

**Jihočeská Univerzita v Českých Budějovicích**

**Přírodovědecká fakulta**



**Problematika poziční stabilizace multikoptéry na  
platformě Crazyflie 2.0**

**Bakalářská práce**

**Autor: Robert Princ**

**Vedoucí práce: PhDr. Milan Novák, Ph.D.**

**České Budějovice 2018**

## **Bibliografické údaje**

Princ, R., 2017: Problematika poziční stabilizace multikoptéry na platformě Crazyflie 2.0.[The issues of positional stabilization of the multicopter on the Crazyflie 2.0 platform. Bc.. Thesis, in Czech.] – 52, Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

## **Anotace**

Bakalářská práce se zabývá problematikou stabilizace multikoptéry Crazyflie 2.0 uvnitř uzavřeného prostoru. Problematika je popsána v práci teoreticky, kde jsou rozebrány možnosti a postupy pro stabilizaci a v praktické části demonstrován postup a testování vybrané stabilizace.

## **Klíčová slova**

Multikoptéra, Crazyflie 2.0, Poziční stabilizace, Flow desk, Barometr, Kalmanův filtr

## **Annotation**

Bachelor thesis deals with a problem of Crazyflie 2.0 multicopter stabilization inside enclosed space. The problem is described in theoretical section, where the possibilities and procedures for stabilization are analysed. In practical section of thesis, the procedure and testing of selected stabilizations are demonstrated.

## **Keywords**

Multicopter, Crazyflie 2.0, Position stabilization, Flow desk, Barometer, Kalman filter

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. V platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejich internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam s porovnáním textu mé kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne .....

Podpis autora .....

# Obsah

1	Úvod.....	1
2	Cíle práce .....	2
2.1	Hlavní cíle .....	2
2.2	Úkoly.....	2
3	Metodika práce .....	3
3.1	Výzkumné metody .....	3
3.2	Výzkumné problémy .....	3
4	Platforma Crazyflie 2.0.....	4
4.1	Specifikace .....	5
4.2	Crazyflie 2.0 systémová architektura .....	6
4.2.1	STM32F405 a interní-MCU komunikace.....	8
4.2.2	Syslink protokol.....	8
4.3	CrazyRadio PA.....	10
4.3.1	Specifikace.....	10
4.4	Crazyflie 2.0 sensory.....	11
4.4.1	MPU-9250 .....	11
4.4.2	Tříosý gyroskop se 16bitovým ADC .....	12
4.4.3	Tříosý akcelerometr se 16bitovým ADC .....	12
4.4.4	Tříosý magnetometr se 16bitovým ADC .....	12
4.4.5	Procesor digitálního pohybu .....	13
4.4.6	Vysoce precizní tlakový sensor (LPS25H).....	13
5	Technologie použité v práci.....	14
5.1	Crazyflie python knihovna .....	14
5.2	PyCharm.....	14

5.2.1	Crazyflie počítačový klient .....	14
6	Poziční stabilizace.....	15
6.1	Poziční stabilizace pomocí Loco pozičního systému.....	15
6.2	Stabilizace pomocí kamery a video transmitterem.....	16
6.3	Flow deck .....	17
6.3.1	VL53L0.....	17
6.3.2	PMW3901 .....	17
6.4	Výsledek analýzy pozičních systémů.....	18
7	Praktická část .....	19
7.1	Konstrukční stabilizace a sestavení multikoptéry .....	19
7.1.1	Kalibrace Vrtulí .....	21
7.1.2	Ochranné prvky.....	21
7.1.3	Použitý model a fáze modelace .....	22
7.2	Poziční stabilizace pomocí optického toku .....	24
7.2.1	Laserové měřicí rovnice.....	26
7.2.2	Adaptivní filtrace pro odstranění polohovaného posunu .....	27
7.2.3	Modely pro měření šumu.....	27
7.2.4	Kalmanův filtr.....	29
7.2.5	Kalmanova filtrace v Crazyflie 2.0.....	30
7.3	Základní testování letu .....	32
7.3.1	Instalace operačního systému ve VirtualBoxu.....	32
7.4	Tlakový snímač LPS25H .....	34
7.4.1	První spuštění a problematika čipu LPS25H .....	35
7.5	Testování letu za pomoci Flow Deck.....	36
7.5.1	Kód k použití optického sensoru na platformě .....	37
7.5.2	Python skript na držení hover módu v 50 cm .....	38

7.5.3	Python skript pro ovládání multikoptéry pomocí klávesnice .....	39
7.6	Barometr LPS25H x Optický tok.....	41
7.6.1	Praktické porovnání stabilizace .....	42
7.6.2	Porovnání s určenou dráhou letu .....	45
8	Závěr .....	46
9	Citace .....	47
10	Seznam použitých obrázků, tabulek a grafů .....	49
11	Přílohy.....	51

# 1 Úvod

Dnešní doba přináší velice rozšířené možnosti v oblasti robotiky a autonomie. Jedním z těchto témat jsou UAV z anglického výrazu Unmanned Aerial Vehicle a MAVs z anglického výrazu Micro Aerial Vehicles, v češtině nazvané jako mini multikoptéry nebo mikro multikoptéry.

Micro Aerial Vehicles (MAVs) jsou ideální platformy pro široký okruh možností použití v uzavřených i venkovních prostorech díky jejich malé velikosti, velmi dobré mobilitě, a jejich schopnosti se vznášet. V mnoha využití je potřeba, aby MAV byl schopen samostatného letu a tím minimalizoval zátěž operátora, který obsluhuje tento stroj. Díky Globálnímu pozičnímu systému (GPS), jsme si osvojili a dosáhli komerčního venkovního využití v mnoha odvětvích dnešního světa. Příkladem těchto odvětví je například: letecké fotografie, mapování a výměry pozemků, dále pak v dopravě a přepravě na které pracuje například Amazon a neposlední řadě inteligentní farmaření. Avšak, dále je zde veliký potenciál použití UAV/MAVs v oblasti dozoru, vyhledávání a záchrany v uzavřených a složitých vnitřních i venkovních prostředích, které jsou nepřístupné nebo nebezpečné pro lidi nebo pozemní vozidla.

V takových to případech při operování v prostředích je dostupnost signálu GPS, omezená nebo žádná. Infrastruktura, existence neznámých překážek v těsné blízkosti, může být dalším důvodem proč je potřeba nutnosti úplné autonomie s minimální lidskou interakcí.

Základem k dosažení plné autonomie je dosáhnout poziční stabilizace multikoptéry. Tato práce se zabývá poziční stabilizací platformy Crazyflie 2.0. za použití dostupných možností jako senzorů a doplňků pro tuto platformu.

## 2 Cíle práce

Cílem této práce je seznámit se s platformou Crazyflie 2.0 a jejími dalšími součásti. Ze zadání práce se potom soustředit na poziční stabilizaci v uzavřených prostorech. V teoretické části se práce bude zaměřovat na dostupné řešení, které již byla realizována. Budou porovnávány důležité prvky poziční stabilizace.

### 2.1 Hlavní cíle

V teoretické části práce se hlavními cíli práce stává:

- Systémová definice platformy Crazyflie 2.0
- Analýza a výzkum realizovaných řešení poziční stabilizace
- Zhodnocení dosavadních možností
- Vyvození závěrů
- Aplikace nejlepší dostupné poziční stabilizace

### 2.2 Úkoly

Úkoly při zpracování budou:

- Studium zdrojů a literatury
- Analýza pozičních systémů
- Analýza využití kalmanova filtru na platformě
- Rozebrání stabilizace za pomoci flow deck, barometru
- Porovnání realizací
- Návrh a realizace ovládacích prvků pro platformu



## **3 Metodika práce**

### **3.1 Výzkumné metody**

V teoretické části bude použita metoda obsahové analýzy. Studováním různých zdrojů a literatury. Ze závěrů vyvozených z analýz nám vyjdou závěry. Na základě těchto závěrů budeme postupovat při praktickém řešení realizace. Technologickým řešením dostaneme výsledek, který otestujeme a vyvodíme podle analýz odpovědi na výzkumné problémy.

### **3.2 Výzkumné problémy**

- Je na platformě Crazyflie 2.0 možnost lepší stabilizace?
- Je možné dosáhnout stabilizace bez GPS modulu?

## 4 Platforma Crazyflie 2.0

Platforma Crazyflie 2.0 je produkt, který vynalezli ve Švédsku a stále se podílí na jejím výzkumu tým, který v roce 2011 založil Bitcraze AB. Jejich cílem bylo zafinancovat a vyvíjet crazyflie desku na open source platformě. S pracemi na multikoptéře začali v roce 2009 jako část projektu Daedalus ve švédské konzultační firmě. Krátce poté co vydali první video na Hackaday.com se urychlil vývoj této platformy. Hlavními třemi zakladateli jsou Arnaud Taffanel, Marcus Eliasson a Tobias Antonsson.

Crazyflie 2.0 je celá postavená na open source platformě a díky tomu má mnoho možností k vývoji vlastních kódů, ale i součástek, které není problém připojit k základní desce. Jelikož se tato práce zabývá pozicováním v uzavřených prostorech je tato platforma ideální. Hlavními výhodami jsou možnosti rozšíření o tzv. desky. Tyto desky přinášejí mnoho výhod v různých oblastech. Jedná se o rozšíření typu SD desky, která slouží například k mapování letů, další deskou je například LED diodová deska. V této práci je však nejdůležitější rozšíření, a to flow deska. Tato deska obsahuje sensory na podporu stabilizace. Jsou zde ale i další desky, které jsou zatím v experimentálním vývoji.

Důvod proč se pro tuto práci využívá platforma Crazyflie 2.0 je také její velikost. Platforma spadá do úrovně Micro/nano koptér. To znamená, že díky svojí velikosti dokáže létat i ne tak velkých prostorách a členitějším terénu. Dalším důvodem je, že platforma je vybavená radiem, které v kombinaci s Crazyradio Pa ulehčuje její ovládání a je možné využít počítač. Toto spojení umožňuje testování, logování v reálném čase, grafy v reálném čase a nastavování multikoptéry při vzletu.

## 4.1 Specifikace

Funkce a její přednosti

- Robustní design
- Snadná montáž bez pájení
- Podporuje rozšiřující desky s automatickou detekcí
- Podporuje létání z iOS a Android pomocí Bluetooth LE, stejně jako z Windows / Mac OSX / Linux s Crazyradio nebo Crazyradio PA
- Testováno více než na 1 kilometrovou rádiovou vzdálenost (LOS) s Crazyradio PA
- Aktualizace lze provádět bez použití kabelů
- Vestavěné nabíjení pomocí standardního uUSB



Obrázek 4-1: Multikoptéra Crazyflie 2.0

### Mechanické specifikace

- 27g váha
- Velikost: 92x92x92mm

### Radiové

- 20 dBm rádio testovaný na více jak 1 km vzdálenost s Crazyradio PA
- Bluetooth nízká energetická podpora s iOS a Androidem

### Mikro kontroléry

- STM32F405 (Cortex-M4, 168 Mhz, 192kb SRAM, 1Mb flash)
- nRF51822 rádio a napájecí management
- LiPo nabíječ 100 mA, 500mA nebo 980mA
- Plno rychlostní USB rozhraní

### IMU

- 3 axis gyroskop (MPU-9250)
- 3 axis akcelerometr (MPU-9250)
- 3 axis magnetometr (MPU-9250)
- Tlakový sensor (LPS25H)

### Létající specifikace

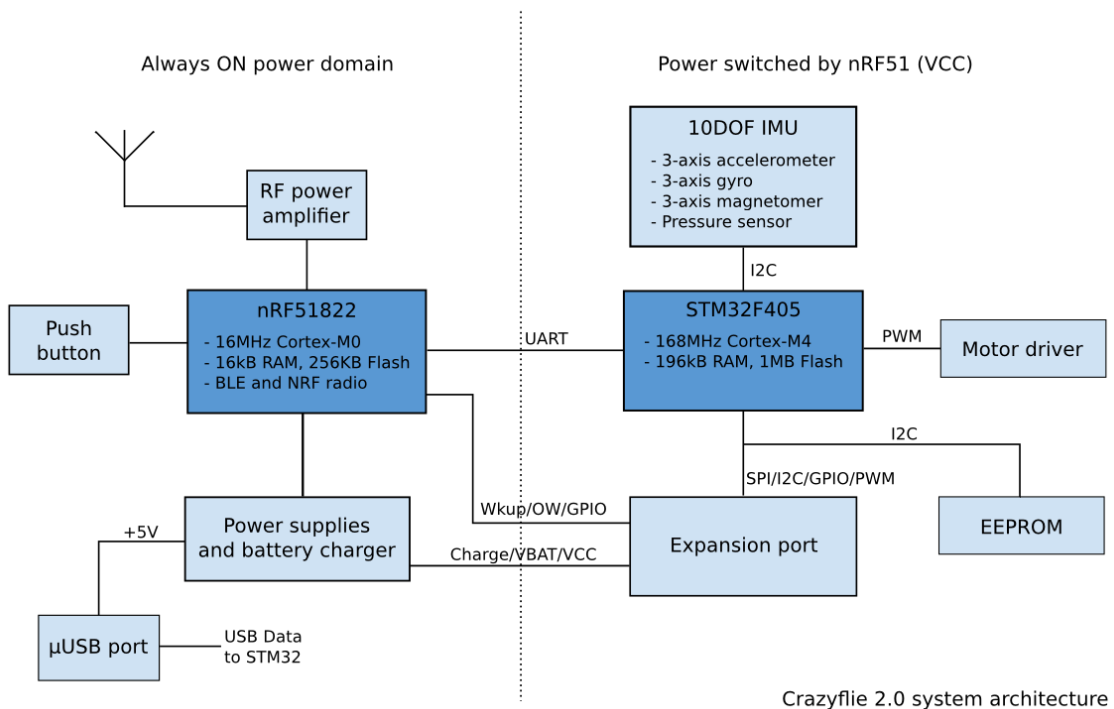
- Čas pro let: 7 minut
- Nabíjení 40 minut
- Maximální náklad: 15 g

## 4.2 Crazyflie 2.0 systémová architektura

Crazyflie 2.0 je postavená na dvou mikrokontrolerech. Jedním z nich je mikrokontroler NRF51, Cortex-M0, který se stará o radiokomunikaci a řízení spotřeby baterie. Má zabudovanou ON/OFF logiku, dále se stará o přístup k napájení v systému, a to konkrétně pro druhý mikrokontroler STM32, senzory a další přídatné desky. Nedílnou součástí je měření napětí a Master rádio bootloader. Obstarává také komunikaci radia a BLE, a

jelikož všechny přídatné desky jsou detekovány automaticky mikrokontroler NRF51 se stará o detekování těchto desek.

