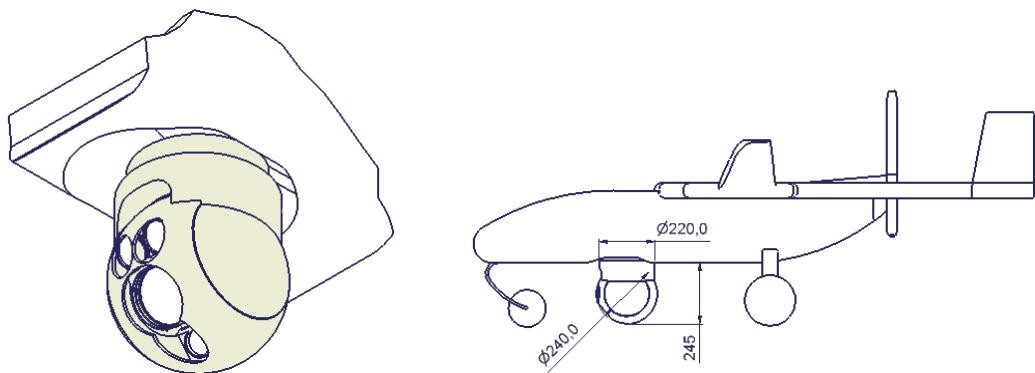


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA ELEKTROTECHNICKÁ



DIPLOMOVÁ PRÁCE

Návrh řízení pro systém stabilizace  
optické osy kamerového systému  
pro bezpilotní letoun.

Praha, 2008

Autor: Martin Řezáč

Autor: Bc. Martin Řezáč  
Vedoucí: Ing. Zdeněk Hurák, Ph.D.  
              Katedra řídicí techniky, Fakulta elektrotechnická,  
              České vysoké učení technické v Praze  
Oponent: Ing. Jiří Nohýl, CSc.  
              VTÚL & PVO, Praha  
Rok: 2008

## **Prohlášení**

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne 23.května 2008

---

podpis

## **Poděkování**

Děkuji především svému vedoucímu diplomové práce Ing. Zdeňkovi Hurákovi, Ph.D. a také všem ostatním kolegům za ochotu a cenné rady a připomínky při řešení této práce.

Dále pak děkuji své rodině a blízkým přátelům za jejich podporu během celé doby mého studia.

# **Abstrakt**

Pro zadavatele projektu – Výzkumný technický ústav letectva (VTÚL) – je vyvíjena stabilizovaná základna pro kamerový sledovací systém obsahující dvě kamery a laserový dálkoměr. Tato základa ve tvaru koule bude umístěna v novém českém bezpilotním letounu MANTA, následníka úspěšného bezpilotního letounu Sojka. Účelem celého systému je sledování pevných i pohybujících se cílů při současném potlačení vlivu pohybu letounu a ostatních poruch jako jsou poryvy větru. Systém je možné rozdělit na dvě vzájemně nezávislé části – vnitřní rychlá inerciální zpětnovazební smyčka a vnější pomalá vizuální zpětná vazba (založená na algoritmech zpracování obrazu a rozpoznávání).

Tato diplomová práce popisuje návrh a také implementaci Stabilizace optické osy kamerového systému měřením inerciálních úhlových rychlostí. Pro navrženou koncepci řídicího systému byl vytvořen simulační model stabilizované základny. Následně tato práce představuje vyrobený hardware a software potřený pro ověření činnosti vytvořených algoritmů ve vyrobené stabilizované základně.

# **Abstract**

For a governmental contractor – Czech Air Force R&D Institute (VTÚL) – the inertially stabilized platform for a tracking camera system is being developed. The system consists of two cameras and a laser range finder. This sphere-shaped platform will be carried by a new Czech unmanned aircraft MANTA, a successor of the successful SOJKA UAV. The purpose of the complete system is to track standing or moving targets while attenuating the influence of motion of an aircraft and other disturbances like wind buffeting. The system is comprised of two relatively independent subsystems – inner fast inertial feedback control loop and outer slow visual feedback control loop (based on image recognition, locking and tracking).

This diploma paper describes a design and also an implementation of the Line-of-sight stabilization of an airborne camera system using inertial measurements. For the designed method of stabilization the simulation model has been created. Consequently, this work introduces realized hardware and software necessary to implement and test all designed algorithms in real developed inertially stabilized platform.

# ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Martin Řezáč

Studijní program: Elektrotechnika a informatika (magisterský), strukturovaný  
Obor: Kybernetika a měření, blok KM1 - Řídicí technika

Název tématu: **Stabilizace optické osy kamerového systému bezpilotního letounu**

Pokyny pro vypracování:

1. Nastudujte problematiku stabilizace optické osy pomocí inerciálních měření. Popište a srovnejte používané koncepce konstrukčního uspořádání takových systémů.
2. Pro zadanou mechanickou koncepci navrhněte vhodnou koncepci řídicího systému (co a kde a čím měřit, co a kde a čím řídit).
3. V souladu s Vámi navrženou koncepcí řídicího systému vyberte vhodné akční členy (motory), snímače (gyroskopy, akcelerometry, magnetometry,) a počítač(e).
4. Vytvořte simulační model pro navržený systém.
5. Navrhněte a realizujte prototyp elektronické řídicí jednotky pro stabilizaci optické osy.
6. Navrhněte a implementujte algoritmus pro stabilizaci optické osy. Otestujte na simulačním modelu, případně i na reálném systému (postaví průmyslový partner).

Seznam odborné literatury:

M. K. Masten, L. A. Stockum (Ed.) Selected papers on Precision Stabilization and Tracking Systems for Acquisition, Pointing, and Control Applications. SPIE, Vol. MS 123, 1996.

Vedoucí: Ing. Zdeněk Hurák, Ph.D.

Platnost zadání: do konce zimního semestru 2008/2009

prof. Ing. Michael Šebek, DrSc.  
vedoucí katedry



v.r. M. Šebek  
doc. Ing. Boris Šimák, CSc.  
děkan



# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Hlavní požadavky VTÚL na konstrukci základny a řídicí algoritmy . . . . .	2
1.2	Cíl diplomové práce . . . . .	3
<b>2</b>	<b>Mechanická konstrukce základny</b>	<b>4</b>
2.1	Přehled základních typů uvažovaných konstrukcí . . . . .	4
2.2	Zvolená konstrukce . . . . .	6
2.3	Symbolika . . . . .	8
2.4	Hlavní elektromechanické části stabilizované základny . . . . .	9
<b>3</b>	<b>Koncepce řídicího systému</b>	<b>12</b>
3.1	Princip a základní možnosti stabilizace obrazu . . . . .	12
3.2	Kinematika stabilizované základny . . . . .	14
3.3	Polohová, rychlostní, a proudová zpětná vazba . . . . .	16
<b>4</b>	<b>Výpočetní návrh řídicího systému</b>	<b>18</b>
4.1	Kaskádní uspořádání . . . . .	18
4.2	Rychlostní zpětná vazba . . . . .	20
4.3	Vizuální zpětná vazba . . . . .	20
4.4	Slepé směrování . . . . .	21
4.4.1	Princip slepého směrování . . . . .	22
4.4.2	Model v simulinku . . . . .	25
4.4.3	Redukovaná inerciální jednotka s EKF (rozšířený Kalmanův filtr) .	28
4.4.3.1	Měřené veličiny, polovodičové gyroskopy a akcelerometry	30
4.4.3.2	Reprezentace polohy - quaterniony vs. Eulerovy úhly .	31
4.4.3.3	Stavový model použitý pro odhadování . . . . .	32
4.4.3.4	Algoritmus rozšířeného Kalmanova filtrov . . . . .	33
4.4.3.5	Realizace algoritmu v jednočipovém počítači . . . . .	33
4.4.3.6	Výsledky . . . . .	34
4.4.3.7	Působení vnějšího zrychlení . . . . .	36
4.4.3.8	Závěr a možná vylepšení . . . . .	38
<b>5</b>	<b>Implementace řídicího systému</b>	<b>39</b>
5.1	Struktura elektronické části . . . . .	39
5.1.1	Elektronika vnitřního závěsu základny . . . . .	41

5.1.2	Elektronika řídicích jednotek . . . . .	42
5.2	Navržený hardware . . . . .	43
5.2.1	Řídicí jednotky pohonů – CPUBoardEL a CPUBoardAZ . . . . .	43
5.2.2	Měřicí jednotka – SPIBoard . . . . .	45
5.2.3	Výkonové spínače motorů – HmustekEL a HmustekAZ . . . . .	46
5.3	Software . . . . .	47
5.3.1	Program řídicí jednotky motorů – CPUBoardAZ a CPUBoardEL . . . . .	47
5.3.2	Program měřicí jednotky – SPIBoard . . . . .	48
5.3.3	Zpracování signálu IRC čidel . . . . .	49
5.3.3.1	Sled kroků při čtení dat IRC čidla z obvodu CPLD . . . . .	50
5.3.4	Grafické uživatelské rozhraní . . . . .	51
<b>6</b>	<b>Experimentální ověření</b>	<b>52</b>
6.1	Výsledná funkčnost stabilizované základny . . . . .	52
6.2	Zapojení experimentu . . . . .	53
6.3	Výsledky . . . . .	53
<b>7</b>	<b>Závěr</b>	<b>57</b>
<b>Literatura</b>		<b>59</b>
<b>A</b>	<b>Část zdrojového kódu provádějící algoritmus EKF</b>	<b>I</b>
<b>B</b>	<b>Fotky a výkres sestavy základny</b>	<b>III</b>
<b>C</b>	<b>Schémata elektroniky</b>	<b>VI</b>
<b>D</b>	<b>Simulinkový model základny</b>	<b>IX</b>
<b>E</b>	<b>Obsah přiloženého CD</b>	<b>XII</b>

# Kapitola 1

## Úvod

Pro zadavatele projektu, Výzkumný technický ústav letectva (VTÚL <http://vtul.cz>), je vyvíjena ve spolupráci kateder řídicí techniky a kybernetiky FEL ČVUT a fy. Opten s.r.o. stabilizovaná základna pro kamerový systém. Tato základna bude umístěna uvnitř nově vyvýjeného bezpilotního letounu MANTA (vizte drak letounu na obr. 1.1) a pomocí speciálního mechanizmu bude za letu vysouvána z trupu letounu ven. Předchůdce letounu MANTA – letoun Sojka je úspěšně provozován již více než 10 let.

