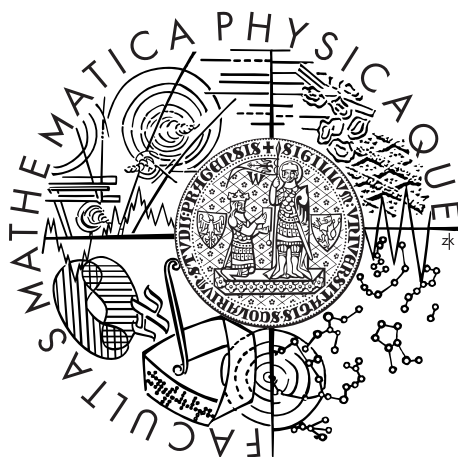


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Jan Škoda

Stabilizace létajícího drona v dynamickém prostředí

Katedra teoretické informatiky a matematické logiky

Vedoucí bakalářské práce: prof. RNDr. Roman Barták, Ph.D.

Studijní program: Informatika

Studijní obor: Obecná informatika

Praha 2014

Děkuji svému vedoucímu prof. RNDr. Romanu Bartákovi, Ph.D. za uvedení do světa robotiky a cenné připomínky při vedení bakalářské práce. Dále děkuji své rodině za podporu při studiu. V neposlední řadě děkuji svým kamarádům Filipovi Štědronskému a Bc. Jakubovi Suchému za připomínkování srozumitelnosti textu práce.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Název práce: Stabilizace létajícího drona v dynamickém prostředí

Autor: Jan Škoda

Katedra: Katedra teoretické informatiky a matematické logiky

Vedoucí bakalářské práce: prof. RNDr. Roman Barták, Ph.D.

Abstrakt: Práce se zabývá implementací a popisem systému vzdáleně řídicího kvadrioptéru, který je schopen ji dlouhodobě udržet na místě navzdory vlivům vnějšího prostředí i nespolehlivosti senzorů. Toho je dosaženo využitím obrazových dat z kamery a informací ze senzorů bez jakékoli předchozí znalosti prostředí a bez navigačního signálu. Jádrem řešení je lokalizační a mapovací systém paralelně sledující rozpoznatelné body v přijímaném obrazu a udržující mapu těchto bodů v trojrozměrném prostoru. Jelikož systém nemůže přesně určit ani polohu bodů, které vidí na kameře, ani polohu své kamery, optimalizuje v reálném čase a hledá hodnoty všech těchto neznámých tak, aby dosáhl minimální celkové chyby. Výstup této vizuální lokalizace je pak Kalmanovým filtrem zkombinován s nevizuálními navigačními daty drona a na základě modelu dynamiky drona ve filtru probíhá predikce pohybu a na jeho základě reaktivní navigace.

Klíčová slova: slam, lokalizace, mapování, ar.drone, kalmanův filtr

Title: Stabilisation of a flying drone in dynamic environments

Author: Jan Škoda

Department: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: prof. RNDr. Roman Barták, Ph.D.

Abstract:

This work describes implementation of a system remotely controlling quadricopter, that is able to stabilize and hold it still regardless of external influences and inaccuracy of sensors. That was achieved by utilizing visual data from camera and information from sensors without any preceding knowledge of the environment and navigational signal. The core of the approach is a localization and mapping system tracking recognizable points in video and maintaining a 3D map of those points in parallel. As the system can't measure the location of points seen through the camera and even the camera itself, it optimizes in real time and estimates the values of all those variables, in order to minimize the total error. Output of visual localization is afterwards combined using the Kalman filter with non-visual navigation data. The filter then predicts future location using drone's dynamics model. Resulting location is then used for reactive navigation.

Keywords: slam, localisation, mapping, ar.drone, kalman filter

Obsah

Úvod	3
Vývoj v oblasti robotiky	3
Cíl práce	3
Struktura práce	4
1 Platforma AR. Drone 2	5
1.1 Zařízení	5
1.2 Senzory drona	5
1.3 Rozhraní	5
1.3.1 Data přijímaná ze senzorů	6
1.3.2 Data přijímaná z kamer	6
1.3.3 Řízení drona	7
1.4 Problémy	8
2 Rozbor řešeného problému	9
2.1 Zavedení pojmů	9
2.2 Stabilizace drona	9
2.2.1 Minimalizace zrychlení	10
2.2.2 Minimalizace rychlosti	10
2.2.3 Minimalizace vzdálenosti	10
2.3 Vizualizace lokalizace	11
2.4 Model řízení	12
2.5 Fúze a filtrování senzorů	13
2.5.1 Alternativy	13
2.5.2 Letový a sensorický model	14
3 Lokalizace a mapování	15
3.1 Problém (vizualizace) SLAM	15
3.2 Průběh lokalizace a mapování	16
3.3 Zvolená metoda	16
3.3.1 Spojení PTAM a EKF	17
3.4 Sledování landmarků	18
3.5 Lokalizace	19
3.5.1 Projekce bodů mapy do roviny přijatého snímku videa	19
3.5.2 Určení polohy drona na dle pozic pozorovaných landmarků	19
3.5.3 Detekce ztráty lokalizace a její důsledky	20
3.6 Mapování	20
3.6.1 Inicializace	20
3.6.2 Přidání keyframu	21
4 Fúze a filtrace senzorů	22
4.1 Základní pojmy a schopnosti filtru	22
4.2 Fungování rozšířeného Kalmanova filtru	23
4.2.1 Stav a měření	24
4.2.2 Údaje o procesu ovlivňující chování EKF	25

4.2.3	Predikční a korekční krok	25
4.2.4	Důsledky predikce a korekce	26
4.3	Chování filtru při ztrátě vizuální lokalizace	27
4.4	Pravděpodobnostní modely	27
4.4.1	Letový model	28
4.4.2	Senzorický model	30
4.5	Střednědobá predikce	31
5	Řízení drona	32
5.1	Rozhraní	32
5.2	Kontrolér	32
5.3	Řízení	33
5.3.1	Vliv nepřesnosti měřítka mapy na řízení	33
6	Uživatelské rozhraní	34
6.1	Popis	34
6.1.1	Interpretace okna č. 2	34
6.2	Řízení	35
6.3	Použití programu z CD	36
6.3.1	Konfigurace parametrů systému	36
7	Verifikace	37
7.1	Experiment	37
7.2	Výsledky experimentu	38
7.2.1	Kolejní pokoj	38
7.2.2	Fasáda domu	39
	Závěr	40
	Seznam použité literatury	43
	Seznam použitých zkratek	44
	Přílohy	45

Úvod

Vývoj v oblasti robotiky

V nedávné době se značná část zájmu mobilní robotiky oklonila od snahy o konstrukci a řízení humanoidních robotů schopných provádět běžné lidské činnosti k létajícím dronům. Tato zařízení jsou již v praxi schopná provádět širokou paletu specializovaných činností od roznášení balíků po logistickém skladu po přesné zaměření a eliminaci teroristů. Výzkum v oblasti skutečně autonomních dronů je ale stále v začátcích. Armádní droni¹ jsou z velké části řízeni lidskou obsluhou, průmysloví jsou často navigováni pomocí externího signálu – uvnitř pomocí radiomajáků, venku pomocí GPS. Tento přístup ale není příliš robustní a naráží na problémy například při častém přerušení GPS signálu v zástavbě.

Aby byl dron skutečně autonomní jednotkou odolnou ztrátě řídicího nebo lokalizačního spojení, je třeba umět co nejspolehlivěji zpracovávat data ze senzorů, které má k dispozici. Nejvýznamnějším a zároveň nejdostupnějším takovým zdrojem dat je kamera. Oproti dalším senzorům, poskytujícím obdobné množství informací z okolí robota, jako například laserové rangefindery², má kamera několik výhod. Předně je to nízká pořizovací cena. Dále nemá omezený dosah, je pasivním senzorem, díky čemuž například odpadá možnost vzájemného rušení několika roboty a ve vojenském prostředí nehrozí detekovatelnost senzoru.

Zpracování vizuálních dat ale přináší i několik podstatných problémů, díky kterým profesionální řešení dávají přednost rangefinderům. Velké množství informace, které kamera řídicímu systému dodává vyžaduje velký výpočetní výkon. Problémy jsou i s kvalitou obrazu při pohybu robota a za horšího osvětlení. To naštěstí většinou uspokojivě vyřeší kvalitní kamera s dostatečně velkým a světelným objektivem. Největším problémem je ale samotné zpracování obrazu a rozpoznání důležitých objektů, které pro mnoho aplikací ještě není dostatečně robustní. Příkladem může být projekt autonomního vozidla společnosti Google, které využívá jako bezpečný zdroj informací o dění před vozem lidar.

Cíl práce

Cílem implementace je systém vzdáleně řídicí kvadrikoptéru AR.Drone 2 (viz kap. 1) přes Wi-Fi, který je schopen ji dlouhodobě udržet na místě navzdory vlivům vnějšího prostředí i nespolehlivosti senzorů. Dron je tedy možné například ručně vychýlit nebo mu dočasně zakrýt kameru. K tomu má systém k dispozici obrazová data z kamery, informace z IMU (akcelerometr a gyroskop), tedy magnetický azimut a informaci o náklonu ve 2 osách, a odhad rychlosti drona podle všech 3 os. Systém naopak nemá žádné předchozí znalosti prostředí ani vnější navigační

¹Slovo *dron* skloňuji stejně jako *robot* podle mužského životného vzoru stejně tak, jak jej začal používat K. Čapek ve svém dramatu *R.U.R.*

²Rangefinder je senzor poskytující informace o vzdálenostech předmětů kolem senzoru

signál. Nemůže předpokládat žádné navigační pomůcky a případnou mapu okolního prostředí si musí budovat sám. Před letem je však možné zkalibrovat nebo iniciovat senzory.

Řídící systém se kromě nespolehlivosti senzorů musí vypořádat i s prodlevou měření a řízení. Díky ní by totiž bez vhodné kompenzace docházelo k velkým překmitům cílové polohy nebo dokonce k soustavnému kmitání.

Struktura práce

Práce obsahuje dvě hlavní části, implementaci a popis fungování stabilizace kvadrikopty Parrot AR.Drone 2. Tento popis je zaměřen hlavně na principy lokalizace, mapování, zpracování měření ze senzorů a řízení robota. V případě, že implementujeme tyto principy nějakým nepřímým nebo zajímavým způsobem, uvádíme v popisu i to. Částem, které jsem neimplementoval vlastnoručně, ale využil abstrakce v podobě nějaké existující knihovny, bude věnováno méně pozornosti a místo jejich podrobného popisu uvádíme jen shrnutí vnitřního fungování a odkaz na podrobnosti od tvůrců knihovny.

1. Platforma AR. Drone 2

1.1 Zařízení

Parrot AR.Drone 2 (dále jen dron) je vzdáleně ovládaná kvadrikoptéra (vrtulník se čtyřmi rotory) vyráběná francouzskou firmou Parrot. Dron je určen k ovládání mobilními nebo tabletovými operačními systémy jako iOS nebo Android přes Wi-Fi spojení. Mezi typické aplikace patří hry s rozšířenou realitou¹ nebo záznam videa na připojenou USB flash paměť. Konstrukce drona je uzpůsobena pro let v interiérech i exteriérech a pro tyto dvě použití jsou dokonce s dronem dodávány dva druhy ochranných krytů. Masivnější z nich je určen do vnitřního prostředí, kde více hrozí náraz, menší do venkovního prostředí, kde je užitečná jeho menší plocha, díky které bude na drona méně působit eventuální vítr.

První verze byla uvedena na trh v roce 2010 v Las Vegas, druhá verze v roce 2012 tamtéž. Vylepšeny byly senzory i funkcionality drona. Za zmínku stojí vyšší kvalita kamery, senzor tlaku vzduchu a magnetický kompas.[16]

1.2 Senzory drona

AR.Dron 2 je oproti svým levnějším konkurentům specifický mj. velkou nabídkou senzorů. Díky tomu se dobře hodí jako platforma pro robotický vývoj. Výrobce[5] uvádí následující specifikaci senzorů:

- Přední HD kamera – nahrává obraz před dronem. Šířka objektivu je 92° diagonálně, kvalita 720p, 30 FPS. Základní profil kódování H264.
- Vertikální kamera s rozlišením QVGA a 60 FPS
- 3-osý gyroskop s přesností 2000°/s
- 3-osý akcelerometr s maximální chybou 50mg
- 3-osý magnetický kompas
- ultrazvukový výškoměr
- senzor tlaku vzduchu s přesností $\pm 10Pa$ (10Pa odpovídá cca $\pm 80cm$ nadmořské výšky)

1.3 Rozhraní

Dron obsahuje operační systém Linux, který obsluhuje senzory a rotory drona a generuje Wi-Fi přístupový bod. Po připojení ovládacího zařízení na tento přístupový bod poskytne DHCP server drona ovládacímu zařízení IP adresu a to

¹„Rozšířená realita (někdy také augmentovaná realita) je označení používané pro reálný obraz světa doplněný počítačem vytvořenými objekty. Jinak řečeno jde o zobrazení reality (např. budovy nasnímané fotoaparátem v mobilním telefonu) a následné přidání digitálních prvků (třeba informací o daném objektu).“ [18]

se můžeme k dronovi připojit pomocí kombinace TCP a UDP spojení. Ovládací programy mohou buď implementovat celý síťový ovládací protokol a připojit se k dronovi přímo, nebo využít některou z knihoven zapouzdřujících tuto síťovou komunikaci. Výrobce drona, Parrot, dodává pro ovládání drona vlastní knihovnu, *AR Drone SDK* (dále jen SDK), která nabízí rozhraní pro ovládání drona pro jazyk C. Toto oficiálně podporované rozhraní používá i implementace této práce. Kromě těchto možností existuje i mnoho více či méně kvalitních nástrojů třetích stran. [7]

1.3.1 Data přijímaná ze senzorů

SDK dodává programu tzv. *navigační data* (označována též jako *navdata*), která obsahují předzpracovaná data z ostatních senzorů. Toto předzpracování ani přesnost jeho výstupu bohužel není výrobcem zdokumentováno, což komplikuje pravděpodobnostní interpretaci přijímaných dat (4.4.2).

