

SLOVENSKÁ TECHNICKÁ UNIVERZITA v BRATISLAVE

Fakulta elektrotechniky a informatiky

Ústav riadenia a priemyselnej informatiky

Ing. Alexandra Šlezárová

Autoreferát dizertačnej práce

MODELOVANIE VO VIRTUÁLNEJ REALITE

RIADENIE SKUPINY AIRBOTOV

Na získanie vedecko-akademickej hodnosti philosophiae doctor

v študijnom odbore 5.2.14 Automatizácia a riadenie

Bratislava, 2012

Dizertačná práca bola vypracovaná v dennej forme doktorandského štúdia na Ústave riadenia a priemyselnej informatiky, Fakulty elektrotechniky a informatiky Slovenskej Technickej Univerzity v Bratislave.

Predkladateľ: Ing. Alexandra Šlezárová
Slovenská Technická Univerzita v Bratislave
Fakulta elektrotechniky a informatiky
Ústav riadenia a priemyselnej informatiky
Ilkovičova 3, 812 19 Bratislava

Školiteľ: doc. Ing. Igor Hantuch, CSc.
URPI, FEI STU v Bratislave

Oponenti: doc. RNDr. Andrej Ferko, CSc.
Katedra algebry, geometrie a didaktiky matematiky
Fakulta matematiky, fyziky a informatiky, Univerzita
Komenského
Mlynská dolina
842 48 Bratislava, Slovenská republika

prof. Ing. Aleš Janota, PhD.
Katedra riadiacich a informačných systémov
Elektrotechnická fakulta, Žilinská Univerzita v Žiline
Univerzitná 8215/1
010 26 Žilina, Slovenská republika

Autoreferát bol odoslaný:

Obhajoba dizertačnej práce sa koná.....26.10.2012.....o.....10.00.....h

na Fakulte elektrotechniky a informatiky
Slovenská Technická Univerzita v Bratislave
Ilkovičova 3,
812 19 Bratislava

.....
doc. RNDr. Gabriel Juhás, PhD.
Dekan Fakulty

OBSAH

ZOZNAM POUŽITÝCH SKRATIEK	6
ÚVOD	7
CIELE DIZERTAČNAJ PRÁCE	7
1. STAV VO SVETE	8
1.1 Kooperácia a koordinácia	8
1.2 Bezpilotné lietadlá	8
1.2.1 Trendy vývoja v doprave	8
1.2.2 Prehľad problematiky UAV vo svete	9
2. FORMY MODELOVANIA	10
2.1 Softvérovo dátový model	10
3. MODEL VO VIRTUÁLNEJ REALITE	11
3.1 Model „SVETA“	12
3.2 Návrh riadenia systému Ab	12
4. PROGRAMOVÉ PROSTREDIE MODELU	14
4.1 Návrh PPM	14
4.1.1 Vlastnosti PPM	14
4.1.2 Funkcie PPM	15
4.1.3 Funkcie Ab	15
4.2 Využitie AirboSim-u pri návrhu PPM	16
4.3 Realizácia PPM	20
5. RIADENIE EXPERIMENTU	21
5.1 Scenár misie	21
5.2 Návrh krivky pohybu Ab	21
5.3 Návrh pravidiel pre oblet prekážky po novej trase	23
5.3.1 pokutový okruh	23
5.3.2 Meranie spotreby množstvo paliva	24
5.3.3 Návrh pravidiel pre oblet prekážky po novej trase	25
5.4 Zdieľanie vedomia	26
5.4.1 Návrh oblasti práškovania	26
5.4.2 Obsah pamäte Ab	27
5.4.3 Ukladanie údajov do súboru	27
6. EXPERIMENTÁLNE OVERENIE	27
6.1 Podmienky experimentov	28
6.1.1 Scenár experimentov	28
6.1.2 Parametre experimentov	28
6.1.3 Úlohy Ab	29
6.1.3.1 Úlohy experimentu I. a II.	30
6.1.3.2 Rozdiely medzi experimentmi I. a II.	30
6.1.4 Meranie	30
6.1.5 Výstupy experimentov	31
6.1.6 Zobrazenie	34

6.1.6.1 Zobrazenie experimentu II.	34
7. VYHODNOTENIE EXPERIMENTOV	34
8. DISKUSIA K VÝSLEDKOM	35
9. SPLNENIE CIEĽOV DIZERTAČNEJ PRÁCE	35
10. ZÁVER	36
LITERATÚRA	36

ZOZNAM POUŽITÝCH SKRATIEK

AI	- Artificial Intelligence, umelá inteligencia
Airboty (Ab)	- Vzdušné roboty s vlastnosťami bezpilotných lietadiel
AirbotSim	- Vývojové softvérové prostredie pre simuláciu Ab
AGENT	- Predstavuje samostatnú entitu zabezpečujúcu určenú činnosť so špecializovaným cieľom
AVATAR	- Reprézntácia používateľa vo virtuálnej realite
CRS	- Centrálné riadiace stredisko
DDF	- Data Decentralized Fusion, Fúzia decentralizovaných dát
DEDS	- Discrete Event Dynamic System, Dynamický systém diskretných udalostí
EKF	- Extended Kalman Filter, Rozšírený Kalmanov filter
EÚ	- Európska Únia
GPS	- Global Position System, Globálna pozícia systémov
HAD	- Krivka podľa pohybu „hada“
HWIL	- Hardware In the Loop, Hardvér v slučke
IB	- Inteligentná budova
JAVA	- Programovací jazyk
KS	- Kolektívne správanie
MČ	- Množina čísiel
MILP	- Mixed-integer Linear Programming, Zmiešané celočíselné lineárne programovanie
MIT	- Massachusetts Institute of Technology, Massachusettský Inštitút Technológie
MSS	- Model stavového stroja
NASA	- National Aeronautics and Space Administration
NAV	- Nano Aerial Vehicle, Nano lietadlo
PDF	- Probability Density Function, Pravdepodobnostná funkcia hustoty
PPM	- Programové prostredie MODELU
PSR	- Programové syntézy riadenia
RE	- Riadenie experimentu
RS	- Riadiaci systém
SVM	- Support Vector Machine, Podpora vektorového zariadenia
SW	- Softvér
SWDM	- Softvérovo dátový model
TAR	- Teória automatického riadenia
TP	- Technologický proces
UAS	- Unmanned Aerial System, Bepilotný systém
UAV	- Unmanned Aerial Vehicle, Bepilotné lietadlá
ULL	- Ultralight Aircraft, Ultra ľahké lietadlo - vetroň
USA	- United States of America, Spojené Štáty Americké
VR	- Virtuálna realita
VŠDS	- Vysoká škola dopravy a spojov

ÚVOD

V súvislosti s lietaním do popredia vystupuje pojem autonómnosť. Dobrým príkladom pre tento výraz by mohli byť vesmírne sondy, ktoré majú najväčšiu mieru adaptívnosti.

Problematika UAV (Unmanned Aerial Vehicle) lietadiel je téma zložitá, ale veľmi aktuálna. Vedie od vývoja autopilotov, riadenia letovej prevádzky až po zariadenia umožňujúce úplnú autonómnosť lietania a plnenia misií.

Hlavným cieľom dizertačnej práce je vytvorenie univerzálneho programového systému, ktorý umožňuje modelovať, simulovať a riadiť skupinu robotov a preskúmať výhodnosť využitia tzv. „kolektívneho vedomia“.

CIELE DIZERTAČNEJ PRÁCE

Cieľom práce je navrhnúť riadenie kolektívneho správania (KS) sa Ab s využitím kolektívneho „vedomia“, ktoré budeme môcť pre porovnanie a vyhodnotenie experimentov prepínať na:

- vypnuté KS
- zapnuté KS

Pre kolektívne správanie, čo znamená napr. zdieľanie informácií, posielanie, zapisovanie a čítanie údajov bolo navrhnuté podľa [19] tzv. „kolektívne vedomie“, ktoré budeme skúmať v dvoch experimentoch I. a II.

Pre potreby simulácie experimentov si vytvoríme tzv. „mapu“, ktorá bude predstavovať terén, nad ktorým budú Ab lietať. Scenár experimentov je navrhnutý tak, aby sme preskúmali vlastnosti kolektívneho správania sa Ab v porovnaní s plnením misie bez neho.

V riadiacom systéme Ab bol pri návrhu využitý tzv. „STAVOVÝ STROJ“, ktorý nám umožňuje v sofistikovanejšej forme využívať pojem agent a agentové riadenie. Programová realizácia s využitím vlastností virtuálnej reality nám umožní simulovať dva druhy experimentov so zapnutým a vypnutým KS, čím preskúame a porovnáme vlastnosti KS.

Očakávame, že experiment II (zapnuté KS) prinesie lepšie a výhodnejšie riešenie ako experiment I. (vypnuté KS), pretože v prípade zapnutého kolektívneho vedomia majú Ab výhodu v tom, že si môžu zdieľať „vedomie“.

1. STAV VO SVETE

1.1 Kooperácia a koordinácia

Práce v oblasti koordinácie a kooperácie skupiny UAV je možné v literatúre sledovať už od roku 2002. V tejto dobe sa začali formulovať aj prvé tézy nedokončenej dizertačnej práce [13] a jej diplomové práce [14], [15], [16].

Vo väčšine publikácií [9], [10], [11] a [12] teória na pozadí riešení vychádza z klasickej TAR diferenciálnych rovníc riešených v transformáciách a maticovom zápise. Náš prístup vychádza z prác v oblasti 5. gen. softvéru a myšlienok akademika Cypkina [24]. Pre využitie princípov 5. gen. softvéru v oblasti modelovania a simulácie procesov reálneho času pomocou využitia tzv. softvérových dát modelu procesu, realizovaného v prostredí virtuálnej reality. Takto je možné „opísať“ chovanie resp. správanie sa procesov z oblasti DEDS bez znalosti matematického opisu (napr. počítačová resp. telekomunikačná sieť a pod.) s využitím tzv. scenára pre definovanie časových postupností, funkcií, udalostí, úkonov a pod. [18], [24].

1.2. Bezpilotné lietadlá

Tak, ako sa v oblasti robotov nachádza podskupina nanorobotov, tak sa aj bezpilotné lietadlá delia na NAV a UAV, kde sú:

- NAV – nano bezpilotné lietadlá
- UAV – bezpilotné lietadlá

Nano Air Vehicle (NAV) - nano vzdušné zariadenia sa vyvíjajú ako extrémne malé, ultral'ahké vzdušné systémy (o veľkosti menej ako 15 cm a vážiace menej ako 20 gramov) s potenciálom vykonávať vojenské misie vo vnútornom aj vonkajšom priestore [23].

UAV je zariadenie, ktoré dokáže lietať bez ľudskej posádky. Sú to zariadenia vhodné na určité letové operácie, kde nie je možná ľudská prítomnosť. Postupne začali tieto autopiloty prenikať z armádnych zdrojov do civilných aplikácií tak, aby spĺňali všetky potrebné úlohy.