Cílem celého systému je sledovat pevné či pohyblivé cíle a přitom potlačovat nežádoucí pohyby letadla (např. zatačení). V současném provedení je kamerový systém pevně přichycen k trupu letounu Sojka, bez možnosti změny orientace a bez schopnosti tlumit nežádoucí rychlé pohyby letounu. Proto bude vyvíjená stabilizovaná základna pro kamerový systém velkým kvalitativním povýšením.



Obrázek 1.1. Letouny Manta a Sojka na veletrhu IDET 2007. Fotka byla převzata z webového portálu <http://www.sekuritaci.cz/>.

Projekt stabilizované základny je rozdělen na tři hlavní části. První částí je návrh a výroba konstrukce základny. Pro tuto část byl k účasti na projektu přizván Ing. Milan Bartoš, CSc., Opten s.r.o.. Dalším úkolem je návrh algoritmů vyhledávaní pevných i pohybujících se cílů v reálném čase ve videosignálu kamery. Tento úkol řeší skupina na katedře kybernetiky pod vedením prof. Hlaváče. Posledním úkolem, řešeným na katedře

řídicí techniky, je úkol návrhu řídicích algoritmů stabilizace obrazu a realizace těchto algoritmů v elektronice umístěně přímo na míru uvnitř stabilizované základny.

Na řídicí části a vývoji elektroniky stabilizované základny jsem podílel coby student společně s kolegou Jaroslavem Žohou. Moje část diplomové práce se zabývá hlavně koncepčním návrhem celého systému a elektroniky, návrhy řídicích strategií a jejich simulacemi. Také jsem se zabýval realizací jednotlivých řídicích a měřicích algoritmů v navrženém hardware. Práce kolegy je naopak více zaměřena na návrh elektronické části systému a návrhu jednotlivých desek plošných spojů [19]. Obě práce dohromady tvorí ucelený popis řídicí části stabilizované základny.

## 1.1 Hlavní požadavky VTÚL na konstrukci základny a řídicí algoritmy

Jeden z požadovaných manévrů letadla je kroužení letadla na místě a přitom sledování 2 km vzdáleného cíle. Z tohoto kroužení plynou požadavky na možnost rotace azimutálního kloubu o  $n \times 360^\circ$  a naklopení elevačního kloubu o úhel až  $110^\circ$  (tj. nad horizont). Hlavní požadavky:

- rotace azimutálního kloubu o  $n \times 360^\circ$
- naklopení elevačního kloubu o úhel až  $110^\circ$ .
- napájení 27 V, 200 W
- rozměry max. průměr 200 mm, max. výška 300 mm
- hmotnost max. 10 kg (včetně stabilizovaných prvků)

Jako příslušenství základny byly nakonec vybrány následující kamery a laserový dálkoměr (foto viz. obr. 1.2).

- infračervená kamera Miricle 110K
- laserový dálkoměr Vectronix LRF42
- optická kamera FCB-IX



Obrázek 1.2. Vlevo: optická kamera, uprostřed: laserový dálkoměr, vpravo: infra kamera

Hlavním požadavkem na řídicí algoritmy je požadavek realizace sledování pohyblivých i pevných cílů v reálném čase pomocí videosignálu kamer. Tento úkol tedy vyžaduje spolupráci mezi systémy vyhledávaní cílů ve videosignálu kamer a řídicího systému realizujícího stabilizaci optické osy. Vedlejším požadavkem je také možnost manuálního ovládání kloubů základny např. pomocí joysticku.

## 1.2 Cíl diplomové práce

Cílem mé diplomové práce bylo provést kompletní návrh systému stabilizace optické osy kamerového systému z obr. 1.2 pomocí inerciálních měření. Pro navržený systém stabilizace vytvořit simulační model a pokud to bude možné také otestovat navržený algoritmus na reálném modelu. Je nutné zdůraznit, že se všechny požadavky podařilo splnit.

Celá diplomová práce byla řešena ve spolupráci s Centrem podpory talentovaných studentů<sup>1</sup>, které velmi přispělo ke kvalitním výsledkům práce.

V kap. 2 je proveden základní rozbor možností jakým způsobem lze stabilizaci optické realizovat. Je zde popsána konstrukce vyrobená průmyslovým partnerem a její hlavní části.

Kap. 3 seznamuje se základní koncepcí navrženého systému, možnostmi umístění jednotlivých senzorů a také se základními režimy stabilizované základny. Na tuto kapitolu navazuje kap. 4, ve které jsou jednotlivé myšlenky rozvinuty více do detailů, jsou zde představeny matematické modely řídicího systému.

Poslední dvě kapitoly se zabývají návrhem jednotlivých řídicích a měřicích bloků stabilizované základny a hlavně realizací navržených algoritmů přímo v hardware. Také jsou předvedeny výsledky dvou krátkých pokusů stabilizace optické osy kamerového systému.

---

<sup>1</sup><http://www.cepot.cz>

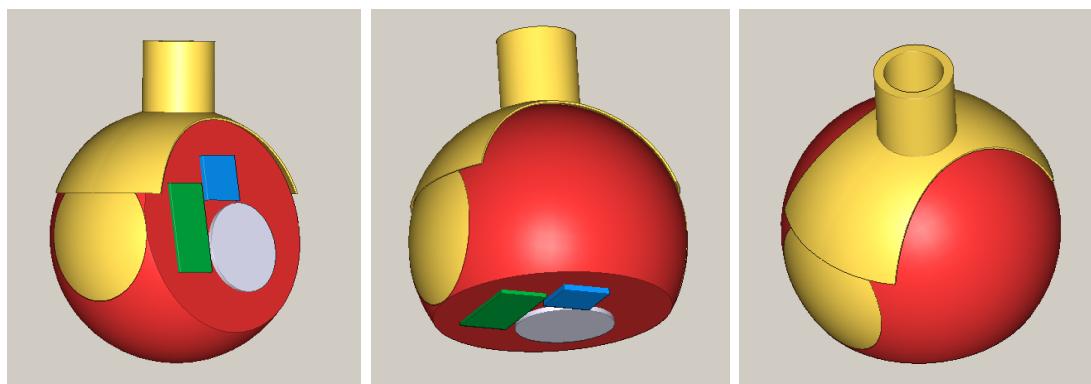
# Kapitola 2

## Mechanická konstrukce základny

V této kapitole budou popsány základní typy konstrukčních uspořádání, které jsou vhodné pro realizaci stabilizace optické osy, jejich výhody a nevýhody. Představena bude také konstrukce základny navržená a vyrobená pro VTÚL konstruktérem Milanem Bartošem, CSc., Opten s.r.o.

### 2.1 Přehled základních typů uvažovaných konstrukcí

Z uspořádání se dvěma stupni volnosti přichází v úvahu dvě nejrozšířenější, označované jako jako *El-Az* (elevation-azimut) příp. *Alt-Az* (altitude-azimut), s otáčením vnějšího závěsu okolo svislé osy a vnitřního závěsu okolo vodorovné osy, a *Tip-Tilt* s otáčením vnějšího i vnitřního závěsu okolo dvou na sebe kolmých vodorovných os.



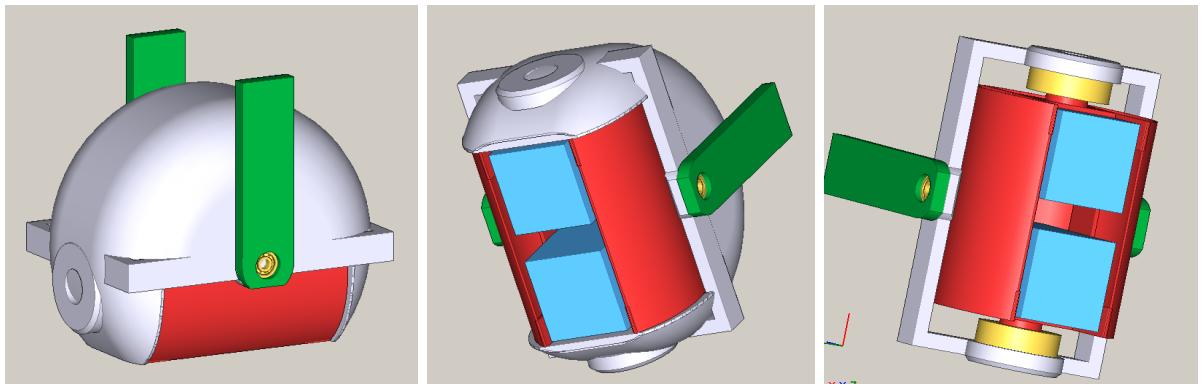
Obrázek 2.1. Konstrukce typu El-Az

Na obr. 2.1 můžete vidět konstrukci typu *El-Az*. Žlutou barvou je označen vnější okolo svislé osy otáčející se závěs. Vnitřní závěs, otáčející se okolo vodorovné osy, je

označen červenou barvou. Jednotlivé příslušenství (kamery) jsou označeny modrou, zelenou a šedou barvou. Jedná se o komerčně velmi rozšířenou konstrukci. Její hlavní výhodou je fakt, že je možné dosáhnout v obou kloubech velkých rozsahů úhlů a to i při velmi malých rozměrech celé konstrukce. Konstrukce je také velmi univerzální co se týče použití. Další velká výhoda této konstrukce se projeví za letu při změně azimuetu letu letadla, kdy při použití této konstrukce nedochází k rotaci obrazu (jako je tomu např. u konstrukce tip-tilt).

Konstrukce má však jednu zásadní nevýhodu. Při pohledu kolmo dolů pod letadlo je kamera nasměrovaná ve směru jedné osy otáčení, což je kinematicky velmi nežádoucí stav (při pohledu kamery do jedné z os o osu přijdeme). V okolí tohoto tzv. inflexního či singulárního bodu je potřeba velmi velkých kroutících momentů k rychlému pohybu, což činí eliminaci nežádoucích kmitů letadla při sledování cíle na zemi (na frekvencích řádově v jednotkách Hz) absolutně nemožnou<sup>1</sup>.

Na následujícím obr. 2.2 můžete vidět konstrukci typu *Tip-Tilt*. Oproti konstrukci *El-Az* má velkou výhodu, že neobsahuje singulární bod při pohledu kamer kolmo dolů. Singulární bod je zde přesunut a projeví se při otočení vnitřního červeného závěsu na horizont. Neexistence singulárního bodu při pohledu kolmo dolů umožní velmi dobrou stabilizaci optické osy v režimu, kdy kamery sledují krajinu pod letadlem a nejsou kladený požadavky na pozorování cílů na horizontu. Hlavní nevýhodou tohoto uspořádání jsou větší rozměry konstrukce oproti např. konstrukci *El-Az*, pokud jsou kladeny stejné požadavky na rozsahy pohledů (viz. požadavek sledování nad horizontem v kap. 1.1).

