -generation limit forces the mutation rate back to its normal value, average fitness tends to increase rapidly, thus usually moving away from the local extreme.

## Experimental Results

Several experiments have been carried out. Fig. 2 shows performance of the search algorithm with the AGA adaptive mechanism proposed in [5] (with constants set according to recommendations). It also shows performance of the search algorithm without any adaptive mechanism and with the AVSMR mechanism proposed in this paper. The maximum fitness value achieved is shown for each of the 5 runs displayed.

As shown, search achieves suboptimal results when running with no adaptive mechanism. This can be ascribed to its inability to escape from local extremes. With no adaptive mechanism the search has not found the global optimum (fitness = 89) in any of the 5 runs.

As mentioned, the AGA mechanism causes further deterioration and its results are thus even worse than those produced by the standard algorithm.

The AVSMR mechanism proposed in this work improves the process of search – in 2 of the runs the global optimum is found, yet in certain cases not even the high mutation rate is guaranteed to help the search to escape from local maxima (runs 2, 3, 4).

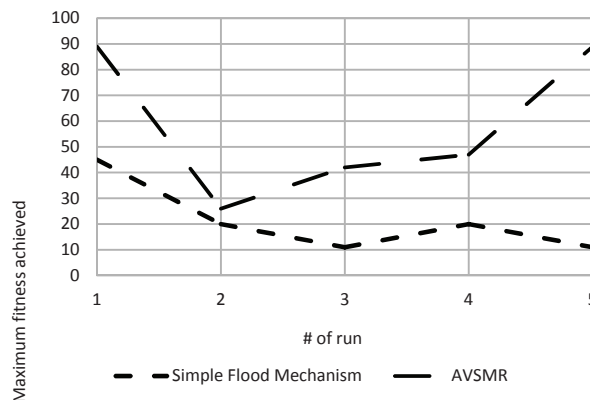


Figure 2 Comparison of the AVSMR and the Simple Flood Mechanism

## 4. THE SIMPLE FLOOD MECHANISM

The AVSMR mechanism described in the previous section is helpful in controlling the search process by helping it to escape from local extremes, yet it is not completely reliable and not always effective. To address these issues, we have developed another adaptive scheme supposed to provide new genetic material to an even larger extent.

### Simple Flood Mechanism

The principle is very straight-forward – once a trapping is detected – a relatively small part of the population is selected: these individuals survive. The rest of the population is destroyed and replaced by newly generated individuals. This method is superior to AVSMR in that a large part of the population is guaranteed to be replaced and the newly generated individuals can be (and need not be) generated in the same way that the initial population was.

The trigger criterion has also been modified. The first requirement is that the criterion only activates for a single generation at a time as it would probably be useless and possibly even counterproductive to activate the flood mechanism for several successive generations.

The new trigger criterion is still based on the average fitness  $\bar{f}_i$  (where  $i$  is the number of generation). The criterion stores average fitness  $\bar{f}_i$  for  $N$  generations (that is,  $N - 1$  previous generations and the current one;  $N = 7$  generations was used in the experiments). The mechanism cannot activate before the  $\bar{f}_i$  for at least  $N$  generations has been collected. Once that is true, the mechanism activates if the following holds:

$$\sum_{i=j}^{j-(N-2)} \bar{f}_i - \bar{f}_{i-1} < \theta \quad (7)$$

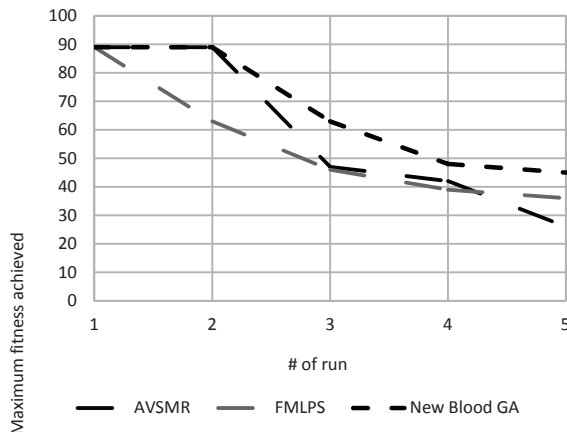
where  $j$  is the number of current generation and  $\theta$  is an activation threshold. It is also possible to interpret the threshold as a relative parameter in which case we can rewrite the equation as follows:

$$\frac{\sum_{i=j}^{j-(N-2)} \bar{f}_i - \bar{f}_{i-1}}{\bar{f}_j} < \theta \quad (8)$$

All experiments were carried out using (8).

It is also important to note that once the mechanism activates, the array storing the previous values of average fitness is cleared so it is guaranteed that the mechanism does not activate for the next  $N$  generations.

Although the approach seems straight-forward and similar in concept to AVSMR, experimental results point out an important issue. As obvious from Fig. 3, the results achieved by the Simple Flood Mechanism are significantly worse than those produced by AVSMR – they are in fact worse than those produced by the standard algorithm.



**Figure 3 AVSMR, FMLPS and the New Blood Mechanism**

The reason behind this is very simple – although we do introduce new genetic material into the process, the newly generated individuals will generally have very low fitness (usually 0, 3, or 4 at most). Therefore if we apply fitness-proportionate selection to these in the next generation, almost every newly generated individual will be discarded. The survivors on the other hand will now dominate the population. This is especially true later in the evolutionary process when fitness score of the best individual will tend to be vastly greater than that of any randomly generated individual. At this point the next generation will be formed almost exclusively by the best individual, which will almost in every case aggravate the problem of getting trapped in a local extreme instead of solving it.

### Flood Mechanism with Low-pressure Scaling and the New-Blood Mechanism

There are several ways to alleviate the problem that the Simple Flood Mechanism faces. The objective is – in any case – to create such scheme in which the newly generated individuals mate with the survivors so as to make use of their potentially useful code.

This paper proposes two different ways to achieve this:

- apply a fitness scaling function with low selection pressure to the GA for several generations following the flood – this mechanism will be referred to as *Flood Mechanism with Low-pressure Scaling (FMLPS)*;
- once the mechanism activates, create only such mating pairs in which at least one individual is newly generated – this mechanism will be referred to as the *New Blood Mechanism*.

The experimental results are shown in Fig. 3. FMLPS uses power scaling of 0,3 as the low-pressure scaling. To make the comparison easier, the values are now ordered by fit-

ness rather than by the number of run. This shows that AVSMR is still superior to FMLPS (although FMLPS is – in contrast to the Simple Flood Mechanism presented earlier – significantly better than standard GP). The New Blood GA on the other hand is definitely superior to AVSMR – although it still gets trapped in local extremes, the maximum fitness values achieved are generally greater than those achieved by the AVSMR.

## 5. SUGGESTIONS FOR FURTHER WORK

It has been shown that the adaptive mechanisms described in this work are able to effect considerable improvements. They are able to prevent getting trapped in local maxima to a certain extent. Further experiments should now be carried out to test usefulness of these approaches in a wider range of applications for which genetic algorithms and genetic programming are prone to getting trapped in local extremes.

It has also become apparent that even with the high mutation rates or with various versions of the flood mechanism it is not always possible to guarantee success. Value-switching, or even piecewise continuous relationships for other parameters could perhaps help to alleviate the problem to a further extent.

The trigger criteria would also require some additional work in order to become more stable and robust.

## 6. CONCLUSION

It has been shown that genetic algorithms and genetic programming may be applied to such tasks in which it is difficult to find a representation that does not map into a highly complex fitness landscape. Evolving algorithms with explicit memory concepts, such as indexed memory, may be considered an example of such task as shown by Teller in 5 and demonstrated on the model problem in this paper.

Tasks with highly complex and discontinuous fitness landscape are most difficult to solve using standard genetic algorithms and genetic programming as both of these methods are likely to get trapped in local extremes when applied to such problems.

The paper presents several adaptive mechanisms – the AVSMR mechanism, the FMLPS mechanism and the New Blood GA. These mechanism are shown to be able to alleviate the problem to a certain extent, thus achieving significantly better results than the standard algorithm.

Although the results compare favorably to those of the standard algorithm, even the proposed mechanisms cannot always guarantee that the search process will indeed escape from every local extreme that it encounters.

Related techniques such as adaptive value-switching or piecewise continuous relationships for other parameters of the search algorithm might provide further improvements. The influence that some of the flood mechanism related parameters (such as the number of survivors, or the selection pressure applied by the low-pressure scaling) have on the process of search may also provide area for further research.

### Acknowledgement

*This contribution/publication is the result of the project implementation:*



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

## 7. References

- [1] KOZA, J.R.: Genetic Programming: On the Programming of Computers by means of Natural Selection. Series in Complex Adaptive Systems, Cambridge, Massachusetts: MIT Press, 1992.
- [2] TELLER: "Genetic programming, indexed memory, the halting problem, and other curiosities," Proceedings of the 7th annual Florida Artificial Intelligence Research Symposium, 1994, p. 270–274.
- [3] EIBEN, A.E., HINTERDING, R., MICHALEWICZ, Z.: Parameter control in evolutionary algorithms, Evolutionary Computation, IEEE Transactions on, vol. 3, 1999, p. 124–141.
- [4] THIENS, D.: "Adaptive mutation rate control schemes in genetic algorithms," Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC'02., IEEE, 2002, p. 980–985.
- [5] SRINIVAS, M.: "Adaptive probabilities of crossover and mutation in genetic algorithms," Systems, Man and Cybernetics, IEEE, vol. 24, 1994, pp. 656-667.
- [6] "Santa Fe trail" problem:  
[https://www.google.sk/search?q=santa+fe+trail+problem&tbm=isch&tbo=u&source=univ&sa=X&ei=2Y60UbbeEshNhAestoC4Cw&ved=0CF0QsAQ&biw=1018&bih=520#facrc=&imgcr=TUcyZ\\_6CzmK4M%3A%3BzWXt988hDB3irM%3Bhttp%253A%252F%252Fupload.wikimedia.org%252Fwikipedia%252Fcommons%252Fthumb%252Fb%252Fba%252FSantaFeTrail.gif%252F220px-SantaFeTrail.gif%3Bhttp%253A%252F%252Fen.wikipedia.org%252Fwiki%252FSanta\\_Fe\\_Trail\\_problem%3B220%3B220](https://www.google.sk/search?q=santa+fe+trail+problem&tbm=isch&tbo=u&source=univ&sa=X&ei=2Y60UbbeEshNhAestoC4Cw&ved=0CF0QsAQ&biw=1018&bih=520#facrc=&imgcr=TUcyZ_6CzmK4M%3A%3BzWXt988hDB3irM%3Bhttp%253A%252F%252Fupload.wikimedia.org%252Fwikipedia%252Fcommons%252Fthumb%252Fb%252Fba%252FSantaFeTrail.gif%252F220px-SantaFeTrail.gif%3Bhttp%253A%252F%252Fen.wikipedia.org%252Fwiki%252FSanta_Fe_Trail_problem%3B220%3B220)

Ing. Michal Gregor

prof. Ing. Juraj Spalek, PhD.

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacích a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
E-mail: michal.gregor@fel.uniza.sk

# Creating the model of real actuator for the purpose of design the controller

Jozef Hrbček, Vojtech Šimák, Ľubomír Pekár

## Abstract

Jet fans are installed on the ceiling or side walls of the tunnel to create the air stream. The jet fans take in tunnel air and blow it out at higher speed along the axis of the tunnel. In the tunnel is mounted several fans with about hundred kW each. The design and industrial implementation of automatic control systems requires powerful and economic techniques together with efficient tools. In order to solve a control problem it is necessary to describe somehow the dynamic behaviors of the system to be controlled. Traditionally this is done in terms of a mathematical model.

**Keywords:** system modeling, jet fan, multivariable identification

## Introduction

During the last years the computer simulations of physical processes have been used in scientific research and in the system analysis and design of the systems controllers. The systems of interest are existing that operate at design conditions, deployment conditions, failure-mode conditions, or accident scenarios. The systems of interest are also the natural systems. Computer simulations are used for validation, analysis, improvement and predictions. These kinds of predictions are beneficial in the development of the control system, in the preparation of safety procedures and in the prevention of failures. This trend of modeling and simulation based design is primarily given by cost economize, e.g., automobiles, propulsion systems, tunnel ventilation systems, etc., where the need to fulfill the desired criterions. The necessary and sufficient amount of airflow ventilated in different traffic conditions in a tunnel must be provided with minimum electric power consumption. In particular, increasing the ventilation efficiency in tunnels is important to reduce the operating costs. The ventilation control system is based on programmable logic controllers. Since ventilation control system is a critical process, control system must meet safety requirements. About safety requirements is mentioned in [13].

## 1. System modeling

Mathematical models are mathematical expressions of essential characteristics of an existing system that describe knowledge about the system in a usable form. Turbulence inside the tunnel, variety of traffic and atmospheric make the system behavior stochastic. To make the models we can use the parametric identification. The tool MATLAB is used not only for models creation but also in model validation. The main tasks of system identification were the choice of model type and model order. It is advantageous obtain a linear time discrete model to consider the discrete character of controller. Although most real systems have non-linear input/output characteristics, many systems, when operated within nominal parameters (not "over-driven") have behavior

that is close enough to linear that LTI system theory is an acceptable representation of the input/output behavior.

System identification is the study of modeling dynamic systems from experimental data.

System (S): A defined part of the real world. Interactions with the environment are described by input signals, output signals and disturbances.

Model (M): A description of a system. The model should capture the essential behavior of the system.

The system Identification procedure:

- Collect Data. If possible choose the input signal such that the data has maximally informative.
- Choose Model Structure. Use application knowledge and engineering intuition. Most important and most difficult step (don't estimate what you know already).
- Choose the identification approach. How would a good model look like?
- Choose best model in model structure (optimization or estimation).
- Model validation. Is the model good enough for our purpose?

### 1.1 Determination of the transfer function

For Single-input Single-output the discrete transfer function can be used, that describe the relation between the input and output of the system with zero initial conditions. The transfer function  $F(z)$  is defined to be the ration of the response amplitude  $Y(z)$  to the input amplitude  $U(z)$ :

$$F(z) = \frac{Y(z)}{U(z)} = \frac{b_m z^m + b_{m-1} z^{m-1} + \dots + b_0}{a_n z^n + a_{n-1} z^{n-1} + \dots + a_0} \quad (1)$$

The transfer function equation (1) is an algebraic rational function of the variable  $z$ .

For Multiple-input Multiple-output system we can write the transfer function matrix. The discrete transfer function (TF) matrix,  $\mathbf{G}(z)$ , relates the  $Z$  transform of the response vector to the  $Z$  transform of the excitation vector for zero initial conditions by the equation:

$$Y(z) = G(z)U(z), \quad (2)$$

where;

$$Y(z) = Z[y_1(k), y_2(k), \dots, y_m(k)]^T; y_i(k) = i\text{th output}, \quad (3)$$

$$U(z) = Z[u_1(k), u_2(k), \dots, u_n(k)]^T; u_j(k) = j\text{th input}, \quad (4)$$

$$G(z) = \begin{bmatrix} \frac{Z_{11}(z)}{D_1(z)} & \frac{Z_{12}(z)}{D_1(z)} & \dots & \frac{Z_{1r}(z)}{D_1(z)} \\ \frac{Z_{21}(z)}{D_1(z)} & \frac{Z_{22}(z)}{D_1(z)} & \dots & \frac{Z_{2r}(z)}{D_1(z)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{Z_{r1}(z)}{D_1(z)} & \frac{Z_{r2}(z)}{D_1(z)} & \dots & \frac{Z_{rr}(z)}{D_1(z)} \end{bmatrix} \quad (5)$$

$D_i(z)$  equals the least common denominator of the  $i$ -th row of  $G(z)$  having the degree  $n_i$ , as seen by;

$$D_i(z) = \sum_{k=0}^{n_i} \alpha_k^i z^{n_i-k}, \quad \alpha_0^i = 1 \quad (6)$$

$Z_{ij}(z)$  equals the polynomial in  $z$  of the maximum degree  $n_{i-1}$ , as seen by

$$Z_{ij}(z) = \sum_{k=1}^{n_i} \beta_k^i z^{n_i-k} \quad (7)$$

System identification requires the determination of  $G(z)$ , i.e.,

$$\{Z_{ij}(z), D_i(z)\}; i = 1, 2, \dots, m; j = 1, 2, \dots, r, \quad (8)$$

from measurements of the system inputs and outputs. Before attempting to present the identification procedure, it is important to note that (5) is not valid for arbitrary initial conditions, i.e., arbitrary states at the start of the identification procedure. Since usually these initial states are neither known nor zero, a means must be provided to circumvent this difficulty.

The impact of jet speed variation on pollutant concentration is the first part of system and pollutant level change due to traffic intensity variation is the second part. The first part requires the jet fan model. The "Jet Fan Model" is a model that characterizes effect of ventilator on CO (carbon monoxide) concentration, NOx (oxides of nitrogen) concentration and visibility (opacity). It is a system with one input  $u(t)$  and three outputs  $y(t)$  (dilution of CO, NOx concentration and opacity). The second part assume a tunnel tube model that be taken as the model of measured disturbance of the ventilation system.

## 2. Real system identification

The "Jet Fan Model" is a multivariable model which can be described by multi-input, multi-output system:

$$S: A(z^{-1})Y(z) = B(z^{-1})U(z), \quad \det A(z^{-1}) \neq 0 \quad (9)$$

With  $p$  inputs  $u(k) \in R^p$  and  $m$  outputs  $y(k) \in R^m$  and polynomial matrices

$$A(z^{-1}) = I_{m \times m} + A_1 z^{-1} + \dots + A_n z^{-n}, \quad A_1, \dots, A_n \in R^{m \times m} \quad (10)$$

$$B(z^{-1}) = B_1 z^{-1} + \dots + B_n z^{-n}, \quad B_1, \dots, B_n \in R^{m \times p} \quad (11)$$

A characteristic problem for multivariable linear system is that, in general, there is no unique factorization ( $A(z^{-1}), B(z^{-1})$ ). For instance, the following two

factorizations represent the same transfer function despite their different parametrizations.

$$\begin{bmatrix} 1 - z^{-1} + z^{-2} & -z^{-1} + z^{-2} \\ -z^{-1} & 1 - z^{-1} \end{bmatrix} Y(z) = \begin{bmatrix} z^{-1} & z^{-1} \\ z^{-1} & -z^{-1} \end{bmatrix} U(z) \quad (12)$$

$$\begin{bmatrix} 1 - z^{-1} & 0 \\ -z^{-1} & 1 - z^{-1} \end{bmatrix} Y(z) = \begin{bmatrix} z^{-1} + z^{-2} & z^{-1} - z^{-2} \\ z^{-1} & -z^{-1} \end{bmatrix} U(z)$$

In fact, the second factorization is obtained from the first by multiplication of the polynomial matrices  $A(z^{-1}), B(z^{-1})$  from the left by the polynomial matrix

$$Q(z^{-1}) = \begin{bmatrix} 1 & z^{-1} \\ 0 & 1 \end{bmatrix}, \quad (13)$$

with

$$\det Q(z^{-1}) = 1 \quad (14)$$

Hence, given a multivariable function  $G(z^{-1})$  it is suitable to define an equivalence class of factorizations ( $Q(z^{-1})A(z^{-1}), Q(z^{-1})B(z^{-1})$ ) of  $G(z^{-1})$  for any stable, causal, and invertible polynomial matrix  $Q(z^{-1})$ . For any member of this equivalence class can be found the transfer function

$$(Q(z^{-1})A(z^{-1}))^{-1} Q(z^{-1})B(z^{-1}) = A^{-1}(z^{-1})B(z^{-1}) = G(z^{-1}) \quad (15)$$

Accordingly, any member of this equivalence class can be used to describe cross-coupling, delay, and other transfer function properties. An important conclusion is that assumptions on unique parametrizations are artificial assumptions that should be avoided unless there is an explicit a priori motivation. However, as for practical reasons it is desirable to use a finite number of well-defined parameters, it is often suitable to choose the parameter set with the smallest 2-norm. For the purpose of least-square identification, then, it is suitable to organize model and data according to [2]:

$$y_k = -A_1 y_{k-1} - \dots - A_n y_{k-n} + B_1 u_{k-1} + \dots + B_n u_{k-n}, \quad (16)$$

$$y_k \in R^m,$$

$$\phi_k = [-y_{k-1}^T \dots -y_{k-n}^T u_{k-1}^T \dots u_{k-n}^T]^T, \quad \phi_k \in R^{n(m+p)}, \quad (17)$$

$$\theta = [A_1 \dots A_n \quad B_1 \dots B_n]^T, \quad \theta \in R^{n(m+p) \times m}, \quad (18)$$

which suggests the linear regression model

$$M: \bar{y}_N = \phi_N \theta, \quad (19)$$

with

$$\bar{y}_N = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_N^T \end{bmatrix} \quad \text{and} \quad \phi_N = \begin{bmatrix} \phi_1^T \\ \phi_2^T \\ \vdots \\ \phi_N^T \end{bmatrix}. \quad (20)$$

The normal equations of the associated least-squares estimation of  $\theta$  will, as a result of the non-uniqueness of parameters, in general exhibit rank deficit.

It is therefore natural to apply the least-squares solution

$$\hat{\theta}_N = (\phi_N^T \phi_N)^+ \phi_N^T \bar{y}_N, \quad (21)$$

where  $(\phi_N^T \phi_N)^+$  denotes the matrix pseudo-inverse of  $W = \phi_N^T \phi_N$ , where  $W$  is  $n \times m$  matrix. If  $r = \text{rank}(W) = n$  then

$$W^+ = (A^T A)^{-1} A^* \quad (22)$$

If  $m=n=\text{rank}(A)$ , then  $A^+ = A^{-1}$ .

Data measured in the control centre of the tunnel are used to create the multivariable model. Effects of jet fan operation on concentration of pollutants in the tunnel tube are subject to identification.

Various gases are emitted by combustion engine. They consist largely of oxides of nitrogen (NOx), carbon monoxide (CO), steam (H<sub>2</sub>O) and particles (opacity). Because of these dangerous gases, it is necessary to provide fresh air in longer tunnels. The fresh air which is used to lower the concentration of CO also serves to improve visibility. The purpose of ventilation is to reduce the noxious fumes in a tunnel to a bearable amount by introducing fresh air. To create the air stream, jet fans are installed on the ceiling or side walls of the tunnel. The jet fans take in tunnel air and blow it out at higher speed along the axis of the tunnel. In the tunnel is mounted several fans with about hundred kW each. For design the control system we need the model of jet fan. Input value is switching on the jet fan and output values are CO (carbon monoxide) concentration, NOx (oxides of nitrogen) concentration and opacity inside the tunnel [5]. Traffic rate of cars and trucks, their speed, the concentrations of CO, NOx and Opacity are also available. These data are measured directly by sensors inside the tunnel tube.

Example of measured data in control center from tunnel tube:

- CCOxxCO - concentration of CO [ppm],
- CCOxOP - measured visibility [m<sup>-1</sup>],
- CCOxNX - oxides of nitrogen (NOx) [ppm],
- CUxxxxQ - airflow inside the tunnel tube [m<sup>3</sup>/s],
- CUxxxxVL - air velocity inside the tunnel tube [m/s],
- KxxxxNNV - number of trucks on camera,
- KxxxxNOS - number of cars on camera,
- KxxxxNV - vehicles speed [m/s],
- KxxxxDOP - total number of vehicles on camera,
- etc.

Measured data timestamp	CCO60 7OP [ppm]	CCO60 7CO [ppm]	CCO60 8OP [ppm]	CU X09 VL [m]	...
00:00:57.887	0.28	4.25	0.20	-1.7	...
00:01:57.600	0.28	3.99	0.20	-1.8	...
00:02:57.907	0.28	3.91	0.20	-1.7	...
00:03:57.322	0.27	3.56	0.21	-1.7	...
...	...	...	...	...	...

Tab.1 Example of measured data inside tunnel tube

The abbreviation *ppm* is a way of expressing very dilute concentrations of substances. Just as per cent means out of a hundred, so parts per million or *ppm* means out of a million. It describes the concentration of something in air.

Traffic intensity is sensed by a camera system and the cars are then counted and sorted by categories in database system. More information about monitoring of the traffic can

be found in [3] and [17]. About the security by transferring the data is discussed in [12] and [8]. Technical support of traffic control system is discussed in [7].

## 2.2 State space model

For Single-input Multiple-output discrete time linear systems we can also write the discrete state space model to describe the jet fan dynamics characteristics:

$$\begin{aligned} \mathbf{x}[(k+1)T] &= \mathbf{A}\mathbf{x}(kT) + \mathbf{B}u(kT) \\ \mathbf{y}(kT) &= \mathbf{C}\mathbf{x}(kT) + \mathbf{D}u(kT). \end{aligned} \quad (23)$$

In the above equations **A** is called the state matrix, **B** the input matrix, **C** the output matrix and **D** the feedforward matrix, whereas the vectors  $\mathbf{x}(t)$ ,  $u(t)$  and  $\mathbf{y}(t)$  are representing the state, input and output vector. The first equation (22) is called state equation and the second is called output equation, respectively.

This model describes the behavior of a system including the initial condition. Using the identification we get the matrixes **A**, **B**, **C** and **D**.

$$\begin{aligned} \bar{\mathbf{A}} &= \begin{bmatrix} -20 & -11.875 & -0.25 & 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1.2 & -0.032 & 0 & 0 \\ 0 & 0 & 0 & 0.0625 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -4.1636 & -0.0611 \\ 0 & 0 & 0 & 0 & 0 & 0.0313 & 0 \end{bmatrix} \\ \bar{\mathbf{B}} &= \begin{bmatrix} 0.25 \\ 0 \\ 0 \\ 0.0156 \\ 0 \\ 0.0313 \\ 0 \end{bmatrix} \\ \bar{\mathbf{C}} &= \begin{bmatrix} 0 & 0 & -0.19 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0225 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.0158 \end{bmatrix} \\ \bar{\mathbf{D}} &= \begin{bmatrix} 1.1 \\ 0.11 \\ 0.1 \end{bmatrix} \end{aligned}$$

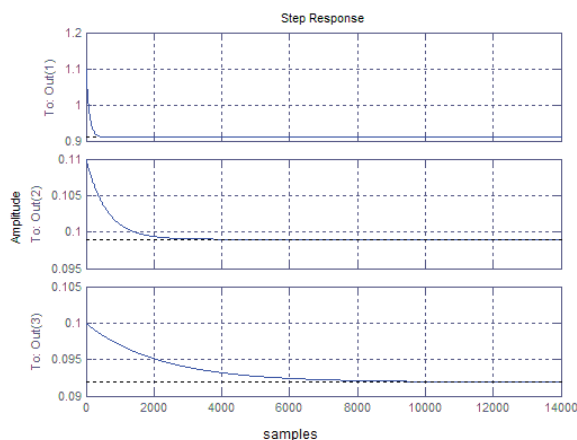
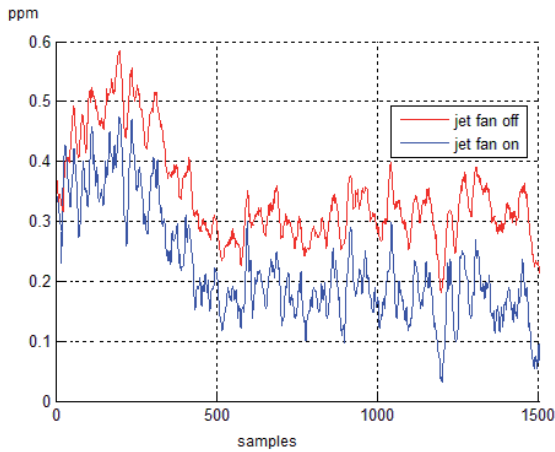


Fig.1 Step response of estimated model (single input, multi-output system)





**Fig. 2 Simulation of influence to CO concentration**

The red lines represent the concentrations of CO pollution without using the jet fan. The blue lines represent the concentrations of CO with jet fan switched on. In this paper we pointed out only to concentration of CO, because this type of pollution is most dangerous for human organism.

### 3. Validation assessment

Validation assessment is the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model. The goal of validation is to quantify confidence in the predictive capability of the model by comparison with experimental data [16].

The approach to validation assessment is to measure the agreement between model predictions and experimental data from appropriately designed and conducted experiments. Agreement is measured, for example, by quantifying the difference (error) between the experimental data and the model output. Uncertainty in both model output and experimental data will confound measurement of the error.

The definition of “validation assessment” given above requires further clarification. The phrase “process of determining” emphasizes that validation assessment is an on-going activity that concludes only when acceptable agreement between experiment and simulation is achieved. The phrase “degree to which” emphasizes that neither the simulation nor the experimental outcomes are known with certainty, and consequently, will be expressed as an uncertainty, e.g., as an expected value with associated confidence limits. Finally, the phrase “intended uses of the model” emphasizes that the validity of a model is defined over the domain of model form, inputs, parameters, and responses. This fact effectively limits use of the model to the particular application for which it was validated; use for any other purpose (than making a prediction) would require the validation assessment to be performed again.

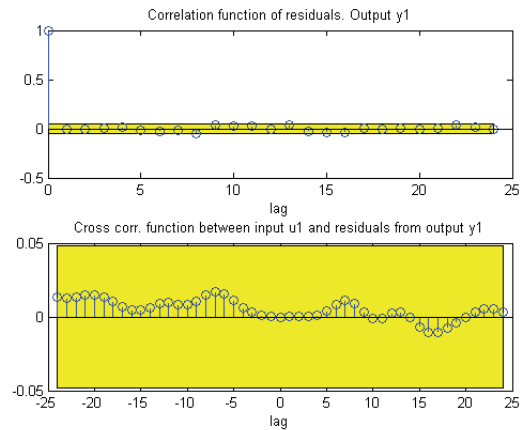
In this paper we are presented these methods of validation: Residual analysis, Akaike Final Prediction Error (FPE), Model and Parameter Accuracy

#### 3.1 Residual test

Residuals are differences between the one-step-predicted output from the model and the measured output from the validation data set. Thus, residuals represent the portion of the validation data not explained by the model. Residual analysis consists of two tests: the whiteness test and the independence test. According to the whiteness test criteria, a good model has the residual autocorrelation function in-

side the confidence interval of the corresponding estimates, indicating that the residuals are uncorrelated [2].

According to the independence test criteria, a good model has residuals uncorrelated with past inputs. Evidence of correlation indicates that the model does not describe how part of the output relates to the corresponding input. For example, a peak outside the confidence interval for lag  $k$  means that the output  $y(t)$  that originates from the input  $u(t-k)$  is not properly described by the model. In Fig. 3 we can see the residual autocorrelation for some model with 95% confidence interval limits for noncorrelated residuals and cross-correlation function between input and residuals from output.



**Fig.3 Residual test**

#### 3.2 The Akaike final prediction error (FPE)

The average prediction error is expected to decrease as the number of estimated parameters increase. One reason for this behavior is that the prediction errors are computed for the data set that was used for parameter estimation. It is now relevant to ask what prediction performance can be expected when the estimated parameters are applied to another data set. It might be suspected that a large overparametrized model might poorly predict the behavior of a new data set. In order to analyze this situation, we consider the expected prediction error based on the  $p$  parameter estimates  $\hat{\theta}$  based on  $N$  data fitted to some linear model

$$y_k = \phi_k^T \theta + w_k, \text{ where}$$

$$\begin{aligned} E\{\epsilon_k^2(\hat{\theta})\} &= E\{(y_k - \phi_k^T \hat{\theta})^2\} = \\ E\{(y_k - \phi_k^T \theta) - (\phi_k^T \hat{\theta} - \phi_k^T \theta)^2\} &= \\ E\{(y_k - \phi_k^T \theta)^2\} + E\{(\phi_k^T \hat{\theta} - \phi_k^T \theta)^2\} &= \\ E\{w_k^2\} + \text{tr}(E\{\tilde{\theta} \tilde{\theta}^T \phi_k \phi_k^T\}) &\approx \sigma^2 + \sigma^2 \frac{p}{N}, \end{aligned} \quad (24)$$

where the term  $\sigma$  derives from the noise variance properties, whereas the contribution  $p\sigma^2/N$  derives from the parameters error. The asymptotic prediction error decreases as the number of observations increases, whereas the prediction error variance increases as the number of estimated parameters increases. However, the expected loss function based on the null hypothesis  $H_0$  when estimating  $p$  parameters based on  $N$  observations is

$$E\{V_n(\hat{\theta}_N)\} = \frac{1}{2} \sigma^2 (N - p) \quad (25)$$

which tends to decrease as the number of parameter increases the final prediction error criterion (FPE) is estimated as

$$FPE(p) = \hat{\sigma}^2 \left( 1 + \frac{p}{N} \right) = \frac{N+p}{N-p} \frac{2}{N} V_N(\hat{\theta}_N) \quad (26)$$

The quality of identification measured as expected prediction accuracy can be improved by introducing new parameters to be estimated as long as each new parameter can be accurately estimated. For that reason we choose

$$\hat{p} = \arg \min FPE(p), \quad (27)$$

where, the second factor increases as the number  $p$  of estimated parameters increases [2].

Model 1 FPE 0.6875	Model 2 FPE 0.09887
-----------------------	------------------------

Fig. 4 FPE comparison

The FPE coefficient of two different models is shown in Fig. 4. This test shows the flexibility of the model structure. We are looking for minimum value of FPE coefficient.

### 3.3 Model and parameter accuracy

In this method we comprise the model performance and behavior with real data. A deterministic simulation can be used, where real data are compared with the model response to the recorded input signal used in the identification. This test should ascertain whether the model response is comparable to real data in magnitude and response delay [2].

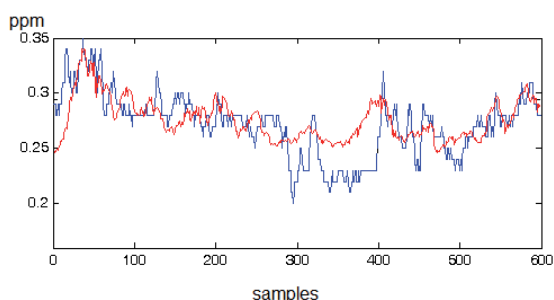


Fig. 5 Deterministic simulation – red line, real data – blue line

This method showed graphically accuracy between simulated values and measured values. Although the simulated and measured data not fit precisely, this result is sufficient for most stochastic system like pollution inside the road tunnel.

The blue values are measured data and the red values are simulated data in Fig. 5. Although the simulated values are not the same as measured data this result is sufficient for this system with most stochastic behavior. Another approach using the mathematical model is described in [5] and [6].

The next work will include finding out the other factors causing to concentrations of pollution inside the tunnel system. We can express the square of the correlation coefficient. This value represents the fraction of the variation in one variable that may be explained by the other factors. We want determinate how affect the atmospheric and velocity to pollution.

## Conclusion

The issue of multi-dimensional modeling system was explained. The different types of models can be converted to other type in order to use better solution. Description by transfer matrix makes calculations faster, but the state space model is more complex and can include the initial values for simulations. The paper presents a methodology that has been used for design parametric models of the jet fan. We need identification of system based on data obtained from the real ventilation system. Jet fan model has been created and verified in MATLAB environment. This part is the ground for best design of ventilation control system.

## Acknowledgment

This publication is the result of the project implementation: **Centre of excellence for systems and services of intelligent transport II.**, ITMS 26220120050 supported by the Research & Development Operational Programme funded by the ERDF.



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

## Literature

- [1] CAMACHO, E. F., BORDONS, C.: Model Predictive Control. 2nd ed., Springer-Verlag London Limited, 2004, pp. 405, ISBN 1-85233-694-3.
- [2] JOHANSON, R.: System modeling and identification, Prentice-Hall, 1993, pp. 512, ISBN 0-13-482308-7.
- [3] PIRNÍK, R., ČAPKA, M. and HALGAŠ, J.: Non-invasive monitoring of calm traffic. Proceedings of international symposium on advanced engineering & applied management – 40th anniversary in higher education, CD ver. S. II-107-112, ISBN 978-973-0-09340-7.
- [4] BUBNICKI, Z.: Modern Control Theory. Springer, 2005, pp. 422, ISBN 3-540-23951-0.
- [5] PAVELKA, M., PŘIBYL, P.: Simulation of Air Motion and pollutions inside the Road Tunnel– Mathematical Model, 2006, OPTUN 228/06-EEG.
- [6] HUANG, Z.Y., WU, K., XU, L.T. : Simulation analysis of longitudinal ventilation system with jet fan speed control for MPC strategy in a road tunnel, In. Proc. of the 15th International IEEE Conference on Intelligent Transportation Systems, Anchorage, Alaska, USA, September 16-19, 2012, pp. 1471-1476, ISBN: 978-1-4673-3064-0, ISSN 2153-0009.
- [7] GAVULOVÁ, A., PIRNÍK, R., HUDEC, R. Technical support of traffic control system of Slovak agglomerations in NaTIS project. In Modern transport telematics; selected papers, Berlin Heidelberg, Springer-Verlag, 2011, pp. 382-

391, ISBN 978-3-642-24659-3. Communications in computer and information science, vol. 239, ISSN 1865-0929.

[8] HOLEČKO, P., BUBENÍKOVÁ, E., PIRNÍK, R.: Communication systems in transport — hybrid ITS interface. In: Proc. of 9th international conference ELEKTRO 2012: Žilina - Rajecké Teplice, Slovakia, May 21st-22nd. IEEE, 2012, pp. 292-298, ISBN 978-1-4673-1178-6.

[9] NOSKIEVIČ, P.: System Modelling and Identification, Ostrava: MONTANEX, a. s., 1999, pp. 276, ISBN 80-7225-030-2.

[10] GODAN, J. at al.: Tunnels, Road Tunnels and Railway Tunnels, 2001, pp. 202, 135882/p-UK/Z.

[11] HRBČEK, J., SPALEK, V., ŠIMÁK, V.: Mathematical description of tunnel systems for the purpose of design the predictive algorithm, Acta Electrotechnica et Informatica, Vol. 10, No. 2, 2010, pp. 52–56, ISSN 1335-8243.

[12] HOLEČKO, P., KRBÍLOVÁ, I.: IT Security Aspects of Industrial Control Systems. Advances in Electrical and Electronic Engineering, No. 1-2 Vol. 5/2006, pp. 136-139, ISSN 1336- 1376.

[13] RÁSTOČNÝ, K., ŽDÁNSKY, J.: Specificities of safety PLC based implementation of the safety function. In: Proceedings of international conference applied electronics. AE 2012, Pilsen, pp. 229-232, ISBN 978-80-261-0038-6, ISSN 1803-7232.

[14] MACIEJOVSKI, J.M.: Predictive Control with constraints, Prentice Hall, 2002, pp. 331, ISBN: 0-201-39823-0.

[15] LIUPING, W.: Model Predictive Control System Design and Implementation Using MATLAB, Springer, 2009, pp. 371, ISBN 978-1-84882-330-3.

[16] THACKER, B.H. at al.: Concepts of Model Verification and Validation. 2004, Technical Report, Report Number LA-14167-MS.

[17] YINGHUA HE, HONG WANG, and BO ZHANG.: Color-Based Road Detection in Urban Traffic Scenes, IEEE Transactions on intelligent transportation systems, vol. 5, no. 4, December 2004, ISSN 1524-9050.

**Ing. Jozef Hrbček, PhD.**

University of Žilina  
Faculty of Electrical Engineering  
Department of Control and Information Systems  
Univerzitná 1  
01026 Žilina, Slovak Republic  
jozef.hrbcek@fel.uniza.sk

**Ing. Vojtech Šimák, PhD.**

University of Žilina  
Faculty of Electrical Engineering  
Department of Control and Information Systems  
Univerzitná 1  
01026 Žilina, Slovak Republic  
vojtech.simak@fel.uniza.sk

**Ing. Ľubomír Pekár**

University of Žilina  
Faculty of Electrical Engineering  
Department of Control and Information Systems  
Univerzitná 1  
01026 Žilina, Slovak Republic  
lubomir.pekar@fel.uniza.sk

# Lineárne riadenie prirodzene nestabilného systému s použitím PID regulátora

Vojtech Šimák, Dušan Nemeč, Jozef Hrbček, Emília Bubeníková

## Abstrakt

Príspevok opisuje parametre a charakteristiky 4-rotorového vrtuľníka. Prirodzene nestabilný charakter systému môže byť účinne potlačený dvojstupňovým PID regulátorom. Ladenie konštánt PID je náročné vzhľadom na zmeny aerodynamiky v blízkosti terénu. Sensorický systém vrtuľníka sa skladá zo štyroch 3-osových MEMS akcelerometrov a dvoch 3-osových MEMS gyroskopov kvôli zvýšeniu presnosti.

**Kľúčové slová:** vrtuľník, PID, regulátor, prenosová funkcia

## Úvod

Tento článok sa zaoberá riadením 4-rotorového vrtuľníka pomocou PID regulátora. Riadenie vrtuľníka aj spracovanie údajov zo snímačov je realizované na 8-bitovom mikropočítači ATMEGA 1284P s taktovacou frekvenciou 20 MHz. Sensorický systém je založený na akcelerometroch a gyroskopoch vyrobených technológiou MEMS. Snímače boli zdvojené kvôli zvýšeniu presnosti merania a potlačeniu diferenciálnych vibrácií. Všetky snímače sú pripojené cez zbernicu I2C s riadiacim mikropočítačom, pričom údaje musia byť vyhodnocované v reálnom čase (so vzorkovacou frekvenciou minimálne 200Hz). Regulátor na stabilizáciu letu má 2 úrovne. Nižšia úroveň reguluje uhlovú rýchlosť otáčania okolo každej zo súradnicových osí ( $x$ ,  $y$ ,  $z$ ). Druhá, vyššia úroveň riadenia riadi náklon vrtuľníka vyjadrený pomocou Eulerových uhlov. Výstup vyššej úrovne riadenia (riadenie uhla) je požadovanou hodnotou pre nižšiu úroveň (regulácia uhlovej rýchlosti). Pre účely ladenia bol využitý jednoosový otočný záves. Toto zariadenie umožňuje testovať hodnoty P, I a D zosilnení, anti-windup algoritmus a nastavenia filtrov.

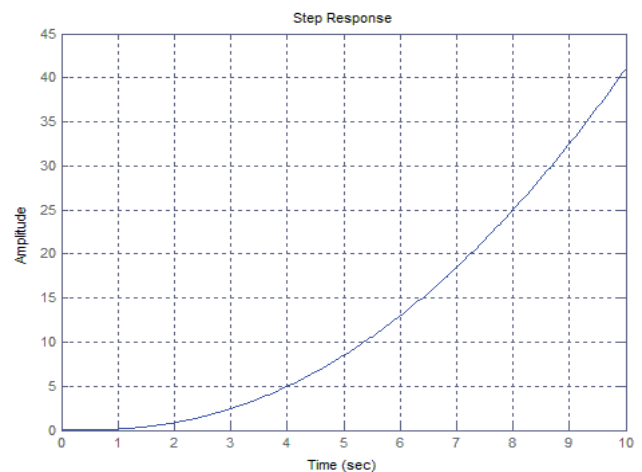
Vďaka symetrickej konštrukcii vrtuľníka sú parametre stabilizácie priečneho a pozdĺžneho náklonu (osi  $x$  a  $y$ ) rovnaké, pričom ich približné hodnoty je možné získať manuálnym ladením, ak je vrtuľník upevnený v otočnom závесе. Regulácia kurzu (os  $z$ ) môže byť naladená pri skúšobnom testovaní letu, pretože jej nastavenie nie je kritické pre stabilitu letu.

## 1. Prirodzene nestabilný lineárny systém

Uvažujme mechanickú sústavu, ktorej výstupnou veličinou je posunutie resp. otočenie, pričom riadiace zásahy aplikované na sústavu ovplyvňujú inú mechanickú veličinu. Pohyb (posun, rotácia) mechanického systému je popísaný niekoľkými premennými (napr. poloha, rýchlosť, zrýchlenie). V prípade, že akčná veličina priamo ovplyvňuje druhú deriváciu požadovanej hodnoty, prenosová funkcia  $F(s)$  je druhého rádu a obsahuje iba dva integračné členy:

$$F(s) \approx \frac{1}{s^2}, \quad (1)$$

kde  $s$  je komplexná premenná, ak  $s = j\omega$ . Prechodová charakteristika tejto skupiny systémov je znázornená na obr.1.



Obr.1 Typická prechodová charakteristika prirodzene nestabilného lineárneho systému

Fig.1 Typical step response of naturally unstable linear system

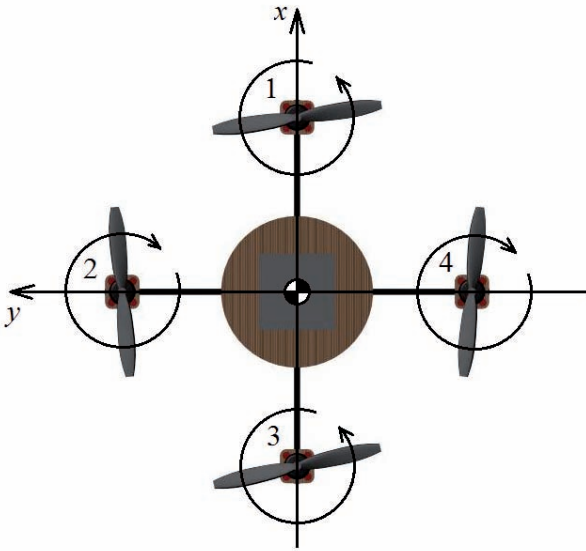
Takýto systém (bez spätnej väzby) je prirodzene nestabilný, pretože jeho prechodová charakteristika  $h(t)$  je divergentná:

$$h(t) \sim L^{-1} \left\{ \frac{1}{s} \cdot \frac{1}{s^2} \right\} = \frac{t^2}{2}. \quad (2)$$

Vzhľadom k tejto vlastnosti je systém veľmi citlivý na rušenie, ktoré musí byť potlačené presným ale zároveň rýchlym riadením.

## 2. Model 4-rotorového vrtuľníka

Mechanický systém 4-rotorového vrtuľníka môžeme opísať ako teleso so šiestimi stupňami voľnosti (v angl. *Degrees of Freedom*, skr. DoF). Motory s rotormi sú umiestnené vo vrcholoch štvorca, pozri obr.2.



**Obr.2 Umiestnenie rotorov**  
**Fig.2 The placement of rotors**

Súradnicová sústava je pravouhlá, s bázou  $(e_x, e_y, e_z)$  a je pevne spojená s vrtuľníkom.

Rotor pôsobí na sústavu dvojakým spôsobom:

- $F$  – užitočný ťah - sila pôsobiaca nahor pozdĺž osi otáčania (u vrtuľníka sú rotory orientované v zvislom smere – rovnobežne s osou  $z$ ).
- $M_R$  – reakčný moment sily pôsobiaci v opačnom smere ako uhlová rýchlosť rotora  $\omega_R$  (vektor uhlovej rýchlosti aj reakčného momentu sú rovnobežné s osou otáčania rotora).

Rotor predstavuje aerodynamický prvok v podzvukovej oblasti rýchlosti, preto sú všetky sily úmerné druhej mocnine otáčok (Revolutions per Minute, RPM).

Kvôli zjednodušeniu neskoršie použitých vzorcov, budeme vyjadrovať RPM ako rotačný pomer  $n$ , ktorý je v rozmedzí 0 až 100% maximálnych otáčok. Potom platí:

$$\omega_R = 2\pi f_{\max} n, \quad (3)$$

$$F \approx F_{\max} n^2, \quad (4)$$

$$M_R \approx -M_{\max} n^2, \quad (5)$$

kde  $f_{\max}$  je maximálna frekvencia otáčania,  $F_{\max}$  je maximálny užitočný ťah a  $M_{\max}$  je maximálny reakčný moment. Vzhľadom k tomu, že sa rotor otáča okolo pevnej osi, je jeho moment zotrvačnosti  $J_R$  konštantný. Táto konštanta zahŕňa aj zotrvačnosť rotačných častí motora. Každý rotor predstavuje rotujúce teleso s relatívne malým momentom hybnosti  $L_R$ , ktorý vypočítame ako:

$$L_R = J_R \omega_R = L_{\max} n e_z, \quad (6)$$

kde  $L_{\max}$  je moment hybnosti pri maximálnych otáčkach. Otáčaním vrtuľníka sú generované gyroscopické momenty, ktoré majú vplyv na systém a spôsobujú značné napätie v nosnom ráme. Krútiaci moment  $M_{\text{gyro}}$  jedného z motorov môže byť približne vyjadrený nasledujúcim vzorcom: (platný pre  $\omega \ll \omega_R$ ):

$$M_{\text{gyro}} \approx -\sum_{i=1}^4 \omega \times J_{Ri} \omega_{Ri} \approx -L_{\max} (n_1 - n_2 + n_3 - n_4) (\omega \times e_z) \quad (7)$$

Z výrazu (7) je jasné, že pre rovnaké otáčky na všetkých rotoroch ( $n_1 = n_2 = n_3 = n_4$ ) sú gyroscopické momenty úplne vykompenzované.