1.2.1 Trendy vývoja v doprave

Podľa [20] ale trendy vývoja v oblasti dopravných systémov pre osobnú aj tovarovú prepravu pôjdu smerom k využívaniu ULL (Ultralight Aircrafts). Svedčia o tom aj súťaže NASA [7].

Dominantné postavenie v tejto priemyselnej oblasti má v súčasnosti Slovinsko, pričom [21] získal 3 ceny NASA za sebou [3] a [2].

Okrem toho významné postavenie, hlavne čo sa týka vývoja a exportu ULL do USA, má Česká republika [6]. Firma PhoenixAir so svojím ULL (motorizovaný vetroň) získala v NASA CAFÉ (2011) 3. miesto.

Taktiež nie je zanedbateľná aktivita Pipistrelu v oblasti výskumu a vývoja a ich spolupráce z významnými univerzitami v EÚ a vo svete. [5], [4], [21] a [22].

Tieto informácie určili aj rámec možnej aplikačnej oblasti tejto dizertačnej práce. Po analýze súčasného stavu palubnej navigačnej a pilotnej výzbroje civilných lietadiel – 4. generácia autopilotov spojených s navigačnými systémami pozemného zabezpečenia letovej prevádzky, kde sú už desaťročia spoľahlivo vyriešené problémy základného riadenia lietadla pomocou autopilota, sme sa zamerali na vyššiu úroveň riadenia a to riadenie skupiny lietadiel.

1.2.2 Prehľad problematiky UAV vo svete

Taktiež v [25] v kapitole Budúcnosť UAV (str. 44) je spomínaná nutnosť nasmerovať výskum a vývoj k vybaveniu UAV vyššou úrovňou AI (umelej inteligencie). Dominancia zaradenia UAV do „leteckých síl“ USA je už súčasnosťou. Otázky do budúcnosti sa týkajú už riešenia praktických riešení kolíznych situácií v rámci riadenia vzdušného priestoru civilného letectva.

Základná téza tejto dizertačnej práce - neriešiť, čo už je vyriešené - nám ohraničila pole nášho záujmu. Praktické otázky autopilota až po autonómne zvládnutie akrobacie je už vyriešené teoreticky aj prakticky. V [26] v kap. 8 sú už jasným spôsobom opísané konštrukcie, kde je využité matematické modelovanie nelineárneho riadenia UAV až do úrovne hybridných lietadiel s kolmým štartom, čo predstavuje v oblasti letectva ťažký oriešok.

V práci [27] na str. 101 je dokonca už opis lietajúceho zariadenia imitujúceho hmyz. Pozoruhodná je absencia fyzikálneho a matematického modelu resp. výpočet vztlakových síl, ktoré sú iba merané.

Nad prácou [28] môžeme iba nostalgicky povzdychnúť, nakoľko do nášho modelu sme prebrali iba výpočtovú reprezentáciu nášho „sveta“ vo virtuálnej realite a na vizualizáciu už nemáme ani technické prostriedky ani spolupracovníkov.

V práci [29] je znovu ukázaná plejáda už existujúcich modelov a riešení vychádzajúcich prevažne z klasickej TAR. Tieto metódy a ich aplikácie nás nasmerovali znovu mimo tento už vyriešený priestor.

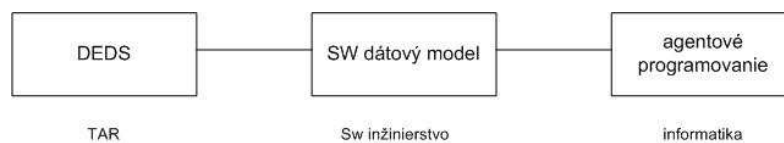
V [30] sa na str. 20 Christopher A. Bohn pokúša pomocou heuristiky overovať model návrhu riadenia UAV, čo už je bližšie k našej téme a aj v jeho práci je evidentná neexistencia vhodnej matematickej podpory riešenia tejto problematiky.

V [31] je už naznačené riešenie plánovania cesty pre skupinu UAV s využitím evolučných algoritmov. Pekne graficky prezentované výsledky ponúkajú vyriešený navigačný problém.

V [32] je precízny a podrobný opis UAV od základov aerodynamiky, cez atmosférické vplyvy a parametre na let až po ovládanie a navigáciu UAV. Aj táto kniha nás utvrdila vo vhodnosti výberu témy a zamerania tejto dizertačnej práce.

2. FORMY MODELOVANIA

Vzhľadom na to, že proces riadenia misie našich Airbotov je už ťažko opísateľný klasickou TAR – teória automatického riadenia, využijeme iné formy modelovania (Obr. 1) procesu a jeho riadenia. Nebudeme používať matematický model vo forme výpočtov zložitých diferenciálnych rovníc. Zameriame sa na problematiku DEDS - Discrete Event Dynamic System, dynamický systém diskretných udalostí.



Obr. 1 – formy modelovania

2.1 Softvérovo-dátový model

V tejto práci bol zvolený softvérovo dátový model procesu riadenia kolektívu Ab.

Pre vysvetlenie základných princípov využitia softvérovo - dátového modelu je potrebné zísť trochu do histórie. V r. 1973 bola doc. S. Petrášom, CSc. na UTK SAV Bratislava sformulovaná výskumná úloha Automatizovaná syntéza programových systémov riadenia (PSR) technologických procesov (TP) v reálnom čase. Už v r. 1977 počas študijného pobytu doc. Igora Hantucha na IPU ANZSSR Moskva akad. Cypkin načrtnol odvážne tézy :

„...keď je k dispozícii TP a generátorom vygenerujeme jeho PSR,

Čo keď neexistuje TP a my vygenerujeme PSR taký, pomocou ktorého tento neexistujúci TP môžeme modelovať ?...“

Ďalej pokračoval :

„...ak máme takýto softvérovo – dátový model TP,

Čo keď nepoznáme TP a pomocou vygenerovaného PSR TP môžeme modelovať ešte neexistujúce technológie ?“

A tak uzrelo svetlo sveta využitie generátora a jeho produktu PSR TP obrátene – nie na riadenie TP ale zrkadlovo otočené na modelovanie.

Práve tento princíp modelovania je opísaný v Kap. 6 a využitý v tejto dizertačnej práci.

Odtiaľto už bol len malý krôčik k aplikácií princípov agentového riadenia a následne agentového programovania, ktoré v podstate majú svoj pôvod v uvedenej histórii.

Na základe uvedených princípov SW dátového modelovania je v dizertačnej práci využitý **AirbotSim** [33], pomocou ktorého boli navrhnuté, zrealizované a odsimulované experimenty.

Pomocou AirbotSim-u boli :

- navrhnuté bloky, ktoré predstavujú fragment modelu (stavový stroj)
- pri návrhu boli definované parametre, ktoré sa využívajú počas vykonávania sa funkcií stavovým strojom a určujú správanie sa Ab
- následne sa program načíta do Ab
- program sa začne vykonávať po spustení (štartu) simulácie,
- počas simulácie program umožňuje napojiť sa na Ab a sledovať tak aktuálne parametre a stavy navrhnuté v stavovom stroji, čím sa testuje resp. overuje navrhnutý model
- výstupy z testovania sú v podstate priebehy stavov a hodnôt parametrov v reálnom čase
- na grafickú reprezentáciu priebehu experimentov bol využitý program pre vizualizáciu priebehu simulácie

3. MODEL VO VIRTUÁLNEJ REALITE

Ako vyplýva z Kap. 1, jediná možnosť ako experimentálne overiť vlastnosti KS sa skupiny Ab, bolo vytvorenie modelu vo VR. Jedná sa ale o komplexný a zložitý problém, preto bolo potrebné zdefinovať viacvrstvový a viacúrovňový model.

3.1 Model „SVETA“

Vo virtuálnej realite bol vytvorený model „SVETA“. „SVETA“ je viacvrstvový a viacúrovňový systém. Návrh dekompozície vychádzal zo základného scenára:

CRS (tvorca) → RS ↔ Ab ↔ Prostredie

kde:

- CRS je úroveň tvorca, odkiaľ sa riadi dianie v celom „Svete“
- RS je vrchná úroveň riadiaceho systému, ktorá z hornej úrovne CRS prijíma a na spodnú úroveň posiela pokyny
- Ab je samotný Airbot, ktorý realizuje prijaté príkazy z RS (vo forme funkcií a parametrov) nad Prostredím
- Prostredie je parametricky (podľa informácií z CRS) definované

Takto navrhnutý Model „SVETA“ slúži pre návrh a tvorbu softvérového prostredia tak, aby v prípade existencie hardvérových vrstiev bolo možné „prepínať“ celý systém z virtuálnej reality do reálneho prostredia a späť.

3.2 Návrh riadenia systému Ab

Kľúčovou úlohou práce je preskúmať možnosti riadenia kolektívneho správania sa skupiny Ab. Pre tieto účely boli navrhnuté a testované dva experimenty správania sa Ab:

- bez zapnutého kolektívneho správania sa (KS)
- so zapnutým kolektívnym správaním (KS)

Základný princíp KS spočíva v definovaní **kolektívneho vedomia**. Pod týmto pojmom budeme rozumieť zdieľanie informácií z **mapy**, vďaka ktorým budú Ab vedieť reagovať na zmeny vykonané v misii. V našom modeli sa jedná o informácie o prekážkach, na ktoré Ab bude reagovať, pri plnení misie rozdielne podľa toho, čo o prekážke „vie“ alebo „nevie“. Mapa ďalej slúži Ab na orientáciu sa v priestore alebo nad terénom. Zároveň Ab z nej získavajú a zaznamenávajú informácie počas letu. Základne vlastnosti mapy, vlastnosti Ab sú zadávané centrálnym riadiacim strediskom (**CRS** - úroveň TVORCU – AUTORA).