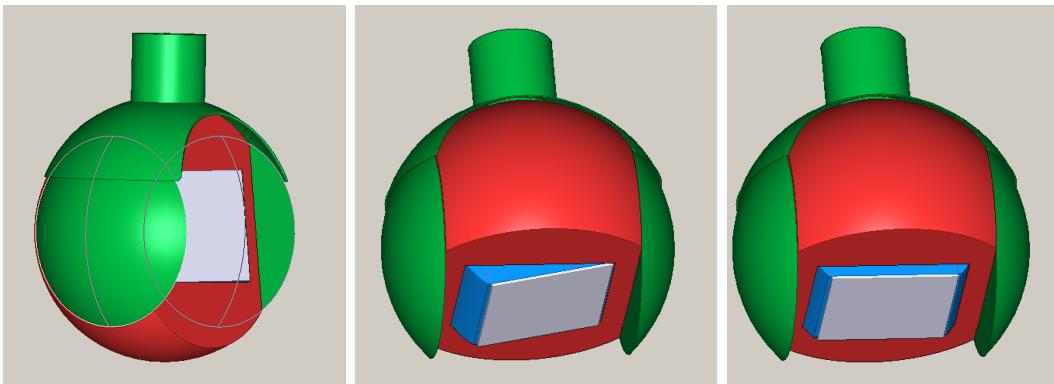


Obrázek 2.2. Konstrukce typu Tip-tilt

Jednou možností jak se vyhnout singulárnímu bodu při pohledu kolmo dolů pod letadlo je použití konstrukce s více (třemi) stupni volnosti, jak můžete vidět na obr. 2.3. Konstrukce velmi připomíná konstrukci *El-Az* s tím rozdílem, že obsahuje ještě jeden vnitřní kloub. Požadavky na rozsah otáčení v tomto kloubu přitom nemusí být příliš velké. Pro eliminaci singulárního bodu a zajištění možnosti stabilizace optické osy při pohledu kolmo dolů pod letadlo stačí rozsah otáčení nejvnitřejšího závěsu o  $\pm 10^\circ$  až  $\pm 20^\circ$  (na obr. 2.3 je označen modrou barvou). Ze všech tří uvedených konstrukcí se tato

<sup>1</sup>Na webových stránkách projektu můžete shlédnout krátké video demonstrující přítomnost singulárního bodu (<http://dce.felk.cvut.cz/mamok/video/singularita.avi>).

jeví pro potřeby stabilizace optické osy jako nejlepší. Bohužel je však díky třem stupňům volnosti nejsložitější a neumožňuje nést všechno požadované příslušenství.



Obrázek 2.3. Konstrukce se třemi stupni volnosti

S kolegy pracujícími společně na projektu stabilizované základny jsme měli možnost shlédnout ve VTÚL jednu hlavici (stabilizovanou základnu) komerčně vyráběnou v Izraeli. Tato konstrukce obsahovala 4 kloubů. Uspořádání vnějších dvou kloubů bylo totožné s konstrukcí typu *El-Az*, uspořádání vnitřních dvou kloubů naopak připomínalo konstrukci typu *Tip-Tilt*. Vnější 2 kloubů obsahovaly pohony s šnekovými převody (příp. jinými samosvornými). Vnitřní kloubů tak byly kryty oproti aerodynamickým účinkům proudění okolního vzduchu a prováděly jemné přesné namíření na cíl.

## 2.2 Zvolená konstrukce

I přes naše doporučení vyrobít konstrukci typu *Tip-Tilt* trval VTÚL na výrobě konstrukce typu *El-Az* ze dvou hlavních důvodů. Základna musí nést celkem objemné příslušenství (viz. kap. 1.1) s nímž je realizace konstrukce *Tip-Tilt* velmi obtížná. Druhým důvodem je požadavek na velké rozsahy pohledů základny, kterých je v případě *Tip-Tilt* kostrukce s objemným příslušenstvím nemožné dosáhnout, pokud má být stabilizovaná základna umístěna v letadle MANTA.

Byla tedy vyrobena konstrukce typu *El-Az* (Elevation – Azimut). Na obr. 2.4 můžete vidět její fotky. Konstruktérem základny je Ing. Milan Bartoš, CSc., Opten s.r.o.. Více fotek a také výkres sestavy konstrukce můžete vidět v příloze B. této diplomové práce. Na webových stránkách projektu [20] dále najeznete VRML<sup>2</sup> modely celé základny ve 3D.

<sup>2</sup>Virtual Reality Modeling Language

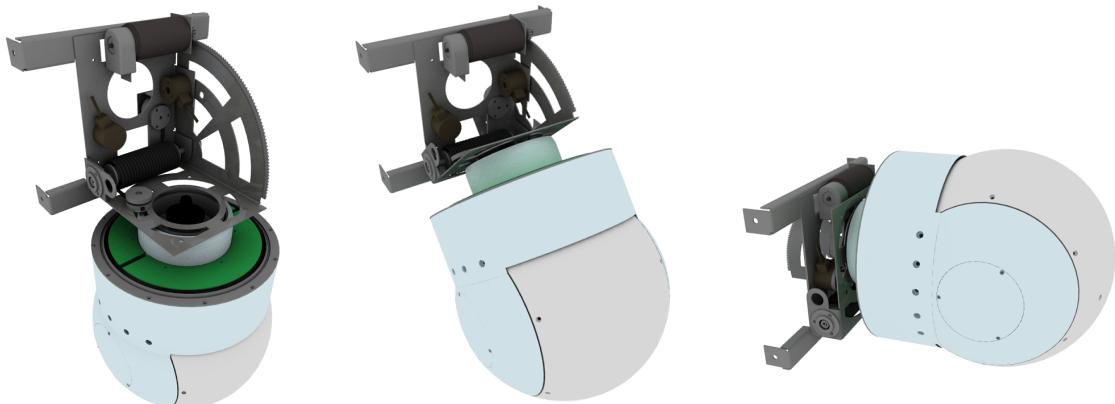
## 2.2. ZVOLENÁ KONSTRUKCE

7



Obrázek 2.4. Vyrobená konstrukce el-az.

Jelikož vyrobená stabilizovaná základna obsahuje singulární bod při pohledu kolmo dolů pod letadlo nebude možné při tomto pohledu zajistit přesnou stabilizaci namíření optické osy na cíl. Možností jak umožnit stabilizaci optické osy i při tomto pohledu je využít dalšího stupně volnosti ve formě výsuvu stabilizované základny z letadla a naklopit celou stabilizovanou základnu o určitý úhel ve směru (příp. v protisměru) letu letadla. Tím dojde k posunutí singularity mimo bod ležící pod letadlem. Model vysouvacího zařízení můžete vidět na obrázku 2.5. Bohužel však díky velkým rozměrům stabilizované základny nakonec toto zařízení nebude pro výsuv základny použito. Zařízení bude nahrazeno konstrukcí se svislým výsuvem stabilizované základny pod letadlo.

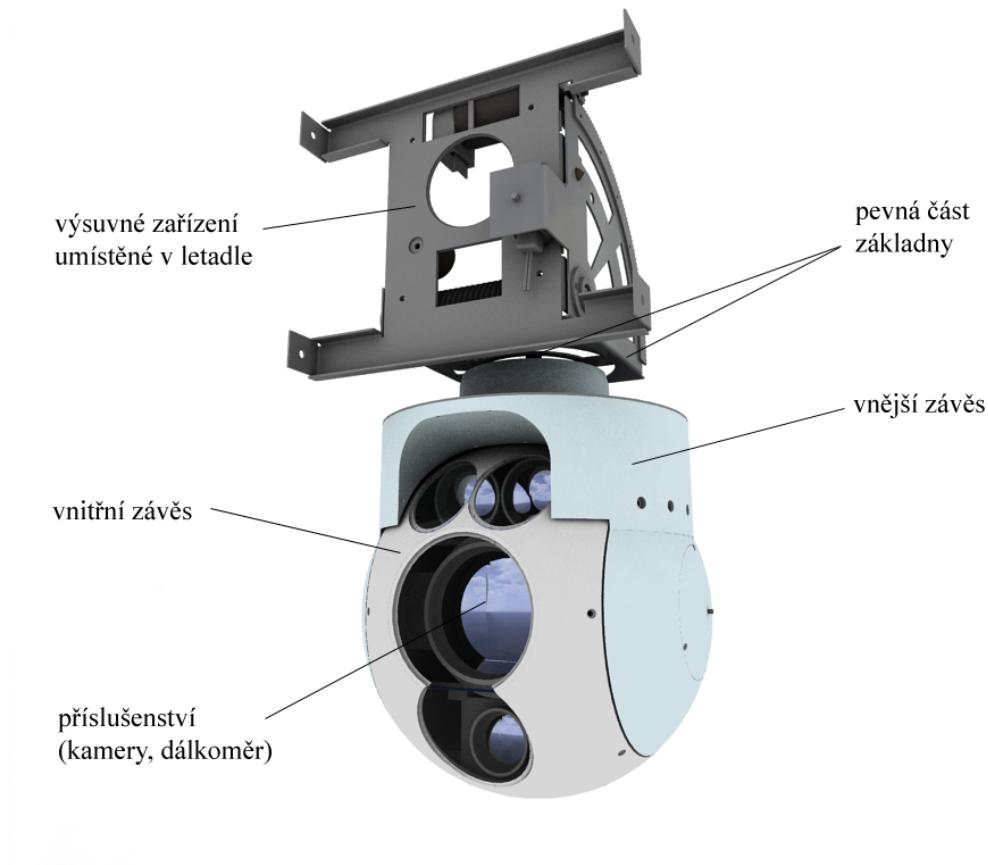


Obrázek 2.5. Zařízení pro výsuv stabilizované základny pod letadlo. Obrázky jsou vytvořeny na základě modelu základny v programu Inventor. Model byl zapůjčen od konstruktéra.

## 2.3 Symbolika

Nyní v krátkosti uvedu hlavní pojmy, které budu používat k popisu stabilizované základny a jednotlivých jejích částí.

- Stabilizovaná základna (někdy také hlavice či základna) – zařízení jako celek.
- Pevná část základny (angl. base) – jedná se o část stabilizované základny, která je pevně spojena s vysouvacím zařízením. Po zaaretování vysouvacího zařízení je tato část pevně spojena s letadlem.
- Vnější závěs (také vnější část či vnější rám) – část stabilizované základny, která se oproti pevné části otáčí okolo azimutálního kloubu.
- Vnitřní závěs (také vnitřní část či vnitřní rám) – část stabilizované základny, která se oproti vnějšímu závěsu otáčí okolo elevačního kloubu. V této části jsou umístěny kamery a veškeré příslušenství.