Riadením otáčok jednotlivých motorov potom vytvárame riadiace momenty v jednotlivých osiach. Rozdielny ťah rotorov spôsobuje moment sily v osiach  $x$  a  $y$ . Na vytvorenie riadiaceho momentu v osi  $z$  sa využívajú reakčné momenty jednotlivých rotorov, pretože protiahlé rotory majú rovnaký zmysel otáčania, a teda aj rovnaký smer reakčného momentu. Na základe rovnice (5) môžeme riadiace momenty vyjadriť ako:

$$M_x = (F_2 - F_4) r_M = F_{\max} (n_2^2 - n_4^2) r_M, \quad (8)$$

$$M_y = (F_3 - F_1) r_M = F_{\max} (n_3^2 - n_1^2) r_M, \quad (9)$$

$$M_z = M_{R1} + M_{R3} - M_{R2} - M_{R4} = -M_{\max} (n_1^2 - n_2^2 + n_3^2 - n_4^2), \quad (10)$$

kde  $r_M$  je vzdialenosť ťažiska vrtuľníka od osi otáčania ľubovoľného rotora.

Z dôvodu linearizácie aktuátorov boli medzi vstup motorov a výstup z regulačného systému zaradené korekčné členy zohľadňujúce kvadratickú závislosť riadiacich momentov od otáčok. Matematicky sa dajú korekčné členy opísať nasledovne:

$$n_1 = \sqrt{u_0 - u_y - u_z}, \quad (11)$$

$$n_2 = \sqrt{u_0 + u_x + u_z}, \quad (12)$$

$$n_3 = \sqrt{u_0 + u_y - u_z}, \quad (13)$$

$$n_4 = \sqrt{u_0 - u_x + u_z}, \quad (14)$$

kde  $u_i$  predstavuje zložku akčného zásahu v  $i$ -tej osi.

Na základe pohybovej rovnice pre rotačný pohyb tuhého telesa môžeme napísať:

$$\frac{d}{dt} [J\omega] = M_{\text{act}} + M_{\text{gyro}}, \quad (15)$$

kde  $\omega$  je vektor okamžitej uhlovej rýchlosti vrtuľníka,  $J$  je diagonálny tenzor momentu zotrvačnosti (kvôli symetrickej konštrukcii sú nediagonálne prvky nulové),  $M_{\text{act}}$  je moment vyvolaný z riadiaceho systému,  $M_{\text{gyro}}$  je gyroscopický moment generovaný diferenciálnou rotáciou rotorov.

Pre jednotlivé zložky s použitím predchádzajúcich vzorcov dostaneme:

$$J_x \frac{d\omega_x}{dt} + (J_z - J_y) \omega_z \omega_y = F_{\max} (n_2^2 - n_4^2) r_M - L_{\max} (n_1 - n_2 + n_3 - n_4) \omega_y, \quad (16)$$

$$J_y \frac{d\omega_y}{dt} + (J_x - J_z) \omega_x \omega_z = F_{\max} (n_3^2 - n_1^2) r_M + L_{\max} (n_1 - n_2 + n_3 - n_4) \omega_x, \quad (17)$$

$$J_z \frac{d\omega_z}{dt} + (J_y - J_x) \omega_y \omega_x = -M_{\max} (n_1^2 - n_2^2 + n_3^2 - n_4^2). \quad (18)$$

Tento systém troch diferenciálnych rovníc opisuje závislosť medzi uhlovou rýchlosťou  $\omega$  vrtuľníka nameranou v sústave pevne spojenej s vrtuľníkom a otáčkami každého rotora  $n_i$ . Ľavá strana týchto rovníc vyjadruje zotrvačnosť telesa so šiestimi stupňami voľnosti, pravé strany rovníc sú špecifické pre systém aktuátorov 4-rotorového vrtuľníka. Je prakticky

nemožné nájsť analytické riešenie rovníc (16) - (18), ale táto sústava rovníc má význam pri numerickom modelovaní správania sa vrtníka pomocou výpočtovej techniky.

Uhlová rýchlosť  $\omega$  sa odzrkadľuje v zmene Eulerových uhlov  $\alpha, \beta, \gamma$ , t.j. priečného náklonu (angl. *Roll*), pozdĺžneho náklonu (angl. *Pitch*) resp. kurzu (angl. *Yaw*) v konvencii *xyz*. Transformácia medzi uhlovou rýchlosťou a časovými deriváciami Eulerových uhlov je nasledovná:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} 1 & \frac{\sin \alpha \sin \beta}{\cos \beta} & \frac{\cos \alpha \sin \beta}{\cos \beta} \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \frac{\sin \alpha}{\cos \beta} & \frac{\cos \alpha}{\cos \beta} \end{bmatrix} \cdot \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}. \quad (19)$$

Výsledné Eulerove uhly možno vypočítať pomocou časovej integrácie derivácií  $\dot{\alpha}, \dot{\beta}, \dot{\gamma}$ .

### 3. Analýza stabilizačného algoritmu 4-rotorového vrtníka

Pre analýzu charakteru systému môžeme aproximovať rovnice (16) – (18) za použitia predpokladu:

$$\alpha = 0, \beta = 0, J_R = 0, \frac{d\omega_i}{dt} \gg \omega_j \omega_k. \quad (20)$$

Potom získame:

$$J_x \frac{d^2 \alpha}{dt^2} \approx F_{\max} (n_2^2 - n_4^2) r_M = C_x u_x, \quad (21)$$

$$J_y \frac{d^2 \beta}{dt^2} \approx F_{\max} (n_3^2 - n_1^2) r_M = C_y u_y, \quad (22)$$

$$J_z \frac{d^2 \gamma}{dt^2} \approx -M_{\max} (n_1^2 - n_2^2 + n_3^2 - n_4^2) = C_z u_z, \quad (23)$$

kde  $u_i$  sú vstupy z linearizačného člena podľa (11) - (14) a  $C_i$  sú konštanty závislé od akčného člena.

V princípe ide o riadený systém druhého rádu s prenosovou funkciou  $F_{Si}(s)$  a má integračný charakter. Zjednodušená prenosová funkcia má dvojnásobný nulový pól. Zanedbali sme vplyv gyroskopického a deviačného momentu, ktoré môžu byť interpretované ako rušenie. V každej osi môžeme vyjadriť vlastnosti systému pomocou Laplaceovej transformácie a prenosových funkcií:

$$F_{Si}(s) \approx C_i \frac{1}{J_i s^2}. \quad (24)$$

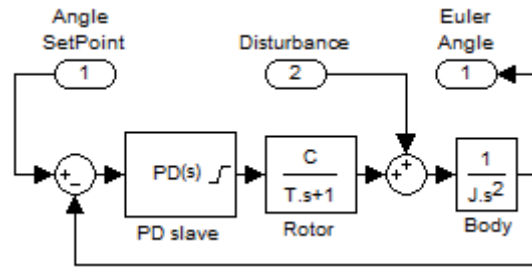
Ak zahrnieme zotrvačnosť aktuátora do tohto zjednodušeného modelu, dostaneme:

$$F_{Si}(s) \approx \frac{C_i}{(Ts+1)} \cdot \frac{1}{J_i s^2} = F_{rotor}(s) \cdot F_{body}(s). \quad (25)$$

### 4. Jednoduchý regulátor

Ak by sme chceli regulovať systém jedným lineárnym regulátorom, môže byť vhodný PD regulátor definovaný ako:

$$F_{PD}(s) \approx K_d + sK_p. \quad (26)$$



Obr.3 Jednoduchá regulácia uhla

Fig.3 Simple angle regulator

Prenosová funkcia systému pre každú os (s uvažovaním nulového rušenia) je:

$$F(s) = \frac{F_{PD}(s)F_{rotor}(s)F_{body}(s)}{1 + F_{PD}(s)F_{rotor}(s)F_{body}(s)} = \frac{K_d Cs + K_p C}{JT s^3 + Js^2 + K_d Cs + K_p C}. \quad (27)$$

Prenosová funkcia rušenia  $F^*(s)$  s nulovým vstupom je:

$$F^*(s) = \frac{F_{body}(s)}{1 + F_{PD}(s)F_{rotor}(s)F_{body}(s)} = \frac{Ts + 1}{JT s^3 + Js^2 + K_d Cs + K_p C}. \quad (28)$$

Na základe algebraických kritérií stability môžeme povedať, že systém bude stabilný, ak platí:

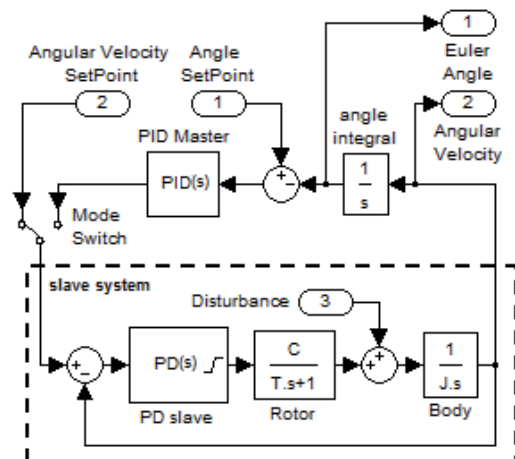
$$K_d > K_p T; \quad K_p > 0. \quad (29)$$

Systém potláča skokové rušenie na hodnotu:

$$h^*(\infty) = \lim_{s \rightarrow 0} F^*(s) = \frac{1}{K_p C}. \quad (30)$$

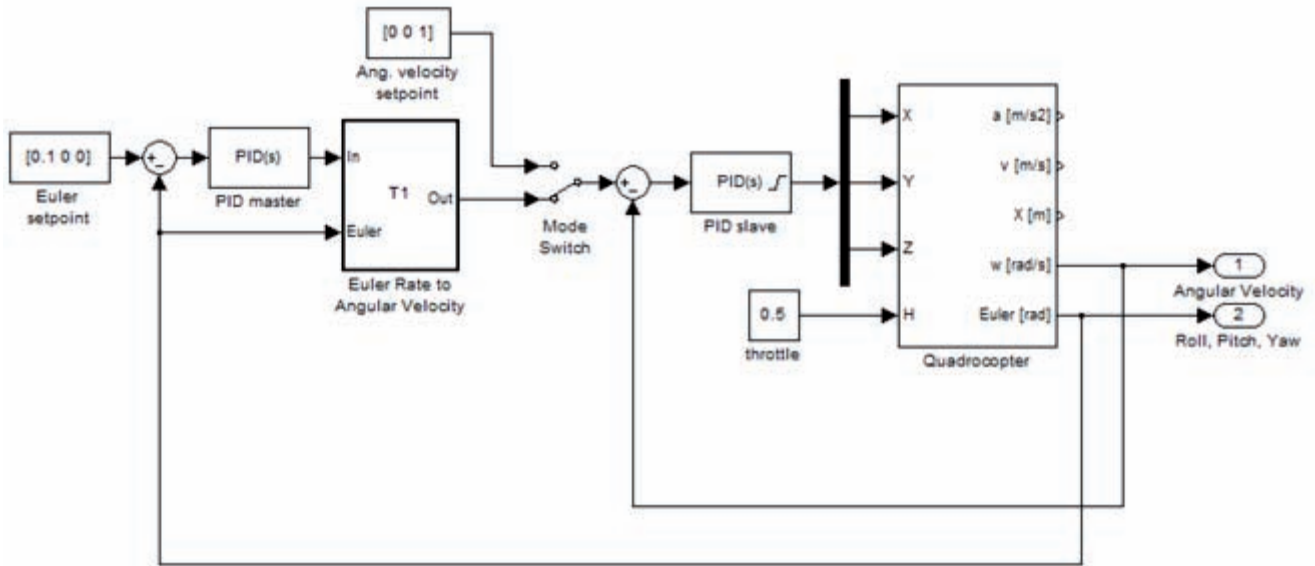
### 5. Dvojúrovňová regulácia

Po rozdelení dvojitého integračného člena na dva jednoduché môžeme vytvoriť dvojstupňový regulátor. Môžeme využiť tú výhodu, že pomocou gyroskopov nemeríme priamo Eulerove uhly ale uhlovú rýchlosť  $\omega$ . PD regulátor na nižšej úrovni bude regulovať uhlovú rýchlosť a PID regulátor na vyššej úrovni bude regulovať uhol (viď obr. 4).



Obr.4 Schéma dvojúrovňovej regulácie

Fig.4 Two-stage regulation schematics



Obr.5 Model regulácie v Simulinku  
Fig.5 Simulink model of regulation

Znížením stupňa integrácie vo vnútornej slučke, môžeme znížiť dobu odozvy (polovica fázového oneskorenia). Prenosová funkcia podriadenej sústavy  $F_1(s)$  potom bude:

$$F_1(s) = \frac{F_{PDslave}(s)F_{rotor}(s)F_{body1}(s)}{1 + F_{PDslave}(s)F_{rotor}(s)F_{body1}(s)} = \frac{K_{d1}Cs + K_{p1}C}{JT_s^2 + (J + K_{d1}C)s + K_{p1}C} \quad (31)$$

Prenosová funkcia rušenia podriadeneho (slave) systému bude:

$$F_1^*(s) = \frac{F_{body1}(s)}{1 + F_{slave}(s)F_{rotor}(s)F_{body1}(s)} = \frac{Ts + 1}{JT_s^2 + (J + K_{d1}C)s + K_{p1}C} \quad (32)$$

Systém je stabilný pre kladné konštanty PD regulátora  $K_{p1}$ ,  $K_{d1}$ . Použitie derivačného člena má jednu významnú nevýhodu a tou je zosilnenie vysokofrekvenčného šumu. Preto je použitie derivačného člena zvyčajne spojené s dolnopriepustným filtrom s časovou konštantou  $T_f$ . Časová konštantka filtra je nižšia ako požadovaná doba regulácie a vyššia ako doba odozvy aktuátorov. V realizovanom prototypu bola hodnota  $T_f = 500$  ms. Upravená prenosová funkcia PD regulátora potom je:

$$F_{PDslave}(s) = K_{p1} + \frac{K_{d1}s}{(T_{f1}s + 1)} \quad (33)$$

Nadradený systém (master) má výslednú prenosovú funkciu  $F_2(s)$ :

$$F_2(s) = \frac{F_{PIDmaster}(s)F_1(s)F_{body2}(s)}{1 + F_{PIDmaster}(s)F_1(s)F_{body2}(s)} \quad (34)$$

Výsledná prenosová funkcia rušenia je:

$$F_2^*(s) = \frac{F_1^*(s)F_{body2}(s)}{1 + F_{PIDmaster}(s)F_1(s)F_{body2}(s)} \quad (35)$$

Reakcia systému na jednotkové rušenie sa ustáli na hodnote:

$$h^*(\infty) = \lim_{s \rightarrow 0} F_2^*(s) = \lim_{s \rightarrow 0} \frac{F_1^*(s)}{F_{master}(s)F_1(s)} = \frac{1}{K_{p1}C} \lim_{s \rightarrow 0} \frac{1}{F_{master}(s)} = \frac{1}{K_{p1}C} \lim_{s \rightarrow 0} \left( K_{p2} + \frac{K_i}{s} + \frac{K_{d2}s}{T_{f2}s + 1} \right) \quad (36)$$

Je jasné, že ak nadradený regulátor obsahuje integračnú zložku so zosilnením  $K_i$ , môže byť rušenie úplne potlačené. To je významná výhoda oproti jednoduchým regulátorom. Pretože sme uvažovali gyroskopický a deviačný moment ako rušenie, pre realizáciu bol zvolený dvojúrovňový regulátor. Obrázok 5 ukazuje kompletnú regulačnú schému, ktorá v sebe zahŕňa aj matematický korekčný člen medzi nadradeným a podradeným regulátorom, t.j. medzi PID a PD regulátorom. Tento je nevyhnutný pre zabezpečenie riadneho fungovania pre všetky náklony. Jeho funkcia je vyjadrená nasledovne:

$$SlaveIN = \begin{bmatrix} 1 & 0 & -\sin \beta \\ 0 & \cos \alpha & \sin \alpha \cos \beta \\ 0 & -\sin \alpha & \cos \alpha \cos \beta \end{bmatrix} \cdot MasterOUT, \quad (37)$$

kde  $\alpha$  je priečny náklon a  $\beta$  je pozdĺžny náklon. Výraz (37) je inverzný k (19).

## 6. Namerané parametre

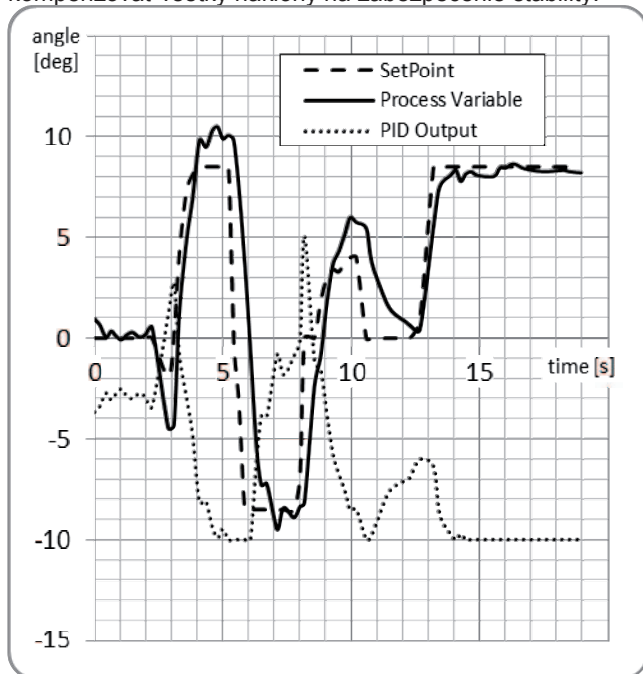
Aby bolo možné stanoviť parametre uvedeného matematického modelu 4-rotorového vrtuľníka, bolo vykonaných niekoľko meraní. Boli pri nich získané nasledovné hodnoty:

- Moment zotrvačnosti v osiach  $x$  a  $y$ :  $J_x = J_y = 0,4 \text{ kg.m}^2$ .
- Moment zotrvačnosti v osi  $z$ :  $J_z = 0,42 \text{ kg.m}^2$ .
- Vzdialenosť medzi rotorom a ťažiskom:  $r_M = 0,23 \text{ m}$ .
- Maximálny ťah jedného rotora:  $F_{max} = 12 \text{ N}$ .

Konštanty PID regulátora pre osi  $x$  a  $y$  pri ladení boli nasledovné:

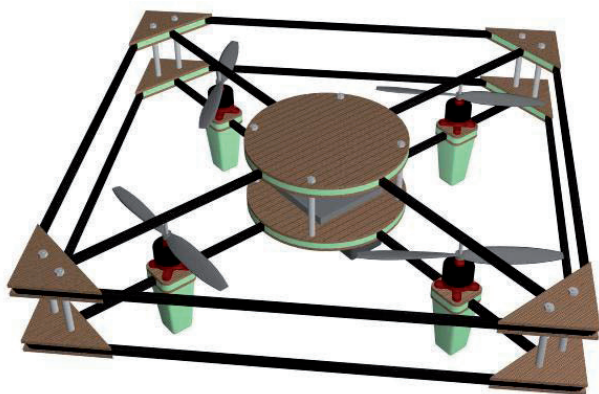
- Master PID:  $K_p = 4,00$ ;  $K_i = 1,00$ ;  $K_d = 0,018$ ;  $T_f = 500$  ms.
- Slave PD:  $K_p = 0,080$ ;  $K_d = 0,020$ ;  $T_f = 300$  ms.

Obrázok 6 ukazuje trendové krivky požadovanej hodnoty (*SetPoint*), aktuálnej hodnoty (*Process Variable*) a výstupu PID regulátora (*PID Output*), ktoré boli získané pri ladení PID regulátora v reálnom čase. Ako je možné vidieť, systém bol vystavený relatívne veľkému rušeniu a tak PID musí kompenzovať všetky náklady na zabezpečenie stability.



Obr.6 Dáta z reálneho procesu regulácie (nadradený PID)

Fig.6 Real-time process regulation data (Master PID)



Obr.7 Model vrtuľníka

Fig.7 Design of the helicopter

## Záver

Tento systém je ťažké naladiť v priebehu letu, preto musí byť ladený v zavesení. Správanie reálneho vrtuľníka upevneného v zavesení nezodpovedá na 100% uvedenému modelu systému. Dôvodom rozdielov môžu byť zložité aerodynamické javy v blízkosti terénu a vplyv samotného zavesenia. Prechod medzi prízemným letom a väčšou výškou je prakticky nemožné matematicky popísať. Druhým významným neznámym parametrom je ťahová charakteristika jedného

rotora. Zníženie časového oneskorenia rotora môže výrazne zvýšiť účinok riadenia. Charakter systému neumožňuje použiť „štandardné“ algoritmy ladenia, ktoré sú založené na stabilnej odozve na skokovú zmenu pri otvorenej regulačnej slučke.

Po naladení regulátorov bolo zistené nasledovné:

- riadiaci systém musí výrazne reagovať na rušenie,
- je nutné zlepšovať dynamickú odozvu aktuátorov (t.j. rotorov vrtuľníka),
- pomocou dvojestupňového regulátora je možné dosiahnuť skokovú zmenu uhla bez výrazného integračného prekmitu.

## Podakovanie

Tento príspevok vznikol za podpory Kultúrnej a edukačnej grantovej agentúry (KEGA) Číslo: 010ŽU-4/2013: Modernizácia didaktického vybavenia a metód vzdelávania so zameraním na oblasť robotiky.

## Abstract

This paper describes parameters and characteristics of 4-rotor helicopter. Unstable character of the system can be suppressed by two-stage PID regulator. The tuning is very complex due to aerodynamic changes near floor. The sensor system consists of four 3-axis MEMS accelerometers and two 3-axis gyroscopes due to increasing accuracy.

## Literatúra

- [1] COMINOS, P.: Control Theory and Applications, IEEE Proceedings 2003, ISSN 1350-2379.
- [2] ROZHDESTVENSKY, V.: Aerodynamics of a Lifting System in Extreme Ground Effect, Springer 2000, ISBN 3-540-66277-4.
- [3] KVASNICA, J. a kol.: Mechanika, Academia Praha 2004, ISBN 80-200-1268-0.
- [4] ŽDÁNSKY, J., NAGY, P.: Influence of the control system structure with safety PLC on its reliability and safety. In: Proceedings of the 9th international conference ELEKTRO 2012, Rajecské Teplice, pp. 395 - 399, ISBN 978-1-4673-1178-6.
- [5] HOLEČKO, P., KRBILOVÁ, I.: IT Security Aspects of Industrial Control Systems. Advances in Electrical and Electronic Engineering, No. 1-2 Vol. 5/2006, pp. 136-139, ISSN 1336-1376.
- [6] RÁSTOČNÝ, K., ŽDÁNSKY, J.: Specificities of safety PLC based implementation of the safety function. In: Proceedings of international conference applied electronics. AE 2012, Pilsen, pp. 229-232, ISBN 978-80-261-0038-6, ISSN 1803-7232.

Ing. Vojtech Šimák, PhD.

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 1  
01026 Žilina  
Tel.: +421 41 513 3304



**Bc. Dušan Nemeč**

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 1  
01026 Žilina

**Ing. Jozef Hrbček, PhD.**

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 1  
01026 Žilina

**Ing. Emília Bubeníková**

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 1  
01026 Žilina

# Virtualization for manufacturing execution systems integrated with distributed control systems

Peter Peniak, Mária Franeková

## Abstract

This article deals with virtualization and its possible application for manufacturing execution systems integrated with distributed control systems. The main focus is paid on virtualization of infrastructure that could provide opportunities for MES systems and enable significant improvements in system availability, fail-over and roll-back features for a manufacturing area.

**Keywords:** manufacturing execution systems, distributed control systems, virtualization, hypervisor, virtual desktops, virtual LAN, hardware partitioning, software PLCs, remote desktop protocol, terminal services

## Introduction

Information systems of nowadays production companies are based on ERP (Enterprise Resource Planning) systems, which are responsible for corporate resources and processes, but as well as on MES (Manufacturing Execution Systems) that are generally used to manage and monitor work-in-process on ShopFloor.

MES system keeps track of all manufacturing information, for instance the production orders, recipes, product and process monitoring and overall product traceability. The main goal is to support the execution in real time and an improvement of overall productivity. MES are typically integrated with various MCS (Machine Control Systems) within DCS, as shown in Fig. 1.

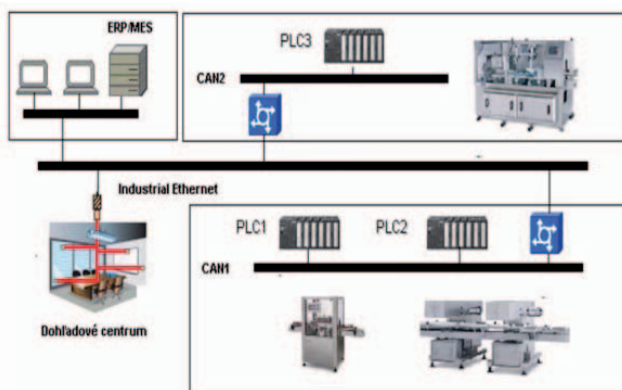


Fig.1 ERP/MES system integration with DCS

Manufacturing Execution Systems are typical by the need for long lifecycles. As soon as IT (Information technology) solution is in production, the companies want to ensure production stability and avoid unnecessary changes in software applications or basic IT infrastructure. The risk and cost of service interruptions become much higher due to their deep integration within production process. Therefore, an availability and performance of mission-critical manufacturing application is vital for advanced production companies with the following requirements:

- High system availability (~99%, 24h/7d).
- Fail-safe setup (system redundancy, fast recovery).

- Prolonged lifecycle (~10 years together with MCS).
- Restricted maintenance windows (e.g. holidays).
- Limited changes (restricted patching, releases).
- Necessity to support various machine control systems with heterogenic hardware, software and interfaces.

To address these challenges, the virtualization is one of the most promising technologies that can improve and secure MES IT infrastructure and provide needed features relevant to this mission-critical area.

## 1. Virtualization

Virtualization can be defined as a creation of a virtual version of the physical object with its all needed features, independently on its physical implementation. In case of IT environment, virtualization decouples software from hardware, enabling run multiple virtual objects (servers, desktops, applications) on a single physical system, and share the resources of that object across multiple environments resulting with

- a) server and storage virtualization,
- b) desktop virtualization,
- c) application virtualization,
- d) network virtualization (VLAN).

### 1.1 Server and storage virtualization

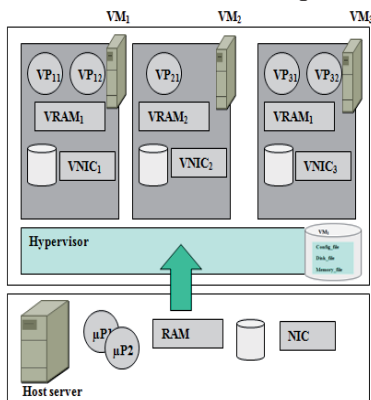
Virtualization of servers was the first successful approach. The consolidation of physical servers and their replacement by virtual servers simplified the hardware operations, maintenance and decreased an overall cost, bringing additional potential improvement in availability and fast service recovery.

The server virtualization is based on so called VM (Virtual Machine) that represents logical server object. This simulated server hardware object behaves exactly like a physical computer or storage. It has its own "virtual" CPU, RAM, NIC (Network Interface Card) and hard disk and runs as an

isolated operating system installation on shared hardware, called a host. The process of virtualization works by inserting a thin layer of virtualization kernel software, also called as “hypervisor”, directly into the computer hardware. The hypervisor layer presents multiple sets of “virtual hardware,” which contain the same components as a regular machine (e.g., motherboard, chipset, etc.), such as:

- virtual processors (VP<sub>i</sub>),
- virtual memory (VRAM<sub>i</sub>),
- virtual file systems (VFS<sub>i</sub>),
- and virtual network interface cards (VNIC<sub>i</sub>),

which are mapped to physical hardware resources of host servers. The model of VM is shown in Fig. 2.



**Fig.2 Server/storage virtualization**

The hardware sizing for needed VM resources must be in a correlation with available physical resources of the host computers. Therefore correct configuration of host servers is a prerequisite for implementation of server virtualization. The simplified model for resource calculation is provided in the following formulas:

$$N_{hosts} \geq 3, \quad (1)$$

$$S_{RAM\_Hosts} = \sum_{i=1}^n VRAM_i + \Delta RAM \times N_{hosts}, \quad (2)$$

$$N_{\mu P\_Hosts} = \sum_{i=1}^n (\sum_{j=1}^{m_i} VP_{ij}) / N_{Cores}, \quad (3)$$

$$N_{VNIC} \leq 4 \times n \geq 1, \quad (4)$$

$$S_{FS\_Hosts} = \sum_{i=1}^n (\sum_{j=1}^{m_i} VMFS_{ij}), \quad (5)$$

where

- $N_{Hosts}$  – is number of physical servers,
- $N_{\mu P\_hosts}$  – is number of processors in host computers (sockets),
- $N_{VNIC}$  – is number of virtual network interface cards,
- $S_{FS\_hosts}$  – is size of file systems connected to hosts (GB),
- $S_{RAM\_hosts}$  – is RAM size assigned in Hosts (GB).

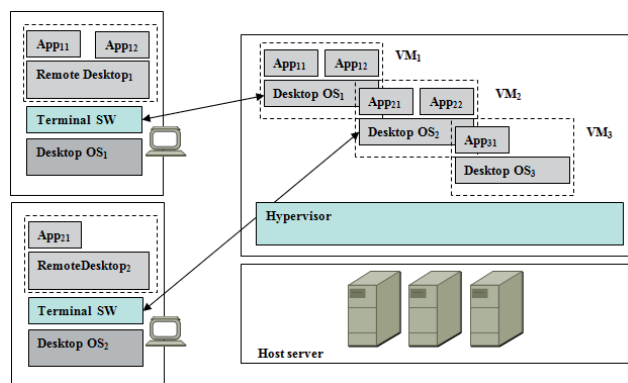
## 1.2 Desktop virtualization

Almost the same model can be applied for desktop visualization. Virtual machines represent virtual desktop for users. In contrast to server virtualization, there is a difference in access layer. The users have to connect to the host computer via a specific virtualization client or through a presentation layer, such as:

- Citrix-ICA,
- Microsoft-Remote Desktop Protocol (RDP),
- PC over IP.

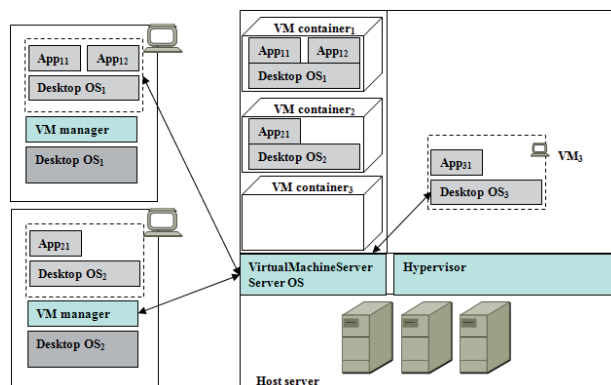
This is quite similar to terminal services, which use the thin client or thick client to access target workplace. The main difference is that users do not share one server computer via Terminal server, but are directly connected to defined virtual machines (one-to-one). The management platform must deal with requirements to maintain different types of virtual desktops, user profiles and data management. Overall solution is then called as VDM (Virtual Desktop Management).

Example of simple VDM scenario is illustrated by Fig. 3. Host servers provide VM resources for Hypervisor, which assigns a dedicated VM<sub>i</sub> for each requested virtual desktop. Virtual desktop runs as dedicated desktop computer with own desktop operation system (Desktop OS) and has installed the requested applications (App<sub>ij</sub>). Remote workstation (PC, thin-client, workstation) can access the virtual desktop via terminal software with standard remote desktop protocol, for example RDP.



**Fig.3 Desktop virtualization -VDM**

As explained, this solution is very similar to the standard terminal services, having several desktops with separated operation system instead of sharing one computer. This approach eliminates performance problems and compatibility issues when different application can cause the conflict within single operation system. However, there is a still one disadvantage, for critical real-time application or very sensitive graphical application can have significant issues due to performance of remote desktop protocols.



**Fig.4 Desktop virtualization-Virtual containers**

For such case, the different model of virtualization is invented, based on virtual container system. All virtual desktops are centrally installed and stored in virtual containers, that can be transferred to any hypervisor system or virtualization software (VM manager), that runs on the top of local operation system. The streaming protocols might be used or direct copy to the target system from USB drive/DVD is possible as well. The main advantage is that VM is started on local computer (desktop OS<sub>1,2</sub>) and there is no need to use remote desktop protocol and compromise performance requirements. The use of virtual container keeps integrity of the image and enables a high security and software deployment. Those desktop that do not require critical re-

response time can still run on central Hypervisor as shown in Fig. 4 (desktop OS<sub>3</sub>).

### 2.1 Application virtualization

The main goal for virtualization of application is to achieve independence between applications and operating system itself. The virtualization layer provides an interface between different applications and operating systems, creating isolation of application from operating system (Sand Box), see Fig. 5. The applications are independent on installed operating system and cannot change its environment causing potential conflicts with the other applications.

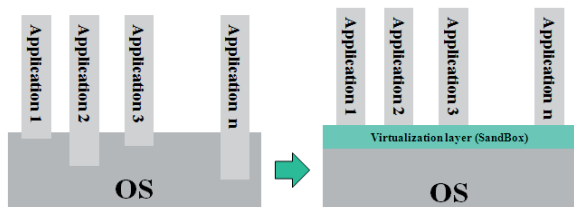


Fig.5 Desktop virtualization

### 1.4 Network virtualization

Network virtualization has been used for long time. There are two basic approaches: Technology of VLANs (Virtual LANs) and virtualization of network devices. VLAN standard has been known since 90s. All major network vendors support creation of virtual LANs, based on the protocol IEEE 802.1q, adding virtualization to the switched network infrastructure, as explained by Fig. 6a. In addition, some vendors offer not only VLAN support, but can provide the virtualization of the active network components (virtual switches, routers). This approach emulates a logical network component for all end-devices that are connected to several physical components represented by virtual logic network device (Fig. 6b).

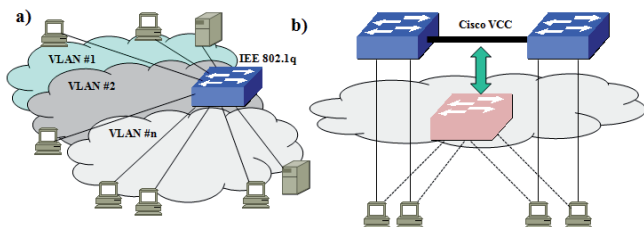


Fig.6 Network virtualization – VLAN/VCC

## 2. Virtualization and MES requirements

As explained in chapter 1, MES systems have very specific requirements and challenges that can be partly addressed by means of virtualization. In this chapter, we will try to list the possible usage of virtualization for MES area.

### 2.1 Resource provisioning on-demand

The needed capacity of MES resources can be extended according to the demands and without unexpected service shortages. A standard virtual machine with the MES application and requested operating system can be copied onto a server in a matter of minutes whenever additional resources are needed. In addition, the virtual machine can be tested in advance to ensure correct function prior its deployment into production environment.

### 2.2 Failover and rollback

A virtual machine's image residing on one physical server can be periodically copied or replicated to another server. This procedure would provide regular system backup and very fast system recovery. In case of a data corruption, dedicated virtual management software can be used to perform point-in-time rollbacks. The roll-back could ensure a transition to the last known good/healthy state of the system what is crucial factor when using MES system with running production.

### 2.3 Upgrades without downtime

Virtualization can support a technological replacement and upgrades without necessity to plan major maintenance windows. Although the system unavailability must be taken into consideration, the needed time is just limited to move of VM machine to the newer hardware with no change to the application or operating system, which is called as a live migration. Live migration works by replicating the system state iteratively while the applications keep running. Shortly before a final copy of the virtual machine is ready for migration, only a brief application blackout (perhaps milliseconds) is necessary to synchronize the second virtual machine with the original. In cases where the operating system is not visible to the application, no interruption to the application and unavailability is perceived by users.

### 2.4 High system availability

Virtualization provides a new and rich set of capabilities for protecting virtual servers and control rooms. This makes it possible to provide faster, more reliable, and more affordable disaster recovery protection. Switching of virtual machines between different host systems is possible, which is much more simplified approach than former clustering.

## 3. Use cases for virtualization in MES/DCS

Today's production facilities have to maintain separate servers to run multiple applications and perform various functions within the production plants. Each server requires space, power and cooling. Hardware and OS (Operating System) needs regular upgrades, operations and maintenance, complications in system management. Because of these issues, reducing the number of physical servers required for plant operations has become a priority. Therefore MES and ERP servers can be virtualized with using hypervisor as explained in chapter 1.1 and shown in Fig. 7b.

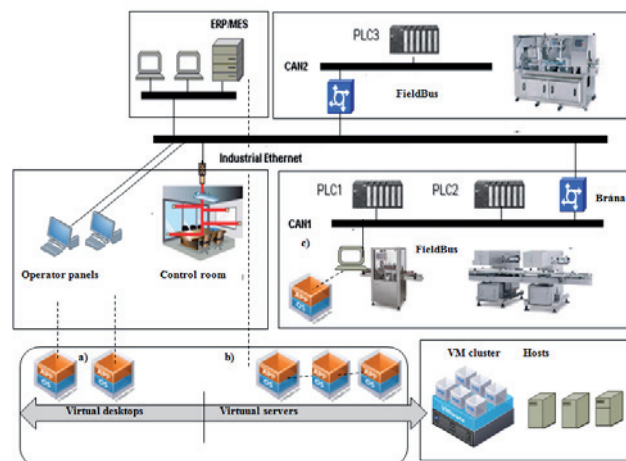
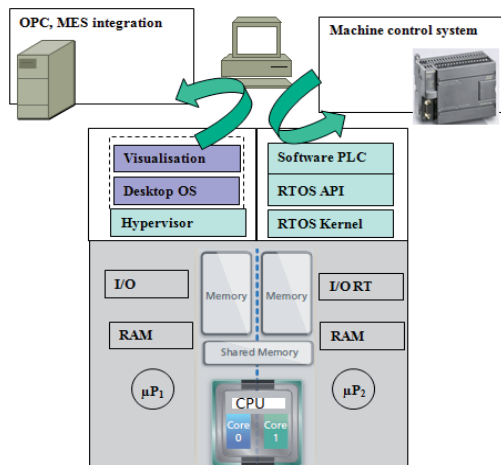


Fig.7 Virtualization for MES and Process control

As well as, various operator, supervisor and batch workstation can be integrated to virtual desktop environment to secure and save infrastructure related and enable independence of used control applications on operating systems. The possible application of virtualization in MES environment is shown in Fig. 7a.

However, there is still open, how to apply virtualization for desktops/servers with core control functions or applications. Distributed control systems have to control machines in real-time with exact timing and deterministic way. The virtualization is based on additional software layer that emulates hardware resources, therefore a direct access and control of hardware resources is not provided.

To deal with this constrain and potential latency, the hybrid solution shall be applied. Workstation, which is used for process control as shown in Fig. 7c, can be based on so called hardware partitioning. Hardware could be divided to two parts. The first part will be assigned for non critical application with VM support (hypervisor), to provide for example process visualization (SCADA), OPC client and reporting. This part will have assigned physical resources to hypervisor. The second hardware part should be used for real time processing (real time OS) with control application (software PLC) for machine control. This part must have a direct access to assigned hardware resources (not assigned to hypervisor) in order to eliminate a latency and ensure a real time deterministic behavior. The HW partitioning is shown in Fig. 8.



**Fig.8 Hardware partitioning for SoftPLC**

## Conclusion

Virtualization offers several opportunities for MES systems integrated with DCS and process control. The main goal is to achieve a hardware consolidation, eliminate high hardware complexity and reduce needed operations and maintenance cost. An additional benefit is represented by high availability, better fail-safe support and fast recovery system in contrast to previous hardware provisioning and standard technologies, such as a clustering. This article tried to review the current virtualization technologies with focus on the use cases for virtualization in MES. As a result, the following benefits can be listed: Physical hardware can be replaced with dissimilar hardware without impacting the virtual machine. There is simplified system maintenance with live migration. The virtual computers can be modified, copied, and replaced quickly and easily just like normal files. As well as, virtualization can reduce security issues and risk environments by providing greater flexibility for testing and development. Despite all listed opportunities, virtualization cannot be simply applied for process control and real time application. There is a still need to use alternative solutions

with dedicated physical hardware, PLCs or hardware partitioning.

## Acknowledgement

*This paper has been supported by the Educational grant agency (KEGA) Nr: 024ŽU-4/2012: Modernisation of educational technologies and methods with focus on cryptography for safety - related applications.*

## References

- [1] PENIAK, P., FRANEKOVÁ, M.: Trendy v integrácií radiaciach a informačných systémov. ATP Journal, 3/2012, s. 32-34. ISSN 1335-2237
- [2] ZOLOTOVÁ, I., KARCH, P., LACIŇÁK, S.: Riadenie a vizualizačné systémy. Elektronická publikácia s podporou Európskeho sociálneho fondu v rámci Operačného programu VZDELÁVANIE. FEI TU Košice. ISBN: 978-80-553-0958-3
- [3] KÁLLAY, F., PENIAK, P.: Počítačové siete LAN, MAN, WAN a jejich aplikace. Monografia, rada Publishing 2003, ISBN 80-247-0545-1
- [4] FRANEKOVÁ M., KÁLLAY F., PENIAK P., VESTENICÝ, P.: Komunikačná bezpečnosť priemyselných sietí. Monografia, EDIS ŽU Žilina 2007, ISBN 978-80-8070-715-6
- [5] RÁSTOČNÝ, K., ŽDÁNSKY, J.: Specificities of safety PLC based implementation of the safety function. In: Proceedings of international conference applied electronics. AE 2012, Pilsen, pp. 229-232, ISBN 978-80-261-0038-6, ISSN 1803-7232, 2012.

## Abstract

The paper deals with problems of virtualization process within manufacturing execution systems integrated with distributed control systems. The main part the solutions of virtualization in MES are analysed on base of practice experience which authors obtained in manufacturing corporation Continental Matador, s. r. o.

### Ing. Peter Peniak, PhD.

Katedra radiaciach a informačných systémov  
Elektrotechnická fakulta Žilinskej univerzity v Žiline  
Univerzitná 1  
010 26 Žilina  
E-mail: peter.peniak@fel.uniza.sk

### Prof. Ing. Mária Franeková, PhD.

Katedra radiaciach a informačných systémov  
Elektrotechnická fakulta Žilinskej univerzity v Žiline  
Univerzitná 1  
010 26 Žilina  
E-mail: maria.franekova@fel.uniza.sk

# Numerical model of high availability system with server virtualization

Peter Peniak

## Abstract

This article deals with the problem of creation high availability systems based on the virtualization. The server and storage virtualization, in contrast to the classical clustering solution, offers much more flexible technique for creation of the redundant systems with supported fast recovery or fallback systems. The main focus of this article is to create the numerical model to explore the features and properties of high availability systems, which could be applied in distributed control systems.

**Keywords:** high availability systems, clusters, virtualization, hypervisor, virtual machine, recovery systems, hardware partitioning, finite population queuing model

## Introduction

Virtualization provides a new and rich set of capabilities for protecting servers and control applications. This is a very good base for faster, more reliable and more affordable disaster recovery protection [1].

The server virtualization is based on so called VM (Virtual Machine) that represents logical server object. This simulated server hardware object behaves exactly like a physical computer or storage and runs as an isolated operating system installation on shared hardware, called a "host". The process of virtualization works by inserting a thin layer of virtualization kernel software, also called as "hypervisor", directly into the computer hardware.

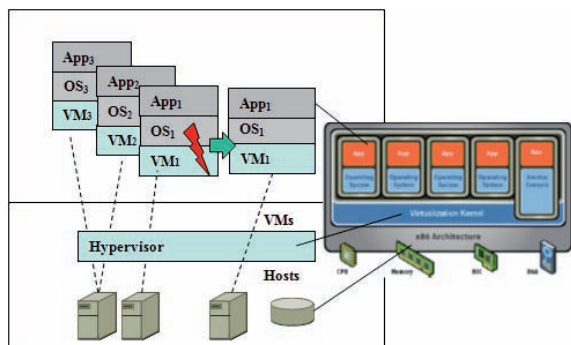


Fig.1 Server/storage virtualization and VM move option

Switching of virtual machines between different host systems (VM move) is also possible. This is much more simplified approach than former clustering [3]. In case of a failure of host computer, all the virtual machines can be automatically moved and restarted on a different physical host server. The switching of VM machines between host systems is managed by specialized virtualization software called "VMotion". However, the common data store with all VM server snapshots has to be available to process fail-over procedure. The described feature is illustrated on Fig. 1, where virtual machine VM<sub>1</sub> has been broken and virtualization management system has initiated the fail-over procedure with restart of VM<sub>1</sub> server on different physical server. This situation would remain till the original server is again brought to production state and VM<sub>1</sub> can migrate back to the former host hardware.

Another option is to provide fail-over operation between hosts in two independent server rooms. In addition to the standard switching, the virtualization has to deal with situation when the data-stores are located in the different storage systems. In such case, there is a need to replicate data stores between data centers with using an additional technology called "Storage VMotion", which has to perform switching of data stores and application data between control rooms in case of failure in one control room. The virtualization is not responsible for replication of file systems with data-stores. That is vital function of interconnected storage systems which can be configured as dedicated storage area network (SAN), based on standard Fiber Channel protocol, or LAN communication is applied with need to use one of the open network protocols for storages such as iSCSI, FCoE (FiberChannel over Ethernet).

This scenario is illustrated on Fig. 2, where the host computers are connected by LAN and storage systems via SAN. If VM<sub>1</sub> has been broken, virtual machine is restarted in another Data Center from replicated data store [1].

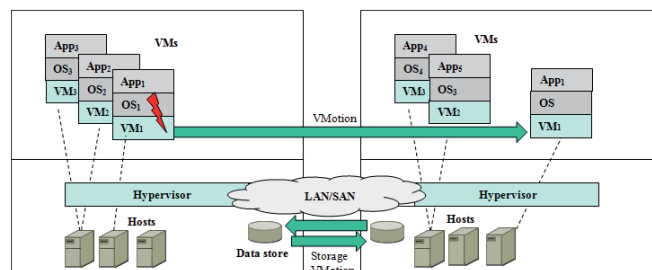


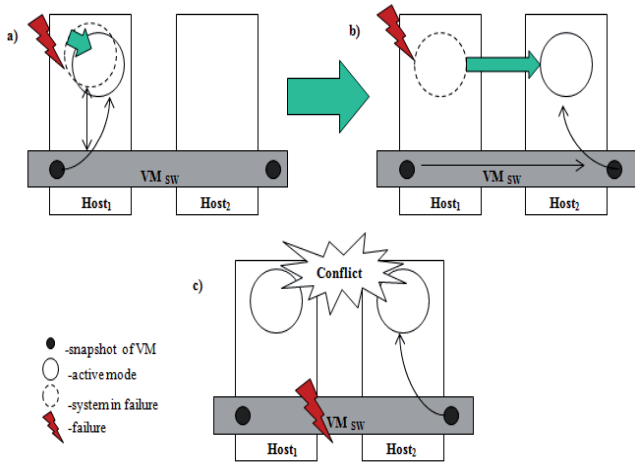
Fig.2 Server/storage virtualization with VM move and Storage Move option between Data centers

## 1. Model for virtual machine move

The procedure for switching of virtual machines between physical host systems has been illustrated on Fig. 1 and Fig. 2. In this chapter, we will try to create a numerical model to evaluate a system behavior and properties of simple solution for virtual server/storage infrastructure with regard to VM move procedure.

