

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ**



B A K A L Á Ř S K Á P R Á C E

Distribuované řízení formace kvadkoptér AR.Drone

2013

Michal Kapral

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Michal Kapral**

Studijní program: Kybernetika a robotika

Obor: Systémy a řízení

Název tématu: **Distribuované řízení formace kvadrokoptér AR.Drone**

Pokyny pro vypracování:

1. Vytvořte aktuální přehled výukových a výzkumných projektů využívajících komerční a částečně otevřenou platformu AR.Drone, a to zejména se zaměřením na formace těchto kvadrokoptér.
2. Navrhněte a implementujte řešení pro částečně distribuované řízení formace pěti kvadrokoptér, přičemž jedna kvadrokoptéra (lídr) bude řízena přímo lidským operátorem, a ostatní čtyři se budou řídit samy v závislosti na vizuálním sledování sousedů a/nebo lídra či přímo i bezdrátové komunikace s ním.
3. Realizujte experimenty s letem ve formaci v různých konfiguracích. Necht' mezi nimi je i jednorozměrný řetězec (angl. string) a pro ni demonstруйте některé známé jevy jako je kupříkladu "stringová nestabilita".
4. Analyzujte možnosti rozšíření palubní instrumentace platformy AR.Drone dodatečným hardwarem s motivací zlepšit schopnost vzájemné lokalizace kvadrokoptér ve formaci. Jednou možností je ultrazvukové měření vzájemné vzdálenosti. V případě proveditelnosti takové rozšíření realizujte.

Seznam odborné literatury:

- [1] J. Halgašík, J. Zemánek, M. Hromčík and Z. Hurák. Affordable multiUAV experiments using AR.Drone quadrotors. Zasláno na ECC 2012.
- [2] R. Mahony, V. Kumar, and P. Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. IEEE Robotics Automation Magazine, 19(3):33–45, September 2012.
- [3] P.-J. Bristeau, F. Callou, D. Vissiere, and N. Petit. The navigation and control technology inside the AR.Drone micro UAV. In Proceedings of the 18th IFAC World Congress, pages 1477–1484, August 2011.

Vedoucí: Ing. Zdeněk Hurák, Ph.D.

Platnost zadání: do konce zimního semestru 2013/2014

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry



prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 1. 2. 2013

Abstrakt

Táto bakalárska práca sa zaoberá využitím komerčne dostupných malých kvadrokoptér Parrot AR.Drone 2.0 pre výskum synchronizovaného letu vo formácii. Vedúci skupiny / formácie preposiela akčné zásahy svojím nasledovníkom a formácia sa synchronizuje pomocou detekcie tágu, ktorým je označená každá kvadrokoptéra. Práca sa zameriava predovšetkým na spracovanie programovej časti. Program dáva používateľovi možnosť zasahovať do vnútorných parametrov regulátorov a je vybavený sofistikovaným debugovacím systémom. Práca pre úplnosť poskytuje prehľad hlavných komerčných a akademických projektov v oblasti malých kvadrokoptér.


Abstract

This thesis deals with use of a commercial open API quadcopter called Parrot AR.Drone 2.0 for synchronized flight in formation. The formation leader forwards the commands to its followers and the whole formation is synchronized via a visual detection of a tag attached to each quadrotor. The main focus of the thesis is on the programming part. The program controls the whole system and provides an access to the inner variables of the controller through a sophisticated debugging system. This thesis also provides a survey of the most popular indoor quadcopters.

Prehlásenie

Prehlasujem, že som svoju bakalársku prácu vypracoval samostatne a použil som len podklady uvedené v priloženom zozname.

V Prahe dňa 22.5.2013



podpis

PodĎakovanie

PodĎakovanie patrí môjmu vedúcemu bakalárskej práce Ing. Zdeňkovi Hurákovi, Ph.D. za vložení dôveru a cenné rady, ktorými ma inšpiroval a usmerňoval pri jej vypracovaní. Rád by som sa týmto tiež poďakoval svojim rodičom a manželke za podporu, pochopenie a mnohokrát aj inšpiráciu a konzultácie pri písaní tejto práce, ale i počas celého štúdia.

Obsah

1	Úvod, motivácia	1
2	Kvadroptéry	2
2.1	História	2
2.2	Súčasný modely	3
2.3	Konštrukčné vlastnosti	7
2.4	Použitie	8
2.4.1	Výskumná platforma	8
2.4.2	Vojenská platforma	8
2.4.3	Komerčná platforma	9
2.5	Formácie	9
3	Parrot AR.Drone 2.0	10
4	Analýza možnosti rozšírenia platformy AR.Drone 2.0.....	12
4.1	Rozšírenie pomocou Arduino	12
4.1.1	Návod	12
5	Distribúované riadenie	15
5.1	Druhy algoritmov	15
5.1.1	Algoritmus preposielania akčného zásahu.....	15
5.1.2	Algoritmus detekovania tagu pomocou kamery.....	16
5.2	Spôsoby kompenzácie polohy	16
5.2.1	Regulátory	17
6	Programové prevedenie	18
6.1	Nadstavba na YADrone	18
6.1.1	Rozšírenie projektu pre viacero pripojených kvadroptér	19
6.2	Grafické užívateľské prostredie – GUI.....	Error! Bookmark not defined.
6.2.1	Hlavné ovládacie okno	20
6.2.2	Nastavenia.....	20
6.2.2.1	Debug – Logger	21
6.3	Ovládanie	22
6.4	Convoy.....	23
6.4.1	Vláknová štruktúra	23
6.4.2	Načítanie nastavení.....	23
6.5	Optimalizácia.....	24
7	Experimenty	25
7.1	Manuálne ovládanie formácie	25
8	Záver.....	27
	Zoznam použitej literatúry	29
	Príloha.....	31

Zoznam obrázkov

Obrázok 1 De Bothezatová kvadroptéra, 1923	3
Obrázok 2 Draganfly X 4-P	4
Obrázok 3 Microdrones model md4-1000.....	5
Obrázok 4 MikroKopter model QuadroKopter XL.....	5
Obrázok 5 Kvadroptéra Starmac (vľavo) a systém Starmac (vpravo)	6
Obrázok 6 Schéma reakčných momentov na každom motore.	7
Obrázok 7 Parrot AR.Drone 2.0.....	10
Obrázok 8 Konfigurácia pinov na matičnej doske AR.Drone 2.0	13
Obrázok 9 Schéma zapojenia napäťového zosilňovača a dosky Arduino na konektory AR.Drone 2.0	13
Obrázok 10 Funkčné pripojenie ultrazvukového snímača na AR.Drone 2.0.....	14
Obrázok 11 Názorná schéma práce systému formácie kvadroptér	15
Obrázok 12 Názorná ukážka práce detekcie tágu	16
Obrázok 13 Grafické znázornenie prerozdelenie kompenzačných rýchlostí pre vychýlenie do strán ..	17
Obrázok 14 Znázornenie vstupov regulátora a výstupných príkazov, ktoré regulátor poskytuje	17
Obrázok 15 YADrone hlavné okno – riadiace centrum	18
Obrázok 16 Convoy/Options/AR.Drone – nastavenia ARDrone	21
Obrázok 17 Convoy/Options/Logger – nastavenia loggeru	22
Obrázok 18 Convoy/Options/Control – nastavenie ovládania	22
Obrázok 19 Práca jednotlivých vlákien	23
Obrázok 20 CPU záťaž jednotlivých metód počítaná z celkovej záťaže aplikácie Convoy	24
Obrázok 21 Nasledovanie detekovaného tágu	25
Obrázok 22 Let formácie štyroch kvadroptér	26

1 Úvod, motivácia

Kvadrokoptéra je zaujímavý lietajúci stroj, v ktorom vidím v budúcnosti veľký potenciál pre jej možné využitie vďaka špecifickému spôsobu lietania. Jej zdokonaľovaním sa zaoberá množstvo špičkových vedeckých tímov na celom svete. Moderné kvadrokoptéry sú populárne vo výskume bezpilotných vzdušných prostriedkov UAV, čo je skratka anglického unmanned aerial vehicle a znamená bezpilotné lietadlo.

Veľkou motiváciou pre zvolenie témy z tejto oblasti je, že kvadrokoptéry v niektorých prípadoch dokážu uľahčiť prácu človeka. Preto by som chcel dokázať, že aj s kvadrokoptérmi je možné lietať vo formácii, čo v reálnom živote môže byť prospešné pri prieskume neznámeho prostredia, či lietanie v obmedzenom priestore.

Hlavným cieľom mojej práce je vytvoriť letuschopnú formáciu piatich kvadrokoptér firmy Parrot, pričom jedna kvadrokoptéra, vedúca skupiny, bude riadená ľudskými zásahmi a nasledovníci budú autonómne nasledovať vedúceho skupiny a zároveň udržiavať formáciu. Ich synchronizáciu bude zabezpečovať regulátor, ktorého vstupom budú navigačné dáta, poloha a vzdialenosť detekovaného tágu, ktorý je pripevnený na každej kvadrokoptére. V nasledujúcom kroku som otestoval naimplementovaný systém v rôznych podmienkach a situáciách.

Práca poskytuje aktuálny prehľad výukových a výskumných projektov využívajúcich sa v komerčnej sfére.

Ďalším cieľom, ktorý som si stanovil, je vytvoriť návod na rozšírenie palubnej inštrumentácie platformy AR.Drone dodatočným hardwarom. Tento návod by mal slúžiť pre ďalšie pokračovanie tohto projektu a to tým, že po jeho uskutočnení sa vylepší vzájomná schopnosť lokalizácie kvadrokoptér vo formácii.

2 Kvadrokoptéry

Kvadrokoptéra je lietajúci stroj poháňaný štyrmi rotormi. Je klasifikovaná ako rotorový stroj, ktorý na rozdiel od lietadiel s krídlami využíva vztlak generovaný sadou otočných úzkych vrtúľ. Na rozdiel od ostatných helikoptér, kvadrokoptéry všeobecne využívajú symetricky osadené vrtule, ktoré pracujú ako skupina. Riadenie pohybu vozidla je dosiahnuté menením výšky a zmenou rýchlosti rotácie jedného alebo viacerých vrtúľ, čím sa mení moment záťaže a vztlakové vlastnosti.

Moderné kvadrokoptéry sa stali populárne vo výskume bezpilotných vzdušných dopravných prostriedkov (takéto prostriedky sa nazývajú UAV). Tieto stroje používajú elektronický kontrolný systém a senzory na stabilizáciu lietadla. S ich malou veľkosťou a jednoduchou ovládateľnosťou, môžu byť ovládané v interiéroch aj exteriéroch.