Obrázek 2.6. Hlavní části základny.

## 2.4 Hlavní elektromechanické části stabilizované základny

Konstrukce je navržena tak, že klouby se mohou protáčet dokola. Aby se přitom nepřekroutily kabely kamer a přídavné elektroniky je každý kloub osazen kluzným kroužkem (angl. Slip Ring, viz. obr. 2.8) AC 6023 od fy. MOOG, který umožňuje snadný průchod kabelů. Abychom snížili počet kabelů procházejících jednotlivými kroužky a vedli kabeláž co nejkratší nutnou vzdálenost, rozhodli jsme se jednotlivé řídicí prvky rozdělit jak je uvedeno v kapitole 5. Pro plošné spoje řídicích jednotek obou pohonů je k dispozici místo ve tvaru mezikruží, které je snadno přístupné po odstranění hliníkového krytu (viz. obr. 2.7).

Pro měření natočení obou kloubů jsou použity optické inkrementální snímače<sup>3</sup> (IRC). Snímače mají přesnost 50000 pulzů<sup>4</sup> na jednu otáčku. Inkrementální snímač elevačního kloubu je umístěn na opačné straně základny než je motor, jeho fotku můžete vidět takéž na obr. 2.7.

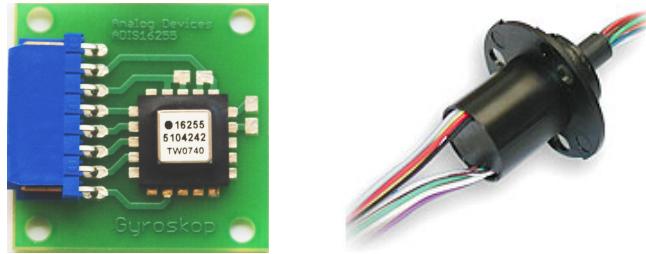


Obrázek 2.7. Vlevo místo pro umístění elektroniky řídicích jednotek přístupné po odstranění hliníkového krytu v horní části základny. Vpravo optický inkrementální snímač elevačního kloubu umístěný na opačné straně než je motor a místo pro průchod kabelů.

Pro měření pohybu vnitřního závěsu používáme senzor úhlové rychlosti ADIS 16255 od fy. Analog Devices (viz. obr. 2.8). Senzor je číslicový jednoosý gyroskop komunikující po sběrnici SPI. Obsahuje vnitřní registry umožňující senzor kalibrovat, čist aktuální hodnotu úhlové rychlosti, teploty, úroveň napájení, aj. Šířka pásma senzoru je 50 Hz a max. vzorkovací frekvence 255 Hz. Senzor obsahuje nastavitelný FIR filtr (filtr s konečnou impulzní odezvou). Je teplotně kalibrován a pracuje v rozsahu teplot od  $-25^{\circ}\text{C}$  do  $+85^{\circ}\text{C}$ .

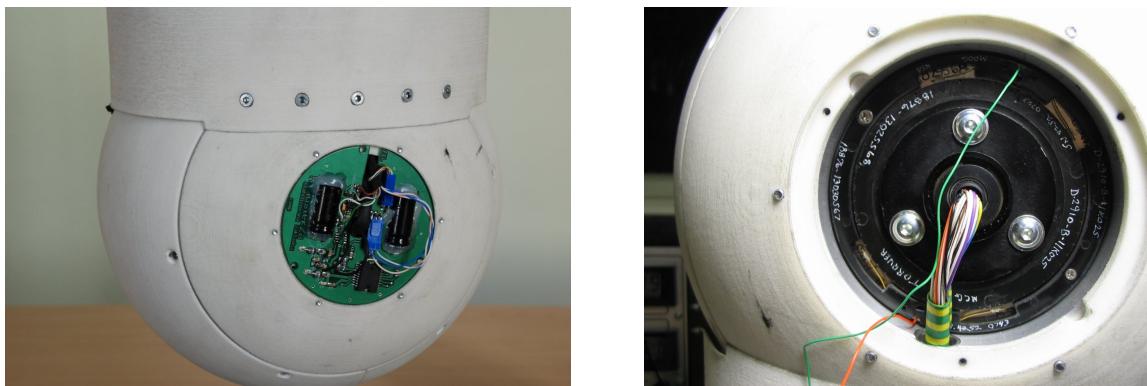
<sup>3</sup>Výrobcem snímačů je firma ESSA Praha.

<sup>4</sup>v případě kvadraturního zpracování signálu až 100 tis pulzů



Obrázek 2.8. Vlevo, fotka osazeného senzoru úhlové rychlosti ADIS 16255 (Velikost obrázku přibližně odpovídá reálným rozměrům senzoru). Vpravo, kluzný kroužek fy. MOOG.

Jako akční členy obou kloubů jsme použili motory D-2910 od fy. MOOG. Jedná se o motor typu direct drive. Jeho velkou výhodou je absence jakéhokoli typu převodovky. Vyhnuli jsme se tak problémům s vůlemi mezi převody a také třením. Je-li vnitřní závěs stabilizované základny dokonale vyvážen, dochází k potlačení velké části nežádoucích pohybů již díky samotnému hladkému uložení a velkému momentu setrvačnosti příslušenství. V případě použití pohonů s převody by se tato výhoda ztratila. Další výhodou motoru jsou jeho rozměry. Velký průměr motoru a malá axiální délka jsou velmi výhodné parametry pro použití v naší aplikaci. Z hlediska řízení se jedná o obyčejný kartáčový motor s permanentními magnety, který je možné řídit PWM signálem. Detail motoru s kluzným kroužkem uvnitř najeznete na obr. 2.9.



Obrázek 2.9. Vlevo umístění motoru v základně – motor je skryt pod H-můstkem. Vpravo detail motoru s kluzným kroužkem po odkrytí H-můstku.

Pro potřeby návrhu řízení bylo nutné sestavit stavový model motoru [9]. Matice v (2.1) jsou matice  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{D}$  lineárního spojitého stavového popisu se stavovými proměnnými proud motorem  $I[\text{A}]$ , úhlová rychlosť  $\omega[\text{°}/\text{s}]$  a poloha hrídele  $\varphi[\text{°}]$ . Vstupní proměnné jsou napětí  $U[\text{V}]$  a zatěžovací moment  $T[\text{Nm}]$ . Při určování matic  $\mathbf{A}$  a  $\mathbf{B}$  bylo nutné uvažovat nejen moment setrvačnosti rotoru (viz. tab.2.1), ale také moment setrvačnosti veškeré připojené zátěže, tedy celé základny. Ten byl určen konstruktérem základny za pomocí výpočetního softwaru Inventor, ve kterém byla celá konstrukce navržena.

Tabulka 2.1. Seznam základních parametrů motoru D-2910.

Parametr	Hodnota	Jednotka
Špičkový moment	59,2(0,42)	oz-in (Nm)
Počet pólů	10	
Elektrická časová konstanta	0,333	ms
Mechanická časová konstanta	19,68	ms
Moment setrvačnosti rotoru	$8,83 \cdot 10^{-5}$	$\text{kg} \cdot \text{m}^2$
Růst teploty s výkonem	3,8	$^{\circ}\text{C}/\text{W}$
Odpor vinutí	18	ohm
Indukčnost vinutí	6	mH
Napětí	26	V
Proud (špičkový)	1,44	A

$$\begin{aligned}
 \mathbf{A} &= \begin{bmatrix} -3000 & -48,33 & 0 \\ 1,47 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \mathbf{B} &= \begin{bmatrix} 166,7 & 0 \\ 0 & -49,78 \\ 0 & 0 \end{bmatrix} \\
 \mathbf{C} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \mathbf{D} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \\
 \mathbf{x} &= [ I[\text{A}] \quad \omega[^{\circ}/\text{s}] \quad \varphi[^{\circ}] ]^T & \mathbf{u} &= [ U[\text{V}] \quad T[\text{Nm}] ]^T
 \end{aligned} \tag{2.1}$$

# Kapitola 3

## Koncepce řídicího systému

Tato kapitola stručně popisuje koncepční schéma stabilizované základny. Jsou zde popsány její základní režimy činnosti, volba umístění senzorů a podrobněji vysvětleny důsledky nevhodnosti konstrukce z hlediska umístění singulárního bodu při pohledu přímo dolů pod letadlo.

### 3.1 Princip a základní možnosti stabilizace obrazu

Pro dosažení namíření středu kamery na pevně stanovený cíl je nutné realizovat stabilizaci optické osy (angl. Line of sight stabilizace, zkratka LOS). Cíl může být definován dvěma způsoby podle režimu činnosti stabilizované základny. Buď může jít o cíl daný zpracováním obrazu kamery nebo může být cíl určen souřadnicemi na zemském povrchu. Abychom stabilizace optické osy dosáhli, je nutné měřit pohyb vnitřního závěsu (pohyb kamery) nebo pohyb letadla. Podle toho rozlišujeme dva základní druhy stabilizace optické osy – přímou a nepřímou [12].

V případě přímé LOS stabilizace jsou senzory umístěny ve vnitřní závěsu stabilizované základny společně s kamery. Senzory úhlové rychlosti tak měří přímo pohyb kamery ve dvou osách (případně i třech osách pokud měříme i rotaci obrazu). Regulátor se pak snaží udržet nulovou úhlovou rychlosť obou senzorů a jedná se o přímou stabilizaci optické osy.

V případě nepřímé LOS stabilizace jsou senzory naopak umístěny v části základny umístěné v letadle a úhlové rychlosti kamery musí být doloženy přes transformační<sup>1</sup> rovnice (3.3). Podle [12] je však nepřímá LOS stabilizace méně výhodná zejména díky nutnosti transformace úhlových rychlostí, což s sebou nese spoustu dalších přídavných chyb, např. při měření relativního natočení jednotlivých kloubů. Rozhodli jsme se realizovat přímou stabilizaci optické osy.

Jelikož stabilizovaná základna je robotické zařízení se dvěma stupni volnosti (dvěma kloubů), není možné dosáhnout kompletní stabilizace namíření na cíl (dva úhly např. vůči zemi) a současně stabilizace rotace obrazu.