Druhým již výše zmíněným mikrokontrolerem je STM32F405, Cortex-M4@S160Mhz, který se stará a kontroluje letové vlastnosti multikoptéry. Mezi jeho práci patří čtení sensorů a následná kontrola a řízení motorů, dále letová kontrola, telemetrie (včetně napětí baterie) a dodatečný uživatelský vývoj. [1]



Obrázek 4-2. Systémová architektura [2]

## nRF51822

Dvěma hlavními úkoly pro tento mikrokontroler je navázání a udržování rádiové komunikace a distribuce napájení. Funguje jako rádiový most (komunikuje prostřednictvím raw data paketů do STM). Platforma Crazyflie 2.0 používá rádio pro Crazy RealTime Protokol (CRTP) i pro Bluetooth LE (BLE), ale hardware podporuje také další protokoly jako je například ANT. CRTP mód je kompatibilní s Crazyradio PA a dokáže tak navázat spojení o 2Mbit/s nízkou latencí. U technologie Bluetooth je minimální dosažitelná latence okolo 7,5ms, ale současná implementace se pohybuje

okolo 20ms. Hlavní výhodou CRTP je snadná implementace, pokud zařízení podporuje USB. Naopak Bluetooth LE se využívá většinou v případech řízení letu za použití mobilních zařízení.

Čip nRF51 je dále navržen tak aby mu jako zdroj napájení vystačila hodinková baterie, což znamená, že je velmi dobře zpracovaný pro nízko-energetické operace. Jelikož je čip odpovědný za řízení spotřeby, zvládá tzv. ON/OFF logiku. ON/OFF logika umožňuje různých akcích i po delším stisknutí tlačítka pro zapnutí. Příkladem je potržení tlačítka pro zapnutí na delší dobu se dostaneme do takzvaného bootloaderu.

#### 4.2.1 STM32F405 a interní-MCU komunikace

Na čipu STM32 běží hlavní software. I když je nastartován čipem NRF51, chová se jako nadřazený systém. Implementuje letovou kontrolu a všechnu komunikační vrstvu algoritmů. Rozšiřující port je připojený k STM32, a tak rozšiřující desky patří do STM také. Čip je vybavený 196kB RAM, což je dostatek paměti pro řízení letu. Díky velikosti paměti je k dispozici místo pro složitější algoritmy jako je například fúze vnitřních sensorů a GPS dat.

Komunikace mezi výše zmíněnými CPU probíhá v rámci syslink protokolu. Je to jednoduchý paketový protokol, který byl vytvořený pro rozšíření schématu komunikace.

#### 4.2.2 Syslink protokol

Protokol použitý pro komunikaci mezi STM32 a NRF51 v platformě Crazyflie 2.0. Zabývá se nízko úrovněnou komunikací na sériovém portu.



Obrázek 4-3: rámeček protokolu syslink

## Fyzický formát

- Start je dvoubytová konstanta, 0xBC 0xCF
  - Type definuje typ paketu
  - Length definuje délku paketu
  - CKSUM je kontrolní součet
- **Paketové typy**

Paketové typy jsou definované v *syslink.h* souboru a jsou organizovány do skupin pro směrování ve firmwaru.

Group	Name	Popis
0x00	RADIO	Rádiové pakety pro data a konfigurace
0x10	PM	Distribuce napájení a spotřeby
0x20	OW	Přístup k paměti pomocí kabelu

*Tabulka 4-1: Názvy a významy paketů*

- Rádiové pakety  
Paket RADIO\_RAW je typu 0x00 a formát je paket poslaný vzduchem. Dalším je RADIO\_CHANNEL typu 0x01 je pro navázání spojení v rámci kanálu, kde rozmezí od 2400Mhz do 2525Mhz. Dalšími pakety jsou například RADIO\_DATARATE a RADIO\_CONTWAVE.
- Distribuce napájení a spotřeby  
Tento typ obsahuje dva možné pakety a to SYSLINK\_OM\_ONOFF\_SWITCHOFF, používaný pro vypnutí systému a přechodu do režimu spánku. Druhým je SYSLINK\_PM\_BATTERY\_STATE, který se stará o baterii a její stav nabití. Tento paket je posílán stokrát za sekundu do mikrokontroleru STM32.

Syslink provádí veškerou komunikaci. STM32 je nadřazený a NRF51 je v módu slave. NRF51 je veden jako co nejjednodušší, aby druhý čip mohl provádět komplexní algoritmy.

## 4.3 CrazyRadio PA

CrazyRadio PA je rádiové zařízení vyvinuté stejnou firmou, která se zabývá vyvíjí platformu crazyflie 2.0. Toto zařízení funguje na bázi USB připojení, které dodává počítači možnost pracovat s rádiovou komunikací mezi počítačem a multikoptérou. Jeho použití lze aplikovat a adaptovat na jiné low-cost platformy. PA v názvu značí zesilovač, který poskytuje možnost rozsahu až na délku jednoho kilometru. [2]

### 4.3.1 Specifikace

nRF24LU1+ čip od Nordic Semiconductor

- 8051 MCU do 16MHz s 32Kb flash a 2Kb SRAM paměti
- 2.4GHz ISM rádiové pásmo
- USB zařízení
- 125 rádiových kanálů
- 2Mbps, 1Mbps and 250Kps komunikačních a přenosových dat

Rádiové specifikace:

- 20dBm výstupní výkon (100mW)
- Low Noise Amplifier (LNA)
- RP-SMA konektor



Obrázek 4-4: CrazyRadio PA



## 4.4 Crazyflie 2.0 sensory

Hlavním důvodem výběru této platformy byly již zabudované sensory, se kterými se již dá od počátku pracovat. Crazyflie 2.0 jako taková má tyto sensory celkem čtyři. Má tříosý



Obrázek 4-5: Čip MPU-9250

gyroskop, tříosí akcelerometr, tříosí magnetometr a vysoce přesný tlakový sensor. Tři z těchto sensorů má na starosti multi-čip MPU-9520.

### 4.4.1 MPU-9250

MPU-9250 je vícečipový modul, který se skládá ze dvou oddílů integrovaných do balíčku QFN (Quad Flat No-leads). První oddíl obsahuje tříosý gyroskop a tříosý akcelerometr. Druhý pak AK8963 tříosý magnetometr od firmy Asahi Kasei Microdevices Corporation. MPU-9250 je devíti osové zařízení, které dokáže monitorovat pohyb, stavěné na kombinaci tříosého gyroskopu, tříosého akcelerometru a tříosého magnetometru. Zároveň poskytuje digitální pohybový procesor. Vše v malém provedení 3x3x1 mm. MPU-9250 je také navržena pro propojení s více digitálními snímači, jako jsou snímače tlaku na pomocném I2C portu. Čip obsahuje tři 16bitové analogově digitální převodníky (ADC)

pro digitalizaci výstupů gyroskopu, tři 16bitové ADC pro digitalizaci výstupů a tři 16bitové ADC pro digitalizaci výstupů magnetometru. Pro přesné sledování jak rychlého, tak pomalého pohybu jsou vybaveny uživateli programovatelnými součásti gyroskopu v plném rozsahu  $\pm 250, \pm 500, \pm 1000$  a  $\pm 2000$  ° / s (dps), uživatelsky programovatelný akcelerometr rozsahu celé stupnice  $\pm 2$  g,  $\pm 4$  g,  $\pm 8$  g a  $\pm 16$  g a rozsah magnetometru v plném rozsahu  $\pm 4800\mu\text{T}$ . [3]

#### **4.4.2 Třiosý gyroskop se 16bitovým ADC**

MPU-9250 se skládá ze tří nezávislých vibračních gyroskopů MEMS (Mikro elektromechanické systémy), které detekují rotaci kolem os X, Y a Z. Když se gyroskop otáčí kolem nějaké osy, způsobí efekt Coriolisovy vibrace, která je detekována kapacitním odběrem. Výsledný signál se zesiluje, demoduluje a filtruje, aby bylo dosaženo napětí, které je úměrné úhlové rychlosti. Toto napětí je digitalizováno pomocí individuálního čipu 16bitové analogově-digitálního převodníku (ADC) pro vzorkování každé osy. Úplný rozsah gyroskopů se naprogramovat až na 2000 stupňů za sekundu (dps), jak bylo zmíněno výše. Vzorkovací frekvence analogově-digitálního převodníku je 8000 vzorků za sekundu, ale dá se naprogramovat až na pouhé 3,9 vzorků za sekundu.

#### **4.4.3 Třiosý akcelerometr se 16bitovým ADC**

Třípolohový akcelerometr používá pro každou osu samostatný výpočet. Zrychlení podél určité osy vyvolá posun, na který kapacitní snímače reagují rozdílně. Architektura MPU-9250 snižuje citlivost akcelerometrů k termálním driftům. Když je zařízení umístěno na rovný povrch, bude měřit hodnotu 0 g na ose X a Y a + 1 g na ose Z. Faktor měřítka akcelerometrů je kalibrován ve výrobním závodě a je nominálně nezávislá na napájecím napětí. Každý senzor má vyhrazenou sigma-delta ADC pro poskytování digitálních výstupů. Plný rozsah lze nastavit až na  $\pm 16$  g, jak bylo uvedeno v úvodní části.

#### **4.4.4 Třiosý magnetometr se 16bitovým ADC**

Třiosý magnetometr využívá velice citlivou technologii snímačů Hall. Magnetometrická část obsahuje magnetické senzory pro detekci zemského magnetismu v osách X-, Y-, a Z-, senzor řízení okruhu, řetězec zesilovače signálu a aritmetický okruh pro zpracování signálu z každého snímače.

#### **4.4.5 Procesor digitálního pohybu**

Vestavěný procesor DMP (Digital Motion Processor) je umístěn v rámci MPU-9250 a vypočítává algoritmy pro zpracování pohybu od hostitelského procesoru. DMP získává data z akcelerometrů, gyroskopů a magnetometrů a dalších sensorů třetích stran. Dále pak zpracovává data. Výsledné údaje mohou být čteny z registrů DMP nebo je lze uložit do vyrovnávací paměti FIFO. Typicky by měly být algoritmy pro zpracování pohybu prováděny rychlostí často kolem 200 Hz, aby se zajistili přesné výsledky s nízkou latencí. To je vyžadováno i v případě, že aplikace se aktualizuje mnohem nižší rychlostí. Například pro uživatelské rozhraní s malým výkonem se aktualizuje pomalu až k 5 Hz, ale zpracování by mělo pokračovat při 200 Hz. DMP lze použít jako nástroj k minimalizaci napájení, zjednodušení načasování, zjednodušení softwarové architektury a uložení cenných MIPS (Mikro Instruction Per Second) na hostitelském procesoru pro použití v aplikaci.

#### **4.4.6 Vysoce precizní tlakový sensor (LPS25H)**

Tento sensor je hojně využíván jako výškoměr a barometr pro přenosná zařízení. Využívají ho také GPS aplikace. Patří i mezi vybavení meteorologických stanic, a dokonce se vyskytuje i ve sportovních hodinkách. LPS25H je ultra kompaktní snímač tlaku. Zahrnuje monolitický snímací prvek a rozhraní IC, které má schopnost dokázat vzít naměřené informace ze snímání a poskytnout je jako digitální signál vnějšímu světu. Jeho snímací prvek se skládá ze závěsné membrány, která měří absolutní tlak. [4]

## 5 Technologie použité v práci

Crazyflie 2.0 je bez spojení s počítačem takřka neovladatelná. K tomu abychom mohli s platformou pracovat potřebujeme pár dostupných technologií. Pro tuto platformu máme několik možností, jak se k ní připojit. Jednou z možností je použít počítač. Tato práce se zabývá Crazyflie klientem pro Windows a dále pak VirtualBoxem, kde je vydáván systém běžící na linuxu přímo od Bitcraze firmy.

### 5.1 Crazyflie python knihovna

Crazyflie python knihovna je API, která se zabývá ovládáním a komunikací s platformou Crazyflie 2.0. Struktura knihovny je asynchronní a je založena na zpětných volání událostí. Funkce, jako je `open_link` se okamžitě vrátí a při otevřeném odkazu se vyvolá spojení zpětného volání. Knihovna neobsahuje žádné podprocesy nebo zámky, které by vedly běžící aplikace. To závisí přímo na aplikaci, která tuto knihovnu využívá.

### 5.2 PyCharm

Pro psaní skriptů, a spouštění zkušebních skriptů se v práci využívá program PyCharm. Pycharm je integrované vývojové prostředí (IDE), používané v počítačovém programování, specificky pro jazyk Python. Vyvíjen je českou společností JetBrains. Poskytuje analýzu kódu, grafický ladící program, integrovaný testovací jednotku, integraci se systémy řízení verzí a podporuje vývoj webových aplikací pomocí Django. Pycharm je platforma s verzemi pro Windows, MacOS a Linux.

#### 5.2.1 Crazyflie počítačový klient

Tento klient využívá knihovny Crazyflie. V této práci je využíván zejména pro testování stabilizace a letu platformy. Funguje také jako bootloader pro aktualizaci firmwaru v Crazyflie 2.0.

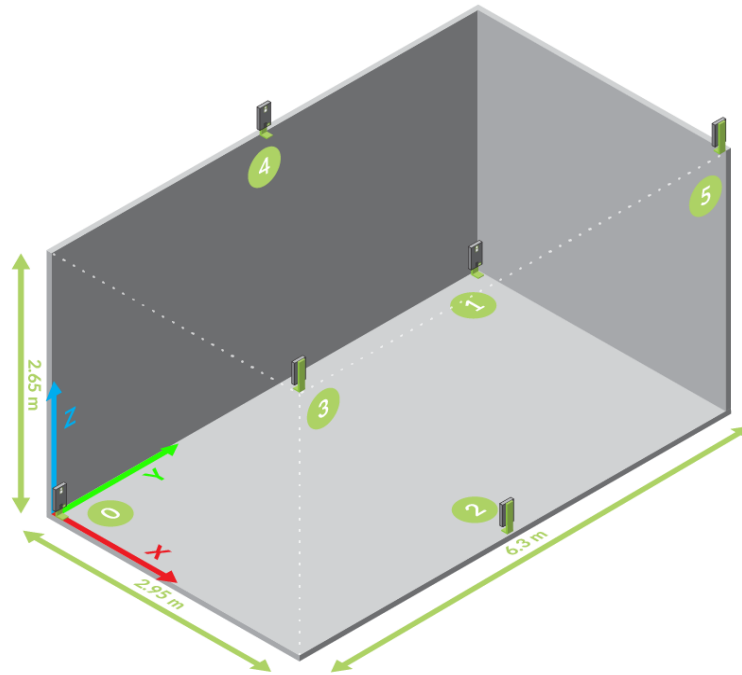
## 6 Poziční stabilizace

Poziční stabilizace multikoptéry na platformě Crazyflie 2.0 se dosáhne pomocí již výše zmíněných IMU (Inertial Measurement Unit). IMU je v zásadě jednotka, kterou obsahuje platforma pro orientaci v prostoru. Díky čipům, které na platformě jsou, lze změřit orientaci multikoptéry za pomoci úhlové rychlosti, síly, magnetického pole a tlaku. Tato práce se zabývá poziční stabilizací uvnitř prostor. Pro stabilizaci uvnitř prostor je bohužel toto vnitřní IMU multikoptéry nedostačující. Zejména kvůli rušení a výkyvům sensorů. Dosáhnout přesné poziční stabilizace v uzavřených prostorách, kde nelze využít například GPS, ani jiného globálního pozičního systému musíme využít fúze sensorů již dostupných (vnitřních) a dalších externích. Tato kapitola rozebere možnosti, které sensory a možnosti jsou dostupné z hledisek a kritérií pro tuto práci.