Existuje niekoľko výhod kvadrokoptér v porovnaní s helikoptérmi.

- kvadrokoptéry nevyužívajú mechanické väzby na menenie uhlu sklonu rotora.
- použitie štyroch rotorov umožňuje zmenšiť dĺžku listu vrtule, čo im umožňuje mať menej kinetickej energie počas letu. To znižuje možné poškodenie spôsobené nárazom vrtule. Pre malé bezpilotné lietadlá to umožňuje manévrovať v úzkych priestoroch.

2.1 História

V minulosti boli kvadrokoptéry vnímané ako možné riešenie na niektoré trvalé problémy v priamom lete. Je jednoduchšie zostrojiť viacero menších rotujúcich listov, ako jeden veľký a zadný rotor, ktorý nevytvára žiadny vztlak. Prvé návrhy sa objavili v rokoch 1920 až 1930. Tieto stroje boli medzi prvými úspešnými lietajúcimi strojmi. Avšak, prvé prototypy trpeli na malý výkon a veľkú nestabilitu. Bol potrebný príliš veľký zásah pilota.

- **Etienne Oehmichen** experimentoval s rotorovými lietadlami v dvadsiatych rokoch dvadsiateho storočia. Spomedzi jeho šiestich návrhov, jeho kvadrokoptéra číslo 2 mala štyri rotory a osem vrtúľ, všetky poháňané jedným motorom. Druhá Oehmichenová kvadrokoptéra používala rám z oceľových trubiek s dvojčepelovými rotormi na koncoch štyroch ramien. Uhol týchto čepiel sa mohol líšiť. Päť vrtúľ, točiacich sa v horizontálnej rovine, stabilizovalo stroj. Ďalšia vrtuľa bola namontovaná na nose pre riadenie. Zostávajúce vrtule boli pre predný pohon. Lietadlo vykazovalo značné miery stability a ovládateľnosti na svoju dobu a urobilo viac ako tisíc skúšobných letov. Roku 1923 bolo schopné zostať vo vzduchu po dobu niekoľkých minút.

- **Dr. George de Bothezat a Ivan Jerome** navrhli lietadlo so šesť čepeľnými rotormi na koncoch rámu tvaru písmena X. Dve malé vrtule s premenným náklonom boli použité pre ťahové a vychýľovacie riadenie. Toto lietadlo využívalo hromadné riadenie náklonu. Kvadrokoptéra spravila svoj prvý let v októbri 1922. Do konca roka 1923 spravila okolo 100 skúšobných letov. Najvyššia výška, ktorú dosiahla bola 5 metrov. Konštrukcia bola mechanicky náročná a náchylná k problémom so spoľahlivosťou. Pracovná záťaž na pilota bola príliš náročná aby sa pokúsil o bočný let.



Obrázok 1 De Bothezatová kvadrokoptéra, 1923

Zdroj: www.wikipedia.org

- **De Bothezatová kvadrokoptéra, 1923** - tento jedinečný vrtuľník bol zamýšľaný byť prototypom pre rad oveľa väčších civilných aj vojenských kvadrokoptér. Dizajn predstavoval dva motory poháňajúce štyri rotory s krídlami, ktoré boli pridané pre ťah v lete dopredu. Toto lietadlo bolo prvou štvor-rotorovou kvadrokoptérou, ktorá ukázala úspešný let dopredu. Kvôli nedostatku zákaziek na obchodných alebo vojenských trhoch bol projekt ukončený.
- **Curtiss-Wright VZ-7** bolo lietadlo navrhnuté pre americkú armádu. VZ-7 bolo kontrolované pomocou zmeny ťahu každej zo štyroch vrtúľ.

2.2 Súčasné modely

V posledných niekoľkých desaťročiach, sa stali malé bezpilotné lietadla UAV viac používané pre mnoho aplikácií. Potreba lietadiel s väčšou manévrovateľnosťou viedla k značnému výskumu v oblasti kvadrokoptér. Štvor-rotorový dizajn umožňuje kvadrokoptéram byť relatívne jednoduchými, ale veľmi spoľahlivými a obratnými. Špičkový výskum stále zvyšuje životaschopnosť kvadrokoptér, zameraním sa na preskúmavanie prostredí a manévrovateľnosť. Ak spojíme všetky tieto vývojové kvality, kvadrokoptéry môžu byť schopné pokročilých autonómnych operácií, ktoré nie sú v súčasnosti uskutočniteľné inými lietadlami.

Niektoré súčasné programy:

- **Aermatica Spa 's Anteos** je prvý vrtuľník typu RPA (diaľkovo pilotovaných lietadiel), ktorý získal oficiálne povolenie k letu vydané v civilnom vzdušnom priestore a bude ako prvé pracovať v nevymedzenom vzdušnom priestore.
- **AeroQuad** je open-source hardwarový a softwarový projekt, ktorý využíva Arduino dosky a voľne poskytuje dizajn hardwaru s softwaru pre konštrukciu kvadrokoptéry
- **ArduCopter** je open-source multicopter (viac rotorový) UAV, založený na Arduino. Podporuje až osem motorov, rovnako ako tradičné helikoptéry a umožňuje plne autonómnu prácu, ale aj ovládanie RC. Jednou zo zaujímavých funkcií je tzv. „return to lunch“ teda návrat na miesto odletu. Výrobca ponúka aj spracované užívateľské rozhranie. Kompletne zdrojové kódy ale aj popis projektu sa nachádza na <https://code.google.com/p/arducopter/wiki/ArduCopter>.
- **Draganfly** je komerčná platforma, ktorá poskytuje rýchle, nízkonákladové kvadrokoptéry, ktoré sú schopné posilať vysokokvalitné informácie na základňu na zemi v reálnom čase. Ich súčasnými modelmi sú ES, X6 či X4-P (obrázok nižšie). Ich lietadlá sú spoľahlivé, pretože sú vyrobené z vysoko kvalitných uhlíkových vlákien a liatych komponentov. Draganfly bol navrhnutý, aby vedel lietať v exteriéroch ale aj v interiéroch s obmedzenou možnosťou pohybu. Napríklad model X4-P je vybavený jedenástimi senzormi, medzi ktoré aj napríklad GPS lokátor. Jeho konštrukcia má 87 x 87 x 32 cm a dokáže odniesť záťaž do 0,7 kg. Zaujímavou funkciou Draganfly kvadrokoptér je, že ich môžeme zaparkovať vo vzduchu, to znamená že kvadrokoptéra sa nebude hýbať a môže napríklad fotografovať. Viac informácií na <http://www.draganfly.com/>.



Draganflyer X4-P

Obrázok 2 Draganfly X 4-P

Zdroj: <http://www.draganfly.com/>

- **The microdrones GmbH** je komerčná spoločnosť, ktorej produkty sú určené na vysoko náročné operácie v oblasti leteckého prieskumu. Palubný štandard tejto spoločnosti je poskytovanie vysokorozlíšených fotografií, videa, termografia alebo multispektrálne senzory microdrones, ktoré zjednodušujú prácu geodeta a zememeračov. Letový čas niektorých modelov je stanovený až na 88 minút a dokážu lietať v teplotách od -20°C do +50°C. Veľkou výhodou microdrones kvadrokoptér je, že sú odolné voči vode a prachu. Viac informácií nájdete na www.microdrones.com.



Obrázok 3 Microdrones model md4-1000

Zdroj: <http://www.microdrones.com>

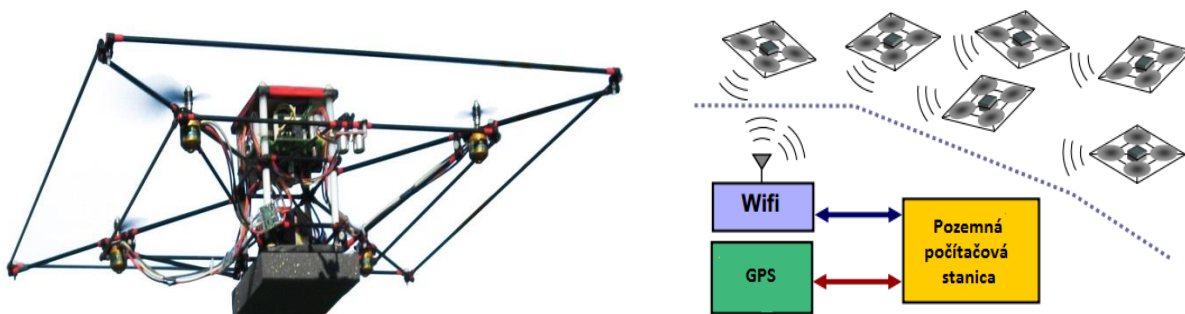
- **Open-source platforma MikroKopter**, poskytuje lietadlá, ktoré sa využívajú v zákazkách pre verejný sektor, t.j. ako držiak pre kamery a senzory. Majú k dispozícii QuadroKopter so štyrmi, HexaKopter so šiestimi alebo OktoKopter s ôsmimi rotormi. MikroKopter môže byť vybavený GPS a následne je schopný automaticky udržať pozíciu a výšku. Letové časy sú v závislosti na integrovanej hmotnosti v rozsahu 15 až 40 minút. MikroKopter unesie až 2,5 kg. Viac informácií na www.mikrokopter.de.