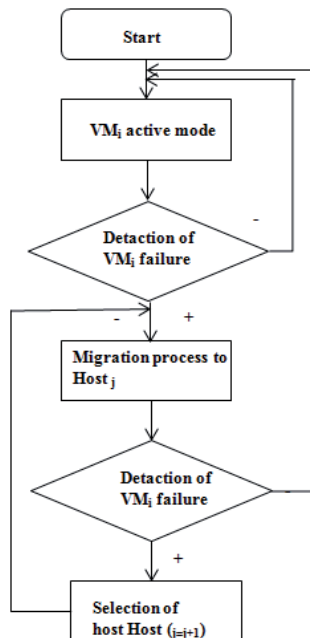
Suppose that VM move procedure can be represented by simple model, as shown on Fig. 3. The proposed model of

system consists of two physical hosts (Host<sub>1</sub>, Host<sub>2</sub>). In case there is a breakdown of one physical server, as illustrated on Fig. 3a, VM move procedure is initiated. Subsequently, the hypervisor of the second host server will detect unavailability of Host<sub>1</sub> and all the virtual machines that were running on Host<sub>1</sub> will be moved to the Host<sub>2</sub>. In this case, each VM machine is started on Host<sub>2</sub> from the snapshot that is available for Host<sub>2</sub> on shared data store (Fig. 3b).



**Fig.3 Model of VM move fail-safe option**

In addition, there is another specific case that must be taken into consideration. If Host<sub>1</sub> and Host<sub>2</sub> cannot communicate each other, for example due to failure of a communication channel (LAN, shared memory), then each physical node might assume that there is breakdown of the other physical server. The result would be much more severe for the same VM servers would be started on host systems causing the confusion for client systems, users and even triggering conflicts of IP addresses on network level and cause full system unavailability. This case is the most critical and represents the major weakness of VM move procedure, as shown on Fig. 3c. The prerequisite for proper system is to ensure high availability and overall stability of communication channel.

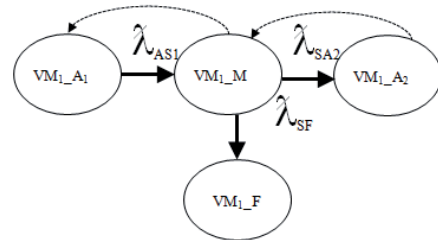


**Fig.4 System flow diagram**

To explain VM procedure in more details, the system flow chart is attached [2], see Fig. 4. Under normal conditions, all VM machines are in active mode. When some potential anomaly happens, the event is detected and migration process is started for all relevant virtual machines belonging to the physical host in failure mode. After successful restart of

VM on another host, the virtual machine is changed to the active mode again. However, there could be a case that restart is not possible. In such case the procedure is repeated for another possible host or even the original host if it is available again.

Another purpose of the model is to determine how VM move approach can enhance the overall availability of the system. For that reason, the state transition model has been prepared [2] to describe the behavior of virtualized system with VM move procedure, as shown in Fig. 5. By mapping through actions to these transition models with stochastic process, the mathematical steady-state solution of the chain can be achieved.



**Fig.5 State diagram**

The model represents three possible states:

- Active state (VM<sub>x</sub>\_A<sub>y</sub>).
- Migration state (VM<sub>x</sub>\_M).
- Failure state (VM<sub>x</sub>\_F).

Where:

- x - is number of virtual machine,
- y - is number of physical host.

Transition from one VM state to the other possible VM state is represented by transition rate ( $\lambda$ ). Initially all of the VMs are in active working state. As time progresses, active VM could eventually transit to unstable state. The detection of failure by system would trigger migration procedure. The VM would move from the active state to the migration state to perform switching to the next available host, with rate ( $\lambda_{AS1}$ ). The migration status can lead to the three possible scenarios with separate rates. First, the VM is successfully activated on next physical host with rate ( $\lambda_{SA2}$ ). Second, the migration is not successful and VM will not be activated, therefore its status will be switched to Failure state with rate ( $\lambda_{SF}$ ). Being in the failure state, all VMs stop running and are not available. Third, in this case the VM goes back from migration state to active state, before switching to the new host. This approach is possible if conditions are quickly corrected, so that VM could recover to the desired original host according to preferred configuration.

Let us use the proposed model to calculate the overall system availability (A). The first equation is obtained by summing the probabilities of all states in the system, which is to be equal to 1, as expressed by formula (1):

$$P_{A1} + P_M + P_{A2} + P_F = 1, \quad (1)$$

where:

- $P_{A1}$  – is probability that VM is active on Host<sub>1</sub>,
- $P_{A2}$  – is probability that VM is active on Host<sub>2</sub>,
- $P_M$  – is probability that VM is in migration phase,
- $P_F$  – is probability that VM is in failed status.

The proposed model can be extended for system with more host systems, which would lead to the more general equation:

$$\sum_{i=1}^n P_{A_i} + P_M + P_F = 1, \quad (2)$$

where  $i$  is represents the number of host systems in VM solution.

As a result, the availability of virtual machine can be derived from (2) as overall probability that system is in active mode, either on physical Host<sub>1</sub> or Host<sub>2</sub>, see relation (3):

$$A = \sum_{i=1}^n P_{A_i} = 1 - (P_M + P_F) \cdot \quad (3)$$

## 2. Finite population queuing model for hypervisor interaction with VMs

Earlier in the chapter we noted that the virtualization is a preferred technology for hardware consolidation and has a potential to support robust fail-safe solutions for MES systems. However, there is still an open question what is a role of virtualization in the field of process control and distributed control systems. The general topic of making virtualization able to support real-time applications with deterministic system response still remains main challenge [3], [4]. For hardware virtualization, an additional virtualization layer has to be added between physical hardware and a virtual machine with dedicated operation system and requested control application software. The new layer might cause not only an additional latency, but could significantly influence overall response time by compromising requested deterministic framework.

There are two possible approaches how to solve this issue. One approach is based on hardware partitioning with physical hardware divided into two independent parts (RAM, processor cores). The first part should be used for hypervisor with virtual machines and non critical applications and the second one for control application with real time requirements, as described in [1]. Another approach could be based on a real-time capable hypervisor (RTH), as shown in Fig. 6.

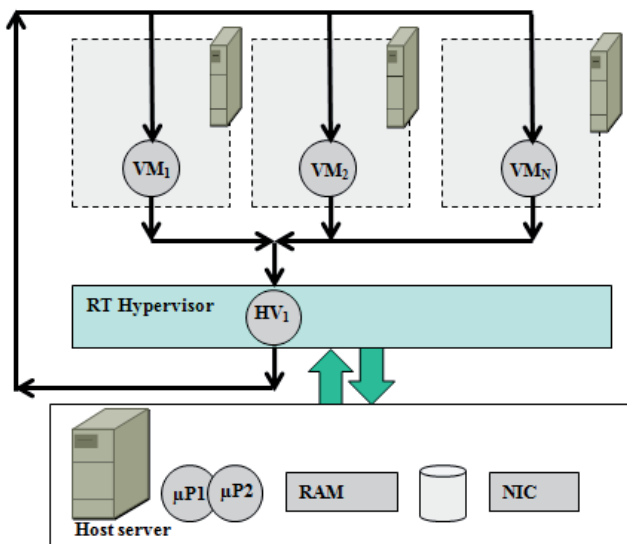


Fig.6 Model of VM server and Hypervisor interaction

RT hypervisor would have to control assignment of physical resources (RAM, NIC, μPC) to virtual machines (VM<sub>1,...,N</sub>) equally in precise time slots, emulating Token based approach among virtual machines. It means that host resources are assigned to each VM cyclically. This approach enables deterministic response time and real-time behavior. According to proposed solution, a numerical model can be created to evaluate basic interaction of hypervisor module with particular VM servers.

Having taken into consideration the request to support real-time properties by hypervisor, let us assume that the model can be represented by finite population queuing model that is often used for interactive systems [5]. Hypervisor subsystem consists of queue for centrally managed hardware resources. Each VM server (VM<sub>i</sub>) interacts with hypervisor (HV<sub>1</sub>) and can be in either an idle status, without request for resources, or requesting status, or receiving status. In order to achieve real time properties, a time-sharing model is suggested. Hypervisor resources are assigned equally among all requesting VM servers. After solving the first assigned interaction, hypervisor serves another interaction in queue. The performance can be evaluated by service rate (S<sub>r</sub>), which is defined as a performance of hypervisor (HV) instantly assigned among all interactions for given physical resource, see relation (4):

$$S_r = \frac{\mu}{n}, \quad (4)$$

where:

S<sub>r</sub> – is hypervisor service rate,

n – is number of interaction among Hypervisor and VMs,

μ – is hypervisor performance per resource (for example MIPS).

The proposed numerical model of finite population queuing system can be expressed by the following equations [5]:

$$W = \frac{N \times E_s}{\rho} - E_t, \quad (5)$$

$$\rho = 1 - p_0, \quad (6)$$

$$\lambda_T = \frac{\rho}{E_s}, \quad (7)$$

$$p_0 = \frac{1}{\sum_{n=0}^N \frac{N!}{(N-n)!} \times \left(\frac{E_s}{E_t}\right)^n}, \quad (8)$$

where:

W – is response time of hypervisor,

E<sub>s</sub> – is service time per VM interaction,

E<sub>t</sub> – is time between finished and new interaction (idle time),

N – is amount of VM servers,

ρ – is hypervisor utilization,

p<sub>0</sub> – is probability that hypervisor is idle,

λ<sub>T</sub> – is throughput in interactions per VM unit time.

The most important parameter, which is used to describe performance, is overall response time of hypervisor, represented by (5). It is calculated from assigned service time per each interaction E<sub>s</sub>, which is multiplied by number of all VM hosts (N) and divided by Hypervisor utilization and time to next request for interaction E<sub>t</sub>. Hypervisor utilization is related to probability that hypervisor is not idle and there are requirements for interactions (n) from possible VM hosts (N), based on (5). Another useful parameter that represents capabilities of hypervisor is its throughput (λ<sub>T</sub>), in interactions per assigned service time for virtual machine.

## Conclusions

Virtualization is one of the most developing information technologies. It is getting popular and gaining significant interest in the enterprise information systems. In spite of enormous success in traditional information technologies, its applications in distributed control systems remains still a challenging field. Therefore the main goal of this article was



to propose possible numerical models, which could provide an abstraction from the real world system and relate only to the behavior and characteristics of interest concerning creation of High availability systems. Two major models were constructed with focus on desired properties of virtualization in field of control systems. The first one stands for fail-safe features of virtualization and is based on the state transition model. This model is applied for VM move procedure with formulation of expected VM availability. The second one uses the finite population queuing model to concentrate on overall system throughput and response time of Virtual machines. The both models can be used for more detailed analysis of virtualization and its influence to distributed control systems. However, this is just first step that is to be followed by experimental evaluation of proposed models in order to verify the selected approach and compare the real results with constructed model.

## References

- [1] PENIAK, P., FRANEKOVÁ, M.: Trendy v integrácií riadiacich a informačných systémov. ATP Journal, 3/2012, s. 32-34. ISSN 1335-2237
- [2] THANDAR THEIN, SUNG-DO CHI, JONG SOU PARK: Availability Modelling and Analysis on Virtualized Clustering with Rejuvenation. IJCSNS International Journal of Computer Science and Network Security, pp. 120-128, VOL.8 No. 9, September 2008
- [3] FRANEKOVÁ, M., KÁLLAY, F., PENIAK, P., VESTENICKÝ, P.: Komunikačná bezpečnosť priemyselných sietí. Monografia, EDIS ŽU Žilina 2007, ISBN 978-80-8070-715-6
- [4] RÁSTOČNÝ, K., ŽDÁNSKY, J.: Specificities of safety PLC based implementation of the safety function. In: Proceedings of international conference applied electronics. AE 2012, Pilsen, pp. 229-232, ISBN 978-80-261-0038-6, ISSN 1803-7232, 2012.
- [5] ALLEN A. O.: Queuing Models of Computer Systems. In: Computer, April 1980, Volume 13 Number 4, ISSN 0018-9162, 1980

## Abstract

The paper deals with creation of numerical model for virtualization, which is applied for creation of High availability systems. The main focus is paid on modeling of availability with fail-safe features (VM move) and quantification of response time and throughput of virtual machine.

## Ing. Peter Peniak, PhD.

Katedra riadiacich a informačných systémov  
Elektrotechnická fakulta Žilinskej univerzity v Žiline  
Univerzitná 1  
010 26 Žilina  
E-mail: peter.peniak@fel.uniza.sk  
Continental Matador Truck Tires , s. r. o.  
Terézie Vansovej 1054  
020 01 Púchov  
E-mail: peter.peniak@conti.sk

# Availability and safety of typical SRCS architectures with safety PLC

Karol Rástočný, Juraj Ždánsky

## Abstract

Safety PLCs (Programmable Logic Controller) are one of the appropriate tools for implementation of safety-related control system (SRCS - Safety Related Control Systems), because safety integrity level (SIL - Safety Integrity Level) is defined by manufacturers. In some cases is necessary to fulfil the requirements for availability of SRCS, except its safety requirements. Achievement of these properties depends on the choice of an appropriate architecture SRCS. Safety PLCs are modular systems and therefore allow creating a number of different architectures. This paper considers the influence of some SRCS architectures with safety PLC to its availability and safety.

**Key words:** SRCS, safety PLC, probability of failure state, probability of dangerous failure, random failure rate, dangerous failure rate

## Introduction

Presently it's possible to see an effort of PLC manufacturers for expansion PLC application possibilities (for example [1], [2], [3]). There are many different categories of PLCs appropriate for control of a group of processes. The safety PLCs are also one of these categories. Primarily they are designed for implementation the SRCS in the industry, but also in other areas (e.g. transport).

Attribute of safety PLC is after failure transition with defined probability into pre-defined safe state (it's attribute marked as fail-safe). For all commercially available safety PLC, safe state is considered as state in which output is disconnected - state without power (logical level 0 in output). (Note: it's property of the safety PLC, not property of the SRCS, which include safety PLC.)

The fundamental difference in manufactures attitude to ensuring required availability and safety of the safety PLC is that some manufacturers observe these properties separately (they offer PLC with increased availability and safety PLC with increased safety) and some manufactures offer safety PLC with modular architecture, which allows synchronously observing of availability and safety increasing of created SRCS.

As the achievement of required safety properties, as well as the achievement of required availability is implemented by appropriate application of redundancy. Literature relating to the SRCS with safety PLC often says about using of redundancy only in the context of increasing availability. This specificity is due to the fact that the redundancy associated with increasing safety properties is mainly applied in the modules of the safety PLC and therefor from user viewpoint it's "invisible" redundancy. Conversely, the redundancy associated with increasing availability is from the user viewpoint "visible", because it's implemented by using number of modules the same type.

From this viewpoint SRCS with safety PLC can be divided to:

- SRCS with safety PLC without redundancy;
- SRCS with safety PLC with redundancy.

The paper deals with the influence of random hardware failure on availability and safety of different architectures of SRCS with safety PLC. Availability and safety of individual architectures are evaluated against the consequences of random hardware failure. Evaluation of software is not the subject of this contribution.

## 1. Decomposition of SRCS with safety PLC

Fig. 1 shows an example of decomposition of SRCS with safety PLC. In this case, its input part consists of the sensors and the F-I module (Fail-Safe Input module), the logic consists of F-CPU (Fail-safe Central Processing Unit) and the output part consists of F-O module (Fail-safe Output module) and actuators. Fig. 1 doesn't show modules that are necessary for the operation of SRCS, but in compliance with specified conditions of use don't affect safety (e.g. power supply).

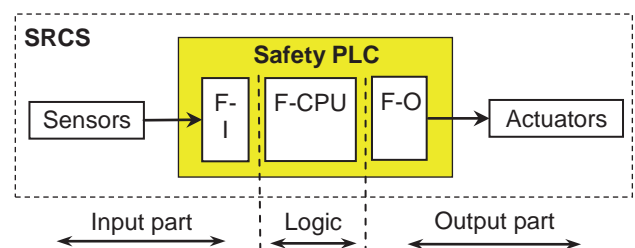


Fig. 1 Decomposition of SRCS with safety PLC

By combining of the different architectures of the input part, the logic and the output part can be achieved a wide range of architectures of SRCS with safety PLC. The paper in consideration of specified issue scale is more detail about availability and safety of the input part of SRCS. A similar method can be followed in the evaluation of availability and safety of the output part or the logic.

More difficult architectures can be created using several safety PLCs or their modules at different control levels. Such a distribution of safety functions has its own specifics (literature [4] refers some of them), which should be taken in consideration during creation the architecture.

In the next section of this paper let's assume that:

- individual considered parts of SRCS are independent of each other in the meaning that an occurrence of failure in one part of the SRCS does not affect the probability of failure in other part of the SRCS;
- SRCS implement one safety function and thus SIL SRCS correspond to the SIL of the safety function.

Then the probability of SRCS failure state (decomposed according to Fig. 1) can be expressed by equation:

$$P(t) = P_I(t) + P_L(t) + P_O(t) - P_I(t) \cdot P_L(t) - P_I(t) \cdot P_O(t) - P_L(t) \cdot P_O(t) + P_I(t) \cdot P_L(t) \cdot P_O(t), \quad (1)$$

where  $P_I(t)$ ,  $P_L(t)$  and  $P_O(t)$  are probabilities of failure state of the input part, the logic and the output part of the SRCS.

The safety integrity level for SRCS is expressed by the dangerous failures rate per hour and the function [5]. The dangerous failures rate for SRCS according to Fig. 1 can be expressed by equation:

$$\lambda_H(t) = \lambda_{HI}(t) + \lambda_{HL}(t) + \lambda_{HO}(t), \quad (2)$$

where  $\lambda_{HI}(t)$ ,  $\lambda_{HL}(t)$  and  $\lambda_{HO}(t)$  are dangerous failures rates of the input part, the logic and the output part of SRCS.

## 2. The input part without redundancy

In examples of architectures we consider scanning of one value, even if to the implementation of specific safety function can be necessary the scanning of several values and thus using several sensors and F-DI modules. Analysis of availability and safety of the input part of SRCS with safety PLC must cover all sensors and F-DI modules that are involved in the implementation of the safety function.

F-I modules of safety PLC enable two-channel connection of sensors. This must correspond to their internal architecture. F-I modules therefore have two independent, galvanically isolated channels with a separate power supply (channels are marked L (Left) and R (Right)). Evaluation of the inputs of the channels L and R leads to their mutual comparison. Comparison takes place directly in the F-I module and can be parameterized depending on the manufacturer and modules type at different scales. Program in the F-CPU operates only with the comparison result (variable I in Fig. 2).

### 2.1 Connection with one sensor

Connection of the input part of SRCS with one sensor (Fig. 2) can be used when the sensor output in basic state is log. level 1 and is eliminated failure as permanent log. 1 with such a probability that corresponds to the required SIL (Safety Integrity Level) of the safety function [5] (this is a necessary but not sufficient condition).

For the probability of the failure state of the input part of SRCS with one sensor (Fig. 2) is valid:

$$P_I^{1S}(t) = 1 - (1 - P_S(t))(1 - P_{DI}(t)), \quad (3)$$

where  $P_S(t)$  is probability of failure state of sensor S and  $P_{DI}(t)$  is probability of failure state of F-DI module.

For the dangerous failure rate of the input part of SRCS with one sensor (Fig. 2) is valid:

$$\lambda_{HI}^{1S}(t) = \lambda_{HS}(t) + \lambda_{HDI}(t), \quad (4)$$

where  $\lambda_{HS}(t)$  is dangerous failure rate of sensor S and  $\lambda_{HDI}(t)$  is dangerous failure rate of F-DI module.

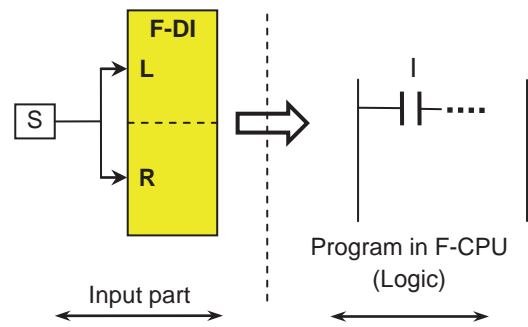


Fig. 2 Connection of the input part with one sensor

### 2.2 Connection with two sensors

Connection of the input part of SRCS with two sensors (Fig. 3) doesn't have any special requirements on the safety properties of sensors. Sensors  $S_L$  and  $S_R$  scan the same value and in F-DI module there is a mutual comparison of the information from the sensors. If the information coming from the sensors  $S_L$  and  $S_R$  is identical (in tolerances defined by module parameters), the variable I (comparison result) in Fig. 3 is equal to information coming from the sensors. If the information is not equal, the variable I obtain permanent pre-defined safe state (logical 0).

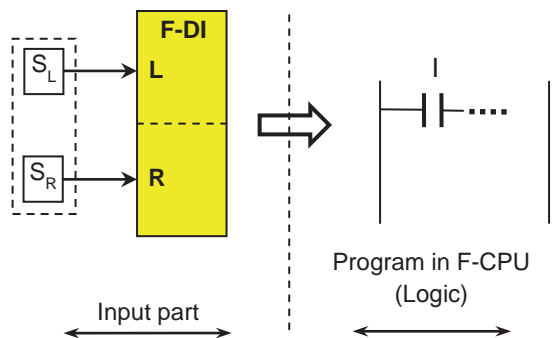


Fig. 3 Connection of the input part with two sensors

For probability of failure state of the input part of SRCS with two sensors (Fig. 3) is valid:

$$P_I^{2S}(t) = 1 - (1 - P_{SL}(t))(1 - P_{SR}(t))(1 - P_{DI}(t)). \quad (5)$$

where  $P_{SL}(t)$  and  $P_{SR}(t)$  are probabilities of failure state of sensors  $S_L$  and  $S_R$  and  $P_{DI}(t)$  is probability of failure state of F-DI module.

For dangerous failure rate of the input part of SRCS with two sensors (Fig. 3) is valid:

$$\lambda_{HI}^{2S}(t) = \frac{dP_{HS}(t)}{1 - P_{HS}(t)} + \lambda_{HDI}, \quad (6)$$

where  $P_{HS}(t)$  is the probability of dangerous failure of two sensors  $S_L$  and  $S_R$ ,  $\lambda_{HDI}$  is the dangerous failure rate of F-DI module (it's a constant which is part of its technical documentation).

The probability of dangerous failure of two sensors  $S_L$  and  $S_R$  can be expressed by term:

$$P_{HS}(t) \leq P_{SL}(t) \cdot P_{SR}(t). \quad (7)$$

The value of the probability  $P_{HS}(t)$  depends on how often sensor functionality is checked. Actually, verifying of sensor functionality occurs when the sensor changes its state. The state change depends on operational conditions. From safety reasons it's necessary to consider the worst case (the

longest time between two state changes of the sensor). If this time cannot be clearly defined, it can be equated with the time between preventive controls or the useful life of SRCS.

For considerations in the next section of this contribution is valid:

- the each sensor failure is detected in the time  $t_{0S}$ ;
- random failure rates of sensors are constants (exponential distribution of failure occurrence).

Then probability of dangerous failure of two sensors  $S_L$  and  $S_R$  can be expressed by term:

$$P_{HS}(t_{0S}) \leq (1 - e^{-\lambda_{SL} \cdot t_{0S}}) \cdot (1 - e^{-\lambda_{SR} \cdot t_{0S}}), \quad (8)$$

where  $\lambda_{SL}$  and  $\lambda_{SR}$  are random failures rates of sensors  $S_L$  and  $S_R$ .

If  $\lambda \cdot t \ll 1$ , the random failures rate of two sensors  $S_L$  and  $S_R$  can be determined according to equation:

$$\lambda_{HS}(t_{0S}) \cong 2 \cdot \lambda_{SL} \cdot \lambda_{SR} \cdot t_{0S}. \quad (9)$$

The dangerous failures rate of the input part of SRCS with two sensors can be determined according to term:

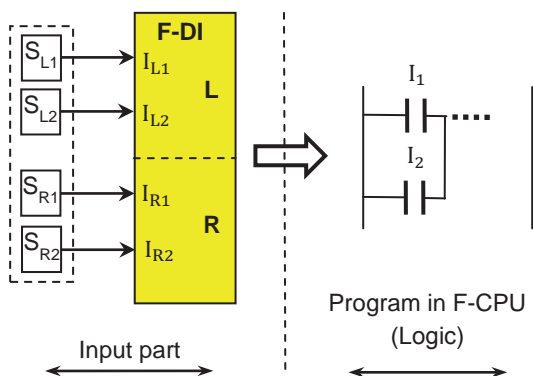
$$\lambda_{HI}^{2S}(t_{0S}) \leq \lambda_{HDI} + \lambda_{HS}(t_{0S}), \quad (10)$$

where  $\lambda_{HDI}$  is random failures rate of F-DI module.

### 3. Input part with redundancy

#### 3.1 Redundancy at the sensors level

Fig. 4 shows connection of the input part of SRCS with redundancy at the sensors level. This connection can be considered as a double connection with two sensors (Fig. 3). Sensors  $S_{L1}$  and  $S_{R1}$  are working together with sensors  $S_{L2}$  and  $S_{R2}$ . Comparison of the sensors  $S_{L1}$  and  $S_{R1}$  (respectively  $S_{L2}$  and  $S_{R2}$ ) takes place directly in the F-DI module (as well as in Fig. 3). Evaluation of the information coming from sensors pairs shown in Fig. 4 is expressed by operation of the logical sum (parallel sequencing of instructions in the ladder program). This evaluation must be realized in the application program, because safety PLCs don't allow configuration of redundancy in a single F-DI module.



**Fig. 4 Connection of the input part with redundancy at the sensors level**

For the probability of failure state of the input part of SRCS with redundancy at the sensors level (Fig. 4) is valid:

$$P_I^{RS}(t) = 1 - (1 - P_{S1}(t) \cdot P_{S2}(t))(1 - P_{DI}(t)), \quad (11)$$

where  $P_{S1}(t)$  is the probability of failure state of two sensors  $S_{L1}$  and  $S_{R1}$  and  $P_{S2}(t)$  is the probability of failure state of

two sensors  $S_{L2}$  and  $S_{R2}$  and  $P_{DI}(t)$  is the probability of failure state of the F-DI module.

The probability of failure state of two sensors  $S_{L1}$  and  $S_{R1}$  can be expressed by equation:

$$P_{S1}(t) = 1 - (1 - P_{SL1}(t))(1 - P_{SR1}(t)), \quad (12)$$

where  $P_{SL1}(t)$  and  $P_{SR1}(t)$  are the probabilities of failure state of sensors  $S_{L1}$  and  $S_{R1}$ .

Similarly, the probability of failure state of two sensors  $S_{L2}$  and  $S_{R2}$  can be expressed by equation:

$$P_{S2}(t) = 1 - (1 - P_{SL2}(t))(1 - P_{SR2}(t)), \quad (13)$$

where  $P_{SL2}(t)$  and  $P_{SR2}(t)$  are the probabilities of failure state of sensors  $S_{L2}$  and  $S_{R2}$ .

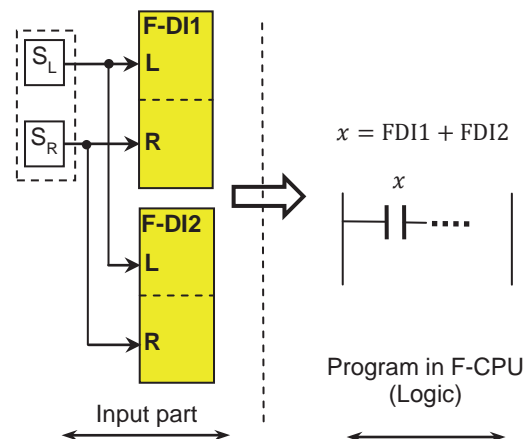
For the dangerous failure rate of the input part of SRCS with redundancy at sensors level (Fig. 4) is valid:

$$\lambda_{HI}^{RS}(t_{0S}) \leq 2 \cdot \lambda_{SL1} \cdot \lambda_{SR1} \cdot t_{0S} + 2 \cdot \lambda_{SL2} \cdot \lambda_{SR2} \cdot t_{0S} + \lambda_{HDI}, \quad (14)$$

where  $\lambda_{SL1}$ ,  $\lambda_{SR1}$ ,  $\lambda_{SL2}$  and  $\lambda_{SR2}$  are the random failure rates of sensors  $S_{L1}$ ,  $S_{R1}$ ,  $S_{L2}$  and  $S_{R2}$ .

#### 3.2 Redundancy at the F-DI modules level

If the reliability of the two sensors  $S_L$  and  $S_R$  is sufficient and the reliability of the F-DI module isn't sufficient, is possible to use connection of the input part of SRCS with redundancy at the F-DI modules level (Fig. 5).



**Fig. 5 Connection of the input part with redundancy at the F-DI modules level**

There are several types of safety PLCs which are enable to parameterize the F-DI modules to implementation the connection shown in Fig. 5. This connection provides to the application program one information about scanned value (variable  $x$  in Fig. 5). This information is given by the logical sum of information provided by the modules F-DI1 and F-DI2. The processing of final information takes place directly in the F-DI modules.

The probability of failure state of the input part of SRCS with redundancy at the F-DI modules level (Fig. 5) can be expressed by equation:

$$P_I^{RM}(t) = 1 - (1 - P_{SL}(t))(1 - P_{SR}(t))(1 - P_{DI1}(t) \cdot P_{DI2}(t)), \quad (15)$$

where  $P_{SL}(t)$  and  $P_{SR}(t)$  are the probabilities of failure state of sensors  $S_L$  and  $S_R$  and  $P_{DI1}(t)$  and  $P_{DI2}(t)$  are probabilities of failure state of modules F-DI1 and F-DI2.

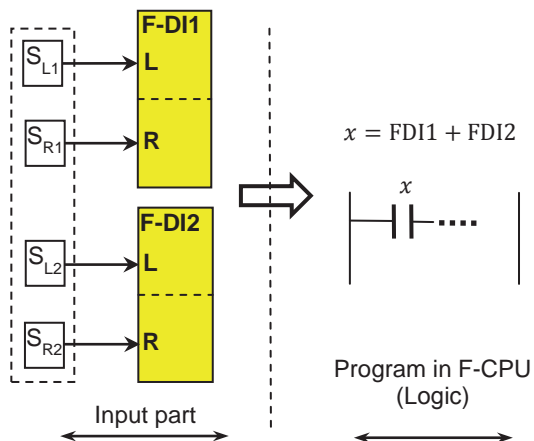
The dangerous failure rate of the input part of SRCS with redundancy at F-DI modules level (Fig. 5) can be expressed by term:

$$\lambda_{HI}^{RM}(t_{0S}) \leq 2 \cdot \lambda_{SL} \cdot \lambda_{SR} \cdot t_{0S} + \lambda_{HDI1} + \lambda_{HDI2}, \quad (16)$$

where  $\lambda_{HDI1}$  is the dangerous failure rate of the F-DI1 module and  $\lambda_{HDI2}$  is the dangerous failure rate of the F-DI2 module.

### 3.3 Redundancy at the sensors and the F-DI modules level

If is necessary to improve the availability of sensors and F-DI module, can be used connection according to Fig. 6. Principle of information processing of the sensors is in this case the same as the connection shown in Fig. 5



**Fig. 6 Connection of the input part with redundancy at the sensors and the F-DI modules level**

The probability of failure state of input part of SRCS with redundancy at the sensors and the F-DI modules level (Fig. 6) can be expressed by equation:

$$P_I^{RSM}(t) = P_{I1}^{2S}(t) \cdot P_{I2}^{2S}(t), \quad (17)$$

where  $P_{I1}^{2S}(t)$  is the probability of failure state of the first channel (components F-DI1, S<sub>L1</sub> and S<sub>R1</sub>) and  $P_{I2}^{2S}(t)$  is the probability of failure state of the second channel (components F-DI2, S<sub>L2</sub> and S<sub>R2</sub>).  $P_{I1}^{2S}(t)$  and  $P_{I2}^{2S}(t)$  can be specified according equation (5).

The dangerous failure rate of input part of SRCS with redundancy at the sensors and the F-DI modules level (Fig. 6) can be expressed by term:

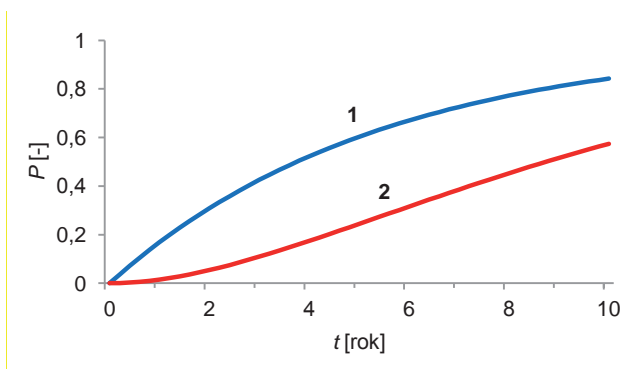
$$\lambda_{HI}^{RSM}(t_{0S}) \leq 2 \cdot \lambda_{SL1} \cdot \lambda_{SR1} \cdot t_{0S} + 2 \cdot \lambda_{SL2} \cdot \lambda_{SR2} \cdot t_{0S} + \lambda_{HDI1} + \lambda_{HDI2}. \quad (18)$$

### 4. Comparison of the selected properties of SRCS architectures with safety PLC

In the previous sections of the contribution were post the availability and safety of typical architectures of the input part of SRCS with safety PLC. In this section are compared the availability and safety of the selected architectures of SRCS (Fig. 1). Curves comparing the selected architectures are shown in Fig. 7 and Fig. 8 and they are compiled for SRCS with safety PLC Simatic (parameters of reliability of the Simatic components can be found in [6] and safety parameters are part of the technical documentation of individual modules). The estimated random failure rate of sensors is  $1.10 \cdot 10^{-6} \text{ h}^{-1}$  and the estimated random failure rate of actuators is  $5.10 \cdot 10^{-6} \text{ h}^{-1}$ .

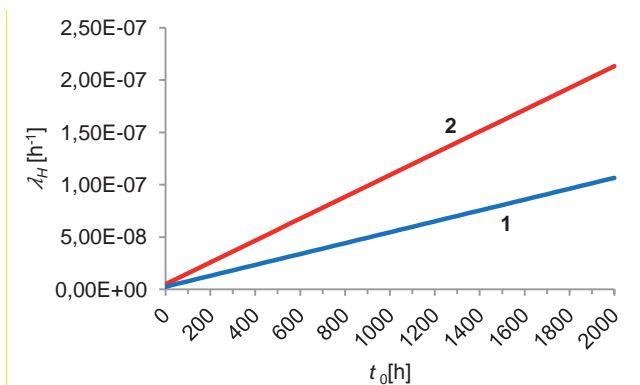
The graph in Fig. 7 shows the probability of failure state of SRCS with safety PLC. Curve 1 is for SRCS without redundancy (compiled of the input part with two sensors (Fig. 3), the logic without redundancy and the output part with two actuators). Curve 2 is for SRCS with redundancy (compiled of the input part with redundancy at the sensors and the F-DI modules level (Fig. 6), the redundant logic and the output part with redundancy at the actuators and the F-DO modules level). More detailed information about architectures of the logic and the output part can be found in the literature [7].

Fig. 7 shows that SRCS with redundancy have a lower probability of failure state than SRCS without redundancy. These models don't take into consideration the recovery process of SRCS. It is obvious that by the SRCS recovery (or its parts) can significantly increase its availability. Influence of the recovery on the availability and safety of SRCS can be modelled by an appropriate method (e.g. Mark's diagram) [8], [9].



**Fig. 7 The probability of failure state of SRCS with safety PLC**

Fig. 8 shows the time curves of the dangerous failure rate for the same SRCS with safety PLC as the Fig. 7. The curve 1 is for SRCS without redundancy and curve 2 is for SRCS with redundancy.



**Fig. 8 The dangerous failure rate of SRCS with safety PLC**

Fig. 8 shows, that dangerous failure rate of SRCS with redundancy (curve 2) is higher as dangerous failure rate of SRCS without redundancy (curve 1). This is caused by the fact that parts forming the reserve affect the safety of SRCS with redundancy.

### Conclusion

An appropriate SRCS architecture for a specific application can generally be estimated by experiences. The following analysis of safety and availability of chosen architecture can confirm the assumed properties, or identify the "weak points". If the analysis don't show the fulfilment of the re-

quired properties, is necessary to change either the architecture of SRCS or apply other appropriate actions to improve these properties (e.g. a change the architecture of only some part of SRCS, application redundancy, creation of test procedures, etc.). The suggested actions should be subject to re-analysis to confirm or refute their sufficiency.

If the requirements for availability and safety of the application cannot be achieved by any of the above typical architectures, is possible with additional hardware and/or software solutions to create a more difficult SRCS architecture (e.g. architecture 2 out of 3). Some manufacturers of safety PLC have for this purpose certified program blocks usable in the application program.

## Acknowledgment

*This publication is the result of the project implementation:*

**Centre of excellence for systems and services of intelligent transport II.,**

*ITMS 26220120050 supported by the Research & Development Operational Programme funded by the ERDF.*



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

## References

- [1] BEZÁK, T. - STRÉMY, M. - HUSÁROVÁ, B.: Distributed control systems modelling using PROFInet CBA. In: Annals of DAAAM and Proceedings of DAAAM Symposium 2010, Vol. 21, No 1., ISSN 1726-9679.
- [2] HRBČEK, J. - ŠIMÁK, V.: Implementation of Multi-dimensional Model Predictive Control for Critical Process with Stochastic Behavior, chapter in: Advanced Model Predictive Control, p.109-124, InTech, Tao Zheng (Ed.), June 2011, ISBN 978-953-307-298-2.
- [3] BÉLAI, I. - DRAHOŠ, P.: The Industrial Communication Systems PROFIBUS and PROFInet. In: Applied Natural Sciences 2009 : International Conference. Trnava, Slovak Republic, pp. 329-336, ISBN 978-80-8105-127-2.

[4] RÁSTOČNÝ, K. - ŽDÁNSKY, J.: Specificities of safety PLC based implementation of the safety functions, Proceedings of International Conference Applied Electronics 2012, 5-7 September 2012, Pilsen, ISBN 978-80-261-0038-6, ISSN 1803-7232, IEEE Catalog Number: CFP1269A-PRT, p. 229-232

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

[6] MTBF\_2009-04.xls, dostupné na [http://www.nwe.siemens.com/denmark/internet/dk/industry/information/Software\\_vejledninger/Document/s/MTBF\\_2009-04.xls](http://www.nwe.siemens.com/denmark/internet/dk/industry/information/Software_vejledninger/Document/s/MTBF_2009-04.xls). Prevzaté 3.12.2012.

[7] 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 2012, IEEE Catalog Number: CFP1248S-ART, p. TA4\_25, ISBN 978-1-4673-1178-6

[8] RÁSTOČNÝ, K. - ILAVSKÝ, J.: Quantification of the Safety Level of a Safety-critical Control System. In: Proceedings of international conference on Applied Electronics 2010. Pages 285-288. IEEE Cat.nr CFP1069A-PRT, Pilsen, 2010, pp.285-288, ISBN 978-0-7043-865-7, ISSN 1803-7232.

[9] RÁSTOČNÝ, K. - ŽDÁNSKY, J.: Application of programmable logic controllers in safety systems. Proceedings of the Scientific Conference ADVANCES IN TRANSPORT SYSTEMS TELEMATICS 2007, pp. 177-182, ISBN 978-83-917156-6-6.

**prof. Ing. Karol Rástočný, PhD.**

University of Žilina  
Faculty of Electrical Engineering  
Department of Control and Information Systems  
Univezitná 8215/1  
010 26 Žilina, Slovak Republic  
Tel.: +421 41 513 3320  
E-mail: karol.rastocny@fel.uniza.sk

**Ing. Juraj Ždánsky, PhD.**

University of Žilina  
Faculty of Electrical Engineering  
Department of Control and Information Systems  
Univezitná 8215/1  
010 26 Žilina, Slovak Republic  
Tel.: +421 41 513 3342  
E-mail: juraj.zdansky@fel.uniza.sk

# Vplyv diagnostiky na integritu **bezpečnosti** SRCS so safety PLC

Juraj Ždánsky, Karol Rástočný

## Abstrakt

Safety PLC (Programmable Logic Controller) sa primárne používajú na realizáciu SRCS (Safety Related Control Systems) na procesnej úrovni, preto treba aj snímače a akčné členy považovať za súčasť SRCS. V takomto prípade môže byť diagnostika ich porúch a včasná reakcia na detegovanú poruchu nevyhnutná na dosiahnutie požadovanej SIL (Safety Integrity Level). V SRCS so safety PLC možno detekciu porúch snímačov a akčných členov riešiť v aplikačnom programe. V takom prípade treba zohľadniť niektoré špecifické vlastnosti safety PLC. Tento príspevok sa venuje niektorým súvislostiam medzi parametrami safety PLC, možnosťami riešenia diagnostiky a integritou bezpečnosti SRCS so safety PLC.

**Kľúčové slová:** SRCS, safety PLC, diagnostika porúch, intenzita nebezpečných porúch, safety program

## Úvod

Diagnostika v SRCS so safety PLC môže byť realizovaná s cieľom dosiahnuť požadovanú SIL (ide o včasnú detekciu a negovania poruchy) a/alebo s cieľom zvýšiť užívateľský komfort počas prevádzky SRCS (ide predovšetkým o presnú lokalizáciu poruchy, čím sa môže výrazne znížiť čas do obnovy SRCS).

V SRCS so safety PLC možno diagnostiku rozdeliť na:

- diagnostiku realizovanú výrobcom (diagnostika realizovaná firmou jednotlivých komponentov SRCS a systémovými funkciami spúšťanými v F-CPU (Fail-safe Central Processing Unit));
- aplikačnú diagnostiku (diagnostika realizovaná používateľom v aplikačnom programe).

Diagnostika realizovaná výrobcom je používateľovi takmer úplne alebo z veľkej časti neznáma. V niektorých prípadoch je používateľovi umožnené nastavovať niektoré parametre diagnostiky a to len v takom rozsahu, aby nedošlo k zníženiu SIL daného komponentu pod hodnotu deklarovanú výrobcom.

Aplikačná diagnostika je zameraná na tie komponenty SRCS, ktoré nemajú diagnostiku realizovanú výrobcom alebo jej rozsah nie je postačujúci na dosiahnutie požadovanej SIL bezpečnostnej funkcie, na realizácii ktorej sa dané komponenty podieľajú. Najčastejšie sú to akčné členy realizujúce pripájanie riadeného objektu (riadených objektov) ku zdroju energie, resp. odpájanie riadeného objektu (riadených objektov) od zdroja energie. Aplikačná diagnostika sa týka aj snímačov, pretože treba kontrolovať schopnosť snímačov reagovať na výskyt udalostí alebo na zmenu veličiny, ktorú vyhodnocujú.

Na základe znalosti spôsobu realizácie bezpečnostnej funkcie, znalosti požadovanej SIL danej bezpečnostnej funkcie a na základe znalosti intenzít náhodných hardvérových porúch prvkov (obvodov), ktoré sa na realizácii bezpečnostnej funkcie podieľajú, možno stanoviť maximálny časový interval, počas ktorého musí byť overená schopnosť snímačov a akčných členov meniť svoj stav [1]. Ak k zmene stavu snímačov a akčných členov počas bežnej prevádzky dochádza častejšie ako je stanovený maximálny časový inter-

val a táto zmena stavu je vhodným spôsobom vyhodnotená riadiacou logikou, tak nie je nutné realizovať testovaciu diagnostiku z dôvodu bezpečnosti. Ak sa aj napriek tomu aplikačná diagnostika realizuje, tak je to z dôvodu zvýšenia pohotovosti SRCS.

Ak zmena stavu snímačov a akčných členov počas bežnej prevádzky nenastáva aspoň raz počas stanoveného časového intervalu, tak treba realizovať testovaciu diagnostiku (formou aplikačnej diagnostiky).

Testovacia diagnostika výstupných obvodov pripojených k safety PLC vyžaduje taký postup, ktorý v rámci stanoveného (vypočítaného) časového intervalu overí schopnosť akčných členov zmeniť svoj stav. Pri realizácii takéhoto testovacieho postupu aplikačným programom safety PLC treba zohľadniť špecifické vlastnosti safety PLC.

Príspevok poukazuje na vplyv niektorých vlastností safety PLC na možnosti aplikačnej diagnostiky, čo môže v konečnom dôsledku ovplyvniť aj dosiahnutú úroveň bezpečnosti SRCS so safety PLC. Príspevok rieši predovšetkým aplikačnú diagnostiku akčných členov. Analogicky však treba postupovať aj pri testovaní snímačov.

## 1. Diagnostika realizovaná výrobcom

Vplyv základných parametrov diagnostiky realizovanej výrobcom možno vhodne ukázať na zapojení výstupnej časti s jedným akčným členom (obr. 1). Takéto zapojenie možno použiť vtedy, ak F-DO (Fail-safe Digital Output) modul a aj pripojený akčný člen vyhovujú požadovanej SIL.

Pre intenzitu nebezpečných porúch výstupnej časti SRCS s jedným akčným členom (obr. 1) platí, že

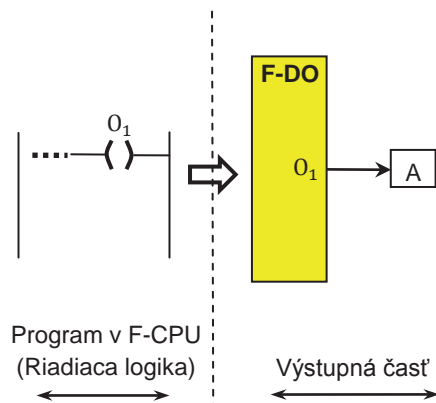
$$\lambda_{HO}^{1A}(t) = \lambda_{HA}(t) + \lambda_{HDO}, \quad (1)$$

kde  $\lambda_{HA}(t)$  je intenzita nebezpečných porúch akčného člena A a  $\lambda_{HDO}$  je intenzita nebezpečných porúch F-DO modulu.

Z dôvodu dosiahnutia požadovanej SIL disponuje F-DO modul vlastnou testovacou diagnostikou (diagnostika realizovaná výrobcom), ktorá v určitých časových intervaloch kontroluje schopnosť F-DO modulu nastaviť úroveň log. 0 na svojich výstupoch (kontrola schopnosti F-DO modulu

prejsť do bezpečného stavu po detekcii poruchy). V prípade detekcie poruchy prechádza F-DO modul do bezpečného stavu – nastavenie signálu log. 0 na príslušnom svojom výstupe (príslušných svojich výstupoch). Tento test prebieha vtedy, keď na výstupe, ktorý má byť testovaný, je signál úrovne log. 1 a treba s ním počítať, pretože:

- môže ovplyvniť aplikačnú diagnostiku akčných členov;
- môže ovplyvniť riadený proces.



Obr. 1 Zapojenie výstupnej časti s jedným akčným členom

Fig. 1 The output part connection with one actuator

Z pohľadu aplikačnej diagnostiky akčných členov by bolo výhodné, ak by sa krátkodobé odpájanie výstupov testovacou diagnostikou modulu dalo využiť aj na vyhodnotenie reakcie akčných členov pomocou aplikačnej diagnostiky. Takéto riešenie si vyžaduje synchronizáciu medzi testovacou diagnostikou modulu a aplikačnou diagnostikou, čo spravidla nie je možné. Preto musí v prípade potreby testovacie impulzy pre akčné členy vytvárať aplikačná logika nezávisle od testovacej diagnostiky modulu.

Z pohľadu bezpečnosti nesmie chybné vygenerovanie tohto impulzu a ani predĺženie času jeho trvania spôsobiť nebezpečný stav (nežiaduca prítomnosť signálu s úrovňou log. 0 na výstupe F-DO modulu). Reakcia akčného člena na takýto impulz však môže nepriaznivo pôsobiť na riadený proces. Aby k takejto reakcii nedošlo, musí platiť, že

$$t_A^{1 \rightarrow 0} > t_O^0, \quad (2)$$

kde  $t_A^{1 \rightarrow 0}$  je čas odozvy akčného člena na zmenu signálu z úrovne log. 1 na úroveň log. 0 a  $t_O^0$  je čas trvania testovacieho impulzu úrovne log. 0.