Na základě průzkumu obsahu a struktury navigačních dat a porovnání se schopnostmi specifikovaných senzorů se zdá, že firmware drona odhaduje naklonění drona oproti horizontální rovině na základě směru gravitačního zrychlení z akcelerometru a okamžité rychlosti naklánění z gyroskopu. Na základě obrazu z vertikální kamery směřující pod drona pak firmware pomocí metody sledování optického toku[9] odhaduje rychlost.

Celkem přijímaná navigační data obsahují tyto údaje:

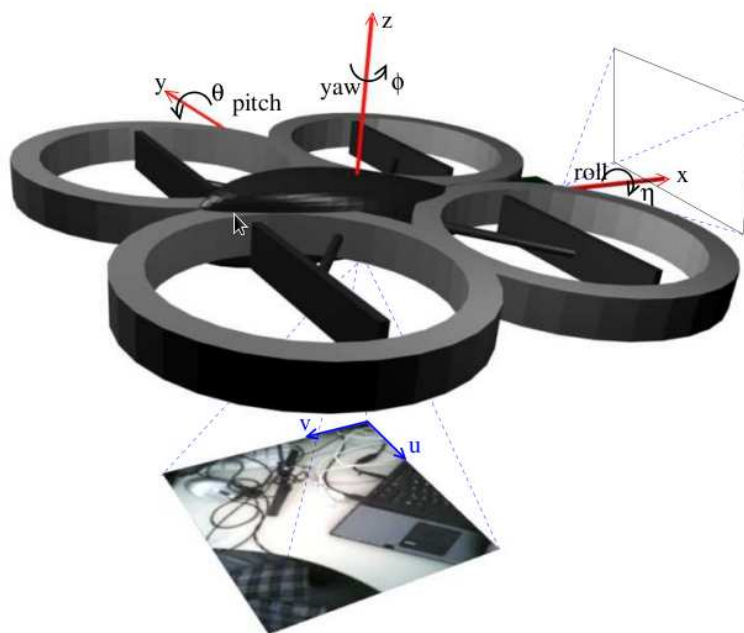
- Úhel náklonu (roll) (obr. 1.1)
- Úhel sklonu (pitch) (obr. 1.1)
- Azimut podle magnetického kompasu (yaw)
- Rychlost ve směru dopředu, doleva a vzhůru (x,y,z).
- Výšku

Poslední 2 uvedené údaje, rychlosti a výšku, dron odesílá v navigačních datech jen za letu. Kromě těchto přijímaných dat je možné přímočaře odhadnout i zrychlení. To je dané náklonem drona, který díky tahu svých vrtulí zrychluje ve směru svého náklonu. Přesný výpočet je uveden v části 4.4.1.

1.3.2 Data přijímaná z kamer

Dron svými dvěma kamerami zaznamenává obraz před a pod sebou. Nedokáže však odesílat oba tyto videozáznamy v plné kvalitě zároveň. Na výběr jsou tyto 3 režimy přenášeného obrazu:

- Obraz z přední kamery
- Obraz z vertikální kamery směřující pod drona



Obrázek 1.1: Rychlosti a úhly drona oproti horizontální rovině [3]

- Přenos typu obraz v obraze, kdy je obraz z vertikální kamery umístěn do rohu obrazu z přední kamery. Nahrazená část obrazu přední kamery pak tedy chybí

Dron je schopný přenášet obraz v rozlišení 640x360 nebo 1280x720 o 15 nebo 30 FPS (snímky za vteřinu). Video je komprimované kodekem H.264 nebo MPEG4.2 a kvalita videa je omezena na 4Mb za sekundu.

1.3.3 Řízení drona

Dron se pohybuje výhradně regulací rychlostí svých 4 rotorů (tyto rotory budeme coby prostředku pohybu robota v práci občas označovat obecným termínem *aktuátory*). Jejich rychlost obsluhuje software drona na základě řídicích příkazů přijatých přes Wi-Fi. Za letu se dron řídí jediným řídicím příkazem pro pohyb, který nastavuje tyto hodnoty:

- Úhel náklonu (roll) (obr. 1.1)
- Úhel sklonu (pitch) (obr. 1.1)
- Rotační rychlost
- Vertikální rychlost

Úhly jsou zadávány desetinným číslem z rozsahu $[-1; 1]$. Úhly odpovídající hraničním intervalu jsou nastaveny v konfiguraci drona[7], mezi těmito body je stupnice

lineární. Rychlosti jsou předávány po řadě v deg/s a mm/s .

Kromě tohoto příkazu jsou k dispozici příkazy pro vzlet, přistání a přepnutí do emergency módu. Při vzletu dron vystoupá do výšky nastavené v konfiguraci[7], přičemž už při stoupaní přijímá a provádí řídicí příkazy pro pohyb. Při přistání dron klesá, dokud ultrazvukový výškoměr nedetekuje blízkost podložky. Pak dron vypne rotory a dopadne na podložku, aby se při přistávání nevznášel na vzduchových vírech vytvořených vzduchem proudícím mezi rotory a podložkou.

1.4 Problémy

AR.Drone má také několik problémů a omezení. Jeho baterie vydrží maximálně 30 minut letu v závislosti na jejím stáří a používání. Pokud je baterie nabita pod 20%, drone vůbec nevzlétne. Pokud drone narazí takovým způsobem, že se byt jen na chvíli zpomalí některá z jeho vrtulí (například nárazem do pružné vnější kostry), firmware takový stav zdetekuje a přepne se do *emergency* módu. To znamená, že okamžitě vypne všechny vrtule a padá k zemi.

Ačkoli má drone poměrně kvalitní 720p přední kameru, před přenosem po Wi-Fi její obraz ztrátově komprimuje na maximálně 4Mb přenesených dat za sekundu. Výsledkem toho je, že při přenosu mnoha rozdílných rámců obrazu není obraz ostrý. To se bohužel děje prakticky při všech manévrech drona, protože každá rotace podle libovolné osy znamená změnu směru kamery a tato změna zase jiný obraz. Systém pro zpracování obrazu se tedy musí dokázat vyrovnat s častou ztrátou dat a ideálně dokázat získávat informace i z méně ostrých snímků.

Dalším problémem je zpoždění sensorických dat i řídicích příkazů. Díky kompresi obrazu, zpoždění Wi-Fi a omezené schopnosti akcelerace motorů dosahuje zpoždění mezi získáním obrazu a vykonáním řídicího příkazu průměrně 250ms[1]! To by bez adekvátní kompenzace ústilo k velkému překmitávání stabilizační pozice (ve které se systém snaží drona udržet), neboť systém by ve chvíli, kdy už dron cílovou pozici proletěl, stále navigoval dál stejným směrem.

2. Rozbor řešeného problému

2.1 Zavedení pojmů

V této kapitole budeme používat některé složitější pojmy z oblasti robotiky a lokalizace robota, které je vhodné zavést předem. Prvně je to klasifikace prostředků lokalizace na relativní a absolutní:

„ Pomocí prostředků *relativní lokalizace* lze odhadnout relativní změnu polohy robota, zpravidla tedy jeho posunutí a rotaci v prostoru vůči jeho předcházející poloze. Celková změna polohy robota vztažená k jeho počáteční poloze se určuje skládáním jednotlivých dílčích změn polohy. Na rozdíl od prostředků absolutní lokalizace neumožňují prostředky relativní lokalizace zjistit absolutní polohu robota v prostředí; je-li ale počáteční pozice robota v prostředí známá, je možné tyto prostředky krátkodobě využívat i k lokalizaci absolutní. Krátkodobě z toho důvodu, že kvůli inkrementálnímu charakteru výpočtu pozice řetězením jednotlivých dílčích změn dochází k akumulaci chyb, takže se celková chyba lokalizace neustále zvětšuje. “ [6]

„ Prostředky *absolutní lokalizace* umožňují zjistit nebo odhadnout absolutní polohu robota v prostoru, bez ohledu na události a stavy, které dosažení této polohy předcházely. “ [6] Díky tomu absolutní lokalizace netrpí akumulovanou chybou.

Robot se ve svém okolí může orientovat pomocí *landmarků* – bodů v okolí, jejichž pozici jsme schopni sledovat i znovu rozpoznat po tom, co se tyto body ztratí z dosahu senzoru.

Stavem robota obecně rozumíme celkovou informaci o jeho pozici, rychlosti, případně třeba i úrovni nabití jeho baterie. Přesněji dále v práci zavádíme stavový vektor, který je definován v kapitole 4.

2.2 Stabilizace drona

Pojmem stabilizace drona chápeme snahu řídicího systému zabraňovat pohybu drona v prostoru a vracet se do původní pozice, pokud je z ní dron vychýlen. Tuto pozici budeme nazývat *stabilizační pozice*.

Představíme si 3 přístupy, které vedou k různě robustní stabilizaci létajícího drona. Postupně se budeme snažit minimalizovat zrychlení, rychlost a vzdálenost od stabilizační pozice.

Pro lepší představu o vztahu těchto přístupů je vhodné dodat, že v ideálním světě s přesnými senzory i aktuátory a bez vnějších vlivů (jako např. vítr) by stačil už první přístup. Nulové zrychlení by zajistilo nulovou rychlost a ta nulovou uraženou vzdálenost.

2.2.1 Minimalizace zrychlení

Nejjednodušší přístup k problému stabilizace je pomocí údajů z gyroskopu držet tělo kvadrikopty v horizontální rovině, čímž se minimalizuje horizontální zrychlení. Pak už stačí udržovat za pomoci výškoměru nulovou vertikální rychlost. Není překvapením, že tento přístup absolutně nefunguje. Senzory ani aktuátory nejsou zdaleka dostatečně přesné na to, aby drona udržely v dokonalé horizontální rovině. Kromě toho na drona neustále působí vnější vlivy jako například přirozené proudění vzduchu a v blízkosti podlahy nebo jiných předmětů i poměrně silné vzdušné víry.

Dron tedy nejen že nebude stabilizován na místě, ale pokud díky nepřesnosti stabilizace nebo vlivu vnějšího prostředí (např. větru) získá rychlost, nebude schopen ji snižovat, zastavit a opět se stabilizovat. Jediné, co tento přístup zajišťuje, je udržování nulového zrychlení.

2.2.2 Minimalizace rychlosti

Velkou vadou minulého řešení je neschopnost snižovat rychlost. Dron ale nemá žádný senzor přímo snímající rychlost¹, takže je možné ji jen nepřímou odhadovat.

První možností je využít přímo odhad rychlosti drona určený pomocí vertikální kamery (1.3.1). Tato metoda však selhává, pokud nejsou pod dronem (např. na podlaze) rozpoznatelné vizuální body. Druhá možnost je využít data z akcelerometru a rychlost určit integrováním vektoru zrychlení.

$$v = \int_0^T a(t) dt. \quad (2.1)$$

Rovnice 2.1: Vztah rychlosti a zrychlení podle Newtonovské fyziky.

Ani tento přístup však není přesný, neboť nepřesná měření zrychlení způsobují akumulovanou chybu rychlosti. Díky ní dron může naopak začít zrychlovat ze stabilizační pozice, pokud vypočítá nenulovou rychlost, ačkoli stojí.

2.2.3 Minimalizace vzdálenosti

Všechna předchozí řešení využívají pouze metody relativní lokalizace [6], trpící akumulací lokalizační chyby. Robustní řešení by ale vyžadovalo možnost lokali-

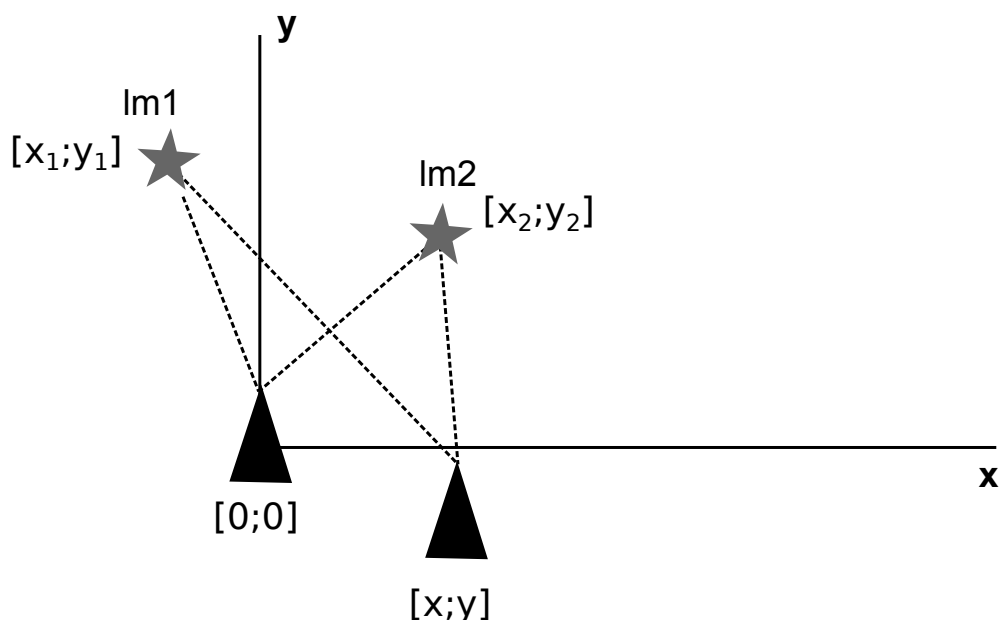
¹A z fyzikálního hlediska je měření rychlosti komplikované. Je totiž třeba ji vždy měřit relativně k nějakému objektu. Buďto k okolnímu vzduchu, který se ale také pohybuje a v malých rychlostech to nelze zanedbat, nebo k nehybnému objektu na zemi, který je ale třeba vybrat a přesně lokalizovat.

zace drona vůči stabilizační pozici s chybou nezávislou na čase, tedy možnost absolutní lokalizace. Taková lokalizace by byla odolná vůči krátkodobé nepřesnosti senzorů, jejich výpadkům nebo nárazům drona do překážek. Pak bychom mohli minimalizovat vzdálenost drona od stabilizační pozice tak, že bychom ji jej do ní přímo navigovali.