Misia a úlohy sú naprogramované a modifikujú sa zmenou programu v riadiacej vrstve systému (RS). RS je miesto, z ktorého sa riadi misia pre Ab. Misia pozostáva z parciálnych úloh – cieľov, do ktorých sa majú Ab dostať. Pre potreby realizácie riadiaceho systému a jeho overenia pomocou experimentov je navrhnuté SW prostredie, v ktorom boli vytvorené nasledovné vrstvy:

- **riadenie experimentov (RE)**: najvyššia vrstva, z ktorej sa riadi beh celej simulácie, zadávajú sa v nej globálne parametre určujúce beh

jednotlivých úrovní, zadávajú a modifikujú sa príkazy modelu stavového stroja

- **model sveta:** softvérová vrstva, ktorá reprezentuje priestor - terén, v ktorom sa vykonáva misia. Je v ňom realizovaná MAPA, ktorú vidia Ab – komunikujú s ňou (čítajú z nej údaje a zapisujú do nej v prípade zapnutia KS)
- **úroveň Ab:** predstavuje softvérová vrstva, v ktorej sú realizované Ab (pohyb, komunikácia, vykonávanie funkcií)
- **riadiaci systém (RS):** ktorý komunikuje s CRS, od ktorého prijíma úlohu z misie
- **mapa** je softvérový priestor, v ktorom je realizovaný „terén“, nad ktorým budú Ab „lietať“ – pohybovať sa, a do ktorej sa budú z RE definovať „prekážky“ a obmedzenia

Schéma základných funkcií je nasledovná:

- **pohyb Ab** predstavuje priemet ich simulovanej polohy do MAPY. Ab má v simulácii súradnice $[x, y, a]$, kde $[x, y]$ sú súradnice z MAPY, a je výška (altitude)
- **zdieľanie mapy** Ab znamená : pre každé pole int (x, y, p) , kde x, y sú súradnice, p – parameter (číslo – napr.: 3 – PREKÁŽKA (farba červená)), význam farieb je určený v RE
- **KS** znamená, že Ab sú schopné počas letu zapisovať do mapy nové informácie a vzájomne ich zdieľať
- v prípade **vypnutého KS** Ab z mapy len čítajú (ciele a prekážky definované z RE)
- v prípade **zapnutého KS** Ab z mapy čítajú a zapisujú do nej (ciele a prekážky definované z RE ako v prípade vypnutého KS a navyše zapisujú do mapy nové údaje, ktoré Ab získali počas letu)
- **kolektívne vedomie** je vlastnosť, ako sa kolektívne správanie Ab realizuje pomocou zdieľania mapy
- **misia** je hlavným cieľom činnosti množiny Ab definovaná z úrovne RE
- **úloha** je časťou misie pre konkrétny Ab
- **funkcie** Ab sú množina činností, ktoré musia Ab vykonať pre splnenie úlohy

4. PROGRAMOVÉ PROSTREDIE MODELU

Táto kapitola obsahuje návrh a programovú realizáciu programového prostredia MODELU (PPM).

4.1 Návrh PPM

Návrh PPM vychádza z princípov softvérovo dátového modelu (SWDM). Na základe filozofie v Kap. 5 a dekompozície bol navrhnutý PPM nasledovne:

- vlastnosti PPM
- funkcie:
 - PPM
 - Ab

4.1.1 Vlastnosti PPM

Po analýze a dekompozícii modelu nám vyšli nasledovné vlastnosti:

- začiatok
- simulácia
- koniec

Začiatok Misie

- definovanie a modifikácia správania sa Ab pomocou blokov a to ich:
 - vytvárať
 - uložiť
 - načítať
- definovanie Mapy, kde v súradniciach a sektoroch sa zadávajú:
 - prekážky
 - oblasti
- načítanie správania sa Ab pre Misiu
- načítanie správání pre Misiu (možnosť modifikovať prostredie)

Simulácia Misie

- načítanie Ab do Mapy
- odobratie Ab z Mapy
- spustenie simulácie misie
- zastavenie simulácie misie

Koniec Misie

- modifikácia rýchlosti behu simulácie (časový zoom)
- sledovanie aktuálnych stavov a parametrov Ab (debugovanie)
- vizualizácia počas simulácie

4.1.2 Funkcie PPM

Po analýze a dekompozícii modelu nám vyšli nasledovné funkcie:

- návrh oblastí
- priradovanie farieb
- vizualizácia mapy počas misie

Návrh Oblastí (Voľba oblastí):

- práškovacia oblasť,
- oblasť pristátia
- čerpacia stanica
- prekážky
- vyznačenie okolia prekážky
- pokutový okruh

Priradovanie Farieb:

- práškovacie pole – zelená
- čerpacia stanica – ružová
- oblasť pristátia – tmavomodrá
- práškovanie – oranžová
- prekážky – žltá
- vyznačenie okolia prekážky – slabomodrá
- z pohľadu Ab
 - vidieť body pokutového okruhu
 - vidieť označenie okolia prekážky (zapnuté KS)
 - odkrývanie trajektórie

Vizualizácia mapy počas misie:

- možnosť vidieť oblasti na mape
 - práškovacie pole
 - čerpaciu stanicu
 - oblasť pristátia
 - vyznačenie okolia prekážok
 - z pohľadu Ab - body pokutového okruhu
 - súradnice a sektory (pravý horný roh)

4.1.3 Funkcie Ab

Pre jednotlivé Ab máme nasledovné funkcie:

- zaznamenávanie dát z experimentov I. a II.
- pohyb na mape
 - pohyb v tvare krivky „HAD“
 - smer A

- smer B
- kolektívne vedomie - zdieľanie globálnych dát
 - stretnutie Ab s prekážkou
 - vypnuté KS
 - zapnuté KS
 - žiadny posun informácie pri vypnutom KS
 - posun informácie pri zapnutom KS
 - označenie prekážky
 - zaznačenie informácie do mapy
 - komunikácia – výmena údajov
 - čítanie z mapy
- plnenie úloh misie pri **vypnutom KS** (1, 2, 3, 4, 7)
 1. úloha – nájdenie počiatočných bodov
 2. úloha – po vstupe do práškovej oblasti začať práškovať
 - **oranžová** farba – práškovanie
 3. úloha – využitie pokutového okruhu
 - **svetlomodrá** farba - Body pokutového okruhu
 4. úloha – dotankovať palivo
 7. úloha – po ukončení práškovania pristáť
 - zníženie rýchlosti
 - zníženie výšky
- plnenie úloh misie pri **zapnutom KS** (1, 2, 3, 4, 5, 6, 7)
 1. úloha – nájdenie počiatočných bodov
 2. úloha – po vstupe do práškovej oblasti začať práškovať
 - **oranžová** farba – práškovanie
 3. úloha – využitie pokutového okruhu
 - **svetlomodrá** farba - Body pokutového okruhu
 4. úloha – dotankovať palivo
 5. úloha – vyznačenie okolia prekážky
 - **svetlomodrá** farba – vyznačenie okolia prekážky
 6. úloha
 - výmena obsahu svojich pamätí (napr. trajektória)
 7. úloha – po ukončení práškovania pristáť
 - zníženie rýchlosti
 - zníženie výšky

4.2 Využitie AirboSim-u pri tvorbe PPM

AirbotSim bol navrhnutý tak, aby v ňom bolo možné postupné vykonávanie rutín programu. Rutina v tomto prípade je časť programu, ktorej aktivitu je možné vyvolať iným programom alebo samé sebou.

Princíp SWDM pozostáva z vytvorenia modelu SW a modelu dát. Model je sám o sebe nefunkčný. Pre jeho oživenie je potrebné z modelu vytvoriť program. O túto činnosť sa stará generátor, ktorý zoberie požiadavky na PPM a na základe zadefinovania parametrov PPM vytvorí funkčný PPM. Pod pojmom funkčný PPM berieme do úvahy celkovo napísaný a preložený program, vykonávajúci všetky požadované funkcie PPM, do ktorého už počas samotnej simulácie nebude možné zasiahnuť.

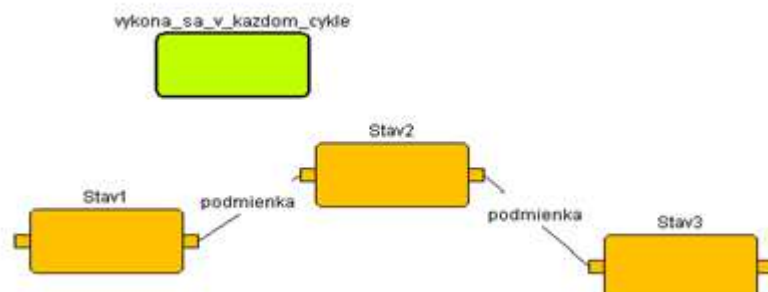
AirbotSim je v podstate vývojové prostredie pre tvorbu - programovanie PPM na základe princípov SWDM a bolo vytvorené v projekte [35], ktorý bol koordinovaný paralelne s touto prácou.

Z princípov SWDM vyplýva, že je v ňom možné vytvoriť PPM, ktorý je nezávislý od programovacieho jazyka a platformy. V projekte **AirbotSim** bola abstrakcia trochu znížená a to tak, že za programovací jazyk bol zvolený jazyk JAVA, čím sa zjednodušila tvorba PPM.

Princípy SWDM použité v projekte **AirbotSim** uvedieme na nasledujúcom príklade, kde je ukázané, ako sa využívajú moduly z **AirbotSim**-u pri tvorbe PPM.

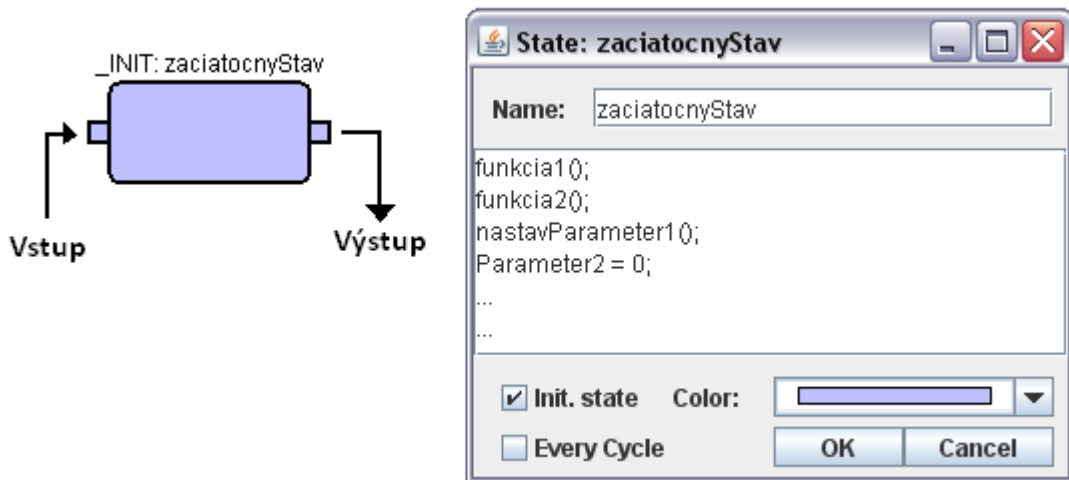
Tvorba PPM prebieha na grafickej úrovni (Obr. 2) a je reprezentovaný navzájom prepojenými blokmi. Pod týmito blokmi sa v **AirbotSim**-e nachádza program v jazyku Java a parametre bloku (Obr. 3). Konkrétne sa jedná o parametre „*name*” meno bloku, časť programu patriaca pod tento blok, parameter „*Init. state*” reprezentujúci či sa jedná o blok, v ktorom sa začne vykonávanie programu a parameter „*every cycle*” označujúci tento blok za blok, ktorý sa vykoná v každom programovom cykle nezávisle od stavu programu.

Po označení bloku parametrom „*every cycle*” nie je možné tento blok prepojiť zo žiadnym iným blokom (Obr. 2 - blok v zelenej farbe).



Obr. 2 – Grafická prezentácia softvérového modelu

V našom prípade **reprezentujú bloky** programu (Obr. 3) jednotlivé **stavy** stavového stroja, preto ich v práci označujeme menom „**stav**“.