### 6.1 Poziční stabilizace pomocí Loco pozičního systému

Loco poziční systém je výsledkem vývoje týmu bitcraze.io. Díky tomuto pozičnímu systému můžeme dosáhnout absolutní poziční stability platformy Crazyflie 2.0. Platforma jako taková nemá žádnou skutečnou představu, kde v prostoru se v reálném čase nachází. Sensory na hlavní desce jsou používány k získání hrubé představy o jejích pohybech, ale postrádají přesnost pro dlouhodobé poznání polohy. Tento systém je jedním z řešení. Jedná se o externí systém, který poskytuje informace o aktuální pozici.

Loco poziční systém je lokální externí polohovací systém, který používá k nalezení absolutní 3D pozice objektů v prostoru. V mnoha směrech může připomínat miniaturní GPS systém. Základem systému je sada kotvících prvků umístěných v místnosti. Ty představují satelity v systému GPS. Druhou částí systému je přijímač, který je upevněn na objekt, v našem případě multikoptéru. Tento prvek umožňuje sledování objektu. Odesláním krátkých vysokofrekvenčních rádiových zpráv mezi kotvy a přijímačem systém měří vzdálenost od každé kotvy ke značkám a vypočítá tak pozici objektu, na kterém se nachází přijímač. Loco na rozdíl od jiných polohovacích systémů má informace potřebné k výpočtu pozice přímo na crazyflie 2.0. Tím se liší od ostatních systémů, které vypočítávají tuto pozici externě v počítači a pak ji teprve předávají platformě. [6]



Obrázek 6-1. Ukázka rozložení kotvících prvků

Obrázek 6-1 popisuje místnost s rozložením kotvících prvků, které následně fungují jako vysílače. Ty poté komunikují s platformou uvnitř této místnosti a předávají jí měřené hodnoty. Poté platforma upravuje svojí pozici podle předaných hodnot. Tímto je vyřešena stabilizace multikoptéry v prostoru a lze ji pak dále programovat trajektorii.

## 6.2 Stabilizace pomocí kamery a video transmitterem

Tuto možnost vytvořil a popsal v práci Oliver Dunkley, Jakob Engel a další. Stabilizace se zakládá na umístění analogové kamery s video transmitterem přímo na platformě Crazyflie 2.0. Autonomní let probíhá touto cestou využitím zabudovaných senzorů IMU a kamery, která zařizuje vizuálně-inerciální odhad stavu multikoptéry. Jednou z výhod této stabilizace je zabudování všech potřebných senzorů a zařízení do multikoptéry. Tím se minimalizuje náročnost na provedení uvnitř prostor. [5]

## 6.3 Flow deck

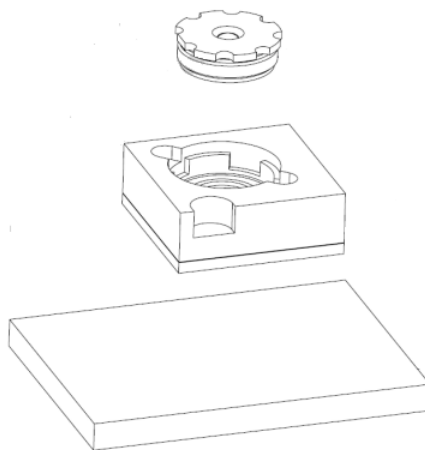
Flow Deck je rozšíření pro platformu CrazyFlie 2.0. Tato deska obsahuje senzor VL53L0, který měří vzdálenost k zemi s vysokou přesností. Dalším senzorem je senzor optického toku PMW3901, který měří pohyby ve vztahu k zemi. Spojením těchto dvou senzorů dosahuje platforma možnosti stabilizace v dané výšce a je možné programovat následný pohyb. [7]

### 6.3.1 VL53L0

VL53L0 je sensor pro přesné měření vzdálenosti bez ohledu na odrazivost materiálu. To znamená, že i velmi reflexivní materiály jako lino tomuto sensoru nevadí. Tento sensor může měřit absolutní vzdálenost do 2 metrů. Sensor integruje jedno-fotonové diody. VL53L0 je laserový emitor, je zcela neviditelný lidskému oku, s infračervenými filtry pro lepší odolnost vůči okolnímu světlu a lepší robustnost optického čtení. [8]

### 6.3.2 PMW3901

Optický snímač průtoku je vizuální snímač schopný měřit optický průtok nebo vizuální pohyb a výstup měřeného signálu na základě optického toku. Snímač optického průtoku PMW3901 je stejně jako sledovací snímač na typické myši, ale zde je určen pro rozsah od 80 mm do nekonečna. Snímač vyžaduje čočku (dodávaný s výstupem), který umožňuje sledování vzdáleného pole viz Obrázek 6-2.



Obrázek 6-2. Rozložení čipu PMW3901

## 6.4 Výsledek analýzy pozičních systémů

V celé šesté kapitole jsou popisované různé možnosti aplikací, jak dosáhnout poziční stabilizace bez GPS modulu. Z analýzy vyplývá, že každá uvedená možnost v kapitole 6. se zabývá stabilizací z jiného úhlu. Nejpreciznější možností poziční stabilizace je využití Loco pozičního systému. Tento a systémy Locu podobné se využívají například pro trajektorii multikoptér v hejnech. Důvodem je, že pro ovládání více multikoptér potřebujeme absolutní poziční stabilizaci, aby nedošlo k případnému střetnutí multikoptér. O tento systém projevila zájem i NASA, která by ho mohla využít ve vesmírném programu. Pro tuto práci je bohužel tento návrh stabilizace nedokonalý. Při autonomním létání v například zřícených dolech, či troskách po zemětřesení není možné předem umístit kotvící zařízení. Dalším důvod je cena. Pro uzavřený prostor je potřeba alespoň čtyř kotev, aby bylo dosaženo výsledku. Optimálně se ale využívá kotev šest. Při ceně jedné kotvy přesahující dva tisíce korun se pro autonomní let tento systém nehodí. Další možností je možnost stabilizace pomocí analogové kamery a bezdrátovým video transmitterem. Bohužel toto provedení má mnoho nevýhod. Největší nevýhodou je snížení času letu ze standartních 7.5 minut na pouhé tři minuty. Využitím kamery se zpoždění dostává kolem 40 ms, což je opravdu vysoké číslo, pokud platforma pomocí těchto informací má dělat výpočty stabilizace. Poslední nevýhodou tohoto provedení je veliký šum prováděný motory způsobující korupci odesílaných snímků kamery. Jediným kladem takového postupu je cena, která je pouze zlomek ceny Loco pozičního systému. Poslední je možnost řešení za použití optického toku. Tyto sensory, které poskytuje Flow Deck jsou pro tuto práci nejlépe vyhovující. Důvodem je že základní rozšiřující deska je hardwarem stejné firmy jako samotná platforma Crazyflie 2.0. Deska má minimální hmotnost 16 gramů a tím nezatěžuje multikoptér. To zvyšuje dobu letu. Dalším důvodem je preciznost těchto sensorů. Za použití kalmanovo filtru lze dosáhnout maximální přesnosti těchto sensorů. Jedná se o zlatou střední mezi poměrem cena/výkon.



## 7 Praktická část

Praktická část práce se věnuje platformě Crazyflie 2.0 jako takové. Jelikož se jedná o případ MAVs (Micro Aerial Vehicles) je nutné co nejlépe zkalibrovat platformu před samotným přidáním ostatních sensorů, v tomto případě flow decku. Důvodem je velikost platformy. Každá odchylka, která se nachází na rotorech, vrtulích, balanci baterie se promítá do stabilizace. Hlavní příčinou je šum, který se vytváří při těchto chybách a sensory pak měří výsledky zahrnující šum. Tyto další kapitoly se budou zabývat eliminací těchto chyb a dosažením optimálních výsledků. Následně na to probíhá testování hover modu a altitude modu za použití flow desky.

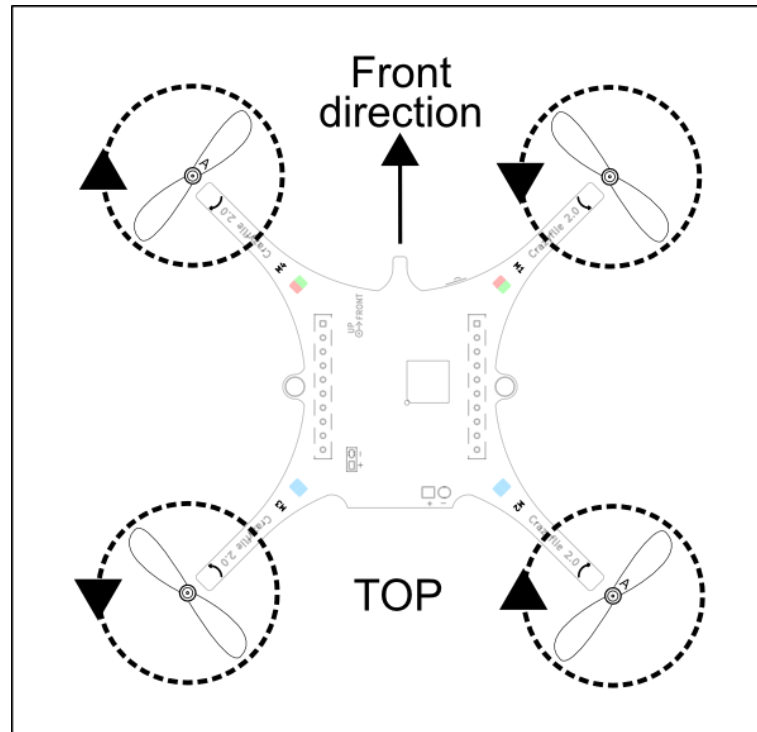
### 7.1 Konstrukční stabilizace a sestavení multikoptéry

Konstrukční stabilizací se práce zabývá v úvodu praktické části. Konstrukční stabilizací je myšlena výroba samotné platformy, kalibrace vrtulí, vyvážení ochranných prvků a odstranění nedostatků stabilizace vykazující chybovost v pozdější fázi. Následuje sestavení multikoptéry. Prvním krokem je takzvaný automatický test. Připojením desky pomocí uUSB portu k napájecímu zdroji se spustí tento test. Na desce se nachází čtyři LED diody, dvě modré, jedna zelená a jedna červená. Pokud zabliká zelená pětikrát rychle, tento test prošel. Tento test se dělá z důvodu, aby se zkontroloval stav desky po rozbalení z krabice. V nějakých případech může dojít k poškození převozem, extrémní teplotou nebo silným magnetickým polem. Význam LED diod:

- Zapnutí a vše proběhlo v pořádku: Modré LED diody jsou plně osvětleny a přední pravá LED dioda bliká červeně dvakrát za sekundu
- Režim pro nahrání firmwaru: Modré zadní LED diody svítí a poté se rozblikají přibližně jednou za sekundu.
- Selhání automatického testu: Pravá přední LED dioda opakovaně bliká v pěti krátkých impulzech s delší pauzou mezi skupinami

Dále přichází na řadu sestavení motorů. Každý motor má napájecí drát, který je potřeba zakroutit, aby se snížil elektronický šum. Poté se motor zasune do umělohmotného držáku, který slouží jako držák, podstava pro platformu a jako spojení s hlavní deskou.

Důležitým krokem je nasazení vrtulí na motory. Existují dva typy vrtulí. Vrtule, které se točí po směru hodinových ručiček a vrtule které se točí proti směru hodinových ručiček. Obrázek 7-1 vysvětluje postup nasazení. Důvod proč se vrtule nasazují tímto způsobem je podstata, proč multikoptéra může létat. Z fyzikálního hlediska je to díky proudu vzduchu, který má menší tlak než vzduch, který je stacionární.

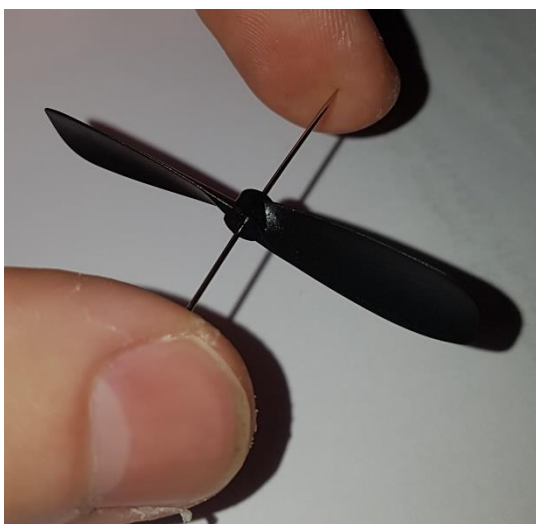


Obrázek 7-1: Směry vrtulí na platformě

V poslední řadě přichází na řadu umístění baterie. Baterie se umísťuje mezi držáky na horní straně desky. Poté je nasazen držák baterie na kolíky. V tomto kroku je potřeba dát si pozor na umístění baterie. Jelikož je multikoptéra takto malá, každý posun baterie o milimetry dopředu, či dozadu má za následky driftování a destabilizaci multikoptéry.

### 7.1.1 Kalibrace Vrtulí

Jedním z kroků, jak snížit vibrace multikotéry a šum senzorů je vylepšení stabilizace a balancování vrtulí. Při testování bez tohoto kroku se stává, že multikoptéra driftuje bez jakýkoliv důvodů, i když je baterie umístěná v pořádku. Multikoptéra pak při zatáčení



*Obrázek 7-2: Ukázka kalibrace vrtule*

vykazuje veliké odchylky. Po dlouhém hledání příčin bylo zjištěno nedostatečné kalibrování samotných vrtulí.

Postup, jakým se kalibrují vrtule je umístění vrtule na jehlu a následné držení vrtule v horizontální poloze viz Obrázek 4-1. Pokud se vrtule po puštění rozhoupe a dostane se do vertikální polohy, není kalibrovaná a znamená to, že jedna část vrtule, v tomto případě ta spodní je těžší než ta horní. Kalibrace probíhá tak že připevníme malý kousek lepenky na horní část vrtule a zkoušíme opět vertikální pozici. Takto se pokračuje do té doby, než vrtule po puštění v horizontální poloze nezůstane. Tehdy je vrtule kalibrována a vyvážena. Takto se postupuje u všech čtyř vrtulí.