Možnosti absolutní lokalizace drona jsou ale poměrně omezené. Jediným senzorem schopným měřit pozici nezatíženou akumulovanou chybou je výškoměr. Jeho problémem ovšem je, že měří relativně k povrchu pod dronem, který ale může mít různou výšku. Výškoměr pak naměří ve stejné výšce různé hodnoty. Díky tomu je i ten sám o sobě nepoužitelný. Je tedy nutné pozici určit nepřímým sledováním okolního prostředí. K tomu má dron k dispozici kameru, jejíž obraz můžeme zpracovávat metodami vizuální lokalizace.

2.3 Vizuální lokalizace

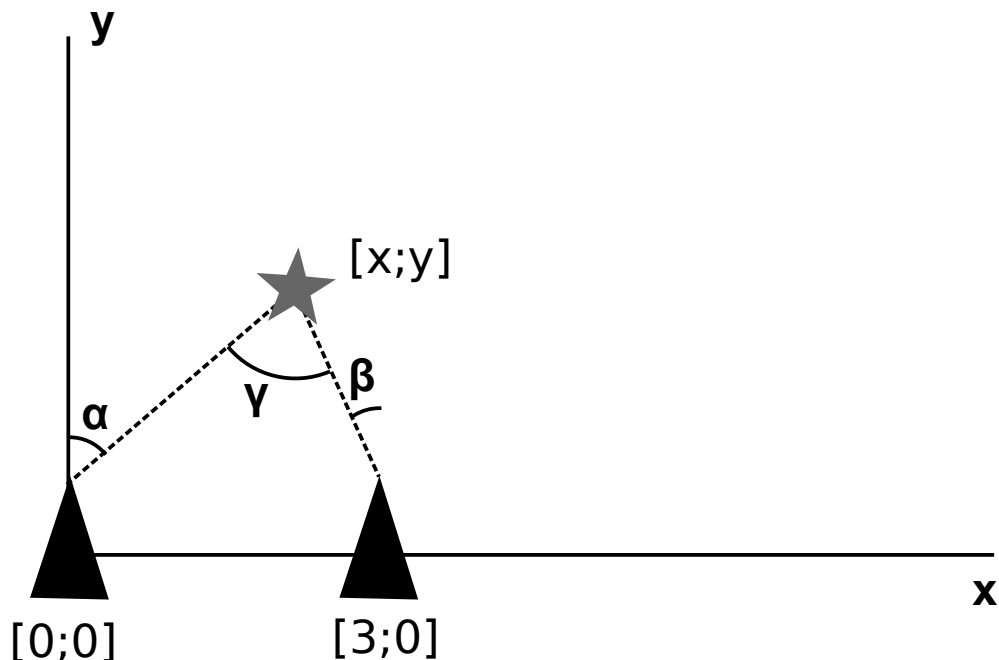
Vizuální lokalizace spočívá v identifikaci a sledování landmarků v okolí robota a orientaci podle nich. Pokud bychom si vytvářeli mapu landmarků, tedy jejich relativních pozic vzhledem k počáteční pozici drona, a následně měřili pozice pozorovaných landmarků za letu, uměli bychom lokalizovat snadno řešením geometrického problému naznačeného na obrázku 2.2. Úhly, pod kterými jsou landmarky pozorovány (přerušované čáry), by se daly vypočítat ze souřadnic landmarků na snímku z kamery. Pozici drona bychom pak snadno určili pomocí goniometrických funkcí.



Obrázek 2.2: Lokalizace robota pomocí mapy.

Uvedený předpoklad, existence mapy landmarků, je ale problém. Kamerou s jedním objektivem nemůžeme přímo určit vzdálenost landmarků od drona, tedy ani

jejich pozici. Vzdálenost je ale možné geometricky určit ze dvou snímků z různých míst (obr. 2.3). Aby to ale bylo přesné, je třeba dostatečný úhel mezi pozicí prvního snímku, pozorovaným bodem a pozicí druhého snímku (v obr. označen γ). To je ale při snaze o stabilizaci drona problém, protože pohyb je naopak nežádoucí.



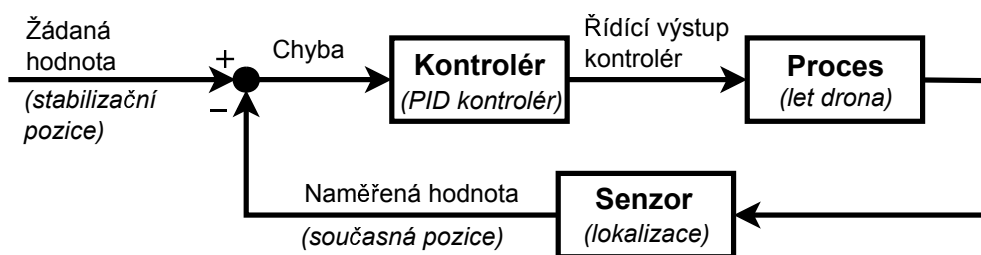
Obrázek 2.3: Lokalizace landmarku pomocí 2 pozic robota.

2.4 Model řízení

V každém okamžiku má robot k dispozici souřadnice stabilizační pozice, odhad aktuální pozice a odhad rychlostí. Na základě těchto informací je třeba robota řídit do stabilizační pozice. Zároveň je třeba zajistit, aby dron tuto pozici nepřelétl a nezačal kolem ní kmitat. To je složitější díky zpoždění spojení mezi dronem a ovládacím počítačem, které může dosahovat řádově desetin sekundy. Tento problém samotné stabilizace na základě známé pozice je již jednoduše řešitelný – je typickým případem použití PID kontroléru.

PID kontrolér [15] je zařízení, které vypočítává *chybu* jako rozdíl mezi měřenou hodnotou kontrolovaného procesu (zde pozicí) a žádanou hodnotou a snaží se tuto *chybu* minimalizovat svým výstupem, kterým je řízen aktuátor ovlivňující kontrolovaný proces (obr. 2.4). Tento cyklus měření a řízení tvoří uzavřenou smyčku nebo také zpětnovazební smyčku. Důležitou vlastností PID kontroléru je schopnost předcházet a utlumovat kmitání. Popis PID kontroléru a detaily jeho funkcionality uvádíme v 5.2.

Kontroléru řídicí systém poskytuje současnou a stabilizační pozici. Na základě jeho výstupu pak vypočítá vektor žádaného zrychlení a ten pak geometricky převede na náklon drona. Díky tomu se bude dron jak nehybně vznášet, pokud bude



Obrázek 2.4: Schéma funkčnosti obecného kontroléru.

v stabilizační pozici, tak i vracet do této pozice po vychýlení. Poletí však vždy přímočaře nehledě na překážky. Drona řízeného PID kontrolérem tedy není možné například odnést za překážku umístěnou mezi jeho novou pozicí a stabilizační pozicí.

Ačkoli PID kontrolér dokáže řešit kmitání kolem stabilizační pozice, je možné tento problém ještě více eliminovat. Jelikož je překmitávání způsobeno zpožděním mezi senzory a aktuátory, bylo by potřeba řídicímu systému dodávat odhad stavu (zejména pozici drona) co nejlépe stavu, ve kterém se bude dron nacházet ve chvíli, kdy se právě vydávaný řídicí příkaz vykoná.

2.5 Fúze a filtrování senzorů

Pro řízení drona potřebujeme odhad stavu, zejména pozice a rychlosti. Měření, která nám ale hardware poskytuje, jsou typicky zašuměná a některá navíc redundantní. Z těchto měření, znalosti předchozího stavu a pravděpodobnostního modelu spolehlivosti senzorů a letové dynamiky chceme získat tento odhad stavu společně s informací o přesnosti odhadu. Pro model řízení navíc potřebujeme i schopnost předpovídat stav v blízké budoucnosti (dále označujeme jako schopnost *predikce*).

Tyto potřeby můžeme realizovat pomocí Bayesovských filtrů. Kvůli nelinearitě letového modelu jsem zvolil rozšířený Kalmanův filtr [14] (dále EKF – Extended Kalman Filter). Ten odhad stavu aproximuje normálním rozložením a dron se pak může řídit mediánem a kovariancí tohoto rozložení. Problémem tohoto filtru je neschopnost pracovat s multimodálními pravděpodobnostními rozloženými a tedy nemožnost udržování více lokačních hypotéz.

2.5.1 Alternativy

Alternativní volbou by mohly být částicové filtry [10], které linearitu vůbec nepředpokládají a hlavně dokáží pracovat s více lokačními hypotézami, tedy uvažovat více různých možných lokací, dokud není možné lokaci přesně určit. Nezvolil jsem je ale proto, že bych v jejich případě očekával problémy s velkým stavovým vektorem a velkou výpočetní náročností potřebnou pro přesnost srovnatelnou

s EKF.

Dále by bylo možné používat tzv. unscented variantu Kalmanova filtru [13]. Ta též nepředpokládá linearitu a navíc zjednodušuje implementaci, neboť oproti EKF není třeba počítat matice parciálních derivací letového a sensorického modelu.

2.5.2 Letový a sensorický model

Abychom mohli predikovat stav (a pro řídicí systém zejména pozici) v budoucnosti, potřebujeme znát letovou dynamiku drona, jakási „pravidla“, podle kterých se stav v čase mění, podle kterých se dron pohybuje. Tuto letovou dynamiku popisuje letový (obecně pohybový) model. Formálněji vzato je letový model funkce zobrazující stav a řídicí pokyn pro drona na novější stav. Tento model je pro drona třeba odvodit na základě fyzikálního chování jeho letu. Tomu se věnuje část 4.4.1.

Sensorický model udává vztah mezi vektory stavu a měření. Říká, jak jsou které senzory spolehlivé a jak vypovídají jednotlivé naměřené veličiny o stavu drona. Tomu se věnuje část 4.4.2.

3. Lokalizace a mapování

3.1 Problém (vizuální) SLAM

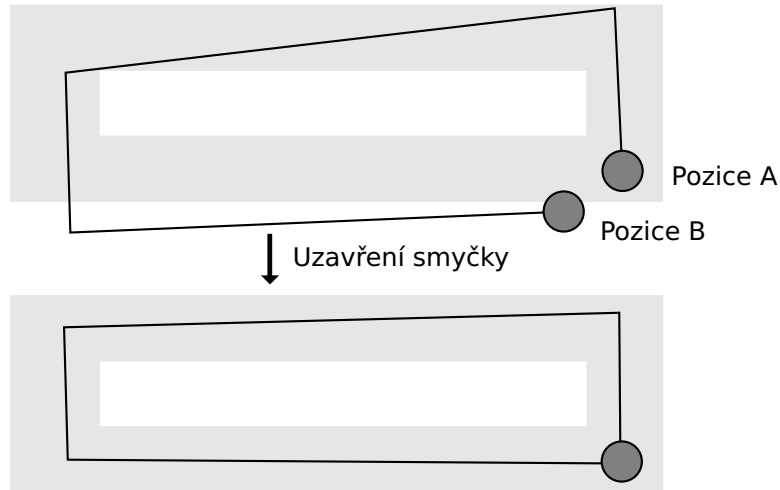
Tato kapitola se zabývá jedním z nejdůležitějších problémů robotiky, simultánní lokalizací a mapováním (dále jen SLAM). Nejprve uvedeme obecný popis tohoto problému přeložený z [11]:

Na SLAM problém narážíme, když chceme lokalizovat robota, který nemá přístup k předvytvořené mapě prostředí kolem robota ani zdroj informací o vlastní pozici. Místo toho má robot k dispozici jen posloupnost měření ze sensorů (značíme z) a posloupnost řídicích příkazů (značíme u), které robot vykonával. Měření je souhrn (ve formální robotice vektor) všech dat naměřených v daném čase. Příkladem části měření může být například rychlost aktuátorů. Řídicím příkazem je souhrn všech data ovlivňujících stav robota, která jsou mu v daném čase odeslána. Příkladem části vektoru řídicího příkazu může být například příkaz pro otáčení robota určitou rychlostí. Označení SLAM pak popisuje výsledný problém: robot vytváří mapu okolního prostředí, zatímco se simultánně lokalizuje v této mapě. SLAM je pro svoji komplexnost jedním z nejobtížnějších problémů pravděpodobnostní mobilní robotiky.

V našem případě jsou měření souřadnice a identifikátor *význačného bodu* v obrazu. Význačné body jsou body v obrazu, zobrazující místa okolního prostředí, která dokážeme rozpoznat z různých úhlů a vzdáleností a párovat s dalšími pozorováními toho samého místa. Místo v prostředí korespondující s význačným bodem označujeme jako *landmark*. Všimněme si, že tyto dva pojmy je často možné zaměnit. Většinou ale používáme označení *význačný bod*, pokud mluvíme o zpracování obrazu a *landmark*, pokud mluvíme o lokalizaci.

Vizuální a monokulární (tedy s kamerou s jediným objektivem) varianta problému SLAM přidává další komplikace. S jediným objektivem není možné okamžitě určit vzdálenost landmarku od drona z jediného snímku z kamery. Můžeme ze souřadnic bodu vypočítat jen směr, kterým je pozorován. Landmark můžeme tedy lokalizovat (obr. 2.3) a zanést do mapy teprve po alespoň 2 pozorováních. Kvůli přesnosti je navíc třeba provést triangulaci pomocí dvou co nejvzdálenějších pozorování landmarku. K tomu je ale třeba opět přesná lokalizace robota.

Od SLAM systému navíc chceme nejen aby byl schopný iterativně budovat mapu a určovat lokaci. Je třeba na základě nových měření dokázat opravovat i své předchozí chyby. Typickým případem potřeby této schopnosti je *uzavření smyčky*. To je situace, která nastane, když se robot pohybuje nějakou dobu neznámým prostředím a poté se vrátí do známé části mapy. Jelikož vytváření mapy není dokonale přesné, je třeba v tomto momentu upravit mapu tak, aby lokace drona v nové a staré části mapy odpovídala. Ilustrace je na obrázku 3.1 a 3.2.