Obr. 3 – Grafické Zobrazenie parametrov, ktoré je možné pridať jednotlivým blokom programu

Tvorba dátového modelu v **AirbotSim**-e je zjednodušená pomocou zapisovania v syntaxe jazyka JAVA (Obr. 4). Vkladaním riadkov pomocou syntax jazyka JAVA sa nám vytvára model dát. Tieto dáta sú prístupné vo všetkých blokoch softwarového modelu v rámci **AirbotSim**-u.

```
parameter1;
parameter2;
parameter3;
...
...
...
```

Obr. 4 – Grafické zobrazenie modelu dát

Proces vytvorenia programu PPM pomocou **AirbotSim**-u sa nazýva generovanie. Generátor navrhnutý v projekte **AirbotSim** vytvára program v jazyku JAVA. Počas generovania sa pospájajú jednotlivé bloky navrhnuté v PPM do logických celkov. Z dátového modelu zadaného v PPM sa stanú atribúty triedy. Ďalej sa realizuje inicializácia programu a pripoja sa testovacie rutiny.

Samotné správanie sa PPM závisí vlastne od vhodne zvolených a zadaných parametrov počas návrhu PPM.

Generátor **AirbotSim**-u v našom prípade spracováva:

- inicializáciu premenných
- vytvorenie premenných (atribúty triedy) z modelu dát
- prepojenie jednotlivých SW blokov podľa SW modelu
- na základe parametrov modelu rozhodne o implementácii jednotlivých SW blokov
- vytvorenie testovacích rutín pre kontrolu behu programu - správania
- vytvorenie rutín pre zaznamenávanie behu programu
- pripojenie knižnice Airbota s obsahom funkcií

Architektúra vygenerovaného PPM bola inšpirovaná riadiacimi systémami PLC. Riadiace systémy PLC [34], [35] pracujú vo veľmi rýchlych cykloch (doba jedného cyklu cca **20 ms**), ktoré stíha aj náš PPM:

- testovanie
- načítanie vstupov
- spracovanie programu
- zapísanie výstupov

Architektúra vygenerovaných programov správania sa PPM je založená na pravidelných cykloch, teda ako v PLC. Jeden cyklus pozostáva z nasledujúcich častí:

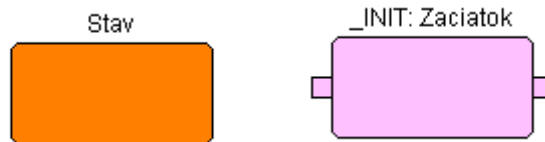
- vykonanie blokov označených parametrom „*every cycle*“ (určenie poradia vykonávania týchto blokov sa určí pri generovaní, ale tvorca modelu k tomuto prístupu nemá)
- vykonanie jedného bloku na základe stavu programu
- spustenie testovacej rutiny, na základe, ktorých môže tvorca sledovať beh programu

Týmto spôsobom (vykonávanie blokov postupne nie súčasne) je vygenerovaný program v našom prípade PPM synchronizovaný lepšie, ako keby sa generovali jednotlivé bloky pomocou **AirbotSim**-u ako viac - vláknový program. Ak by sme totiž generovali program ako viac - vláknový museli by sme sa starať aj o synchronizáciu prístupu k údajom.

Najväčšia výhoda používania princípov SWDM je v našom prípade jednoduchosť vygenerovania akéhokoľvek programu s ľubovoľnou architektúrou a to pre ľubovoľnú SW platformu. Toto je dosiahnuté jednoduchým modifikovaním samotného generátora. Z toho vyplýva, že pri migrácii SW z PPM do reálneho modelu lietadla už nie je potrebná softvérová úprava PPM. V **AirbotSim**-e sa modifikácia generátora vykoná zmenou jednej triedy napísanej v jazyku JAVA.

4.3 Realizácia PPM

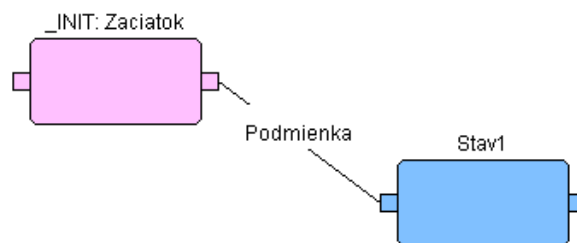
Podľa Kap. 4.1.1. bol navrhnutý PPM s nasledovnými vlastnosťami a funkciami. Pri návrhu vlastností PPM a Ab môžeme použiť dva typy blokov (Obr. 5):



Obr. 5 – Grafické zobrazenie blokov pre vytvorenie vlastností PPM a Ab

Oranžový blok s názvom *stav* sa vykonáva stále dookola, nie je závislý od stavu programu. Ružový blok je závislý od stavu programu a vykoná sa ako prvý, pretože je označený prefixom *_INIT:*. Prefix bloku *_INIT:* sa priradí k bloku parametrom *initRun*. Blok označený prefixom *_INIT:* sa pri návrhu môže použiť iba jeden.

Ružový blok (Obr. 6) s názvom *_INIT:Zaciatok* sa taktiež vykonáva stále dookola, ale je závislý od stavu programu. To znamená, že ak po splnení prechodovej podmienky (*Podmienka*) prejde do nasledujúceho stavu *stav1*, v ďalšom cykle programu sa nevykoná blok *_INIT:Zaciatok*, pretože sa vykonáva blok, do ktorého prešiel (*stav1*).



Obr. 6 – Grafické zobrazenie blokov

Pri návrhu správania sa jedného Ab využívame funkcie uložené v knižnici *Airbot.lib*, ktoré vkladáme do blokov:

- komunikačné – *vymenUdaje()*, *zmazUdajeOMape()*, *skenuj()*
- navigačné – *letDoBodu()*, *letDoSektora()*
- senzorické – *Sektor1-9()*
- operačné – *nastavStav()*

Postupným vkladáním funkcií, parametrov, príkazov a premenných do blokov a pospájaním blokov, závislých od stavu programu dostaneme celkový program, ktorý charakterizuje alebo ktorým ovládame správanie sa PPM a Ab.

5. RIADENIE EXPERIMENTU

Pre riadenie experimentu sme zvolili nasledovný opis :

- scenár misie - úroveň CRS
- funkcie misie - úroveň MSS
- nastavenie parametrov misie - úroveň Ab

5.1 Scenár misie

Na misiu vyšleme štyroch Ab. Ich úlohou je práškovať vyhradenú oblasť, definovanú súradnicami mapy. Pohybovať sa budú podľa krivky „HAD“, vid' Obr. 7 a 8. Úlohou je preskúmať, ako sa bude plniť misia v prípadoch:

- **s vypnutým KS**
- **so zapnutým KS**

V prípade výskytu **prekážky** Ab zareagujú podľa toho, či je prekážka:

- známa – existuje na MAPE
- neznáma – neexistuje na MAPE

V prípade „**známa PREKÁŽKA**“ Ab:

- zvolí body, ktorými prekážku obíde
- bude letieť po novej upravenej trase

V prípade „**neznáma PREKÁŽKA**“ Ab

- pre identifikáciu prekážky využije tzv. **pokutový okruh**, a potom (v prípade **vypnutého KS**) túto novú prekážku obletí ako známu a do globálnej mapy ju **nezapíše**
- v prípade **zapnutého KS** Ab postupuje ako v hore uvedenom prípade až na to, že informáciu o prekážke **zapíše** do globálnej mapy, čím umožní **zdieľanie** tejto informácie všetkým Ab, ktoré KS majú zapnuté (takto je realizované kolektívne „vedomie“)
- pokutovým okruhom sa predĺži letová trasa Ab, pri čom sa mu zvýši aj celkové množstvo spotrebovaného paliva

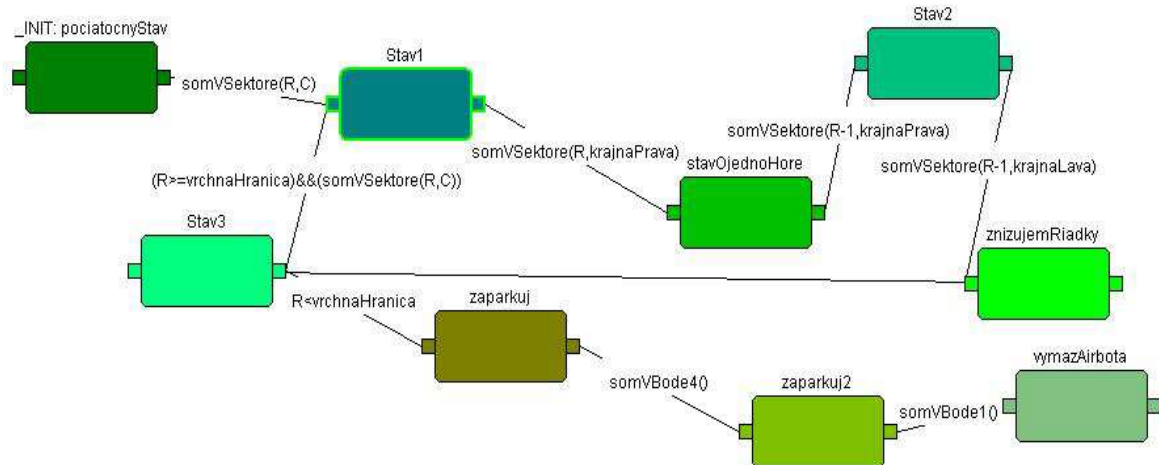
5.2 Návrh krivky trajektórie pohybu Ab „HAD“

Krivka trajektórie letu Ab „HAD“ je opísaná nasledujúcimi stavmi a podmienkami v Tab. 1.

Tab. 1 – zobrazuje potrebné stavy a podmienky pre návrh trajektórie letu Ab „HAD“

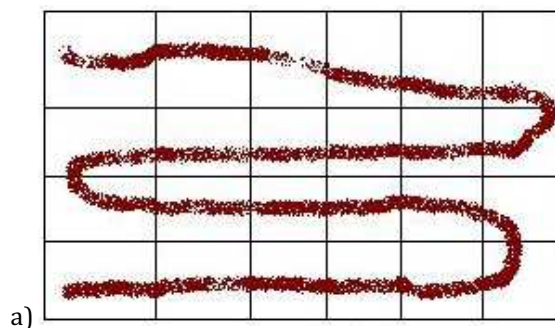
STAVY	PODMIENKY
<i>pociatocnyStav</i>	$somVSektore(R, C)$
<i>Stav1</i>	$somVSektore(R, krajnaPrava)$
<i>Stav0JednoHore</i>	$somVSektore(R-1, krajnaPrava)$
<i>Stav2</i>	$somVSektore(R-1, krajnaLava)$
<i>znizujemRiadky</i>	$(R \geq vrchnaHranica) \&\& (somVSektore(R, C))$
<i>Stav3</i>	$R < vrchnaHranica$
<i>Zaparkuj</i>	$somVBode4()$
<i>Zaparkuj2</i>	$somVBode1()$
<i>vymazAirbota</i>	

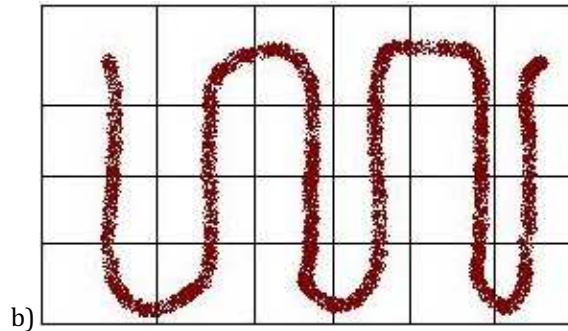
Na Obr. 7 sú uvedené stavy a podmienky pre návrh krivky trajektórie letu pre Ab „HAD“ znázornené graficky v modeli stavového stroja.