Obrázok 4 MikroKopter model QuadroKopter XL

Zdroj: <http://www.mikrokopter.de/ucwiki/MK-QuadroXL>

- Projekt FMA** – jednou z najvýznamnejších osobností, ktoré sa zaujímajú o let formácii kvadrokoptér je profesor Dr. Raffaello D'Andrea z ETH v Zürichu. Jeho projekt FMA (aréna lietajúcich strojov) je prenosný priestor (10 x 10 x 10 metrov) zariadený pre autonómny let. Priestor pozostáva z vysoko presného systému zachytávania pohybu (anglicky motion capture), bezdrôtovej komunikačnej siete a programového vybavenia vykonávajúceho sofistikované algoritmy pre odhad a kontrolu. Systém zachytávania pohybu môže nájsť viacero objektov v priestore s rýchlosťou viac ako 200 snímok za sekundu. Aj keď sa to môže zdať veľmi rýchle, objekty v priestore sa môžu pohybovať rýchlosťou väčšou ako 10 m/s, čo má za následok posunutie väčšie ako 5 cm medzi nadväzujúcimi snímkami. Táto informácia sa miesi s ďalšími dátami a modelom dynamiky systému, čo umožňuje predpovedať stav objektov do budúcnosti. Systém používa tieto poznatky aby zistil, aké príkazy má kvadrokoptéra vykonávať k dosiahnutiu ich požadovaného správania, čo môže byť napríklad vyrovňovanie objektov, vysokorýchlostný vzlet, stavebné konštrukcie. Príkazy sú nasledovne odoslané do palubných počítačov, kde sú ďalej spracované. Funkčnosť projektu FMA je možné vidieť vo videu na internetovej stránke <http://vimeo.com/58001498>.
- Starmac** – jedná z prvých akademických platforiem a bola vyvinutá na Standfordskej univerzite. Cieľom tohto projektu bolo dokázať niektoré teórie hybridných systémov a hlavne dokázať či demonštrovať analytické výhody prehľadávania priestoru v reálnom svete. Rám kvadrokoptéry je vyrobený z karbonových vlákien, palubný počítač pozostáva s procesorov PC104 a Atmega128, pre stabilizáciu využíva PID regulátor. Podrobné informácie o tejto platforme sa nachádzajú na <http://www.flyingrobots.org/mavws/slides/08-Hoffmann.pdf>.



Obrázok 5 Kvadrokoptéra Starmac (vľavo) a systém Starmac (vpravo)

Zdroje:

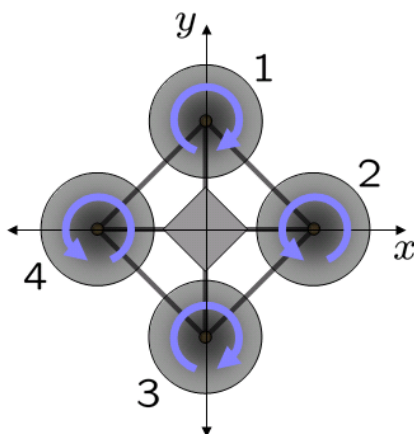
<http://www.flyingrobots.org/mavws/slides/08-Hoffmann.pdf>

<http://autonomous.uwaterloo.ca/waslander/newsmodule/view/id/3/src/@random48888d6be5f4a/>

2.3 Konštrukčné vlastnosti

Uhlové zrýchlenie na náklonovej a otáčacích osiach môže byť spôsobené samostatne, bez ovplyvnenia zvislej osi. Každý pár vrtúľ rotujúcich v rovnakom smere ovláda jednu os a rastúci ťah jedného rotora zároveň znižuje ťah pre ostatné, čo udržiava rovnováhu krútiaceho momentu potrebného pre stabilitu otáčania a navodzujú čistý krútiaci moment na otáčacích a náklonovej osi. Týmto spôsobom môžu pevné listy rotora spôsobovať manévrovanie kvadrokoptéry vo všetkých smeroch. Translačné zrýchlenie sa dosiahne tým, že sa udržiava nenulový náklon alebo uhol rotácie.

- Hlavné mechanické komponenty potrebné pre výstavbu sú rám, vrtule a elektrické motory. Pre dosiahnutie čo najlepších výsledkov a najjednoduchších kontrolných algoritmov, by motory a vrtule mali byť umiestnené v rovnakej vzdialenosti medzi nimi. Rôzne materiály môžu byť použité pre všetky diely. V súčasnosti sa stali populárnymi používať komponenty z karbónového vlákna, vďaka ich nízkej hmotnosti a vysokej konštrukčnej tuhosti.
- Elektrické komponenty potrebné k vytvoreniu funkčnej kvadrokoptéry sú podobné tým, ktoré používajú moderné modelárske vrtuľníky RC. A to elektrické regulátory rýchlosti, palubný počítač, či riadiaca doska a batéria. Obvykle sa pre zásah človeka používa diaľkové ovládanie.
- Každý rotor produkuje ťah a krútiaci moment okolo stredu jeho otáčania, rovnako ako aj posuvnú silu opačnú k smeru letu vozidla. Ak sa všetky rotory točia rovnakou uhlovou rýchlosťou, tak motory jedna a tri sa točia v smere ručičkových hodínok a rotory dva a štyri proti smeru ručičiek. Preto čistý aerodynamický moment a uhlové zrýchlenie okolo osi odchýlenia / vybočenia je presne nula, čo znamená, že nie je potrebný stabilizačný rotor ako je to u konvenčných vrtuľníkov. Vychýlenie je spôsobované nesúlalom rovnováhy v aerodynamických momentoch.



Obrázok 6 Schéma reakčných momentov na každom motore.

Zdroj: www.wikipedia.org

2.4 Použitie

2.4.1 Výskumná platforma

Kvadrokopty sú užitočným nástrojom pre univerzitných výskumníkov. Používajú sa na testovanie a hodnotenie nových myšlienok v mnohých rôznych oblastiach, vrátane teórie riadenia letu, navigácie, systémy reálneho času a robotiky. V posledných rokoch veľa univerzít ukázalo mnohé zlepšenia komplexnej vzdušnej manipulácie. Formácie kvadrokopty sa môžu vznášať vo vzduchu, samostatne vykonávať zložité rutiny ako premety, či iné akrobatické prvky.

Existujú mnohé výhody pre použitie kvadrokopty ako univerzálnych platforiem. Sú relatívne lacné, dostupné v rôznych veľkostiach a ich jednoduché mechanické prevedenie umožňuje ľahkú obsluhu a údržbu aj amatérom. Vzhľadom k multidisciplinárnej povahe prevádzkovania kvadrokopty, musia akademici z rôznych odvetví pracovať spoločne. Tieto projekty sú väčšinou spoluprácou medzi počítačovou vedou, elektrotechnikou a strojárstvom.

Pretože sú kvadrokopty tak obrátne, môžu pracovať vo všetkých druhoch situácií a prostredí. Kvadrokopty schopné autonómneho letu môžu pomôcť odstrániť potrebu ľudí v nebezpečných pozíciách. Toto je aj hlavným dôvodom, prečo výskum stále pokračuje.

V súčasnosti existujú špecializované laboratória pre rozvoj pokročilej riadiacej techniky a aplikácií pre kvadrokopty. Jedná sa najmä o MIT Aerospace Controls Lab, ETH's Flying Machine Area a Pensilvanská univerzita všeobecnej robotiky, automatizácie, senzorov a snímania.

2.4.2 Vojenská platforma

Kvadrokopty bezpilotných vzdušných dopravných prostriedkov sa používajú pre sledovanie a prieskum pre vojenské a policajné útvary, rovnako ako aj pre záchranné, či pátracie misie v mestskom prostredí. Jeden taký príklad je Aeryon Scout, čo je malá UAV kvadrokopty vytvorené kanadskou spoločnosťou Labs Aeryon, ktorá sa môže pokojne vznášať na mieste a používať fotoaparát na pozorovanie ľudí a objektov na zemi. Spoločnosť tvrdí, že stroj hral kľúčovú úlohu v odhalení drogového kartelu v Strednej Amerike a to tým, že poskytoval vizuálny dohľad nad drogovým priekupníkom hlboko v džungli.

2.4.3 Komerčná platforma

Najväčšie použitie kvadrokoptér je v oblasti leteckých snímok. Kvadrokoptéry sú vhodné pre túto prácu, vďaka ich autonómnej povahe a obrovskej úspore nákladov. Zachytávanie leteckých snímok pomocou kvadrokoptér je tak jednoduché, ako naprogramovať GPS súradnice a stlačiť tlačidlo. Používanie palubných kamier dáva užívateľovi možnosť sledovania videa v reálnom čase. Veľa spoločností využíva túto výhodu od fotografovania až po vykonávanie technických inšpekcií.

2.5 Formácie

Formácia je disciplinovaný let dvoch alebo viacerých lietadiel pod velením jedného vodcu. Vojenský piloti používajú útvary pre vzájomnú obranu a maximalizáciu palebnej sily. V civilnom letectve sa formácie používajú len na leteckých dňoch, či pri rekreačnom letaní. Let formácie má vplyv na spotrebu paliva, pretože znižuje odpor, ktorý musia lietadla prekonať.

Najznámejšou formáciou je určite V formácia, ktorú využíva vtáctvo pri migrácii skupín. Ak je v skupine 25 členov, každý člen môže dosiahnuť zníženie odporu až o 65%.

Pre túto prácu sú však skôr zaujímavé formácie, ktoré maximalizujú prehľadanie prostredia. Predstavme si napríklad prehľadávanie väčšej plochy. Ideálna formácia pre túto úlohu je formácia uložená na jednej osi. Kvadrokoptéry sa na takúto úlohu výborne hodia, dokážu prísť na určité miesto, zbierať potrebné dáta a v prípade potreby sa vrátiť naspäť.

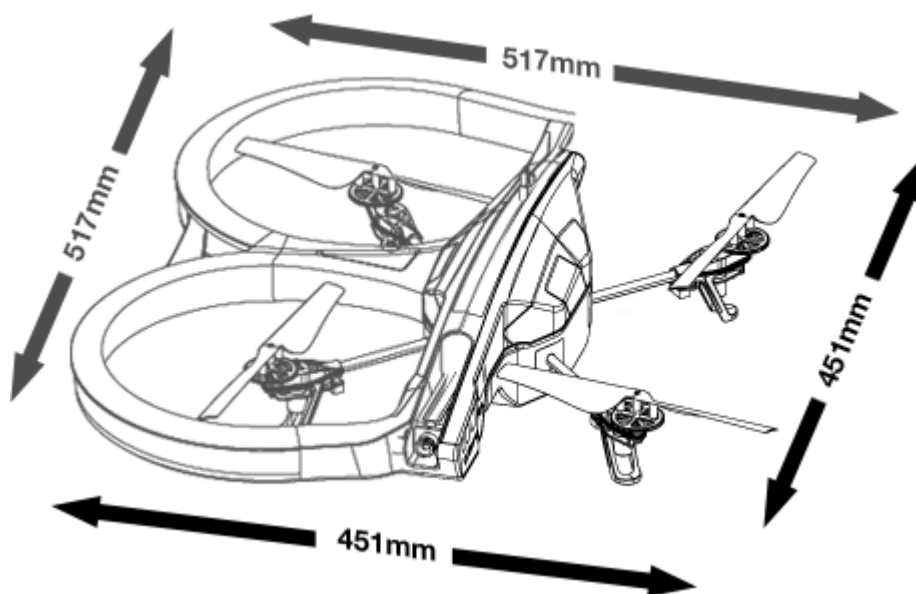
V tejto bakalárskej práci budú kvadrokoptéry lietať v rade za sebou. Táto formácia umožňuje synchronizáciu pomocou vzájomnej viditeľnosti.