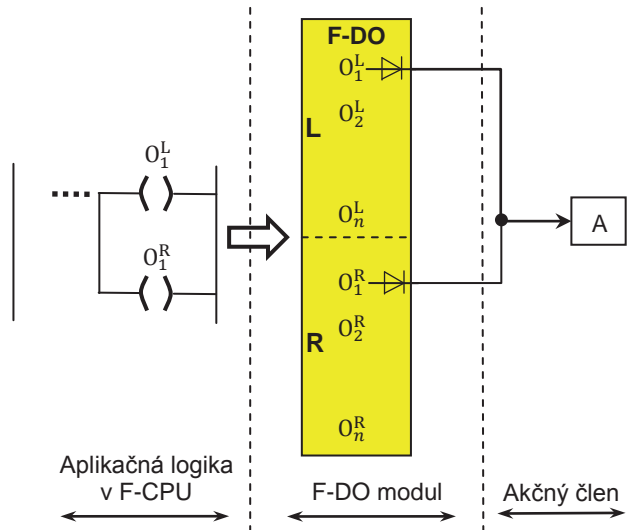
V prípade, že akčný člen nespĺňa podmienku (2), možno použiť riešenie znázornené na obr. 2. Riešenie spočíva v pripojení akčného člena na dva výstupy, pričom treba zaisťiť, aby boli testovacie impulzy na výstupoch vzájomne časovo posunuté a aby nedochádzalo k vzájomnému ovplyvňovaniu výstupov (skrat jedného výstupu druhým; na obr. 2 je tento problém vyriešený použitím oddeľovacích polovodičových diód). Časové posunutie testovacích impulzov vyžaduje vzájomnú synchronizáciu testovania, čo nie je užívateľsky možné. Existujú však F-DO moduly, ktoré majú internú synchronizáciu testovania medzi príslúchajúcimi výstupmi kanálov L a R (ľavý a pravý kanál F-DO modulu). Vplyv takéhoto riešenia na bezpečnosť treba podrobiť analýze bezpečnosti, pretože k chybnému ovplyvneniu akčného člena môže dôjsť vplyvom poruchy jedného alebo druhého výstupu.

F-DO moduly spravidla disponujú nielen diagnostikou nevyhnutnou na dosiahnutie požadovanej úrovne bezpečnosti, ale aj diagnostikou slúžiacou na včasnú detekciu nie nebezpečnej poruchy. Na takúto diagnostiku slúžia aj testovacie impulzy úrovne log. 1. Testovací impulz úrovne log. 1 má význam vzhľadom na pohotovosť SRCS, pretože ním F-DO

modul testuje, či nedošlo k odpojeniu záťaže (akčných členov) v stave, keď je na výstupe modulu úroveň log. 0. Aby nedošlo k ohrozeniu bezpečnosti riadeného procesu, treba použiť akčný člen spĺňajúci podmienku:

$$t_A^{0 \rightarrow 1} > t_O^1, \quad (3)$$

kde  $t_A^{0 \rightarrow 1}$  je čas odozvy akčného člena na zmenu signálu z úrovne log. 0 na úroveň log. 1 a  $t_O^1$  je čas trvania testovacieho impulzu úrovne log. 1.



Obr. 2 Pripojenie akčného člena na dva výstupy

Fig. 2 The actuator connection to two outputs

Ak pripojené akčné členy nevyhovujú podmienke (3), nemôžno z dôvodu bezpečnosti tento typ testovania použiť, pretože by vplyvom testovacích impulzov dochádzalo k ovplyvneniu akčných členov v stave, keď má byť na výstupe modulu úroveň log. 0.

## 2. Aplikačná diagnostika

### 2.1 Vplyv aplikačnej diagnostiky na bezpečnosť SRCS

Na obr. 3 je znázornené zapojenie výstupnej časti SRCS s dvoma akčnými členmi. Toto zapojenie nekladie špeciálne požiadavky na bezpečnostné vlastnosti akčných členov a predpokladá bezpečné odpájanie riadeného objektu RO od zdroja energie (vzhľadom na sériové radenie akčných členov) [2]. Možno ho použiť vtedy, ak nie je k dispozícii akčný člen s vhodnými bezpečnostnými vlastnosťami.

Pre intenzitu nebezpečných porúch výstupnej časti SRCS s dvoma akčnými členmi (obr. 3) platí, že

$$\lambda_{HO}^{2A}(t) = \frac{dP_{HA}(t)}{1 - P_{HA}(t)} + \lambda_{HDO}, \quad (4)$$

kde  $P_{HA}(t)$  je pravdepodobnosť nebezpečnej poruchy dvojice akčných členov  $A_L$  a  $A_R$ ,  $\lambda_{HDO}$  je intenzita nebezpečných porúch F-DO modulu.

Pravdepodobnosť nebezpečnej poruchy dvojice navzájom nezávislých akčných členov  $A_L$  a  $A_R$  možno vyjadriť vzťahom:

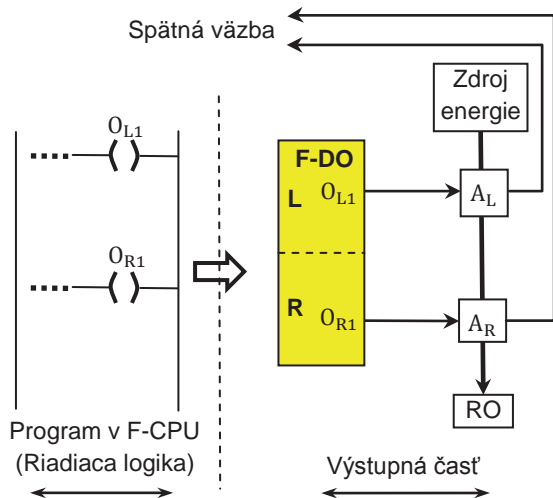
$$P_{HA}(t) \leq P_{AL}(t) \cdot P_{AR}(t), \quad (5)$$

kde  $P_{AL}(t)$ , resp.  $P_{AR}(t)$  je pravdepodobnosť poruchového stavu akčného člena  $A_L$ , resp.  $A_R$ .

Hodnota pravdepodobnosti  $P_{HA}(t)$  závisí od času detekcie a negovania poruchy akčných členov (čas negovania poruchy je vzhľadom na čas detekcie poruchy spravidla zane-



dbateľný). Detekcia poruchy akčných členov môže byť realizovaná funkčnou alebo testovacou aplikačnou diagnostikou. V prípade realizácie funkčnej diagnostiky možno za čas detekcie poruchy považovať maximálny čas medzi prevádzkovými poveleniami na zmenu stavu akčných členov. V prípade realizácie testovacej diagnostiky možno za čas detekcie poruchy považovať maximálny čas medzi vykonaním testovacích procedúr.



Obr. 3 Dvojkanálové pripojenie akčných členov k F-DO modulu

Fig. 3 Two-channel connection of actuators to F-DO module

Realizácia funkčnej aj testovacej diagnostiky vyžaduje snímanie stavu akčných členov (spätná väzba na obr. 3). Snímanie stavu akčných členov môže byť riešené priamo (ak to akčný člen umožňuje) alebo nepriamo (doplnením vhodných snímačov). V niektorých prípadoch (vzhľadom na požadovanú SIL a parametre použitých akčných členov) nemusí byť diagnostika nevyhnutná a čas detekcie poruchy možno stotožniť s časom medzi preventívnymi kontrolami, prípadne s užitočným životom SRCS.

Nech pre ďalšie úvahy platí, že:

- čas detekcie a negovania poruchy akčného člena je  $t_{0A}$ ;
- intenzity náhodných porúch akčných členov sú konštantné (exponenciálne rozdelenie výskytu porúch).

Potom pravdepodobnosť nebezpečnej poruchy dvojice akčných členov  $A_L$  a  $A_R$  možno vyjadriť vzťahom:

$$P_{HA}(t_{0A}) \leq (1 - e^{-\lambda_{AL} \cdot t_{0A}}) \cdot (1 - e^{-\lambda_{AR} \cdot t_{0A}}), \quad (6)$$

kde  $\lambda_{AL}$ , resp.  $\lambda_{AR}$  je intenzita náhodných porúch akčného člena  $A_L$ , resp.  $A_R$ .

Ak  $\lambda \cdot t \ll 1$ , tak možno intenzitu nebezpečných porúch dvojice akčných členov  $A_L$  a  $A_R$  stanoviť podľa vzťahu:

$$\lambda_{HALR}(t_{0A}) \cong 2 \cdot \lambda_{AL} \cdot \lambda_{AR} \cdot t_{0A}. \quad (7)$$

Intenzitu nebezpečných porúch výstupnej časti SRCS s dvoma akčnými členmi (F-DO modul a akčné členy  $A_L$  a  $A_R$ ) možno určiť podľa vzťahu:

$$\lambda_{HO}^{2A}(t_{0A}) \leq \lambda_{HDO} + \lambda_{HALR}(t_{0A}), \quad (8)$$

kde  $\lambda_{HDO}$  je intenzita nebezpečných porúch F-DO modulu.

### 3.2 Realizácia aplikačnej diagnostiky v safety PLC

Zo vzťahov (6), (7) a (8) je zrejmé, že intenzita nebezpečných porúch výstupnej časti SRCS so safety PLC je závislá od času detekcie a negovania poruchy akčného člena je  $t_{0A}$ .

Čas detekcie a negovania poruchy akčného člena je závislý od parametrov aplikačnej diagnostiky. Aplikačná diagnostika sa realizuje v aplikačnom programe safety PLC (safety program) a jej možnosti sú limitované vlastnosťami safety PLC vyplývajúcimi zo spôsobu spracovania aplikačného programu a vstupných a výstupných signálov safety PLC. Vlastnosti safety PLC majú obmedzujúci vplyv predovšetkým na čas trvania testovacích impulzov, na schopnosť safety PLC zachytiť zmenu stavu akčného člena vyvolanú testovacím impulzom a tiež majú vplyv na čas medzi testovacími procedúrami.

#### 3.2.1 Spracovanie safety programu

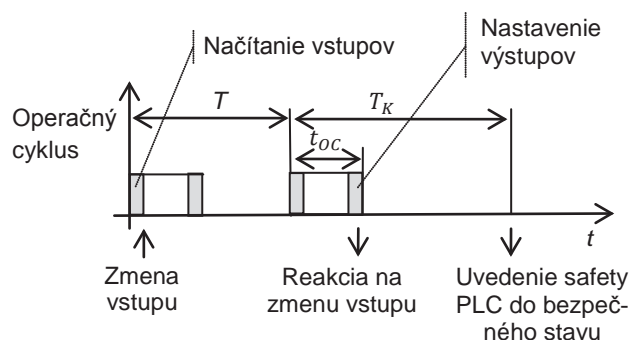
Spracovanie safety programu je, podobne ako aj spracovanie štandardného programu, založené na operačnom cykle (najpoužívanejším programovacím jazykom pri safety PLC je jazyk LD (Ladder Diagram); štandardné PLC majú spravidla širšie možnosti). V štandardných PLC zvyčajne existuje jedna programová úloha (tzv. kontinuálna úloha), ktorá sa spúšťa v operačnom cykle bezprostredne po ukončení predchádzajúceho operačného cyklu, pričom načítanie vstupov a aktualizácia výstupov sa môže (ale nemusí) viazať na operačný cyklus. Naopak, v safety PLC sa operačný cyklus vykonáva v definovanej perióde a načítanie vstupov a aktualizácia výstupov sa vždy viaže na operačný cyklus.

Periodické vykonávanie operačného cyklu ovplyvňuje aj čas odozvy safety PLC (pri realizácii aplikačnej diagnostiky to treba zohľadniť), preto musí byť zaistené, aby v každej perióde došlo k vykonaniu operačného cyklu. Nevykonanie operačného cyklu v stanovenej perióde musí byť detegované a musí byť vyvolaná bezpečná reakcia.

Mnoho safety PLC používa na detekciu chyby cyklického vykonávania operačného cyklu kontrolný časovač. Tento časovač je parametrizovateľný a musí platiť, že

$$t_{0C} < T < T_K, \quad (9)$$

kde  $t_{0C}$  je čas vykonávania operačného cyklu,  $T$  je perióda volania operačného cyklu a  $T_K$  je čas kontrolného časovača. Ak do času  $T_K$  nedôjde k volaniu operačného cyklu, kontrolný časovač vyvolá bezpečnú reakciu safety PLC (nastavenie výstupov do stavu log. 0; situáciu znázorňuje obr. 4).



Obr. 4 Vykonávanie operačného cyklu safety PLC

Fig. 4 Implementation of the operational cycle of safety PLC

Vzhľadom na spôsob spracovania vstupov a výstupov má perióda volania operačného cyklu vplyv na čas odozvy safety PLC [4]. Niekedy sa používa zjednodušený predpoklad, že

$$t_{sPLC} \cong T, \quad (10)$$

kde  $t_{sPLC}$  je čas odozvy safety PLC a  $T$  je perióda volania operačného cyklu. Tento predpoklad vychádza z úvahy, že zmena vstupu bude v najnepriaznivejšom prípade spraco-

vaná až pri ďalšom vykonávaní operačného cyklu (situáciu znázorňuje obr. 4).

Vzťah (10) možno akceptovať len za predpokladu, že platí

$$t_I + t_O \ll t_{sPLC} \quad (11)$$

a zároveň je garantované vykonanie operačného cyklu v perióde  $T$ . Vo vzťahu (11) je  $t_I$  čas odozvy F-DI modulu a  $t_O$  je čas odozvy F-DO modulu. Ak na periodické vykonávanie operačného cyklu dohliada kontrolný časovač, tak je z pohľadu bezpečnosti akceptovateľné tvrdenie, že safety PLC zareaguje na zmenu vstupu v najnepriaznivejšom prípade (pri chybe volania operačného cyklu) po čase  $T_K$  a nie po čase  $T$ . Preto z pohľadu realizácie bezpečnostných funkcií pre čas odozvy safety PLC (za predpokladu lokálnych vstupov a výstupov) platí:

$$t_{sPLC} = t_I + t_O + T_K \quad (12)$$

Ak sú vstupy a výstupy safety PLC pripojené po komunikačnej zbernici, treba to vo vzťahu (12) zohľadniť. Podrobnejšie sa problematike komunikácie venuje napr. literatúra [5], [6].

### 3.2.2 Možnosti aplikačnej diagnostiky

Čas trvania testovacích impulzov, ktorú možno na safety PLC realizovať, vyplýva zo spôsobu spracovania vstupov a výstupov v operačnom cykle a je daný vzťahom:

$$t_{DT} = k_D \cdot T, \quad (13)$$

kde  $t_{DT}$  je čas trvania testovacích impulzov,  $k_D$  je celé kladné číslo a  $T$  je perióda volania operačného cyklu.

Ak safety PLC používa na detekciu chyby cyklického vykonávania operačného cyklu kontrolný časovač, tak pri analýze vplyvu testovacích impulzov na akčné členy (a následne aj na riadený objekt) treba zohľadniť, že môže dôjsť k predĺženiu času trvania testovacieho impulzu. V najnepriaznivejšom prípade (za najnepriaznivejší prípad treba považovať najdlhší testovací impulz) môže byť čas trvania testovacieho impulzu

$$t_{DT} = k_D \cdot T_K, \quad (14)$$

kde  $T_K$  je čas kontrolného časovača.

Čas trvania testovacích impulzov možno nastaviť predovšetkým vhodnou voľbou čísla  $k_D$  (v aplikačnom softvéri), pretože časy  $T$  a  $T_K$  musia zohľadňovať aj iné skutočnosti. Číslo  $k_D$  treba zvoliť tak, aby:

- čas aktivácie akčného člena v čase, keď nemá byť riadený objekt pripojený na zdroj energie, bol dostatočne dlhý na vyhodnotenie stavu akčného člena riadiacou logikou SRCS, ale dostatočne krátky na to, aby riadený objekt nezareagoval na pripojenie energie;
- čas deaktivácie akčného člena v čase, keď má byť riadený objekt pripojený na zdroj energie, bol dostatočne dlhý na vyhodnotenie stavu akčného člena riadiacou logikou SRCS, ale dostatočne krátky na to, aby riadený objekt nezareagoval na odpojenie energie.

Súčasne s týmito podmienkami treba zohľadniť, že pre čas potrebný na vyhodnotenie stavu akčného člena v aplikačnom programe (čas  $t_V$ ; čas medzi vydaním povelu na zmenu stavu akčného člena a „zачytením“ tejto zmeny aplikačným programom) platí, že

$$t_V \leq t_O + t_A + t_{S_A} + t_{min} + t_N + T_K, \quad (15)$$

kde  $t_O$  je čas odozvy F-DO modulu,  $t_A$  je čas odozvy akčného člena,  $t_{S_A}$  je čas odozvy snímača stavu akčného člena,  $t_{min}$  je minimálny čas trvania signálu na vstupe modulu potrebný na spracovanie signálu modulom a  $t_N$  je povolený čas nezhody (parameter F-DI modulu).

Vzťah (15) platí za predpokladu, že F-DI a F-DO modul sú lokálne moduly safety PLC. V prípade, že tieto moduly budú pripojené cez komunikačnú zbernici, treba vo vzťahu (15) zohľadniť aj čas prenosu po komunikačnej zbernici.

Ďalším dôležitým parametrom aplikačnej diagnostiky je čas medzi testovacími procedúrami. Za predpokladu, že čas negovania poruchy je omnoho menší ako čas jej detekcie (tento predpoklad je spravidla splnený), možno čas medzi testovacími procedúrami stotožniť s časom detekcie a negovania poruchy akčného člena  $t_{0A}$ .

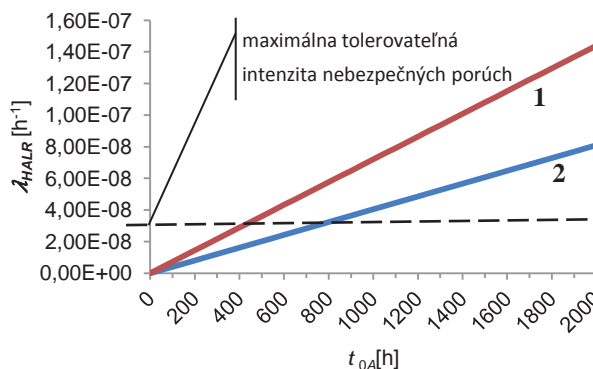
Limitujúcim faktorom pre minimálny čas medzi testovacími procedúrami je dĺžka trvania testovacích impulzov (vzťahy (13) a (14)) a čas vyhodnotenia stavu akčného člena (vzťah (15)), pretože k vykonaniu testovacej procedúry môže dôjsť až po ukončení predchádzajúcej testovacej procedúry.

Spúšťanie testovacích procedúr na safety PLC môže byť viazané buď na reálny čas, alebo na násobky času vykonávania operačného cyklu. Je výhodné, ak sú testovacie procedúry realizované na viacúlohovom safety PLC, pretože v tomto prípade môže testovacie procedúry vykonávať samostatná bezpečnostne relevantná úloha.

## 4. Príklad vplyvu parametrov diagnostiky na integritu bezpečnosti akčných členov

Predpokladajme zapojenie dvojice akčných členov podľa obr. 3. Potom možno intenzitu nebezpečných porúch dvojice akčných členov  $A_L$  a  $A_R$  stanoviť podľa vzťahu (7). Na obr. 5 je znázornená závislosť tejto intenzity od času detekcie a negovania poruchy akčného člena  $t_{0A}$ . Prvý priebeh zodpovedá hodnotám pre  $\lambda_{AL} = \lambda_{AR} = 6 \cdot 10^{-6} \text{ h}^{-1}$  (akčné členy s horšími ukazovateľmi bezporuchovosti) a druhý priebeh pre  $\lambda_{AL} = \lambda_{AR} = 4,5 \cdot 10^{-6} \text{ h}^{-1}$  (akčné členy s lepšími ukazovateľmi bezporuchovosti).

Na obr. 5 je znázornená aj maximálna tolerovateľná intenzita nebezpečných porúch dvojice akčných členov (možno ju stanoviť na základe maximálnej tolerovateľnej intenzity nebezpečných porúch SRCS a intenzít nebezpečných porúch jednotlivých komponentov SRCS). Z priebehov vidieť, že dosiahnutie dostatočne nízkej intenzity nebezpečných porúch dvojice akčných členov možno zaistiť skrátením času detekcie a negovania poruchy akčného člena  $t_{0A}$ . Možno povedať, že pri danej bezporuchovosti akčných členov bude minimálna dosiahnuteľná intenzita nebezpečných porúch dvojice akčných členov závisieť od parametrov safety PLC (parametre safety PLC sú pri skracovaní času  $t_{0A}$  limitujúce).



Obr. 5 Intenzita nebezpečných porúch dvojice akčných členov

Fig. 5 The dangerous failures rate of two actuators

Ak vzhľadom na parametre použitého safety PLC nemožno dosiahnuť požadované parametre aplikačnej diagnostiky, tak možno:

- použiť akčné členy s lepšími ukazovateľmi bezporucho-  
vosti, čím sa predĺži maximálny časový interval, počas  
ktorého musí byť overená schopnosť akčných členov  
zmeniť svoj stav, resp. použiť jeden akčný člen, ale so  
zodpovedajúcou SIL vzhľadom na realizovanú bezpeč-  
nostnú funkciu [1];
- zaisťiť test organizačnými opatreniami (preventívnou  
údržbou), čo však nemusí byť akceptovateľné z hľadiska  
prevádzky.

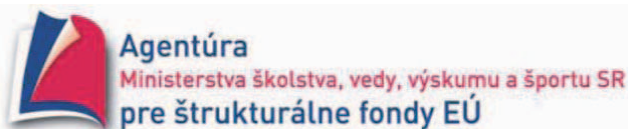
## Záver

Aplikačná diagnostika safety PLC je v niektorých prípadoch nevyhnutná na dosiahnutie požadovanej úrovne bezpečnosti SRCS so safety PLC. Realizácia diagnostiky (predovšetkým testovacej) si spravidla vyžaduje prispôbenie parametrov internej testovacej diagnostiky safety PLC. Vo všeobecnosti možno povedať, že čím má safety PLC menej nastaviteľných parametrov vzťahujúcich sa na diagnostiku, tým je jeho použitie na realizáciu SRCS vyžadujúceho aplikačnú diagnostiku problematickejšie.

## PodĎakovanie

*Táto 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] ŽDÁNSKY, J., RÁSTOČNÝ, K., HRBČEK, J.: Safety related control systems for railway signalling applications with a safety PLC. In: Archives of transport system telematics. Vol. 4, Iss. 3, pp. 43-49, ISSN 1899-8208, 2011.
- [2] ŽDÁNSKY, J., NAGY, P.: Influence of the control system structure with safety PLC on its reliability and safety. In: Proceedings of the 9<sup>th</sup> international conference ELEKTRO 2012. Rajecské Teplice, IEEE Catalog Number: CFP1248S-ART, pp. TA4\_25, ISBN 978-1-4673-1178-6, 2012.
- [3] HRBČEK, J., ŠIMÁK, V.: Implementation of Multi-dimensional Model Predictive Control for Critical Process

with Stochastic Behavior, chapter in: Advanced Model Predictive Control, pp.109-124, InTech, Tao Zheng (Ed.), ISBN 978-953-307-298-2, 2011.

[4] RÁSTOČNÝ, K., ŽDÁNSKY, J.: Specificities of safety PLC based implementation of the safety function. In: Proceedings of international conference applied electronics. AE 2012, Pilsen, pp. 229-232, ISBN 978-80-261-0038-6, ISSN 1803-7232, 2012.

[5] BEZÁK, T., STRÉMY, M., HUSÁROVÁ, B.: Distributed control systems modelling using PROFInet CBA. In: Annals of DAAAM and Proceedings of DAAAM Symposium 2010, Vol. 21, No 1., ISSN 1726-9679.

[6] BÉLAI, I., DRAHOŠ, P.: The Industrial Communication Systems PROFIBUS and PROFInet. In: Applied Natural Sciences 2009 : International Conference. Trnava, Slovak Republic, pp. 329-336, ISBN 978-80-8105-127-2.

## Abstrakt

Safety PLCs (Programmable Logic Controller) are primarily used to implement the SRCS (Safety Related Control Systems) at the process level, therefore is necessary to consider also the sensors and the actuators as part of the SRCS. In this case, the diagnosis of their failures and early response to a detected failure can be necessary to achieve the required SIL (Safety Integrity Level). In the SRCS with safety PLC the detection of sensors and actuators failures can be solved in the application program. In this case, is necessary to take into account some specific properties of safety PLC. This article looks at some relations between parameters of safety PLC, solution possibilities of the diagnostics and safety integrity of SRCS with safety PLC.

**Ing. Juraj Ždánsky, PhD.**

University of Žilina  
Faculty of Electrical Engineering  
Department of Control and Information Systems  
Univezitná 8215/1  
010 26 Žilina, Slovak Republic  
Tel.: +421 41 513 3342  
E-mail: juraj.zdansky@fel.uniza.sk

**prof. Ing. Karol Rástočný, PhD.**

University of Žilina  
Faculty of Electrical Engineering  
Department of Control and Information Systems  
Univezitná 8215/1  
010 26 Žilina, Slovak Republic  
Tel.: +421 41 513 3320  
E-mail: karol.rastocny@fel.uniza.sk

# Špecifické požiadavky na softvér SRCS so safety PLC

Juraj Ždánsky, Karol Rástočný

## Abstrakt

Široké aplikačné možnosti safety PLC (Programmable Logic Controller) predstavujú nespornú výhodu pri realizácii SRCS (Safety Related Control Systems) so safety PLC. Na druhej strane však kladú väčšie nároky na optimalizáciu funkcií SRCS z rôznych hľadísk. Vzhľadom na to, že ide o bezpečnostné funkcie, je nevyhnutné postupovať tak, aby bola preukázateľným spôsobom zaistená minimálna požadovaná úroveň všetkých relevantných vlastností SRCS. Niektoré z týchto vlastností sa dajú vyjadriť kvantitatívne, niektoré kvalitatívne a ich minimálna úroveň musí byť stanovená vzhľadom na bezpečnosť riadenej aplikácie. Dosiachnutie mnohých sledovaných vlastností (predovšetkým z oblasti funkčnej bezpečnosti) je závislé od softvéru safety PLC, čomu sa venuje aj tento príspevok.

**Kľúčové slová:** SRCS, safety PLC, bezpečnostná funkcia, safety program, čas odzvy

## Úvod

Podobne ako aj pri iných programovateľných riadiacich systémoch, tak aj pri SRCS so safety PLC sú požadované funkcie zaistené programom. SRCS so safety PLC realizuje bezpečnostné funkcie a preto musia byť pri tvorbe programu dodržané určité postupy a opatrenia, ktorých rozsah závisí od požadovanej SIL. Niektoré z týchto postupov a opatrení sú závislé od konkrétneho typu safety PLC a príslušného vývojového softvérového prostredia a sú pre používateľa skryté, prípadne rozpoznateľné len vo veľmi obmedzenej miere (napríklad postupy súvisiace so zaistením integrity a dôverylosti programu a pod.). Niektoré postupy a opatrenia však môže ovplyvniť užívateľ. Vzhľadom na ich aplikačnú závislosť je však nevyhnutné poznať vlastnosti safety PLC a vhodne ich použiť vzhľadom na bezpečnosť konkrétnej aplikácie.

Pri programovateľných SRCS (takým je aj safety PLC) je realizácia bezpečnostnej funkcie (príp. funkcií) o to náročnejšia, že testovaním SRCS nemožno odhaliť všetky potenciálne nebezpečné stavy SRCS, ktoré môžu počas života systému vzniknúť. Preto treba pri realizácii safety programu (program pre safety PLC) vychádzať z takých predpokladov, ktoré akceptujú najnepriaznivejší prípad z pohľadu bezpečnosti vzhľadom na danú aplikáciu. Je vhodné, ak je program pre safety PLC vytvorený systematicky. Na tento účel možno použiť napríklad postupy založené na teórii konečných automatov [1].

## 1. Deterministické spracovanie safety programu

Deterministické spracovanie safety programu musí garantovať priamo výrobca safety PLC. Preto musia existovať presné pravidlá, na základe ktorých dochádza k spracovaniu safety programu. Tieto pravidlá treba aplikovať tak, aby bola požadovaná bezpečnostná funkcia vykonávaná správne nielen po funkčnej stránke, ale aj v časovej oblasti [2]. Preto sa treba zaoberať aj časovými nárokmi safety programu, spôsobom spracovania časovo závislých inštrukcií, ako aj

spôsobom spracovania safety programu (poradie spracovania jednotlivých inštrukcií a pod.).

### 1.1 Časové nároky safety programu

Periódou vykonávania safety programu je okrem iného závislá aj od času vykonávania operačného cyklu. Orientačne možno stanoviť čas vykonávania operačného cyklu ( $t_{OC}$ ) pomocou diagnostických nástrojov safety PLC po spustení safety programu. Týmto spôsobom sa však nedá stanoviť najnepriaznivejší prípad, ktorý môže byť nevyhnutný pre teoretický dôkaz bezpečnosti. Najnepriaznivejší prípad (za predpokladu, že safety program nemôže byť prerušený inou úlohou; tento predpoklad je spravidla splnený) je daný vzťahom

$$t_{OC} = t_v + \sum_{i=1}^w t_{SI_i}, \quad (1)$$

kde  $t_v$  je čas vykonávania systémových funkcií súvisiacich so zaistením bezpečnosti (používateľ ich nemôže ovplyvniť),  $t_{SI_i}$  je čas vykonávania  $i$ -tej inštrukcie, prípadne funkčného bloku a  $w$  je počet inštrukcií a/alebo funkčných blokov použitých v safety programe.

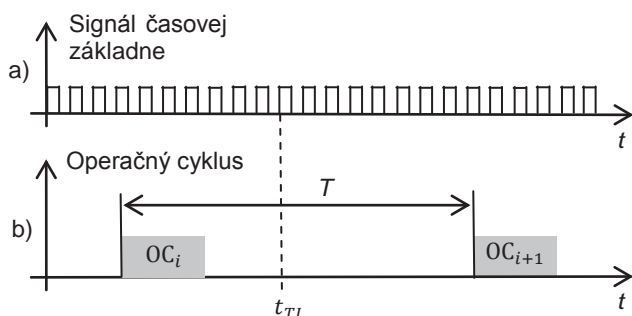
Čas  $t_{OC}$  je závislý aj od typu F-CPU, preto možno skrátenie času vykonávania safety programu dosiahnuť aj použitím výkonnejšieho procesora (pri rozsiahlejších bezpečnostne relevantných úlohách je skrátenie výraznejšie).

### 1.2 Spracovanie časovo závislých inštrukcií

Riadenie reálnych procesov si často vyžaduje realizáciu bezpečnostných funkcií obsahujúcich časové závislosti. Tieto sa realizujú pomocou časovo závislých inštrukcií. Sú to inštrukcie schopné merať čas a v závislosti od času meniť hodnoty vnútorných premenných, resp. hodnoty výstupov safety PLC.

Dôležitým faktorom, ktorý treba zohľadniť pri použití časovo závislých inštrukcií, je presnosť, s akou sú schopné merať čas. Bežne sa tento údaj spája s časovou základňou, ktorá generuje impulzy pre časovo závislé inštrukcie.

Periodické vykonávanie operačného cyklu, charakteristické pre safety PLC, do značnej miery ovplyvňuje aj spracovanie časovo závislých inštrukcií. Spôsob spracovania časovo závislých inštrukcií zachytáva obr. 1. Na obr. 1a) je znázornený signál časovej základne. Na obr. 1b) je znázornené vykonávanie safety programu (pre jednoduchosť uvažujeme s jednou bezpečnostne relevantnou úlohou). Predpokladáme, že safety program obsahuje časovo závislú inštrukciu vyhodnocujúcu dosiahnutie času  $t_{TI}$  od definovanej udalosti (vzhľadom na uvedený problém nie je podstatné, o akú udalosť ide). V čase, keď je vykonávaný  $i$ -tý operačný cyklus ( $OC_i$ ), je čas od definovanej udalosti menší ako čas  $t_{TI}$ . Čas  $t_{TI}$  od definovanej udalosti uplynie po ukončení  $i$ -tého operačného cyklu. V tomto okamihu nie je vyvolaná žiadna akcia, pretože safety program nie je práve vykonávaný. K vykonaniu požadovanej akcie dôjde až v nasledujúcom operačnom cykle ( $OC_{i+1}$ ).



Obr. 1 Spracovanie časovo závislých inštrukcií safety programom

Fig. 1 The processing time-dependent instructions in safety program

Z obr. 1 je zrejmé, že čas spracovania časovo závislých inštrukcií je viac ovplyvnený periódou volania operačného cyklu ako periódou časovej základne (tá je spravidla dostatočne malá). Preto treba uvažovať s tým, že

$$t_{RA} \in \langle t_{TI}, t_{TI} + T \rangle, \quad (2)$$

kde  $t_{RA}$  je reálny čas medzi definovanou udalosťou a vyvolaním požadovanej akcie,  $t_{TI}$  je čas nastavený v safety programe a vyhodnocovaný časovo závislou inštrukciou a  $T$  je perióda volania operačného cyklu.

Ak zmena času  $t_{RA}$  môže ovplyvniť bezpečnosť riadeného procesu (ak je vykonanie bezpečnej reakcie časovo závislé; napríklad v niektorých prípadoch treba jednotlivé časti technológie odpájať, resp. pripájať postupne v definovaných časových okamihoch), tak treba zohľadniť skutočnosť, že v mnohých safety PLC nie je čas  $T$  garantovaný, ale je strážený kontrolným časovačom. Potom v najnepriaznivejšom prípade (z pohľadu bezpečnosti) pre čas  $t_{RA}$  platí:

$$t_{RA} \in \langle t_{TI}, t_{TI} + T_K \rangle, \quad (3)$$

kde  $T_K$  je čas kontrolného časovača.

Bezpečná reakcia je teda za normálnych okolností vyvolaná v najbližšom operačnom cykle po uplynutí času  $t_{TI}$  (čas  $t_{RA}$  sa v tomto prípade môže pohybovať v intervale danom výrazom (2)), alebo v prípade nevykonania operačného cyklu do času  $T_K$  kontrolným časovačom (čas  $t_{RA}$  sa v tomto prípade môže pohybovať v intervale danom výrazom (3)).

### 1.3 Spôsob spracovania safety programu

Preferovanými jazykmi pre komerčne dostupné safety PLC sú jazyky založené na jazykoch LD (*ladder diagram*) a FBD (*function block diagram*). Stručná charakteristika jazykov LD a FBD je uvedená napr. v [3]. Modifikácie týchto jazykov určené na tvorbu safety programu sa môžu líšiť názvom

v závislosti od výrobcu a typu safety PLC (napríklad F-LD (*fail-safe ladder diagram*), F-FBD (*fail-safe function block diagram*)).

Procesor spracováva safety program v akomkoľvek jazyku sekvenčne. Základný rozdiel medzi jazykmi F-LD a F-FBD je v tom, že poradie spracovania jednotlivých inštrukcií safety programu je v jazyku F-LD dané jeho zápisom, naopak pri jazyku F-FBD musia byť použité dodatočné prostriedky na jednoznačné vyjadrenie poradia spracovania jednotlivých inštrukcií (napríklad každá inštrukcia má číslo, ktoré určuje poradie jej spracovania). Túto skutočnosť treba zohľadniť pri tvorbe safety programu, pretože zmena poradia spracovania jednotlivých inštrukcií v safety PLC (vizuálne poradie inštrukcií nemusí zodpovedať poradiu spracovania) môže viesť až k zmene funkcie SRCS.

## 2. Programovanie komunikácie

V niektorých prípadoch sa nemožno pri realizácii bezpečnostnej funkcie vyhnúť komunikácii medzi dvomi, resp. viacerými safety PLC. Dôvodom realizácie distribuovaného SRCS môže byť:

- rozsiahlosť riadeného procesu;
- sprehľadnenie safety programu;
- nepostačujúci výpočtový výkon jedného safety PLC;
- obmedzenie rozsahu výpadku SRCS v prípade poruchy.

Dôvodom realizácie distribuovaného SRCS spravidla býva kombinácia uvedených faktorov. Z hľadiska času reakcie bezpečnostných funkcií je ideálne, ak bezpečnostnú funkciu realizuje len jeden safety PLC. Vždy sa nedá vyhnúť prípadom, keď je časť bezpečnostnej funkcie realizovaná na jednom a časť na druhom, prípadne ďalších safety PLC. Predpokladom rozdelenia bezpečnostnej funkcie na dve, prípadne viac safety PLC je možnosť bezpečnej komunikácie medzi nimi.

Na dosiahnutie bezpečnej komunikácie s požadovanou SIL výrobcovia používajú bezpečnostnú nadstavbu štandardných protokolov. Bezpečnostná nadstavba sa pri safety PLC realizuje v aplikačnej vrstve pomocou programových funkcií certifikovaných na tento účel. Používateľ môže tieto programové funkcie používať v aplikačnej časti safety programu pomocou presne definovaných rozhraní (tieto programové funkcie musia byť zaistené tak, aby ich používateľ nemohol meniť). Spôsob ich použitia síce neohrozuje bezpečnosť komunikácie, ale môže výrazne ovplyvniť čas prenosu informácií a tým aj čas odozvy bezpečnostnej funkcie.

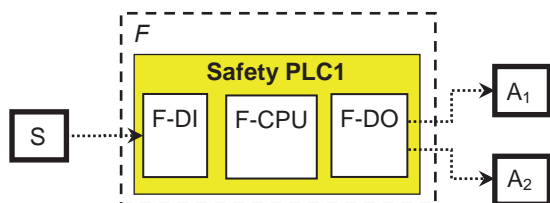
### 2.1 Lokálna realizácia bezpečnostnej funkcie

Na obr. 2 je znázornená realizácia jednoduchej bezpečnostnej funkcie  $F$ . SRCS je v tomto prípade tvorený jedným safety PLC, snímačom  $S$  a akčnými členmi  $A_1$  a  $A_2$ . Safety PLC monitoruje ovplyvnenie snímača  $S$ . V prípade jeho ovplyvnenia dôjde k ovplyvneniu akčného člena  $A_2$  a následne k ovplyvneniu akčného člena  $A_1$  (predpokladáme, že časovú následnosť ovplyvnenie akčných členov si vyžaduje riadený proces). Bezpečnostnú funkciu budeme považovať za vykonanú po ovplyvnení obidvoch akčných členov. V prípade, že akčné členy riadia sústavu ktorá je zotrvačná, tak treba uvažovať aj so zotrvačnosťou riadenej sústavy. V tomto príspevku však nebudeme uvažovať zotrvačnosť riadenej sústavy, pretože táto prispieje k času odozvy bezpečnostnej funkcie v akejkoľvek architektúre SRCS rovnako.

Čas odozvy bezpečnostnej funkcie realizovanej podľa obr. 2 možno vyjadriť vzťahom:

$$t_F = t_S + t_{SPLC} + t_{A1} + t_{A2}, \quad (4)$$

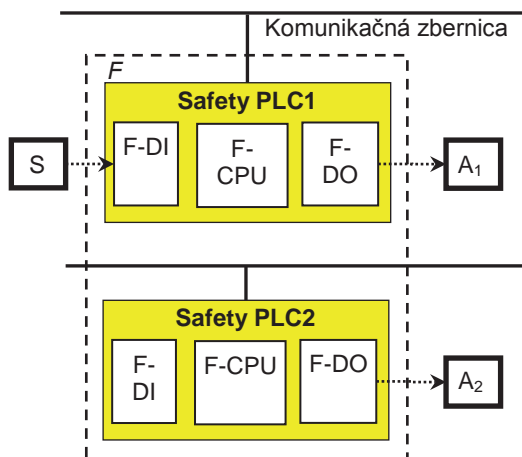
kde  $t_S$  je čas odozvy snímača S (katalógový údaj),  $t_{sPLC}$  je čas odozvy safety PLC (vzťah (12) v príspevku „Vplyv diagnostiky na integritu bezpečnosti SRCS so safety PLC“),  $t_{A1}$  je čas odozvy aktuátora  $A_1$  (katalógový údaj) a  $t_{A2}$  je čas odozvy aktuátora  $A_2$  (katalógový údaj).



Obr. 2 Realizácia bezpečnostnej funkcie  
Fig. 2 The realization of safety function

## 2.2 Distribuovaná realizácia bezpečnostnej funkcie

Na obr. 3 je znázornená realizácia bezpečnostnej funkcie F pomocou dvoch navzájom komunikujúcich safety PLC, snímača S a akčných členov  $A_1$  a  $A_2$ . Safety PLC1 monitoruje ovplyvnenie snímača S. V prípade jeho ovplyvnenia dôjde k odoslaniu tejto informácie na safety PLC2, kde sa táto informácia spracuje a na jej základe dôjde k ovplyvneniu akčného člena  $A_2$  a následnému odoslaniu informácie o ovplyvnení akčného člena  $A_2$  do safety PLC1. Safety PLC1 po prijatí informácie zo safety PLC2 ovplyvní akčný člen  $A_1$ . Bezpečnostnú funkciu budeme považovať za vykonanú po ovplyvnení oboch akčných členov.



Obr. 3 Realizácia bezpečnostnej funkcie pomocou dvoch safety PLC

Fig. 3 The realization of safety function using two safety PLC

Čas odozvy bezpečnostnej funkcie realizovanej podľa obr. 3 možno vyjadriť vzťahom:

$$t_F = t_S + t_{sPLC} + t_{A1} + t_{A2} + 2 \cdot t_K, \quad (5)$$

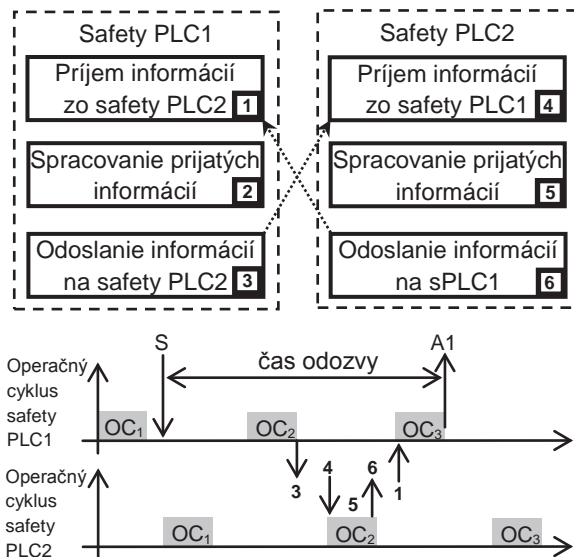
kde  $t_K$  je čas prenosu informácie medzi safety PLC1 a safety PLC 2.

Vo vzťahu (5) je uvažovaný dvojnásobok času  $t_K$ , pretože funkcia SRCS si vyžaduje prenos informácie zo safety PLC1 na safety PLC2 a naspäť.

Čas  $t_K$  je závislý od mnohých faktorov, z pohľadu programovania safety PLC však významnou mierou na čas  $t_K$  vplyva aj usporiadanie safety programu.

Vplyv usporiadania safety programu na čas prenosu informácií znázorňuje obr. 4 a obr. 5. V oboch prípadoch je znázornené principiálne usporiadanie safety programu s komunikačnými funkciami a časové diagramy prenosu informácie medzi safety PLC1 a safety PLC2.

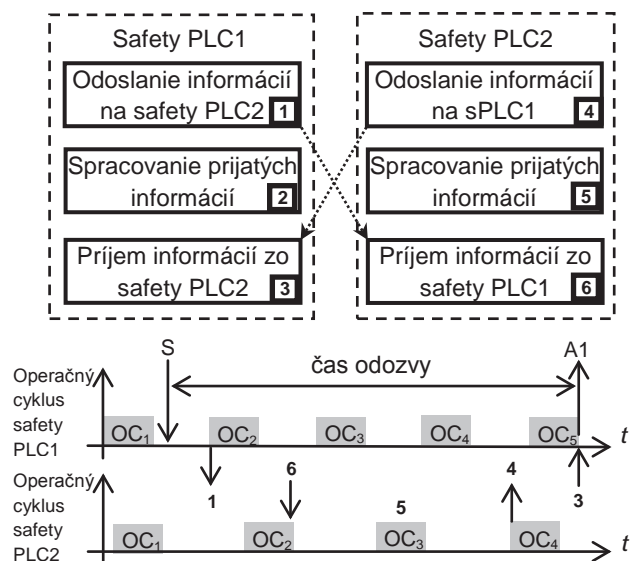
Usporiadanie safety programu podľa obr. 4 predstavuje optimálny spôsob usporiadania vzhľadom na čas prenosu informácií. Informácia o ovplyvnení snímača S sa odošle na safety PLC2 vykonaním safety programu v najbližšom operačnom cykle po tejto udalosti (OC<sub>2</sub> na safety PLC1). Tu je informácia prijatá, spracovaná a odoslaná späť na safety PLC1 v rámci jedného operačného cyklu (OC<sub>2</sub> na safety PLC2). Safety PLC1 po jej prijatí (v operačnom cykle OC<sub>3</sub>) ovplyvní akčný člen  $A_1$ .



Obr. 4 Optimálne usporiadanie safety programu vzhľadom na čas prenosu informácií

Fig. 4 The optimal arrangement of safety program considering the information transfer time

Usporiadanie safety programu podľa obr. 5 znázorňuje prenos informácií pri opačnom poradí prijímania a odosielania. Informácia o ovplyvnení snímača S je, podobne ako na obr. 4, odoslaná vykonaním safety programu v najbližšom operačnom cykle po tejto udalosti (OC<sub>2</sub> na safety PLC1). Na safety PLC2 dôjde k jej prijatiu až na konci operačného cyklu OC<sub>2</sub>. Z tohto dôvodu môže byť spracovaná až v ďalšom operačnom cykle (OC<sub>3</sub> na safety PLC2) a až v nasledujúcom operačnom cykle (OC<sub>4</sub> na safety PLC2) môže dôjsť k jej odoslaniu. Safety PLC1 po jej prijatí (v operačnom cykle OC<sub>5</sub>) ovplyvní akčný člen  $A_1$ .



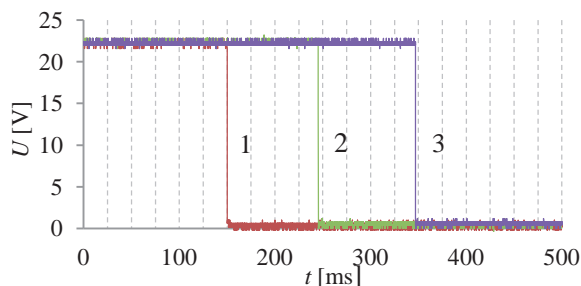
Obr. 5 Nevhodné usporiadanie safety programu vzhľadom na čas prenosu informácií

Fig. 5 The inappropriate arrangement of safety program considering the information transfer time

Cieľom obr. 4 a obr. 5 je poukázať na významný vplyv usporiadania komunikačných funkcií v safety programe na čas odozvy bezpečnostnej funkcie. Časové diagramy na obr. 4 a obr. 5 nezachytávajú najnepriaznivejší prípad. Najnepriaznivejší prípad času odozvy je závislý nielen od usporiadania safety programu, ale aj od

- periódy vykonávania operačného cyklu na safety PLC1 a safety PLC2;
- času kontrolných časovačov na safety PLC1 a safety PLC2;
- fázového posunu medzi vykonávaním operačných cyklov na safety PLC1 a safety PLC2 (fázový posun môže byť premenný v čase);
- rozsiahlosti safety programov;
- prenosovej rýchlosti použitej komunikačnej zbernice.

Na obr. 6 je znázornená reálna časová odozva bezpečnostnej funkcie  $F$ . Priebeh 1 zachytáva ovplyvnenie snímača  $S$  a priebeh 3 zachytáva reakciu akčného člena  $A_1$  (za predpokladu usporiadania safety programu podľa obr. 4). Kvôli porovnaniu je na obr. 6 uvedený aj priebeh 2 zachytávajúci reakciu akčného člena  $A_1$  v prípade, že bezpečnostná funkcia  $F$  je realizovaná na jednom safety PLC. Bezpečnostná funkcia bola v oboch prípadoch realizovaná na safety PLC Simatic S7-300F (CPU 315F-2DP).