Obr. 7 – Grafické zobrazenie stavov a podmienok krivky trajektórie Ab „HAD“ v modeli stavového stroja

Ďalší Obr. 8 a) a b) zobrazuje grafické prevedenie pohybu Ab v smere určenom v misii A a B.





Obr. 8 a) Grafické zobrazenie pohybu Ab v smere A,
b) Grafické zobrazenie pohybu Ab v smere B

5.3 Návrh pravidiel na identifikáciu prekážky

Obvyklý letový manéver pri zistení prekážky je prieskumný okruh počas ktorého pilot identifikuje prekážku a nájde spôsob ako ju obísť. V prípade Ab pre túto činnosť definujeme **pokutový okruh**. V nasledujúcej časti definujeme pokutový okruh pre obídenie prekážky.

5.3.1 Pokutový okruh

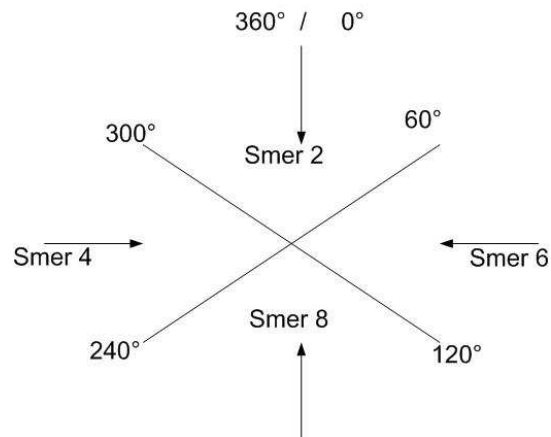
Pravidlá pre pokutový okruh sú uvedené v Tab. 11, Obr. 34 graficky zobrazuje body, ktoré definujú pokutový okruh. Ab cez ne obletí prekážku, kde modrou farbou sú vyznačené body obchádzania prekážky¹.

Rozsah stupňov rozdelených podľa Tab. 3 predstavuje kurz, z ktorého môže Ab priletieť (Obr. 9).

Tab. 2 - rozdelenie sektorov a rozsah stupňov

Rozdelenie sektorov	Rozsah stupňov
<i>obsahSektora2()</i> - smer 2	300° - 0°, 0° - 60°
<i>obsahSektora4()</i> - smer 4	300° - 240°
<i>obsahSektora6()</i> - smer 6	60° - 120°
<i>obsahSektora8()</i> - smer 8	120° - 240°

¹ Ružová bodka znázorňuje Airbota, čierna čiara vyznačuje trajektóriu obchádzania



Obr. 9 - Grafické zobrazenie rozdelenia kurzov a smerov

r (riadok) = súradnica $y()$, c (stĺpec) = súradnica $x()$

Tab. 3 – zobrazuje body pokutového okruhu

Súradnice bodov a rozsah stupňov tvoriace body a smery pre Ab na identifikáciu prekážky			
60° - 120°	240° - 300°	120° - 240°	0° - 60° a 300° - 360°
sprava - doľava	zl'ava - doprava	zdola - hore	zhora - dole
$y-2, x$	$y+2, x$	$y, x-2$	$y, x+2$
$y-2, x-2$	$y+2, x+2$	$y+2, x+2$	$y-2, x+2$
$y, x-2$	$y, x+2$	$y+2, x$	$y-2, x$
$y-1, x+1$	$y+1, x-1$	$y-1, x+1$	$y-1, x-1$
$y, x+2$	$y, x-2$	$y, x-2$	$y, x-2$

5.3.2 Meranie spotrebovaného množstva paliva

Jednou z funkcií Ab je aj sledovanie spotreby paliva. Ak množstvo paliva v nádrži klesne pod parametrom určenú minimálnu hladinu začne sa vykonávať funkcia `<DoplPalivo>`.

Ab zmení stav a tým aj letový režim a vybočí zo stavu trasy Ab „HAD“ (zo smeru A alebo B) a prejde na trasu smerom na čerpaciu stanicu, ktorá je znázornená na mape purpurovým štvorcem.

Pri obchádzaní prekážky pokutovým okruhom sa zvýši spotrebované množstvo paliva, podľa parametra `<spotrebovanePalivo2>`. Vid' Tab. 4.

Tab. 4 – stavy a podmienky prechodu používané pri kontrole paliva

STAVY	PODMIENKY PRECHODU
<code>letiPrePalivo</code>	<code>somVSektore2()</code>
<code>DoplPalivo</code>	

5.3.3 Návrh pravidiel pre oblet prekážky po novej trase

Operácia *<oblet>* (obletenie prekážky) sa vykonáva pri zapnutom KS, kedy Ab zapíše do mapy novú trasu. Pre vykonanie tejto operácie boli použité premenné a funkcie uvedené v Tab. 5, 6.

Tab. 5 - premenné používané vo fragmente kódu, obletenie prekážky.

PREMENNÉ
<i>obchadzanie</i>
<i>oblet</i>
<i>poziciaVPoliObletu</i>
<i>airbot.heading</i>
<i>poziciaPrekazky</i>
<i>poleObletu</i>
<i>somVSektore().y-1, somVSektore().x+1</i>
<i>poslednyStav</i>
<i>airbot.state</i>

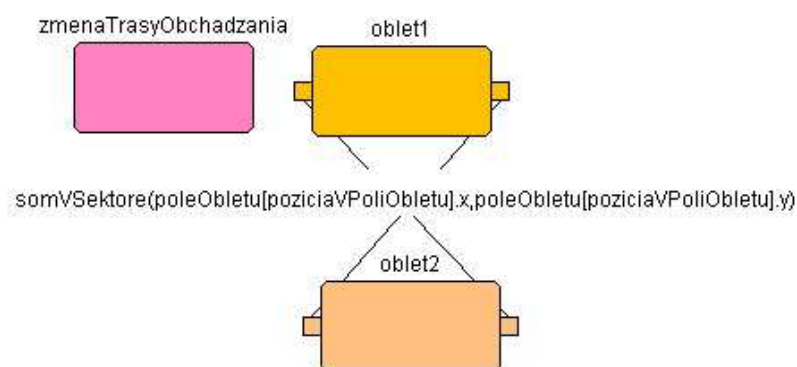
Tab. 6 – funkcie používané v bloku *<zmenaTrasyObchadzania>*

FUNKCIE
<i>obsahSektora6()</i>
<i>obsahSektora3()</i>
<i>obsahSektora9()</i>
<i>nastavStav()</i>

Operácia *<oblet>* sa vykonáva stavmi a podmienkami uvedených v Tab. 7 navrhnuté v modeli stavového stroja (Obr. 10).

Tab. 7 – stavy a podmienky, ktoré vykonávajú obletenie prekážky

STAVY	PODMIENKY PRECHODU
<i>oblet1</i>	<i>somVSektore(poleObletu[poziciaVPoliObletu].x, poleObletu[poziciaVPoliObletu].y)</i>
<i>oblet2</i>	<i>poziciaVPoliObletu < poleObletu.length</i>



Obr. 10 – Grafické zobrazenie blokov v modeli stavového stroja

5.4 Zdieľanie „vedomia“

Dôležitá funkcia pri **zapnutom KS** je prepojenie „**vedomia**“ globálnej mapy tak, aby z nej mohol Ab zdieľať informácie.

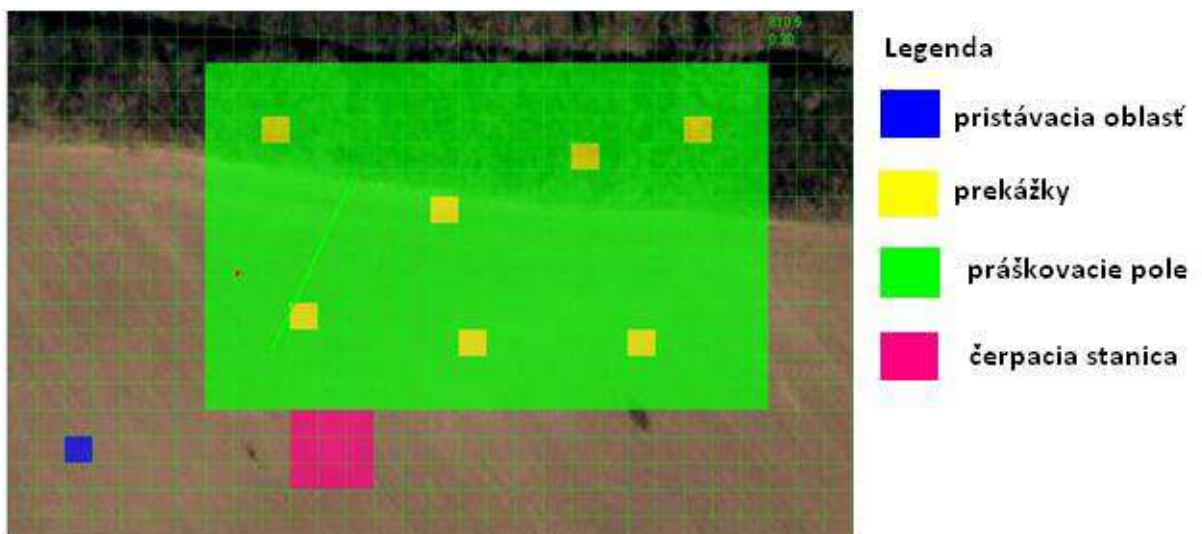
Premenná

`<consciousness.mapGPS.grid.area>`

umožňuje Ab **zdieľať** informácie z globálnej mapy, ktorá je aktualizovaná pri zapnutom KS každým Ab, ktorý nájde prekážku.

V globálnej mape sú zadefinované oblasti z úrovne CSR, kde každá oblasť má svoju funkciu a je označená inou farbou, viď Obr. 11:

- **práškovacie pole – zelená farba**
- **prekážky – žltá farba**
- **čerpacia stanica – purpurová farba**
- **pristávanie – tmavomodrá farba**



Obr. 11 – Grafické znázornenie „terénu“ – mapy, jeho oblasti a prekážky, v ktorom sa Ab pohybuje

5.4.1 Návrh oblasti práškovania

Oblasť práškovania je zadefinovaná v úrovni CSR a je uložená vo „vedomí“ globálnej mapy, kde Ab pri zapnutom KS vyznačí bodmi novú trasu (okolie prekážky) a uloží ju do nej.