3 Parrot AR.Drone 2.0

Kvadrokoptéra od firmy Parrot, ktorá tvorí základ mojej práce, má označenie AR.Drone 2.0. Je vybavená 1GHz ARM Cortex A8 procesorom a systémom Linux 2.6.32. Má 1Gb RAM, USB 2.0 a Wi-Fi. Na palube sa ďalej nachádzajú snímače:

- 3-osý gyroskop s presnosťou 2000°/s
- 3-osý akcelerometer s presnosťou ± 50 mg
- 3-osý magnetometer s presnosťou 6°
- Snímač tlaku s presnosťou ± 10 Pa
- Ultrazvukové senzory pre meranie výšky
- 60fps vertikálna QVGA kamera pre meranie pozemnej rýchlosti

Celková hmotnosť je 380g, s ochranným rámom je to 420g. Jej motorovú časť tvoria 4 elektromotory o výkone 1405W, 28500RMP.



Obrázok 7 Parrot AR.Drone 2.0

Zdroj: www.ar drone2.parrot.com

Vybavenie kvadrokoptéry

- Jediným zdrojom energie je lithium-polymerový akumulátor, je trojčlánkový, s celkovou kapacitou 1000 mAh. Priemerná doba lietania s nabitým akumulátorom je 10 – 12 minút.
- Každá vrtuľa má samostatný elektromotor, tzn., že má 4 striedavé motory s výkonom 15 W, má otáčky 28 000 RPM. Pri najväčšom zrýchlení kvadrokoptéry dosahuje až 41 400 RPM (otáčok za minútu).
- Vrtule nie sú pripevnené priamo na výstupnú hriadeľ motora, ale sú poháňané ozubenými kolieskami.
- Kvadrokoptéra má dve kamery. Jedna smeruje dopredu – čip typu CMOS s rozlíšením VGA, zorný uhol 93°. Druhá kamera je na spodku kvadrokoptéry – čip CMOS s maximálnym rozlíšením OCIF, zorný uhol 64 °.
- Pod krytom kvadrokoptéry sú dve dosky plošných spojov – navigačná a základná.
 - Navigačná, ktorá obsahuje súčiastky poskytujúce údaje o lete – gyroskop Invensense IDG 500 a výškomer.
 - Základná doska, na ktorej je procesor, operačná pamäť, wifi modul atď., zabezpečuje komunikáciu, spracovanie informácií zo senzorov a stabilizáciu. Systém využíva operačnú pamäť DDR s veľkosťou 128 MB s časovaním 200 MHz.
- ultrazvuk – prenos zvukového signálu s frekvenciou 40 kHz na určenie letovej výšky. Je umiestnený na spodnej strane kvadrokoptéry. Odrazom od zeme sa meria letová výška.
- snímače polohy a natočenia - akcelerometre a gyroskopy.
 - Akcelerometre sú senzory merajúce dynamické zrýchlenie – akceleráciu, alebo statické zrýchlenie – polohu v gravitačnom poli predmetu. Pracujú na princípe zmeny kapacity vnútorného premenlivého kondenzátora vplyvom pôsobiacej sily, ktorá vznikla zrýchlením puzdra senzoru.
 - Gyroskopy sa používajú na zistenie zmeny natočenia alebo uhlovej rýchlosti predmetu, ku ktorému sú pripevnené.
- Oficiálne API firmy Parrot nepodporuje formáciu kvadrokoptér.

4 Analýza možnosti rozšírenia platformy AR.Drone 2.0

Oficiálne nie je rozšírenie platformy AR.Drone 2.0 podporované. Napriek tomu existuje niekoľko úspešných pokusov. AR.Drone ponúka dve možnosti ako pridať nový senzor či zariadenie:

- Použiť USB rozhranie. Vhodné je použiť TTL-USB konvertor a správne skompilovať USB sériový ovládač. Potom je možné čítať dáta z nového zariadenia cez sériové rozhranie *ttyUSB0*. K tomuto riešeniu, ktoré sa navonok javí ako pomerne jednoduché sa však neobjavujú žiadne špecifické postupy či manuály.
- Druhou možnosťou je použitie dosky Arduino (viď 4.1).

4.1 Rozšírenie pomocou Arduino

Arduino je open-source platforma založená na mikročítači ATmega od firmy Atmel a grafickom vývojovom prostredí, ktoré vychádza zo známeho prostredia Wiring. Arduino môže byť použité k vytváraniu samostatných interaktívnych zapojení alebo môže byť pripojené k softvéru na počítači (napr. Macromedia Flash, Processing, Max/MSP, Pure Data, SuperCollider). Arduino je pripojované k počítaču pomocou rozhrania USB, ale má aj softwarovo simulované sériovú komunikáciu cez linku RS-232.

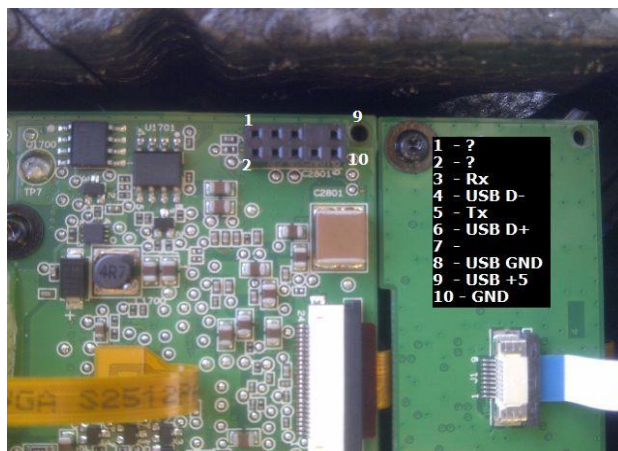
4.1.1 Návod

Najjednoduchší spôsob, ktorým sa dá popísať spôsob rozšírenia našej platformy je vytvoriť návod:

- Ako prvé si musíme vytvoriť podklady pre komunikáciu Arduino s AR.Drone (vytvoriť tzv. otlačok v kvadrokoptére (node_modules)). Je potrebné stiahnuť potrebné ovládače a testovacie programy z internetovej adresy <https://github.com/felixge/node-cross-compiler/downloads>. Následne sa pripojiť na wifi kvadrokoptéry a potom zadať v príkazovom riadku:

```
telnet 192.168.1.1
cd/data/video
chmod +x node
mkdir node_modules
cp -r usb/node-serialport/ node_modules/
```

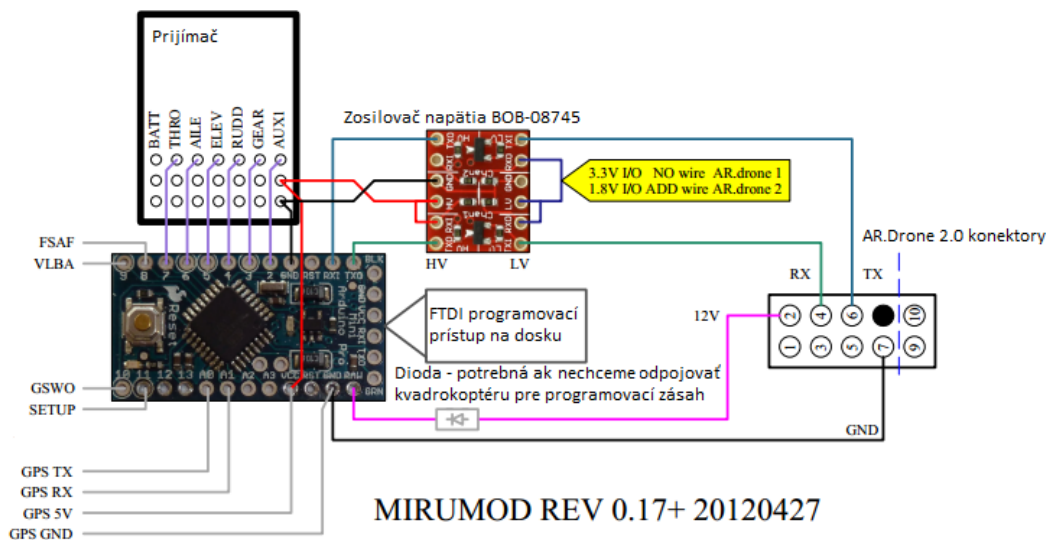
- Ak odstránime spodný kryt kvadrokoptéry, dostaneme sa k jej matičnej doske, na ktorej sa nachádza servisné rozhranie, ktoré má piny pre TTL a USB sériovú komunikáciu (viď obrázok nižšie).



Obrázok 8 Konfigurácia pinov na matičnej doske AR.Drone 2.0

Zdroj: <https://gist.github.com/maxogden/4152815>

- Je nutné si uvedomiť, že piny Tx/Rx majú napätie 1.8V, čo znamená, že budeme potrebovať napäťový prevodník (napr. BOB-08745) pre komunikáciu s Arduino. Ideálna doska je Arduino Uno alebo Due, ktoré poskytujú 5V alebo 3.3V TTL cez digitálne porty Tx/Rx. Zapojíme ho podľa nasledovnej schémy.



Obrázok 9 Schéma zapojenia napäťového zosilňovača a dosky Arduino na konektory AR.Drone 2.0

Zdroj:

<http://people.eecs.ku.edu/~jpince/Project%20Files/Serial%20Port%20&%20Power%20wiring%20diagram.pdf>

- Ďalším krokom je nahrať skúšobného programu *helloworld.pde* do dosky Arduino.

V otvorenom telnete zadáme nasledovné príkazy:

<code>cat /proc/cmdline</code>	- zistí, ktoré TTY zariadenie sa používa, vráti napr. <i>tty03</i>
<code>stty -F /dev/tty -raw</code>	- nastaví tty socket na plný, čistý mód
<code>stty -F /dev/tty03</code>	- overí prenosovú rýchlosť (napr. 9600)

- Prejdeme do zložky *./node* a nakopírujeme tam *helloworld.js*. Po spustení by sme mali dostať výstupný text „hello“.

Ak sme sa dostali až do tohto bodu, tak už úspešne komunikujeme s doskou Arduino cez kvadrokoptéru. Ďalším krokom je pripojenie nového senzoru alebo zariadenia, ktoré chceme pripojiť a správne nastaviť všetky jeho parametre.