Obrázek 3.1: Uzavření smyčky. Pozice A a B sdílí pozorování některých landmarků.

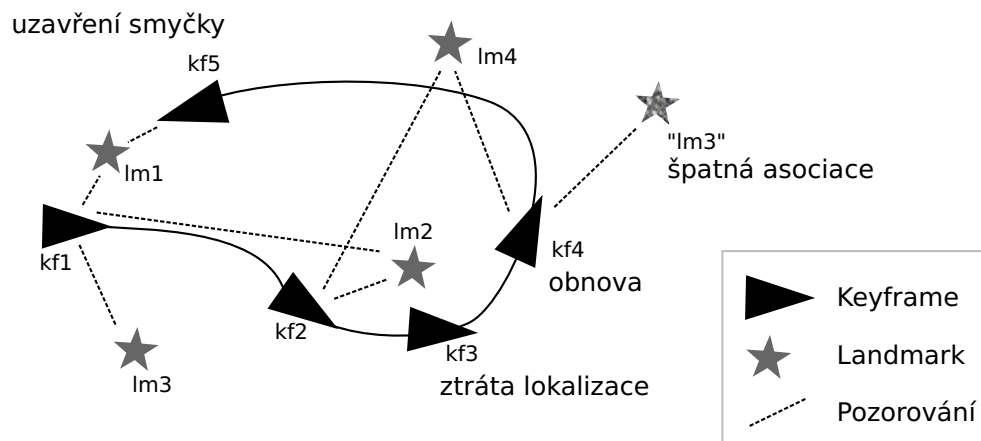
3.2 Průběh lokalizace a mapování

Nejdříve obecně popíšeme fungování SLAM systému používajícího (stejně jako námi použitý PTAM) grafovou reprezentaci mapy (obr. 3.2). Mapa je graf obsahující údaje o jednotlivých landmarcích. Tyto údaje obsahují zejména polohu landmarku a informaci o jeho podobě, která se používá pro jeho rozpoznání. Spolu s landmarky jsou v mapě uloženy i *keyframy*, což jsou informace o momentu, ve kterém byly landmarky pozorovány. Keyframe obsahuje snímek z kamery a lokaci robota. Vrcholy grafu, landmarky a keyframy, jsou propojeny 2 typy hran. Mezi keyframy vedou hrany (znázorněné v obrázku černou křivkou) reprezentující pohyb drona. Dva keyframy jsou tedy spojeny hranou, pokud za sebou bezprostředně následují v čase (není mezi nimi další keyframe). Mezi keyframem a landmarkem vede hrana (znázorněná v obrázku přerušovanou úsečkou), pokud byl landmark robotem pozorován na snímku kamery daného keyframu.

Keyframem může být teoreticky každý snímek z kamery, ale z výkonnostních důvodů tomu tak obvykle není. Přidání nového keyframu je totiž jediný moment, kdy SLAM systém přidává nové landmarky do mapy. Navíc zpřesňuje i pozice ostatních vrcholů mapy způsobem, který rozebereme dále. Právě výpočet tohoto zpřesnění je výkonnostně náročný.

3.3 Zvolená metoda

Systém pro absolutní vizuální lokalizaci v této práci využívá metodu PTAM (Parallel Tracking And Mapping) publikovanou v [2] společně se stejnojmennou zveřejněnou knihovnou. PTAM je navržen pro lokalizaci pozice kamery v neznámém prostředí a je obecnější než SLAM v tom smyslu, že v původní implementaci nevyužívá ani řídicí příkaz, který by u kamery přirozeně nedával smysl. Původním účelem bylo vytváření prostředí s rozšířenou realitou používané hlavně pro hry.



Obrázek 3.2: Graf průběhu lokalizace a mapování robota.

PTAM v dvou oddělených vláknech řeší lokalizaci a mapování. To umožňuje využít pro zpřesňování mapy na základě nových pozorování výpočetně náročné optimalizační techniky většinou nepoužívané v aplikacích fungujících v reálném čase. Díky tomu systém svou přesností a robustností zdatně konkuruje současným SLAM implementacím. To jeho autoři dokládají srovnáním přesnosti s konkurenčním EKF-SLAM popsaným např. v [11], kde PTAM přesvědčivě vítězí. Kromě přesnosti se dokáže i rychle zotavit ze ztráty lokalizace, což je vzhledem ke kvalitě obrazu přijímaného z drona velmi důležité. [2] (volně přeloženo)

3.3.1 Spojení PTAM a EKF

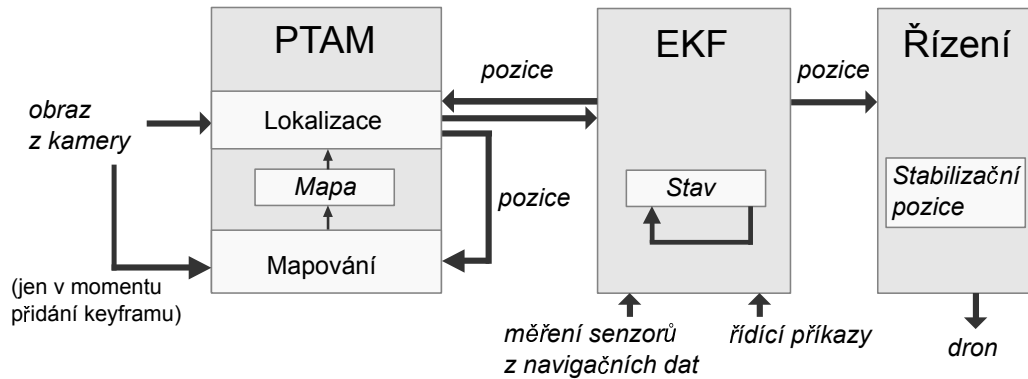
Kromě vizuální lokalizace náš systém obsahuje i relativní lokalizaci (viz 2.1) pomocí rozšířeného Kalmanova filtru (EKF), který pracuje s daty ze senzorů a odhaduje změny polohy na základě svého pohybového modelu (viz 4). Tyto 2 metody lokalizace nejsou redundantní, naopak se vhodně doplňují a přispívají k robustnosti a přesnosti výsledného odhadu pozice.

Různé části našeho systému využívají informaci o lokaci z ostatní částí takto (obr. 3.3:

- Řídicí systém ovládající pohyb drona se rozhoduje na základě odhadu stavu stanoveného EKF. Ten je tedy finálním výstupem celého systému lokalizace v práci.
- PTAM využívá odhad pozice na výstupu EKF, aby odhadl, kde v obrazu budou pozorovány které landmarky a tuto informaci použil jako heuristiku při párování pozorovaných landmarků v obrazu se známými landmarky

v mapě. To by jinak bylo pomalé, neboť by se musely porovnávat na shodu každý pozorovaný landmark s každým landmarkem v mapě. Více o tomto algoritmu uvádíme v části 3.5.

- EKF využívá pozici z PTAM jako měření z pomyslného senzoru měřícího pozici. EKF jeho výstup filtrací zpřesňuje a v případě, že PTAM ztratí lokalizaci, např. díky rychlému pohybu a rozmazání, EKF bude dál poskytovat řízení odhad pozice aktualizovaný relativní lokalizací.



Obrázek 3.3: Diagram komunikace jednotlivých částí lokalizace a řízení drona.

3.4 Sledování landmarků

Lokalizace a mapování ze snímku videa potřebuje získat seznam konkrétních úspěšně pozorovaných landmarků a jejich souřadnice v obraze. Je přitom žádoucí minimalizovat počet špatných asociací, tedy falešných detekcí známého význačného bodu v jiném místě. Takový jev totiž může způsobit, že systém určí špatnou lokaci a pak poškodí mapu přidáním nových landmarků na špatná místa mapy. Bohužel tento problém nelze úplně eliminovat, neboť v reálném světě se přirozeně objevují opakované vzory.

Na práci PTAMu s detekcí landmarků v obraze je charakteristické to, že jsou přijaté snímky přeškálovány na 4 různá rozlišení – snímek se postupně 3x přeškáluje tak, že hrany každého dalšího snímku budou mít poloviční velikost. Této čtveřici snímků říkáme *pyramida*. Každý keyframe pak obsahuje tyto 4 snímky. Účelem pyramidy je zvýšit robustnost metody, neboť malé přeškálované snímky tolik netrpí například rozmazáním. Navíc se s menšími snímky rychleji pracuje, takže jsou využívány i z výkonnostních důvodů.

V jednotlivých snímcích pyramidy se pak pomocí FAST detektoru hledají význačné body. Se známými landmarky se pak porovnávají a párují na základě podobnosti určené pomocí *SSD skóre*[4][2] v lokalizačním algoritmu popsáném v části 3.5.

3.5 Lokalizace

Lokalizační systém přijímá od zpracování obrazu snímky z kamery a udržuje v reálném čase odhad pozice kamery v kartézských souřadnicích uvnitř trojrozměrné mapy (těmto souřadnicím budeme říkat *globální*). Mapu lokalizační systém potřebuje mít už vytvořenou (byť se může měnit). Inicializaci a úpravy mapy zajišťuje mapovací část popsaná v 3.6.

Pro každý snímek provede systém tuto dvoufázovou lokalizační proceduru[2]:

1. Predikční krok EKF dodá PTAMu odhad pozice. Viz 3.3.1.
2. Snímek se 3x přeškáluje na poloviční velikost. Tyto snímky dohromady tvoří *pyramidu*.
3. Ve všech snímcích pyramidy nalezneme pomocí FAST[12] detektoru význačné body – možná pozorování landmarků.
4. Landmarky z mapy jsou promítnuty (viz 3.5.1) do roviny snímku na základě odhadu pozice z bodu 1). Tím získáme očekávané souřadnice (v px), kde by měly být landmarky pozorovány. V okolí těchto souřadnic se pak pro landmarky pokusíme najít odpovídající bod v nejmenším snímku pyramidy. Toto provedeme z výkonostních důvodů s maximálně 50 landmarky.
5. Odhadovaná lokace drona se vypočte podle pozorovaných landmarků a jejich pozic (viz 3.5.2).
6. Landmarky v mapě se znovu promítnou do roviny snímku na základě zpřesněné pozice. Ve snímku se pak hledá až 1000 landmarků ve všech snímcích pyramidy. Poměr počtu nalezených a hledaných landmarků je použit jako indikátor kvality lokalizace.
7. Konečný odhad lokace drona se vypočte podle všech pozorovaných landmarků a jejich pozic (viz 3.5.2). Tím se docílí vyšší přesnosti než v první fázi s 50 landmarky.

3.5.1 Projekce bodů mapy do roviny přijatého snímku videa

Abychom dokázali rychle identifikovat známé landmarky (uložené v mapě) ve snímku přijatém z kamery, potřebujeme odhadnout jejich souřadnice ve snímku na základě současné pozice drona – *promítnout* tyto body mapy do obrazu. Používaná knihovna při projekci bere v úvahu i *soudkovitost*, vadu obrazu, kterou kamera drona trpí. Výpočet projekce je popsán v [2] (sekce 5.2).

3.5.2 Určení polohy drona na dle pozic pozorovaných landmarků

K určení pozice drona máme k dispozici souřadnice polohy landmarků v mapě a odpovídající souřadnice jejich pozorování v obrazu. Snažíme se pak najít takovou pozici drona (souřadnice a rotaci), aby souřadnice pozorovaných landmarků

promítnutých do mapy maximálně odpovídaly souřadnicím jejich pozorování. Detaily implementace této optimalizační procedury knihovny PTAM uvádí [2] (sekce 5.4).

3.5.3 Detekce ztráty lokalizace a její důsledky

Pokud hodnota indikátoru kvality lokalizace vypočteného v lokalizační proceduře (viz 3.5) klesne pod určitou hranici, je lokalizace označena za *špatnou* nebo pod další hranicí za *ztracenou*.

Pokud je lokalizace *špatná*, mapovací část popsaná v části 3.6 nesmí přidávat nové body do mapy. Pokud by totiž odhad pozice byl nepřesný, vložené body by měly nepřesnou polohu a mapa i další lokalizace by tím byla poškozena.

Pokud je lokalizace *ztracená*, přestane PTAM poskytovat použitelný odhad pozice a systém se přestane řídit výsledky (zejména polohou) vizuální lokalizace. Bude se dočasně řídit prostředky relativní lokalizace (viz 2.1) popsanými v kapitole 4, dokud nebude lokalizace obnovena.

3.6 Mapování

Mapování je proces budování trojrozměrné mapy landmarků a keyframů. Má 2 fáze, inicializaci před spuštěním lokalizace (a před vzletem drona) a průběžné přidávání nově pozorovaných bodů.

3.6.1 Inicializace

Inicializace je provedena pomocí 2 keyframů. Ty jsou potřeba, neboť kamerou s jedním objektivem nemůžeme přímo určit vzdálenost pozorovaných bodů od drona, tedy ani jejich pozici. Vzdálenost se dá určit geometricky ze dvou různých keyframů. (obr. 2.3). Aby to ale bylo přesné, je třeba dostatečný úhel mezi pozicí prvního snímku, pozorovaným bodem a pozicí druhého snímku (v obr. označen γ). To je ale při snaze o stabilizaci drona problém, protože pohyb je naopak nežádoucí. Otázkou, kdy přidat keyframe, se zabýváme dále v části 3.6.2.

Uživatel obsluhující drona pomocí rozhraní programu vytvoří první keyframe, provede translační pohyb (posune drona) a vytvoří druhý keyframe. PTAM pak vytvoří počáteční mapu obsahující tyto 2 keyframy a landmarky pozorované z obou z nich. Polohu těchto landmarků určí pomocí *5 point algoritmu* popsaného v [8] na základě pozic keyframů. Zbývá tedy určit ty. Pozice prvního keyframu přidaného při inicializaci bude prohlášena za počátek souřadnic, pozici druhého keyframu je však třeba určit na základě směru a délky translačního pohybu.