5.4.2 Obsah pamäte Ab

Táto kapitola popisuje, obsah pamäte Ab, ktorý predstavuje jeho „videnie“. Ab má okolo seba kvadranty, kde každý kvadrant má priradené svoje číslo. Tieto čísla predstavujú **smery sektorov**, cez ktoré sa Ab pohybuje:

- 1-4
- 6-9

5.4.3 Ukladanie údajov do súboru

Po skončení misie sa do súboru **zapíšu údaje**, ktoré Ab nadobudol počas misie:

- spotrebované palivo počas misie
- súradnica x
- súradnica y
- počet prejdenných prekážok
- výška Ab
- rýchlosť Ab

6. EXPERIMENTÁLNE OVERENIE

Cieľom kapitoly je ukázať na dvoch príkladoch navrhnuté správanie Ab, ako sa „správajú“ – plnia misiu.

Priebeh experimentov v reálnom čase je v plnej verzii v prílohe I (**video_Experimentov.avi**). Nakoľko sa jedná o dynamický priebeh experimentov v 3D prostredí je problém s jeho prezentáciou na 2D papieri tejto práce.

Preto má táto kapitola charakter „reportáže“ o experimentoch, kde jednotlivé obrázky sú len dôležité fragmenty celkového priebehu experimentov.

Táto kapitola je v podstate úvodom k úplnej dynamickej prezentácii priebehu výsledkov v reálnom čase počas obhajoby a prezentácie tejto dizertačnej práce.

Scenár „reportáže“

Na preukázanie ako budú plniť misiu 4 Ab s (**vypnutým/zapnutým**) **KS** a v rôznych smeroch misie (A, B) bol navrhnutý:

- experiment I. - **vypnuté KS** (Ab **nezapisuje** do MAPY zmeny, ktoré sa udejú počas vykonávania MISIE)

- experiment II. - **zapnuté KS** (Ab **zapisuje** do MAPY zmeny, ktoré sa udejú počas vykonávania MISIE)

6.1 Podmienky experimentov

Spoločné podmienky experimentov I. a II. boli nastavované nasledovne:

- prostredie
- parametre
- úlohy
- meranie
- výstupy
- zobrazenie

6.1.1 Scenár experimentov

Celý experiment prebieha podľa modelu vo virtuálnej realite z Kap 3. Prostredie je vrstva, ktorá nám predstavuje ako keby „terén“, v ktorom sa uskutočnia obidva experimenty.

Scenár experimentov má dve úlohy ako realizovať misiu:

- s vypnutým KS
- so zapnutým KS

Prvou úlohou je, že Ab odštartujú z počiatočného bodu na mape, ktorý reprezentuje základné letisko. Druhou úlohou je prelet z okruhu letiska na začiatok oblasti plnenia hlavnej časti misie – v našom prípade bola ako typická úloha vybraná „PRÁŠKOVANIE“.

6.1.2 Parametre experimentov

Spoločné parametre používané pre experiment I. a II. vychádzajú z dekompozície, kde uvažujeme o viacvrstvovej parametrizácii [2]. Nastavujeme ich tak, aby definovali celý požadovaný priebeh experimentov od vlastností prostredia (kde sa Ab pohybujú), cez správanie sa Ab (ako sa pohybujú) až po riadenie celého experimentu.

V experimente I. a II. sú použité **statické** (Tab. 8) a **dynamicky** sa meniace parametre (Tab. 9), zadané úrovňou CRS. Týmito parametrami ovládame správanie Ab v misii. Tab. 8, 9 obsahuje parametre, ktoré využíva Ab pri vykonávaní misie. Parametre sú modifikovateľné z úrovne CRS.

Tab. 8 – statické parametre

STATICKÉ PARAMETRE	
<code>int krajnaPrava = 26</code>	hranice práškovacej oblasti
<code>int krajnaLava = 7</code>	
<code>int vrchnaHranica = 2</code>	
<code>int spodnaHranica = 14</code>	

Tab. 9 – dynamicky sa meniace parametre Ab

DYNAMICKÉ PARAMETRE	
<code>int R = 2</code>	čísla predstavujú počiatočné body Ab, od ktorých začal práškovať
<code>int C = 7</code>	
<code>double palivo = 1000</code>	hodnota paliva na začiatku
<code>double rychlost = 0.2</code>	rýchlosť Ab
<code>int cyklus = 0</code>	premenná pre počet cyklov
<code>double vyska = 5.0, 3.0, 7.0, 9.0</code>	letová hladina

6.1.3 Úlohy Ab

V rámci experimentov sa snažíme modelovať a simulovať nasledovné úlohy. Úlohou Ab po vpustení do misie je nájsť začiatočné body práškovacej oblasti. Po vstupe do práškovacej oblasti majú Ab za úlohu popráškovat' územie a popri tom identifikovať prekážky, obletieť ich podľa stanoveného algoritmu (napr. pokutovým okruhom). Cieľom misie je sledovanie spotreby paliva a trajektórie Ab popráškovania územia tak, aby sa vyhýbali prekážkam.

Celková úloha bola dekomponovaná na nasledovné časti:

Úloha 1:

- prvou úlohou Ab je vzlietnuť do požadovanej letovej hladiny a nájsť začiatočné body, od ktorých Ab začne práškovať.

Úloha 2:

- druhou úlohou Ab je popráškovat' vyhradenú oblasť.

Úloha 3:

- treťou úlohou Ab je po identifikácii prekážky urobiť predpísaný manéver – v našom prípade pokutový okruh.

Úloha 4:

- štvrtá úloha Ab je ísť dotankovať palivo, ak dôjde k splneniu podmienky poklesu paliva pod stanovenú minimálnu hodnotu.

Úloha 5:

- piatou úlohou Ab (pri zapnutom KS) je označiť prekážku a jej okolie a zapísať tento údaj do mapy.

Úloha 6:

- šiestou úlohou Ab (pri zapnutom KS) je zabezpečiť výmenu obsahu svojich pamätí. Táto úloha slúži na výmenu informácií (napr. pomocou vzájomnej komunikácie o významných informáciách počas letu) mimo mapy

Úloha 7:

- siedmou úlohou Ab je po dokončení misie vrátiť sa na bázové letisko a pristáť.

6.1.3.1 Úlohy experimentu I a II.

Experiment I., ktorý je definovaný na splnenie misie s **vypnutým KS** vykonáva nasledovné úlohy (1, 2, 3, 4, 7):

Experiment II., ktorý je definovaný pre splnenie misie so **zapnutým KS** vykonáva nasledovné úlohy (1,2,3,4,5,6,7):

6.1.3.2 Rozdiely medzi experimentom I. a II.

Pri **vypnutom** KS (experiment I.) Ab:

- nemá k dispozícii informácie od ostatných Ab
- „nevie“ o prekážkach
- pri každej prekážke musí urobiť obchádzku – pokutový okruh

Pri **zapnutom** KS (experiment II.) Ab :

- má k dispozícii informácie od ostatných Ab
- „vie“ o prekážkach
- pri vyznačenom okolí prekážky neurobí obchádzku – pokutový okruh
- vymieňajú si obsah svojich pamätí

6.1.4 Meranie

Pre potreby výpočtov, tvorby grafov, ktoré slúžia na prezentáciu výsledkov experimentov ale hlavne pre záverečné porovnanie experimentu I. a II. a ich vyhodnotenie sa merajú nasledovné údaje:

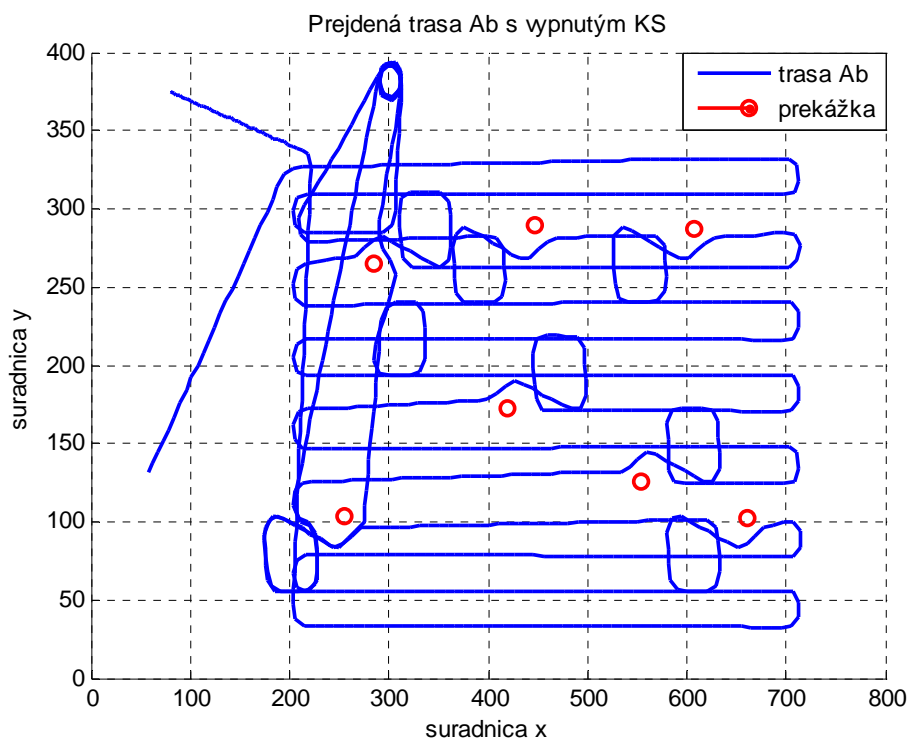
- množstvo spotrebovaného paliva
- počet obídených prekážok s vypnutým a zapnutým KS
- doba simulácie
- prejdená trasa: {súradnica x, súradnica y}
- výška
- rýchlosť