Obr. 6 Čas odozvy bezpečnostnej funkcie

Fig. 6 The response time of safety function

Čas odozvy sa môže ešte výraznejšie predĺžiť, ak bude bezpečnostná funkcia realizovaná na viac ako dvoch safety PLC. Obmedzenie komunikácie medzi safety PLC možno dosiahnuť napríklad pridaním hardvérových modulov (opätovné načítanie informácie priamo z procesu môže byť rýchlejšie ako jej prenos z iného safety PLC). Treba si uvedomiť, že takéto riešenie môže ovplyvniť architektúru realizovaného SRCS, čo môže mať vplyv na jeho technickú bezpečnosť (vplyvu architektúry SRCS na jeho bezpečnosť sa venuje literatúra [4] a vplyvu obnovy a údržby literatúra [5] a [6]). Preto je nevyhnutné už pri návrhu SRCS uvažovať aj s požadovaným časom odozvy pre jednotlivé bezpečnostné funkcie a prispôbiť tomu aj architektúru SRCS so safety PLC.

## Záver

Štruktúra safety programu, spôsob jeho tvorby ako aj nastavenie súvisiacich parametrov môžu významne ovplyvniť nielen časové relácie vykonávaných bezpečnostných funkcií, ale aj samotnú funkciu SRCS. Táto skutočnosť je o to závažnejšia, že nie všetky vlastnosti SRCS sa nedajú overiť testovaním. Preto je dôležité pri realizácii bezpečnostných funkcií zohľadniť nielen základné princípy tvorby programu na safety PLC, ale aj špecifické vlastnosti konkrétneho typu safety PLC.

## PodĎakovanie

This paper was supported by the scientific grant agency VEGA, grant No. VEGA-1/0388/12 "Quantitative safety

integrity level evaluation of control systems in railway application".

## Literatúra

- [1] RÁSTOČNÝ, K., ŽDÁNSKY, J.: Použitie konečného automatu pri programovaní PLC. In: Advances in electrical and electronic engineering. Vol. 3/2004, No.1, pp. 45-49, ISSN 1336-1376, 2004.
- [2] RÁSTOČNÝ, K., ŽDÁNSKY, J.: Specificities of safety PLC based implementation of the safety function. In: Proceedings of international conference applied electronics. AE 2012, Pilsen, pp. 229-232, ISBN 978-80-261-0038-6, ISSN 1803-7232, 2012.
- [3] RÁSTOČNÝ, K., ŽDÁNSKY, J.: Riadiace systémy so safety PLC. EDIS - vydavateľstvo ŽU, Žilina, ISBN 978-80-554-0681-7, 2013.
- [4] ŽDÁNSKY, J., NAGY, P.: Influence of the control system structure with safety PLC on its reliability and safety. In: Proceedings of the 9<sup>th</sup> international conference ELEKTRO 2012. Rajecké Teplice, IEEE Catalog Number: CFP1248S-ART, pp. TA4\_25, ISBN 978-1-4673-1178-6, 2012.
- [5] ILAVSKÝ, J., RÁSTOČNÝ, K.: Considerations of the recovery in 2-out-of-3 safety-related control system. In: Proceedings of 11<sup>th</sup> IFAC/IEEE international conference on programmable devices and embedded systems. PDeS 2012, Brno, pp. 132-137, ISBN: 978-3-902823-21-2, 2012.
- [6] RÁSTOČNÝ, K., ILAVSKÝ, J.: Effects of a periodic maintenance on the safety integrity level of a control system. In: Proceedings of the 8<sup>th</sup> symposium on formal methods for automation and safety in railway and automotive systems. FORMS/FORAMAT 2010, Braunschweig. pp. 77-85, Springer-Verlag, Berlin, ISBN 978-3-642-14260-42011, 2010.

## Abstrakt

Wide application possibilities of safety PLC (Programmable Logic Controller) represent undisputed advantage in implementation of the SRCS (Safety Related Control Systems) with safety PLC. On the other hand, they places greater claims for optimization of the SRCS functions from different viewpoints. Due to the fact that these are safety functions, it is necessary to proceed in a provable way to ensure a minimum required level of all relevant properties of the SRCS. Some of these properties can be expressed quantitative, some qualitative and their minimum level must be determined considering the safety of the control application. The achievement of many observed properties (especially in the field of functional safety) is dependent on the safety PLC software, what is also content of this paper.

## Ing. Juraj Ždánsky, PhD.

University of Žilina, Faculty of Electrical Engineering  
Department of Control and Information Systems  
Univezitná 8215/1, 010 26 Žilina, Slovak Republic  
Tel.: +421 41 513 3342  
E-mail: juraj.zdansky@fel.uniza.sk

## prof. Ing. Karol Rástočný, PhD.

University of Žilina, Faculty of Electrical Engineering  
Department of Control and Information Systems  
Univezitná 8215/1, 010 26 Žilina, Slovak Republic  
Tel.: +421 41 513 3320  
E-mail: karol.rastocny@fel.uniza.sk

# Optimalizácia plánu broadcast vysielania v bezdrôtových senzorových sieťach

Peter Holečko

## Abstrakt

Príspevok sa zaoberá problematikou optimalizácie riadenia prístupu k rádiovému prenosovému kanálu metódou TDMA (Time Division Multiple Access) v rámci bezdrôtových senzorových sietí tvoriacich súčasť procesnej úrovne riadenia za predpokladu neuniformných požiadaviek na prístup. Primárne optimalizačné kritériá (minimalizácia dĺžky TDMA rámca a maximalizácia jeho využitia) sú vyšetřované použitím heuristického algoritmu.

**Kľúčové slová:** bezdrôtové senzorové siete, plán broadcast vysielania, optimalizácia

## Úvod

Technológie snímania a riadenia majú veľký potenciál rastu nielen vo vede a riadení, ale i v širokom spektre aplikácií súvisiacich so ochranou a bezpečnosťou kritickej infraštruktúry, energetikou, spracovaním tovarov, dopravou, zdravotníctvom, environmentalistikou a ďalšími oblastami.

Senzorová sieť je infraštruktúra skladajúca sa zo snímacích (meracích), výpočtových a komunikačných prvkov poskytujúcich možnosť merania, monitorovania a reakcie na udalosti a javy v špecifikovanom prostredí.

K základným komponentom sensorovej siete patrí sústava distribuovaných alebo lokalizovaných senzorov, prepojovacia sieť (obvykle bezdrôtová), centrálny uzol zoskupujúci informácie a sústava výpočtových prostriedkov majúcich za úlohu monitorovanie stavov, koreláciu dát, sledovanie trendov a dolovanie dát. Komunikačná a výpočtová infraštruktúra senzorových sietí je často špecifická v závislosti od prostredia ich nasadenia a aplikácie.

Z dôvodu distribuovaného charakteru bezdrôtových senzorových sietí vystupuje do popredia problém optimalizácie komunikačných procedúr medzi jednotlivými jej prvkami.

## Problém časovania broadcast vysielania

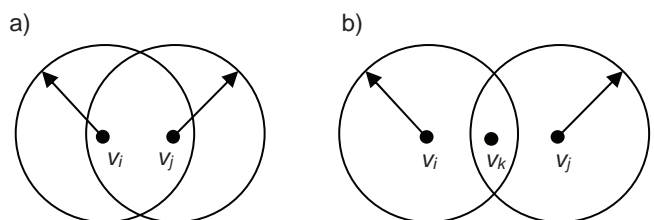
Bezdrôtová sieť môže byť reprezentovaná grafom  $G=(V,E)$ , kde  $V = \{v_1, v_2, \dots, v_i, \dots, v_N\}$  je množina uzlov, pričom  $N$  je počet uzlov a  $E = \{e_1, e_2, \dots, e_L\}$  je množina neorientovaných hrán, kde  $L$  je počet neorientovaných hrán [1]. Existencia hrany medzi dvoma uzlami znamená, že obidva uzly môžu priamo navzájom prijímať pakety vysielané druhým uzlom. Konektivita medzi sieťovými uzlami je vyjadrená symetrickou maticou  $C$  dimenzie  $N \times N$ , kde prvok

$$c_{ij} = \begin{cases} 1, & \text{ak } v_i \text{ a } v_j \text{ sú spojené} \\ 0, & \text{inak} \end{cases}$$

V prípade, že sú dva uzly priamo spojené, hovoríme, že sú vzdialené jeden skok. Predpokladajme, že čas rozdelený na sloty a paket s konštantnou dĺžkou, z čoho vyplýva, že vyslanie resp. prijatie jedného paketu trvá jeden časový slot. Rámec TDMA sa skladá z fixného počtu takýchto časových slotov. Pakety môžu byť rôznymi stanicami vysielané súčasne v tom istom časovom slotu len v tom prípade, ak nedochádza k ich vzájomnej interferencii. Ak je stanovená optimálna prenosová schéma TDMA rámca, rámec sa v čase opakuje. Označme takýto TDMA rámec maticou  $T$  rozmerov  $M \times N$ , ktorej prvky

$$t_{mj} = \begin{cases} 1, & \text{ak } v_j \text{ vysielala v časovom slotu } m \\ 0, & \text{inak} \end{cases}$$

Ak uzol  $v_i$  vysielala paket, žiadny z jeho susedov, t. j. uzlov vzdialených jeden skok, nemá dovolené vysielat' simultánne, čo by spôsobilo tzv. *primárny konflikt* [1], v niektorých prácach označovaný ako *priama kolízia* [4] (obr. 1a). Všetky uzly vzdialené dva skoky od uzla  $v_i$  taktiež nesmú vysielat' zároveň s  $v_i$ , keďže by to viedlo k tzv. *sekundárnemu konfliktu* označovanému aj *kolízia skrytej stanice* alebo *skrytá kolízia* (obr. 1b) z dôvodu viacnásobného príjmu na medziľahlých uzloch ( $v_k$ ).



Obr. 1 Znáznornenie a) primárneho a b) sekundárneho konfliktu

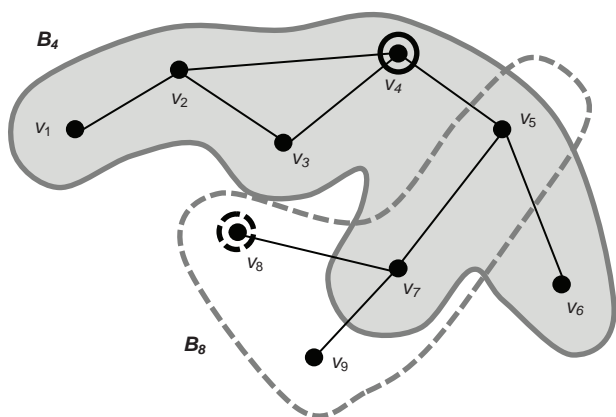
Fig. 1 Representation of a) primary and b) secondary conflict

Všetky tieto uzly vzdialené jeden alebo dva skoky od uzla  $v_i$  sú označované ako *broadcast zóna* [1] uzla  $v_i$ . Množinu týchto uzlov potom označujeme ako  $B_i$ . Je zrejmé, že



potreba eliminácie interferencií vyžaduje, aby žiadny z uzlov nachádzajúcich sa v  $B_i$  nevysielal simultánne s uzlom  $v_i$ . Obr. 2 ilustruje princíp tvorenia broadcast zóny uzla  $v_4$  (šedá oblasť  $B_4$ ) a uzla  $v_8$  (oblasť  $B_8$ ). Broadcast zóna uzla  $v_4$ , označená ako  $B_4$ , obsahuje všetky uzly, ktoré sú od  $v_4$  vzdialené jeden alebo dva skoky. To znamená, že vymedzuje množinu uzlov, pri ktorých súčasnom vysielaní by došlo ku primárnemu alebo sekundárnemu konfliktu. Na základe tohto konceptu je možné vytvoriť maticu kompatibility  $D$  rozmerov  $N \times N$ , ktorej prvky sú dané

$$d_{ij} = \begin{cases} 1, & \text{ak } v_j \in B_i \\ 0, & \text{inak} \end{cases}$$



Obr. 2 Princíp vytvorenia broadcast zón  
Fig. 2 Principle of broadcast zones

### Optimalizačné kritériá

Cieľom je nájsť najkratší TDMA cyklus spĺňajúci nasledovné podmienky:

- každý uzol musí mať počas jedného TDMA cyklu naplánované aspoň jedno vysielanie

$$\sum_{m=1}^M t_{mi} \geq 1 \quad \text{pre } \forall i \quad (1)$$

- za účelom vyhnutia sa primárnym konfliktom uzol nesmie vysielat' a prijímať pakety v tom istom časovom slotě,
- uzol nesmie prijímať dva alebo viac prenosov simultánne, čím sa eliminujú sekundárne konflikty.

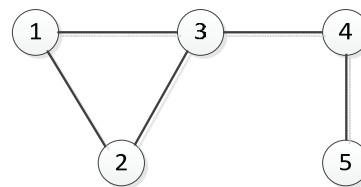
Formálne teda ak  $t_{mi} = 1$ , potom  $\sum_{j=1}^N t_{mj} d_{ij} = 0 \quad \forall m, i$

$$\text{t.j. } \sum_{m=1}^M \sum_{i=1}^N \sum_{j=1}^N t_{mi} t_{mj} d_{ij} = 0 \quad (2)$$

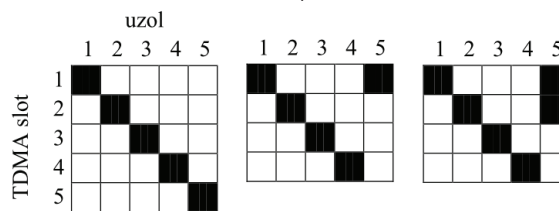
Posledným dvom uvedeným konfliktom je možné sa vyhnúť vytvorením TDMA rámca  $T$  so štruktúrou spĺňajúcou rovnicu (2). Triviálne riešenie spĺňajúce všetky tri podmienky predstavuje  $N$ -slotový TDMA rámec, kde  $N$  rôznych uzlov vysielajú v  $N$  rôznych časových slotoch (obr. 3). Hodnota  $N$  predstavuje zároveň horné ohraničenie dĺžky TDMA rámca.

Primárne optimalizačné kritérium je minimalizácia dĺžky TDMA cyklu, t. j.  $M$  by malo byť čo najmenšie. Jedným z ďalších možných optimalizačných cieľov je maximalizácia celkového počtu prenosov. Túto požiadavku je možné reprezentovať indexom využitia kanála  $\rho$ , pričom

$$\rho = \frac{1}{MN} \sum_{m=1}^M \sum_{j=1}^N t_{mj} \quad (3)$$



a)



b)

c)

d)

Obr. 3 Príklad a) topológie siete, b) triviálny TDMA rámec, c) a d) optimálne TDMA rámce

Fig. 3 Example of a) network topology, b) trivial TDMA frame, c) and d) optimal TDMA frames

Tretím ukazovateľom charakterizujúcim nájdený plán broadcast vysielania je priemerné oneskorenie vysielania každého uzla, označované ako  $\eta$  a definované nasledovným vzťahom

$$\eta = \frac{1}{N} \sum_{i=1}^N \left( \frac{M}{\sum_{j=1}^M t_{ij}} \right) = \frac{M}{N} \sum_{i=1}^N \left( \frac{1}{\sum_{j=1}^M t_{ij}} \right) \quad (4)$$

### Dolné ohraničenie dĺžky TDMA rámca

Pre dĺžku rámca  $M$  možno stanoviť dolné ohraničenie na základe niekoľkých viet. V [1] bol navrhnutý spôsob stanovenia dolného ohraničenia založený na stupňoch vrcholov grafu. Pre danú sieť  $G=(V,E)$  je stupeň daného vrcholu  $v \in V$ , označovaný ako  $\text{deg}(v)$ , počet hrán incidenčných s vrcholom  $v$ . Potom dĺžka rámca  $M$  spĺňa nasledujúcu nerovnosť:

$$M \geq \delta(G) + 1, \quad (5)$$

kde  $\delta(G) = \max_{v \in V} \text{deg}(v)$ .

### Neuniformné požiadavky na prenos

V doterajšom texte sa predpokladalo, že všetky uzly siete majú rovnaké požiadavky na broadcast vysielanie, konkrétne jedno vysielanie (jeden časový slot) v rámci TDMA rámca. Jedná sa teda o uniformné požiadavky na broadcast vysielanie.

Uvažujme ale prípad, kedy tieto požiadavky uniformné nie sú a pre každý uzol je definovaný počet požadovaných vysielaní v rámci. Tieto požiadavky môžu byť stanovené pomocou vektora požiadaviek na vysielanie  $P$  s  $N$  prvkami:

$$P = (p_1, p_2, \dots, p_N),$$

pričom  $p_i \in \mathbb{N}$ ,  $\forall i = 1, 2, \dots, N$ . Vzťah (1) potom prejde na

$$\sum_{m=1}^M t_{mi} \geq p_i \quad \text{pre } \forall i, \quad (6)$$

čo pre  $P = (1, 1, \dots, 1)$  dáva špeciálny prípad – uniformné prenosové požiadavky. Z uvedeného je zrejme, že musí platiť nerovnosť

$$\sum_{m=1}^M \sum_{i=1}^N t_{mi} \geq \sum_{i=1}^N p_i \quad (7)$$

V nasledujúcej časti budú doteraz formulované kritériá a požiadavky transformované do heuristického algoritmu hľadajúceho optimálne TDMA rámce. Implementácia bola realizovaná v programovom prostredí Matlab s využitím rozširujúcich knižnic funkcií.

## Navrhnutý heuristický algoritmus

Navrhnutý algoritmus pozostáva z nasledujúcich krokov a procedúr a vychádza z matematického aparátu naznačeného v predchádzajúcich častiach.

**Vstupné dáta.** Do algoritmu vstupuje hodnota  $N \in \mathbb{N}$  predstavujúca počet uzlov siete a vektor požiadaviek  $P = (p_1, p_2, \dots, p_N)$ ,  $p_i \in \mathbb{N}$ ,  $\forall i = 1, 2, \dots, N$ .

**Generovanie matice konektivity.** Blok generuje náhodnú štvorcovú symetrickú maticu konektivity (koincidenčnú maticu)  $C (N \times N)$ , reprezentujúcu topologický model siete uzlov resp. graf siete. Použitá je funkcia `sprandsym(n,density)`, ktorá vracia rozptýlenú symetrickú štvorcovú maticu rozmerov  $n \times n$  s približne  $density * n * n$  nenulovými prvkami s normálnym rozdelením ( $\mu = 0$ ,  $0 \leq \sigma^2 \leq 1$ ). Ďalšími operáciami je táto matica transformovaná na maticu prvkov 0 a 1.

**Testovanie a úprava súvislosti siete.** Časť algoritmu overuje, či je sieť vygenerovaná v predchádzajúcom kroku súvislá, to znamená, či jej graf má práve jeden komponent. Prípad nesúvislej siete s  $x$  komponentmi prechádza na  $x$  samostatných sietí. Použitá sú funkcie Sieť je preto po zistení nesúvislosti pomocou funkcie `make_connected` (z množiny `matlab_bgl`) upravená na súvislú.

**Testovanie planárnosti siete.** Časť algoritmu testuje, či je možné topológiu siete (grafu) získanú v predchádzajúcich krokoch zakresliť v rovine bez toho, aby sa jednotlivé hrany križovali, t.j. testuje jej planárnosť.

**Výpočet parametrov siete.** Blok počíta parametre finálnej siete (grafu) reprezentovanej prostredníctvom matice  $C$ . Týmito parametrami sú: stupeň všetkých uzlov  $\deg(v_i)$ , priemerný stupeň uzlov, maximálny stupeň uzlov  $G = \max(\deg(v_i))$ , počet hrán, dolné ohraničenie dĺžky TDMA rámca (5). Tieto parametre budú neskôr použité pri hodnotení optimality dosiahnutého výsledku.

**Generovanie grafickej reprezentácie siete.** Pre zlepšenie predstavy o štruktúre siete je implementovaný blok jej grafickej reprezentácie pomocou funkcie `graph_draw` z množiny `graphviz`.

**Výpočet matice kompatibility.** Hlavnou vstupnou štruktúrou do procesu určenia TDMA rámca je matica kompatibility  $D (N \times N)$  vyjadrujúca, ktoré uzly môžu v danom časovom slotе súčasne vysielat' bez toho, aby medzi nimi nastal primárny alebo sekundárny konflikt (obr. 1). Výsledkom je matica obsahujúca v danom riadku  $i$  hodnotu „1“ v stĺpcoch zodpovedajúcich uzlom, ktoré by boli v konflikte s uzlom  $i$ . V prípade neexistencie konfliktu má prvok hodnotu „0“.

**Hľadanie TDMA rámca.** Blok predstavuje samostatný heuristický algoritmus pre nájdenie optimálnej štruktúry TDMA rámca na základe zadaných kritérií.

**Výpočet indexu využitia kanála.** Blok počíta druhý z parametrov optimalizácie – index využitia kanála  $\rho$ , definovaný vzťahom (3).

**Výpočet priemerného oneskorenia.** Blok počíta ďalší z parametrov optimalizácie – priemerné oneskorenie  $\eta$ , definované vzťahom (4).

**Výpis dát do GUI a pracovného priestoru.** Vypočítané a nájdené hodnoty a dáta sú znázornené v grafickom rozhraní a uložené do pracovného prostredia MATLAB-u.

## Výsledky simulácií

Pre účely testovania algoritmu bolo potrebné zostaviť množinu testovacích vstupných matic konektivity  $C$  takým spôsobom, aby bolo možné porovnanie výsledkov dosiahnutých inými metódami. Použitá a implementovaná bola metóda generujúca náhodné inštalácie podľa [5]. Metóda sa skladá z nasledujúcich krokov:

1. Daný je počet uzlov  $N$  a parameter pre generovanie hrán  $r$ . Ďalej sú stanovené 4 skupiny testovacích prípadov s parametrom  $r$  určeným ako  $r = 1/\sqrt{N}$ ,  $r = 2/\sqrt{N}$ ,  $r = 3/\sqrt{N}$  a  $r = 4/\sqrt{N}$ .

2. Generovanie  $N$  náhodných dvojrozmerných súradníc uzla

$$x_i = \text{random}[0,1] \quad y_i = \text{random}[0,1], \text{ pre } i = 1, \dots, N \quad (8)$$

kde  $x_i$  a  $y_i$  sú  $x$ -ová a  $y$ -ová súradnica  $i$ -teho uzla a funkcia `random[a,b]` generuje náhodné číslo s rovnomerným rozdelením medzi  $a$  a  $b$ .

3. Priradenie hrany dvojici uzlov, ktorých vzdialenosť je menšia než  $r$ .

$$c_{ij} = \begin{cases} 1, & \text{ak } \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} < r \\ 0, & \text{inak} \end{cases} \quad (9)$$

pričom  $i = 1, \dots, N$  a  $j = 1, \dots, N$ .

S použitím stanovených testovacích topológií boli realizované simulácie, ktoré sumarizuje tab. 1. Pre jednotlivé testovacie skupiny boli zvolené topológie s počtom uzlov ( $N$ ) 100-500 resp. 1000-1000, pričom na základe vzťahov z predchádzajúcej kapitoly boli generované topológie so zodpovedajúcim počtom hrán ( $E$ ). Pre účely porovnania sú uvedené parametre minimálny resp. maximálny stupeň uzla danej topológie.

č.m.	TT	N	E	min(deg(v))	max(deg(v))	M
1	1	100	162	2	8	9
	2	300	476	2	9	10
	3	500	827	2	11	12
2	4	100	560	4	18	19
	5	300	1795	4	24	25
	6	500	2812	2	22	23
	7	750	4348	3	24	25
3	8	1000	5826	3	22	23
	9	100	1117	10	38	39
	10	300	3809	5	39	40
	11	500	6214	5	38	39
4	12	750	9997	6	49	50
	13	1000	13028	7	42	43
	14	100	1712	10	56	57
	15	300	6076	12	62	63
	16	500	10869	15	71	72
	17	750	16932	15	70	71
18	1000	22837	15	72	73	

Tab.1 Výsledné dĺžky TDMA rámcov pre jednotlivé testovacie topológie

Ak vychádzame zo vzťahu (5) stanovujúceho dolné ohraničenie dĺžky rámca, ktoré je možné zapísať i ako

$$M \geq \max(\deg(v_i)) + 1, \quad i = 1, 2, \dots, N \quad (10)$$

potom je zrejmé, že pre jednotlivé testovacie topológie boli s použitím navrhnutého algoritmu dosiahnuté optimálne dĺžky TDMA rámca  $M$ .

## Záver

Analýza problematiky ukázala vhodnosť aplikácie mechanizmu mnohonásobného prístupu TDMA na riešenie efektívneho a spoľahlivého zdieľania rádiového prenosového média medzi jednotlivými uzlami bezdrôtovej senzorovej siete. Jednoduché pridelenie časového slotu každému uzlu (triviálny TDMA rámec) vedie pri väčšom počte uzlov k neefektívnemu využívaniu prenosového média a k výraznému oneskoreniu. Dôvodom je neumožnenie vysielania uzlom, ktoré by vysielali mohli bez toho, aby došlo ku konfliktu s uzlom aktuálne vysielajúcim. Na detekciu primárnych a sekundárnych konfliktov bola navrhnutá metóda a z nej vychádzajúci algoritmus lokalizujúci tieto konflikty v zadanej sieťovej topológii reprezentovanej maticou konektivity. Riešenie problému ďalej spočíva v nájdení takého TDMA rámca, ktorý v súlade so zadanými vstupnými údajmi (matica konektivity, matica kompatibility) umožní každému uzlu siete vysielat' aspoň raz počas trvania rámca (pridelí mu aspoň jeden slot) a zároveň má tento TDMA rámec minimálnu dĺžku. Proces optimalizácie tohto primárneho optimalizačného parametra je navyše doplnený o maximalizáciu využitia kanála a minimalizáciu priemerného oneskorenia. Do navrhnutého heuristického algoritmu bol pomocou vektora neuniformných požiadaviek zapracovaný mechanizmus na pridelenie odlišných počtov slotov v rámci daného TDMA rámca. Táto vlastnosť algoritmu umožňuje zvýšiť požiadavky na vysielanie pre uzly s vyššou konektivitou (vyšším stupňom), agregáčné uzly alebo uzly implementujúce integritné resp. bezpečnostné mechanizmy zvyšujúce množstvo potrebných prenesených dát [6]. Navrhnutý algoritmus preukázal v porovnaní s inými algoritmami dosahovanie porovnateľných výsledkov, pričom na testovacích sieťových topológiách bola dosiahnutá optimálna dĺžka TDMA rámca.

## Podakovanie

Táto publikácia vznikla s priamou podporou projektu ITMS 26220120050 s názvom "Centre of excellence for systems and services of intelligent transport II" riešeného v rámci Agentúry Ministerstva školstva, vedy, výskumu a športu SR pre štrukturálne fondy EÚ.

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



## Literatúra

- [1] CHAKRABORTY, G., CHAKRABORTY, D., SHIRATORI, N.: A heuristic algorithm for optimum transmission schedule in broadcast packet radio networks. In *Computer Communications*. ISSN 0140-3664, 2005, vol. 28, no. 1, pp. 74-85.
- [2] EPHREMIDES, A., TRUONG, T.V.: Scheduling broadcast in multihop radio networks. In *IEEE Transactions on Communications*. ISSN 0090-6778, 1990, vol. 38, no. 4, pp. 456-460.
- [3] WANG, G., ANSARI, N.: Optimal Broadcast Scheduling in Packet Radio Networks Using Mean Field Annealing. In *IEEE Journal on Selected Areas in Communications*. ISSN 0733-8716, 1997, vol. 15, no. 2, pp. 250-260.
- [4] YEO, J., LEE, H., KIM, S.: An efficient broadcast scheduling algorithm for TDMA ad-hoc networks. In *Computers & Operations Research*. ISSN 0305-0548, 2002, vol. 29, no. 13, pp. 1793-1806.
- [5] FUNABIKI, N., KITAMICHI, J.: A Gradual Neural Network Algorithm for Broadcast Scheduling Problems in Packet Radio Networks. In *IEICE Trans. Fundamentals*. ISSN 0916-8508, 1999, vol. E82-A, no. 5, pp. 815-824.
- [6] PIRNÍK, R., HALGAŠ, J., ČAPKA, M.: Non-invasive monitoring of calm traffic [Neinvazívne monitorovanie dopravy v klude] In: *International symposium on advanced engineering & applied management*: 4-5 November, 2010, Hunedoara, Romania. - Hunedoara: University Politehnica Timisoara - Faculty of engineering, 2010. - ISBN 978-973-0-09340-7. - S. II-107-II-111.

## Abstract

The paper focuses on the problem of optimisation of radio transmission media access control using the Time Division Multiple Access (TDMA) method within the process control level providing non-uniform access requirements. The primary optimisation criteria (the TDMA frame length minimisation and maximisation of its utilisation) have been examined using a heuristic algorithm.

Ing. Peter Holečko, PhD.

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 1  
010 26 Žilina  
holecko@fel.uniza.sk

# Je **budúcnosť** inteligentných domov v automatizácii rutinných **ovládacích úloh**?

Tomáš Mikluščák, Aleš Janota

## Abstrakt

Výrazný rozvoj informačných a komunikačných technológií v poslednom období umožnil vznik konceptu inteligentných budov. Spojením nových informačno-komunikačných technológií a matematického predikčného aparátu sa koncept inteligentných budov dostal na novú úroveň. Na úroveň, kde sa inteligentné budovy učia návyky užívateľa a časom vedia automatizovať rutinné ovládacie úlohy, napríklad ovládanie systémov HVAC (vykurovanie, ventilácia a klimatizácia) či ovládanie osvetlenia. Tento článok prináša vo svojej úvodnej časti pohľad na state-of-the-art danej oblasti, hodnotí potenciál týchto technológií, porovnáva ich výhody ako aj nevýhody a prináša výsledky vlastných originálnych experimentov v danej oblasti.

**Kľúčové slová:** Ubiquitous Computing, Smart Environment, Smart Building, Machine Learning

## Úvod

V roku 1991 Mark Weiser [1] predpovedal vývoj informačných a komunikačných technológií takým smerom, ktorý zaisťuje, že budú bežní ľudia pri bežných činnostiach používať veľké množstvo výpočtovej techniky – často bez toho, aby si to uvedomovali. Všetky tieto výpočtové zariadenia budú zapojené v jednej komunikačnej sieti. Zaviedol termín „ubiquitous computing“, teda všadeprítomné počítače. Vývoj mu dal za pravdu a 21. storočie môžeme nazvať aj storočím „ubiquitous computing-u“. Mark Weiser však vo svojej prelomovej práci naznačil aj inú vlastnosť počítačov zapojených v danej sieti, ktorú jeho nasledovníci nazvali „context awareness“. V skratke by sa dalo povedať, že úlohou strojov nebude iba zbierať a skladovať surové dáta, ale aj hľadať v nich súvislosti, teda uvedomovať si kontext. Od uvedomenia si kontextu je následne iba krok k hľadaniu vzorov následnej predikcie, ktorá umožňuje proaktívne správanie sa danej a všadeprítomnej siete počítačov. V odbornej anglickej literatúre sa zaužíval pojem „context prediction“.

Prvým priekopníkom, ktorý si uvedomil potenciál novo prichádzajúcej éry všadeprítomných informačno-komunikačných technológií v oblasti automatizácie budov bol Michael Mozer [2]. Vo svojom projekte „The Neural Network House“ už v roku 1998 umiestnil v bežnom obytnom dome veľké množstvo senzorov a predikčný riadiaci systém založený na neurónových sieťach, pričom dokázal úspešne automatizovať ovládanie vetrania, vykurovania, klimatizácie a osvetlenia. Riadiaci systém snímал pomocou senzorov aktivitu v dome a postupne začal automatizovať jednotlivé ovládacie a regulačné úkony. V konečnom dôsledku bolo ovládanie osvetlenia, vykurovania, vetrania a klimatizácie takmer úplne automatické. Riadiaci systém sledoval dva protichodné ciele: prvým bola úspora energie, druhým užívateľský komfort. Úlohou bolo riadiť systém tak, aby mal užívateľ vždy svoj komfort, avšak bez zbytočne vynakladanej energie. Takáto stratégia mala za následok úsporu energie, nakoľko takéto ovládanie je bližšie k optimu

a zároveň rapídne zvýšenie komfortu obyvateľov. Úspora energie vyplynula z faktu, že klimatizované, vykurované či vetrané miestnosti boli iba tie, kde sa obyvatelia zdržiavali alebo tie, kde bol predpoklad, že sa budú v najbližšom čase zdržiavať. Analogicky sa pracovalo s osvetlením. Zvýšený komfort vyplynul z faktu, že užívateľ nemusí nič ovládať ani nič nastavovať.

V podobných experimentoch neskôr pokračovali a stále pokračujú výskumné tímy po celom svete. Medzi najvýznamnejšie projekty patria „The MavHome“ [3], „SOCAM“ [4] alebo „The Aware Home“ [5]. V závislosti od podmienok bola dosiahnutá značná úspora energií, v istých prípadoch dosahovala spotreba celého systému menej než 40% spotreby dosahovanej za normálnych podmienok.

Čiastkové problémy existujúce v danej oblasti, ako napríklad problém predikcie či optimálneho riadenia, riešia aj menšie tímy, ktoré nie sú súčasťou veľkých projektov.

## 1. Ciele projektu

Faktom ostáva, že aj 15 rokov po úspešnom projekte už spomenutého Michaela Mozera [2] ľudia stále trávia mnoho času nastavovaním a regulovaním zariadení vo svojich obydliach a zároveň sú obydliá stále energeticky veľmi náročné, regulácia je ďaleko od optima a istým synonymom pre inteligentný dom sa v komerčnej sfére stal smartfón, ktorým sa dá ovládať osvetlenie. Príčinou tohto stavu môže byť akademické uviaznutie v zložitosti a honba za čím lepšími výsledkami na úkor realizovateľnosti. Spoločným znakom spomenutých projektov je mohutná a drahá technika, ktorej cena a spotreba v bežnom dome nikdy nevyváži ušetrenú energiu a čiastočne aj použitie techniky, ktorú väčšina obyvateľov určite odmietne, napr. kamery či mikrofóny.

Cieľom nášho projektu je vytvoriť čo najjednoduchší riadiaci systém, ktorý aplikuje znalosti, metódy a technológie získané v spomenutých projektoch a upraví ich na použitie v

bežnej domácnosti, t. j. riadiaci systém, ktorý na jednej strane nebude potrebovať tak mohutnú senzoricú sieť a drahé technológie, na strane druhej však bude schopný automatizovať ovládanie vykurovania, vetrania, klimatizácie a osvetlenia s minimálnou potrebou zásahov do ovládania zo strany používateľa - obyvateľa budovy.

Ideálnym prístupom pre komplexné zvládnutie a realizáciu uvedenej myšlienky by bolo skúmanie správania sa reálnych inteligentných riadiacich systémov v rôznych klimatických a iných podmienkach, v rôznych typoch budov a s rôznymi obyvateľmi. Taký prístup si však vyžaduje prostriedky, ktorými náš tím žiaľ nedisponuje. Východiskom z tejto situácie je pre nás tvorba simulačného modelu, ktorý umožňuje odladenie systému v laboratórnych podmienkach, bez potreby drahých experimentov v reálnom svete. Primárnym účelom simulačného modelu nie je ani tak náhrada experimentov realizovaných v reálnom svete na iných pracoviskách, ako skôr ich doplnenie, doladenie a modifikácia, pretože vďaka simulácii je možné v krátkom čase otestovať množstvo situácií, ktoré by si v reálnych podmienkach vyžiadali dlhé roky experimentov.

Predpokladáme, že senzorová technika používaná v moderných bezpečnostných systémoch je schopná zisťovať prítomnosť obyvateľa v jednotlivých miestnostiach domu, ako už bolo viackrát dokázané [10]. Návrh konkrétnej senzorickej siete nie je predmetom tejto práce.

## 2. Simulačný model

Pre overenie nášho riadiaceho systému sme v špecifických podmienkach vytvorili agentovo orientovaný softvérový simulačný model, ktorý sa skladá z viacerých nezávislých agentov, ktoré konajú na základe daných motívácií a vo výsledku by mali tvoriť čo najreálnejší model vlastností, správania sa a interakcií riadiaceho systému, budovy a jej obyvateľov.

Simulácia je založená na prístupe, ktorý patrí do rodiny typu Monte Carlo a možno ho stručne charakterizovať nasledovným spôsobom: keďže každá simulovaná udalosť má istú pravdepodobnosť výskytu, musí sa simulácia s rovnakými vstupnými parametrami mnohokrát opakovať – vytvárať replikácie - a ako výsledok simulácie sa berie štatisticky spracovaný súbor sledovaných udalostí tvorený výsledkami zo všetkých replikácií simulácie.

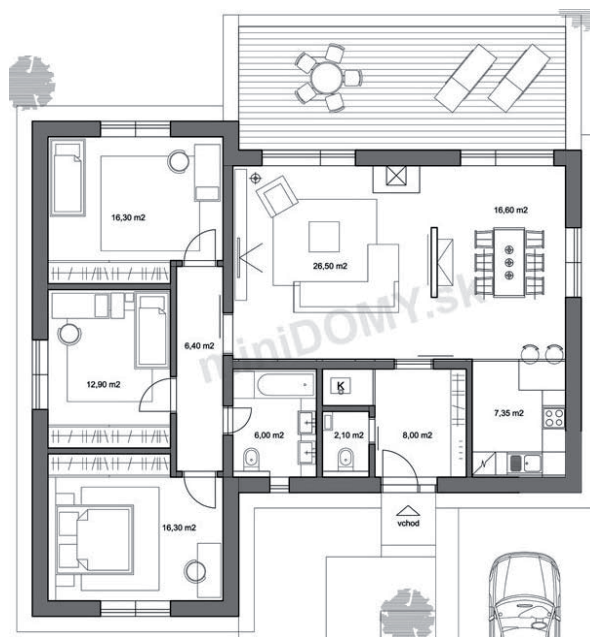
Pre samotnú realizáciu sme využili objektovo - orientované programovanie v jazyku C# na platforme .NET.

### 2.1 Dom

Model domu reprezentuje fyzikálne vlastnosti budovy ako takej. Hlavná vlastnosť, ktorú modelujeme, je naakumulovaná energia v podobe tepla v danom čase a jej zmena pôsobením vonkajších aj vnútorných faktorov. K vonkajším faktorom patrí vonkajšia teplota, k vnútorným energia, ktorú budove dodáva vykurovací systém v podobe tepla, prípadne klimatizácia, ktorá vzduch a budovu ochladzuje. Aktuálna teplota teda závisí od výkonu vykurovacích telies a klimatizácie, vonkajšej teploty a naakumulovaného tepla v budove a vo vzduchu. Pretože dom tvorí množstvo materiálov s odlišnými akumuláčnymi a tepelnoizolačnými vlastnosťami, napr. obvodové múry, vonkajšia izolácia, nábytok, podlaha,... je pre výpočet aktuálnej teploty v danom čase a na danom mieste v dome nutné použiť špecializovaný softvér. Príkladom takéhoto softvéru je eQuest [6].

K ďalším modelovaným vlastnostiam domu patrí jeho fyzický tvar, teda jeho geometria, ktorá determinuje pohyb obyvateľov v rámci budovy.

V porovnaní s inými experimentmi, ktoré boli spomenuté v úvode, v našom prípade sa nemusíme obmedzovať na jednu fyzickú budovu, ale môžeme robiť pokusy s viacerými typmi stavieb. V tomto prípade je to dôležité, pretože pre každý typ stavby je potrebná iná stratégia na ovládanie vykurovania, vetrania a klimatizácie. Všeobecne platí, že čím má stavba vyššiu schopnosť akumulovať teplo, tým viac sa oplatí udržiavať konštantnú teplotu počas celej vykurovacej sezóny. Príkladom takejto stavby je murovaný dom z plnej tehly. Naopak, čím má stavba ako celok nižšiu schopnosť akumulovať teplo, tým viac sa oplatí znižovať teplotu počas neprítomnosti obyvateľov, prípadne v noci. Príkladom takejto stavby sú moderné montované domy všetkých typov.



Obr.1 Pôdorys modelovaného domu

Fig.1 Floor plan of the model house

### 2.2 Obyvatel'

V budove existuje niekoľko obyvateľov, ktorých úlohou je, jednoducho povedané, žiť svoj život. Každý obyvateľ má daný istý typ správania, ktoré sa dá vopred nastaviť. Z reálnych skúseností vieme, že typ správania sa je rôzny od obyvateľa k obyvateľovi. Niektorí jedinci pracujú dlhé hodiny a domov prídu iba prespať. Iní napríklad pracujú z domu či na čiastočný úväzok. Výhodou oproti experimentálnym projektom používajúcich reálne osoby, zvyčajne študentov, je fakt, že typ správania môžeme nastaviť pred simuláciou podľa cieľov experimentu.

Podobne ako v simulácii, ktorú realizoval Erickson [7], pohyb a správanie obyvateľa riadi viacero úrovní rozhodovania. V našom prípade sme však jeho úroveň „Patch choice“ a „Walking behaviour“ spojili do jednej, ktorú sme nazvali realizačná. Vo výsledku máme dve úrovne rozhodovania – cieľovú a realizačnú. Cieľová úroveň riadenia rozhoduje o aktuálnom ciele, ktorý je potrebné dosiahnuť. Realizačná úroveň je zodpovedná za samotnú realizáciu zvoleného cieľa.

#### 2.2.1 Cieľová úroveň rozhodovania

Túto úroveň by sme mohli nazvať aj makro rozhodovanie. Agent sa na tejto úrovni rozhoduje, či svoju aktuálnu činnosť zmení a ak áno, ktorú činnosť si zvolí za svoj ďalší cieľ.

Každá činnosť je spojená s konkrétnym miestom v dome a konkrétnymi aktivitami pri samotnej realizácii. Okrem aktív v dome môže agent vykonávať aj nižšie uvedené aktivity mimo domu.

#### Agent si vyberá z nasledujúcich činnosti v dome:

- Spánok
- Práca s PC
- WC a následná hygiena
- Príprava jedla
- Jedenie
- Pozeranie TV
- Hygiena
- Pohyb na terase
- Práca v záhrade
- Upratovanie
- Popoludňajší oddych

#### Aktivity mimo domu:

- Odchod do práce
- Odchod z domu za nákupom
- Odchod z domu za zábavou
- Príchod domov
- Odchod z domu za iným účelom

Rozhodnutie, ktorú činnosť si obyvateľ zvolí, súvisí s parametrami typu správania, ktoré sú zvolené pred začatím simulácie podľa cieľov experimentu. Parametre správania sú udávané v pravdepodobnostnom rozdelení pre všetky časové intervaly a zároveň sériou podmienok, ktoré určujú, ktorá činnosť obvyčajne nasleduje po inej činnosti, ktoré činnosti sa musia počas dňa vykonať aspoň  $n$ -krát, práve  $n$ -krát alebo najviac  $n$ -krát. Samotné rozhodnutie je, ako to už v podobných simuláciách býva, na správne nastavenom generátore náhodných čísel.

#### 2.2.2 Realizačná úroveň rozhodovania

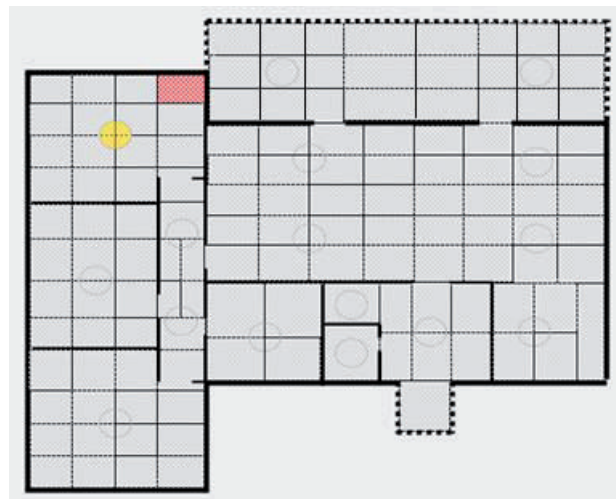
Na rozdiel od cieľovej úrovne rozhodovania, ktorá je do určitej miery stochastická – v závislosti od vstupných parametrov, realizačná úroveň je prísne deterministická. Realizačná úroveň rozhoduje, ako sa zvolená činnosť má vykonať. Predpokladáme pritom, že ak sa obyvateľ potrebuje premiestniť z bodu A do bodu B, tak zvolí najkratšiu trasu a vždy tú istú. Taktiež predpokladáme, že väčšina činnosti má prísnu postupnosť, napr. samotné jedenie sa nemôže odohrať skôr ako príprava jedla.

Pre každú činnosť je priradená jednoznačná lokalita a jednotlivé aktivity. Ešte predtým ako ich obyvateľ začne vykonávať, musí sa na dané miesto dostať. Realizačná úroveň ma okrem iného za úlohu rozhodnúť, ako sa zo súčasnej pozície dostať do určenej lokality. Aby sme vedeli určiť presnú lokalitu a vybrať optimálnu trasu, dom bolo potrebné rozdeliť na jednotlivé segmenty. Segment tvorí štvorec s rozmerom jeden meter. Jeden meter štvorcový sa považuje za osobnú zónu a preto sa nepredpokladá, že by sa v jednom metri štvorcovom nachádzalo viac obyvateľov.

Následne môžeme z takto segmentovaného domu vytvoriť matematický neorientovaný graf

$$\vec{G} = (V, E, C), \quad (1)$$

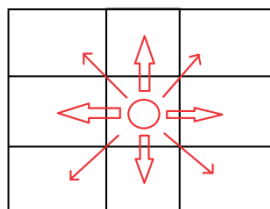
kde  $V$  je neprázdna konečná množina vrcholov grafu, v našom prípade všetky segmenty domu,  $E$  je množina usporiadaných dvojíc typu  $\{u, v\}$ , kde  $u \neq v$ ,  $u \in V$ ,  $v \in V$  nazývaných neorientované hrany neorientovaného grafu, v našom prípade ide o prechody medzi jednotlivými segmentmi domu,  $C$  je množina ocenení hrán z množiny  $E$ .



Obr.2 Pôdorys modelovaného domu rozdelený na jednotlivé segmenty. Červený štvorček označuje obsadenú plochu. Kružnice označujú jednotlivé svietidlá: ak sú vyplnené žltou, tak jednotlivé svietidlo je zasvietené

Fig.2 Floor plan of the model house divided into segments. The red square represents an actual position of the inhabitant. Circles represent particular lights: the yellow circle indicates the light is on

V našom prípade určuje vzdialenosť medzi jednotlivými susednými segmentmi. Pre vzdialenosť platí, že ak sa medzi segmentmi nedá prejsť kvôli prekážke, obvyčajne kvôli stene, nábytku alebo iného obyvateľa, tak je vzdialenosť nekonečno. Ak sa prejsť dá, tak vzdialenosť kolmých a vodorovných pohybov je 1, šikmých naopak  $\sqrt{2}$  – dĺžka základne pomyselného rovnoramenného trojuholníka s dĺžkou ramien 1. Hrany medzi nesusednými vrcholmi neexistujú. Vzdialenosť je zobrazená na obr. 3.



Obr.3 Vzdialenosti medzi segmentmi. Hrubé šípky určujú vzdialenosť 1, tenké naopak  $\sqrt{2}$

Fig.3 Distances between segments. Wider arrows represent distance 1, thin arrows determine distance  $\sqrt{2}$

Orientovaný digraf v bežnom dome nepredpokladáme, nakoľko z empirickej skúsenosti vieme, že ak je bod A dosiahnuteľný z bodu B, potom je obvykle aj bod B dosiahnuteľný z bodu A s rovnakým ocenením hrany.