Obrázok 10 Funkčné pripojenie ultrazvukového snímača na AR.Drone 2.0

Zdroj: <https://gist.github.com/maxogden/4152815>

Pre viac informácií je možné navštíviť webové stránky, ktoré priamo popisujú a zobrazujú podobné návody:

- <https://gist.github.com/maxogden/4152815>
- <http://forum.parrot.com/ardrone/en/viewtopic.php?pid=22490>

5 Distribuované riadenie

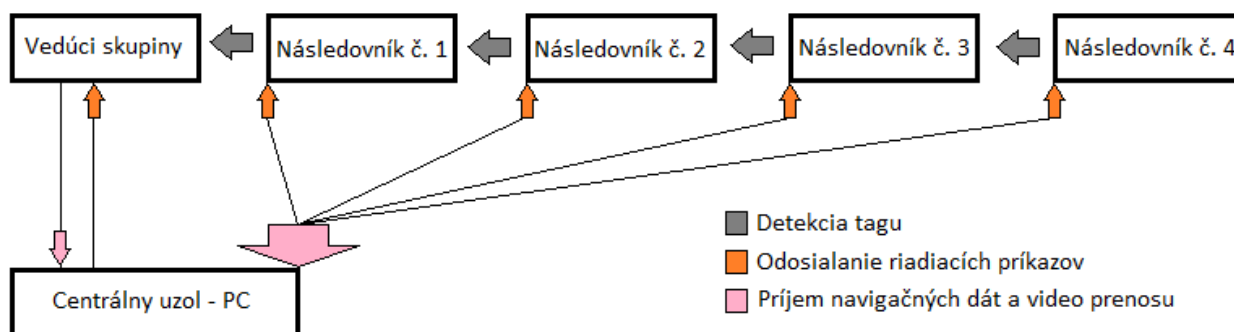
5.1 Druhy algoritmov

V súčasnej dobe sú naimplementované dva druhy synchronizácie:

- Algoritmus preposielania akčného zásahu (viď. 5.1.1)
- Algoritmus detekovania tagu pomocou kamery (viď. 5.1.2)

Oba algoritmy sú navzájom previazané. Základným prvkom je počítač, ktorý zo zeme ovláda celý systém. Priame ovládanie je spojené len s vedúcim dronom. Všetky jeho úkony sú v reálnom čase automaticky preposielané aj jeho nasledovníkom. Detekcia tagu je spustená len pre nasledovníkov. Vedúci skupiny túto možnosť nepotrebuje, respektíve nemá k dispozícii tag k detekcii.

Kvadrokoptéry:



Obrázok 11 Názorná schéma práce systému formácie kvadrokoptér

5.1.1 Algoritmus preposielania akčného zásahu

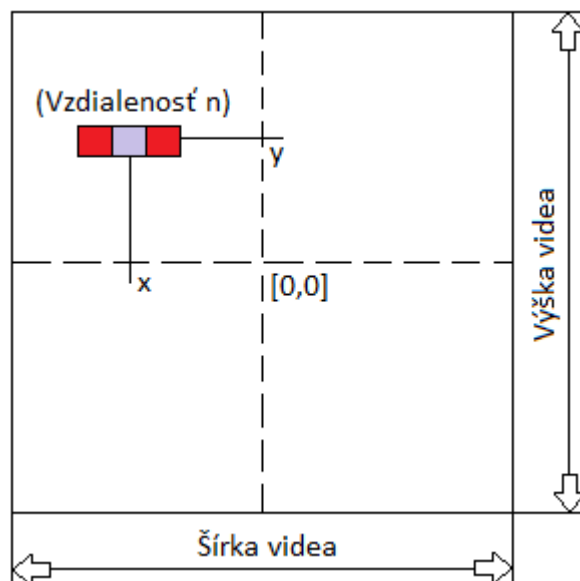
Algoritmus preposielania akčného zásahu pracuje na veľmi jednoduchom princípe. Klávesnicou sa ovláda len vedúci skupiny. Nasledovníkom je ovládací príkaz preposielaný automaticky. Dôležité je zvážiť problém so zastavovaním jednotlivých príkazov. Napríklad, ak odošlem príkaz na pohyb vpred, je potrebné poslať aj príkaz na zastavenie. V opačnom prípade sa kvadrokoptéra nezastaví. Pre vedúceho skupiny je to vyriešené uvoľnením klávesy. Avšak vzhľadom na dobu trvania preposielania tagu je pre príkaz, ktorý trvá len niekoľko milisekúnd, nemožné hneď preposlať príkaz stop – v tomto prípade sa príkaz nevykoná.

5.1.2 Algoritmus detekovania tagu pomocou kamery

Algoritmus zbiera snímky z videa. Vlákno detekcie tagu si vezme nový snímok, vždy keď dokončí detekciu starého. Ostatné snímky sa jednoducho zahadzujú, respektíve priamo prenášajú do grafického prostredia bez detekcie. Výsledkom tohto systému je plné využitie detekcie. Pozorovaním som zistil, že sa týmto spôsobom spravuje približne každý druhý snímok. Rozlíšenie videa bolo zmenené na 640x480.

Súradnice detekovaného tagu sa prepočítavajú, aby nulový bod $[0,0]$ bol v strede videa. Súradnice tagu tak bezprostredne udávajú pokyn, ktorým smerom sa má kvadrokoptéra natočiť. Detekcia vzdialenosti tagu je relatívne presná, odchýlka je okolo 1 až 2 cm. Maximálna vzdialenosť, do ktorej sa dá na detekciu vzdialenosti spoľahnúť je 2 metre.

Detekcia tagu ponúka aj možnosť vykreslenia nájdeného tagu vo videu. Táto možnosť však zaberá veľa operačnej pamäte a spôsobuje časové oneskorenie videa, preto je táto funkcia vypnutá. Informácie o nájdenom tagu sa zobrazia v informačnom paneli priamo v ovládacom okne.



Obrázok 12 Názorná ukážka práce detekcie tagu

5.2 Spôsoby kompenzácie polohy

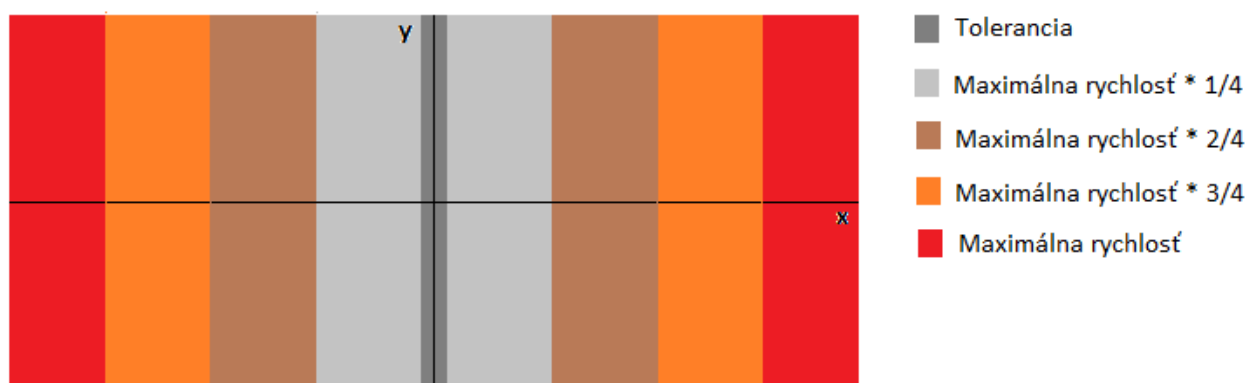
Spôsob kompenzácie polohy je spracovaný pomocou regulátora. Každá dvojica má svoj regulátor (predná – zadná kvadrokoptéra), ktorý má päť rôznych režimov. Synchronizácia pomocou navigačných

dát (rýchlosť, náklon dopredu/dozadu, náklon do strán a nadmorská výška) sa ukazuje ako veľmi nepresná, nakoľko dáta jednotlivých kvadrokoptér sa výrazne líšia. Najspoľahlivejším spôsobom sa zdá byť synchronizácia pomocou detekcie tágu a to konkrétne jeho poloha a vzdialenosť.

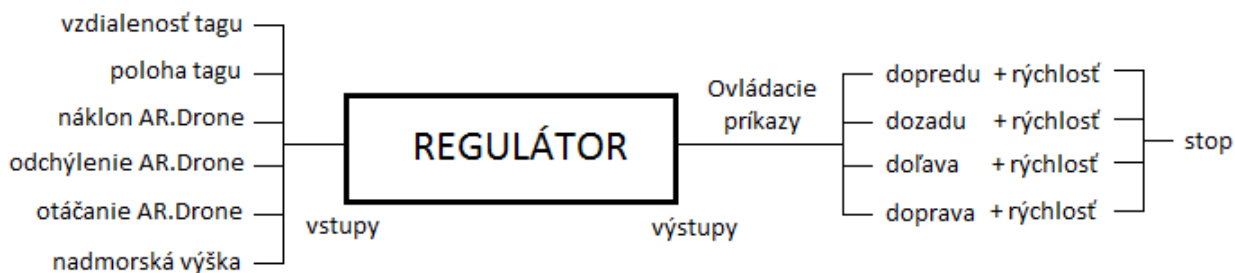
5.2.1 Regulátory

Každá z možností regulátora má svoju vlastnú rýchlosť a toleranciu, ktoré sa dajú dynamicky meniť v nastaveniach. Dôležitým parametrom je letová vzdialenosť, ktorá bolo nastavená na jeden meter vzhľadom k spoľahlivosti detekcie tágu.

Regulátor sa spustí ako náhle sú obe kvadrokoptéry vo vzduchu v stabilizovanej polohe, tzn. že počas vzletu regulátor nepracuje. Rýchlosť kompenzácie polohy sa znásobuje s veľkosťou výchylky. Povedzme, že maximálna rýchlosť je rozdelená do štyroch rýchlostných skupín podľa vzdialenosti od požadovaného stredu. Pri aktivácii nejakého letového príkazu sa jeho rýchlosť a smer zapíše do výstupného loggeru pod kategóriou ovládania (anglicky control).