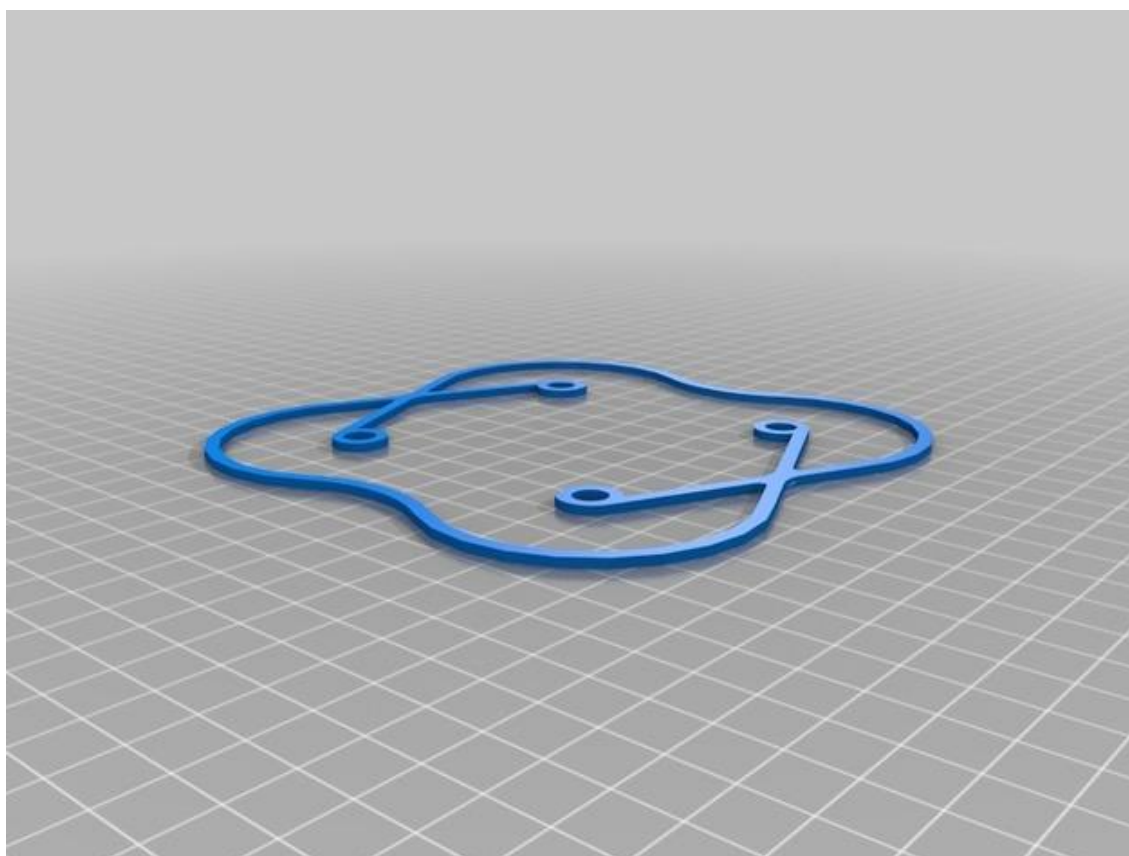
### 7.1.2 Ochranné prvky

V praktické části práce se věnuje kapitola ochranným prvkům platformy Crazyflie 2.0. Multikoptéra nemá žádný základní ochranný prvek, který by zabránil při pádu nebo při naražení do překážky, která se nachází ze strany multikotéry. Pád směrem dolů odlehčují držáky motorů, které mají výstupky pro přistání. Pro nárazy přední, zadní nebo bočních

stran však neexistuje žádný ochranný prvek. Jelikož při testování skriptů pro multikoptéru je možnost, že dojde k velkým odchylkám sensorů nebo driftům může snadno dojít k nárazu a poškození multikoptéry. V této práci se přišlo s řešením ochranných prvků pro multikoptéru na bázi 3D modelace a tisku na 3D tiskárně.

### 7.1.3 Použitý model a fáze modelace

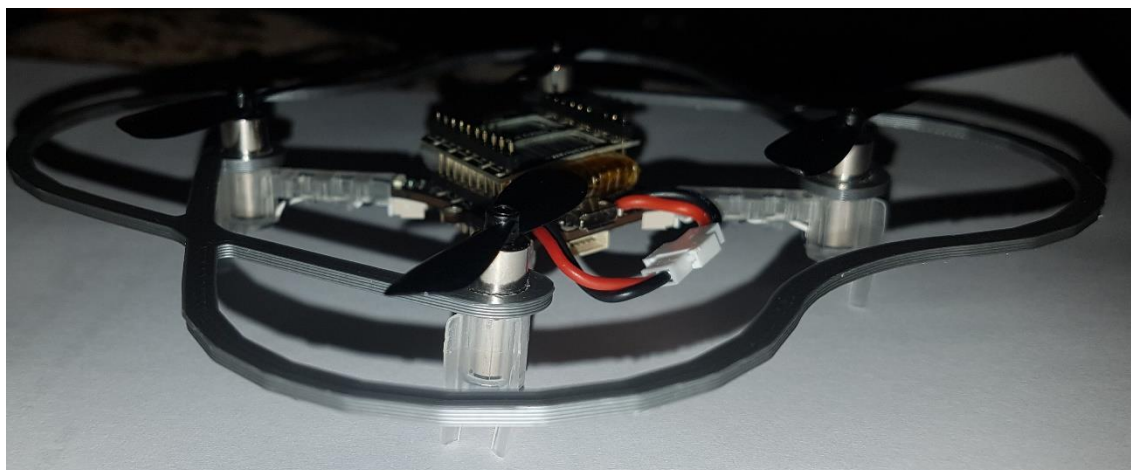
Model, který práce využívá byl inspirován webovou stránkou [www.thingiverse.com](http://www.thingiverse.com), kde se nacházejí modely připravené pro tisk na 3D tiskárně. Model, který byl vymodelován v Blendru, zabraňuje poškození vrtulí na vektorech X a Y, což znamená boční, přední a



Obrázek 7-3: Model ochranného prvku

zadní protekci viděno viz Obrázek 7-3. Montáž ochranného prvku na multikoptéru byla vyřešena tak, že upevňovací části se nasunou přímo na motory a tím eliminujeme lepení a máme možnost prvek opětovně sundat.

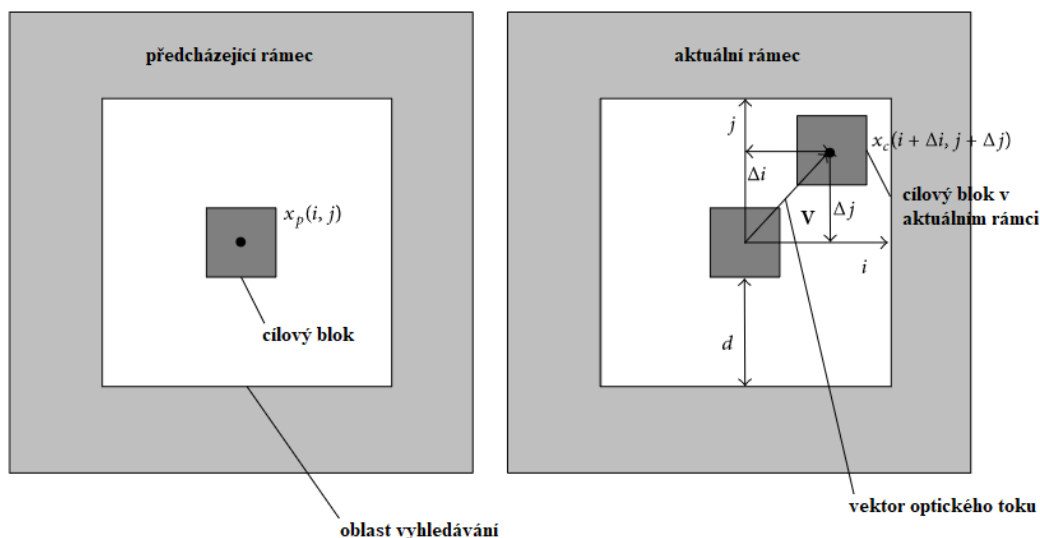
Tento model byl dále testován a přemodelován na užší konstrukci, aby nezatěžoval svojí vahou samotnou multikoptéru. Výsledný produkt byl vytištěn na 3D tiskárně, použitím tvrdšího filamentu. Obrázek 7-4 ukazuje ochranný prvek připevněn na platformu.



*Obrázek 7-4: Reálná fotografie s ochranným prvkem*

## 7.2 Poziční stabilizace pomocí optického toku

Optický tok je projevem 3D relativního pohybu na 2D rovinu obrazu. Pro výpočet optického toku se používá algoritmus pro porovnávání bloků. Jak je znázorněno viz Obrázek 7-5, kde nastavena jako šedá hodnota cílového bloku vybraného z předchozího rámce a je šedá hodnota cílového bloku, která má být porovnána ve vyhledávací oblasti aktuálního rámce.



Obrázek 7-5. Rámce zobrazující vektor optického toku

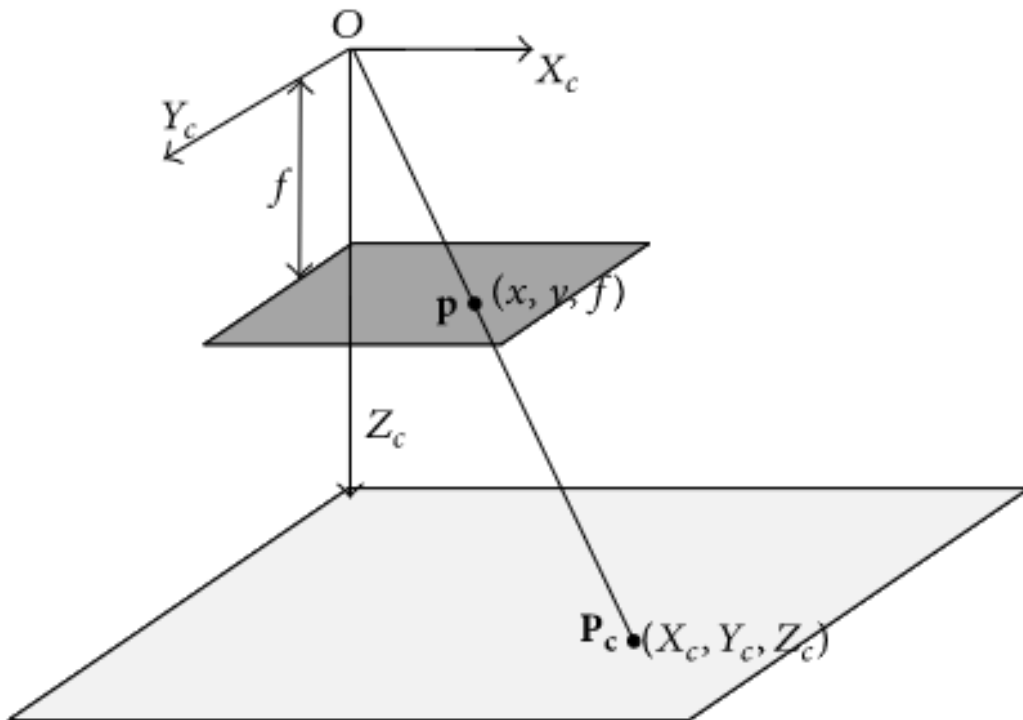
$$SAD(\Delta i, \Delta j) = \sum_{i=1}^n \sum_{j=1}^n |x_c(i + \Delta i, j + \Delta j) - x_p(i, j)|$$

V počátečním stavu je cíl vybrán v původním bodě zobrazovací roviny. Cílový blok se bude pohybovat, když se MAV přesune. Ve vyhledávací oblasti aktuálního rámce lze získat optický vektor proudu cílového bloku výpočtem minimálního SAD mezi aktuálním blokem a předchozím blokem. V experimentu jsou shromažďovány snímky, které jsou kolmé k čočce. Po získání snímku každého snímku se vypočítají střední absolutní chyby každého vektoru a minimální hodnota se volí jako vektor optického toku.

$$U = \min\{SAD(\Delta i, \Delta j)\}$$

$$V = r(\Delta i, \Delta j)^T|_U$$

Model pohybu optického toku promítá trojrozměrný pohyb na dvojrozměrnou rovinu obrazu. Existují dva společné modely optického odhadu průtoku: jeden je přístup v rovině obrazových čar, který je odvozen od principu vizuálního systému hmyzu a obratlovců; druhá je sférická zobrazovací plocha, která je odvozena od očí hmyzu. V naší studii je přístup k obrazové rovině pin-hole používán k odhadu pohybu MAV v geografickém souřadném systému.



Obrázek 7-6. Znáznorněný model pohybu optického toku

$\mathbf{P}_c = [X_c, Y_c, Z_c]^T$  obraz je nastaven jako bod pod souřadným systémem.  $f$  značí ohniskovou vzdálenost, takže  $\mathbf{P}_c$  může být zapsáno jako  $\mathbf{p} = [x, y, f]^T$ :

$$p = f \frac{P_c}{Z_c} \quad x = f \frac{X_c}{Z_c} \quad y = f \frac{Y_c}{Z_c}$$

Vezmeme-li v úvahu jakýkoli bod  $\mathbf{P}$  na zemi,  $\mathbf{P}$  má vztažný bod k MAV pod souřadnicovým systémem následující vztah:

$$V_c = -T_c - \omega \times P_c$$

Rovnice je následně rozšířena do tří rozměrů:

$$V_{xc} = -T_{xc} - (\omega_y Z_c - \omega_z Y_c)$$

$$V_{yc} = -T_{yc} - (\omega_z X_c - \omega_x Z_c)$$

$$V_{zc} = -T_{zc} - (\omega_x Y_c - \omega_y X_c)$$

Kde  $\omega = [\omega_x, \omega_y, \omega_z]^T$  je úhlová rychlost MAV a  $T = [T_x, T_y, T_z]^T$  je průměrná rychlost MAV pod souřadnicovou soustavou. Po derivačním počtu lze získat vztah mezi rychlostí pod souřadnicovým systémem a rychlostí pod obrazovou rovinou:

$$\frac{flow}{\Delta time} = v = f \frac{ZV_c - V_z P_c}{Z_c^2}$$

Kde  $v = [v_x, v_y, v_z]^T$  je rozšířením vzorce:

$$v_x = \frac{f}{Z_c^2} (V_{xc} Z_c - V_z Z_x)$$

$$v_y = \frac{f}{Z_c^2} (V_{yc} Z_c - V_z Z_y)$$

$$v_z = 0$$

A poté substitucí tří rozměrů na:

$$v_x = \frac{T_{zc} x - T_{xc} f}{Z_c} - \omega_y f + \omega_z y + \frac{\omega_x x y - \omega_y x^2}{f}$$

$$v_y = \frac{T_{zc} x - T_{yc} f}{Z_c} - \omega_x f + \omega_x y + \frac{\omega_x x y - \omega_y x^2}{f}$$

V substituovaných vzorcích  $v_x$  a  $v_y$  jsou optické komponenty toku na směrech a které lze vypočítat pomocí SAD a BMA.  $Z_c$  lze získat laserovým měřičem, který byl integrován do optického snímače průtoku.

### 7.2.1 Laserové měřicí rovnice

Pro zjištění výšky  $z$  musíme vzít měření laserového sensoru  $\alpha = \arccos[(Rz_f) \cdot z_f]$ .