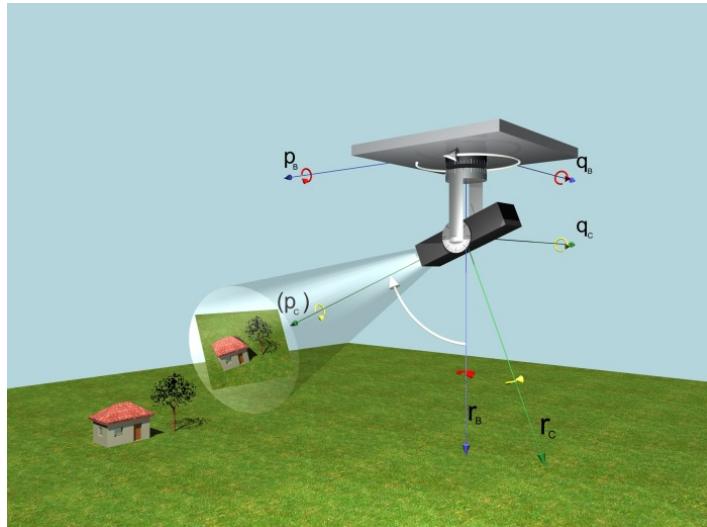
---

<sup>1</sup>V rovnici (3.3) se uvažuje svislý výsuv základny z letadla.

Chceme-li tedy přesně stabilizovat polohu optické osy (namíření středu kamery na daný cíl) je třeba přesně měřit rotace kamery okolo dvou kolmých os, které nazýváme „elevace“ a „příčná elevace“ (kolmo k elevační ose). Na obr. 3.1 je osa elevace označena zelenou šipkou  $q_c$  a osa příčné elevace zelenou šipkou  $r_c$ . Zbylá třetí osa rotace  $p_c$  je rotace obrazu na jejíž stabilizaci musíme bohužel rezignovat, protože vyrobená stabilizovaná základna obsahuje pouze dva stupně volnosti. Zde se tedy jedná o *stabilizaci optické osy*.

Druhá možnost stabilizace kamer se více uplatní při běžném pozorování, kdy tolik nezáleží na přesném namíření středu obrazu na danou pozici. V této situaci je výhodnější stabilizovat jiné dvě osy než osy elevace a příčné elevace. Dosáhne se tím stabilního chování v okolí singularity základny, kterého v případě stabilizace os elevace a příčné elevace není možné dosáhnout. Další výhodou tohoto režimu je skutečnost, že dochází k menší rotaci obrazu. Pro běžné pozorování je tak mnohem výhodnější stabilizovat osy, ve kterých jsou umístěny pohony. Jedná se tedy o osy „elevace“ a „azimutu“ (na obr. 3.1 je  $q_c$  elevace a  $r_b$  azimut). Tento režim stabilizace budu dále nazývat *stabilizace elevace-azimut*<sup>2</sup>.

Výsledný regulátor by tedy měl přepínat mezi těmito dvěma režimy v závislosti na tom, zda operátor požaduje přesné sledování určitého cíle (angl. tracking) nebo jde jen o běžné pozorování a nebo je stabilizovaná základna namířena poblíž singulárního bodu.

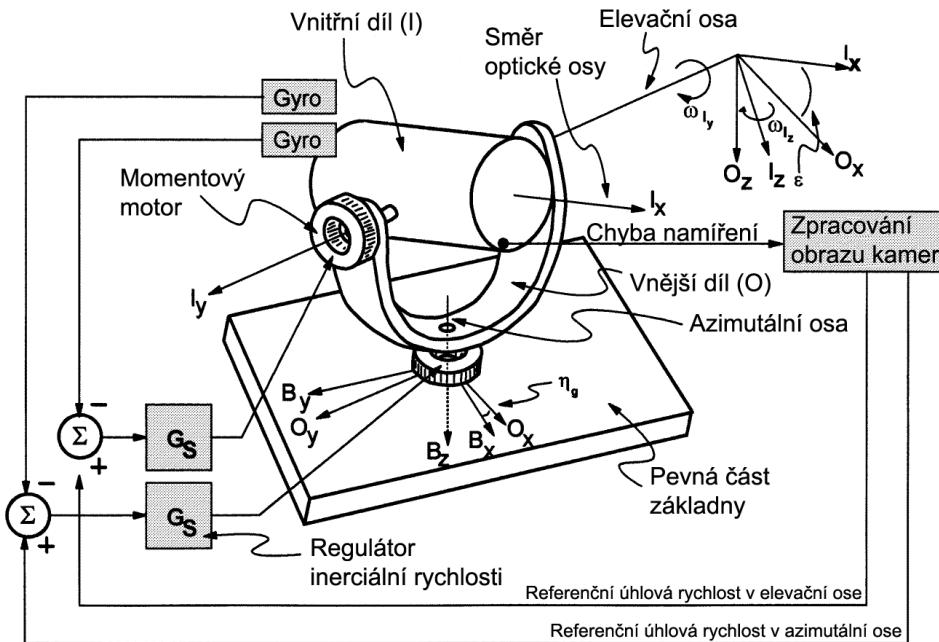


Obrázek 3.1. Souřadné soustavy v základně

<sup>2</sup>Při namíření kloubů do singularity se nabízí ve skutečnosti ještě jedna možnost, které osy stabilizovat. Je to stabilizace os elevace( $q_c$ ) a obrazu( $p_c$ ). Při namíření přesně do singularity jsou osy azimutu ( $r_b$ ) a obrazu ( $p_c$ ) totožné. Pokud bychom namířili optickou osu na horizont, pak osa azimutu ( $r_b$ ) je totožná s osou příčné elevace( $r_c$ ), avšak rotaci okolo osy obrazu ( $p_c$ ) při namíření na horizont není možné klouby základny ovlivnit – jedná se opět o singulární bod. Proto je výhodnější stabilizovat osy  $q_c$  elevace a  $r_b$  azimutu.

## 3.2 Kinematika stabilizované základny

Postupujeme-li podle [12, 10] je možné odvodit rovnice popisující kinematické závislosti jednotlivých souřadných systémů ve stabilizované základně. Obrázek obr. 3.2 znázorňuje tři souřadné systémy – vnitřní (I), ve kterém jsou umístěny kamery, vnější (O) a souřadný systém pevné části základny (B).



Obrázek 3.2. Konstrukční uspořádání a souřadné soustavy. Obrázek byl převzat z článku [12].

Význam jednotlivých úhlových rychlostí je následující:

- $\omega_{I_x}$  – úhlová rychlosť rotace obrazu,
- $\omega_{I_y}$  – úhlová rychlosť mērené elevace,
- $\omega_{I_z}$  – úhlová rychlosť mērené příčné elevace,
- $\omega_{O_z}$  – úhlová rychlosť senzoru azimutu,
- $\omega_{B_x}$  – úhlová rychlosť stranových kryv letadla (příčný sklon),
- $\omega_{B_y}$  – úhlová rychlosť podélného sklonu,
- $\omega_{B_z}$  – úhlová rychlosť azimutu letadla,
- $\eta$  – úhel azimutu základny (relativní úhel mezi vnějším závěsem a pevnou částí),
- $\varepsilon$  – úhel elevace základny (relativní úhel mezi vnitřním a vnějším závěsem).

Úhlové rychlosť pevné časti základny ( $\omega_B$ ) jsou svázány následující rovnicí s úhlovými rychlosťmi v souřadném systému vnějšího závěsu ( $\omega_O$ ).

$$\omega_O = R(\eta) \cdot \omega_B + [0 \ 0 \ \dot{\eta}]^T, \quad (3.1)$$

kde

$$R(\eta) = \begin{bmatrix} c_\eta & s_\eta & 0 \\ -s_\eta & c_\eta & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad s_\eta = \sin \eta; \quad c_\eta = \cos \eta;$$

Podobně úhlové rychlosti v souřadném systému vnějšího závěsu ( $\omega_O$ ) jsou svázány pomocí rovnice (3.2) s úhlovými rychlosti ve vnitřním závěsu (I).

$$\omega_I = R(\varepsilon) \cdot \omega_O + [ 0 \ \dot{\varepsilon} \ 0 ]^T, \quad (3.2)$$

kde

$$R(\varepsilon) = \begin{bmatrix} c_\varepsilon & 0 & -s_\varepsilon \\ 0 & 1 & 0 \\ s_\varepsilon & 0 & c_\varepsilon \end{bmatrix}; \quad s_\varepsilon = \sin \varepsilon; \quad c_\varepsilon = \cos \varepsilon;$$

Rozepíšeme-li jednotlivé rovnice (3.2) a (3.1) po složkách, dostaneme následující rovnice popisující transformace úhlových rychlostí pohybu letadla na úhlové rychlosti jednotlivých výše zmíněných senzorů

$$\begin{aligned} \omega_{I_x} &= \cos(\varepsilon) \cdot (\cos(\eta) \cdot \omega_{B_x} + \sin(\eta) \cdot \omega_{B_y}) - \sin(\varepsilon) \cdot (\omega_{B_z} + \dot{\eta}) \\ \omega_{I_y} &= -\sin(\eta) \cdot \omega_{B_x} + \cos(\eta) \cdot \omega_{B_y} + \dot{\varepsilon} \\ \omega_{I_z} &= \sin(\varepsilon) \cdot (\cos(\eta) \cdot \omega_{B_x} + \sin(\eta) \cdot \omega_{B_y}) + \cos(\varepsilon) \cdot (\omega_{B_z} + \dot{\eta}) \\ \omega_{O_z} &= \omega_{B_z} + \dot{\eta}. \end{aligned} \quad (3.3)$$

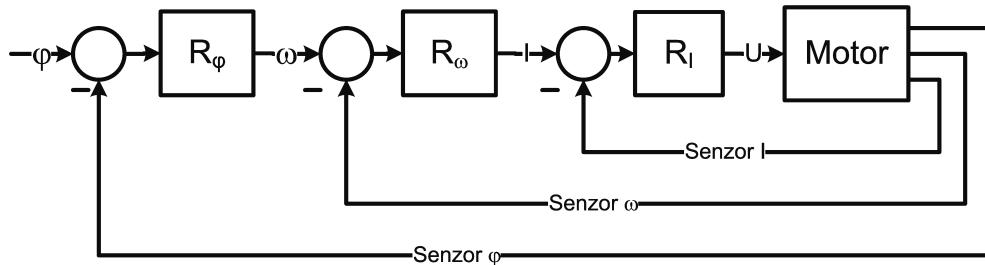
Hlavním úkolem regulátoru je v případě stabilizace optické osy udržet úhlové rychlosti  $\omega_{I_y}$  a  $\omega_{I_z}$  co nejmenší. Toho můžeme dosáhnout působením úhlových rychlostí jednotlivých kloubů  $\dot{\eta}$  a  $\dot{\varepsilon}$ . Podrobnějším průzkumem třetí rovnice z (3.3) však můžete vidět, že vliv jakým  $\dot{\eta}$  působí na  $\omega_{I_x}$  je úměrný konstantě  $\cos(\varepsilon)$ . Pokud se bude úhel elevace  $\varepsilon$  blížit k  $90^\circ$  nebude mít řízená úhlová rychlosť azimutu  $\dot{\eta}$  žádný vliv na měřenou rychlosť  $\omega_{I_z}$  a systém se stává neřiditelný. Tento stav systému je v kapitole 2 již zmíněný singulární bod, ve kterém není možné stabilizaci optické osy pomocí vyrobené kostrukce realizovat.