Obrázok 13 Grafické znázornenie prerozdelenie kompenzačných rýchlostí pre vychýlenie do strán



Obrázok 14 Znáznornenie vstupov regulátora a výstupných príkazov, ktoré regulátor poskytuje

6 Programové prevedenie

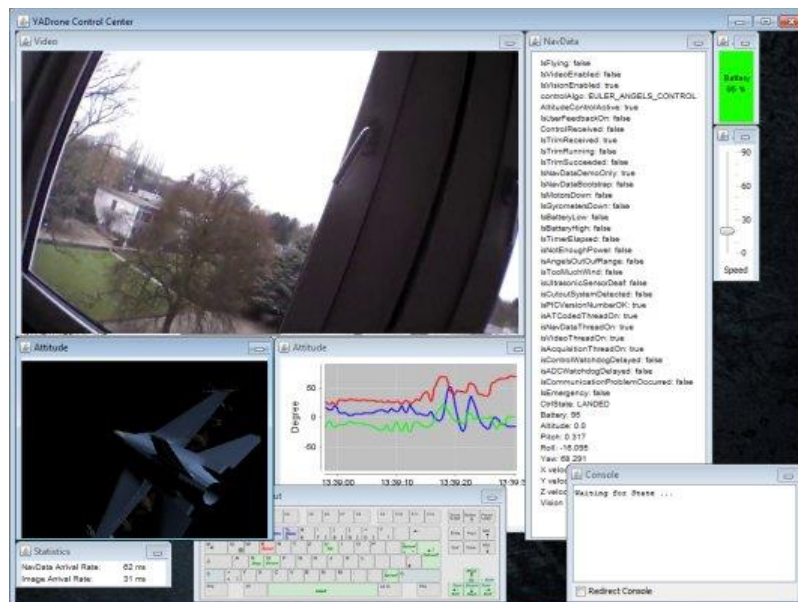
6.1 Nadstavba na YADrone

YADrone je ďalšia voľne šíriteľná open-source štruktúra pre platformu Parrot AR.Drone2. Je založená predovšetkým na ARDroneForP5 projekte a pridáva niektoré opravy chýb a nové funkcie. ARDroneForP5 je štruktúra, ktorá bola navrhnutá pre Parrot AR.Drone 1. YADrone opravuje:

- Video podpora.
- Implementuje príjem navigačných dát z Javadrone projektu pre prístup ku všetkým senzorum.
- Opravený správca príkazov, aby fungovalo zastavovanie platnosti jednotlivých príkazov.

YADrone implementuje len základné ovládacie prvky ako sú let vpred a vzad, let do strán a otáčanie kvadrokoptéry okolo vlastnej osi. Tiež ponúka možnosť zmenu kamery pre video kanál.

YADrone je stále vo vývoji, stále má niektoré známe chyby, ktoré sa týkajú predovšetkým jeho špecifickej grafiky, ktorú môj program, Convoy, nevyužíva. YADrone má zrejme chybu v dekódovaní navigačných dát, pretože aktuálna hodnota výšky je vždy 0. V najnovšej verzii rozširuje použiteľné platformy o android.



Obrázok 15 YADrone hlavné okno – riadiace centrum

Zdroj: <http://vsiis-www.informatik.uni-hamburg.de/projects/yadrone/index.html>

6.1.1 Rozšírenie projektu pre viacero pripojených kvadrokoptér

YADrone vytvára inštanciu triedy ARDrone, ktorá spravuje 3 základné funkcie, respektíve vlákna pre ovládanie kvadrokoptéry – video kanál, príjem navigačných dát a správca príkazov.

V prvom rade bolo nutné vytvoriť list inštancií dostupných kvadrokoptér s prístupom k ich službám. Video manažér pracuje na princípe TCP spojenia na porte 5555. Toto spojenie je bezproblémové a nevyžaduje bezprostredné blokovanie portu, to znamená, že sa jednoducho vytvorí viacero inštancií video manažéra so špecifickými inet adresami.

Správca príkazov využíva spojenie UDP na porte 5556. YADrone využíval *datagramsocket*, ktorý si zarezuje daný port a nikomu inému ho nedovolil používať. Moje riešenie bolo použiť *multicastsocket*, ktorý spravuje jednotlivé *datagramsockets*, ale nerezervuje port natrvalo a tým umožňuje pracovať viacerým správcami príkazov v rovnakom čase.

Riešenie príjmu navigačných dát bolo veľmi komplikované. YADrone používal spoločnú abstraktnú triedu pre správcu príkazov a príjem navigačných dát. Ak som ale použil *multicastsocket*, program prijímal dáta len z poslednej nainicializovanej kvadrokoptéry, ostatné dáta jednoducho ignoroval. Nakoniec bol vytvorený jeden spoločný prijímač pre navigačné dáta všetkých kvadrokoptér, ktorý používa *datagramsocket* a počúva na porte 5554. Po prijatí nejakých dát, prijímač skontroluje od ktorej adresy boli dáta prijaté a na základe tejto informácie, posunie dáta na rozdekódovanie do jednotlivých manažérov navigačných dát.

Je dobre si uvedomiť, že všetky časti boli naprogramované pre ovládanie jednej kvadrokoptéry a preto boli potrebné drobné zásahy do takmer každej triedy pôvodného projektu.

Užívateľovi bolo nutné poskytnúť nové grafické prostredie, ktoré je schopné zobrazovať všetky informácie a spracovávať nové ovládacie prvky (viď 6.2).

- Main control – hlavné ovládacie okno (viď 6.1.2)
- Options – nastavenia (viď 6.1.3)
- Help – pomoc
- Exit – koniec

Programové riešenie využíva java knižnicu swing, ktorá poskytuje potrebné grafické prvky. Pamäťová veľkosť, ktorú zaberá grafické prostredie v pamäti RAM je okolo 700MB. Grafické prostredie nebolo optimalizované, nakoľko moderné stroje by takúto záťaž mali bezproblémovo zvládnuť.

6.1.2 Hlavné ovládacie okno

Hlavné ovládacie okno má na ľavo informačný panel, zobrazujúci informácie o počte pripojených kvadrokoptér, ich rýchlosti, spôsobu riadenia a stavu programu.

V spodnej časti sa nachádza terminálové okno, v ktorom sa zobrazujú aplikačné výpisy.

Hlavnú časť ovládacieho prostredia tvorí jej stred. Je rozdelený do piatich častí, z ktorých je každá určená pre jednu kvadrokoptéru. Každá z častí umožňuje zobraziť plné informácie z navigačných dát, plus zobrazenie základných vlastností. Vedúci skupiny má namiesto informácie o nájdenom tágu nastavenie rýchlosti pohybu pre všetky pripojené kvadrokoptéry.

Okno celej aplikácie je premennej veľkosti. Videá z jednotlivých kvadrokoptér sa dynamicky prekresľujú.

6.1.3 Nastavenia

Nastavenia sú členené podľa funkcionality do štyroch základných skupín:

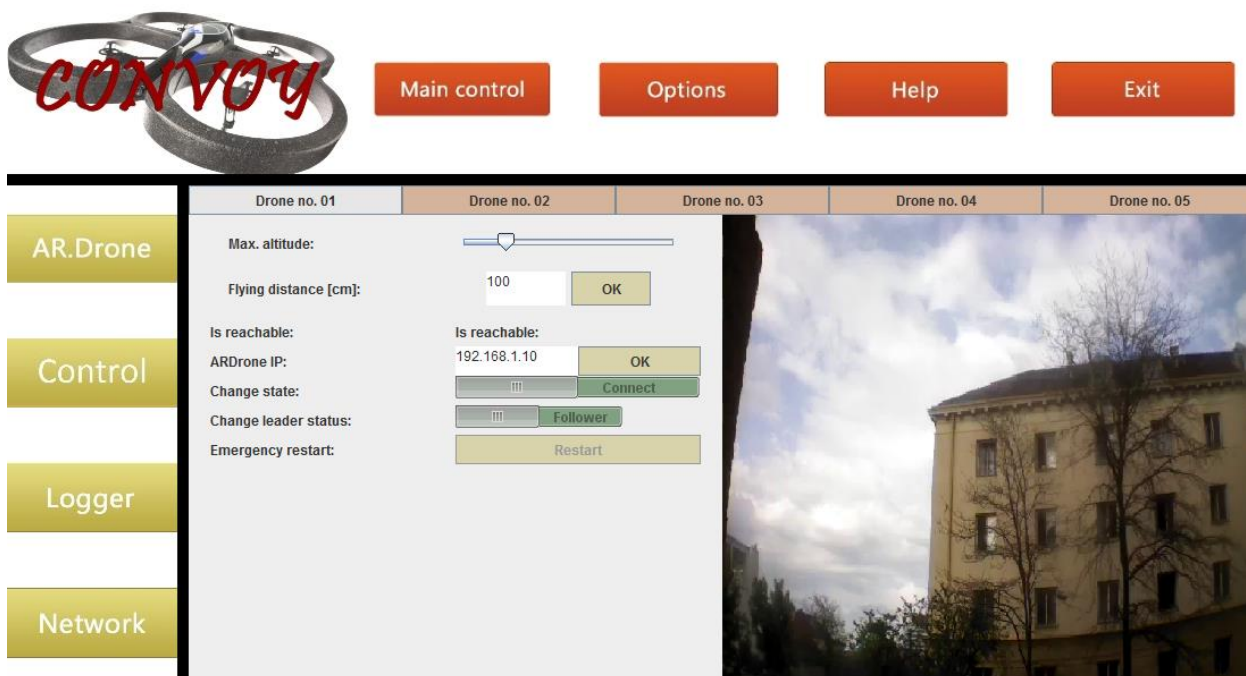
- Network – sieť.
- Control – ovládanie.
- ARDrone
- Logger

Sieť jednoducho zobrazuje stav siete a to kompletne pre všetky sieťové adaptéry. Výstup sa dá prirovnať k terminálovému príkazu *ipconfig -all*.

Ovládanie má dve záložky – klávesnica a regulátor. Klávesnica jednoducho zobrazí klávesovú mapu s preddefinovanými nastaveniami. Záložka regulátor zapína či vypína jednotlivé regulátory, nastavuje im toleranciu a rýchlosť (viď 5.2.1).

ARDrone obsahuje nastavenia, ktoré majú všetky kvadrokoptéry spoločné (napr. výškovú hladinu letu) a zároveň aj nastavenia špecifické pre jednotlivé kvadrokoptéry. V nastavovacom okne sme schopní zmeniť IP adresu (len programovej časti, nie samotného stroja), reštartovať pripojene či zmeniť vedúcu kvadrokoptéru.

Logger poskytuje nastavenia spojené s debugovacími a informačnými výpismi, viď 6.1.3.1.