Pokud je laserové zorné pole  $\theta_{pz} \rightarrow 0$ , následující měření bude:

$$z = h z_\beta \cdot z_f = h (Rz_f) \cdot z_f \Leftrightarrow h = \frac{z}{(Rz_f) \cdot z_f} = \frac{z}{\cos(\alpha)}$$

A však pokud je zorné pole výrazně větší, než nula jako je například  $\theta_{pz} \rightarrow 25^\circ$  uvedené ve specifikacích v1530x, je zřejmé, že  $h = z$ , pokud  $\alpha = \theta_{pz}/2$ . Když vektor  $z$  není



obsažen v laserovém kuželu vyzařování, měřená vzdálenost  $h$  bude korespondovat k průměru měření nejbližší k ose  $z$  takto:

$$h^{row}(x, 0) = h = \begin{cases} \frac{z}{z} & \text{if } |\alpha| < \theta_{pz}/2 \\ \frac{z}{\cos(|\alpha| - \theta_{pz}/2)} & \text{if } |\alpha| \geq \theta_{pz}/2 \end{cases}$$

Determinování měření Jacobian je triviální, pokud nezohledňuje změny v  $R$ . Například v Tait-Bryan reprezentaci  $(Rz_f) \cdot z_f = \cos(\varphi)\cos(\theta)$ , ukazuje pouze možnou opravu kovariance s respektováním výšky a naklání, zatímco otáčení zůstane nepozorovatelné. Tak, že rotace odpovídající k náklonu a výšce je vyčtena z informací gyroskopu a akcelerometru, změny v  $R$  jsou vynechány při měření Jacobian.

### 7.2.2 Adaptivní filtrace pro odstranění polohovaného posunu

Problém pro odhad optického toku vychází z odhadu nepozorovatelných stavů, které nebudou konvergovat na žádné smysluplné řešení. V případě optického toku se toto týká pozičních stavů, kde je vnitřní globální odhad  $(x, y)$  odchýlen od pravé globální pozice  $(x_t, y_t)$ , v čase, který  $(\Delta x, \Delta y) = (x - x_t, y - y_t) \neq 0$ . Nicméně pokud je tento drift dostatečně malý tak, že během životnosti baterie  $T[s]$ , existuje zde kruhová vazba pro konstantní  $C$ . Díky tomu si můžeme být jistí, že na konci každého letu můžeme kontrolovat multikoptéru do bodu  $(x_r, y_r)$  v rádiu kruhu  $C$  okolo zamýšleného bodu.

### 7.2.3 Modely pro měření šumu

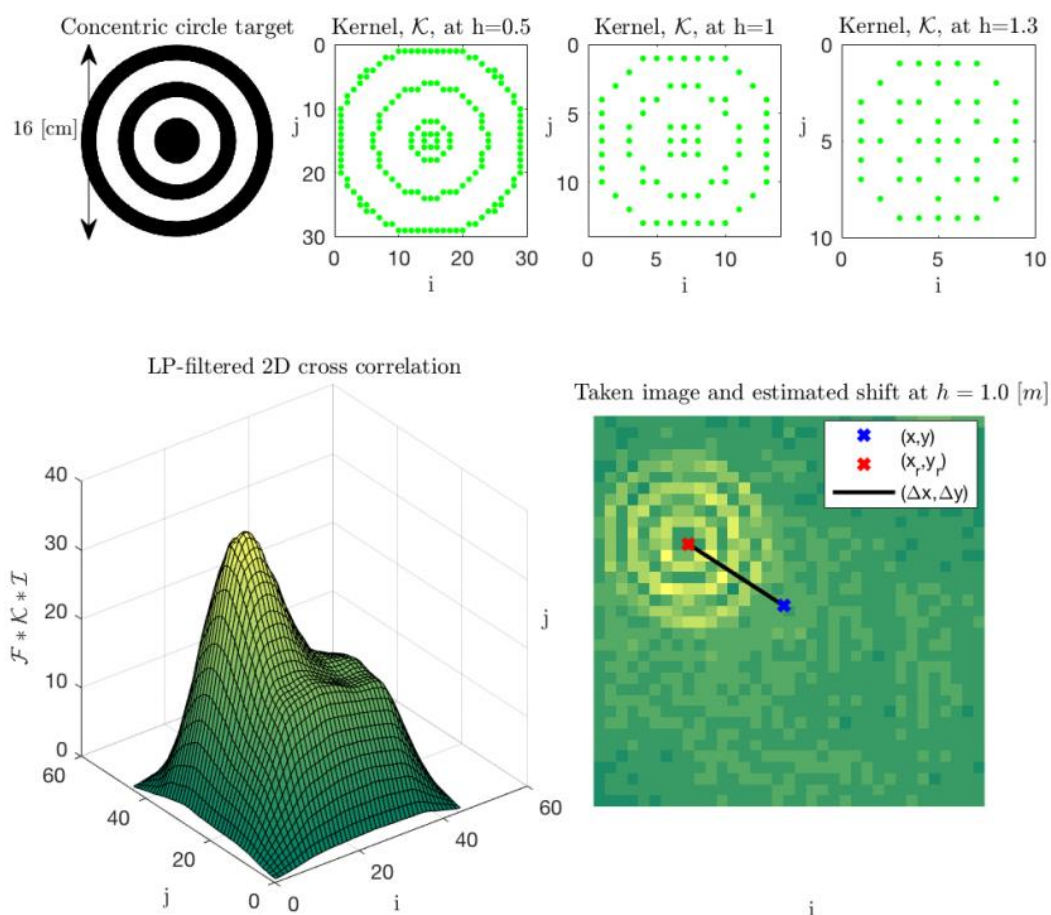
V případě laserového měření byly experimentálně ověřeny měřené šumy jako nulové průměry a normálně distribuovány. Variance šumů závisí na světelných podmínkách (hlavně v infračerveném spektru), reflexních prvcích povrchu a vzdálenosti od něj. Naproti tomu měření šumu optického toku je obtížnější měřit a modelovat. Některé informace mohou být odvozeny z optického sensoru, který poskytuje faktor šedi a faktor kvalitu obrazu, kde je nízký faktor v stupních šedi a vysoká kvalita obrazu indikuje menší standardní odchylku měření šumu. Kromě toho snímač průtoku poskytuje informace o hodnotě uzávěry, která udává světelnost prostředí, které je relevantní pro měření rozsahu laseru. Model pro laserové měření za použití dat z optického sensoru:

$$\mathbf{h}^{row}(\mathbf{0}, \mathbf{v}) = \left( 1 + \alpha_h \frac{1 + Q_s + h - Q_i}{Q_s^{max} + h^{max} + Q_i^{max}} \right) v_h(t) = V_k^{row} v_h(t)$$

Kde  $V_k^{row}$  je diagonální element měření šumu pro odpovídající řádek v Jacobiho matici, v lineární rovnici  $v_h(t)$ . V optickém toku řešíme pouze faktor obrazu, kde je diagonální prvek měření šumu za pomoci Jacobiho matice lehce identifikovaný: [13]

$$\begin{bmatrix} h^{row}(0, v) \\ h^{row+1}(0, v) \end{bmatrix} = \left(1 + \alpha_{\Delta n} \frac{Q_g}{Q_g^{max}}\right) \begin{bmatrix} v_{\Delta x}(t) \\ v_{\Delta y}(t) \end{bmatrix}$$

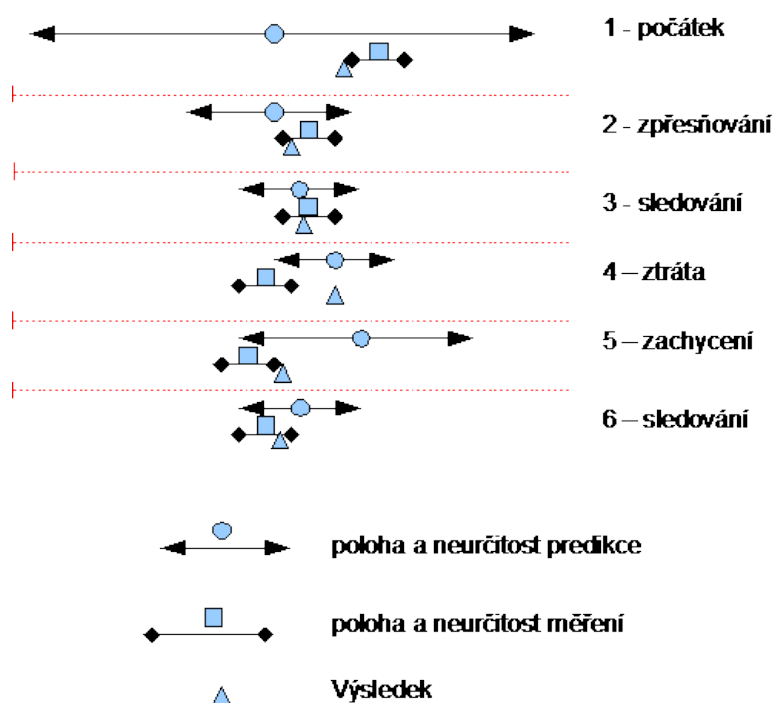
Obrázek 7-7 je koncentrický kruhový cíl a filtrační jádra  $\mathbf{K}$  na různých místech nadmořské výšky  $h$ . Dolní levý obrázek je LP filtrovaná křížová kovariance pořizovaného snímku a filtrační jádro,  $\mathbf{F} * \mathbf{K} * \mathbf{I}$ . Pravá dolní část obrázku je snímek s odhadovaným středem, soustředný kruhový prvek z výšky  $h = 1,0$  s odpovídajícím posunem polohy, jak je vypočítán ve firmware UAV, což způsobuje chybu jediného pixel, odpovídající přesnosti  $\pm 4$  [cm]. [13]



Obrázek 7-7. Model měření optického šumu [13]

## 7.2.4 Kalmanův filtr

Tato práce se zabývá kalmanovo filtrací signálů, které přichází ze sensorů. Kalmanova filtrace je speciální matematický algoritmus pro filtraci signálů v časové oblasti. Výhodou tohoto systému je schopnost získat určitý signál a hodnoty ze zašuměného signálu nebo jinak znehodnoceného souboru hodnot, i bez jakéhokoliv poznatku o rušení. Prakticky lze takto zjistit hodnoty, které jsou přímým měřením těžko zjistitelné, protože při samotném aktu měření do získaných hodnot indukují chyby měřících přístrojů nebo okolní působící šum a rušení. Ve zjednodušené podobě jde o predikčně-estimační algoritmus, který z předchozích, a i následných budoucích dat snaží předpovědět průběh signálu, tedy vždy z průběhu minulých vzorků signálu vzorky následující, např. neznámí čistý signál bez rušení jen z měřeného. Tyto jsou pak konfrontovány s opravdu naměřenými vzorky a jejich rozdíl se opět využívá ke zlepšení odhadu dalších budoucích vzorků. [10] Obrázek 7-8. znázorňuje obecný postup kalmanova filtru.



Obrázek 7-8. Teoretické znázornění postupu výpočtu KF

**Počátek:** určíme nejlepší předpokládanou polohu (predikce polohy) a její neurčitost stanovíme podle toho, jak jí věříme. Jak je vidět viz Obrázek 7-8. Predikci se nevěří,

jelikož je velká. Následně v rozmezí neurčitosti hledáme (měříme) aktuální polohu. Ta má neurčitost danou kvalitou měření. Výsledná poloha se určí jako kombinace predikce a měření.

**Zpřesňování:** na základě minulých hodnot a stanoveného modelu se vypočte predikce. Neurčitost je dána vývojem minulé neurčitosti podle daného modelu (a tedy se měřením zpřesní, ale vlivem neznámého/nekontrolovaného „vývoje“ opět vzroste. Proběhne měření a výpočet nové výsledné polohy.

**Sledování:** predikce se při delším sledování zpřesňuje (váhy měření a predikce se sbližují, filtrace („vyhlazení“) průběhu.

**Ztráta:** v případě, kdy nedojde k měření není možné korigovat polohu, a proto nedojde ani k jejímu zpřesnění. Dochází tedy k nárůstu neurčitosti predikce výsledku, což se projeví i v dalším kroku. Ke ztrátě dochází při nepředvídané změně polohy (mimo popis modelem a jeho parametry), nebo při překrytí nebo splynutí objektů vedoucímu k tomu, že nedojde k měření.

**Zachycení** – jelikož se při ztrátě zvětší neurčitost predikce, narůstá i okruh, ve kterém hledáme změřenou hodnotu. Po jednom nebo více krocích dojde k zachycení, a na jeho základě i ke zpřesnění polohy.

Kalmanův filtr může být napsán jako jediná rovnice, avšak nejčastěji se koncipuje jako dvě odlišné fáze: "Predikovat" a "Aktualizovat". Fáze předpovědi používá odhad stavu z předešlého časového rozvrhu pro vytvoření odhadu stavu v aktuálním časovém rozvrhu. [11,12]

### 7.2.5 Kalmanova filtrace v Crazyflie 2.0

Kalmanova filtrace se na platformě nachází přímo ve firmwaru Crazyflie 2.0. Přesná cesta je `crazyflie-firmware/src/modules/src/estimator_kalman.c`. Při psaní skriptu potom voláme tento modul. V modulu je popsána a naprogramována filtrace pro akcelerometr s gyroskopem i pro optický sensor, který pomáhá při vyrovnávání multikoptéry.

EKF (Extended Kalman Filter) se používá na platformě k fúzi akcelerometru a optického toku s laserovým měřením. Rychlost a poloha vypočítaná pomocí akcelerometru jsou vybrána jako stavová hodnota a rychlost a poloha vypočítaná

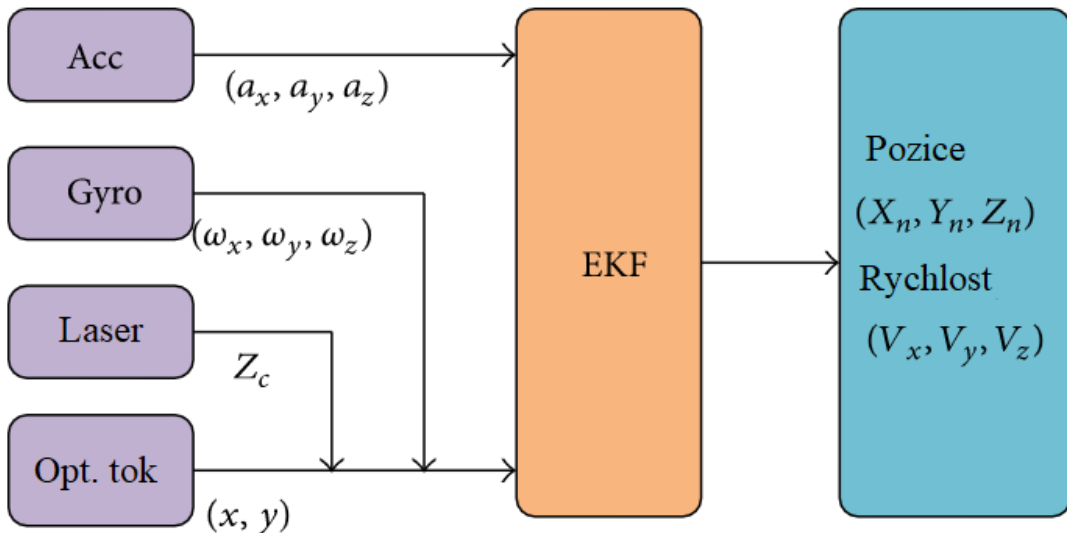
optickým senzorem průtoku jsou zvolena jako hodnoty pozorování. Proces odhadu je uveden viz Obrázek 7-9.