V teorii systému je situace, kdy stavové rovnice celého systému jsou v podstatě lineární a nelineární je pouze výstupní rovnice systému, dobře známá [16, 10]. Řešení je takové, že regulátor musí obsahovat nelinearitu opačnou takovou, aby původní nelinearitu vykrátil. Tedy regulátor musí obsahovat člen  $\frac{1}{\cos(\varepsilon)}$ . Prakticky však takový regulátor není schopný činnosti pokud se k singulárnímu bodu blížíme v úhlu  $\varepsilon$  o více než asi  $10 - 15^\circ$ . Vliv rychlosti otáčení azimutem  $\dot{\eta}$  je již příliš malý a v této situaci je požadavek na rychlosť otáčení motoru příliš velký a dochází k saturaci akčního zásahu. Dalším problémem je i skutečnost, že při tak rychlých pohybech azimutálním kloubem dochází k velké rychlosti rotace obrazu a tím i k jeho rozmazání. Navazující algoritmus zpracování obrazu potom není schopen činnosti s takto rozmazaným obrazem.

V případě stabilizace elevace-azimut je situace jednoduchá. Tato stabilizace se snaží udržet nulové úhlové rychlosti  $\omega_{I_y}$  a  $\omega_{O_z}$  působením kloubů  $\dot{\eta}$  a  $\dot{\varepsilon}$ . Vliv rychlostí kloubů na měřené rychlosti je však přímoúměrný.

### 3.3 Polohová, rychlostní, a proudová zpětná vazba

Regulátor pro většinu servosystémů se velmi často navrhuje kaskádním stylem [15], se třemi kaskádně řazenými regulátory proudu, rychlosti a polohy, jak je zobrazeno na obr. 3.3.



Obrázek 3.3. Kaskádní regulátor motoru

Toto rozdelení má velkou výhodu v tom, že je možné jednotlivé regulátory navrhovat odděleně. Ve stabilizované základně jsme tohoto rozdelení využili hlavně z důvodu možnosti přepínat regulátor polohy podle pracovního režimu, ve kterém se základna nachází (viz. níže). Přitom však regulátor rychlosti od senzorů úhlové rychlosti zůstává v obou režimech vždy stejný, snažící se co nejvíce potlačit pohyb (úhlové rychlosti) v elevační a příčné elevační ose.

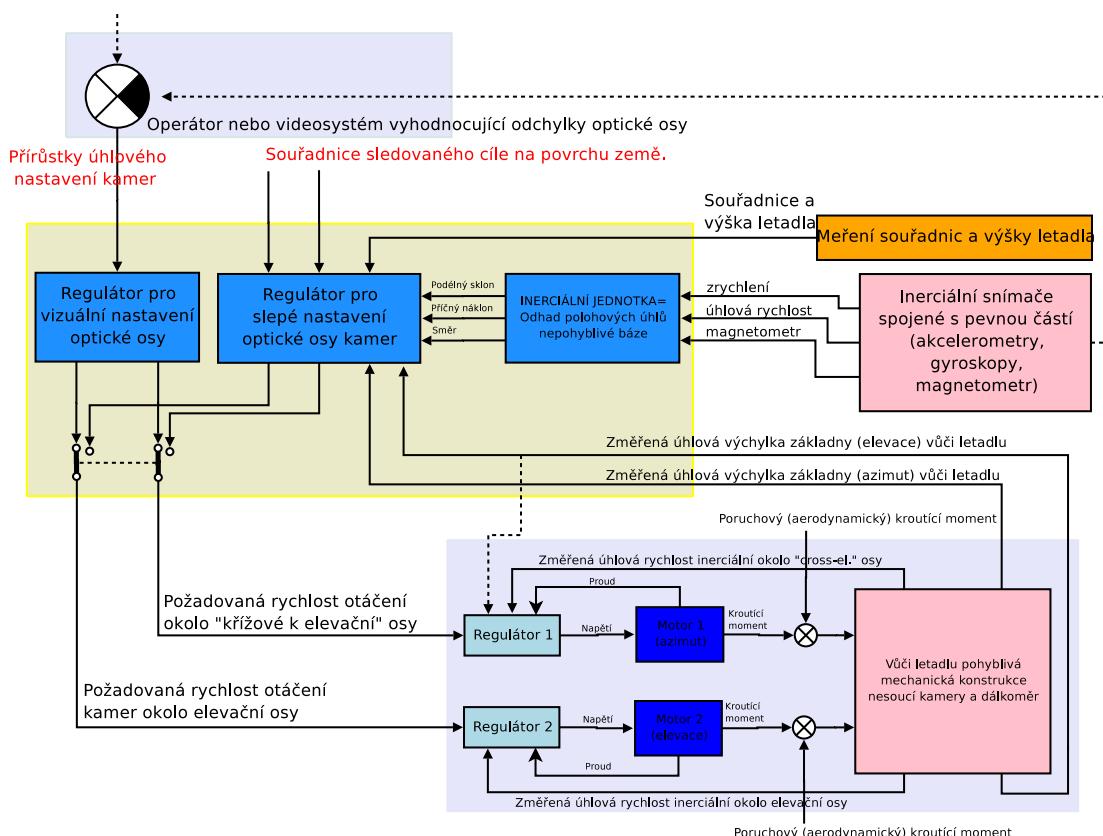
Velikost aktuálního momentu vytvářeného motorem je přímoúměrná proudu procházejícímu kotvou motoru. V případě řízení motoru napětím však není velikost proudu motorem přímoúměrná napětí, ale je zpožděna díky indukčnosti motoru (elektrická časová konstanta). Regulátor proudu (viz. obr. 3.3 a obr. 3.4 bloky **Regulator1** a **Regulator2**) se proto velmi často nastavuje tak, aby zpožďující vliv indukčnosti eliminoval. S nahlédnutím na stavové rovnice motoru (2.1) si můžete všimnout, že elektrická časová konstanta našeho motoru motoru je rovna asi 0,3 ms. Časová konstanta je tedy velmi malá (o řadu menší než perioda vzorkování). Z tohoto hlediska je možné při návrhu regulátor proudu zanedbat.

Druhým důvodem, proč se regulátor proudu zavádí je skutečnost, že při nenulových otáčkách motoru se na kotvě motoru indukuje určité napětí. Jsou-li tedy např. svorky motoru ve zkratu, prochází motorem určitý proud. Pokud svorky nejsou ve zkratu, ale je dokonce přiloženo napětí opačné polarity oproti indukovanému napětí kotvy (brzdění protiproudem), může snadno proud vzrůst nad velikost proudu potřebného pro maximální moment. Z hlediska proudové a výkonové ochrany motoru je proto velmi výhodné regulátor proudu zavést. Měření proudu motorem je tedy nutné v hardwaru realizovat.

Na obr. 3.4 můžete vidět blokové schéma činnosti základny. Jedná se o rozkreslení schématu z obr. 3.3 tak, aby odpovídalo funkčním blokům v základně. Dolní část blokového schématu (světle modrý obdélník) symbolizuje rychlostní zpětnou vazbu. Ta je složena, jak bylo již zmíněno, ze dvou senzorů úhlové rychlosti elevace a příčné elevace. Vstupem do této dvou zpětnovazebních smyček jsou požadované úhlové rychlosti z nadřazených regulátorů polohy. Přepínač v levé horní části schématu symbolizuje přepínání následujících dvou režimů polohového regulátoru.

Prvním režimem polohového regulátoru je *vizuální zpětná vazba*, tedy stabilizace od obrazového systému. Jedná se o hlavní úkol požadovaný v zadání projektu stabilizované základny (viz. kap. 1.1). Je důležité si uvědomit, že vzdálenost cíle od středu obrazu v pixelech ve svislé a vodorovné ose přímo odpovídá pohybu v osách elevace a příčné elevace. Tedy v případě použití P regulátoru se jedná jen o přepočet vzdálenosti v pixelech přes aktuální přiblížení kamery na úhel opravy a přenásobení konstantou P regulátoru. Algoritmus identifikace cíle ve snímku a tím i výpočtu polohové regulační odchylky je realizován kolegy na katedře kybernetiky pod vedením prof. Hlaváče. Regulátor polohy ve formě vizuální zpětné vazby pak využívá této regulační odchylky zamíření cíle v obraze na výpočet požadovaných akčních zásahů pro regulátory rychlosti obou os (viz. obr.3.4). Pro činnost režimu vizuální zpětné vazby proto bude nutná spolupráce mezi podsystémy řídicí části a zpracování videosignálu.

Druhým režimem polohového regulátoru je tzv. „slepé směrování“ (angl. blind pointing). Po přepnutí přepínače v obr.3.4 regulátor podle zadané souřadnice cíle na povrchu země a ze znalosti své polohy z inerciální jednotky a aktuálního natočení jednotlivých kloubů základny vypočte požadované úhlové rychlosti v osách elevace a příčné elevace. Ke své činnosti regulátor potřebuje znát polohu letadla z inerciální jednotky (viz. obr. 3.4).



Obrázek 3.4. Blokové schéma činnosti základny v režimu stabilizace optické osy

# Kapitola 4

## Výpočetní návrh řídicího systému

V předešlé kapitole byla představena základní koncepce řídicího systému stabilizované základny. Pro pochopení činnosti celého řídicího systému je velmi důležité zejména schéma na obr. 3.4 znázorňující dva režimy činnosti stabilizované základny (slepé směrování a vizuální zpětnou vazbu). Tato kapitola podrobně popisuje oba režimy.

Jak již bylo zmíněno, pro činnost algoritmu slepého směrování je důležité mít k dispozici přesný odhad polohy (6 stupňů volnosti) pevné části základny. V případě umístění stabilizované základny v bezpilotním prostředku MANTA řeší úkol odhadu polohy diplomová práce kolegy Tomáše Haniše [7]. Pro laboratorní testování činnosti stabilizované základny, kdy základna není v pohybu a nesplňuje dynamické rovnice pohybu letadla, je však tento algoritmus těžce aplikovatelný. Proto jsem se rozhodl zjednodušit model pro odhad polohových úhlů rozšířeným Kalmanovým filtrem tak, aby byl použitelný i mimo letadlo (i s patřičnými omezeními) a provedl implementaci algoritmu v jednočipovém mikropočítači ARM LPC2119. Tento algoritmus redukované inerciální jednotky (redukován proto, že odhaduje jen 3 st. volnosti ve formě polohových úhlů) umožní vyzkoušet algoritmus slepého směrování i v laboratorních podmírkách.