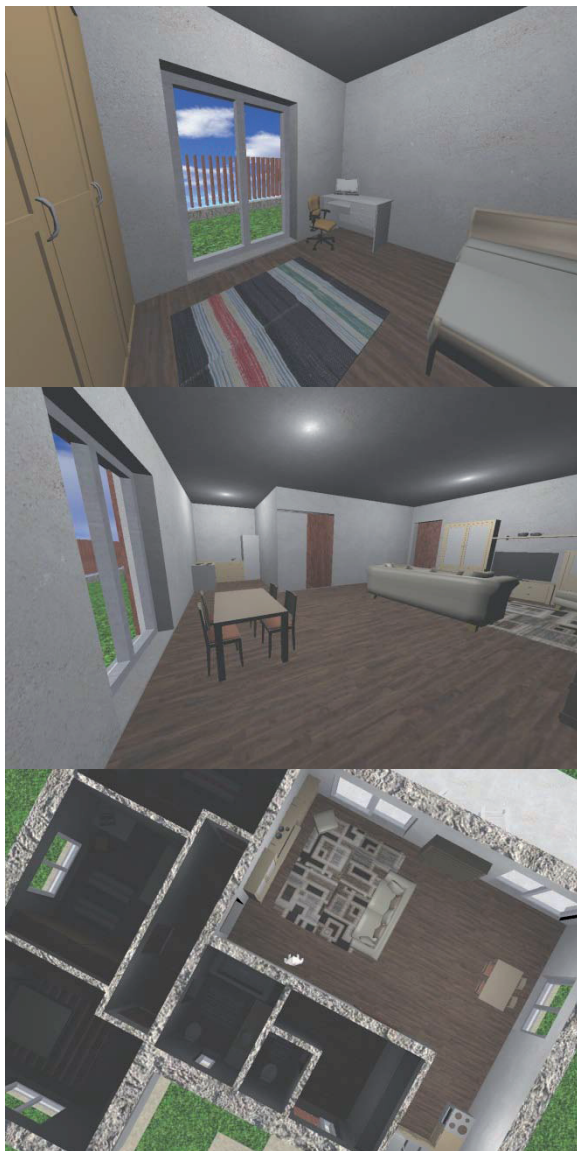
Po transformácii na matematický graf môžeme použiť teóriu grafov k nájdeniu cesty zo segmentu A do segmentu B príslušného domu. Predpokladáme, že obyvateľ zvolí najkratšiu cestu, preto používame základný algoritmus na nájdenie najkratšej cesty.

V prípade konfliktných situácií, teda situácií, keď chce viacerých obyvateľov ísť na rovnaké miesto v rovnakom čase, využívame aparát teórie jednobunkových (celulárnych) automátov, ktorá sa často využíva v príbuznom type simulácií, v simuláciách pohybu chodcov [8] a [9].

### 2.3 Riadiaci systém a vizualizácia simulácie

Tento agent v simulácii sleduje dva ciele: prvým je maximalizácia komfortu obyvateľov a druhým je minimalizácia energetickej spotreby. Riadiaci systém ovplyvňuje iba komfort obyvateľov prostredníctvom domu. Viac informácií o riadiacom systéme je uvedených nižšie.

Štatisticky spracované výsledky simulácie majú veľkú výpočtovú a porovnávaciu hodnotu, niektoré veci je lepšie vidieť animované. Napokon, simulačné nástroje často obsahujú aj grafické animácie simulovaných procesov. Preto sme sa rozhodli vytvoriť grafický animačný model domu a interakcie jeho obyvateľov s ním. Nakoľko samotná simulácia trvá v simulačnom čase obyčajne rok a takýto rok sa v simulácii opakuje (replikuje) stovky krát, vizualizujú sa iba zaujímavé pasáže. Niekoľko ukážok z vizualizácie je na obrázku 4.



Obr. 4 Ukážky z vizualizácie  
Fig. 4 Samples of visualization

### 3. Riadiaci systém

Ako už bolo spomenuté, riadiaci systém sleduje dva ciele. Tým prvým je maximalizácia komfortu obyvateľov a druhým minimalizácia energetickej spotreby. Obidva ciele sa dajú matematicky zapísať v podobe nasledujúcej rovnice, Cieľom riadiaceho systému je následne túto funkciu maximalizovať.

$$f(x) = \sum_{t=1}^n \left( \sum_{i=1}^m (y_i(t)) - \sum_{j=1}^k (u_j(t)) \right) \quad (2)$$

kde:

- $k$  je počet energetických spotrebičov
- $m$  je počet výstupov systému, ktoré majú vplyv na užívateľov komfort
- $n$  je počet časových okamžikov, pre ktorý systém optimalizuje riadenie maximalizáciou funkcie. Je potrebné zvažovať viac časových okamžikov dopredu, keďže riadenie teploty v dome má vďaka tepelnej akumulácii oneskorenú odozvu a teda terajšie rozhodnutia ovplyvňujú budúce náklady.
- $y_i(t)$  je  $i$ -ty výstup systému v čase  $t$ . Ide o príspevok  $i$ -teho výstupu systému (energetického spotrebiča) na komfort užívateľa v čase  $t$
- $\sum_{i=1}^m (y_i(t))$  je suma všetkých príspevkov komfortu užívateľa v čase  $t$ . Je to súhrnná výška komfortu, aký poskytuje systém užívateľovi v danom čase  $t$
- $u_j(t)$  je  $j$ -ty vstup systému v čase  $t$ , teda je to  $j$ -ty odber energie systémom v danom čase a  $\sum_{j=1}^k (u_j(t))$  je súhrnná energetická spotreba systému v čase  $t$ .

Aby sme mohli porovnávať komfort obyvateľa a spotrebu energie, potrebujeme spoločnú jednotku. Spotreba energie sa ľahko prevedie na peniaze a komfort taktiež, pretože ak má obyvateľ na výber medzi zvýšeným komfortom a energetickou úsporou, vždy si vyberie podľa svojej preferencie. Takže, tento pomer sa nastaví sám v učiacej fáze, keď systém iba sleduje správanie sa obyvateľov. Táto téma je podrobnejšie spracovaná v publikácii [12].

Zo vzťahu (2) je zrejmé, že ak má riadiaci systém fungovať správne, musí predikovať správanie sa obyvateľov na istý čas dopredu. Existuje množstvo predikčných metód, ktoré boli použité v danej oblasti. Ich podrobným opisom sa zaoberá napr. publikácia [11].

### 4. Výsledky experimentov

Súčasný stav implementácie nášho simulačného modelu a riadiaceho systému nám zatiaľ dovoľuje skúmať iba možnosť automatického ovládania teploty. Na túto oblasť sme sa doteraz prioritne sústredili preto, lebo v našich klimatických podmienkach je energeticky najnáročnejšia. Doteraz získané výsledky možno považovať za prinajmenšom uspokojivé.

#### 4.1 Opis podmienok

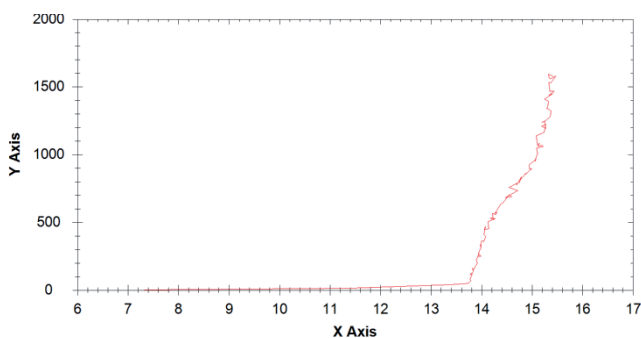
V našom experimente sme simulovali bežný moderný montovaný dom s betónovým základom v klimatických podmienkach, kde vykurovacia sezóna činí 230 dní do roka, čo zodpovedá oblasti severného Slovenska.

Obyvateľstvo domu tvorila štvorčlenná rodina, dvaja pracujúci rodičia a dve školou povinné deti. Obyvatelia sa cítia príjemne, ak je vnútorná teplota na úrovni 23°C, v noci keď spia, teplota môže klesnúť na 17°C. Počas pracovného týždňa zvykne prvý obyvateľ vstávať medzi 6:58 a 7:10 podľa Gaussovho rozdelenia pravdepodobnosti. Dom opúšťajú medzi 7:50 až 7:59, znova podľa Gaussovho rozdelenia pravdepodobnosti s vrcholom v 7:54. Z práce či školy sa zvyknú vrátiť poobede, počas víkendov sú obyčajne doma.

Jedna replikácia simulácie trvá 290 dní, 60 dní je na tréningovanie systému, 230 dní je následná vykurovacia sezóna. Simulácia obsahuje 50 takýchto replikácií. Prebehlo viacero

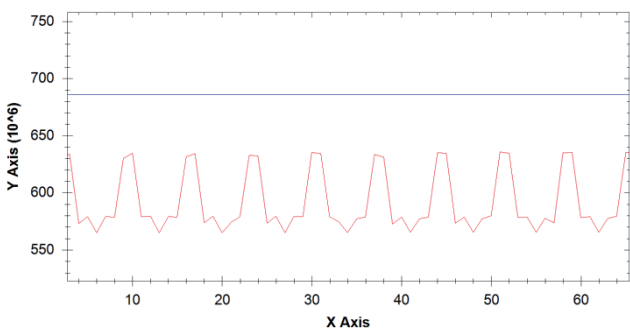
takýchto simulácií s 50 replikáciami, kde sa menil preferenčný pomer komfort/úspora obyvateľov.

Vonkajšia teplota bola počas celej doby nastavená na  $-10^{\circ}\text{C}$ .



**Obr. 5** Závislosť medzi časom, kedy teplota v dome nebola optimálna (Os y) napriek prítomnosti obyvateľov a energetickými úsporami (Os x) v %

**Fig. 5** The dependency between missed time and energy save. The Y axis represents missed time in minutes. The X axis represents energy savings in %



**Obr. 6** Os Y reprezentuje energiu potrebnú na udržanie zvolenej teploty vo Wattoch. Os X reprezentuje dni. Nižšia krivka reprezentuje dynamickú reguláciu teploty, kedy sa teplota znižuje, ak sa nepredpokladá prítomnosť obyvateľov. Vrchná krivka reprezentuje udržiavanie konštantnej teploty  $23^{\circ}\text{C}$

**Fig. 6** The Y axis represents energy needed to keep set temperature in Watts. The X axis represents days. The lower curve represents dynamic control of temperature according to occupancy patterns. The upper curve represents udržiavanie konštantnej teploty  $23^{\circ}\text{C}$  all the time

#### 4.2 Opis dosiahnutých výsledkov

Výsledky simulácie zobrazujú grafy uvedené na obrázkoch 5 a 6. Ako je z výsledných priebehov vidieť, riadiaci systém môže značne zredukovať náklady na vykurovanie len s minimálnych znížením komfortu obyvateľov bez toho, aby musel byť zložito nastavovaný, pretože sa nastaví sám podľa správania obyvateľov. Naše výsledky potvrdili zníženie nákladov, no nie na úrovni niektorých iných porovnateľných štúdií, napr. [13] a [14]. Je to spôsobené vyššou komplexnosťou nášho simulačného modelu, či už ide o uvažované správanie sa obyvateľov alebo o fyzikálne vlastnosti budovy – výsledky sú teda pravdepodobne bližšie realite.

Nami dosiahnuté výsledky naznačujú, že je možné redukovat' náklady na vykurovanie o 13.74% takmer bez straty komfortu a o takmer 8% bez žiadnej straty komfortu. Cena za ušetrenie 13.74% nákladov je 50 minút v dome, ktorý nemá požadovanú teplotu počas obdobia 230 dní, teda

približne každá 6700. minúta. A to všetko s technikou, ktorú je možné zadovážiť za 100 - 200 eur.

#### Záver

Faktom je, že väčšina ľudí má svoje vzory správania, do práce chodia a každý deň sa z nej vracia zväčša v približne rovnakom čase. Rutiny je možné vystopovať na úrovni dní či týždňov. Naše výsledky ukazujú, že aj s použitím minimálnej technológie je možné zvýšiť komfort a znížiť pritom energetickú spotrebu. V budúcnosti budeme pokračovať v implementácii ďalších modulov do nášho riadiaceho systému tak, aby bol časom schopný úplne automatizovať ovládanie vykurovania, vetrania, klimatizácie a osvetlenia.

Záverom možno konštatovať, že odpoveď na otázku položenú v názve článku môže byť: „Zdá sa, že áno“.

#### Literatúra

- [1] WEISSER, M.: The Computer for the 21<sup>st</sup> Century. Scientific American, vol. 265, no. 3, 1991, s. 66-75
- [2] MOZER, M. C., The neural network house: An environment that adapts to its inhabitants. Proc. of the AAAI Spring Symposium on Intelligent Environments, 1998, s. 110–114
- [3] ROY, A., et al.: A Predictive Framework for Location-Aware Resource Management in Smart Homes. IEEE Transactions on Mobile Computing, vol. 6, no. 11, 2007, s. 1270-1283
- [4] GU, T., PUNG, H., K., ZHANG, D., Wang, X.: A Middleware for building context-aware mobile services. Proc. of the IEEE Vehicular Technology Conference (VTC Spring 2004), Minal, Italy, May 17-19, 2004
- [5] Aware Home Research Initiative. Dostupné na: <http://www.awarehome.gatech.edu/>
- [6] eQuest: Building Energy Analysis Tool. Dostupné na: <http://www.doe2.com/>
- [7] ERICKSON, V. L., et al.: Energy efficient building environment control strategies using real-time occupancy measurements. Proc. of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, 2009, s. 19–24
- [8] BIERLAIRE, G. A., WEBERS, M.: Behavioral dynamics for pedestrians. Proc. of the 10<sup>th</sup> International Conference on Travel Behavior Research, August 2003
- [9] SCHADSCHNEIDER, A.: Pedestrian and Evacuation Dynamics, chapter Cellular automaton approach to pedestrian dynamics theory, Springer, 2002, s. 75–86
- [10] SRINIVASAN, V., STANKOVIC, J., WHITEHOUSE, K.: Protecting your daily in-home activity information from a wireless snooping attack. Proc. of the 10<sup>th</sup> international conference on Ubiquitous computing, ACM New York, NY, USA, 2008, s. 202–211
- [11] MIKLUŠČAK, T.: The Predictive Control System in Smart Environment (Minimal Doctoral Thesis). KRIS EF ŽU v Žiline, 2012
- [12] MIKLUŠČAK, T.: Návrh a implementácia užívateľského rozhrania inteligentného domu (Diplomová práca). KRIS EF ŽU v Žiline, 2011
- [13] ALRAZGAN, A., NAGARAJAN, A., BRODSKY, A., EGGE, N.: Learning Occupancy Prediction Models with Decision – Guidance Query Language. Proc. of the 44<sup>th</sup>



Hawaii International Conference on System Sciences (HICSS), 2011, s. 1-10

[14] GAO, G., WHITEHOUSE, K.: The Self-Programming Thermostat: Optimizing Setback Schedules based on Home Occupancy Patterns. Proc. of the First ACM Workshop on Embedded Sensing Systems For Energy-Efficiency in Buildings BuildSys'09, 2009, s. 67-72

**Ing. Tomáš Miklušček**

Žilinská univerzita  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
Email: tomas.mikluscak@fel.uniza.sk

## **Abstract**

Significant developments of technology over the last years, especially in pervasive/ubiquitous computing, have made the dream of smart environment technology true. Merging of this technology with mathematical prediction apparatus allowed raising a new level of smart environments - automating the routine control tasks and pro-active interaction. This paper deals with the potential of automation of the routine control tasks to significantly improve human experience in our houses, offices, buildings and environments in general. We present advantages, shortcomings and upcoming challenges of this technology, state-of-the-art research results as well as the results of our original research in focused area.

**prof. Ing. Aleš Janota, PhD., Eurlng.**

Žilinská univerzita  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
Tel.: ++421-41 513 3356  
Fax: ++421-41 513 1515  
Email: ales.janota@fel.uniza.sk

# Návrh inteligentného používateľského rozhrania pre inteligentný dom

Daniel Michalovič, Tomáš Miklušcak, Aleš Janota

## Abstrakt

Článok sa zaoberá problematikou tvorby inteligentného používateľského rozhrania, pomocou ktorého je možné zadávať príkazy pre inteligentný dom vo forme prirodzeného jazyka. Najskôr sú teoreticky opísané možné spôsoby spracovania prirodzenej reči. Potom autori prezentujú návrh a implementáciu modelu vlastného používateľského rozhrania pre hypotetický inteligentný dom.

**Kľúčové slová:** používateľské rozhranie, inteligentný dom, spracovanie prirodzeného jazyka, pravidlový systém

## Úvod

V komunikácii s technickými systémami sa v súčasnosti môžeme stretnúť s množstvom spôsobov zadávania, prenosu a prijímania informácií za pomoci rozličných používateľských rozhraní. Jedným zo súčasných trendov je snaha vyvíjať také rozhrania, pomocou ktorých môže človek komunikovať so strojom (počítačom) prostredníctvom prirodzenej reči. V takom prípade nie je potrebné učiť sa presné znenie príkazov alebo akýchkoľvek predpísaných kľúčových slov, ale jednoducho prirodzeným spôsobom vyjadriť, aká úloha má byť vykonaná. Využitie tohto prístupu sa dnes hľadá najmä v najnovších aplikáciách smartphonov a v počítačovej technike, kde sa firmy predbiehajú v inováciách a vo zvyšovaní efektivity. Trend budovania inteligentných rozhraní (odhliadnuc od existujúcich výhrad k použitiu a významu slova „inteligencia“ v tomto obľúbenom slovnom spojení) neobchádza ani oblasť inteligentných domov a domácností, kde sa otvárajú možnosti pre ďalšie zvyšovanie komfortu pri ovládaní rôznych zariadení. Veľkú pridanú hodnotu možno vidieť najmä vtedy, ak uvažujeme o aplikáciách pre staršie osoby alebo osoby s určitým telesným hendikepom, pre ktoré zdaniivo bežné úkony v domácnosti môžu predstavovať závažný až neprekonateľný problém.

Rozhranie človek - stroj (angl. *Human-Machine Interface, HMI*) je miesto, kde sa stretáva technológia s človekom. Jeho úlohou je urobiť funkcionality danej technológie jasnou, prehľadnou a zrozumiteľnou. Malo by preto byť čo najjednoduchšie a umožniť tak intuitívne používanie, bez potreby študovania obsiahlych manuálov. Efektívnosť HMI pre takmer všetky technologické riešenia môže predurčiť mieru akceptovateľnosti produktu zo strany budúcich zákazníkov (používateľov) a ako taká je merateľná prostredníctvom viacerých charakteristík. Tieto charakteristiky sa niekedy súhrnne spájajú do jedného spoločného pojmu, ktorým je „použitelnosť“, vyjadrujúca celkovú kvalitu daného produktu. Podľa ISO štandardu [1] táto zahŕňa vlastnosti:

- **užitočnosť:** Vykonáva produkt to, čo vyžaduje používateľ? Vykonáva HMI správne veci?
- **efektívnosť:** Dokážu sa používatelia rýchlo naučiť pracovať s daným HMI? Sú schopní realizovať svoje úlohy s minimálnym úsilím, vrátane minimálnej chybovosti? Zvyšuje to pomer produktivita/úsilie? Vykonáva rozhranie veci správnym spôsobom?

- **spokojnosť:** Vyjadrujú používatelia spokojnosť s produktom? Znižuje nový produkt stres? Majú teraz koncoví používatelia väčšie uspokojenie z práce?

Prieskumy naznačujú, že okolo 50% zdrojového programu v nových softvérových aplikáciách sa venuje práve používateľskému rozhraniu, čo predstavuje významnú nákladovú položku. Aplikácie sú čoraz zložitejšie a práve preto sa v súčasnosti stáva veľkou výzvou zabezpečenie jednoduchosti použitia ako protipól k rastúcej komplexnosti [2].

K dlhodobo používaným tradičným textovým a grafickým rozhraniam pribúdajú v súčasnosti mnohé ďalšie využívajúce možnosti nových technológií, napr.:

- dotykové obrazovky a panely;
- rozhrania snímajúce tváre alebo gestikuláciu;
- rozhrania využívajúce hmatové zmysly (tzv. haptické technológie);
- rozhrania snímajúce svalovú aktivitu alebo odozvy mozgových vln (angl. *brain-computer interface*);
- dialógové rozhrania, atď.

Predmetom nášho záujmu je vývoj hlasového rozhrania, v rámci ktorého by mohol človek komunikovať s riadiacim systémom inteligentného domu prirodzenou rečou.

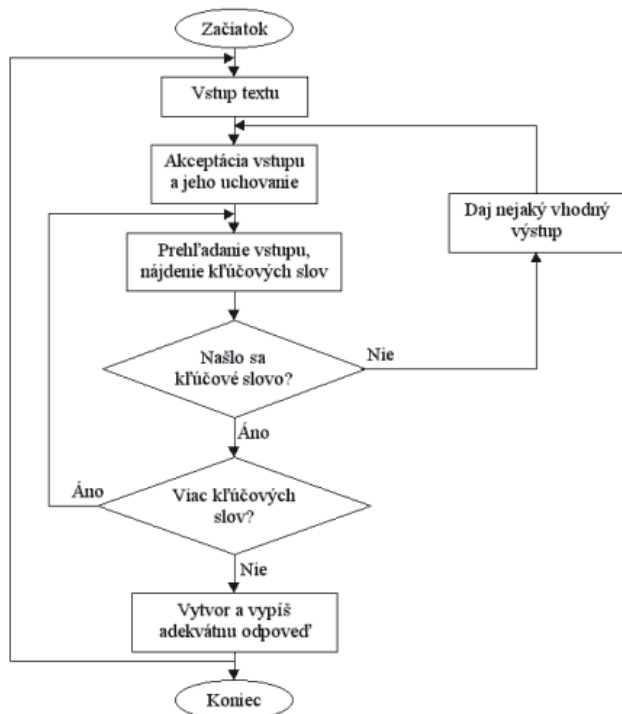
## 1. Spracovanie prirodzeného jazyka

Pri riešení problému spracovania prirodzeného jazyka (angl. *Natural Language Processing*) sa využívajú nielen poznatky z počítačovej vedy, ale aj ostatných oblastí – jazykovedy, psychológie a pod. Spracovanie prirodzeného jazyka je po expertných systémoch najrozšírenejšou aplikáciou umelej inteligencie. Rozhrania využívajúce prirodzenú reč pracujú na princípe prevodu neformálnej reči na formálne príkazy zrozumiteľné pre stroj a naopak. Rôzne rozhrania na tento účel využívajú rôzne techniky a algoritmy. Aby však stroj (počítač) mohol pochopiť otázku alebo príkaz v prirodzenom jazyku, musí mať dostatok vedomostí pre analýzu a interpretáciu vstupu. Musí mať množstvo všeobecných poznatkov o svete, ale taktiež špecifický slovník pre oblasť, v ktorej sa systém používa [3]. Keďže ľudský jazyk je veľmi zložitá forma komunikácie, vznikajú aj mnohé problémy, ktoré je potrebné riešiť pri vytváraní jazykovej interakcie medzi človekom a počítačom. V zásade sa používajú hlavné dva

prístupy k riešeniu, a to hľadanie kľúčových slov a syntakticko-sémantická analýza.

### 1.1 Analýza kľúčových slov

Pri tomto spôsobe riešenia sa hľadajú vo vstupnom reťazci kľúčové slová a frázy, ktoré systém porovnáva s tými, ktoré má uložené v slovníku. Podľa rozpoznaných slov sa potom vykoná príslušná špecifická akcia. Dôležitou vlastnosťou je veľkosť existujúceho slovníka, čo môže predstavovať veľký problém pri procesoch určených pre všeobecné účely, nemusí to však byť až tak limitujúci faktor v prípade používateľského rozhrania konkrétnej aplikácie, akou je nami uvažovaný inteligentný dom. Preto aj tieto jednoduché formy spracovania prirodzeného jazyka majú v praxi svoje opodstatnenie. Takýto systém môže poskytovať textovú alebo rečovú (syntéza reči) spätnú väzbu na základe čiastočného kopírovania kontextu kľúčového slova zo vstupu. Program rozpozná špecifický vstup, ktorý potom použije na tvorbu výstupného textu alebo spustenie určitej akcie. V prípade inteligentného domu bude spätnou väzbou informácia o vykonaní zadaného príkazu, odpoveď na stav, príp. dotaz vyvolaný neúplnosťou poskytnutej informácie (obr. 1).



Obr.1 Analýza prirodzeného jazyka pomocou kľúčových slov [3]

Fig. 1 Analysis of natural language based on key-words processing [3]

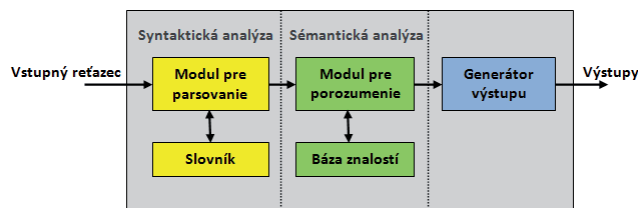
### 1.2 Syntaktická a sémantická analýza

Napriek tomu, že prístup pomocou kľúčových slov je často používanou technikou pri spracovaní jazyka, jeho užitočnosť a použitie sú značne limitované, pretože naráža na neporozumenie rôznym variáciami bežne sa vyskytujúcim v prirodzenom jazyku. Aj tento analytický prístup má však silné obmedzenia vyplývajúce z povahy a čít prirodzeného jazyka. K najväčším problémom patrí skutočnosť, že v jazyku neexistuje jednoznačný vzťah medzi výrazom a významom a takmer nekonečná možnosť kombinovania prvkov medzi sebou (spájanie slov do viet). Okrem toho obrovský rozsah slovnej zásoby nepriaznivo vplyva najmä na rýchlosť analýzy a náročnosť prípravy slovníkov.

Samotná analýza prirodzeného jazyka sa dá rozdeliť do viacerých rovin [3]:

- *fonologická analýza*: rozbor zvukového vstupu a jeho transformácia do elektronického textu (zväčša riešené samostatným programom);
- *morfológická analýza*: analýza tvaroslovia, pomocou nej snaha identifikovať gramatické kategórie jednotlivých slov;
- *syntaktická analýza*: identifikácia vetných členov vo vete a ich skladba do fráz;
- *morfeatická analýza*: rozbor morfológickej stavby slov a identifikácia významu morfému v slovách;
- *sémantická analýza*: rozbor významových častí vo vete a identifikácia sémantických pádov;
- *kontextová analýza*: uvažovanie vety v súvislosti s kontextom.

Základná bloková schéma programu určeného na porozumenie prirodzenému jazyku je uvedená na obr. 2.

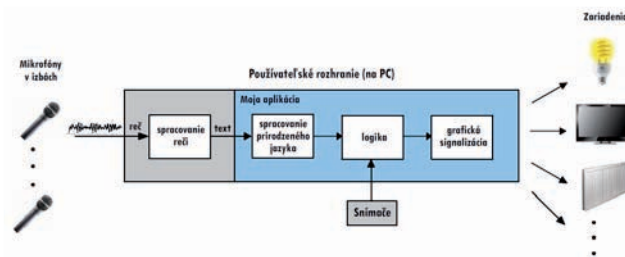


Obr.2 Základná schéma programu pre porozumenie prirodzenému jazyku

Fig. 2 Basic scheme of the program for natural language understanding

## 2. Tvorba a činnosť rozhrania

Primárnym zameraním tu prezentovaného výskumu je návrh a vytvorenie programovej implementácie inteligentného používateľského rozhrania inteligentného domu, ktoré by malo byť v maximálnej možnej miere schopné rozpoznávať príkazy zadávané v prirodzenej reči. Rozhraním je možné simulovať ovládanie osvetlenia, médií, alarmu alebo nastavenie teploty v jednotlivých izbách domu pri zohľadnení vnútornej a vonkajšej teploty (obr. 3).



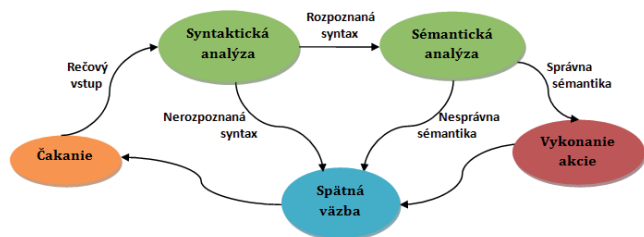
Obr.3 Zjednodušená štruktúra systému

Fig. 3 A simplified structure of the system

Proces spracovania prirodzenej reči pre vyvíjané inteligentné rozhranie inteligentného domu (sivá časť systému na obr. 3) môžeme znázorniť diagramom na obr. 4. Najskôr je reč zachytená cez mikrofón a následne spracovaná technikami frekvenčnej analýzy a napr. skrytých Markovových modelov, o čo sa postará vybraná knižnica pre daný jazyk. Takto získame z hovorenej reči slová a vety, ktoré je potrebné podrobiť ďalšej analýze. Analýza sa uskutočňuje v 2 fázach - syntaktickej a sémantickej.

Syntaktická analýza najskôr definuje, či je vstupom otázka alebo príkaz. Potom analyzuje jednotlivé slová, ktoré opisujú vstup (akcia, spotrebič, miesto, parameter, hodnota, čas, atď.). Pre jeden príkaz môžeme použiť rôzne syntaktické štruktúry vety alebo také štruktúry, ktoré sa líšia iba poradím

slov a ich tvarom. Ak si používateľ želá napr. zapnúť svetlá v spálni, môže tento svoj zámer prezentovať viacerými spôsobmi, a v rôznom poradí slov vo vete (najmä v angličtine má poradie slov veľký význam). Mohli by byť napr. vyslovené vety ako „zapni svetlo v spálni“ alebo „prosím, mohol by si zapnúť svetlo?“ alebo „nevidím, je tu tmá“.



Obr.4 Proces spracovania prirodzenej reči pre aplikáciu „inteligentný dom“

Fig. 4 Procedure for natural language processing in the “smart house” application

Štruktúra vety rozpoznanej systémom musí zodpovedať aj kritériám sémantickej správnosti, tzn. mala by dávať významovo nejaký zmysel. Je oveľa náročnejšie a dôležitejšie nájsť význam vety (príkazu), o čo sa stará práve sémantická analýza. Tá hľadá vzťah medzi slovnými spojeniami vo vstupnom reťazci a aktuálnou situáciou v inteligentnom dome. Systém môže akceptovať aj vetu ako "zapni okno v kuchyni" a sémantická interpretácia rozhrania môže porozumieť takémuto príkazu. To isté platí pre reláciu medzi príkazom a skutočným stavom domu – systém by mal vedieť, že niektoré akcie nemôže vykonať (napr. ak bude vydaný príkaz na vypnutie zariadenia, ktoré je však už vypnuté). Systém môže prostredníctvom rozhrania klásť používateľovi aj doplňujúce otázky v prípade nesúladu medzi príkazom a tým, čo môže systém vykonať, alebo v prípade, ak mu chýba potrebná informácia. Akcia sa vykoná, až keď budú k dispozícii všetky potrebné informácie na jej realizáciu. Po jej vykonaní nasleduje odoslanie informácie používateľovi.

Ak sa v rámci uskutočnenej príkaz vyhodnotí ako syntakticky aj sémanticky správny, môže dôjsť k dvom rôznym reakciám:

- systém vyhodnotí vstup ako príkaz na vykonanie akcie, a to okamžité (vypnúť svetlo, zvýšiť hlasitosť televízora) alebo oneskorené (programovanie kúrenia) a zaznamená nový stav do príslušnej databázy;
- systém vyhodnotí vstup ako otázku (na stavy, parametre alebo programovanie), a rozhranie sa v tomto prípade len pozrie na aktuálne hodnoty v databáze a poskytne ich používateľovi [4].

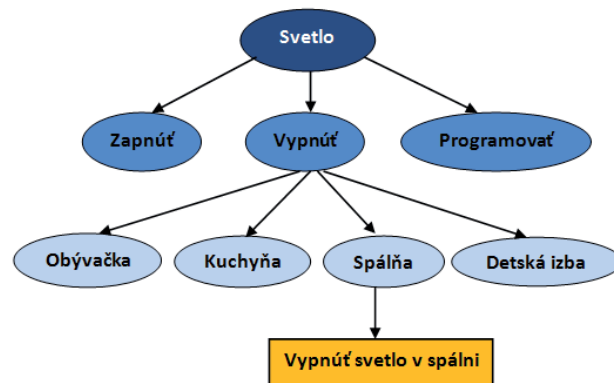
### 2.1 Analýza príkazu na vykonanie akcie

Na analýzu akcií a ich vzťahov ku zariadeniam a miestnostiam môžeme s výhodou použiť názorný diagram so stromovou štruktúrou, ktorý pozostáva z koreňa a nižších úrovní (listov). Stromovú štruktúru tak môžeme zobrazit 2 spôsobmi:

- štruktúra je centralizovaná v zariadení, o úroveň nižšie sú všetky možné akcie zariadenia a na najnižšej úrovni je lokalita, ktorá nakoniec identifikuje konkrétne zariadenie (príklad na obr. 5);
- štruktúra je centralizovaná v akcii, na nižšej úrovni sú možné zariadenia, ktoré dokážu vykonať túto akciu a na najvyššej úrovni je tiež lokalita, ktorá určí zariadenie (príklad na obr. 6).

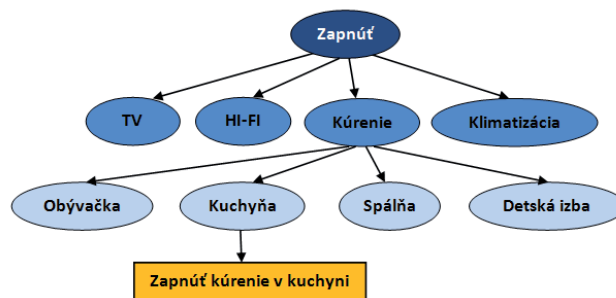
Či už prostredníctvom prvej alebo druhej štruktúry sa zdefiniuje úplný príkaz, po vykonaní ktorého dostane používateľ

spätnú informáciu - buď o jeho splnení alebo o aktuálnom stave zariadenia.



Obr.5 Štruktúra centralizovaná v zariadení

Fig. 5 The equipment-centralized structure



Obr.6 Štruktúra centralizovaná v akcii

Fig. 6 The action-centralized structure

Konečným cieľom je simulovať čo najprirodzenejší dialóg a preto je vhodné dávať premenlivé spätné väzby, ktoré náhodne menia svoju štruktúru. Pri použití angličtiny je situácia jednoduchšia – môže sa napríklad meniť len začiatok vety (*I have already..., I just..., I have...*). Všeobecná štruktúra potom vyzerá takto: (Už som ..., Práve som ..) + Akcia + Zariadenie + Izba (v prípadoch, kde je to potrebné uviesť) [4]. Pri použití slovenčiny je k dispozícii oveľa viac spôsobov, ako by mohol systém odpovedať, a to rôznym skloňovaním slov, zmenou skladby viet a pod.

### 2.2 Analýza otázky (na stavy, parametre, programovanie)

Existujú dva typy otázok, ktoré môže rozhranie rozpoznať. Preto aj pri otázkach využívame dve stromové štruktúry, ktorými je možné otázku opísať. Môže to byť otázka centralizovaná v zariadení, tzn. používateľ sa pýta na konkrétne zariadenie. Napríklad „Je okno v kuchyni otvorené?“. Tu systém očakáva štruktúru vety: („je ...“, „sú ...“, „aký je ...“) + zariadenie + umiestnenie (v prípadoch nejednoznačnosti) + druh stavu, ktorý chce používateľ zistiť + otáznik na konci vety. Na poradí slov nezáleží. Odpoveďou bude stav zariadenia. Druhý typ otázky je centralizovaný v akcii. Používateľ chce vedieť, ktoré zariadenia sú v určitom stave, napríklad: „Ktoré zariadenia sú zapnuté v spálni?“. Odpoveďou na tieto otázky bude potom zoznam zariadení, ktoré majú konkrétny stav a vyhovujú špecifikácii. Stromové štruktúry sa teda pri otázkach používajú na to, aby systém vedel zdefiniovať, o akú požiadavku používateľa ide.

Pozornosť autorov v nasledujúcej časti tohto článku sa sústreďuje na spracovanie výstupov procesov analýzy reči, t.j. príkazov vstupujúcich do systému v textovej podobe (modro vyznačená časť systému na Obr. 3) na základe konkrétnej uskutočnenej implementácie prezentovanej v [5].

### 2.3 Spracovanie zadaného textu

Prvotnou snahou o spracovanie prirodzeného jazyka bolo získať čo najviac informácií o slovách a morfémech. Prostredníctvom analýzy jednotlivých slovných druhov boli získavané pravidlá, ktoré by sa dali využiť v algoritmoch rozpoznávajúcich, či ide o sloveso, podstatné meno alebo o iný slovný druh. Ako kľúčové sa javili napríklad prípony ako *-ť*, *-ni*, *-ie*, ktoré umožňujú určiť, že ide o sloveso v textovom reťazci. Ďalej sa analyzovali morfémy, t.j. najmenšie viditeľné časti slov, ktoré by poslúžili pri rôzne skloňovaných slovách. Pomocou nich by sa dali učiť konkrétne akcie, spotrebiče alebo parametre. Podmienkou úspešnosti analýzy bolo získanie a spracovanie obrovského objemu rôznych textov, či už z odbornej literatúry, záverečných školských prác, bulváru, beletrie alebo iných zdrojov, nakoľko z týchto textov sa dá vytvoriť jazykový model použiteľný pri spracovaní prirodzenej reči a práci s kontextom. Výsledky podobného prístupu možno nájsť napríklad v SAV, IBM a mnohých ďalších firmách, ktoré vytvárajú softvérové nástroje na spracovanie prirodzeného jazyka (Siri, Voice Actions, Google Assistant – projekt Google Majel, Iris, Vlingo Virtual Assistant, Skyvi, Evi, S Voice, Q Voice, Dragon, atď.). Primárnym zámerom bolo získanie max. množstva slov, vyhodnotenie početnosti ich výskytu a vytvorenie potrebnej ontológie. Po úvodnom zhromaždení a automatizovanom spracovaní prvých 300 strán textu (čo je stále nedostatočné množstvo) sa síce získali prvé relevantné údaje o početnosti jednotlivých slov, no pre prílišnú náročnosť a nepresnosť celého procesu bol tento prístup v rámci implementácie [5] zamietnutý a nahradený manuálnou tvorbou slovníka. Slová v slovníku boli kategorizované do 4 skupín reprezentujúcich možné vstupy pre rozhranie v inteligentnom dome:

- akcie;
- predmety;
- parametre;
- miestnosti.

Slovník bol budovaný tak, aby obsahoval slová vo všetkých tvaroch spolu so synonymami a slangovými výrazmi. Vo fáze implementácie boli všetky výrazy vkladané do XML súboru využívaného samotnou aplikáciou.

#### 2.3.1 Analýza pred tvorbou slovníka

Pred vytvorením slovníka bolo potrebné analyzovať:

- priestorové usporiadanie inteligentného domu (aké miestnosti sú k dispozícii);
- jednotlivé prvky interakcie (zariadenia, spotrebiče);
- parametre, ktoré bude možné nastavovať;
- akcie, ktoré bude možné vykonať;
- používateľov, ktorí budú systém (prostredníctvom rozhrania) využívať.

Analýza pozostávala z 3 fáz:

a) Analýza kontextu „Čo a kde?“

Otázka „Čo“ identifikuje rôzne interakčné prvky (svetlo, televízia, klimatizácia, atď.) alebo parametre (hlasitosť, teplota, stanica, atď.). Otázka „Kde“ slúži na rozlíšenie prvkov, ktoré sú rovnaké alebo majú rovnakú funkciu. Je zrejmé, že dve zariadenia nemôžu byť na rovnakom mieste v rovnaký čas alebo jedno zariadenie na dvoch miestach (napríklad svetlo v kuchyni a svetlo v spálni).

b) Analýza kontextu „Aká?“

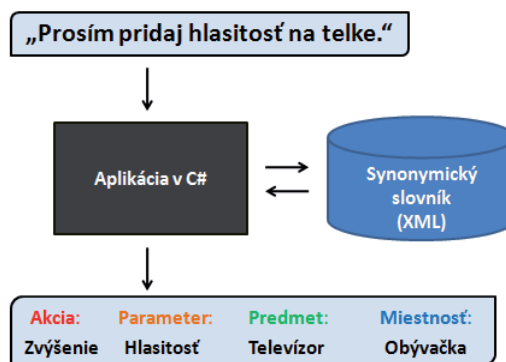
Analýza úloh slúži na identifikáciu jednotlivých akcií, ktoré je používateľ schopný vykonať pomocou rozhrania (nastavenie teploty v izbe, vypnutie svetla, zapnutie televízie a pod.).

c) Analýza kontextu „Kto?“

Dôležitou otázkou môže byť aj kto používa rozhranie. Vďaka identifikácii osoby možno meniť prístupové práva. Táto problematika nebola v opisovanej implementácii osobitne sledovaná.

#### 2.3.2 Princíp porozumenia vstupnému reťazcu

Cieľom bolo hľadanie kľúčových slov a fráz vo vstupnom reťazci, ktoré systém porovnáva s tými, ktoré sú vyskloňované a uložené v slovníku. Ide teda o extrakciu príkazov (vybraných slov) z neformálnej reči, s ktorými by dokázala logika systému ďalej pracovať. Systém vykoná syntaktickú analýzu a na základe identifikovaných slov sa ďalej snaží porozumieť významu a kontextu. Vstupom do rozhrania môže byť napríklad slangová veta: „Prosím pridaj hlasitosť na telke“ (Obr. 7). Program vyhodnotí, že ide o akciu „zvýšenie“, parameter „hlasitosť“ a predmet „televízor“. Pretože ale systém rozumie kontextu, nie je nutné zadávať také úplné znenie vety: ak je televízor zapnutý a príkaz bol zadávaný v obývačke, je zrejmé, že ide o televízor v obývačke. Zadanie skrátenej formy v tvare „pridaj zvuk“ alebo „daj hlasnejšie“ by v takom prípade malo rovnaký efekt.

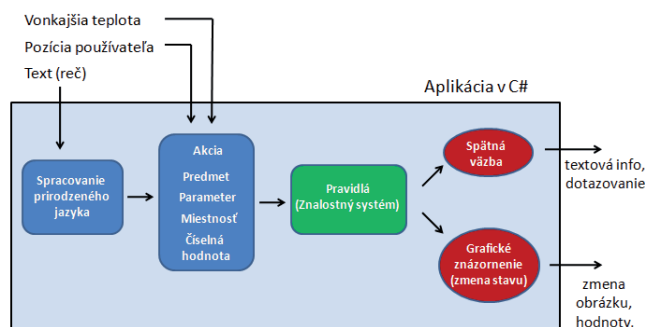


Obr.7 Príklad spracovania príkazu v prirodzenej reči

Fig. 7 An example of a command given in the natural language

#### 2.3.3 Štruktúra rozhrania

Zjednodušenú štruktúru rozhrania môžeme vidieť na obr. 8. Vstupmi sú text (príkazy), pozícia používateľa a vonkajšia teplota (senzory), ktoré bude možné nastavovať v samotnom používateľskom rozhraní, keďže aplikácia nie je (zatiaľ) realizovaná v skutočnom dome.



Obr.8 Štruktúra rozhranie aplikácie

Fig. 8 The structure of the application interface

Kľúčové slová získané analýzou prirodzeného jazyka vstupujú do pravidlového systému v podobe textových reťazcov, prípadne aj ako číselné hodnoty. Množstvo pravidiel tvoriacich pravidlový znalostný systém vyhodnotí požadovanú inštrukciu a následne sa vyšle informácia (spätná väzba) v podobe textovej informácie o prijatí/neprijatí príkazu alebo dodatočné vyžiadanie si chýbajúcej informácie. Zmenu

grafických symbolov sa signalizuje zmena stavov modelovaných zariadení s doplnkovými textovými informáciami.

### 2.3.4 Vytvorenie synonymického slovníka v XML

Výhoda zápisu kľúčových slov do XML formátu je v tom, že dáta sú hierarchicky štruktúrované a teda prístup aplikácie k nim je jednoduchý a zapisovanie dát je prehľadné (na rozdiel od použitia databázy, kde je potrebné riešiť otázky prístupu, riadenia databázy atď.). Rozdelenie slov do 4 hlavných kategórií zodpovedá vyššie uvedenému členeniu na akcie, predmety, parametre a miestnosti (obr. 9, obr. 10).

```

<?xml version="1.0" encoding="utf-8"?>
<VSETKY_SLOVA>
  <AKCIE>...</AKCIE>
  <PREDMETY>...</PREDMETY>
  <PARAMETRE>...</PARAMETRE>
  <MIESTNOSTI>...</MIESTNOSTI>
</VSETKY_SLOVA>

```

Obr.9 Členenie XML slovníka

Fig. 9 Structuring of the XML vocabulary

```

<VSETKY_SLOVA>
  <AKCIE>
    <SVETLÁ>
      <ZAPNÚŤ>
        <SLOVO>zapnúť</SLOVO>
        <SLOVO>zapni</SLOVO>
        <SLOVO>zapnutie</SLOVO>
        <SLOVO>pozapínať</SLOVO>
        <SLOVO>pozapínaj</SLOVO>
        <SLOVO>pozapínanie</SLOVO>
        <SLOVO>zapáliť</SLOVO>
        <SLOVO>zapáľ</SLOVO>
        <SLOVO>zapálenie</SLOVO>

```

Obr.10 Ukážka fragmentu XML slovníka

Fig. 10 The sample of the XML vocabulary

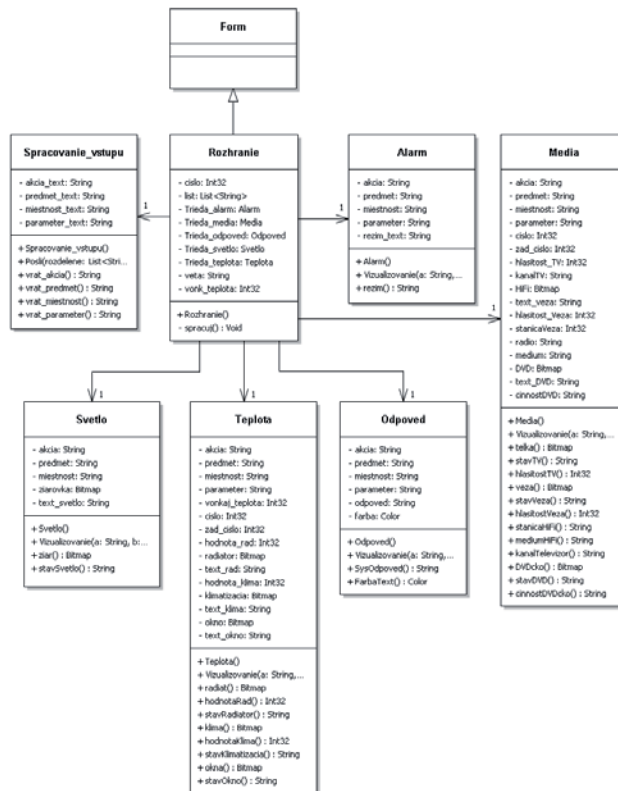
## 2.4 Implementácia

Pre potreby implementácie bol zvolený jazyk C# na platforme .NET Framework. Ako vývojové prostredie poslúžilo Microsoft Visual Studio 2010, ktoré sa vyznačuje prehľadnosťou, užívateľsky prijateľným prostredím a obsahuje množstvo nástrojov na uľahčenie procesu programovania a jednoduchú tvorbu GUI. Aplikácia využíva synonymický slovník vytvorený v XML formáte. Logo aplikácie a obrázky jednotlivých prvkov, ktoré znázorňujú ich stav, boli nakreslené v programe Adobe Flash CS3.

### 2.4.1 Diagram tried a koncepcia

Pre lepšie vysvetlenie princípu fungovania programu a celkovej koncepcie môže poslúžiť diagram tried znázornený na obr. 11. Aplikácia pozostáva z hlavného formulára (trieda Rozhranie), ktorý nielen volá jednotlivé triedy, ale aj vytvára okno grafickej aplikácie Form. Návrh obsahuje 6 tried, z ktorých každá má svoju špecifickú úlohu v rozhraní:

- Spracovanie\_vstupu: spracovanie prirodzeného jazyka (využíva XML slovník)
- Svetlo: ovládanie svetiel
- Teplota: ovládanie teploty (radiátor, klimatizácia, okná)
- Media: ovládanie médií (TV, DVD prehrávač, HiFi veža)
- Alarm: ovládanie alarmu
- Odpoved: systémové odpovede (spätná väzba).