Vzhledem k nelineární stavové rovnici systému a pozorované rovnici:

$$X_{k+1} = f[X_k, k] + B_k U_k + T_k W_k$$

$$Z_{k+1} = h[X_{k+1}, k + 1] + V_{k+1}$$

Kde  $X_k$  je vektor stavu, který obsahuje informace o rychlosti a informaci o pozici MAV.  $Z_k$  je vektor pozorování, který obsahuje informace z měření optickým a laserovým senzorem.  $U_k$  je kontrolující vektor systému který nabyde hodnoty po transformaci dat akcelerometru v souřadnicové matici.  $B_k$  je matice kontroly.  $T_k$  je maticí šumu a  $W_k$  maticí procesních šumů.  $V_k$  je šum pozorování.  $f$  reprezentuje systémovou stavovou funkci a  $h$  reprezentující funkci pozorování.



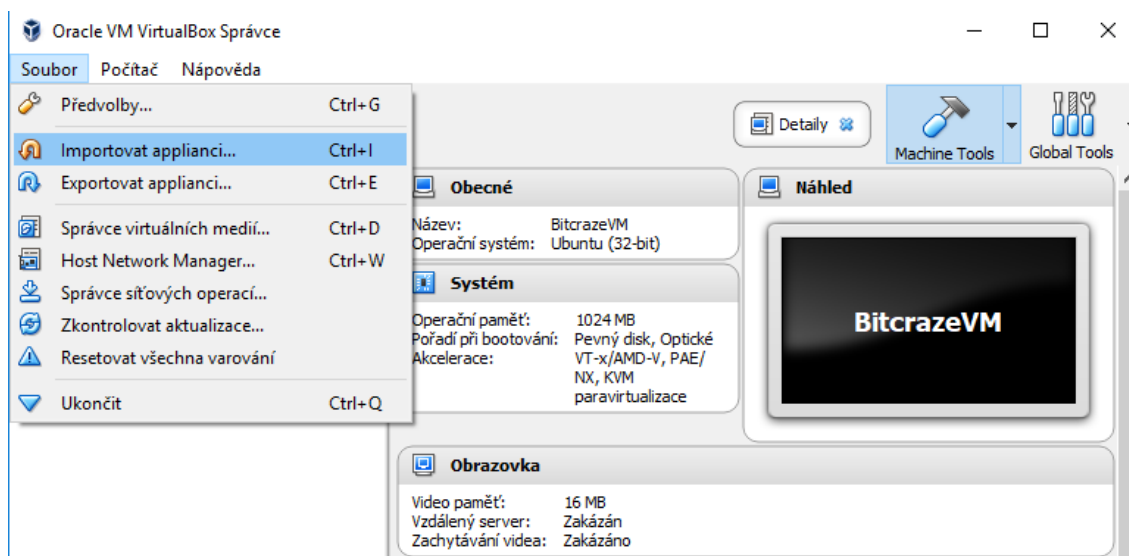
Obrázek 7-9. Proces odhadu EKF

## 7.3 Základní testování letu

K základnímu testování letu se v této práci přistupuje ve třech bodech. Nainstalováním operačního systému Ubuntu na virtuálním počítači za použití Crazyflie klienta, Crazyflie rádia a Xbox one ovladače. Práce se věnuje testování letu bez jakéhokoliv senzorického pozicování. Druhým krokem je testování stabilizace za pomoci senzoru LPS25H, tedy barometru. Třetím krokem bude základní testování platformy se senzorem optického toku.

### 7.3.1 Instalace operačního systému ve VirtualBoxu

Pro základní testování se v této práci přistupuje pomocí systému navrženým firmou Bitcraze. Tento operační systém obsahuje základní součásti, které vyhovují základním testům letu a stabilizace. VirtualBox je výkonný produkt pro virtualizaci x86 a AMD64 / Intel64. Je volně k dispozici jako Open Source Software za podmínek GNU General Public License (GPL2). V současné době VirtualBox běží na systémech Windows, Linux, Macintosh a Solaris a podporuje velké množství hostujících operačních systémů včetně Windows NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8 a Windows 10). VirtualBox se aktivně rozvíjí s častými verzemi a má stále rostoucí seznam funkcí, podporovaných hostovacích operačních systémů a platforem, na kterých běží. [9]



Obrázek 7-10: Import ve VirtualBox

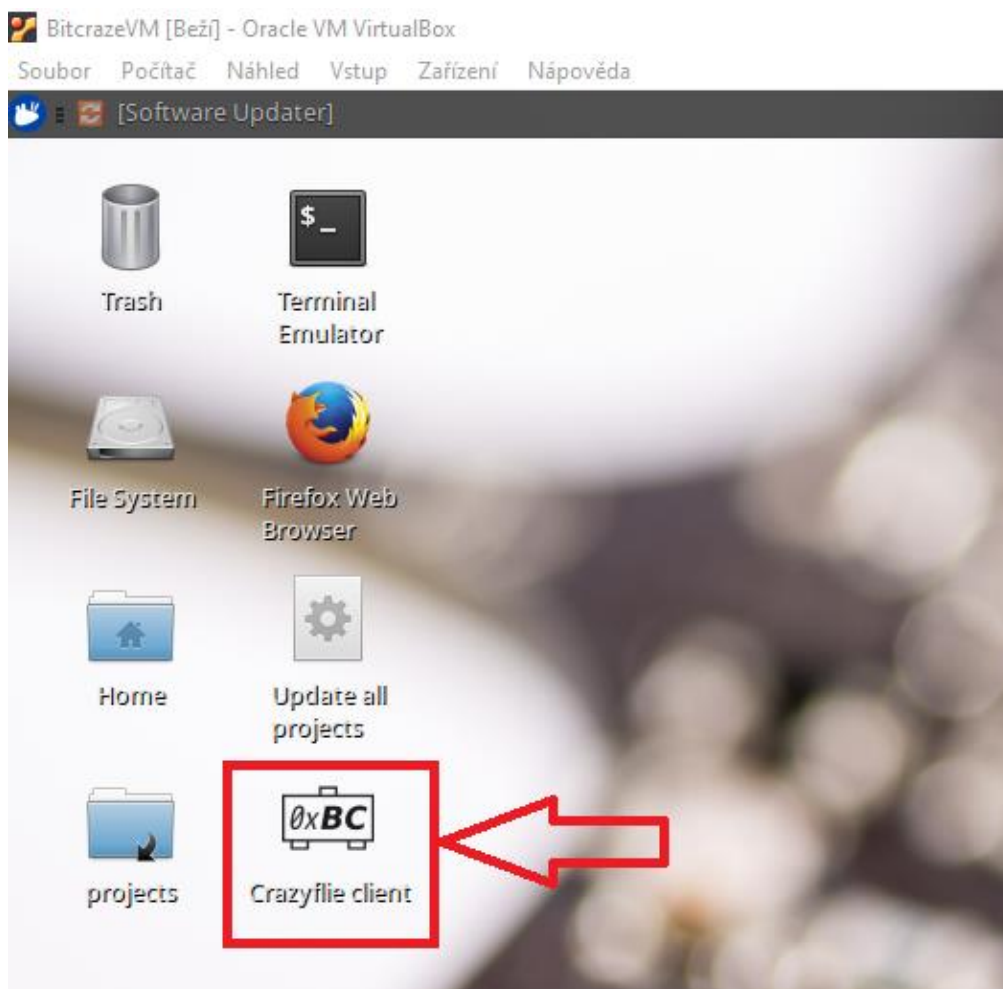
Webová stránka: <https://www.virtualbox.org/>

Ke stažení: <https://www.virtualbox.org/wiki/Downloads/>

Operační systém: cross-platform

Licence: GPL2

Po nainstalování VirtualBoxu potřebujeme operační systém. Ten do VirtualBoxu jednoduše naimportujeme viz Obrázek 7-10. Po spuštění operačního systému je na ploše Crazyflie klient, který je možné spustit, jak je vidět viz Obrázek 7-11.



Obrázek 7-11: Spuštění Crazyflie klienta

## 7.4 Tlakový snímač LPS25H

### Převedení tlaku na nadmořskou výšku

Převedení tlaku na nadmořskou výšku je užitečné pro letadla, rakety, satelity, balóny počasí a mnoho dalších aplikací. Všechny tyto aplikace využívají vztah mezi změnami tlaku ve vztahu k nadmořské výšce. Tento vztah se řídí následující rovnicí:

$$h = \left(1 - \left(\frac{P}{P_{ref}}\right)^{0.190284}\right) \times 145366.45 \text{ ft}$$

Tato rovnice je kalibrována pro výškoměr až do 11 000 m. Mimo tohoto rozsahu se zavede chyba, kterou lze vypočítat odlišně pro každý snímač tlaku. Tyto chybné výpočty budou zohledňovat chybu způsobenou změnou teploty. Snímače barometrického tlaku mohou mít rozlišení nadmořské výšky menší než 1 metr, což je výrazně lepší než systémy GPS (přibližně 20 metrů nadmořská výška). Jedná se o nejjednodušší a nejpoužívanější vzorec barometru (nadmořské výšky). Příklad pseudo-zdrojového kódu viz Obrázek 7-12.

```
void From_Pressure_mb_To_Altitude (double* Pressure_mb,
double* Altitude_ft) {
*Altitude_ft = (1-
pow(*Pressure_mb/1013.25,0.190284))*145366.45; }

void From_ft_To_m(double* ft, double* m) {
*m = *ft/3.280839895;
```

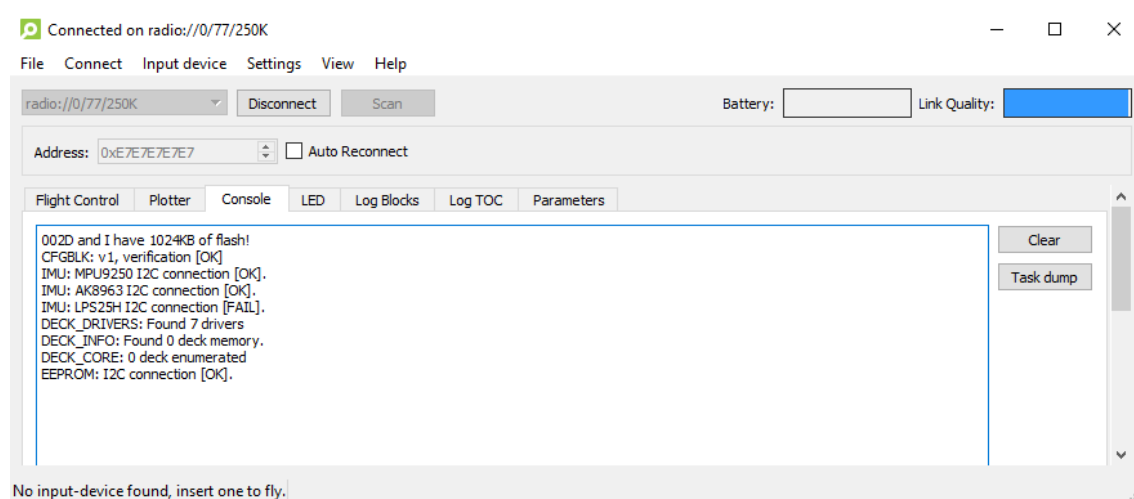
Obrázek 7-12. Pseudo kód pro výpočet nadmořské výšky



### 7.4.1 První spuštění a problematika čipu LPS25H

První spuštění bylo prováděno v rámci python klienta ve VirtualBoxu. Důvodem využití VirtualBoxu pro první spuštění a testování je fakt, že Bitcraze.io tým udělal virtuální operační systém, který obsahuje všechny dostupné kódy, klienty a knihovny. Po zapnutí klienta a nastavení driverů pro CrazyRadio se pokračovalo vyhledáním a připojením multikoptéry. V tomto kroku proběhla aktualizace firmwaru platformy a testování. Aktualizací firmwaru se zobrazila v klientovi možnost altitude módu, který má na starosti právě čip LPS25H. První vzlet se uskutečnil bez altitude módu a byl úspěšný. V konzoli při navazování spojení proběhlo vše v pořádku. Když ale mělo dojít po dobití baterie k druhému testu nastal problém. Konzole při připojení platformy začala vyhazovat selhání konektivity čipu LPS25H viz Obrázek 7-13. Po opětovném přehrání firmwaru se nic nezměnilo. S platformou se dalo létat, ale altitude mód nebyl k dispozici.

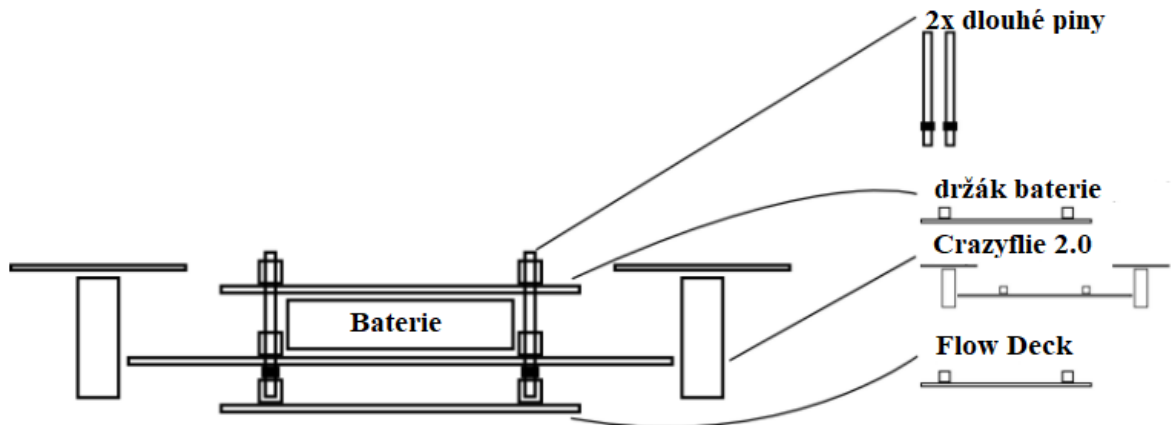
Řešením tohoto problému se zabývalo oficiální fórum řešící podobnou problematiku. Důvodem tohoto selhání bylo nesprávné měření a kontrola na výrobních linkách. V početné vydané edici platforem se nacházel vadně pájený čip LPS25H, který po několika málo vzlétnutí vykazoval poruchovost. Tento problém byl v práci vyřešen kontaktováním developera platformy a následná reklamace.



Obrázek 7-13. Error způsobený výrobní vadou

## 7.5 Testování letu za pomoci Flow Deck

V této části práce se bude probírat nastavování a zprovoznění flow deck rozšíření na platformě. Na platformu se Flow deck montuje ze spodní části, kde se uchytlí pomocí dvou dlouhých pinů, jak je vidět na obecném schéma viz Obrázek 7-15 a Obrázek 7-14.