Jelikož tuto vzdálenost nelze určit z měřítka obrazu, jak uvádí ??, je třeba ji určit jiným způsobem. Nejjednodušším způsobem, jak to udělat, je určit směr

a vzdálenost, kterým bude translační pohyb učiněn. Uživatel tedy při inicializace lokalizace provede pohyb dronem vzhledem k pohledu kamery doprava o zhruba 10cm. Tím je poloha druhého keyframu jednoznačně určena. Případná odchylka délky translace o několik centimetrů metodě nevadí, odůvodnění je uvedeno v části 5.3.1.

3.6.2 Přidání keyframu

Průběžné mapování pak probíhá tak, že pokud je vytvořen keyframe, je přidán do grafu (obr. 3.2) a landmarky, jejichž pozice ještě nebyla určena, ale už jsou pozorované podruhé, se do mapy přidají. K výpočtu jejich pozice, je využito epipolární hledání popsané v [2] (část 6.2). To využívá dvě pozorování landmarků z různých pozic drona a geometricky určuje pozice landmarků. Ilustrace je na obrázku 2.3 na začátku kapitoly.

Vhodná volba okamžiku vložení keyframu je velmi důležitá pro rychlost a robustnost metody. Na keyframy je totiž kladeno několik částečně protichůdných požadavků:

- Nové landmarky jsou přidávány pouze z keyframů, místa pozorovaná mezi keyframy nejsou zpracována.
- Mezi pozicemi sejmutí po sobě přidaných keyframů by měl být pro správnou funkčnost epipolárního hledání dostatečně dlouhý translační pohyb jeho délka závisí na vzdálenosti landmarků od objektivu.
- Příliš mnoho keyframů způsobuje zpomalení celého systému.

Druhý bod je při stabilizaci drona velmi problematický. Proto jsme se při implementaci rozhodli, že moment přidání keyframu bude čistě v režii uživatele. Ten tedy může jak zmapovat prostředí letu přidáním keyframů ještě před vzletem drona, tak přidávat keyframy průběžně tak, jak bude dron objevovat scénu. Keyframy uživatel přidává klávesou k nebo příslušným tlačítkem.

4. Fúze a filtrace senzorů

Pro řídicí systém potřebujeme co nejpřesnější resp. nejpravděpodobnější odhad stavu drona, zejména jeho pozice a rychlosti. Tento stav vyjadřujeme pravděpodobnostním rozdělením a k jeho odhadu máme k dispozici mnoho vstupních dat: výstup vizuální lokalizace, navigační data 1.3.1 a předchozí stav. Tato kapitola tedy řeší, jak na základě těchto vstupů průběžně určovat nejpravděpodobnější stav a jeho přesnost. To realizuje filtr.

4.1 Základní pojmy a schopnosti filtru

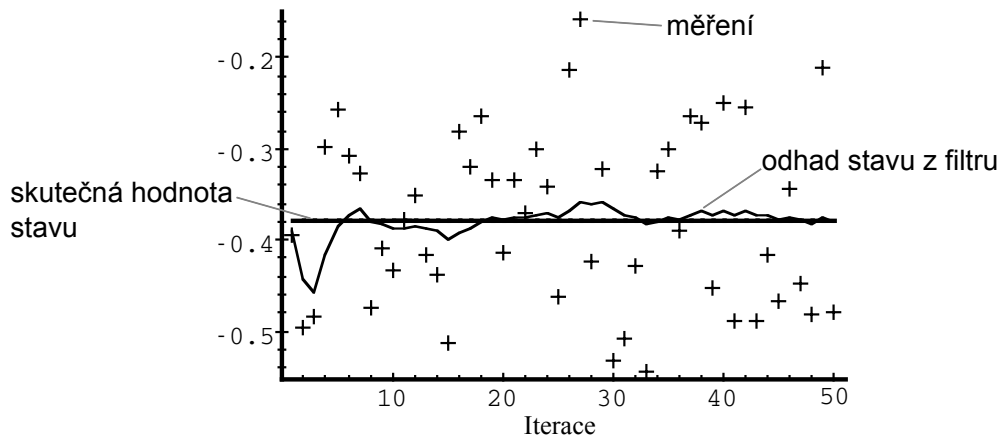
Jedním z problémů drona je zpoždění spojení mezi ním a ovládacím počítačem. Prodleva mezi měřením resp. zaznamenáním snímku videa na dronovi a příjmem těchto informací na počítači je řádově desatina sekundy. Stejně tak je zpožděné vykonání řídicího příkazu odeslaného na drona. Proto bychom pro řídicí systém, který odesílá řídicí příkaz dronovi na základě znalosti jeho stavu, potřebovali znát odhad stavu aktuální k okamžiku vykonání tohoto řídicího příkazu. Této schopnosti filtru předpovídat budoucí stav říkáme *predikce*.

Abychom mohli predikovat stav v budoucnosti, potřebujeme znát letovou dynamiku drona, jakási „pravidla“, podle kterých se stav v čase mění, podle kterých se dron pohybuje. Tuto letovou dynamiku popisuje letový (obecně pohybový) model. Formálně vzato je letový model funkce zobrazující pravděpodobnostní rozdělení stavu a řídicí příkaz pro drona na novější pravděpodobnostní rozdělení stavu. Tento model je pro drona třeba odvodit na základě fyzikálního chování jeho letu.

Kromě pohybového modelu obsahuje filtr sensorický model, který definuje vztah mezi stavem a měřením. Říká, jak jsou které senzory spolehlivé a jak vypovídají jednotlivé naměřené veličiny o stavu drona. Formálně vzato je pohybový model funkce zobrazující pravděpodobnostní rozdělení stavu na měření, které tomuto stavu odpovídá. Tomuto modelu se věnuje část 4.4.2.

Fúzí označujeme schopnost filtru odvodit jednotlivé části stavu z více různých vstupů filtru. Například rychlost drona se dá odhadnout z navigačních dat, která přímo obsahují její odhad na základě spodní kamery, dále z výstupu vizuální lokalizace (odečtením posledních dvou poloh drona a vydělení časem), nebo na základě pohybového modelu drona.

Filtrací označujeme schopnost filtru odhadovat stav přesněji, než jej měří jednotlivá měření. Tím filtr prakticky potlačuje nepřesnosti (nebo také šum) měření. Příklad výsledku filtrace při odhadování hodnoty stavu konstantního procesu na základě měření s velkou mírou šumu je na obr. 4.1.



Obrázek 4.1: Výsledek filtrace při odhadování hodnoty stavu konstantního procesu. Měření obsahuje šum.[14]

4.2 Fungování rozšířeného Kalmanova filtru

Jako filtr používáme v práci rozšířený Kalmanův filtr (dále jen EKF – Extended Kalman Filter), variantu Kalmanova filtru pro nelineární procesy. Chování tohoto filtru bude popsáno v následujících částech. Vnitřní fungování a důvody, proč filtr funguje, jsou svou komplexností nad rámec této práce a proto sahá náš popis jen na úroveň potřebnou k pochopení chování a implementaci filtru pro konkrétní užití (v tomto případě odhad stavu letícího drona). V práci nepopisujeme konkrétně například výpočty přesnosti a problémy způsobené nelineárností procesu. Díky tomu velká část popisu platí jak pro EKF, tak pro KF. Hlubší detaily fungování rozšířeného Kalmanova filtru dobře popisuje například [14].

EKF přijímá měření ze senzorů drona, výstup vizuální lokalizace (který je také brán jako měření) a řídicí příkazy a tyto informace zpracovává pomocí znalosti letového a sensorického modelu a odhaduje stav drona. EKF pracuje ve smyčce, v jejíž každé iteraci přijme měření a odhadne stav. Pořadí této iterace budeme k . Výpočet stavu je rekurzivní, stav v kroku k je určen na základě minulého stavu v kroku $k - 1$. Na výstupu filtru je pak nejpravděpodobnější stav filtrovaného procesu (letu drona) daný střední hodnotou *multivariačního normálního rozdělení*, které odhad (vícerozměrného) stavu reprezentuje, a kovarianční matice tohoto rozdělení, která poskytuje informaci o přesnosti odhadu.

Multivariační normální rozdělení je zobecnění jednorozměrného normálního rozdělení pro více rozměrů. Jedna z možných definic je, že náhodný vektor (vektor náhodných proměnných) je k -variačně normálně rozdělený, pokud je každá lineární kombinace jeho k složek normálně rozdělená (tuto vlastnost ale v práci neuvyžíváme). Multivariační rozdělení je pak takové k -variační rozdělení, kde k je alespoň 2.

Kovarianční matice těchto rozdělení jsou matice obsahující na pozici (i, j) kovarianci i -té a j -té složky náhodného vektoru. Pokud jsou dokonce jednotlivé složky náhodného vektoru navzájem nezávislé, obsahuje kovarianční matice nenulová čísla jen na hlavní diagonále. Hodnota matice na pozici (i, i) pak reprezentuje

rozptyl i -té složky náhodného vektoru daného rozdělení.

4.2.1 Stav a měření

Stav odhadovaný filtrem v kroku k je reprezentován vektorem $x_k \in \mathbf{R}^{10}$ obsahujícím po řadě 3 složky polohy v globálních souřadnicích (viz 3.5), rychlosti podél 3 os stejného souřadnicového systému, náklon, sklon, rotaci (obr. 1.1) a rotační rychlost: $(x, y, z, dx, dy, dz, \phi, \theta, \psi, d\psi)$. Podobně je reprezentováno i měření z_k , které EKF přijímá. Složky jeho vektoru měření jsou: $(dx', dy', dz', \phi, \theta, \psi, x, y, z)$. První 3 složky dx', dy', dz' , rychlosti drona, jsou na rozdíl od rychlostí vektoru stavu v souřadnicovém systému relativním k dronovi, jeho osy neodpovídají osám souřadnic použitým ve stavu. To je řešeno dál v části 4.4

Filtr uchovává odhad stavu jako multivariační normální rozdělení nad prostorem stavů \mathbf{R}^{10} dané střední hodnotou a kovarianční maticí. Předpokládáme, že střední hodnota stavu se vyvíjí v čase podle rekurentní rovnice 4.2.

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1})$$

Rovnice 4.2: Predikce stavu.

Funkce f , která vypočítává nový stav (přesněji jeho střední hodnotu), je hlavní částí pohybového modelu, který programátor do filtru implementuje a který určuje nový stav na základě současného stavu a řídicího příkazu. Náhodná proměnná w_{k-1} reprezentuje *šum procesu*, tedy změny stavu způsobené vnějšími vlivy. Při použití funkce f pro výpočet x_k samozřejmě neznáme vliv šumu (např. proudění vzduchu) na hodnotu stavu v aktuálním okamžiku, takže za w_{k-1} dosazujeme 0. Šum (označovaný maticí Q) místo toho použijeme k výpočtu druhé části pravděpodobnostního rozdělení stavu, kovarianční matice, tedy k určení přesnosti stavu.

Hodnoty stavového vektoru lze určovat z přijatých měření. Měření ale mohou být nepřesná a mohou o různých složkách stavového vektoru vypovídat jen nepřímo, nebo dokonce vůbec. Nepřímým měřením myslíme situaci, kdy lze složku stavu určit z vektoru měření, ale není v něm přímo obsažena. Příkladem může být například měření v jiných jednotkách než v jakých je uchováván stav. Vztah měření z_k a stavu (přesněji jeho střední hodnoty) x_k je vyjádřen funkcí h v rovnici 4.3. V uvedeném příkladě bude funkce h realizovat převod jednotek.

Náhodná proměnná v_k reprezentuje šum měření, nepřesnosti způsobené nedokonalostí senzorů. Při výpočtu z_k popsaném dále dosazujeme za v_k 0, podobně jako u funkce f . Šum (označovaný maticí R) ale využijeme dále. Za povšimnutí stojí, že filtr funkcí h v rovnici 4.3 nevypočítává stav na základě měření, ale naopak. Z odhadovaného stavu vypočítá uvedenou rovnicí dle funkce h odhad měření a to

$$z_k = h(x_k, v_k)$$

Rovnice 4.3: Vztah měření a stavu.

porovná s měřením přijatým (v tomto případě z drona).

Předpokládá se, že šum procesu a měření jsou na sobě nezávislé multivariačně normálně rozdělené náhodné proměnné s vektorem střední hodnoty obsahujícím samé nuly a kovariančními maticemi Q pro šum procesu a R pro šum měření, jak je uvedeno v 4.4.

$$\begin{aligned} w &\sim N(0, Q) \\ v &\sim N(0, R) \end{aligned}$$

Rovnice 4.4: Šum procesu a měření.

4.2.2 Údaje o procesu ovlivňující chování EKF

Aby filtr mohl odhadovat stav procesu, potřebujeme v kódu implementace filtru naprogramovat nebo uvést pohybový model (funkci f), sensorický model (funkci h , informaci o šumu procesu (matici Q) a šumu sensorů (matici R)¹. V případě drona šum procesu vypovídá o tom, jak moc se jednotlivé složky stavu drona za letu samovolně mění. Šum sensorů zase vyjadřuje jejich přesnost.

V naší implementaci Kalmanova filtru považujeme pro zjednodušení všechny složky šumu za nezávislé. V kovariančních maticích jsou pak vyplněny jen diagonální prvky – rozptyly jednotlivých složek šumu (viz 4.2).

Rozptyly kovarianční matice šumu procesu Q udávají míru vlivu prostředí na jednotlivé složky procesu, tedy zejména souřadnice, rychlosti a rotaci. Rozptyly kovarianční matice šumu měření R udávají míru nepřesnosti sensorů tak, že větší hodnoty těchto rozptylů způsobují menší „důvěryhodnost“ příslušné složky vektoru měření. Tomu se budeme věnovat v části 4.2.3.