Obr.11 Diagram tried

Fig. 11 The class diagram

Hlavný formulár (Rozhranie) je srdcom celej aplikácie, volá a využíva funkcionality ostatných tried a vytvára grafické okno. Z vety, ktorú napíše používateľ do vstupného poľa (TextBoxu) sa najskôr odstránia všetky interpunkčné znamienka, potom je veta rozdelená na slová, ktoré sa uložia do poľa s textovými dátami. Taktiež sa vo vete vyhládajú aj čísla, ktoré je potom možné použiť pri zmene niektorých nastavení. Spomínané pole so slovami od používateľa je spracované triedou Spracovanie\_vstupu. Z prirodzenej reči tak dostaneme slová v základnom stave, ktoré následne idú cez Rozhranie do pravidlového systému, čiže jednotlivých tried na ovládanie zariadení. Výstupom z týchto tried sú príkazy na zmenu obrázku, textovej informácie alebo hodnoty.

### 2.4.2 Spracovanie vstupu

Úlohou triedy Spracovanie\_vstupu je spracovať prirodzenú reč na exaktné príkazy. Trieda prijme od Rozhrania pole so slovami od používateľa, no aby sa mohli porovnať s tými, ktoré sú uložené v XML dokumente, musia sa najskôr všetky slová z XML načítať do 4 slovníkov v aplikácii.

```
Dictionary<string,int> akcie = new Dictionary<string, int>();
Dictionary<string,int> predmety = new Dictionary<string, int>();
Dictionary<string,int> miestnosti = new Dictionary<string, int>();
Dictionary<string,int> parametre = new Dictionary<string, int>();
```

Nasledujúca ukážka zo zdrojového kódu zobrazuje načítanie do slovníka akcie pre slová zapnúť a vypnúť. Xml-TextReader prechádza postupne riadky v XML a switch prepína podľa toho, či narazí na začiatkový tag, text v jednotlivých akciách alebo koncový tag. Každý prvok v slovníku má textovú a číselnú informáciu. Číslo je pre celú množinu synonymm jedného slova rovnaké. Každá množina konkrétnej akcie, predmetu alebo miestnosti má teda svoje unikátne číslo. Napríklad zapnúť, zapni, zapnutie, zapni, atď. budú mať všetky v slovníku akcie pridelené číslo 1, podľa ktorého je možné zistiť exaktné slovo v základnom tvare.

```
XmlTextReader reader = new XmlTextReader("slova.xml");
string kontrola = "";

while (reader.Read())
{
    switch (reader.NodeType)
    {
        case XmlNodeType.Element: // názov začiatkového tagu
            if (reader.Name != "SLOVO")
                kontrola = reader.Name;
            break;

        case XmlNodeType.Text: // text medzi jednotlivými tagmi
            string a;

            if (kontrola == "ZAPNÚŤ")
            {
                a = reader.Value;

                if (!akcie.ContainsKey(a))
                    akcie.Add(a, 1);
            }

            if (kontrola == "VYPNÚŤ")
            {
                a = reader.Value;

                if (!akcie.ContainsKey(a))
                    akcie.Add(a, 2);
            }

            break;

        case XmlNodeType.EndElement: //názov koncového tagu
            if (reader.Name != "SLOVO")
                kontrola = "";
            break;
    }
}
```

Ďalej nasleduje časť kódu, ktorá zabezpečuje samotné spracovávanie prirodzeného jazyka. Ide o porovnávanie používateľom zadovaných slov, ktoré sa nachádzajú v poli rozdelené so slovami v slovníkoch. Ak sa napríklad nájde slovo v množine, ktorá má číslo 1, do premennej akcia\_text sa uloží slovo v základnom tvare, t. j. „Zapnúť“.

```
foreach (string s in rozdelené)
{
    if (akcie.ContainsKey(s))
    {
        jeAkcia = true;

        if (akcie[s] == 1)
        {
            akcia_text = "Zapnúť";
        }

        if (akcie[s] == 2)
        {
            akcia_text = "Vypnúť";
        }
    }
    ...
}

if (!jeAkcia)
{
    akcia_text = "Neznáma akcia"; //žiadnu akciu nerozpoznal
}
...
```

Ak sa nenájde v slovníku žiadna zhoda, vyhodnotí sa neznáma akcia. S týmito slovami v základnom tvare potom pracuje logika v jednotlivých triedach na vyhodnocovanie príkazov.

Triedy Teplota, Svetlo, Media a Alarm tvoria spomínaný pravidlový systém, ktorý je vlastne vnútornou vyhodnocovacou logikou, na základe ktorej dostávame z používateľského rozhrania adekvátny výstup, čiže signalizáciu zmeny

stavu zariadenia (obrázok, text alebo hodnotu). Princíp týchto tried je približne rovnaký, preto je na ukážku bližšie opísaná iba jedna z nich - Teplota.

Konkrétne do tejto triedy vstupujú slová v základnom stave (akcia, predmet, parameter, miestnosť), vonkajšia teplota a číselná hodnota. Číselné vstupy sú ošetrené proti zadávaniu hodnôt mimo akceptovateľného rozsahu. Keďže pri zadávaní rôznych jednoslovných fráz by program podľa nasledovnej logiky nevedel určiť, čo chce používateľ povedať, niektoré slová (ako napr. „zakúť“) sú obsiahnuté v akciách aj predmetoch, čiže dôjde k vyhodnoteniu „zapnúť“ a „radiátor“.

V nižšie uvedenej ukážke kódu môžeme vidieť pravidlo, ktoré pri určitých podmienkach zapne radiátor v obývačke a ak nie je zadaná presná hodnota, na akú sa má vykurovať, zvýši cieľovú teplotu o 1 stupeň. Aby nevznikli tepelné straty, okamžite sa zavrie okno (ak je otvorené) a vypne sa klimatizácia.

```
if (akcia == "Zvýšenie" & parameter == "Teplota" & miestnosť == "Obývačka" & vonkaj_teplota < 27)
{
    radiator1 = Properties.Resources.radiator_zap; //obrázok zap. radiátor
    text_rad1 = "Kúrenie: zapnuté"; //text

    if (zad_cislo == -1 & cislo < 34) //zad_cislo=-1 ak nebolo zadané číslo
    {
        cislo = cislo + 1; //teplota +1 stupeň
    }
    hodnota_rad1 = cislo; //hodnota na ktorú vykuruje

    okno1 = Properties.Resources.okno_vyp; //vypnutie klímy a zatvorenie okna
    text_okno1 = "Okno: zatvorené";
    klimatizacia1 = Properties.Resources.klima_vyp;
    text_klima1 = "Klimatizácia: vypnutá";
    hodnota_klima1 = cislo;
}
```

Množina veľkého počtu takýchto pravidiel v každej triede zabezpečuje realizáciu rozhodovacieho procesu. Keďže kúrenie, klimatizácia a vetranie spolu vzájomne súvisia a môžu sa ovplyvňovať, nachádzajú sa v jednej triede. Podobne sú spolu aj jednotlivé médiá.

Úlohou triedy Odpoved je vytvorenie spätnej väzby pre používateľa. Obsahuje textovú informáciu (farebne odlišenú podľa typu zelená/červená) o tom, či bol zadaný správny príkaz, no môže ísť aj o dotazovanie sa na určitú chýbajúcu informáciu potrebnú pre vykonanie akcie. Riešenie je veľmi jednoduché, skladá sa tiež z pravidiel, ktoré na základe toho, ak sú vstupné akcie, predmety, atď., vyhodnotenú ako neznáme, vyvodí nesprávny príkaz alebo absenciu časti informácie. Presnosť tohto spôsobu nie je 100 percentná, pretože môže vyhodnotiť príkaz za správny aj vtedy, keď je zadaný nezmysel ako „zasviet okno“, no ako orientačná informácia pre používateľa by mal byť tento prístup postačujúci.

Predvolené vety a ich podmienky sú nasledovné:

- Príkaz prijatý (našli sa slová aspoň v dvoch slovníkoch).
- Nesprávny príkaz (nenašlo sa žiadne kľúčové slovo).
- S čím mám vykonať túto akciu? (známa iba akcia).
- Čo s ňou/ním mám urobiť? (známy iba parameter alebo predmet).

Ukážka implementovaného používateľského rozhrania je na Obr. 12. V hornej časti je okno na vpisovanie textu, do ktorého používateľ zadáva príkazy (simulácia výstupov z procesu analýzy reči). Pre potvrdenie stačí stlačiť kláves *Enter* alebo kliknúť na tlačidlo *Spracuj*. O riadok nižšie je viditeľná spätná väzba aplikácie, ktorá nás v tomto prípade informuje, že príkaz bol prijatý. V rozhraní môžeme vidieť aj výsledky spracovania prirodzenej reči pre akciu, predmet, parameter a miestnosť. K dispozícii je možnosť zmeny pozície používateľa a vonkajšej teploty, ktoré by boli v skutočne implementovanom systéme zabezpečované prostredníctvom snímačov. Pozícia je dôležitá vtedy, ak používateľ nezadal miesto, kde sa má akcia vykonať, v takom prípade sa automaticky zoberie miesto, odkiaľ bol príkaz zadaný. V strednej

časti je k dispozícii možnosť prepínania kariet s názvami miestností, čím možno zisťovať aktuálne stavy zariadení v jednotlivých častiach domu/domácnosti.



Obr.12 Ukážka používateľského rozhrania  
Fig. 12 Demonstration of the user interface

Vytvorená aplikácia je spustiteľná na operačných systémoch Windows XP a vyššie, pričom k dispozícii musí byť aj (.XML) súbor „slova.xml“ s vytvoreným slovníkom a (.DLL) knižnica „Microsoft.VisualBasic.PowerPacks.Vs.dll“. Vytvorená SW implementácia bola otestovaná na limitovanej vzorke osôb rôzneho veku. Výsledky sú zhrnuté v tab. 1.

Pohlavie - vek	Prijaté príkazy	Príkazy prijaté po dotazovaní	Neprijaté príkazy	Úspešnosť [%]
Žena – 21	26	3	9	0,763
Muž – 24	20	2	8	0,733
Muž – 24	22	1	6	0,793
Muž – 25	34	5	8	0,829
Žena – 47	20	3	4	0,852
Muž – 51	22	4	10	0,722
Žena – 78	17	3	7	0,741

Tab.1 Výsledky testovania

## Záver

Vytvorené rozhranie opísané v tomto článku predstavuje prototyp, ktorý dokáže spracovávať a vyhodnocovať príkazy v prirodzenej reči. Prototyp pracuje s viacerými miestnosťami hypotetického inteligentného domu - obývačkou, spálňou, detskou izbou, kuchyňou, kúpeľňou, WC, chodbou a garážou. Pri zadávaní príkazov sa berie do úvahy aj pozícia používateľa a vonkajšia teplota.

## Literatúra

[1] STN EN ISO 9241-x: Ergonomické požiadavky na kancelárske práce so zobrazovacími terminálmi. SÚTN Bratislava

[2] Rozhranie človek-stroj (Prehľadová štúdia). Máj 2009 Dostupné na: [http://www.kasr.elf.stuba.sk/predmety/dasr/viz/Studia\\_2009\\_02.pdf](http://www.kasr.elf.stuba.sk/predmety/dasr/viz/Studia_2009_02.pdf)

[3] TRABALKA, M.: Spracovanie prirodzeného jazyka. KAIT. Fakulta elektrotechniky a informatiky. Bratislava: Slovenská technická univerzita, 1988. Dostupné na: <http://www2.fiit.stuba.sk/~bielik/courses/zs-slov/clanok/ref-new98/trabalka/index.html#JazykoveJednotky>

[4] FERNÁNDEZ, M., MONTALVÁ, J. B., CABRERA-UMPIERREZ, M. F., ARRENDONDO, M.T.: Natural Language Interface for Smart Homes. C. Stephanidis (Ed.): Universal Access in HCI, Part II, HCI 2009, LNCS 5615, Springer-Verlag Heidelberg Berlin, 2009, s. 49–56

[5] MICHALOVIČ, D.: Návrh a implementácia inteligentného používateľského rozhrania inteligentnej budovy. Diplomová práca č. 28260220132015, KRIS EF ŽU v Žiline, 2013, 60 s.

## Abstract

The paper deals with design of a smart user interface for commanding the smart house in the form of a natural language. At first available ways of natural speech processing are being theoretically described. Then authors present design and implementation of the model of the smart user interface for the particular hypothetic smart house.

## Ing. Daniel Michalovič

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
E-mail: [d.michalovic@gmail.com](mailto:d.michalovic@gmail.com)

## Ing. Tomáš Miklušček

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
E-mail: [tomas.mikluscak@fel.uniza.sk](mailto:tomas.mikluscak@fel.uniza.sk)

## prof. Ing. Aleš Janota, PhD.

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
Fax: +421-41- 513 1515  
E-mail: [ales.janota@fel.uniza.sk](mailto:ales.janota@fel.uniza.sk)



# Virtuálny 3D model inteligentnej budovy

Matúš Kostka, Tomáš Miklušček, Aleš Janota

## Abstrakt

Článok opisuje krok za krokom postup návrhu interaktívneho 3D modelu inteligentnej budovy. Vstupné dáta sú do modelu zadávané v podobe textového súboru obsahujúceho scenáre, ktoré reprezentujú pseudonáhodný pohyb osoby po budove, spolu s informáciami o stave osvetlenia a teplote. Výstupná aplikácia je v podobe .EXE súboru.

**Kľúčové slová:** virtuálny 3D model, inteligentná budova, interaktivita, Blender, Python

## Úvod

Pri návrhu zložitých systémov často vzniká potreba tvorby modelu, ktorý by poslúžil na simuláciu a skúmanie rôznych stavov či procesov odohrávajúcich sa vo vyvíjanom systéme, na vizualizáciu budúcich riešení, na prezentáciu vybraných aspektov návrhu zákazníčkovi a pod. 3D vizualizácia je dôležitým nástrojom pre vedcov, inžinierov a technikov, ktorí môžu vytvárať modely, ktoré by inak nebolo možné pozorovať ľudským okom. K dispozícii je pre tento účel množstvo podporných softvérových nástrojov. Tento článok opisuje krok po kroku spôsob tvorby interaktívneho 3D modelu inteligentného domu (budovy) určeného na vizualizáciu dát získavaných simuláciou iného modelu a opisujúcich pseudonáhodný pohyb človeka v rámci budovy, ovládanie osvetlenia a riadenie teploty. Vstup do modelu má podobu textového súboru obsahujúceho opis rôznych scenárov.

## 1. Voľba prostredia pre prácu s 3D grafikou

K dispozícii je v súčasnosti rad softvérových nástrojov umožňujúcich modelovanie, vykresľovanie 3D počítačovej grafiky a animácie. Okrem štandardných odlišností, akými sú cena, typ verzie či platforma, na ktorej je nástroj spustiteľný (tab. 1), možno nájsť hlavné odlišnosti v dostupnosti knižníc, type použiteľných objektov (plochy, krivky, polygóny, ...), type súborových formátov pre export a import, podpore interaktivity aplikácií, a pod.

Pre návrh prezentovaný nižšie bolo zvolené prostredie Blender [1], a to z viacerých dôvodov:

- open-source charakter;
- multiplatformový;
- podpora interaktivity (implementácia výkonného Game Engine s fyzikálnym Bullet enginom [3] podporujúcim simuláciu dynamiky pevných telies);
- funkčná rozšíriteľnosť pomocou programovacieho jazyka Python [2];
- silná podpora formátov pre import a export;
- komplexnosť;
- animovateľnosť objektov;
- prispôsobiteľnosť používateľského prostredia.

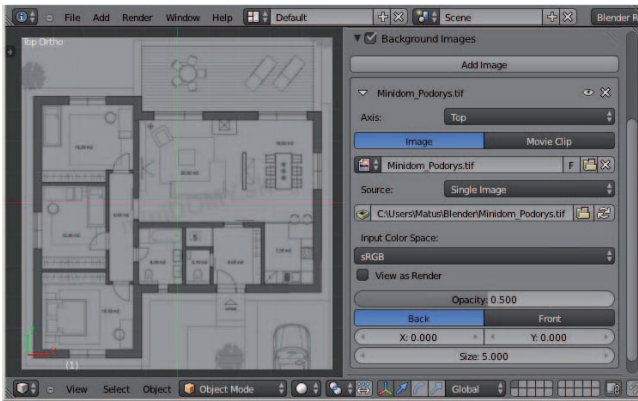
Článok prezentuje výsledky návrhu v podobe ako boli prezentované v rámci kvalifikačnej práce [4].

SW	Licencia / orientačná cena	Platforma	Vývoj
Blender	Freeware zdarma	Microsoft Windows, Mac OS, Linux, BSD, Solaris	Blender Foundation
3Ds Max	Proprietárny SW / 3000€	Microsoft Windows, Mac OS, Linux	Autodesk
Rhinoceros 3D	Proprietárny SW / 1000€	Microsoft Windows, Mac OS	Robert McNeel & Associates
Cinema 4D	Proprietárny SW / 2500€	Microsoft Windows, Mac OS, Linux	Maxon
SketchUp Free	Freeware zdarma	Microsoft Windows, Mac OS	Trimble
SketchUp Pro	Proprietárny SW / 400€		

Tab.1 Prehľad vybraných grafických nástrojov

## 2. Tvorba 3D modelu budovy

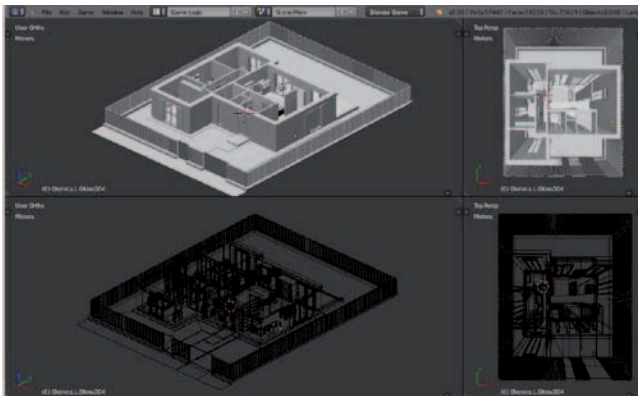
Ako už bolo vyššie uvedené, interaktívny 3D model bol vytváraný ako nástroj na vizualizáciu dát opisujúcich pohyb človeka a stav osvetlenia a teploty v inteligentnom dome. V prvom kroku preto bolo potrebné vytvoriť samotnú budovu, v druhom kroku ju doplniť o funkcie, ktoré zabezpečia interaktívnosť a umožnia používateľovi zasahovať do bežiackej simulácie. Pre čo najpresnejšie vykreslenie budovy bol na začiatku do scény vložený pôdorys budovy vo forme obrázku, aby bola k dispozícii predloha pre tvorbu 3D modelu (obr. 1). Z hľadiska nastavení bola priehľadnosť (*Opacity*) definovaná hodnotou 0,5 a zvolená viditeľnosť z horného pohľadu (položka *Axis*). To umožnilo presnú editáciu plôch (stien budovy) do podoby zodnej s vloženým pôdorysom – najskôr ako 2D obraz podľa osí X a Y (obkreslenie pôdorysu s využitím funkcií *Edge select*, *Vertex select a/alebo Face select*), následne ako 3D obraz rozpínaním plôch (*Extrude*) do výšky podľa osi Z.



Obr.1 Nastavenie viditeľnosti a priehľadnosti obrázku pozadia – pôdorysu budovy

Fig. 1 Adjusting of visibility and opacity of the background – ground plan of the building

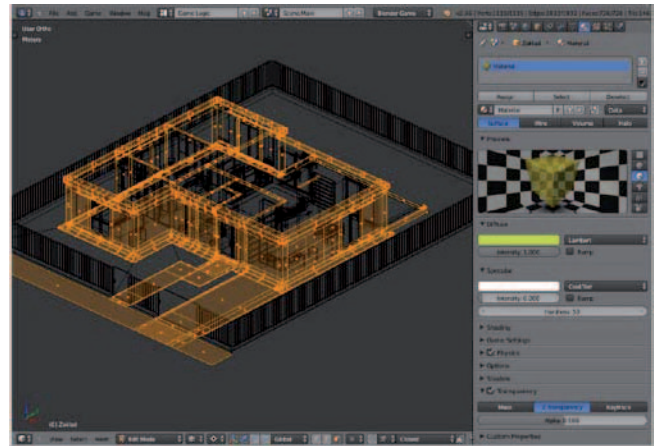
Ďalším čiastkovým krokom bolo zakreslenie okien, dverí a oplotenia ako samostatných objektov, ktoré bude možné animovať (zobrazovať ich pohyb). Pre získanie realistickejšieho obrazu boli pridané aj objekty reprezentujúce nábytok a domáce spotrebiče. Keďže pohyb postavy človeka by mal byť vizualizovaný len vo vnútri jednopodlažnej budovy, strecha nebola do 3D modelu zakomponovaná – znížilo sa tým celkové množstvo objektov v 3D modeli a tým aj zaťaženie hardvérových prostriedkov počítača (CPU, RAM, GPU). Prvý krok smerujúci k vytvoreniu 3D modelu budovy tak vyústil do podoby na obr. 2.



Obr.2 Celistvý (hore) a spojnicový (dole) pohľad na vytvorený 3D model budovy

Fig. 2 Solid (top) and wireframe (bottom) view at the created 3D model of the building

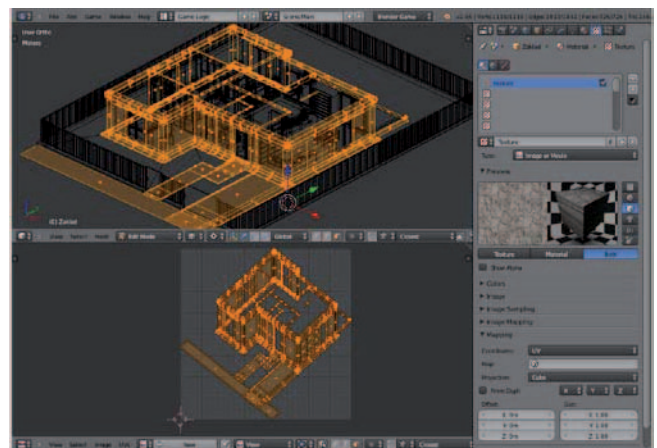
Výsledkom tejto fázy návrhu bol model pozostávajúci iba z množstva navzájom prepojených bodov, ktorý bol navyše jednofarebný, s nedostatočnou odlišiteľnosťou jednotlivých častí pre účely vizualizácie. Preto nasledoval ďalší krok – priradenie materiálov a textúry jednotlivým plochám. Pod materiálom sa rozumie vlastnosť pridávaná povrchu objektu, napr. farba, odraz svetla alebo tieňovanie objektu. To si vyžaduje zaistenie viditeľnosti povrchu, t.j. jeho osvetlenie, preto boli do modelu pridané dva objekty *lamp* simulujúce vyžarovanie svetla a osvetľujúce scénu v 3D modeli. Až potom sa mohlo prikrčiť k priradovaniu materiálov všetkým plochám objektov, ktoré budú pre užívateľa vo vizualizácii viditeľné.



Obr.3 Priradenie a nastavenie materiálu plochám objektu

Fig. 3 Assignment and adjusting of materials to object surfaces

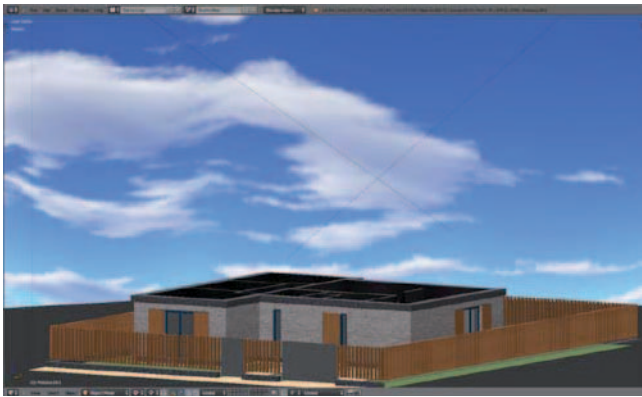
Výsledný tvar modelu získal vlastnosť lepšej rozlíšiteľnosti jednotlivých častí, nepôsobil však príliš realisticky. Preto sa prikrčilo k definícii textúr. Textúra je 2D objekt pokrývajúci povrch 3D objektu, používaný na dosiahnutie maximálnej podobnosti povrchu so skutočnosťou. Dôležitým parametrom textúry je rozlíšenie (veľkosť v pixeloch). Čím väčšia je veľkosť textúry, tým realistickejšie bude povrch objektu pôsobiť. So zvyšujúcou sa veľkosťou textúry však rastú požiadavky na výpočtový výkon počítača pri spracovaní (renderovaní). V našom prípade sa použili pre textúry obrázky s rozlíšením 64 x 64 pixelov (textúra dreva pre dvere, oplotenia a nábytok, textúra trávy pre trávnik, a pod.). S výhodou bol použitý nástroj tzv. UV mapovania, aby sa textúra zobrazovala korektné na rôzne natočených plochách (U a V je označenie súradníc textúry). Proces UV mapovania a aplikovania textúr je znázornený na obr. 4.



Obr.4 UV mapovanie, priradenie a nastavenie textúry plochám objektu

Fig. 4 UV mapping, assignment and adjusting of textures to object surfaces

Vyššie opísanými postupmi sme získali realistickejší vzhľad modelu, avšak za cenu zvýšenia nárokov na výpočtový výkon. Výsledný 3D model po aplikovaní textúr možno vidieť na obr. 5.



**Obr.5 Výsledný 3D model budovy po aplikovaní textúr**  
**Fig. 5 The final 3D model of the building after textures application**

### 3. Interaktivita 3D modelu

Aby bol 3D model interaktívny, je potrebné, aby ho mohol používateľ určitým spôsobom ovplyvňovať. V našom prípade preto bolo potrebné 3D model rozšíriť o funkcie, ktoré by umožňovali ovládať pohyb postavy človeka po jednotlivých miestnostiach budovy, rozsvetovať, resp. zhasínať svetlá v miestnostiach, otvárať, resp. zatvárať dvere a meniť rôzne pohľady na 3D model spolu so zobrazovaním informačného panelu s textovými informáciami a orientačnou mapou. Ovládanie vizualizácie sa predpokladalo vstupmi z vybraných tlačidiel klávesnice a obsahom textového súboru.

Hlavnými komponentmi programu Blender na tvorbu interaktívnych aplikácií boli programovací jazyk Python a vstavaný Blender Game Engine. Tieto komponenty umožňujú ľubovoľne ovládať všetky objekty a ich vlastnosti. V našom prípade sa použil Blender ver. 2.66a podporujúci použitie programovacieho jazyka Python verzie 3.3.

#### 3.1 Zdroj údajov pre ovládanie modelu

Na ovládanie interaktívneho 3D modelu sa použila šablóna (scenár), ktorá má podobu textového súboru (obr. 6).

```
0,86400,75,24,13,0000000000000000
1,54000,110,26,18,1111111111111111
1,86400,75,24,13,0000000000000000
2,54000,110,26,18,1111111111111111
2,3964,26,29,14,0000000000000000
2,1234,75,28,15,0100000000000000
3,12345,110,32,16,0010000000000000
3,521,112,11,17,0001000000000000
4,54000,75,26,18,0000100000000000
4,1234,110,28,15,0000010000000000
5,12345,26,32,16,0000001000000000
6,521,75,30,17,0000000100000000
6,54000,75,26,18,0000000100000000
```

**Obr.6 Štruktúra údajov v textovom súbore**

**Fig. 6 The structure of data in the text file**

Údaje v textovom súbore sú ľahko čitateľné a majú jednoduchú štruktúru totožnú so súbormi formátu CSV (*Comma Separated Values*), ktoré pozostávajú z ľubovoľného počtu riadkov, oddelených znakom nového riadka. Každý riadok obsahuje stĺpce, ktoré sú oddelené čiarkou alebo tabulátorom. Základom sú simulačné kroky, ktoré sú v súbore umiestnené do jednotlivých riadkov. Každý simulačný krok obsahuje údaj o simulačnom dni, čase v sekundách, pozícii postavy, teplote v danej miestnosti, vonkajšej teplote a stave všetkých 16 svetiel. Každý simulačný krok si preto vyža-

duje znalosť 6 hodnôt, počet simulačných krokov nie je obmedzený.

Pre načítanie textového súboru bol vytvorený nový skript nazvaný `pohyb.py` a obsahujúci funkciu s názvom `_citajTXT(nazov_souboru)`, ktorej vstupným parametrom je názov zdrojového textového súboru. Hlavnou časťou uvedenej funkcie je preddefinovaná funkcia jazyka Python - `csv.reader`. Funkcia načíta z textového súboru jednotlivé riadky do jednotlivých položiek (riadkov) jednorozmerného zoznamu. Potom sa odstránia oddeľovacie znaky (čiarky), čím sa jednotlivé položky (riadky) rozdelia na ďalšie položky (stĺpce). Tým vznikne dvojrozmerný zoznam, v ktorom sú údaje zo súboru uložené. Prvý rozmer určuje simulačný krok, druhý rozmer určuje daný údaj v danom simulačnom kroku. Funkcia tak v princípe prevedie číselné hodnoty v textovom súbore na hodnoty dvojrozmerného poľa, s ktorým sa ďalej jednoduchšie pracuje. Názov textového súboru použitého ako zdroj dát pre vizualizáciu bol pevne zadefinovaný ako `3Dmodel.txt`.

```
def _citajTXT(filename):
    #Vytvorenie prázdneho jednorozmerného zoznamu riadok
    riadok = []
    #Otvorenie súboru pre čítanie
    f = open(filename)
    #Každý riadok súboru sa uloží do položky zoznamu (riadok)
    prem = csv.reader(f, delimiter=",")
    #Každá položka zoznamu (riadok) sa rozdelí sa stĺpce
    for stlpec in prem:
        riadok.append(stlpec)
    #Výstupom je dvojrozmerný zoznam
    return riadok
```

**Obr.7 Skript pohyb.py - funkcia \_citajTXT(súbor)**

**Fig. 7 The skript pohyb.py - function \_citajTXT(filename)**

#### 3.2 Pohyb postavy človeka

Zobrazovanie pohybu postavy človeka po budove je jednou z hlavných funkcií vizualizácie. Od jej pozície závisí, v akom stave budú ostatné ovládané objekty ako svetlá a obsah textových polí na informačnom paneli. Po načítaní údajov jedného simulačného kroku sa čaká, kým postava dosiahne aktuálnu pozíciu, ktorá je určená údajom v súbore. Po jej dosiahnutí sa načítajú údaje ďalšieho simulačného kroku.

Aby bola možná navigácia pohybujúcej sa postavy v modelovanom priestore, bola plocha modelovanej budovy rozdelená mriežkou a jednotlivé štvorce boli očíslované číslami 1 až 118 (obr. 8). Každý štvorec predstavuje možnú pozíciu postavy. Vytvorený objekt postavy človeka pri pohybe mení pozíciu vzhľadom k osi X a Y (vodorovné osi) a rotáciu vzhľadom k osi Z (zvislá os). Pre samotný pohyb postavy je dôležité poznať všetky miesta v budove, kde sa môže postava nachádzať. Preto bola celá plocha budovy rozdelená mriežkou, kde každý štvorec predstavuje jednu možnú pozíciu postavy. Pozície boli označené poradovými číslami od 1 po 118 (obr. 8.). Na realizáciu samotného pohybu postavy po jednotlivých pozíciách sa využili možnosti nástroja Blender, ktorý vie pre objekt nájsť cestu k danému cieľu a na navigáciu využíva tzv. navigačnú sieť (angl. *Navigation Mesh*). Ide o graf, ktorý opisuje, odkiaľ kam sa možno s daným objektom (v tomto prípade s postavou človeka) dostať

V objektovom móde (*object mode*) sa označili všetky miestnosti. Potom sa v okne *Properties* v záložke *World* tlačidlom *Build navigation mesh* vytvorila navigačná sieť, ktorú pre hľadanie cesty využíva aktuátor typu *Steering*, v ktorom je možné nastaviť rôzne parametre pohybu (napr. rýchlosť, zrýchlenie, cieľ, ...). Pretože však tento aktuátor nájde cestu vždy len pre jeden cieľ a po jeho dosiahnutí sa deaktivuje, bolo potrebné zabezpečiť postupné prepisovanie cieľov

jeden za druhým. Pre tento účel boli v skripte pohyb.py vytvorené funkcie init() a hlavna().



Obr.8 Rozloženie a označenie miest pre navigáciu postavy v 3D modeli

Fig. 8 Distribution and labelling of places for the purpose of person navigation within the 3D model

```
def init():
    global data, m, vzdialenost, rychlost, p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, p14, p15, p16, pom
    data = citejTXT('3dmodel.txt') #Načítanie dát zo súboru do zoznamu 'data'
    cont = bge.logic.getCurrentController()
    own = cont.owner
    scene = logic.getCurrentScene() #Načítanie aktuálnej scény
    scenes = bge.logic.getSceneList() #Načítanie zoznamu všetkých scén
    m = 0 #Počiatočný riadok na čítanie zo zoznamu
    vzdialenost = 0.0 #Vzdialenosť od cieľa
    rychlost = 1 #Hraničná rýchlosť
    #Počiatočný stav svetiel 1-16 nastavený na zhasnutie
    pom = "0000000000000000"
    p1 = "svietil"
    ...
    #Načítanie scény Scene.HUD pre zobrazenie informácií
    hud = [scene for scene in scenes if scene.name=="Scenes.HUD"][0]
    ciselnik01 = hud.objects['Datum'] #Deň
    ciselnik02 = hud.objects['Cas'] #Čas
    ciselnik03 = hud.objects['TeplotaVon'] #Vonkajšia teplota
    ciselnik04 = hud.objects['TeplotaDnu'] #Vnútorná teplota
    ciselnik01.text = "" #Počiatočná hodnota textového poľa Deň
    ciselnik02.text = "" #Počiatočná hodnota textového poľa Čas
    ciselnik03.text = "" #Počiatočná hodnota textového poľa Vonkajšia teplota
    ciselnik04.text = "" #Počiatočná hodnota textového poľa Vnútorná teplota
    cube = scene.objects['Postava.001']
    cube.worldPosition = [0.0, 0.0, 0.0] #Predvolená pozícia postavy človeka
    cube.worldOrientation = [0.0, 0.0, 0.0] #Predvolená rotácia postavy človeka
```

Obr.9 Skript pohyb.py – funkcia init()

Fig. 9 The script pohyb.py – function init()

Funkcia init() (obr. 9) slúži na načítanie údajov z textového súboru do premennej s názvom data. Funkcia init() tiež nastaví všetky používané premenné na predvolené hodnoty. To znamená, že sa vynuluje počítadlo simulačných krokov (m), stav všetkých svetiel sa nastaví tak, aby boli zhasnuté, vymaže sa obsah textových polí a nastaví sa počiatočná pozícia a rotácia postavy človeka. Ku spusteniu tejto funkcie dôjde vždy len pri spustení vizualizácie, resp. jej reštartovaní.

```
def hlavna():
    global data, m, vzdialenost, rychlost, pom
    cont = bge.logic.getCurrentController()
    own = cont.owner
    scene = logic.getCurrentScene() #Načítanie aktuálnej scény
    scenes = bge.logic.getSceneList() #Načítanie zoznamu všetkých scén
    #Načítanie scény - Scene.HUD pre zobrazenie informácií
    hud = [scene for scene in scenes if scene.name=="Scenes.HUD"][0]
    ciselnik01 = hud.objects['Datum'] #Deň
    ciselnik02 = hud.objects['Cas'] #Čas
    ciselnik03 = hud.objects['TeplotaVon'] #Vonkajšia teplota
    ciselnik04 = hud.objects['TeplotaDnu'] #Vnútorná teplota
    ciselnik04.color = (0,0,1)
    ciselnik04.color = (1,0,0)
    Kracaj = cont.actuators['Steering'] #Načítanie aktuátora postavy
    posta = cont.actuators['sprava'] #Načítanie aktuátora
    if m < len(data):
        #Načítanie dát do jednotlivých textových objektov
        ciselnik01.text = data[m][0]
        ciselnik02.text = _getInMM5(int(data[m][1]))
        ciselnik03.text = data[m][3]
        ciselnik04.text = data[m][4]
        pom = data[m][5] #Stav svetiel
        Kracaj.target = scene.objects[data[m][2]] #Aktuálny cieľ
        cont.activate(kracaj) #Aktivovanie aktuátora 'steering' - aktivovanie pohybu
        #Kontrola aktuálnej rýchlosti a vzdialenosti od cieľa
        Xspeed, Yspeed, Zspeed = own.getLinearVelocity(False)
        linSum = Xspeed + Yspeed + Zspeed
        akt_rychlost = linSum
        if akt_rychlost < rychlost and own.getDistanceTo(data[m][2]) < vzdialenost:
            m += 1
        #Kontrola dostupnosti dát pre vizualizáciu
        if m == len(data):
            posta.subject = 'restart'
            cont.activate(posta)
```

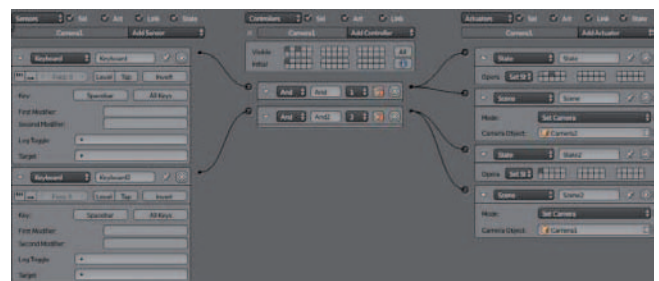
Obr.10 Skript pohyb.py – funkcia hlavna()

Fig. 10 The script pohyb.py – function hlavna()

Funkcia hlavna() slúži na aktualizáciu simulačného kroku (obr. 10), tzn. vždy keď postava dosiahne daný cieľ, aktualizuje sa cieľ pre pohyb postavy, stav svetiel a informácie o čase a teplote. Dosiachnutie cieľa sa testuje tak, že počas pohybu postavy sa sleduje jej rýchlosť a (priamočiara) vzdialenosť od cieľa. Ak je rýchlosť a súčasne vzdialenosť menšia, ako je to uvedené premennými vzdialenost a rychlost vo funkcii init(), aktualizuje sa simulačný krok. Okrem toho sa testuje, či aktuálny simulačný krok nie je posledný. Ak áno, odošle sa správa „restart“ a načíta sa funkcia init(), ktorá nastaví všetky používané premenné na predvolené hodnoty.

### 3.3 Prepínanie pohľadov na model

Aby bolo možné vo vizualizácii prepínať klávesom medzi dvoma rôznymi pohľadmi na 3D model, boli do scény vložené dve kamery. Jedna kamera (Camera1) simuluje pohľad osoby, druhá kamera (Camera2) bola umiestnená nad postavu s cieľom snímať ju z horného pohľadu. Ako „rodič“ bol pre obidve kamery zadaný objekt postavy, takže kamery súčasne s postavou človeka menia svoju pozíciu na osi X a Y a vykonávajú rotáciu podľa osi Z. Na prepínanie medzi obidvoma pohľadmi sa používa kláves „SPACE“.



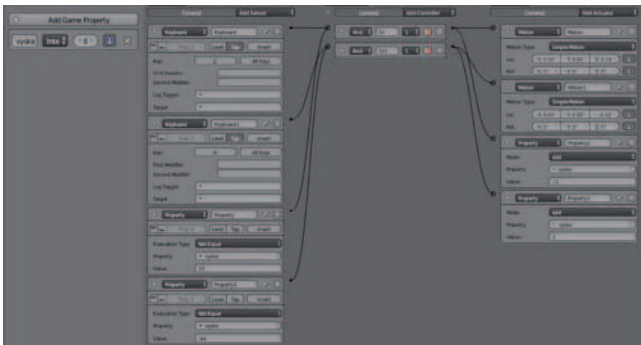
Obr.11 Prepínanie medzi 2 kamerami - zapojenie logických blokov

Fig. 11 Switching-over between 2 cameras - connection of logical blocks

Pri realizácii prepínania sa použilo zapojenie logických blokov vstavaného Blender Game Engine. V jednom stave sa na sníma obrazu scény objektom Camera1 a v druhom stave objektom Camera2 (obr. 11).

Pri hornom pohľade je potrebné meniť výšku objektu Camera2, na čo sa opäť využilo zapojenie logických blokov (obr.

12). Senzor typu Keyboard je zapojený cez blok typu AND na aktuátor typu Motion. Stlačením klávesu „Z“ sa objekt *Camera2* pohne smerom hore (horný pohľad sa oddiali), stlačením klávesu „A“ sa objekt *Camera2* pohne smerom dolu (horný pohľad sa priblíži). Aby nebolo možné pri približovaní postupne prejsť modelom budovy a pri oddaľovaní sa vzdialiť natoľko, že by model budovy nebol viditeľný, pridala sa ešte vlastnosť (*Add Game Property*), v ktorej sa ukladá aktuálna hodnota približenia a pomocou tejto hodnoty je možné sledovať mieru približenia. Vlastnosť bola pomenovaná *vyska* a jej typ nastavený na *integer* a počiatočnú hodnotu na 0. Každému z pohybov (priblíženie a oddialenie) kamery bol pridaný senzor typu *Property* a aktuátor typu *Property*.

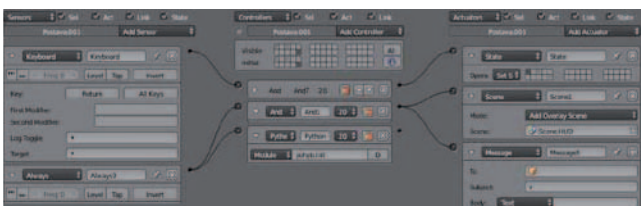


Obr.12 Zmena výšky kamery - zapojenie logických blokov

Fig. 12 Change of the camera height - connection of logical blocks

### 3.4 Ovládanie stavu vizualizácie

Vizualizácia modelu sa môže nachádzať v jednom z troch stavov: spustená, pozastavená a reštartovaná. Aktuálny stav sa zobrazuje na informačnej lište. Prechody medzi spusteným a pozastaveným stavom sa odvíjajú od opakovaného stlačenia klávesu „ENTER“. V spustenom alebo pozastavenom stave je možné kedykoľvek klávesom „R“ prepnúť vizualizáciu do reštartovaného stavu. Na ovládanie sa použili logické bloky zapojené v objekte postavy človeka a čiastočne aj funkcia *hlavna()* v skripte *pohyb.py*. Po spustení aplikácie (stav reštartovaná – obr. 13) sa spustí funkcia *init()* a čaká sa na kláves „ENTER“, ktorý nastaví druhý vnútorný stav objektu a vizualizácia sa spustí (obr. 14).



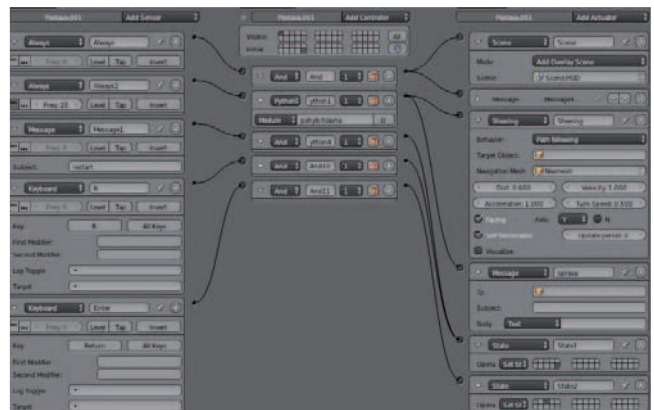
Obr.13 Zapojenie logických blokov pre stav reštartovanej vizualizácie

Fig. 13 Connection of logical blocks for the state of restarted visualization

### 3.5 Ovládanie dverí

Aby sa postava mohla jednoducho a plynule pohybovať medzi jednotlivými miestnosťami, bolo zvolené automatické otváranie resp. zatváranie dverí v závislosti od priblíženia postavy k, resp. vzdialenia od nejakého objektu dverí. Pre každé dvere preto musela byť vytvorená animácia (trvanie každej animácie bolo nastavené na 20 snímok, t.j. pri snímkovacej frekvencii aplikácie 25 snímok/s v trvaní necelých 1s). Na časovej osi sa najprv nastavil počiatočný snímok pre

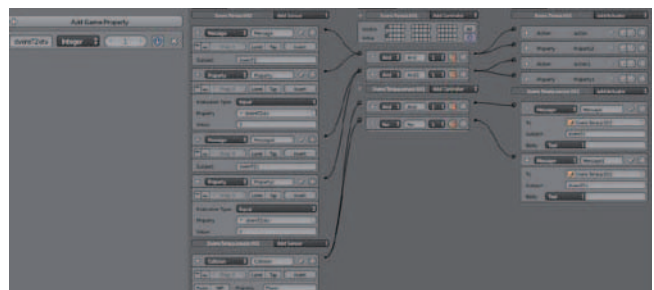
animáciu dverí, potom sa nastavila poloha zavretých dverí a pridal „kľúčový snímok“ (kláves I) pre zmenu polohy a rotácie (*LocRot*). Na časovej osi sa zvýšila hodnota aktuálnej snímky o 20. Nastavila sa poloha otvorených dverí a znova sa pridal ďalší „kľúčový snímok“ (*LocRot*). Aplikácia si sama vygeneruje snímky medzi dvoma „kľúčovými snímkami“.



Obr.14 Zapojenie logických blokov pre stav spustenej vizualizácie

Fig. 14 Connection of logical blocks for the state of running visualization

Pre potreby ovládania sa na podlahu pod každé dvere pridal jednoduchý objekt – plocha (*Plane*), slúžiaci ako senzor sledovania prítomnosti, resp. neprítomnosti postavy človeka. Logický blok sa nastavil tak, aby reagoval s objektom obsahujúcim vlastnosť (*Property*) s názvom *Player*, čiže s postavou človeka. Blok *Collision* sa cez bloky (*Controllers*) AND a NOR prepojil na dva Aktuátory (*Actuators*). Podľa toho, či sa postava plochy/senzora dotýka alebo nie, sa objektu dvere odošle jedna z dvoch rôznych správ (otváranie, keď je hodnota rovná „1“, a zatváranie, keď je hodnota rovná „0“). Konkrétne zapojenie logických blokov pre jednotlivé dvere a ich senzory je na obr. 3.15.



Obr.15 Ovládanie dverí - zapojenie logických blokov

Fig. 15 Door control - connection of logical blocks

### 3.6 Ovládanie osvetlenia miestnosti

Blender Game Engine nepodporuje Ray-Tracing (metódu používanú v 3D grafike na simuláciu zachytávania svetla objektmi, takže svetlo cez objekt neprejde - napr. keď je osvetlená budova zvonku, vo vnútri je tma). Pre dosiahnutie efektu rozsvietenia resp. zhasnutia svetiel v jednotlivých miestnostiach domu sa najskôr každá miestnosť umiestnila na osobitnú vrstvu. Na tieto vrstvy sa tiež umiestnili objekty predstavujúce zariadenie (nábytok a spotrebiče) jednotlivých miestností. V *Edit mode* sa označili vnútorné plochy jednotlivých miestností a vyseparovali sa klávesom *P* do osobitných objektov. Následne sa do jednotlivých miestností pridali svetlá (objekty lamp), spolu 16 týchto objektov. V paneli *Properties* v záložke *Object Data* sa nastavilo, aby svetlá osvetľovali len vrstvu, v ktorej sú umiestnené. Potom sa obdobne ako pri animáciách dverí vytvorili animácie zmeny intenzity osvetlenia jednotlivých svetiel v miestnostiach (obr. 16).



Obr.16 Ovládanie osvetlenia miestností - zapojenie logických blokov

Fig. 16 Room illumination control - connection of logical blocks

Pre jednotlivé objekty lúčov boli v skripte `pohyb.py` vytvorené jednotlivé funkcie (Obr. 17). Funkcia `sv1()` pre lampu 1, funkcia `sv2()` pre lampu 2, atď. Tieto funkcie ovládajú spúšťanie animácie rozsvietenia resp. zhasnutia danej lampy podľa údajov z textového súboru. Python kontrolér je napojený na senzor *Always* a aktuátory *Message* a *Action* (obr. 16). Tzn., že ak je napr. v zdrojovom textovom súbore hodnota stavu danej lampy rovná 1 (rozsvietenie lampy) a zároveň je lampa zhasnutá, spustí sa daná animácia pre rozsvietenie lampy. V opačnom prípade, ak je v textovom súbore hodnota stavu danej lampy rovná 0 (zhasnutá) a zároveň je lampa rozsvietená, spustí sa daná animácia pre zhasnutie lampy. Každá funkcia má premennú (`p1, ..., p16`), kde je uložený aktuálny stav danej lampy („rozsvietená“ alebo „zhasnutá“).