Obrázek 7-15: Umístění rozšíření



Obrázek 7-14. Fotografie multikoptéry s flow desk

## 7.5.1 Kód k použití optického sensoru na platformě

```
static void pamotionTask(void *param)
{
    systemWaitStart();

    while(1) {
        vTaskDelay(10);
        readMotion(&currentMotion);
        int16_t accpx = -currentMotion.deltaY;
        int16_t accpy = -currentMotion.deltaX;
        if (abs(accpx) < OULIER_LIMIT && abs(accpy) < OULIER_LIMIT) {

            flowMeasurement_t flowData;
            flowData.stdDevX = 0.25;
            flowData.stdDevY = 0.25;
            flowData.dt = 0.01;

#ifdef USE_MA_SMOOTHING
            pixelAverages.averageX[pixelAverages.ptr] = (float32_t)accpx;
            pixelAverages.averageY[pixelAverages.ptr] = (float32_t)accpy;
            float32_t meanX;
            float32_t meanY;

            arm_mean_f32(pixelAverages.averageX, AVERAGE_HISTORY_LENGTH,
&meanX);
            arm_mean_f32(pixelAverages.averageY, AVERAGE_HISTORY_LENGTH,
&meanY);

            pixelAverages.ptr = (pixelAverages.ptr + 1) %
AVERAGE_HISTORY_LENGTH;

            flowData.dpixelx = (float)meanX; // [pixels]
            flowData.dpixely = (float)meanY; // [pixels]
#endif
#ifdef USE_LP_FILTER
            // Use LP filter measurements
            flowData.dpixelx = LP_CONSTANT * dpixelx_previous + (1.0f -
LP_CONSTANT) * (float)accpx;
            flowData.dpixely = LP_CONSTANT * dpixely_previous + (1.0f -
LP_CONSTANT) * (float)accpy;
            dpixelx_previous = flowData.dpixelx;
            dpixely_previous = flowData.dpixely;
#else
            flowData.dpixelx = (float)accpx;
            flowData.dpixely = (float)accpy;
#endif
        }
        // Push measurements into the Kalman filter
        if (!useFlowDisabled) {
            estimatorKalmanEnqueueFlow(&flowData);
        }
        else {
            outlierCount++;
        }
    }
}
```

Obrázek 7-16. Hlavní jádro kódu pro flow desk

## 7.5.2 Python skript na držení hover módu v 50 cm

Tento jednoduchý python skript je napsán a rozebrán z důvodu porozumění, jak implementovat python skript s povely pro multikoptéru. Ve skriptu jsou vidět importy času, logování a import knihovny pro Crazyrádio. Dále je skript psán tak aby navázal spojení s multikoptérou a dále nastavil kalmanovy filtry. Posledním krokem je vyslání komandu *send\_hover\_point*, což znamená aktivování rozšíření flow deck.

```
import logging
import time

import cflib.crtp
from cflib.crazyflie.syncCrazyflie import SyncCrazyflie

logging.basicConfig(level=logging.ERROR)

cflib.crtp.init_drivers(enable_debug_driver=False)

print("Scanning interfaces for Crazyflies...")
available = cflib.crtp.scan_interfaces()

print("Crazyflies found:")
for i in available:
    print(i[0])

if len(available) > 0:
    link_uri = available[0][0]
    print(link_uri)

if __name__ == '__main__':
    with SyncCrazyflie(link_uri) as scf:
        cf = scf.cf

        cf.param.set_value('kalman.resetEstimation', '1')
        time.sleep(0.1)
        cf.param.set_value('kalman.resetEstimation', '0')
        time.sleep(2)

        while True:
            cf.commander.send_hover_setpoint(0, 0, 0, 0.5)
            time.sleep(0.1)
```

Obrázek 7-17. Zdrojový kód skriptu

### 7.5.3 Python skript pro ovládání multikoptéry pomocí klávesnice

Tento skript je naprogramován tak, aby bylo možné po spuštění skriptu ovládat multikoptéru v režimu hover módu. Ke puštění skriptu potřebujeme mít nainstalovaný python. Skript je napsán tak aby po spuštění navázal spojení s multikoptérou pomocí rádiového signálu. Po navázání spojení multikoptéra zahájí stoupání a pak ji lze ovládat pomocí kláves.

```
import time
import sys
import tty
import termios
import argparse
import cflib

from cflib.crazyflie import Crazyflie
from threading import Thread, Event
from datetime import datetime

accelvaluesX = []
accelvaluesY = []
accelvaluesZ = []

import logging
logging.basicConfig(level=logging.WARNING)
```

Obrázek 7-18. Seznam Importovaných knihoven pro let za pomoci klávesnice

klávesa	povel	rozsah
w	thrust+	0,20 cm -1,5m výška
s	thrust-	0,20 -1,5 m výška
d	yaw+	(-200,200) stupňů
a	yaw-	(-200,200) stupňů
l	roll+	(-50,50) stupňů
j	roll-	(-50,50) stupňů
i	pitch+	(-50,50) stupňů
k	pitch-	(-50,50) stupňů
q	ukončení	-

Tabulka 7-1. Seznam ovládacích prvků

Obrázek 7-18 ukazuje potřebné knihovny pro tento skript. Skript používá knihovnu *cflib*, ze které volá potřebné funkce, které se poté posílají na platformu, kde je přebírá firmware.

Obrázek 7-19 ukazující definice proměnných použité v hlavní funkci kódu.

Hlavní část kódu se zabývá ovládáním multikoptéry za použití kláves. Tabulka 7-1 popisuje, jak se multikoptéra ovládá.

```
def increasing_step(self):
    global key
    global accelvaluesX
    global accelvaluesY
    global accelvaluesZ
    start_thrust = 0.2 # (blades start to rotate after 10000)
    min_thrust = 0.1
    max_thrust = 0.5
    thrust_increment = 0.05
    start_roll = 0
    roll_increment = 0.1
    min_roll = -50
    max_roll = 50
    start_pitch = 0
    pitch_increment = 0.1
    min_pitch = -50
    max_pitch = 50
    start_yaw = 0
    yaw_increment = 30
    min_yaw = -200
    max_yaw = 200
    stop_moving_count = 0
    pitch = start_pitch
    roll = start_roll
    thrust = start_thrust
    yaw = start_yaw
```

Obrázek 7-19. Definice proměnných

## 7.6 Barometr LPS25H x Optický tok

Barometr LPS25H na platformě funguje jako doplňkový sensor, s kterým je možné řešit poziční stabilizaci v rámci vektoru  $z$ . V kódu je nastaven výpočet tlaku, ze kterého se bere aktuální výška od hladiny moře. V příloze práce je přiloženo video a výsledné záznamy z průběhu testů.

V rámci tohoto sensoru je naprogramován limit pro rychlost rotace motorů, podle kterých se drží výška. Po vzletnutí multikoptéry se po dosažení požadované výšky sepne sensor barometrického tlaku a ten si udržuje aktuální výšku tak, že drží rychlost rotaci motorů na čísle **32767**. Toto číslo je stav rychlosti motorů, kdy dochází ke stabilizaci multikoptéry. Multikoptéra poté vypočítává a přidává nebo ubírá rychlost podle aktuálního barometrického tlaku digitálně převáděného ze sensoru.

Tento postup bohužel nefunguje pro drift multikoptéry do os  $x$ ,  $y$ . Jelikož multikoptéra nemá žádné další ukazatele, podle kterých by se řídila. Ovládat Crazyflie 2.0 za pomoci skriptu, když je používán barometr se nedoporučuje právě z těchto důvodů. V práci je v příloze uveden skript, který spustí a vylétne s multikoptérou do určité výšky a poté loguje záznamy ze stabilizátorů. Výsledkem je udržení výšky v rozmezí 30 až 60 sekund, než multikoptéra ztratí rovnováhu a drift se stane tak velký, že multikoptéra havaruje. Jiný případ je, když se tento sensor použije s dostupným ovládním, kdy operátor vyvažuje nedokonalosti způsobené driftem.

Dalším důvodem pro nevyužití barometru je jeho náchylnost na změny tlaku. Při testování je potřeba mít stabilní prostředí. Jakákoliv změna tlaku v rámci otevření/zavření okna, či dveří v místnosti může způsobit velkou změnu tlaku se kterou se barometr poté musí vypořádat a dojde tím k chybovosti ve výpočtu stabilizace a následný drift a možnost havarování multikoptéry. Dalším problémem může být například změna počasí, a tedy i tlaku.

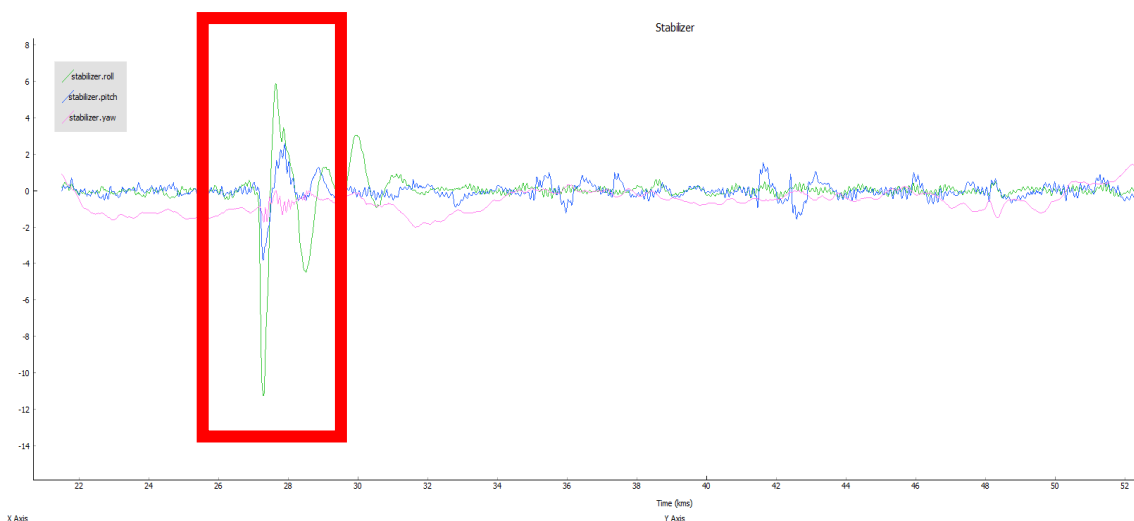
Naproti tomu se v práci pracuje s optickým tokem, který je oproti barometru externě napojený sensor. Při testech dosahoval stabilních výsledků. Z grafů v reálném čase je jasně vidět minimální vybočení z pozice do které je multikoptéra uvedena i v rámci pohybu po ose  $x$  nebo  $y$ . Jelikož se na flow desce se nachází i laserový zaměřovač

pro osu z dochází zde k velmi přesnému měření vzdálenosti osy z a tím na základě výpočtů v rozšířeném kalmanovo filtru stabilní poziční stabilizaci.

K nevýhodám tohoto řešení je nutno doplnit, že na stejnorodém povrchu může dojít k malému vychylování, jelikož čočka optického toku nedokáže rozpoznat pixely kterými by se řídila. Přesto je tento problém v rámci stabilizace minimální a přijatelný. Další nevýhodou je světelnost místnosti. Při šerejších podmínkách může docházet k driftu na ose z, jelikož se o ní stará laser. Při porovnání těchto nevýhod vůči nevýhodám barometru je jasné, že optický tok přijatelnějším, kvalitnějším a z hlediska stabilizace výkonnějším.

### 7.6.1 Praktické porovnání stabilizace

V porovnání je platforma po dobu 30 sekund v stabilizované poloze a jsou porovnávány odchylky od pozice za pomoci stabilizačního grafu běžícího v reálném čase. Jak je vidět viz Graf 7-1, multikoptéra v rámci tohoto testování za pomoci optického toku je velice stabilní. Jediný problém nastal v době, kdy multikoptéra dostala lehký drift viz červený obdélník v grafu. Multikoptéra se s tím vypořádala a dále je vidět stabilní let až do uplynutí časového úseku vyhraněného pro let. Celý průběh je uveden v příloze.

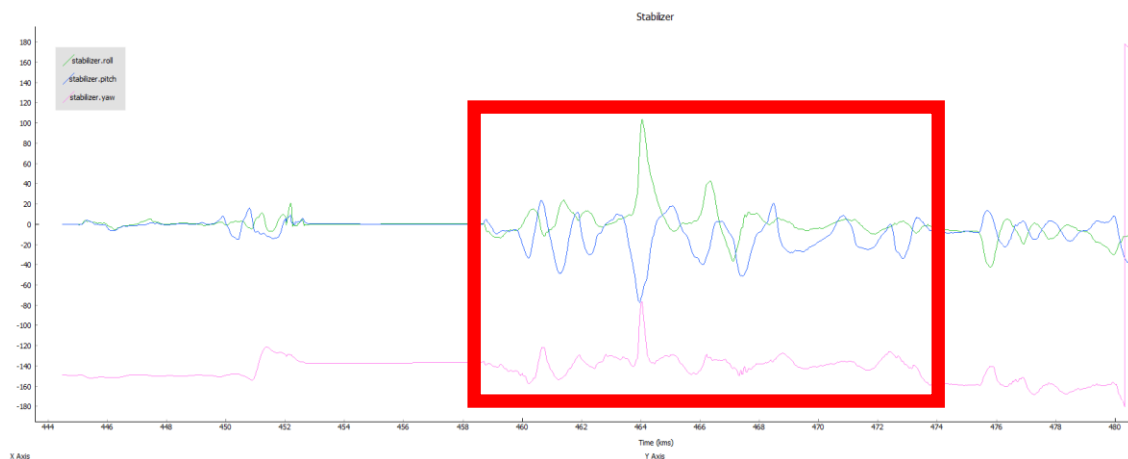


Graf 7-1. Výřez grafu stabilizátoru pomocí optického toku

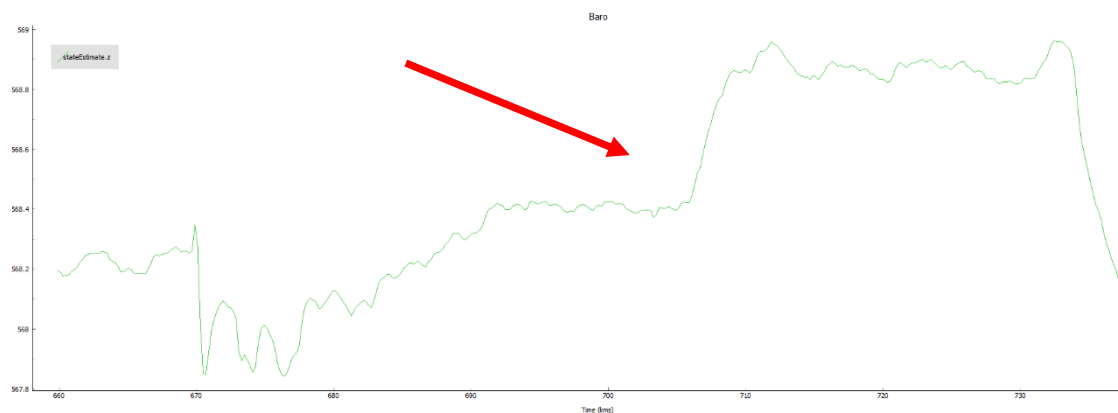
Druhým testováním je za pomoci barometru. Graf 7-2 ukazuje, že start a průběh v pořádku, ale později se projeví nedokonalost barometrického sensoru, kdy zaznamená větší výkyv tlaku a Crazyflie si nedokáže dále poradit se stabilizací a následně se odchýlí z pozice a havaruje. Graf 7-3 ukazující změnu tlaku a nárůst hodnot na sensoru viz



červená šipka. Všechny testovací případy jsou nahrány v rámci grafů v reálném čase, videí a logů v příloze.