Pro simulaci chování, testování a návrh regulátorů byl vyvinut model stabilizované základny v režimu slepého směrování v prostředí simulink, který bude podrobněji popsán na konci této kapitoly.

### 4.1 Kaskádní uspořádání

Při simulačních pokusech s návrhem regulátorů jsem narazil na problém způsobený vyšší vzorkovací frekvencí vnitřní rychlostní smyčky než vnější polohové. Pomalá vnější smyčka poskytuje referenční signál pro vnitřní rychlou smyčku a tak je ZOH<sup>1</sup> approximace naprostě nedostačující, protože vnitřní smyčka téměř dokonale sleduje schodový průběh referenčního signálu vnější smyčky. Tento problém je možné vyřešit dvěma způsoby.

První možností, která nevyžaduje zásah do algoritmu polohového regulátoru, je proložit výstup regulátoru polohy mezi jednotlivými kroky lineární funkcí FOH<sup>2</sup> nebo počkat

---

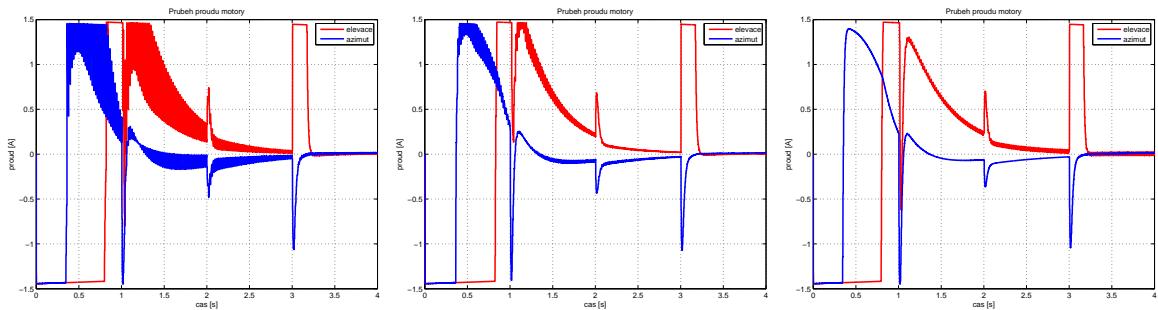
<sup>1</sup>zero order hold – tvarovač nultého řádu

<sup>2</sup>first order hold – tvarovač prvního řádu

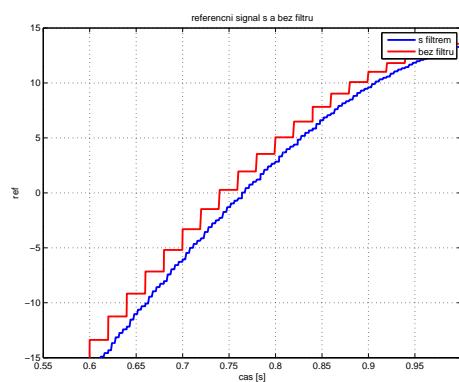
jeden krok vzorkovací periody vnější smyčky [4], převzorkovat signál na vyšší frekvenci vnitřní smyčky a přefiltrovat libovolným dolnopropustním filtrem.

Druhou možností je provést zásah do regulátoru polohy a přinutit jej, aby prováděl akční zásahy na frekvenci vnitřní smyčky. Jelikož však rychlejšího měření či výpočtu nových zásahů není regulátor schopen, může vyšší vzorkovací frekvence akčního zásahu dosáhnout jedině tím, že nahradí měření odhadem např. Kalmanovým filtrem z modelu. Modelem mohou být dynamické rovnice pohyb letounu (v případě slepého směrování), nebo model pohybu sledovaného objektu ve snímku (v případě vizuální zpětné vazby). Druhá možnost modelu pohybu objektu ve snímku může být navíc výhodná v situaci, kdy sledovaný cíl zmizí ze záběru úplně (např. při průjezdu pod stromy nebo mostem).

Následující obr. 4.1 ukazuje průběhy proudů (akčních zásahů) oběma motory v jednotlivých situacích, na obr. 4.2 je průběh schodovitého referenčního signálu.



Obrázek 4.1. Průběh proudů oběma motory. Vlevo bez použití převzorkování, uprostřed při použití převzorkování referenčního signálu na 250 Hz a filtrace dolní propustí, vpravo výsledky po úpravě regulátoru.



Obrázek 4.2. Referenční signály pro regulátory úhlové rychlosti s a bez použití převzorkování a filtrace.

## 4.2 Rychlostní zpětná vazba

Hlavním přínosem použití rychlostní zpětné vazby, navržené a realizované ve stabilizované základně je potlačení poruch nežádoucího pohybu letadla na mnohem vyšších frekvencích, než by bylo dosaženo pouze s použitím polohového regulátoru. Regulátor rychlosti elevace měří úhlovou rychlosť rotace okolo elevační osy a řídí elevační motor. Regulátor rychlosti příčné elevace měří úhlovou rychlosť příčné elevace a řídí azimutální motor. Oba regulátory jsou realizovány jako P/PI regulátory.

Použití integrační složky v regulátoru rychlosti je velmi žádoucí. Pokud totiž nedojde k dokonalému potlačení nežádoucího pohybu pomocí P složky regulátoru rychlosti (a k dokonalému potlačení poruchového pohybu nedojde nikdy), zapůsobí zde integrační složka regulátoru rychlosti jako P složka regulátoru polohy (ve skutečnosti nejde o polohu, ale integrál rychlosti), a způsobí dosažení původní polohy až na ustálenou regulační odchylku, kterou P regulátor způsobuje. Odstranění i této regulační odchylky by vyžadovalo použití dvojitě integrační složky což je již zbytečné, zvláště z toho důvodu, že potlačení regulační odchylky PI regulátoru rychlosti dosáhne polohový regulátor (vizuální zpětná vazba či slepé směrování).

Jedinou nevýhodou použití integrační složky v rychlostní zpětné vazbě je přítomnost ofsetu v senzoru úhlové rychlosti – gyroskopu AD 16255. Integrací tohoto ofsetu proto dochází k „ubíhání“ polohy optické osy, které lze odstranit jedině<sup>3</sup> polohovou zpětnou vazbou.

Hlavním rozdílem mezi regulátory elevace a azimu je ten, že azimutální motor podle předešlé kapitoly navíc obsahuje konstantu  $\frac{1}{\cos(\varepsilon)}$ , kde  $\varepsilon$  je aktuální úhel natočení elevačního kloubu<sup>4</sup>. S rostoucí hodnotou natočení elevačního kloubu tak roste člen  $\frac{1}{\cos(\varepsilon)}$  nad všechny meze a požadavky na rychlosť rotace azimutálním kloubem jsou obrovské. Stabilizovaná základna má proto sklon k nestabilitě pokud se pracovní poloha optické osy přibližuje k singularitě.

Aby k této nestabilitě nedocházelo<sup>5</sup>, je výhodné při namíření kamery poblíž singulárního budu automaticky vypnout stabilizaci optické osy a skokově přepnout na stabilizaci elevace-azimut, stabilizující místo rotace příčné elevace rotační pohyb azimu.

## 4.3 Vizuální zpětná vazba

Základní princip činnosti vizuální zpětné vazby již byl částečně popsán v kapitole 3. Vizuální zpětná vazba jako polohový regulátor využívá k výpočtu regulační odchylky namíření na cíl algoritmy vyvíjené kolegy na katedře kybernetiky pod vedením prof. Hlaváče. Jejich algoritmy jsou schopny po výběru určité části snímku tuto část najít i v

<sup>3</sup>Samozřejmě se počítá s kalibrací ofsetu senzoru, avšak ofset senzoru se v závislosti na poloze a čase mění, jak je podrobněji popsáno v kapitole 4.4.3

<sup>4</sup>Samozřejmý je rozdíl v použití mírně odlišných konstant z důvodu odlišných momentů setrvačnosti mechanické konstrukce v obou kloubech základny.

<sup>5</sup>Hlavice s původem výroby v Izraeli zmíněná v kap. 2 touto nestabilitou trpěla také i přesto, že obsahovala další dva vnitřní klouby.

dalsích snímcích a to i v případě, že určený cíl v po sobě jdoucích snímcích je otočen o určitý úhel. Výstupem těchto algoritmů jsou dvě vzdálenosti (vertikální a horizontální) sledovaného cíle od středu obrazu v pixelech. Spolu s údajem aktuálního přiblížení záběru dávají tyto tři údaje informaci o velikosti úhlových oprav v osách elevace a příčné elevace. Je totiž důležité si uvědomit, že rotace okolo os elevace resp. příčné elevace přímo odpovídá vertikálnímu resp. horizontálnímu pohybu obrazu kamery.

Podle již zmíněného obr. 3.4 vizuální zpětná vazba symbolizuje polohovou zpětnovazební smyčku, která nastavuje referenční úhlové rychlosti pro regulátory rychlosti popsané v předešlém odstavci. Pokud pro realizaci regulátoru polohy využijeme P regulátor je situace jednoduchá. Akční zásah – referenční úhlová rychlosť pro regulátor úhlové rychlosti elevace je získána jednoduše přenásobením hodnoty úhlové opravy ve svislém směru konstantou P regulátoru. Obdobným způsobem je určena referenční úhlová rychlosť příčné elevace.

Po konzultacích s kolegy realizující algoritmy zpracování obrazu kamer jsem se dozvěděl, že rychlosť zpracování obrazu bude s největší pravděpodobností v řádu jednoho až dvou snímků za sekundu. Vzorkování kompozitního obrazového signálu kamer a také přesuny výsledného obrazu mohou trvat až stovky milisekund. Bude tedy docházet k poněkud velkému dopravnímu zpoždění. Takové zpoždění by dokonce mohlo vést k nestabilitě zpětnovazební smyčky. Pokud by bylo zpoždění předem známé a vždy stejné, je možné realizovat tzv. Smithův regulátor [3]. V podstatě se jedná o to, že se známé zpoždění zahrne i do regulátoru vizuální zpětné vazby. Regulátor pak neprovádí akční zásah jen na základě zpožděného měření, ale i na základě modelu vývoje systému. Zpoždění však předem známé není a může i kolísat, protože hardware na kterém budou výpočty zpracování obrazu prováděny nezaručuje dobu odezvy.