Obrázok 16 Convoy/Options/AR.Drone – nastavenia ARDrone

6.1.3.1 Debug – Logger

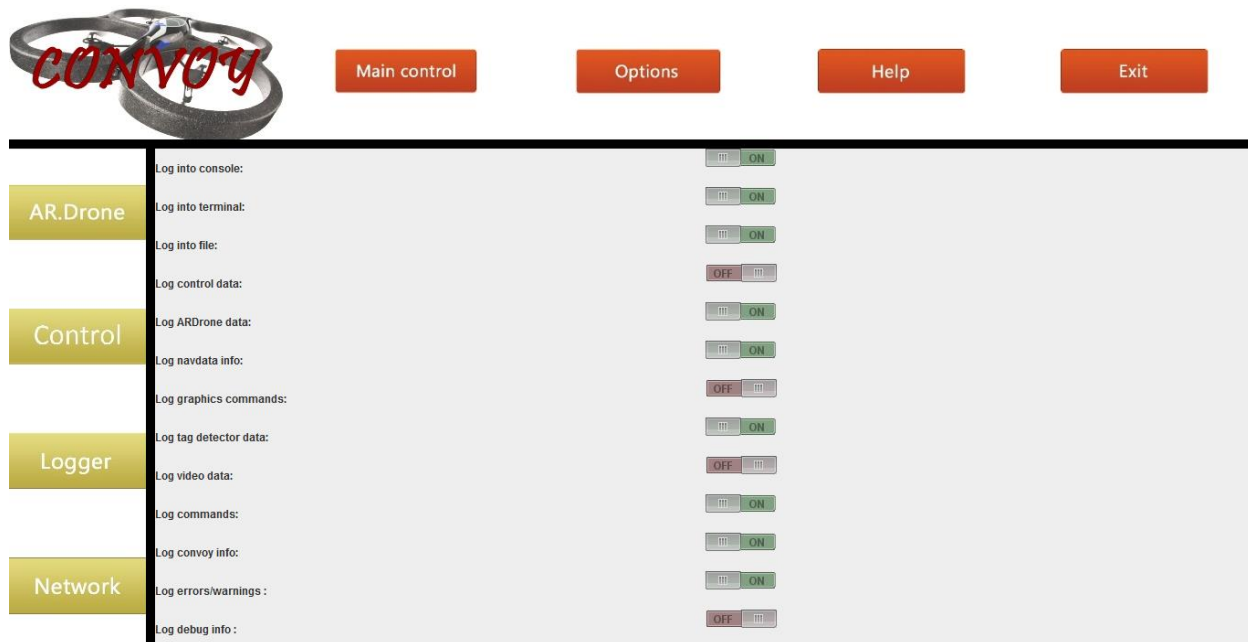
Funkcionalita triedy Logger plne implementuje zapisovanie debugovacích či informačných výpisov. Je možnosť výberu výstupného média:

- Zápis do súboru.
- Zápis do terminálu.
- Zápis do terminálu v aplikácii.

Preťažená metóda *toTerminal* má dva alebo jeden parameter. Pri jednom parametri sa jednoducho zapíše zadaný text. Zaujímavejšia je však skladba s dvoma parametrami, pretože prvý parameter určuje kategóriu zdroja výpisu. Kategórie umožňujú užívateľovi či vývojárovi filtrovať výpisy a zapisovať len vybrané. Všetky nastavenia triedy Logger sa spravujú v programe *Control/Logger*. Kategórie sú:

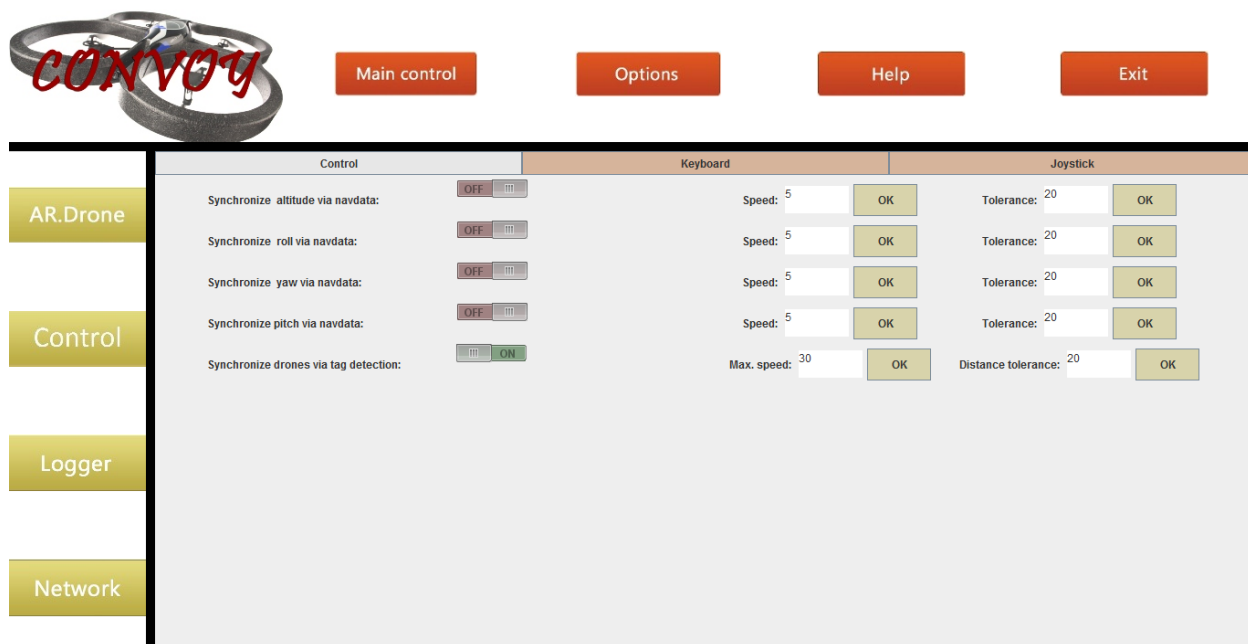
- Control – ovládanie
- Graphics – grafika
- Tag detection – detekcia tagu
- Debug – debugovanie
- Navdata – navigačné data
- Commands - príkazy
- Error – chyby

- ARDrone
- Video
- Convoy



Obrázok 17 Convoy/Options/Logger – nastavenia loggeru

6.2 Ovládanie



Obrázok 18 Convoy/Options/Control – nastavenie ovládania

6.3 Convoy

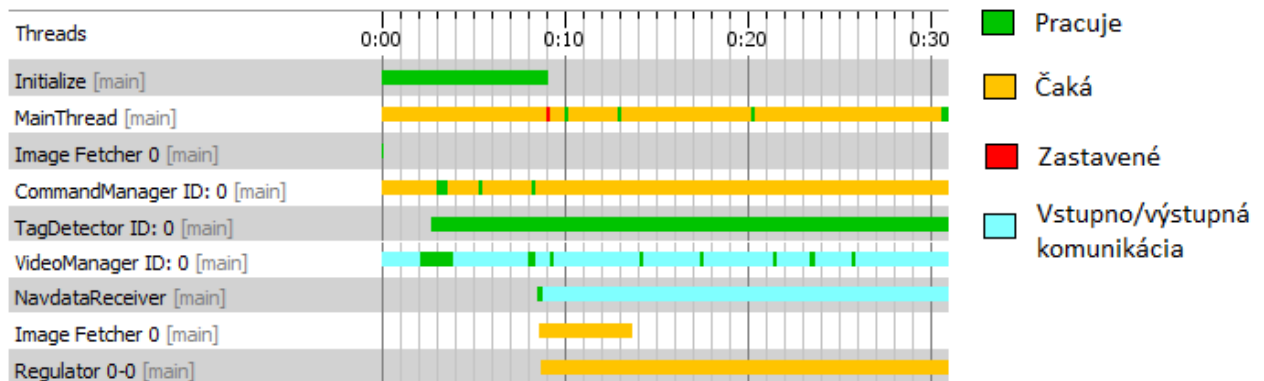
6.3.1 Vlákňová štruktúra

Aplikácia je písaná formou viac vlákňového procesu (anglicky multi-thread). Každé vlákno má svoju jedinečnú funkciu. Niektoré vlákna sa v aplikácii vyskytujú aj niekoľkokrát, s rôznym ID. Je to preto, lebo každé vlákno ovláda funkciu jedného, či dvoch kvadrokoptér.

Základný zoznam vlákien:

- VideoManager – video manažér
- Navdata receiver – prijímač navigačných dát
- CommandManager – manažér odosielania príkazov
- MainThread – grafika
- Regulator X-X – regulátor, kde X-X sú ID kvadrokoptér, ktoré vlákno spravuje
- Tag detector – detekcia tagu
- Initialize – inicializuje program

Grafické znázornenie práce a komunikáciu jednotlivých vlákien môžeme vidieť na obrázku 18. Ak je pripojená len jedna kvadrokoptéra, vlákna TagDetector a Regulátor sú debugovacie, ale znázorňujú skutočnú prácu.



Obrázok 19 Práca jednotlivých vlákien

6.3.2 Načítanie nastavení

Pri zapínaní programu sa z XML súboru načítajú nastavenia, ktoré sa týkajú regulátorov, loggeru a základných informácií a údajov o jednotlivých kvadrokoptéroch.