Graf 7-2. Výřez grafu stabilizátoru za pomoci barometru



Graf 7-3. Výřez grafu zaměřování senzorem barometru

Tyto odchylky a nesrovnalosti lze vyčíst i z logování. Tabulka 4-1 označující část logů, kde dochází k odchylce měření tlaku za pomoci barometrického senzoru.

Log: time(200ms).(Z range)
442099,566.5360717773438
442299,566.5369262695312
442499,566.547607421875
442699,566.56689453125
442899,566.5814208984375
443099,566.5926513671875
443299,566.6209716796875
443499,566.6497192382812
443699,566.6764526367188
443899,566.6763305664062
444099,566.6835327148438
444299,566.688232421875
444499,566.7031860351562
444699,566.7220458984375
444899,566.7630615234375
445099,566.7542724609375
445299,566.6669921875
445499,566.6072998046875
445699,566.6206665039062
445899,566.5902099609375
446099,566.543701171875
446299,566.4969482421875
446499,566.5172729492188
446699,566.6082763671875
446899,566.6574096679688
447099,566.6510620117188
447299,566.6431884765625
447499,566.6574096679688
447699,566.6790771484375
447899,566.7206420898438
448099,566.73486328125
448299,566.7425537109375
448499,566.76123046875

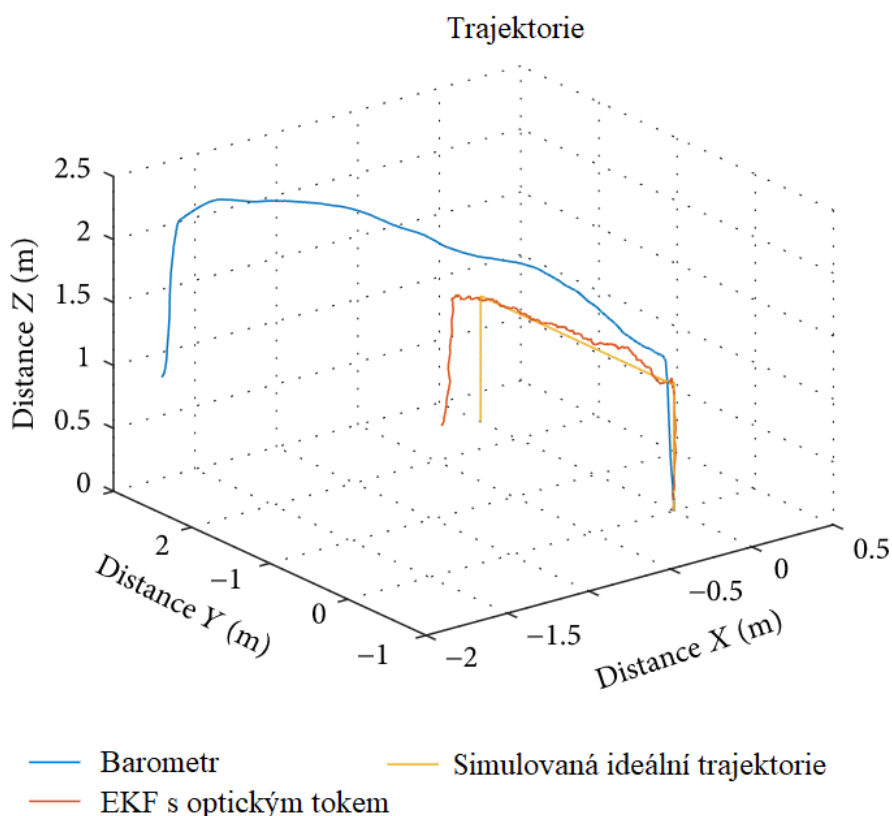
Tabulka 7-2. Tabula Logu Sensoru

Timestamp, stabilizer.roll, stabilizer.pitch, stabilizer.yaw
325866,-0.01560290064662695,- 0.017866546288132668,14.83346939086914
325886,-0.016632767394185066,- 0.01746150106191635,14.835411071777344
325906,-0.01639455556869507,- 0.016647057607769966,14.83617115020752
325926,-0.015534395352005959,- 0.016812333837151527,14.837417602539062
325946,-0.01566576398909092,- 0.017956089228391647,14.839077949523926
325966,-0.015725886449217796,- 0.018649350851774216,14.838750839233398
325986,-0.01553517859429121,- 0.01879742369055748,14.84034538269043
326006,-0.015280650928616524,- 0.019301841035485268,14.841723442077637
326026,-0.015998512506484985,- 0.01847529038786888,14.843399047851562
326046,-0.01645490527153015,- 0.017996899783611298,14.844884872436523
326066,-0.017551694065332413,- 0.017360983416438103,14.845297813415527
326086,-0.01738884113729,- 0.018132338300347328,14.8454008102417
326106,-0.01678638905286789,- 0.01756267063319683,14.84752082824707
326126,-0.018163492903113365,- 0.01851625367999077,14.84847354888916
326146,-0.018372273072600365,- 0.02014111913740635,14.849701881408691
326166,-0.019341424107551575,- 0.020673148334026337,14.85067081451416
326186,-0.02024839259684086,- 0.02052689529955387,14.851627349853516
326206,-0.02069776877760887,- 0.019792376086115837,14.852424621582031
326226,-0.020153217017650604,- 0.018156681209802628,14.853415489196777
326246,-0.02043880708515644,- 0.01912282593548298,14.85425853729248
326266,-0.020187193527817726,- 0.019584733992815018,14.856769561767578
326286,-0.019545825198292732,- 0.020090732723474503,14.856301307678223
326306,-0.01959146559238434,- 0.020733928307890892,14.857763290405273

Tabulka 7-3. Log stabilizátoru při barometru

## 7.6.2 Porovnání s určenou dráhou letu

Před porovnáním bylo v rámci testu určena ideální letová trajektorie, kterou by měla platforma urazit. Graf 7-4 zobrazuje ideální trajektorii jako oranžovou barvu v grafu. První testovací pokus byl proveden na multikoptéře, která byla stabilizována barometrem, na grafu viděna jako modrá barva trajektorie. Jak je vidět první metr drží trajektorii, ale dále pak začala vybočovat směrem vzhůru a později zcela mimo kurz. Barometr reaguje i na malé změny atmosférického tlaku. Toto odchýlení bylo způsobeno právě tím. Dále proběhl test letu za pomoci optického toku. V porovnání trajektorií je jasně vidět, že až na malé kolísání a překročení vzdálenosti byla trajektorie dodržena. Tento test byl zpuštěn za pomoci skriptu a poslán jako komand na multikoptéru a operátor dále nezasahoval.



Graf 7-4. Trajektorie porovnání

## 8 Závěr

Tato práce se zabývá poziční stabilizací multikoptér. Hlavním cílem byla analýza dostupných možností poziční stabilizace, které neobsahují GPS systém. Sepsáním analýz a východisek byla vybrána aplikace poziční stabilizace na platformě Crazyflie 2.0 za pomoci optického toku. Pro tuto možnost byla použita Flow deska, která obsahuje sensor optického toku a laserového zaměrování.

V teoretické části je popsána platforma Crazyflie 2.0 hardwarově i systémově. V práci je popsána problematika poziční stabilizace pomocí vnitřních sensorů IMU, jejich rozdělení, komunikace a následná analýza možností řešení, jak je možné tento systém vnitřních sensorů rozšířit o externí sensory. Na základě analýzy byla vybrána možnost externího sensoru optického toku. Později v praktické části jsou popsány fyzikální a matematické výpočty pro fungování optického toku a laserového zaměrování fúzi interních systémů s externími. Kapitola se věnuje i šumu, se kterým se tyto sensory potýkají. Později je vysvětlen Kalmanův filtr, který se zabývá fúzí sensorů a výpočtů, které udržují poziční stabilizaci multikoptéry Crazyflie 2.0.

Po pochopení problematiky optického toku se práce později věnuje testování čidel a letu. V praktická část se zaměřovala především na stabilizaci platformy, její vyvážení od umístění baterie, až pro co nejdokonalejší kalibraci vrtulí, aby nedocházelo k problémům. Po vyzkoušení testovacích letů ve VirtualBoxu, kde byla možnost otestovat funkčnost všech systémů a sensorů na linuxovém operačním systému vytvořeném pro platformu Crazyflie 2.0 se praktická část práce přesunula na Windows, kde se v programu Pycharm kódovali skripty pro ovládání letu a pozičního stabilizování.

Výsledkem práce bylo vytvoření a zpracování možností poziční stabilizace na platformě Crazyflie 2.0. Za pomoci flow desky byly popsány základní funkce a možnosti optického toku. Jelikož je platforma Crazyflie 2.0 stále ve vývoji, je dosažení poziční stabilizace jen dalším možným krokem k autonomnímu letu. Tímto je možné vylepšení a rozšíření platformy o další sensory tématem pro další výzkum. Za hlavní přínos této práce lze tedy považovat základní principy poziční stabilizace, které mohou posloužit jako základ pro další možnosti využití a výzkumu.

## 9 Citace

- [1] Crazyflie 2.0. In: Bitcraze Wiki [online]. [cit. 2017-03-06]. Dostupné z: <https://wiki.bitcraze.io/projects:crazyflie2:index>
- [2] Crazyradio PA. In: Bitcraze Wiki [online]. [cit. 2017-03-06]. Dostupné z: <https://wiki.bitcraze.io/projects:crazyradiopa:index>
- [3] MPU-9250. *MPU-9250 Product Specification Revision 1.1* [online]. 2016, s. 5-22 [cit. 2017-05-07]. Dostupné z: <https://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf>
- [4] *MEMS pressure sensor: 260-1260 hPa absolute digital output barometer: STMicroelectronics* [online]. 2016, 1-2 [cit. 2017-08-08]. Dostupné z: <http://www.st.com/en/mems-and-sensors/lps25h.html>
- [5] DUNKLEY, Oliver, Jakob ENGEL a Daniel CREMES. *Visual-Inertial Navigation for a Camera-Equipped 25 g Nano-Quadrotor* [online]. Munich [cit. 2017-09-08]. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/>. Technical University Munich.
- [6] Loco Positioning system: The system. *Bitcraze.io* [online]. Švédsko: Bitcraze AB, 2017 [cit. 2017-11-08]. Dostupné z: <https://www.bitcraze.io/loco-pos-system/>
- [7] Bitcraze. *Bitcraze Store: Flow deck* [online]. Švédsko: Bitcraze.io, 2017 [cit. 2017-11-1]. Dostupné z: <https://store.bitcraze.io/collections/decks/products/flow-deck>
- [8] World smallest Time-of-Flight (ToF) ranging sensor: VL53L0X. *Life.Augmented: VL53L0X* [online]. Morocco: STMicroelectronics, 2017 [cit. 2017-09-18]. Dostupné z: <http://www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html>
- [9] *VirtualBox: Welcome to VirtualBox.org!* [online]. Toronto a Redwood City: Oracle, 2017 [cit. 2017-11-19]. Dostupné z: <https://www.virtualbox.org/>

- [10] VOJÁČEK, Antonín. Co je to Kalmanova filtrace? [online]. 2007 [cit. 2017-08-04]. Dostupné z: <http://automatizace.hw.cz/clanek/2007042901>
- [11] KELLY, Alonzo (1994). "A 3D state space formulation of a navigation Kalman filter for autonomous vehicles". DTIC Document: 13. 2006 Corrected Version. [online]. 2007 [cit. 2017-09-09] Dostupné z: [http://www.frc.ri.cmu.edu/~alonzo/pubs/reports/kalman\\_V2.pdf](http://www.frc.ri.cmu.edu/~alonzo/pubs/reports/kalman_V2.pdf)
- [12] REID, Ian; Term, Hilary. "Estimation II" . www.robots.ox.ac.uk. Oxford University [online]. 2007 [cit. 2017-09-09] Dostupné z: <http://www.robots.ox.ac.uk/~ian/Teaching/Estimation/LectureNotes2.pdf>
- [13] Verbandt, M., B. Theys, and J. De Schutter (2014). "Robust marker-tracking system for vision-based autonomous landing of vtol uavs". In: Proceedings of the International Micro Air Vehicle Conference and Competition 2014. Delft University of Technology, pp. 84–91

## 10 Seznam použitých obrázků, tabulek a grafů

### Obrázky:

Obrázek 4-1: Multikoptéra Crazyflie 2.0.....	5
Obrázek 4-2. Systémová architektura [2] .....	7
Obrázek 4-3: rámec protokolu syslink.....	8
Obrázek 4-4: CrazyRadio PA .....	10
Obrázek 4-5: Čip MPU-9250.....	11
Obrázek 6-1. Úkázka rozložení kotvicích prvků .....	16
Obrázek 6-2. Rozložení čipu PMW3901 .....	17
Obrázek 7-1: Směry vrtulí na platformě .....	20
Obrázek 7-2: Ukázka kalibrace vrtule .....	21
Obrázek 7-3: Model ochranného prvku .....	22
Obrázek 7-4: Reálná fotografie s ochranným prvkem.....	23
Obrázek 7-5. Rámce zobrazující vektor optického toku.....	24
Obrázek 7-6. Znázorněný model pohybu optického toku.....	25
Obrázek 7-7. Model měření optického šumu [13].....	28
Obrázek 7-8. Teoretické znázornění postupu výpočtu KF .....	29
Obrázek 7-9. Proces odhadu EKF.....	31
Obrázek 7-10: Import ve VirtualBox.....	32
Obrázek 7-11: Spuštění Crazyflie klienta.....	33
Obrázek 7-12. Pseudo kód pro výpočet nadmořské výšky .....	34
Obrázek 7-13. Error způsobený výrobní vadou.....	35
Obrázek 7-14. Fotografie multikoptéry s flow desk .....	36
Obrázek 7-15: Umístění rozšíření.....	36
Obrázek 7-16. Hlavní jádro kódu pro flow desk .....	37
Obrázek 7-17. Zdrojový kód skriptu.....	38
Obrázek 7-18. Seznam Importovaných knihoven pro let za pomoci klávesnice .....	39
Obrázek 7-20. Definice proměnných.....	40

**Tabulky:**

Tabulka 4-1: Názvy a významy paketů .....	9
Tabulka 7-1. Seznam ovládacích prvků.....	39
Tabulka 7-2. Tabula Logu Sensoru.....	44
Tabulka 7-3. Log stabilizátoru při barometru .....	44

**Grafy:**

Graf 7-1. Výřez grafu stabilizátoru pomocí optického toku.....	42
Graf 7-2. Výřez grafu stabilizátoru za pomoci barometru.....	43
Graf 7-3. Výřez grafu zaměřování senzorem barometru .....	43
Graf 7-4. Trajektorie porovnání.....	45



## 11 Přílohy

Příložené CD obsahuje elektronickou podobu práce ve formátu PDF, zdrojové kódy jednotlivých součástí aplikace, videa a tabulky logů z testování. Zdrojové kódy jsou ve formě prostého textu.

Elektronická podoba bakalářské práce je umístěna v adresáři */Dokumenty*.

Soubory týkající se testování jsou umístěny ve složkách */Videa* a */Logy*.

Zdrojové kódy jsou umístěné ve složce */Kódy*.