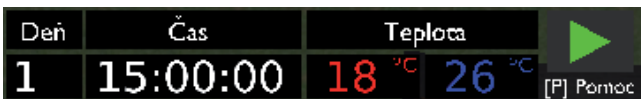
```
def sv1():
    global p1, pom
    cont1 = bge.logic.getCurrentController()
    own1 = cont1.owner
    scene = logic.getCurrentScene() #Načítanie aktuálnej scény
    svetl01 = cont1.actuators['osvetlenie1'] #Načíta aktuátor pre ovládanie animácie
    svetl01.action = 'svetlo.001.Action' #Animácia, ktorá sa bude ovládať
    postal = cont1.actuators['sprava1'] #Načíta aktuátor odoslania správy
    if m < len(data): #Ak sú údaje ...
        if pom[0] == '1' and p1 == 'nesvieti': #Ak sa má svetlo rozsvietiť a ešte nesvieti
            svetl01.frameStart = 430 #Počiatočný snímok animácie
            svetl01.frameEnd = 440 #Koncový snímok animácie
            cont1.activate(svetl01) #Spustí animáciu rozsvietenia
            p1 = 'svieti' #Nastav stav osvetlenia na svieti
            postal.subject = 'svetlo' #Správa pre zapnutie indikácie osvetlenia
            cont1.activate(postal) #Odošlali správu
        if pom[0] == '0' and p1 == 'svieti': #Ak sa má svetlo rozsvietiť a ešte nesvieti
            svetl01.frameStart = 440 #Počiatočný snímok animácie
            svetl01.frameEnd = 430 #Koncový snímok animácie
            cont1.activate(svetl01) #Spustí animáciu zhasnutia
            p1 = 'nesvieti' #Nastav stav osvetlenia na nesvieti
            postal.subject = 'svetlo' #Správa pre vypnutie indikácie osvetlenia
            cont1.activate(postal) #Odošlali správu
```

Obr.17 Skript pohyb.py – funkcia sv1() na ovládanie osvetlenia jednej lampy

Fig. 17 The script pohyb.py – function sv1() for control of one lamp

### 3.6 Informačná lišta

Tento model slúži na vizualizáciu dát, ktoré sú výsledkom simulácie ovládania teploty a osvetlenia inteligentnou budovou. Preto je vo vizualizácii potrebné zobrazovať informácie o dátume, čase, teplote a prípadne stav, v ktorom sa vizualizácia nachádza. Na zobrazenie týchto informácií slúži panel umiestnený v ľavej hornej časti okna. V paneli sa zobrazuje dátum (formát YYYY-MM-DD), čas (formát HH-MM-SS), teplota v miestnosti, v ktorej sa postava nachádza, vonkajšia teplota a symbol označujúci jeden z troch stavov, v ktorých sa vizualizácia nachádza (spustená, pozastavená, reštartovaná) (obr. 18).



Obr.18 Informačná lišta

Fig. 18 The information bar

Pretože scéna s informačnou lištou má prekryvať časť obrazu hlavnej scény, čo vytvorí efekt obrazu v obraze, museli byť informačná lišta a hlavná scéna definované ako dve nezávislé scény. V skripte `pohyb.py` sa do funkcie `hlav-`

`na()` doplnila časť kódu, ktorý textovým objektom priradí v každom simulačnom kroku novú hodnotu vo forme reťazca (obr. 19). Každý z týchto textových objektov zobrazuje informáciu o simulačnom dni, čase a teplote v jednotlivých simulačných krokoch.

```
#Načítanie scény - Scene.HUD pre zobrazenie informácií
hud = [scene for scene in scenes if scene.name=="Scene.HUD"][0]
ciselnik01 = hud.objects['Datum'] #Dátum
ciselnik02 = hud.objects['Cas'] #Čas
ciselnik03 = hud.objects['TeplotaVon'] #Vonkajšia Teplota
ciselnik04 = hud.objects['TeplotaDnu'] #Vnútorná teplota

#Načítanie dát do jednotlivých textových objektov
ciselnik01.text = data[m][0] #Textový objekt pre dátum
ciselnik02.text = _getInHMS(int(data[m][1])) #Textový objekt pre čas
ciselnik03.text = data[m][3] #Textový objekt pre vnútornú teplotu
ciselnik04.text = data[m][4] #Textový objekt pre vonkajšiu teplotu
```

Obr. 19 Skript pohyb.py – časť funkcie `hlavna()` – vloženie údajov do textových objektov

Fig. 19 The script pohyb.py – a part of the function `hlavna()` – data entry to text objects

Všetky hodnoty sa do textových objektov vkladajú v nezmenenej forme, okrem informácií o čase. Zo súboru sa údaj o čase načítal ako počet uplynutých sekúnd daného simulačného dňa. Vo vizualizácii sa ale bude zobrazovať vo formáte HH-MM-SS (hodina-minúta-sekunda). Na prepočet sekúnd na hodiny, minúty a sekundy bola v skripte `pohyb.py` vytvorená jednoduchá funkcia `_HMS(sekundy)` (obr. 20). Funkcia má jeden vstupný parameter, ktorým je počet sekúnd a výstupným parametrom je čas v hodinách, minútach a sekundách. Na prepočet sa používa funkcia `divmod`, určená na delenie so zvyškom. Údaj o čase v sekundách je dátového typu `string` a vstupný parameter tejto funkcie musí byť dátového typu `integer`. Preto som musel údaj o čase vždy najprv prekonvertovať z dátového typu `string` na dátový typ `integer` pomocou funkcie `int()`.

```
def _HMS(sekundy):
    m, s = divmod(sekundy, 60)
    h, m = divmod(m, 60)
    return "%02d:%02d:%02d" % (h, m, s)
```

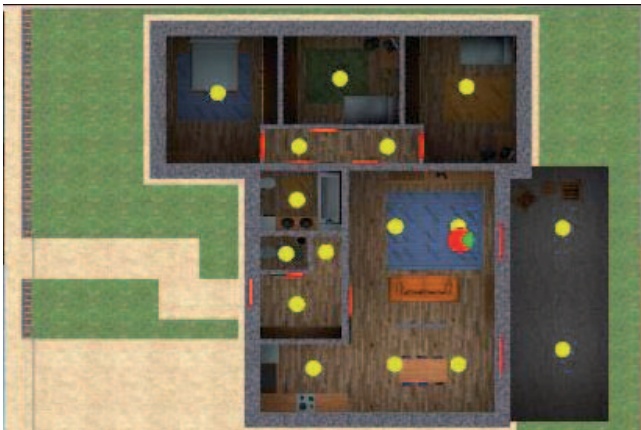
Obr. 20 Skript pohyb.py – funkcia `_HMS(sekundy)` – prepočet sekúnd na hodiny, minúty a sekundy

Fig. 20 The script pohyb.py – the function `_HMM(seconds)`- recalculation of seconds to hours, minutes and seconds

### 3.7 Orientačná mapa

Orientačná mapa je dvojrozmerný pohľad zhora na celú budovu, kde súčasne vidieť všetky miestnosti (Obr. 21). Mapa neposkytuje tak detailné zobrazenie jednotlivých objektov ako hlavný 3D pohľad. Poskytuje však používateľovi informáciu o aktuálnom stave osvetlenia a pozícii postavy človeka v rámci celého objektu budovy. Pritom pri hornom pohľade možno vidieť len časť budovy, resp. blízke okolie postavy človeka. Mapa je umiestnená v pravej hornej časti okna vedľa informačnej lišty.

Zriadenie orientačnej mapy si vyžiadalo vloženie ďalšej kamery v hlavnej scéne (Camera3), ktorá zhora sníma naraz celú budovu. Aby bola postava človeka a stav osvetlenia jednotlivých miestností budovy na mape dobre viditeľné, nad postavu a všetky objekty lúčov sa doplnili objekty v tvare kruhu. Postava je potom na mape zobrazená ako červený kruh so zeleným výrezom, rozsvietenú lampu reprezentuje žltý kruh - ak je lampa rozsvietená, resp. zhasnutá, tak je kruh viditeľný, resp. neviditeľný. Celej ploche sa priradil nový materiál s názvom „mat“ a bola mu priradená textúra, na ktorú sa použil vygenerovaný obrázok `map.png`. Python skript `mapa.py` slúži na to, aby sa pohľad kamery z hlavnej scény ukladal do súboru `map.png`.



Obr. 21 Orientačná mapa

Fig. 21 The orientation map

```
import bpy
from bpy import logic, texture

cont = bpy.logic.getCurrentController()
own = cont.owner
scene_main = None
#Načítanie scény Scene.Main s kamerou, ktorej obraz snímame
for scene in logic.getSceneList():
    if scene.name == 'Scene.Main':
        scene_main = scene
        break
minicam = scene_main.objects['Camera.003'] #Načítanie kamery Camera.003
matID = texture.materialID(own, 'IDmap.png') #Načítanie obrázku pre textúru
map_png = texture.Texture(own, matID) #Načítanie textúry
source = texture.ImageRender(scene_main, minicam) #Použi snímaný obraz ako textúru
map_png.source = source

def main():
    map_png.refresh(True) #Obnovuj snímaný obraz do textúry
```

Obr. 22 Skript mapa.py – obraz snímaný kamerou použitý ako textúra

Fig. 22 The script mapa.py – image shot by the camera is used as a texture

### 3.8 Export aplikácie

Po dokončení všetkých častí 3D modelu a všetkých funkcií vizualizácie sa celý projekt mohol exportovať v súbore s koncovkou `.blend` ako samostatná spustiteľná aplikácia. Tým sa zabezpečilo vytvorenie súboru s koncovkou `.exe`, spustiteľného na operačných systémoch Windows, Linux, Mac OS, BSD, Solaris. Na záver tvorby interaktívneho 3D modelu budovy bolo možné realizovať fázu testovania, ktorá potvrdila správnosť vytvoreného modelu.

### Záver

Cieľom článku bolo krok za krokom opísať postup tvorby interaktívneho 3D modelu inteligentnej budovy slúžiaceho na vizualizáciu dát. Vizualizácia má podobu spustiteľnej aplikácie, ktorá prevádza číselné hodnoty z textového súboru (šablóny) do 3D grafickej podoby, a vychádza z možností softvérového nástroja Blender. V tejto podobe sú výsledky simulácie jasnejšie, čo môže prispieť k lepšiemu pochopeniu skúmaného problému.

### Literatúra

- [1] Blender [online 16. 6. 2013]. Dostupné na: <http://cs.wikipedia.org/wiki/Blender>
- [2] Python - popis jazyka [online 16. 6. 2013]. Dostupné na: <http://programujte.com/clanek/1970010106-python-popis-jazyka/>
- [3] Bullet physics library [online 16. 6 2013]. Dostupné na: <https://code.google.com/p/bullet/>
- [4] KOSTKA, M.: Virtuálny 3d model inteligentnej budovy. Diplomová práca č. 28260220132011, KRIS EF ŽU v Žiline, 2013

### Abstract

The paper describes a step-by-step design of the 3D interactive model of the intelligent building. Input data to the model are in the form of a text file containing scenarios representing a pseudorandom movement of a person in the building, together with information about temperature and lightning. The application is in the form of .EXE file-type.

### Ing. Matúš Kostka

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
E-mail: [matuskostka@gmail.com](mailto:matuskostka@gmail.com)

### Ing. Tomáš Miklušček

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
E-mail: [tomas.mikluscak@fel.uniza.sk](mailto:tomas.mikluscak@fel.uniza.sk)

### prof. Ing. Aleš Janota, PhD.

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 8215/1  
010 26 Žilina  
Fax: +421-41- 513 1515  
E-mail: [ales.janota@fel.uniza.sk](mailto:ales.janota@fel.uniza.sk)

# Bezpečnosť siete ZigBee a jej použitie v inteligentnom dome

Ján Ďurech, Mária Franeková

## Abstrakt

Príspevok je zameraný na oblasť bezdrôtových sietí používaných v inteligentných budovách. Nosná časť je venovaná bezdrôtovej sieti ZigBee, analýze jej bezpečnostných vlastností a opisu jej možností nasadenia v inteligentnom dome (intelligent house). Návrh bezdrôtovej siete pre potreby riadenia v inteligentnom dome je realizovaný pomocou vývojového kitu CC2530ZDK od firmy Texas Instruments. Praktická aplikácia je zameraná na riadenie osvetlenia, kúrenia a externých zariadení. V príspevku sú opísané laboratórne experimenty z pohľadu bezpečnostných útokov na sieť, ako aj potrebné opatrenia proti jednotlivým útokom. Pri realizovaných útokoch na sieť bol použitý program SmarRTF Studio.

**Kľúčové slová:** inteligentný dom, riadenie, ZigBee, bezpečnosť, vývojový kit Texas Instruments, senzor, SW realizácia, linkový kľúč, sieťový kľúč, AES, kryptoanalytické útoky

## Úvod

Posledné dve desaťročia prebieha intenzívny výskum v oblasti inteligentných domov, ktorý je zameraný na nové technológie z oblasti senzorových systémov, metód riadenia, energetického manažmentu za účelom zvýšenia komfortu ľudí. Vložením „inteligencie“ do budov (*intelligent house*) je možné zabezpečiť dôkladnejšiu správu a kontrolu využitia energií. Integrované senzorové systémy vyhodnocujú vnútorné a vonkajšie podmienky budov, ktoré spracovávajú automatizované systémy, čo pozitívne vplyva na zníženie spotreby. Keďže je energia čoraz drahšia, táto problematika púta stále väčšiu pozornosť [1]. Na senzorové systémy pre potreby inteligentnej budovy sú kladené odlišné požiadavky ako na priemyselné senzorové systémy. Zatiaľ čo v priemyselnom prostredí je kladený dôraz na odolnosť voči elektromagnetickému rušeniu, v domácnostiach, alebo kanceláriách sú prioritou nasledujúce požiadavky:

- Cena uzla/ veľký počet uzlov: V inteligentnej budove niekedy potrebujeme stovky uzlov.
- Výdrž batérie: Na dosiahnutie energetickej úspornosti má efekt výber vhodného protokolu.
- Veľké pokrytie: zariadenia sú často rozložené na rozľahlej ploche.

Bezdrôtové siete určené do inteligentnej budovy pozostávajú zo senzorov, pohonov a regulátorov. Vo veľkej väčšine aplikácii potrebujeme prenášať iba malé množstvo údajov, čo sa zohľadňuje vo výbere protokolu. Rozšírené a obľúbené siete ako sú WLAN, alebo Bluetooth sú nevyhovujúce z nasledujúcich príčin:

- Žiadna z týchto sietí nespĺňa požiadavku na potrebnú životnosť batérie, pretože tieto protokoly nepodporujú, resp. podporujú len z malej časti úsporné metódy, ako je napríklad spiaci režim.
- Oba protokoly pracujú najmä v pásme 2,4 GHz ktoré je určené hlavne pre multimedialne aplikácie. Aplikácie v inteligentnej budove nevyžadujú veľkú priepustnosť

dát, čiže im stačí aj nižšie frekvenčné pásmo, napr. 868 MHz s nižším vysielačím výkonom.

T	D [m]	F	R [kb/s]	N	M
6LowPan	75	2,4GHz, 868/915 MHz	250	216	ASK/ BPSK/ O-QPSK
DASH 7	250	433,04-434,79 MHz	27,8	232	FSK/ GFSK
EnOcean	300	868/315 MHz	125	232	ASK
Insteon	50	902-924 MHz	2,9	224	FSK
Knx RF	70	868 MHz	32,7	256	FSK
ONE-NET	70	868/915 MHz	38,4	4096	FSK
Wavenis	1000	400/868/915 MHz	9,6	2000	GFSK
Z-Wave	300	868/908 MHz	9,6	232	FSK
ZigBee	75	2,4GHz, 868/915 MHz	250	264	ASK/ BPSK/ O-QPSK

**Tab. 1 Porovnanie komunikačných technológií používaných v inteligentných budovách**

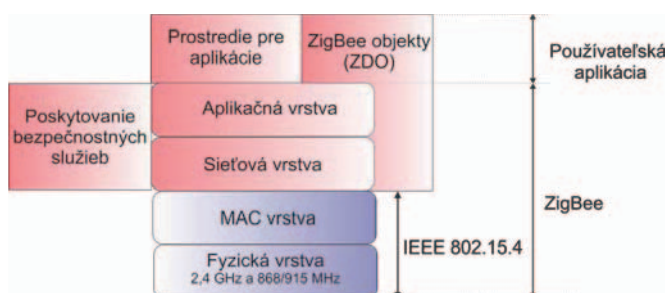
V tab.1 je uvedené porovnanie bezdrôtových technológií používaných v inteligentných budovách z pohľadu dosahu



(D), frekvenčného rozsahu (F), nominálnej rýchlosti (R), počtu použitých zariadení (N) a spôsobov podporovanej modulácie (M).

## 1. Technológia ZigBee a jej vývoj

ZigBee a IEEE 802.15.4 sú štandardy, ktoré poskytujú sieťové rozhranie potrebné pre senzorové siete. Ako je znázornené na obr. 1, štandard IEEE 802.15.4 definuje fyzickú a MAC (Medium Access Control) vrstvu, ZigBee definuje sieťovú a aplikačnú vrstvu. ZigBee Aliancia vznikla v roku 2002 ako nezisková organizácia. Štandard IEEE 802.15.4 bol schválený v roku 2003, (neskôr upravený v roku 2006), ktorý si osvojila ZigBee Aliancia a 14. decembra 2004 bola schválená prvá špecifikácia ZigBee 1.0 [2].



Obr.1 Model siete ZigBee

Fig.1 Model of ZigBee network

Prvá verzia štandardu, známa aj ako ZigBee 2004 bola verejne dostupná od 13. júna 2005. V praxi sa vyskytli mnohé nedostatky [3], preto bol štandard inovovaný. V septembri 2006 bola uvoľnená verzia ZigBee 1.1 (nazývaná aj ZigBee 2006), ktorá znamenala zásadný zlom v nasadzovaní ZigBee technológie v praxi. Veľkou zmenou vo vývoji štandardu bol prechod zo stromovej adresnej štruktúry na spôsob náhodne pridelovaných adries s mechanizmom detekcie kolízií, čo zvýšilo stabilitu a vytvorilo možnosť vytvárania väčších sietí. V októbri 2007 bola vydaná posledná verzia, označovaná aj ako ZigBee 2007 s podporou MESH sietí. Definuje dva profily. Profil jedna, nazývaný aj ako ZigBee je určený pre domáce, jednoduché komerčné použitie s nízkymi nárokmi na pamäť. Profil dva, nazývaný ZigBee PRO ponúka ďalšie funkcie, ako smerovanie *many-to-one*, podporu komunikačného režimu typu *multicast* a vysokú bezpečnosť použitím šifrovanej výmeny kľúčov SKKE (*Symmetric-Key Key Exchange*). Hlavné rozdiely medzi špecifikáciami ZigBee 2006 a ZigBee 2007 sú prídanie fragmentácie a rýchlejšej frekvenčnej preladiťelnosti kanála, ktoré sú potrebné na spätnú kompatibilitu so ZigBee 2006 (v ZigBee 2006 môžu zariadenia v jednej sieti pracovať na rôznych frekvenciách). Fragmentácia je schopnosť spracovávať dáta, ktoré sú väčšie ako je užitočná kapacita rámca. Verzie ZigBee 2006 a ZigBee 2007 sú plne kompatibilné, avšak zariadenia pracujúce na špecifikácii ZigBee 2006 môžu vystupovať iba ako koncové zariadenia. Nemôžu smerovať pakety, ani mať pod sebou z topologického hľadiska iné zariadenia.

### 1.1 Typy zariadení používané v sieti ZigBee

Z logického pohľadu rozdeľujeme zariadenia v sieti ZigBee na:

- **ZigBee Koordinátor ZC** (ZigBee Coordinator) - je primárne zariadenie každej ZigBee siete. Jeho úlohou je smerovanie paketov, správa a monitorovanie siete, plní úlohu centra zabezpečenia (*trust center*). Vytvára tiež

most do iných sietí. V každej sieti je iba jeden koordinátor. Vyžaduje trvalé napájanie.

- **ZigBee smerovač ZR** (ZigBee Router) - slúži na rozšírenie siete. Smeruje pakety medzi koordinátorom a koncovými zariadeniami. Môže tiež plniť funkciu koncového zariadenia. Smerovač je účastník siete a pre koncové zariadenia sa javí ako lokálny koordinátor. Vyžaduje tiež trvalé napájanie.
- **ZigBee koncové zariadenie ZED** (ZigBee End Device) - má najnižšie pamäťové a výpočtové nároky. Dokáže komunikovať iba s koordinátorom, alebo smerovačom. Nedokáže prenášať dáta z iných zariadení. Väčšinou býva napájaný z akumulátorov.

Štandard ZigBee používa tri typy sieťových topológií. Prvou podporovanou topológiou je topológia typu hviezda, v nej je jeden koordinátor a ostatné zariadenia sú vo funkcii koncových zariadení. V topológii strom koncové zariadenia na rozdiel od topológie hviezda nemusia komunikovať priamo s koordinátorom siete, ale môžu využiť smerovač ako sprostredkovateľa. Vďaka tomu je možné zväčšiť vzdialenosti medzi koncovými zariadeniami a koordinátorom. Posledná topológia je typu MESH. Kombinuje vlastnosti topológií strom a hviezda. Táto topológia umožňuje najväčšiu funkčnosť, pretože umožňuje zostaviť sieť ľubovoľným spôsobom [4].

### 1.2 Bezpečnostné mechanizmy siete ZigBee

V bezdrôtových sieťach potrebujeme riešiť dva problémy. Prvým problémom je bezchybné vysielanie dát a ich kontrola integrity, druhým problémom je zabezpečenie dát proti ich odchyťavaniu. Na tieto účely sa v technológii ZigBee používa technika FCS (*Frame Check Sequence*), čo je tvorba kontrolného súčtu rámca a šifrovací štandard AES (*Advanced Encryption Standard*).

ZigBee štandard podporuje používanie nasledujúcich bezpečnostných mechanizmov:

- Šifrovanie dát.
- Verifikácia dát a zariadení.
- Ochrana proti duplicitným rámcem.

V zabezpečenej sieti môže byť použitých päť rôznych kľúčov:

- Linkový kľúč (*link key*) – spoločný medzi dvomi zariadeniami. Môže byť použitý na *unicast* komunikáciu.
- Sieťový kľúč (*network key*) – spoločný pre celú sieť. Používa sa pri *broadcast* komunikácii.
- Hlavný kľúč - používaný pri zavádzaní linkového kľúča medzi dvomi zariadeniami.
- Kľúč pri doprave kľúča (*key-transport key*) - kľúč, ktorý slúži na zabezpečenie prenosu ľubovoľného kľúča z centra zabezpečenia do cieľového zariadenia.
- Načítavací kľúč (*key-load key*) - kľúč používaný na zabezpečený prenos hlavného kľúča.

#### 1.2.1 Kontrolný súčet rámca – FCS

Podľa IEEE 802.15.4 generovanie kontrolného súčtu FSP a jeho kontrola prebieha podľa nasledujúcich krokov:

- Na strane vysielateľa sa z informačnej časti vytvorí polynóm  $M(x)$ , ktorý sa určí z MAC hlavičky paketu a užitočných dát.
- Polynóm  $M(x)$  sa vynásobí najvyššou mocninou generujúceho ireducibilného polynómu  $g(x)$ , ktorý je definovaný:  $g(x) = x^{16} + x^{12} + x^5 + 1$ .
- Takto získaný polynóm sa delí  $g(x)$ , čím vznikne zvyšok po delení  $r(x)$ , ktorý sa pridá k informačnej časti.

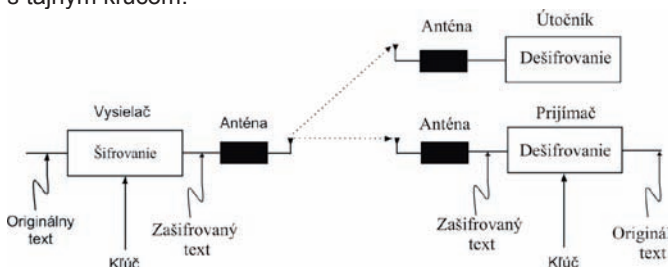
- Na prijímacej strane prebehne znovu delenie prijatého polynómu, odpovedajúcemu prijatým dátam, dohodnutým generujúcim polynómom  $g(x)$ .
- Ak je delenie bezo zvyšku v prijatom pakete nebola detegovaná chyba, ak je delenie so zvyškom paket bol pri prenose narušený.

Podľa poznatkov z teórie kódovania [5] tento spôsob zabezpečenia – generujúcim polynómom 16. rádu, by mal byť schopný detegovať v pakete nielen jednoduchú chybu, ale aj zhuk chýb odpovedajúci stupni generujúceho polynómu. Opisovaný CRC (*Cyclic Redundancy Check*) kód slúži na detekciu chyby prijatého paketu na úrovni linkovej vrstvy. Na verifikáciu údajov, ktorých integrity by mohla byť narušená úmyselne, štandard ZigBee používa hašovaciu funkciu MIC (*Message Integrity Code*).

### 1.2.2 Zabezpečenie dôvernosti

Štandard IEEE 802.15.4 definuje na zabezpečenie dôvernosti prenášaných správ šifrovací štandard AES. Algoritmus AES je 128-bitová bloková šifra s tromi podporovanými dĺžkami kľúča (128, 256, 512 bitov) schválená inštitútom NIST (*National Institute of Standards and Technology*) [6], ktorú používa aj štandard ZigBee s dĺžkou kľúča 128-bitov. Najväčšie obmedzenie v implementácii bezpečnostných funkcií v bezdrôtovej sieti ZigBee sú limitované zdroje. Koncové zariadenia sú prevažne napájané batériou, majú obmedzenú výpočtovú silu a veľkosť pamäte. ZigBee je určená pre lacné aplikácie a hardvér v uzloch nemusí byť zabezpečený. Ak útočník získa prístup k uzlu, ktorý nie je zabezpečený, kľúč môže byť získaný z prístrojovej pamäti [7].

Na obr. 2 je znázornený základný koncept šifrovania v ZigBee sieti na báze symetrického šifrovacieho systému s tajným kľúčom.



Obr. 2 Šifrovacia schéma v sieti ZigBee

Fig. 2 Encryption scheme in ZigBee network

Ako už bolo zdôraznené, ZigBee špecifikuje dva typy kľúčov: linkový a sieťový kľúč. Linkový kľúč sa zdieľa medzi dvoma zariadeniami a sieťový kľúč je spoločný pre celú sieť. Každá zabezpečená ZigBee sieť obsahuje zariadenie nazývané centrum zabezpečenia (*trust center*), ktoré má na starosti distribúciu linkového a sieťového kľúča k iným zariadeniam. Väčšinou úlohu centra zabezpečenia plní koordinátor, no môže to byť aj externé zariadenie, kde ZigBee koordinátor určí adresu centra zabezpečenia [7].

### 1.2.3 Zabezpečenie integrity

Zabezpečenie integrity dát sa dosiahne aplikovaním hašovacej funkcie a pridaním hašovacieho kódu MIC k odosielanému rámcu. Ak sa MIC poskytnuté vysielačom zhoduje s MIC vypočítaným príjemcom, údaje budú považované za autentické. Vyššiu úroveň bezpečnosti dosahujeme zvyšovaním počtu bitov v MIC. Štandardy ZigBee a IEEE 802.15.4 podporujú 32-bitové, 64-bitové a 128-bitové možnosti MIC.

Pozn.: MIC je niekedy označovaný ako MAC\*, ale v ZigBee a IEEE 802.15.4 sa používa viac skratka MIC namiesto

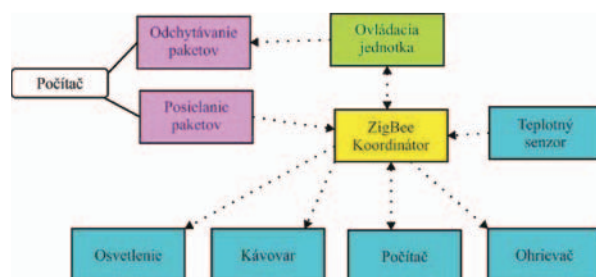
MAC, aby sa zabránilo zámene výrazov MAC\* (*Message Authentication Code*) a MAC (*Medium Access Control*).

MIC je z dôvodu zvýšenia bezpečnosti počítaný v režime CCM\* (*Counter with Cipher block chaining Message authentication code*). Zreťazený režim CCM\* sa používa so 128 bitovým štandardom AES, s kľúčom 128 bitov. AES-CCM\* slúži na šifrovanie dát a zároveň generovanie pridruženého MIC, ktoré je zaslané do prijímača spolu s rámcu. Vo vysielači vstupuje do bloku AES-CCM\* nešifrovaný 128-bitový blok dát. Prijímač používa AES-CCM\* k dešifrovaniu dát a generovaniu vlastného MIC z prijatého rámcu, pre porovnanie s prijatým MIC. CCM\* schéma v sebe kombinuje šifrovanie dát a autentifikáciu.

## 2. Návrh bezdrôtovej siete ZigBee pre potreby riadenia inteligentného domu

Praktická aplikácia bola realizovaná pomocou vývojového kitu CC2530ZDK [8] od firmy Texas Instruments. Vývoj aplikácií spočíval v návrhu a zostrojení vhodných hardvérových doplnkov, naprogramovania aplikácie pre každé zariadenie v sieti, skompilovania a nahratia týchto aplikácií do zariadení. Firma Texas Instrument je jedna zo zakladateľov ZigBee štandardu, má veľkú medzinárodnú podporu produktov, fórum pre riešenie otázok okolo firemných produktov a vývojový kit distribuuje s podprogramami pre ovládanie periférii. Texas Instrument tiež vlastní značku ChipCon, ktorá je výrobcom senzornej dosky CC2430DB [9], ktorá bola tiež použitá v HW a SW realizácii.

Jednoduchá aplikácia je zameraná na riadenie osvetlenia, kúrenia a externých zariadení v inteligentnej budove komunikujúcich podľa obr. 3.



Obr. 3 Koncept realizovanej aplikácie ZigBee siete

Fig.3 Concept of realised application of ZigBee network

Na riadenie bola použitá doska SmartRF05EB. Softvérová realizácia všetkých prvkov bola uskutočnená v programe IAR Embedded Workbench. Pri návrhu bolo potrebné určiť základné parametre siete ZigBee: komunikačný kanál (bol zvolený kanál 25), identifikátor siete PAN ID (bol nastavený na hodnotu 2013) a adresy jednotlivých zariadení (sú uvedené v tab. 2).

Zariadenie	Adresa zariadenia
Koordinátor	0x2520
Ovládač	0x1111
Osvetlenie	0xBEE1
Ohrievač	0x2222
Senzor	0xBEE7
Kávovar	0xBEE6
Počítač	0xBEE5

Tab. 2 Adresy zariadení

Pri realizácii koncových stupňov je použitá doska SmartRF05BB, cez ktorú prebieha komunikácia s riadiacou jednotkou. Pre spínanie výkonového stupňa bolo navrhnuté vlastné zapojenie.

### 3. SW realizácia aplikácií

Pre každé zariadenie v ZigBee sieti bolo nutné vytvoriť samostatný kód. Základná štruktúra tohto kódu sa môže realizovať buď získaním balíčkov od výrobcu ZigBee čipov, alebo pomocou vývojového nástroja, akým je napríklad IAR Embedded Workbench. Pri vytváraní navrhutej ZigBee siete boli použité časti kódov dodávané spolu s kitom. Tieto kódy slúžili na inicializáciu zariadení a riadenie periférií. Dosky boli programované pomocou SmartRF Flash Programmer. Program ovládača je zložený z menu, ktoré slúži používateľovi na ovládanie zariadení.

Realizovaný program na ovládanie definovaných koncových zariadení bol realizovaný v jazyku C, pretože sú v ňom už napísané podprogramy na ovládanie periférií:

```
#define PAN_ID          0x2013
#define SWITCH_ADDR    0x2520
#define LIGHT_ADDR1    0xBEE1
#define SENSOR         0xBEE7
#define OHRIEVAC_ADDR  0x2222
#define OVLADAC_ADDR   0x1111
#define PC_ADDR        0xBEE5
#define KAVOVAR_ADDR   0xBEE6
```

Na začiatku programu boli zadefinované adresy všetkých zariadení, podľa tab. 2. V programe bolo použité skrátené adresovanie, ktoré je výhodnejšie pri menšom počte zariadení. Celý program riadiacej jednotky je založený na spracovávaní prijatých dát zo senzoru, riadiacej jednotky a zabezpečuje komunikáciu s koncovými zariadeniami. Podrobnejšie je v príspevku uvedená len jedna aplikácia - riadenie kávovaru prostredníctvom siete ZigBee. Ostatné aplikácie sú podrobne opísané v [10].

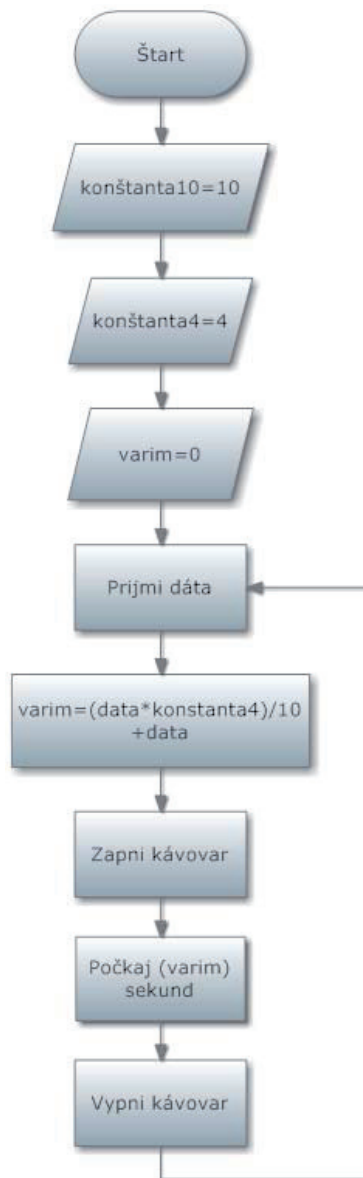
#### 3.1 Príklad realizácie aplikácie – riadenie kávovaru

Princíp riadenia kávovaru spočíva v zopínaní napájania na čas potrebný pre spravenie daného množstva kávy. Po počítačovej analýze časových premenných pre jednotlivé množstvá kávy bolo určené, že za 38 s kávovar spraví 100 ml kávy. V ovládacej jednotke bolo vytvorené menu na nastavenie požadovaného množstva kávy, pozri obr.4.



Obr.4 Displej v riadiacej jednotke kávovaru  
Fig. 4 Display in control unit of coffeemaker

Na displeji v menu sú nastavené dve položky. Veľkosť šálky a počet porcií. Po výbere položiek v ovládacej jednotke sa najprv vynásobí počet porcií s veľkosťou nastavenej šálky, čím dostaneme požadované množstvo kávy, ktorú potrebuje kávovar urobiť. V programe bolo pomocou podmienky ošetrené, aby sa neprekročila maximálna kapacita kávovaru. V ďalšom kroku bol riešený spôsob prenosu. Štandard ZigBee podporuje maximálne 8 bitovú zložku užitočných dát. Preto výsledný údaj bol v ovládacej jednotke kávovaru delený desiatimi a následne násobený príslušnou konštantou a až potom odoslaný koordinátorovi, ktorý ho posiela ku kávovaru. Pribeh programu na ovládanie kávovaru je znázornený pomocou vývojového diagramu obr. 5.



Obr.5. Vývojový diagram ovládania kávovaru  
Fig.5 Flow graph of coffeemaker control

Po prenesení dát z riadiacej jednotky je v jednotke na ovládanie kávovaru vypočítaný príslušný čas potrebný na navařenie daného množstva kávy. Oneskorenie sa realizuje pomocou cyklu „for“ v sekundových intervaloch.

### 4. Výsledky realizácií útokov na sieť ZigBee

Pokiaľ sa podarí útočníkovi odchytiť komunikáciu, získava prístup k celej sieti. Útočníkovi stačí vedieť zdrojovú

a cieľovú adresu a môže začať preposielať odchytené dáta, alebo môže generovať vlastné.

Na odchytyvanie bol použitý špeciálny USB kľúč, ktorý môže byť po pripojení k počítaču použitý na odchytyvanie paketov, pripojenie externých zariadení alebo komunikáciu počítača so ZigBee sieťou. Preposielanie odchyteného nezašifrovaného paketu možno vidieť na obr. 6. Paket bol odchytený medzi ovládacou jednotkou a koordinátorom. Pri preposielaní paketu boli postupne menené poradové číslo paketu (*sequence number*), pričom sa úspešne podarilo ovládať osvetlenie. Pokiaľ by bol paket odchytený viac krát, alebo by boli iba náhodne posielané dáta, tak by sa mohla postupne získať kontrola nad celou sieťou.

#### 4.1 Fyzický útok

Pre zabezpečenie siete vývojový kit CC2530ZDK používa AES šifrovanie so 64 bitovým MIC pre autenticitu dát. Výhodou tohto spôsobu je že do šifrovania sa započítava aj sekvenčné číslo rámca NONCE, čiže šifrovanie dvoch rovnakých textov dáva dva rôzne výsledky. Táto vlastnosť je známa ako sémantická bezpečnosť (*semantic security*). Prelomenie takto šifrovanej siete je takmer nemožné. Nevýhodou siete ZigBee je že všetky heslá uschováva vo svojej pamäti nezašifrované. Pokiaľ sa útočník dostane fyzicky k zariadeniu, môže si ľahko skopírovať pamäť programu do počítača, v ktorej si môže nájsť kľúč.

Realizovaný fyzický útok bol viazaný na jedno zariadenie s USB výstupom, pomocou ktorého bola nahratá celá pamäť zariadenia do počítača. V nej bol nájdený nezašifrovaný kľúč: c0c1c2c3c4c5c6c7c8c9cacbcccdcecf (obr. 6).

```

:1003D0000000233040A3B040B03040C0B040D/304E4
:1003E0000E7B040F8304108B04119304129B0413DF
:1003F000A30414AB0415B30416BB0417C30418CB31
:100400000419D3041ADB041BE3041CEB041DF304DE
:100410001E9C0615FB0401FD0402FF0403010504F4
:100420000305050505060705070905080B05090D60
:10043000050ADE060AC0C1C2C3C4C5C6C7C8C9CA48
:10044000CBCCDCECF0000000010000000A0000A0
:100450000030313233343536373839A14243444540
:100460004600286E756C6C20706F696E7465722919
:100470000030313233343536373839616263646580
:100480006600464C4F4154533F2077726F6E672091
:10049000666F726D617474657220696E7374616CDD
:1004A0006C656421003F3F3F00100000000800002D
:.....

```

Obr.6 Kľúč získaný z pamäte prístroja

Fig.6 Key obtained from memory of equipment

#### 4.2 Rovnaký Nonce útok

Ak však z akéhokoľvek dôvodu zariadenie vyšle dve po sebe idúce správy  $m_1$  a  $m_2$  s rovnakým sekvenčným číslom, útočník má k dispozícii rôzne zašifrované ( $E$ ) dáta  $c_1$  a  $c_2$  rovnakým kľúčom  $key$  a pomocou operácie XOR (1) „odpočúvacie“ zariadenie môže obnoviť čiastočné informácie týkajúce sa pôvodného textu výpočtom:

$$c_1 \oplus c_2 = [m_1 \oplus E(key, nonce)] \oplus [m_2 \oplus E(key, nonce)] = m_1 \oplus m_2. \quad (1)$$

Tento útok je známy ako rovnaký - NONCE útok [11]. Jeden z prípadov, u ktorých sa rovnaký - NONCE útok môže aplikovať, je po výpadku prúdu, ktorý vedie k vynulovaniu akumulátora. Ak je po výpadku napájania posledný NONCE stav neznámy, systém môže resetovať NONCE stav na predvolenú hodnotu. Tento reset zvyšuje pravdepodobnosť opätovného použitia rovnakého NONCE s kľúčom, ktorý bol použitý pred výpadkom. Príklad použitia operácie XOR na originálne dáta po aplikácii útoku, kedy vždy po odoslaní správy bolo resetované zariadenie na originálne dáta 102 (66h) a 103 (67h) je:

$$66 \oplus 67 = 01. \quad (2)$$

Príklad použitia operácie XOR na odchytené dáta:

06 00 00 00 00 C5 15 E5 F9 DF 51 93 C5 58  
je uvedený v (3), kde je možno vidieť vo výsledku získané originálne dáta 01 po aplikovaní operácie XOR.

$$06 \ 00 \ 00 \ 00 \ 00 \ C5 \ 15 \ E5 \ F9 \ DF \ 51 \ 93 \ C5 \ 58 \\ \oplus \ 06 \ 00 \ 00 \ 00 \ 00 \ 51 \ C5 \ 15 \ E5 \ F9 \ DF \ 51 \ 93 \ C5 \ 58 = \\ 00 \ 00 \ 00 \ 00 \ 00 \ \mathbf{01} \ 90 \ f5 \ 83 \ 80 \ eb \ 49 \ 94 \ 09. \quad (3)$$

Pokiaľ by útočník poznal časť dát, vie si ľahko dopočítať chýbajúce. Proti tomuto útoku je možné sa chrániť ukladaním hodnoty sekvenčného čísla v energeticky nezávislej pamäti (*NonVolatile Memory*) a jeho získavanie späť po každom výpadku napájania.

#### 4.2 Denial of Service (DoS) útok

DoS útok spôsobuje odmietnutie všetkých prijatých správ uzlom [11]. Útok sa pokúša zabrániť legitímnemu používateľovi používať sieť. Najbežnejšia metóda na zrealizovanie spočíva v zahltení systému požiadavkami, ktoré spôsobí následný pád systému. Základné typy útokov tohto typu sú zahltenie pásma alebo zahltenie procesora. V testovanej aplikácii bol realizovaný útok na zahltenie procesora.

Útok bol realizovaný pomocou programu SmartRF Studio 7, kde boli najskôr odchytený paket, ktorý bol následne preposielaný rýchlosťou 30 paketov za sekundu. Keďže režim prenosu pracuje na princípe potvrdzovania rámcov, t. j. koordinátor musí na každý paket poslať potvrdzovací rámec, aj keď podvrhnutý paket je hneď po prijatí zahodený, pretože ide o o duplicitný rámec, po 250 s útoku a 7442 odoslaných paketov koordinátor prestal reagovať. Prevenciou proti útoku by bola zmena kanála po prijatí sto duplicitných paketov.

#### 4.3 Replay útok

Jednoduché siete, ktoré nepoužívajú centrum zabezpečenia a celý čas používajú jeden kľúč sú zraniteľné proti útoku pracujúcim na princípe preposielania starých odchytených paketov. Útočníkovi stačí odchytiť paket staršieho dáta a po nejakom čase ho znova preposlať. Útok bol tiež realizovaný pomocou programu SmartRF Studio 7. Bol odchytený šifrovaný paket pre ovládanie osvetlenia. Po následnom preposlaní tohto paketu bolo bez problémov zapnuté osvetlenie. Ochrana proti tomuto útoku spočíva v kontrole sekvenčného čísla na úrovni aplikačnej vrstvy. V programe bolo nadefinované, že každé sekvenčné číslo prijatého paketu, musí byť väčšie ako sekvenčné číslo predchádzajúceho paketu, príp. sekvenčné číslo môže byť nulové, pre prípad reset zariadenia.

#### Záver

Cieľom príspevku bolo uskutočniť analýzu technologických možností realizácie bezdrôtovej siete pre použitie v inteligentnom dome so zameraním na výhody a nevýhody jednotlivých štandardov, pričom bola podrobnejšie opísaná technológia ZigBee. Štandard ZigBee ponúka bohaté riešenia pre monitorovanie, riadenie a obsluhu rôznych zariadení. Jeho výhodou je, že zabezpečuje kompatibilitu veľkého množstva zariadení, podporuje energeticky úspornú komunikáciu s dostatočným zabezpečením. Zároveň však ponúka výrobcovi možnosť vytvorenia vlastných aplikácií. Bola vytvorená aplikácia v pásme 2.4 GHz, ktorá slúži na ukážku riadenia zariadení používaných v inteligentnej budove. Podrobný opis hardvérovej realizácie jednotlivých modulov pre ovládanie zariadení, ako aj opis programu potrebného pre ich riadenie možno nájsť v [10].

V príspevku boli opísané bezpečnostné mechanizmy siete ako je šifrovanie komunikácie, verifikácia údajov a zabezpečenie integrity dát pri prenose. V laboratórnych podmienkach bolo overené, že pri plnom zabezpečení je takmer nemožné sieť prelomiť. Sú uvedené štyri príklady bezpečnostných útokov na sieť ZigBee, ako aj odporúčania ako im predchádzať.

#### PodĎakovanie

*Príspevok vznikol s podporou Edukačnej grantovej agentúry Slovenskej republiky (KEGA) číslo: 024ŽU-4/2012: Modernizácia technológií a metód vzdelávania so zameraním na oblasť kryptografie pre bezpečnostne kritické aplikácie.*

#### Literatúra

- [1] ZigBee Alliance: [online]. In: <http://www.zigbee.org>.
- [2] IEEE 802.15.4: Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs): IEEE Computer Society, 2006. ISBN 0-7381-4996-9.
- [3] GASSMANN, O., MEIXNER, H.: Sensors in Intelligent Buildin, Wiley-VCH Verlag GmbH, 2001, s. 593 ISBN 3-527-29557-7.
- [4] REINISCH, CH.: Wireless Communication in Home and Building Automation: DP. Viedeň: Technische Universität Wien, 2007, s. 116
- [5] MUZIKÁŘOVÁ, L., FRANEKOVÁ, M.: Teória informácií a signálov. Vydavateľstvo ŽU v Žiline, EDIS, 2009 ISBN 978-80-554-0075-4
- [6] FIPS197: Advanced Encryption Standrard. 2001. In: <http://csrc.nist.gov/publications/fips/fips197/fips197.pdf>
- [7] FARAHANI, S.: ZigBee Wireless Networks and Transceivers, Oxford: Newnes, Elsevier Inc, 2008, s. 448, ISBN: 978-0-7506-8393-7.
- [8] CC2530 ZigBee Development Kit User's Guide: In: <http://focus.ti.com/lit/ug/swru209b/swru209b.pdf>
- [9] CC2430DB Demonstration Board User Manual: In: <http://www.ti.com/lit/ug/swru125/swru125.pdf>

- [10] ĎURECH, J. Použitie siete ZigBee v inteligentnej budove: DP. Žilina: Žilinská univerzita v Žiline, 2013
- [11] HERNANDO, ORTEGA, NAVARRO, LÓPEZ: Problem Solving for Wireless Sensor Networks, London: Springer-Verlag, 2008, s. 225, ISBN 978-1-84800-202-9.

#### Abstract

Paper deals with area of wireless networks used in intelligent buildings. Mainly part is orientated to wireless technology ZigBee, which is described and is analysed its safety features and its possibility for using in intelligent house. Solution of wireless network was realised on the base of development kit CC2530ZDK made by firm Texas Instruments. Practical realisation is focusing on control of lighting, heating and external equipment. In the paper realised experiments in laboratory from the view of safety attacks to network are described and measures needs against particular attacks. Within realised cryptoanalytic's attacks program SmarRTF Studio was used.

#### Ing. Ján Ďurech

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 1  
010 26 Žilina  
[durech.jan@gmail.com](mailto:durech.jan@gmail.com)

#### Prof. Ing. Mária Franeková, PhD.

Žilinská univerzita v Žiline  
Elektrotechnická fakulta  
Katedra riadiacich a informačných systémov  
Univerzitná 1  
010 26 Žilina  
Tel.: 0421425133346  
[maria.franekova@fel.uniza.sk](mailto:maria.franekova@fel.uniza.sk)

- 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, diskrétné 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)  
Artificial 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)