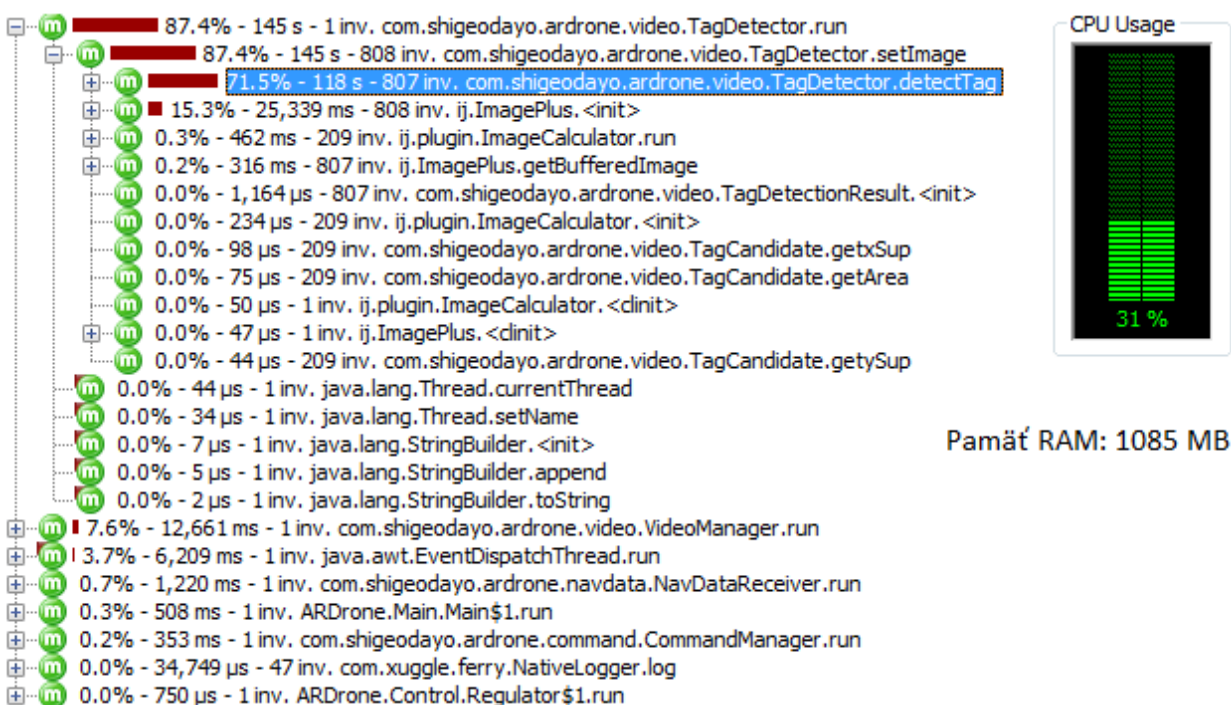
XML súbor obsahuje parameter *pos* ako pozícia, ktorá odpovedá pevne danému ID premennej, ktorej hodnota je sa má načítať.

Ukladanie nastavení sa robí pri ukončovaní programu korektným spôsobom.

6.4 Optimalizácia

Po naimplementovaní všetkých potrebných možností pre program Convoy, som dospel k problému, že využitie procesoru sa konštantne držalo na 100%. Program som pripojil na profiler *JProfiler*, kde som zistil, že nie je nutné mať samostatné vlákno pre všetky druhy regulátorov. Vlákno regulátora bolo zjednotené a v súčasnej podobe má každá dvojica len jedno vlákno. Týmto krokom bolo ušetrené cca. 15% využitia CPU.

Najdôležitejším problémom, ktorý však sčasti naďalej pretrváva je problém so spracovaním videa a detekcie tágu. Najviac času trvalo, kým sa prekreslili pixely nájdeného tágu na čiernu farbu pre jeho vyznačenie, čo spôsobovalo čoraz väčšie a väčšie oneskorenie videa. Preto bolo toto prekresľovanie zakázané.



Obrázok 20 CPU záťaž jednotlivých metód počítaná z celkovej záťaže aplikácie Convoy

7 Experimenty

V priebehu vývoja boli absolvované pravidelné lety s jednou a viacerými kvadrokoptérmi. Najväčším problémom je, že kvadrokoptéry sa po štarte zvyknú vychýliť zo štartovacej pozície a tým sa poruší celistvosť radu.

Ovládacie prvky, ako sú základné pohyby do strán, vpred, vzad či otáčanie okolo vlastnej osi sú prevediteľné a viditeľne synchronizované. Pri lete je potrebné dávať pozor na nastavenie spoločnej rýchlosti kvadrokoptér, ak je príliš nízka, systém sa môže javiť akoby neodpovedal, naopak ak je príliš vysoká, systém je veľmi prchký a pocitovo labilný. Pri lietaní v exteriéroch je problémom vietor, aj pokiaľ sa jedná len o slabý vánok, pri štarte môže spôsobiť výrazné vychýlenie. Jediným dôležitým problémom teda ostáva udržať formáciu.

Detekcia tágu sa javila ako spoľahlivá za prijateľných aj zhoršených podmienkach, čo znamená, že som to otestoval pri zamračenom aj slnečnom počasí v interiéroch pri dennom svetle aj umelom osvetlení. Samozrejme si detekcia tágu neporadí v hmle či tme.



Obrázok 21 Nasledovanie detekovaného tágu

7.1 Manuálne ovládanie formácie

Manuálne ovládanie formácie poskytuje viditeľne synchronizovaný pohyb všetkých pripojených kvadrokoptér. Kvadrokoptéry dokážu v rovnakom čase zmeniť polohu, výšku či rýchlosť. Pri testovaní v priestorovo obmedzených miestnostiach a pri nerovnostiach podlahy (prelet nad podlahou

a následne nad napr. skriňou alebo posteľou) má každá kvadrokoptéra problém ostať stáť na mieste formácia sa začína rozpadáť.

Pri synchronizovanom lete treba dávať pozor na náhlu zmenu smeru letu alebo na príliš krátko trvajúce príkazy. Ak jedna z týchto situácií nastane, niektorí nasledovníci nie sú schopní zareagovať a buď ostanú stáť na mieste, alebo vzdialenosť, ktorú preletia je rozdielna od vzdialenosti vedúceho skupiny.

Synchronizácia formácie pomocou regulátorov, najmä detekcie tágu v momentálnej implementácii nie je úplne schopná zabezpečiť stabilitu systému, čo znamená, že pokiaľ sa kvadrokoptéry od seba vzdialia na väčšiu vzdialenosť, detekcia tágu sa stane nespoľahlivou a formácia sa rozpadne. Avšak preposielanie akčného zásahu zabezpečuje stály synchronizovaný pohyb.

Pozorovanie stringovej nestability je vzhľadom k aktuálnej funkčnosti programu veľmi náročná, avšak typické prejavy tohto problému sú potlačené funkcionalitou preposielania akčného zásahu. Je možné pozorovať istú nestabilitu, respektíve rozdiel medzi dĺžkou kroku vedúceho skupiny a jeho nasledovníkov. Všetci nasledovníci sa vždy pohybujú ako celok, pretože sú im posielané rovnaké príkazy v rovnaký čas. Regulátory spôsobujú kmitanie nasledovníkov okolo rovnovážnej polohy. Tento problém sa však čiastočne podarilo odstrániť zavedením tolerancie.



Obrázok 22 Let formácie štyroch kvadrokoptér

8 Záver

V práci som plne implementoval grafické prostredie so sofistikovaným debugovacím systémom. Otestoval som let štyroch kvadrokoptér (piata kvadrokoptéra mala problém so softwarom) a môžem konštatovať, že synchronizácia letu sa podarila, ale problém udržať formáciu sa stále objavuje, pretože synchronizácia pomocou detekcie tágu nie je dostačujúca ani stopercentná a sama o sebe môže vyvolať rozkmitanie systému. Implementoval som regulátor, ktorý na základe navigačných dát a detekcie tagu stabilizuje systém. Je nutné však poznamenať, že navigačné dáta jednotlivých kvadrokoptér sa výrazne líšia (napríklad rôzny náklon) a preto majú tieto časti nastavené veľké tolerancie alebo sú úplne vypnuté. Celý program bol následne optimalizovaný, avšak rýchlosť odozvy silno závisí na sile signálu pripojenia.

Pre odstránenie tohto problému som vytvoril návod, ako importovať ďalšie senzory k palubnému počítaču kvadrokoptéry AR.Drone pomocou dosky Arduino, ktorá bude použitá v ďalšom pokračovaní tejto témy.

Podarilo sa mi dokázať, že komerčná, pomerne lacná platforma dokáže tvoriť formáciu a vykonávať synchronizovaný pohyb.

Neoddeliteľnou časťou tejto práce je prehľad súčasných výrobcov, či akademikov, ktorí pracujú s témou kvadrokoptér.

Zoznam použitej literatúry

- [1] Hashimoto, S. ARDroneForP5: AR.Drone library for Processing. kougaku-navi. http://kougaku-navi.net/ARDroneForP5/index_en.html (citované Máj 01, 2013).
- [2] Quadrotor. wikipedia. <http://en.wikipedia.org/wiki/Quadrotor> (citované Máj 01, 2013).
- [3] Sharp, R. droneduino, 2013. Github. <https://gist.github.com/maxogden/4152815> (citované Máj 01, 2013).
- [4] Parrot AR.Drone. wikipedia. http://en.wikipedia.org/wiki/Parrot_AR.Drone (citované Máj 01, 2013).
- [5] YADrone, 2013. informatik.uni-hamburg. <http://vsis-www.informatik.uni-hamburg.de/projects/yadrone/index.html> (citované Máj 01, 2013).
- [6] YADrone, Yet Another Drone framework for AR.Drone 2 in Java, 2012. ardrone. <https://projects.ardrone.org/boards/1/topics/show/5259> (citované Máj 01, 2013).
- [7] Products, 2013. Draganfly. <http://www.draganfly.com/industrial/products.php> (citované Máj 01, 2013).
- [8] Products, 2013. Microdrones. <http://www.microdrones.com/products/products.php> (citované Máj 01, 2013).
- [9] Raffaello, D. Flying machine area, 2013. Raffaello. <http://raffaello.name/dynamic-works/flying-machine-arena> (citované Máj 01, 2013).
- [10] Z. Hurák a M. Šebek, "2D polynomial approach to stability of platoons of vehicles," in Proceedings of the 2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems, vol. 2, Centre de Congr'es de L'Impérial Palace, Annecy, France, 2010.
- [11] Franchi, A.; Secchi, C.; Ryll, M.; H.Bulthoff, H.; Robuffo Giordiano, P. Shared control. IEEE Robotics & Robotics 2012, 19 (3), 57–69.
- [12] Dvořák, J. Micro Quadrotor: Design, Modelling, Identification and Control. Diplomová práce, České vysoké učení technické, 2011.
- [13] Arnošt, P. Návrh řídicího modulu UAV robotu. Diplomová práce, Vysoké učení technické v Brně, 2012.
- [14] ArduCopter, 2013. code.google.com. <https://code.google.com/p/arducopter/> (citované Máj 22, 2013).
- [15] Tomlin, C.; Huang, H.; Vitus, M.; et al. Starmac, 2013. <http://hybrid.eecs.berkeley.edu/>. <http://hybrid.eecs.berkeley.edu/starmac/> (citované Máj 22, 2013).

Príloha

Obsah priloženého CD:

- Obrázky z programu a experimentov
- Video nasledovania tagu
- Zdrojový kód aplikácie
- Aplikácia
- Bakalárska práca vo formáte PDF