6.1.5 Výstupy experimentov I. a II.

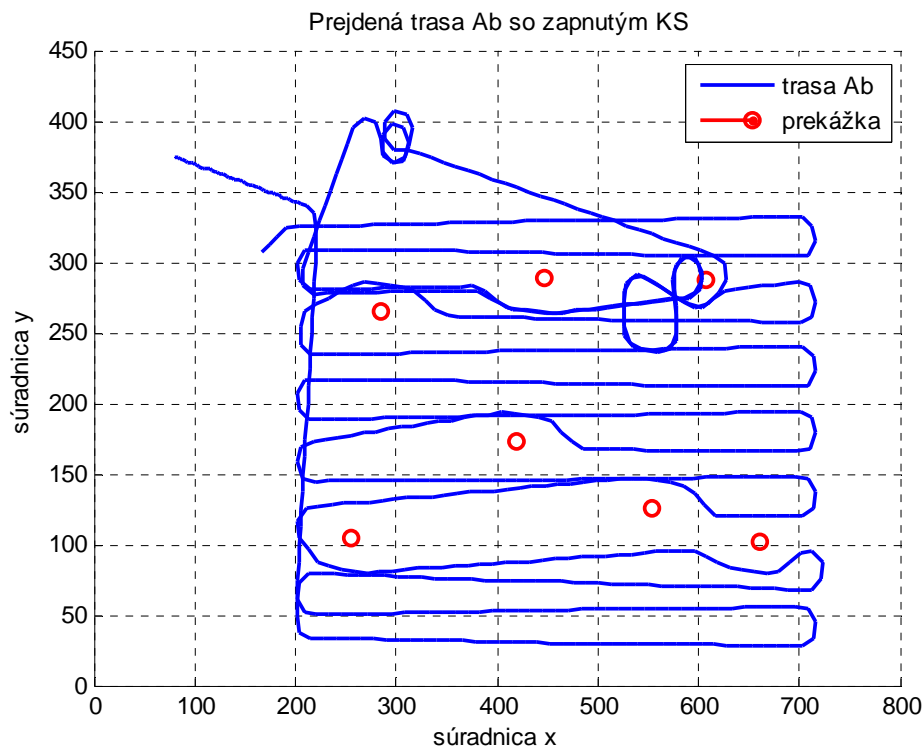
Namerané údaje sú pripravené na následnú grafickú prezentáciu. Nasledujúce údaje zobrazené v grafickej prezentácii porovnávajú rozdiel medzi experimentom I. a II. Do grafov sa vyniesli namerané údaje (Kap. 6.1.4), z ktorých sme dostali nasledujúce grafické zobrazenia.

a) Experiment I. – pri identifikácii každej prekážky urobil Ab jeden pokutový okruh, pretože nemal zapnuté KS (Graf 1 – priemet 3D trajektórie do roviny základne)

b) Experiment II. – so zapnutým KS Ab neurobil pokutový okruh na identifikáciu prekážky nakoľko ju už „poznal“ z mapy a efektívne ju obletel s minimálnou zmenou kurzu a minimálnych vychýlení sa z trasy (Graf 2 – priemet 3D trajektórie do roviny základne)

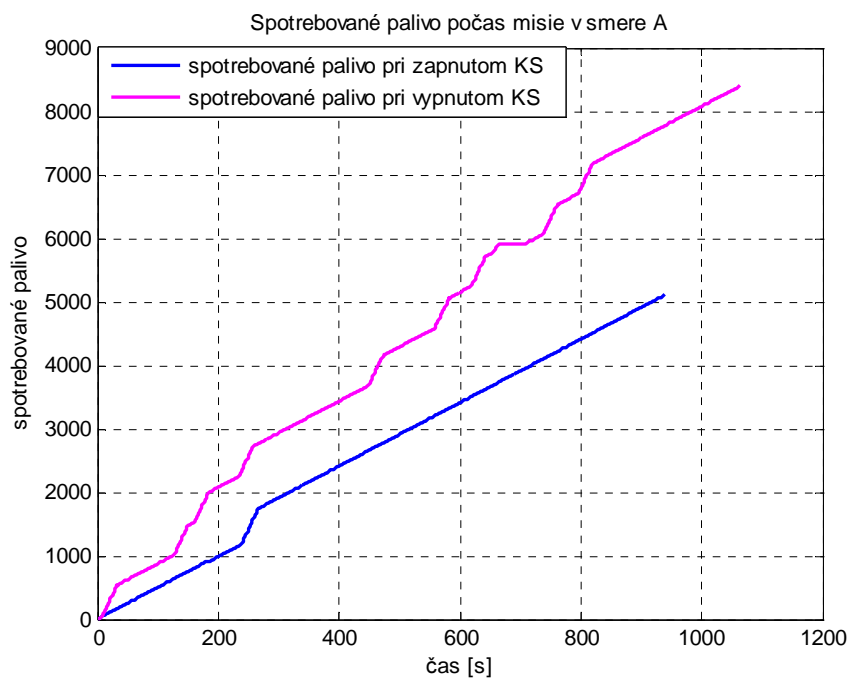


Graf 1 – Grafické zobrazenie pohybu Ab s vypnutým KS v smere A



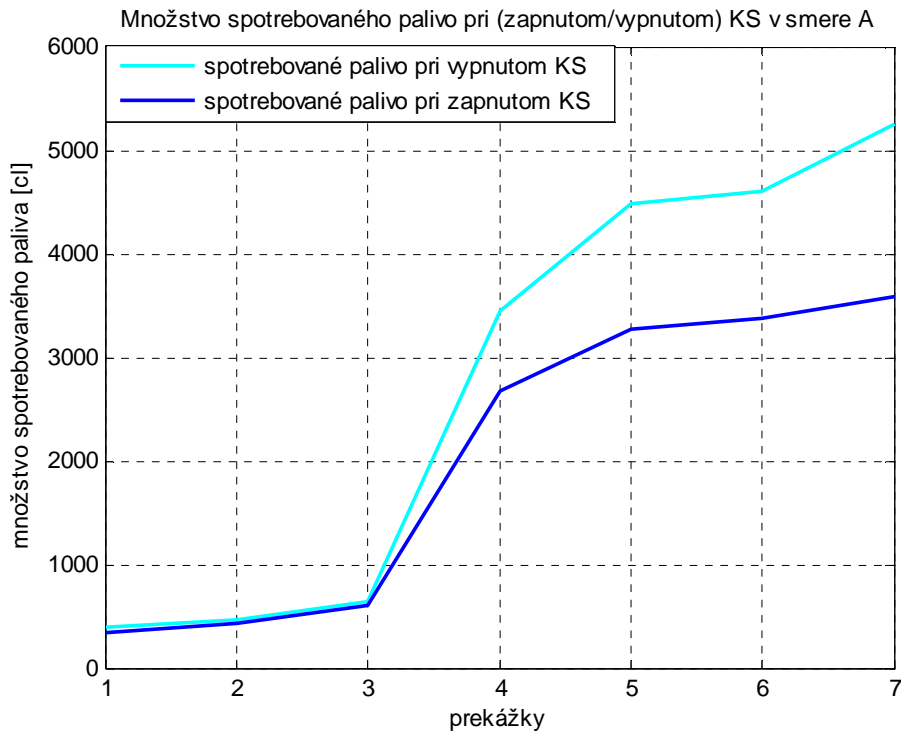
Graf 2 – Grafické zobrazenie pohybu Ab so zapnutým KS v smere A

Graf 3 je vyhodnotenie množstva spotrebovaného paliva pri zapnutom a vypnutom KS, počas experimentov I. a II.

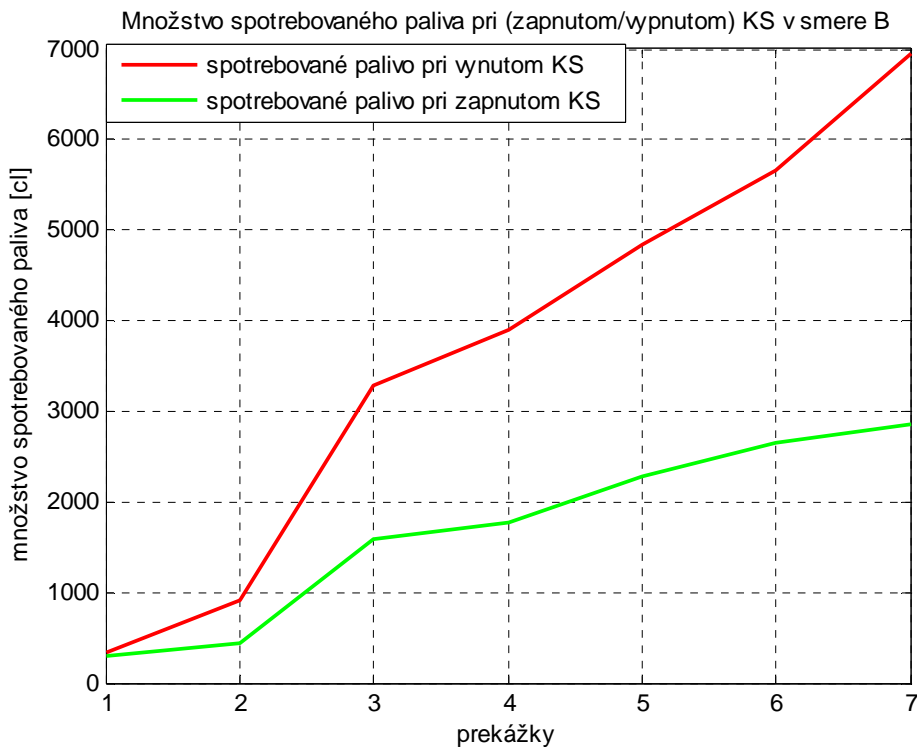


Graf 3 – Zobrazuje množstvo spotrebovaného paliva pri zapnutom/vypnutom KS

Ďalšie dva grafy (Graf 4, 5) predstavujú výsledné rozdiely oboch experimentov pri rôznych smeroch misie (A, B) a (vypnutom/zapnutom) KS, ktorým sme ukázali, že experiment so zapnutým KS je v oboch prípadoch a v rôznych smeroch misie (A, B) pohybu Ab efektívnejší, keď má zapnuté KS.



Graf 4 – Grafické zobrazenie rozdielu experimentu I. a II. v smere A



Graf 5 – Grafické zobrazenie rozdielu experimentu I. a II. v smere B

6.1.6 Zobrazenie

Namerané výstupy pre ilustráciu priebehov experimentu II. Obrázky zobrazujú dva pohľady:

- pohľad na mapu
- pohľad z Ab

6.1.6.1 Zobrazenie experimentu II.

Stratégiu tohto experimentu so zapnutým KS je, že obidva Ab letia do oblasti práškovania, kde sa nachádzajú pre nich neznáme prekážky. Keď prvý Ab narazí na prekážku, vykoná predpísaný okruh pred neznámou prekážkou **ale na rozdiel od experimentu I. zapíše túto informáciu do mapy**. Tak druhý Ab už má informáciu o prekážke a môže ju obletieť úsporným manévrom len s minimálnou zmenou kurzu.

7. VYHODNOTENIE EXPERIMENTOV

Počas experimentov I. a II. boli získané údaje uvedené v Tab. 10.

Tab. 10 – zobrazuje rozdiel medzi experimentom I. a II.

	GRAF 4		GRAF 5	
	Experiment I.		Experiment II.	
	Ab1	Ab2	Ab3	Ab4
Čas simulácie [s]	1100	1100	750	750
Spotrebované palivo [cl]	8398.4000000 0975	9000.20000001 0644	4483.49999999 9323	5108.90000000 1379

Pričom:

Ab1 a Ab4 – sa pohybovali v smere A (vodorovne)

Ab2 a Ab3 – sa pohybovali v smere B (zvisle)

Kde **rozdiely** spotrebovaných množstiev paliva sú nasledovné:

Smer A

Medzi Ab1 (vypnuté KS) a Ab4 (zapnuté KS)

$$x = 8398.4 - 5108.9 = 3289.5 \text{ [cl]} = 32.895 \text{ [l]} \quad [1]$$

Smer B

Medzi Ab2 (vypnuté KS) a Ab3 (zapnuté KS)

$$x = 9000.2 - 4483.5 = 4516.7 [cl] = 45.167 [l] \quad [2]$$

Spracovaním veličín z experimentov je možné tieto experimenty vyhodnotiť aj percentuálne.