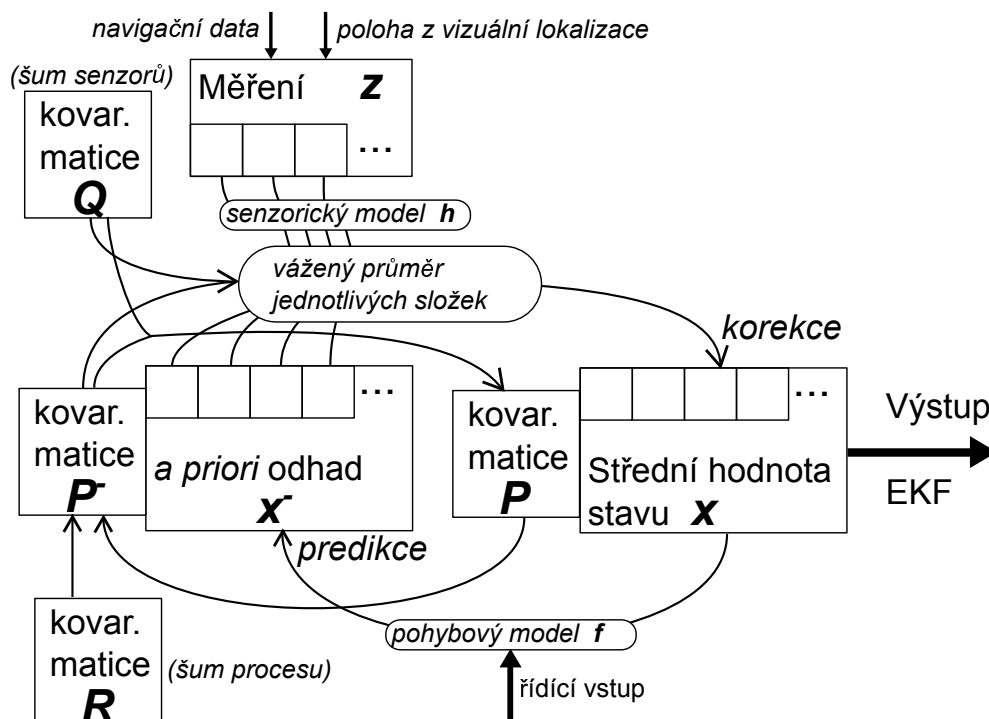
4.2.3 Predikční a korekční krok

Stavový vektor x_k je odhadován pomocí střídání *predikčního a korekčního kroku*. V predikčním kroku se aplikací pohybového modelu odhadne budoucí stav.

¹V některých publikacích se informace o šumu procesu resp. sensorů považuje za součást letového resp. sensorického modelu. To je však jen terminologický detail.

V korekčním kroku se odhad upraví podle přijatého měření. Tyto kroky se stále opakují a tím průběžně odhadují stav.

V *predikčním kroku* se aplikuje pohybový model, funkce f , na původní stav x_{k-1} , čímž se získá tzv. *a priori* odhad budoucího stavu x_k^- . Zároveň se získá *a priori* kovarianční matice odhadu stavu P_k^- na základě kovarianční matice stavu x_{k-1} a kovarianční matice procesního šumu Q . V korekčním kroku se *a priori* odhad stavu upraví podle přijatého měření z_k , čímž vznikne *a posteriori* odhad stavu x_k . Obdobně se z *a priori* kovarianční matice stavu vypočítá *a posteriori* kovarianční matice stavu P_k . Cyklus těchto dvou kroků je znázorněn na obrázku 4.5.



Obrázek 4.5: Schématické znázornění fungování EKF, predikčního a korekčního kroku.

V této práci záměrně nebudeme rozebírat podrobnosti predikčního a korekčního kroku. Podrobnosti lze nastudovat v [14]. Naopak se zaměříme na důsledky korekčního kroku, které jsou pro filtr důležité – ty jsou rozebrány dále v části 4.2.4.

4.2.4 Důsledky predikce a korekce

Vnitřní fungování filtru, predikční a korekční krok, má několik důležitých důsledků, které zde uvedeme. První důležitou vlastností filtru, konkrétně jeho korekčního kroku, je způsob stanovení *a posteriori* odhadu stavu na základě predikovaného *a priori* odhadu, měření a přesnosti těchto dvou hodnot dané jejich kovariancemi. Filtr informaci z odhadu stavu a měření zkombinuje podle jejich přesnosti. Pro lepší představu o chování stanovování odhadu stavu uvedeme příklad:

Když filtr vytváří odhad složku stavu, o které ví, že její hodnotu dokáže předpovědět velmi přesně (na základě hodnoty kovarianční matice odhadu stavu P_k) a zároveň ví, že měření této složky stavu je nepřesné, bere v úvahu víc *a priori* odhad predikce. Naopak pokud by bylo přesné měření a odhad by byl nepřesný například kvůli šumu procesu (který je neodhadnutelný), filtr by více bral v úvahu měření.

4.3 Chování filtru při ztrátě vizuální lokalizace

Důsledky predikce a korekce, konkrétně způsob určení *a posteriori* odhadu stavu na základě predikce a měření využívá systém při předávání pozice drona odhadnuté vizuální lokalizací filtru (viz 3.3.1). Vizuální lokalizační systém sleduje kvalitu své lokalizace a pokud je ta nízká, nebo byla lokalizace dokonce úplně ztracena (viz 3.5.3), je třeba zajistit, aby se filtr měřením pozice získaným z vizuální lokalizace neřídil a dočasně se řídil podle predikce na základě pohybového modelu. Díky tomu by filtr např. při odhadování pozice drona letícího směrem vpravo postupně pozici posouval vpravo. Chybu naakumulovanou v době bez měřené pozice by opravil ve chvíli, kdy se vizuální lokalizace opět obnoví a začne poskytovat filtru přesné informace o poloze.

Popisovaná schopnost filtru překlenout krátká období bez měření pozice drona je realizována úpravami kovarianční matice měření R při změně kvality vizuální lokalizace. Upravují se rozptyly složek měření, která obsahují informace o pozici z vizuální lokalizace. Díky tomu tato měření filtr při ztrátě lokalizace de-facto ignoruje.

4.4 Pravděpodobnostní modely

Filtr potřebuje pro predikční a korekční krok pohybový (v našem případě letový) a senzorický model. Ty jsou definovány funkcemi f a h . Funkce f zobrazuje aktuální stav a řídicí vstup v kroku k na budoucí stav v kroku $k + 1$, popisuje letový model drona. Funkce h definuje vztah mezi stavem a měřením, zobrazuje stav na měření. V předchozích částech přijímaly obě tyto funkce ještě jednu proměnnou, šum. Za ten je ale ve výpočtu vždy dosazena nula, takže jej jako proměnnou nemusíme uvažovat.

Funkce f definující letový model lze sestavit aplikací Newtonovské fyziky na let drona. Byl s drobnými úpravami převzat společně s několika konstantami ($c_1 - c_7$) přesně určujícími chování drona z [1]. Senzorický model je částečně přímočarý (hodnota složky měření je rovna hodnotě složky stavu, který měření odpovídá) v případech, kdy jsou hodnoty stavu přímo měřeny – například azimut. Částečně je získaný geometrickou rotací mezi *dron-centrickými* a *globálními souřadnicemi* – například zrychlení v horizontálních směrech.

Jako *globální souřadnice* označujeme jako souřadnice, se kterými pracuje vizuální

lokalizace. Jsou to souřadnice, jejichž počátek v místě, kde se spustila inicializace vizuální lokalizace. Oproti tomu *dron-centrické souřadnice* jsou souřadnice vztažené vzhledem k aktuální pozici drona. Jejich počátek je (zhruba) ve středu drona a osy směřují dopředu, doleva a vzhůru vzhledem k aktuální rotaci drona. Dron-centrický souřadnicový systém je ilustrován na obr. 1.1.

Kovariance procesního a sensorického šumu byly odhadnuty na základě složek stavového vektoru a povahy jednotlivých měření. Způsob zpracování sensorických dat softwarem drona před odesláním ovládacímu systému (této aplikaci) totiž není zdokumentován.

4.4.1 Letový model

Coby letový model hledáme funkci f zobrazující střední hodnotu stavu $(x, y, z, dx, dy, dz, \phi, \theta, \psi, d\psi)$ a řídicí vstup $(\bar{\phi}, \bar{\theta}, \bar{\psi}, \bar{dz})$ na predikci střední hodnoty stavu za čas dt . Tyto symboly mají následující význam:

- x, y, z jsou souřadnice polohy drona v globálních souřadnicích (viz 3.5).
- dx, dy, dz jsou rychlosti drona ve stejném souřadnicovém systému
- d^2x, d^2y jsou zrychlení ve stejném souřadnicovém systému
- ϕ a θ je náklon a sklon drona tak, jak jsou definovány na obrázku 1.1.
- ψ je azimut magnetického kompasu.
- $d\psi$ je rychlost změny azimutu v $^\circ/s$.
- is_flying je proměnná nabývající hodnot 0, pokud je dron na zemi, nebo 1, pokud letí. Je zavedena proto, aby, například při položení drona na nerovný povrch, systém ještě před vzletem neposouval pozici ve směru náklonu.

Horním indexem $+$ (např. x^+) značíme predikci dané složky stavu, tedy příslušnou složku vektoru *a priori* odhadu stavu x_k^- .

Funkci f resp. její výpočet budu uvádět pro přehlednost po částech dle jednotlivých souvisejících složek stavu.

Pro první 3 složky stavového vektoru jsou souřadnice drona v globálních souřadnicích. Platí pro ně dle mechaniky pevného tělesa pro krátké časové intervaly² vztah uvedený v rovnici 4.6.

Další 2 složky jsou horizontální rychlost drona opět v globálních souřadnicích. Rychlost je vypočtena na základě známého vztahu klasické mechaniky $v = v_0 + t \cdot a$. Zrychlení určíme jako kombinaci dvou složek: první je určena na základě náklonu drona v daném směru. Druhá je přímo úměrná zrychlení (v minulém kroku rekurzivního výpočtu), míří v opačném směru než toto zrychlení a reprezentuje odpor vzduchu. Konstanty pro tyto úměry byly určeny experimentálně v [1].

²V našem případě je krok EKF zhruba $5ms$.

$$\begin{pmatrix} x^+ \\ y^+ \\ z^+ \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} + dt \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix}$$

Rovnice 4.6: Predikce pozice

Pro výpočet zrychlení je použita projekce jednotkového vektoru ve směru osy z dron-centrické soustavy³ na horizontální rovinu $z = 0$ globálních souřadnic (viz 4.7). To můžeme udělat, neboť projekcí vzniklý vektor má směr shodný s vektorem zrychlení a jeho velikost vyjadřuje míru náklonu.

$$\begin{pmatrix} d^2x \\ d^2y \\ 0 \end{pmatrix} = c_2 \begin{pmatrix} \cos(\psi) \sin(\phi) \cos(\theta) & -\sin(\psi) \sin(\theta) \\ -\sin(\psi) \sin(\phi) \cos(\theta) & -\cos(\psi) \sin(\theta) \end{pmatrix}$$

$$\begin{pmatrix} dx^+ \\ dy^+ \end{pmatrix} = \begin{pmatrix} dx \\ dy \end{pmatrix} + dt \cdot is_flying \cdot c_1 \left(\begin{pmatrix} d^2x \\ d^2y \end{pmatrix} - \begin{pmatrix} dx \\ dy \end{pmatrix} \right)$$

Rovnice 4.7: Predikce vodorovného zrychlení

Další složka, vertikální rychlost, je určena opět dle vztahu $v = v_0 + t \cdot a$. Neaplikuje se ale žádná rotace, tentokrát předpokládáme, že dron letí přibližně vodorovně. Zrychlení určujeme přímo na základě námi odeslaného řídicího vstupu přímoú úměrou. Tak jako v minulém případě odečítáme obdobným způsobem odpor vzduchu. Výsledek je uveden v rovnici 4.8.

$$dz = dz + dt \cdot is_flying (c_7 \cdot \bar{dz} - c_8 \cdot dz)$$

Rovnice 4.8: Predikce vertikální rychlosti

Dalšími dvěma složkami je náklon a sklon (roll a pitch). Jeho změna je opět vypočtena jako součet dvou složek. První složka je přímo úměrná řídicímu příkazu, druhá reprezentuje tendenci drona vracet se do horizontální polohy (viz 4.9).

Poslední dvě složky stavu je azimut a rotační rychlost. Změna této rychlosti je opět určena jako součet zrychlení úměrného řídicímu příkazu a odporu (viz 4.10).

³Tedy vektor mířící z drona vzhůru kolmo na jeho povrch.

$$\begin{pmatrix} \phi^+ \\ \theta^+ \end{pmatrix} = \begin{pmatrix} \phi \\ \theta \end{pmatrix} + dt \cdot is_flying \cdot \left(c_3 \begin{pmatrix} d\bar{\phi} \\ d\bar{\theta} \end{pmatrix} - c_4 \begin{pmatrix} d\phi \\ d\theta \end{pmatrix} \right)$$

Rovnice 4.9: Predikce náklonu a sklonu

$$\begin{pmatrix} \psi^+ \\ d\psi^+ \end{pmatrix} = \begin{pmatrix} \psi \\ d\psi \end{pmatrix} + dt \begin{pmatrix} d\psi \\ c_5 \cdot d\bar{\psi} - c_6 \cdot d\psi \end{pmatrix}$$

Rovnice 4.10: Predikce azimutu a rotační rychlosti

4.4.2 Senzorický model

Hledáme funkci h zobrazující střední hodnotu stavu na střední hodnotu vektoru měření $(dx', dy', dz', \phi, \theta, \psi, x, y, z)$. Tyto symboly mají následující význam:

- x, y, z jsou souřadnice polohy drona v globálních souřadnicích (viz 3.5).
- dx', dy', dz' jsou rychlosti drona v drone-centrických souřadnicích
- ϕ a θ je náklon a sklon drona tak, jak jsou definovány na obrázku 1.1.
- ψ je azimut magnetického kompasu.
- $d\psi$ je rychlost změny azimutu v $^\circ/s$.

Tuto funkci resp. její výpočet budu uvádět pro přehlednost po částech dle jednotlivých souvisejících složek vektoru měření.