Z Grafu 5 pre smer A vychádza percentuálny rozdiel misie so zapnutým a vypnutým KS je $V = 39.16817 \%$

Rozdiel experimentu I. v smere A je o 1.6438 väčší.

Z Grafu 6 pre smer B vychádza percentuálny rozdiel misie so zapnutým a vypnutým KS je $V = 50.18444 \%$

Rozdiel experimentu I. v smere B je o 2.01 väčší.

Tieto výsledky potvrdzujú teoretické predpoklady a sú v súlade s [8].

8. DISKUSIA K VÝSLEDKOM

Vyhodnotenie výsledkov experimentov potvrdilo naše teoretické predpoklady o výhodnosti zavedenia zdieľaného vedomia.

Využitie princípov softvérového - dátového modelovania pri existencii softvéru s niektorými vlastnosťami 5. gen. softvéru [1] a modelu stavového stroja z DEDS - ov, znova preukázalo [36], [17] svoju vysokú opodstatnenosť

Využitie princípov 5. gen. softvéru pri návrhu a softvérovej realizácii celej tejto práce počas experimentov preukázal vysoký stupeň modifikovateľnosti jednak modelu sveta, taktiež „generovania“ Ab cez stavové stroje softvérového realizované modulárnym princípom, ako aj výhodnosť parametrizácie „scenárového“ riadenia samotných experimentov pomocou časovej postupnosti úloh.

9. SPLNENIE CIEĽOV DIZERTAČNEJ PRÁCE

Ciele dizertačnej práce boli splnené v plnom rozsahu, navyše od predpokladaného riešenia sa podarilo pri softvérovej realizácii aplikovať princípy 5. gen. softvéru, pričom sa využili vlastnosti viacúrovňovej parametrizácie, modulárnej tvorby až prakticky do úrovne generovania samotnej aplikácie. Táto problematika ale bude ďalej rozpracovaná a využitá v ďalšej dizertačnej práci na „reálnejších“ zariadeniach - modeloch lietadiel)[33].

Bola navrhnutá, zrealizovaná a experimentálne overená metóda riadenia kolektívneho správania sa Ab s tzv. kolektívnym „vedomím“, ktoré bolo možné prepínať (vypnuté a zapnuté KS).

Výsledky experimentov s kolektívnym správaním, kde spoločným cieľom bolo zdieľanie informácií, preukázali, že koordinácia skupiny Ab so zapnutým spoločným vedomím je oveľa efektívnejšia a priniesla lepšie výsledky ako bez zapnutého spoločného vedomia. Ab splnili úlohy misie lepšie a rýchlejšie (zapnuté KS) pri existencii spoločného vedomia – vzájomná komunikácia so spoločnou mapou.

10. ZÁVER

Experimentmi I. a II. sme preukázali, že zdieľanie vedomia bude mať oveľa väčší význam v oblasti mobilných robotických systémov (vzdušné – vodné - pozemné), pretože sa preukázali schopnosti robotov spolupracovať v skupine efektívnejšie.

V uvedenej problematike sa pokračuje v ďalších prácach (dizertačných a diplomových), kde misie budú zložitejšie, robotov bude viac ale hlavne prekážky budú dynamické (bude sa meniť ich výskyt a pozícia). Všetky oblasti nasadenia nie sme v súčasnosti schopní kvalifikovane odhadnúť.

LITERATÚRA

- [1] Hantuch I., Reguli D. – Generátor PSR: Metodika a generovanie PSR, Textová dokumentácia 2. časť, Bratislava 1981
- [2] Boscarol I., Boscarol A. T. – Pipistrel d.o.o. Ajdovščina, Slovenia, 2011 [online] Dostupné na internete <<http://www.pipistrel.si/news/pipistrel-big-winner-of-nasa-challenge>>
- [3] Boscarol I., Boscarol A. T. – Pipistrel d.o.o. Ajdovščina, Pipistrel is the most innovative company in the European Union!, Slovenia, 2011 [online] Dostupné na internete <<http://www.pipistrel.si/news/pipistrel-is-the-most-innovative-company-in-the-european-uni>>
- [4] Vývoj dopravy do r. 2024 – Budúcnosť ULL: „Automotive Simulation Centre Stuttgart (ASCS)“, [online] Dostupné na internete <<http://www.uni-stuttgart.de/home/>>
- [5] Univerzita Heidelberg – Ruprecht-Karls-Universität: Vývoj elektrických zdrojov pohonu, 2011 [online], dostupné na internete <<http://www.uni-heidelberg.de/forschung/transfer/industrie/>>
- [6] informácia online (2011) - http://www.phoenixair.cz/Phoenix_Air_main.htm
- [7] PhoenixAir na NASA CAFE, 2011, [online], dostupné na internete <http://www.nasa.gov/offices/oct/early_stage_innovation/centennial_challenges/general_aviation/index.html>

- [8] Cole, D. T., Thompson, P., Göktoğan, A. H., Sukkarieh S. – System Development and Demonstration of a Cooperative UAV Team for Mapping and Tracking, *The International Journal of Robotics Research*, 29(11) 1371–1399, § The Author(s) 2010, Reprints and permission: sagepub.co.uk/journalsPermissions.nav DOI: 10.1177/0278364910364685, ijr.sagepub.com
- [9] Chandler P.R., Patcher M., Swaroop D., Fowler J. M., Howlett J. K., Rasmussen S., Schumacher C., Nygard K. – Complexity in UAV Cooperative Control in *Proceedings of the American Control Conference*, Anchorage AK, May 2002
- [10] Boskovic, J. D., Prasanth, R., Mehra, R. K. – A multilayer control architecture for unmanned aerial vehicles, *Proceedings of the American Control Conference*, Anchorage, AK, May 2002, pp. 1825-1830
- [11] Bourgault, F., Furukawa, T., Durrant-Whyte, H. F. – Decentralized Bayesian negotiation for cooperative search, *International Conference on Robotics and Automation*, 2004
- [12] Bryson, M. T., Sukkarieh, S. – Co-operative localization and mapping for multiple UAVs in unknown environments, *IEEE/AIAA Aerospace Conference*, Big Sky, MT March 2007
- [13] Koššuth F. – Agentovo orientované riadenie systémov diskretných udalostí, *Dizertačná práca* (1999)
- [14] Šebo T., Hantuch I., Hantuch I. jr. – GO-CAD: Virtual Reality Simulation *IFAC Conference „Control System Design 2003“* Smolenice, 2003, Pergamon Press 2004
- [15] Jakab A. – *Diplomová práca: Modelovanie kolektívneho chovania sa robotov s decentrálnym riadením*, 2004
- [16] Halász V. – *Diplomová práca: Modelovanie kolektívneho chovania sa robotov s centrálnym riadením*, 2004
- [17] Števo S., Hantuch I. jr., Mikula M. – Využitie modelovania IB vo virtuálnej realite pri prevádzke budov, *Facility Managment 2005*, Zborník prednášok z 3.konferencie so zahraničnou účasťou, *Stavebná fakulta STU Bratislava*, November 2005, ISBN 80-89216-03-X, Str.84-86
- [18] Hantuch I., Koššuth F. – Možnosti využitia spätnej väzby v cykle tvorby PSR, *Riadenie v energetike '96*, Bratislava, Máj 1996
- [19] Steiner R. – *Aus der Akasha Chronik* (2008), ISBN: 8086340272
- [20] Boscarol I., Boscarol A. T. – Pipistrel d.o.o. Ajdovščina, Slovenia, 2011 [online] Dostupné na internete <<http://www.pipistrel.si/about-us/history>>
- [21] odborný seminár I. Hantucha konaný na URPI IKS FEI STU v Júny 2009 “Poznatky a závery z návštevy firmy PIPISTREL Vizionárňa v Ajdovščine – Slovinsko máj 2009“
- [22] odborný seminár I. Hantucha konaný na URPI IKS FEI STU Sep. 2011 “Poznatky a závery z návštevy firmy PhoenixAir Ústí nad Orlicí, Česká republika
- [23] Defense Advanced Research Projects Agency, Arlington, 2012, [online] Dostupné na internete <[http://www.darpa.mil/OurWork/DSO/Programs/Nano Air Vehicle %28NAV%29.aspx](http://www.darpa.mil/OurWork/DSO/Programs/NanoAirVehicle%28NAV%29.aspx)>

- [24] Vyžiadaná prednáška, IPU ANZSSR Moskva, akad. Cypkin, odd. prof Ickoviča – Problemy syntezi program upravlenia, jan1977
- [25] Zaloga J. S. – Unmanned Aerial Vehicles: Robotic Air Warfare 1917 - 2007, str. 44, Great Britain, ISBN 978-1-84603-243-1, 2008
- [26] Nonami K., Kendoul F., Suzuki S., Wang W., Nakazawa D. – Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles, 2010, str. 161, ISBN 978-4-431-53855-4, Tokyo
- [27] Budiyo A., Riyanto B., Joelianto E., Intelligent Unmanned Systems – Theory and Applications, časť na str. 101 autori: Nguyen V. Q., Park Ch. H., Goo S. N., Byun D. – An Insect-like Flapping-Wing Device Actuated by a Compressed Unimorph Piezoelectric Composite Actuator, ISBN 978-3-642-00263-2, 2009
- [28] Craig B. A., Sherman R. W., Will D. J. – Developing Virtual Reality Applications, 2009, ISBN 978-0-12-374943-7, USA
- [29] Qu Z. – Cooperative Control of Dynamical Systems: Application to Autonomous Vehicles, University of Central Florida, Orlando, 2009 ISBN 978-1-84882-325-9
- [30] Grundel D., Murphey R., Pardalos P., Prokopyev O. – Advances in Unmanned Aerial Vehicles: Control And Optimization, 2007, USA, ISBN978-3-540-48270-3, str. 20
- [31] Valavanis P. K., Advances in Unmanned Aerial Vehicles – State of the Art and the Road to Autonomy, časť na str. 309 autori: Nikolos K. I., Tsourveloudis C. N., Valavanis P. K., Chapter 10: Evolutionary Algorithm Based Path Planning for Multiple UAV Cooperation, 2007, USA, ISBN 978-1-4020-6113-4
- [32] Nissen D. H. – Instrumentation and Control of Unmanned Air Vehicles, PhD thesis, Denmark, 2002, ISBN 87-91184-15-0
- [33] Projekt pripravovanej dizertačnej práce Bc. Ľ. Čapuchu
- [34] Hugh J. – Automating Manufacturing Systems with PLCs, 2005, Version 4.7, April 14
- [35] E. A. Parr – Programmable Controllers An engineer's guide, Third edition, ISBN 0 7506 5757 X, 2003, Great Britain
- [36] Šebo T. – Dizertačná práca: Modelovanie vo virtuálnej realite, FEI STU, Bratislava 2004