První dvě složky stavu jsou rychlosti drona v jeho lokálních, drone-centrických, souřadnicích. Toto zde značíme apostrofem. Zde osa x míří vpravo od drona, osa y vpřed a osa z vzhůru. Horizontální rychlosti tedy získám ze stavového vektoru otočením souřadnicového systému o azimut[17], jak je vyjádřeno v rovnici 4.11.

$$\begin{pmatrix} dx' \\ dy' \end{pmatrix} = \begin{pmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{pmatrix} + dt \begin{pmatrix} dx \\ dy \end{pmatrix}$$

Rovnice 4.11: Výpočet horizontální rychlosti v dron-centrických souřadnicích

Zbýlých 7 složek vektoru měření jsou po řadě vertikální rychlost, náklon, sklon, azimut a poloha v globálních souřadnicích určená vizuální lokalizací: x , y a z . Tyto hodnoty přímo obsahuje stavový vektor, takže je mohou použít beze změny.

4.5 Střednědobá predikce

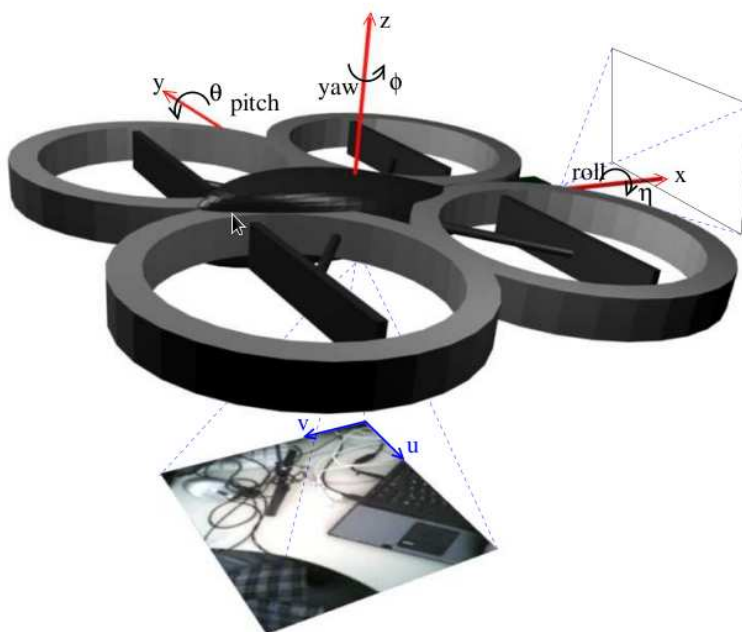
Kalmanův filtr sám o sobě dokáže předpovídat budoucí stav na krátké časové intervaly iterací predikčního kroku. Pokud chceme předpověď na delší čas, je možné triviálně nastavit v rovnici větší dt a provést jednu iteraci predikčního kroku. Taková předpověď je ale díky nelinearitě procesu velmi nepřesná, neboť EKF nelineární funkce aproximuje lineárními[14], což funguje dobře jen na malých intervalech, kde lineární funkce nelineární funkci téměř odpovídá. Čím větší je ale část nelineární funkce, tím větší je nepřesnost. Tento problém se dá vyřešit opakováním predikčního kroku s původním malým dt . Tím se ale zvyšuje výpočetní náročnost lineárně s velikostí dt .

Zvoleným řešením je oba přístupy zkombinovat. Prodloužit interval dt a provést několik predikčních kroků.

5. Řízení drona

5.1 Rozhraní

System drona umožňuje řízení pomocí jediného řídicího příkazu (posaného v 1.3.3), který nastavuje náklon, sklon, rotační rychlost a vertikální rychlost drona. Tyto hodnoty pak drone udržuje do další změny.



Obrázek 5.1: Souřadnicový systém drona [3]

Lokalizační systém oproti tomu dodává řídicímu systému souřadnice stabilizační a současné pozice. Úkolem řízení je pak plynule navigovat drona do stabilizační pozice a minimalizovat případný překmit (přelet stabilizační pozice) způsobený hlavně zpožděním spojení. Tento požadavek je typickým využitím PID kontroléru.

5.2 Kontrolér

PID kontrolér[15] je zařízení, které vypočítává *chybu* jako rozdíl mezi měřenou hodnotou kontrolovaného procesu a žádanou hodnotou a snaží se tuto *chybu* minimalizovat svým výstupem, kterým je řízen aktuátor ovlivňující proces. Tento cyklus měření a řízení tvoří uzavřenou smyčku nebo také zpětnovazební smyčku.

Pro PID kontrolér je charakteristický výpočet řídicího výstupu pomocí 3 složek a jejich konstant. Proporční, integrační a derivační. Proporční složka určuje míru reakce na velikost chyby, integrační složka míru reakce na integrál chyby a de-

rivační složka míru reakce na derivaci chyby, tedy rychlost její změny. Důsledky tohoto výpočtu a konstant jeho složek jsou rozebrány v 5.3.

$$x(t) = Pe(t) + I \int_0^t e(t)dt + D \frac{de(t)}{dt} \quad (5.1)$$

Rovnice 5.2: Výpočet řídicího výstupu PID kontroléru [15]

kde t je současný čas, $x(t)$ řídicí výstup kontroléru v čase t a $e(t)$ je chyba kontrolované veličiny v čase t .

5.3 Řízení

Cílem řídicího systému je přesunout drona z pozice (x, y, z, rot) získané lokalizací do cílové pozice (x', y', z', rot') , ve které chceme drona stabilizovat. Aby se zamezilo překmitům drona kolem cílové pozice způsobeným latencí síťového spojení, vychází řídicí systém z pozice (x, y, z, rot) , ve které se bude dron nacházet v budoucnosti. K tomu je využita predikční schopnost Kalmanova filtru popsána v sekci 4.5.

Jelikož jsou hodnoty x, y, z, rot navzájem nezávislé (jedna neovlivňuje druhou), je pohyb v těchto 4 stupních volnosti realizován nezávislými PID kontroléry s nastavitelnými parametry. Pomocí proporcionální složky, parametru P , je možné nastavit rychlost, se kterou se drona bude vracet do cílové pozice při vychýlení. Integrovaná složka, parametr I , nastavuje schopnost drona reagovat na trvalé vychylující vlivy jako například vítr. Derivační složka, parametr D , určuje sílu reakce na rychlost vychylování – zpomaluje přibližování a brání oddalování od cílové pozice.

5.3.1 Vliv nepřesnosti měřítka mapy na řízení

Při inicializaci vizuální lokalizace musí uživatel rukou provést translační pohyb (posunutí) dronem délky zhruba 10cm. Požadované délky ale nelze rukou dosáhnout přesně, což způsobí (jak uvádíme v 3.6) nepřesnost měřítka souřadnic dodávaných řídicímu systému lokalizací. Díky tomu nebude chyba vypočtená PID kontrolérem přesně v metrech a dron se bude vracet díky výpočtu PID kontroléru (5.2) do stabilizační pozice rychleji nebo pomaleji.

6. Uživatelské rozhraní

V této kapitole se budeme zabývat popisem rozhraní a ovládání našeho systému. Ten obsahuje jednoduché uživatelské rozhraní, jehož účelem je hlavně umožnit uživateli řízení drona a vizualizovat výstup lokalizačního systému. Rozhraní je realizováno pomocí knihovny SDL2. Tato knihovna původně určená k tvorbě počítačových her byla zvolena kvůli její výkonnosti. Její nevýhodou je naopak špatná podpora formulářových prvků. Rozhraní tedy takové prvky téměř neobsahuje.

6.1 Popis

Po spuštění aplikace si můžete v prvním dialogovém okně vybrat mezi režimem ovládání drona nebo simulačním režimem a zadat jméno záznamu, který bude v prvním případě vytvořen a v druhém přehrán. Po stisku tlačítka (nebo dvojitým odenterováním) se program připojí k dronovi nebo spustí simulaci.

Uživatelské rozhraní je rozděleno do několika oken (obr. 6.1):

1. Video z přední kamery drona
2. Vizualizace pohybu drona, pohled shora. Stavové informace jako sklon, náklon, rychlost nebo azimut.
3. Výstup lokalizace a mapování v 3D. Pohled tohoto okna můžete měnit táhnutím myši a přibližovat jejím kolečkem.

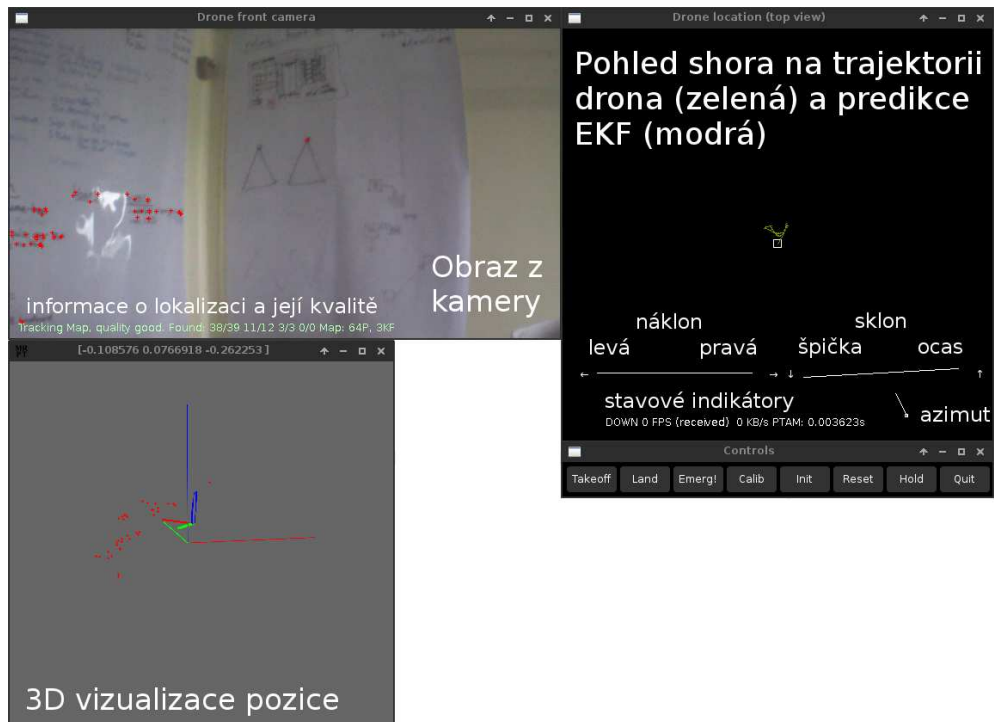
Díky rozdělení do jednotlivých oken můžete některá z nich zobrazovat např. na externím monitoru nebo měnit velikost okna č. 3 s 3D vizualizací. Kromě oken vypisuje program informace o svém průběhu na standardní výstup příkazové řádky. To lze využít například při ladění.

6.1.1 Interpretace okna č. 2

V druhém okně (vpravo) vidíme vizualizaci pohybu drona při pohledu shora a stavové informace.

Čtverec v jeho středu symbolizuje startovní pozici drona. V dolní části je vlevo indikátor náklonu ve směru doleva a doprava a červeně rychlosti ve stejném směru. Vpravo je indikátor náklonu ve směru dopředu a dozadu opět s rychlostmi. Náklonu dopředu odpovídá náklon indikátoru doleva. Vprostřed mezi nimi se nachází indikátor azimutu magnetického kompasu drona. Nakonec vlevo dole jsou indikátory stavů jako například let, nízký stav baterie nebo aktivovaná stabilizace.

V průběhu letu se v okně bude vykreslovat trajektorie drona. Přitom se zohledňuje azimut směru letu, takže pokud ukazuje indikátor azimutu nahoru a drone letí vpřed, bude se v okně vykreslovat trajektorie směrem od středu nahoru. Místa,



Obrázek 6.1: Uživatelské rozhraní

kde byla aktivována stabilizace, jsou vyznačena zeleným čtverečkem.

6.2 Řízení

K řízení se používají tlačítka, klávesy a klávesové zkratky. Zde je jejich seznam:

- X - Spuštění režimu stabilizace (indikován slovem HOLD). Ukončen bude libovolnou z kláves W/S/A/D.
- W/S/A/D - Náklon drona ve směru vpřed/vzad/vlevo/vpravo. Ihned opustí režim stabilizace.
- Q/Y - Let vzhůru/dolů.
- Shift-Q/Y - Vzlet/přistání.
- E/R - Natáčení (kolem svislé osy) vlevo/vpravo.
- Shift-E - Přepnutí emergency stavu. Dron okamžitě vypne motory a kontrolky svítí červeně.
- T - Kalibrace gyroskopu. Spouštět jen pokud je dron v rovině.
- Esc - Ukončení aplikace.

Tlačítka jsou k dispozici pro vzlet, přistání, emergency mód, aktivaci stabilizace, kalibraci gyroskopu a ukončení aplikace.

6.3 Použití programu z CD

Program je možné vyzkoušet pomocí přiloženého CD (Příloha 1). CD stačí vložit do počítače s procesorem Intel a Wi-Fi kartou a provést restart. Pokud váš počítač automaticky nespustí z médií, je to třeba ještě povolit v BIOSu počítače nebo zvolit start z CD klávesovou zkratkou. To je závislé na výrobci počítače. Systém z CD nastartuje bez jakéhokoli instalování. To může trvat několik minut. Na konci tohoto procesu uvidíte plochu a ikony pro spuštění aplikace.

Nyní zapněte AR.Drone 2 a položte jej na rovnou plochu dostatečně daleko od překážek. Připojte se k příslušné WiFi síti (*ardrone2...*) a spusťte aplikaci `drone_hold`. V zobrazeném okně zvolte režim flight a název nahrávky, která bude pořízena. Potvrďte tlačítkem OK nebo dvojitým stiskem klávesy Enter.

Ujistěte se, že má aplikace zaměření (focus) a zkalibrujte gyroskop stiskem klávesy T¹. Nyní můžete kombinací Shift-Q iniciovat vzlet. Pokud dron nevzlétne, pravděpodobně je vybitá baterie, což je indikováno textem ve spodní části lokalizačního okna. Pak je třeba ji nabít na více než 20% kapacity. Až doletíte do místa stabilizace, stiskněte X. Nyní je dron řízen stabilizačním systémem až do stisku některé z řídicích kláves W/S/A/D.

Přistát je možné tlačítkem Shift-Z. Drone pak začne svisle klesat a pár centimetrů nad povrchem vypne motory. V případě problémů je možné drona kdykoli okamžitě vypnout zkratkou Shift-E. Pak okamžitě vypne motory a eventuálně začne padat.

6.3.1 Konfigurace parametrů systému

Pokročilá konfigurace se provádí úpravou textového souboru `drone_hold.cfg`. Pro běžné použití programu však žádný zásah do konfigurace není třeba. Příklad se všemi podporovanými parametry a jejich významem následuje:

```
[control]
pid = [0.5 0 0.5] //Parametry P,I,D stabilizačního kontroleru
print_control = YES //Vypisovat ladící výstupy stabilizace drona?

[application]
replay_file = replay123 //Defaultní název nahrávky.
print_navdata = NO //Vypisovat navigační data přijatá z drona.
slowdown_times = 2 //Kolikrát zpomalit přehrávání nahraného letu
//oproti reálnému času.
no_interactive = NO //Vypne první okno s výběrem názvu nahrávky.
//Spustí rovnou simulaci.
```

¹Všechny následující akce kromě samotného řízení letu lze provést i tlačítkem

7. Verifikace

V práci byl představena komplexní systém, schopný stabilizovat drona na předem zvolené *stabilizační pozici*, tedy zabraňovat pohybu drona směrem od stabilizační pozice a zároveň jej navigovat zpět, pokud dojde k jeho vychýlení. Tak se může stát i vlivem vnějšího prostředí, ať už je to vliv přirozený, jako třeba vítr, nebo cíleně vyvinutý, jako odstrčení drona z pozice. Odchylce drona od stabilizační pozice budeme v této kapitole říkat zkráceně *odchylka*

Abychom mohli hodnotit kvalitu vytvořeného řešení, je třeba si nejdřív položit otázku, jaké vlastnosti od stabilizačního systému očekáváme, jakým způsobem je měřit a na čem je závislá schopnost systému dobře pracovat. Nároky kladené na stabilizační systém byly identifikovány takto:

1. Odchylka by neměla narůstat s časem.
2. Průměrná odchylka by měla být co nejmenší. Toto kritérium se však špatně měří, neboť je potřeba další systém, který bude lokalizovat drona přesně a odchylku v čase určovat, aby mohla být vypočítána průměrná odchylka.
3. Odchylka po nějakém stanoveném čase by měla být minimální.
4. Odchylka po ručním vychýlení by měla být po určitém čase na návrat do stabilizační pozice minimální
5. Dron by se měl po vychýlení o stanovenou vzdálenost vrátit do stabilizační pozice co nejrychleji. V tomhle případě je ale problém s překmitem, protože ten se těžko měří a detekuje. Není totiž jasné, jestli je rychlý návrat s velkým překmitem žádanější než pomalejší návrat menším překmitem.
6. Systém by měl být schopný provozu v co nejširším spektru různých prostředí. Typickými problémy pro vizuální zpracování obrazu jsou nízká úroveň světla, málo sledovatelných vizuálních bodů v prostředí nebo příliš blízké předměty zakrývající výhled.

7.1 Experiment

Na základě identifikovaných požadavků kladených na stabilizační systém jsme sestavili experiment, který komplexně ověřuje požadavky uvedené v bodech 1,3,4 a 6. Tento experiment spočívá v otestování robustnosti stabilizace několika opakovanými následujícího měření v různých prostředích:

1. Spustíme drona i ovládací aplikaci v režimu letu a navážeme spojení.
2. Inicializujeme vizuální lokalizaci translačním pohybem směrem vpravo od pohledu kamery a délky zhruba 10 cm.
3. Přidáme do mapy několik keyframů okolního prostředí. Je třeba pokrýt zejména úhel pohledu drona ve směru, kam bude kamera směřovat ve stabilizační pozici.

4. Položíme drona na vodorovnou podložku a zkalibrujeme gyroskop.
5. S dronem vzletíme a pomocí ručního ovládání klávesami doletíme do vhodné stabilizační pozice.
6. Spustíme odpočet 20ti sekund, aktivujeme režim stabilizace a olovnicí (nebo závěsem s malým, těžkým předmětem) určíme bod na podlaze, nad kterým se dron začal stabilizovat.
7. Dron rukou odstrčíme zhruba 20 cm od stabilizační pozice.
8. Po 20 sekundách olovnicí opět určíme bod na podlaze, nad kterým se dron aktuálně vznáší.
9. Změřením vzdálenosti těchto dvou bodů získáme odchylku pozic drona promítnutých svisle do roviny podlahy. Tím sice nejsme schopni měřit vertikální složku odchylku, ale to nám pro účely experimentu nevadí. Přesné měření ve vzduchu by totiž bylo těžké realizovat.
10. Znovu přehrajeme nahraný záznam letu a zhruba určíme, kolik landmarků dron při aktivaci stabilizační pozice pozoroval. Ze záznamu letu můžeme vypočítat i další údaje, jako například počet ztrát lokalizace a podobně.

Toto měření bylo provedeno několikrát v různých prostředích s různou světelnou intenzitou a množstvím význačných bodů. V interiéru i v exteriéru s vlivem větru.

7.2 Výsledky experimentu

7.2.1 Kolejní pokoj

Prostředí	Vizuálně bohaté prostředí, šero, malý interiér.
Počet význačných bodů	50
Naměřená odchylka	8cm
Poznámky	Vizuální lokalizace mezi kalibrací a letem ztratila lokalizaci, protože drone položený na podlaze nepozoroval dostatečný počet landmarků. Po vzletu se ale lokalizace okamžitě obnovila. Odchylka kolísala, ale nevykazovala trend zvětšování v čase. Za zmínku stojí, že přes velkou vizuální bohatost FAST detektor našel jen poměrně málo význačných bodů.

7.2.2 Fasáda domu

Prostředí	Nepříliš členitá zeď, dostatek světla, lehký vítr.
Počet význačných bodů	100
Naměřená odchylka	10cm
Poznámky	System po celou dobu měření neztratil lokalizaci. Na samotné zdi však nebyl schopný najít téměř žádné význačné body.

TODO: vepsat do práce zbytek měření

Závěr

Na začátku práce byla stručně představena platforma kvadrikopty *AR.Drone 2*, zejména její senzory a ovládání. Pozornost byla v této části zaměřena i na problémy, kterými platforma trpí a které musely být v práci zvláštním způsobem řešeny.

Hlavním cílem práce byla implementace systému řídicího kvadrikopty, který by ji byl schopen dlouhodobě udržet na místě a to bez akumulované chyby typické pro relativní lokalizaci (viz 2.1). Systém by měl být robustní proti vlivům vnějšího prostředí jako je vítr nebo vychýlení drona rukou. Měl by se dokázat vyrovnat s nepřesností senzorů, zpožděním komunikace s dronem a špatnou kvalitou obrazu z kamery, kterou je dron vybaven. To vše by měl dokázat v neznámých prostředích jen s informacemi, které si sám vytvoří.

Požadavek na stabilizaci drona vede na nutnost určení lokace. V prvních částech práce ukazujeme, že dron není schopný se bez akumulované chyby lokalizovat jinak než s použitím kamery a vizuální lokalizace. Aby se navíc dron dokázal orientovat i ve větší části okolního prostředí než jen v té, kterou zrovna objektivem kamery vidí, bylo by třeba si informace o prostředí zapamatovávat, vytvářet mapu.

Řešení stabilizace tedy nakonec vedlo na kompletní simultánní lokalizaci a mapování (známou pod zkratkou SLAM). Díky tomu, že má dron jen jednu kameru, potřebujeme řešit dokonce složitější variantu SLAMu: monokulární vizuální SLAM, tedy lokalizaci a mapování s použitím jediné kamery. Ta je komplikovaná především kvůli mapování, neboť z jediného objektivu se nedá přímočaře určit vzdálenost pozorovaných bodů. Velká část práce se tedy zabývá lokalizací, mapováním a zpracováním obrazu.

Problémy s nespolehlivostí senzorů i vizuální lokalizace a zpožděním spojení s dronem v práci řeší rozšířený Kalmanův filtr. Je to další velmi silný a v robotice používaný mechanismus, který pro řídicí systém odhaduje nejpravděpodobnější polohu drona. Tu dokáže určovat přesněji než samotná měření nebo výstup lokalizace mj. na základě znalosti fyzikálního modelu pohybu drona, který byl ve filtru implementován, a přesností používaných senzorů.

Konečně systém řeší i otázku samotného řízení – stabilizaci. Ta je ale ve chvíli, kdy známe polohu drona, prakticky jen triviální aplikací PID kontroléru, jednoho ze základních nástrojů teorie řízení. PID kontrolér je v systému užitečný i proto, že bude tlumit tendence drona překmitávat polohu, ve které má být stabilizován.

Všechny výše zmíněné části se podařilo implementovat a vyřešit tak všechny problémy, na které tento na první pohled jednoduchý problém stabilizace létajícího drona vede. Vzhledem k rozsáhlosti implementace ale nezbylo tolik času na ladění robustnosti systému nebo uživatelského rozhraní. Tam jsou tedy ještě mezery.

Na druhou stranu má implementovaná funkcionality velký potenciál na další rozšiřování. Jakmile se létající dron dokáže přesně lokalizovat a mapovat prostor kolem sebe, otvírá se mnoho možností pro implementaci další funkcionality. Popisovaný systém by se dal již relativně snadno rozšířit například o schopnost létat mezi zadanými místy (waypointy), prozkoumávat a vytvářet 3D mapu neznámých prostředí, transportovat náklad z místa na místo a podobně. Nejkomplikovanějším součástí všech těchto problémů je totiž právě lokalizace a mapování. V tomto směru tedy vidím největší možnost využití nebo nadstavby této práce.

Seznam použité literatury

- [1] ENGEL, J. – STURM, J. – CREMERS, D. Camera-based navigation of a low-cost quadcopter. V *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, s. 2815–2821. IEEE, 2012.
- [2] KLEIN, G. – MURRAY, D. Parallel tracking and mapping for small AR workspaces. V *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium*, s. 225–234. IEEE, 2007.
- [3] KRAJNÍK, T. et al. AR-drone as a platform for robotic research and education. V *Research and Education in Robotics-EUROBOT 2011*, s. 172–186. Springer, 2011.
- [4] NICKELS, K. – HUTCHINSON, S. Estimating uncertainty in SSD-based feature tracking. *Image and vision computing*. 2002, 20, 1, s. 47–58.
- [5] PARROT. *Parrot AR.Drone 2 specification* [online]. [cit. 8. 7. 2014]. Dostupné z: <<http://ardrone2.parrot-ardrone.cz/specifications/>>.
- [6] SKALKA, M. *Srovnání lokalizačních technik*. Diplomová práce, Univerzita Karlova v Praze, Matematicko-fyzikální fakulta, 2011.
- [7] STEPHANE, P. et al. *AR.Drone Developer Guide* [online]. 2014. [cit. 25. 6. 2014]. Revize SDK 2.0. Dostupné z: <https://projects.ardrone.org/attachments/download/514/ARDrone_SDK_2_0_1.tar.gz>.
- [8] STEWENIUS, H. – ENGELS, C. – NISTÉR, D. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2006, 60, 4, s. 284–294.
- [9] SUN, M. *Optical Flow* [online]. 2008. [cit. 8. 7. 2014]. Dostupné z: <<http://www.cs.princeton.edu/courses/archive/fall08/cos429/optiflow.pdf>>.
- [10] THRUN, S. et al. Robust Monte Carlo localization for mobile robots. *Artificial intelligence*. 2001, 128, 1, s. 99–141.
- [11] THRUN, S. – BURGARD, W. – FOX, D. *Probabilistic robotics*. MIT press, 2005. ISBN 978-0262201629.
- [12] TRAJKOVIĆ, MIROSLAV AND HEDLEY, MARK. Fast corner detection. *Image and vision computing*. 1998, 16, 2, s. 75–87.
- [13] WAN, E. A. – VAN DER MERWE, R. The unscented Kalman filter for non-linear estimation. V *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, s. 153–158. IEEE, 2000.
- [14] WELCH, G. – BISHOP, G. *An introduction to the Kalman filter*, 1995.

- [15] WIKIPEDIA. *PID controller* [online]. 2014. [cit. 7.7.2014]. Dostupné z: <http://en.wikipedia.org/w/index.php?title=PID_controller&oldid=613213646>.
- [16] WIKIPEDIA. *Parrot AR.Drone* [online]. 2014. [cit. 8.7.2014]. Dostupné z: <http://en.wikipedia.org/w/index.php?title=Parrot_AR.Drone&oldid=614465880>.
- [17] WIKIPEDIA. *Rotation matrix* [online]. 2014. [cit. 21.7.2014]. Dostupné z: <http://en.wikipedia.org/w/index.php?title=Rotation_matrix&oldid=617044306>.
- [18] WIKIPEDIE. *Rozšířená realita* [online]. 2014. [cit. 8.7.2014]. Dostupné z: <http://cs.wikipedia.org/w/index.php?title=Rozšířená_realita&oldid=11338957>.

Seznam použitých zkratek

- GPS – Globální poziční systém
- IMU – Inerciální jednotka. Senzor integrující gyroskop a akcelerometr a měřící zrychlení robota.
- SLAM – Simultánní lokalizace a mapování
- EKF – Rozšířený (extended) Kalmanův filtr
- HD – Vysoké rozlišení (High-definition)
- SDK – Vývojářské nástroje (Software development kit)
- IP – Internetový protokol

Přílohy

Součástí práce označenou jako „Příloha 1“ je CD obsahující

- elektronickou verzi práce
- zdrojové kódy implementované aplikace vč. některých nestandardních knihoven
- operační systém spustitelný přímo z CD s předinstalovanou aplikací