V našem případě proto můžeme postupovat trochu jiným způsobem. Model vývoje systému ze Smithova regulátoru můžeme nahradit přímo měřeními vnitřní rychlostní smyčky. Požadavkem je znalost přesného času, kdy došlo ke sejmoutí snímku. Tato znalost by měla být k dispozici. Od té doby začne vnitřní rychlostní smyčka zaznamenávat pohyb kamer pomocí gyroskopů tak, že při příchodu nových úhlových oprav bude regulátor vědět, jak se již kamery během zpoždění zpracováním snímku pohybovaly. Obě informace je možné sloučit dohromady a vytvořit nový údaj úhlové opravy s respektováním pohybu kamer v době mezi pořízením snímku a doručením údaje úhlových oprav.

## 4.4 Slepé směrování

Původní zadání zadavatele projektu (VTÚL) neobsahuje požadavek na režim slepého směrování (angl. blind pointing). Primárním požadavkem je návrh a realizace algoritmu vizuální zpětné vazby. Tím se ale ztrácí velká výhoda celé stabilizované základny v možnosti určení souřadnic cíle, který kamery sledují, ale také možnosti namíření kamer na pozici zadanou podle známých souřadnic objektu např. v mapě.

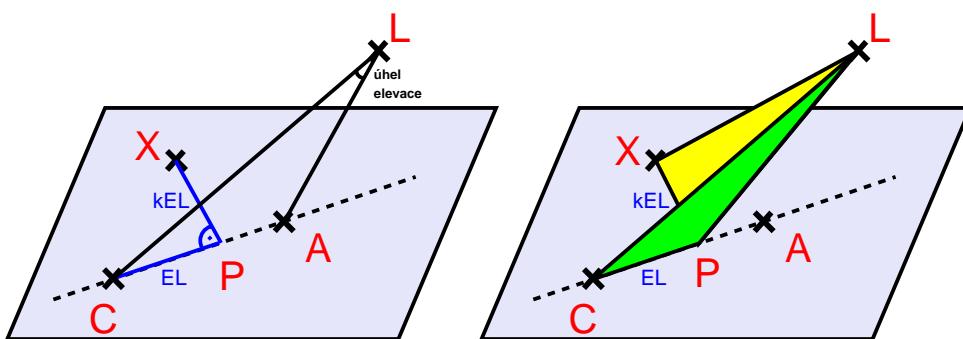
Regulátor polohy v režimu slepého směrování řeší problém jak nastavit referenční úhlovou rychlosť pohybu v osách elevace a příčné elevace tak, aby byly kamery správně

navedeny na zadaný cíl. Jedná se vlastně o problém inverzní kinematické úlohy s řešením odchylek zaměření v úhlech elevace a příčné elevace.

#### 4.4.1 Princip slepého směrování

Existuje několik možností jak problém vyřešit.

První z nich je analytická metoda, která ze znalosti cílové souřadnice, natočení letadla a natočení kloubů základny určí pomocí analytické geometrie o jaký úhel je potřeba otočit kamerou v elevaci a o jaký v příčné elevaci (pozor, skutečně se nejdá se o klouby základny, ale úhly ve zmíněných osách). Výsledné úhly vynásobené konstantou P regulátoru jsou požadované referenční rychlosti. Na obr. 4.3 je schéma symbolizující postup výpočtu.



Obrázek 4.3. Princip analytické metody slepého směrování

**L** symbolizuje letadlo

**A** místo na zemském povrchu kam míří kamery při nulovém vychýlení elevačního kloubu (jakoby byly kamery namířeny do singularity)

**C** místo na zemském povrchu kam jsou aktuálně kamery namířeny

**X** cílová souřadnice v rovině země, která má být zaměřena

**P** kolmý průsečík přímky AC s bodem X.

Po přímce AC se pohybuje místo kam míří kamery otáčením okolo elevační osy. Kolmo na tuto přímku se pohybuje obraz rotací okolo příčné elevační osy. Úkolem je tedy určit úhly CLP (odchylka v elevaci – uvnitř zeleného trojúhelníku) a XLP (odchylka v příčné elevaci – uvnitř žlutého trojúhelníku). Tyto úhly je možné určit se základní znalostí analytické geometrie. Přenásobením úhlů CLP a XLP konstantou P regulátoru dostaneme požadované akční zásahy pro regulátory rychlosti. Celý tento algoritmus provádí matlabská f-ce `position_regul`, kterou najdete v přiloženém CD.

První metoda má obrovskou nevýhodu pokud dojde k namíření kamery na či nad horizont. V takovém případě neexistuje průsečík s rovinou Země a algoritmus nefunguje. Tento

problém jsem se pokusil vyřešit jiným způsobem výpočtu obou požadovaných úhlů za pomocí rotačních matic. Cílová souřadnice pak není zadána jako bod na rovině Země, ale jako úhly azimutu a elevace stabilizované základny v letadle jakoby rovnoměrně letícího v konstantní výšce. Takto je potom možno zadat cílovou souřadnici i na horizontu volbou vstupní elevace rovné  $90^\circ$ . Algoritmus je postaven na následující rovnici.

$$R_{aircraft} \cdot R_{joints} \cdot R_{EL,CEL} = R_{target}(\varphi_{roll}), \quad (4.1)$$

kde

$R_{aircraft}$  je rotační matice popisující polohu letadla. Může být sestavena např. ze znalosti Eulerových úhlů či quaternionů letadla.

$R_{joints}$  je rotační matice popisující rotaci dvěma klouby základny.

$R_{target}(\varphi_{roll})$  je cílová souřadnice definovaná jako poloha kamery pomocí rotační matice.

Tato matice je závislá na proměnné – rotaci obrazu ( $\varphi_{roll}$ ), již není možné explicitně specifikovat.

$R_{EL,CEL}$  je hledaná rotační matice skrývající v sobě rotace okolo os elevace a příčné elevace.

Pokud rozepíšeme matici  $R_{target}$  po jednotlivých složkách dostaneme následující vztahy.

$$\begin{aligned} R_{target}(\varphi_{roll}) &= R_{\varphi_{az}} \cdot R_{\varphi_{el}} \cdot R_{\varphi_{roll}} \\ &= \begin{bmatrix} \cos \varphi_{az} & \sin \varphi_{az} & 0 \\ -\sin \varphi_{az} & \cos \varphi_{az} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \varphi_{el} & 0 & -\sin \varphi_{el} \\ 0 & 1 & 0 \\ \sin \varphi_{el} & 0 & \cos \varphi_{el} \end{bmatrix} \cdot \begin{bmatrix} \cos \varphi_{roll} & \sin \varphi_{roll} & 0 \\ -\sin \varphi_{roll} & \cos \varphi_{roll} & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ R_{target}(\varphi_{roll}) &= \begin{bmatrix} f_{11}(\varphi_{roll}) & f_{12}(\varphi_{roll}) & \cos \varphi_{az} \cdot \sin \varphi_{el} \\ f_{21}(\varphi_{roll}) & f_{22}(\varphi_{roll}) & \sin \varphi_{az} \cdot \sin \varphi_{el} \\ f_{31}(\varphi_{roll}) & f_{32}(\varphi_{roll}) & \cos \varphi_{el} \end{bmatrix} \end{aligned} \quad (4.2)$$

V rov. (4.2) si můžete povšimnout, že poslední sloupec transformační matice není závislý na proměnné  $\varphi_{roll}$ . Ostatní prvky matice jsou díky závislosti na  $\varphi_{roll}$  nezajímavé. K obdobnému vztahu dojdeme při rozepsání matice  $R_{EL,CEL}$  po složkách (viz. rov. (4.3)).

$$R_{EL,CEL}(\varphi_{roll}) = \begin{bmatrix} g_{11}(\varphi_{roll}) & g_{12}(\varphi_{roll}) & \cos \varphi_{CEL} \cdot \sin \varphi_{EL} \\ g_{21}(\varphi_{roll}) & g_{22}(\varphi_{roll}) & -\sin \varphi_{CEL} \\ g_{31}(\varphi_{roll}) & g_{32}(\varphi_{roll}) & \cos \varphi_{CEL} \cdot \cos \varphi_{EL} \end{bmatrix} \quad (4.3)$$

Po úpravě rov.(4.1) do následujícího tvaru

$$R_{EL,CEL} = R_{joints}^{-1} \cdot R_{aircraft}^{-1} \cdot R_{target}(\varphi_{roll})$$

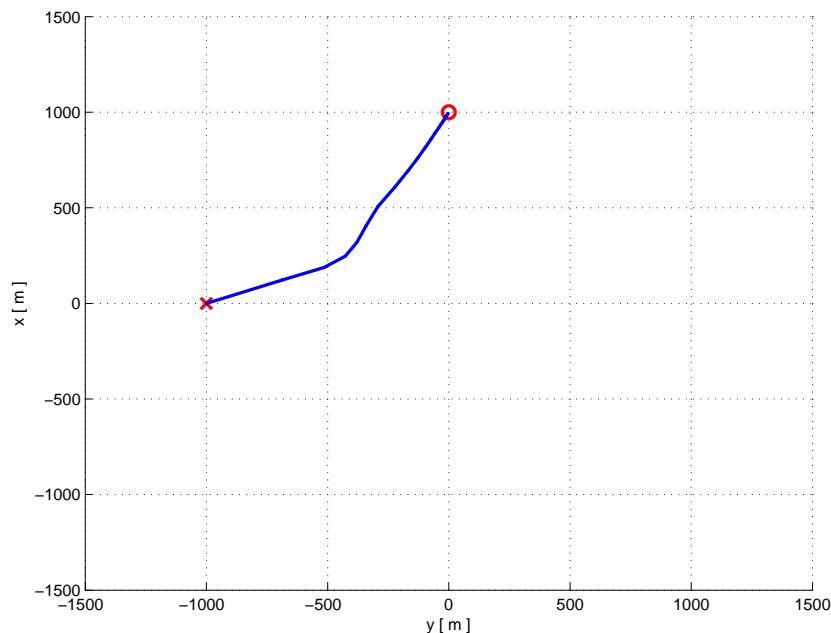
a s uvážením, že poslední sloupce obou stran rovnice jsou nezávislé na rotaci obrazu  $\varphi_{roll}$  je možné určit oba požadované úhly elevace ( $\varphi_{EL}$ ) a příčné elevace ( $\varphi_{CEL}$ ) potřebné pro natočení na daný cíl.

Tento nyní popsaný druhý algoritmus provádí f-ce `position_regul_matrix`, která je také uvedena v přiloženém CD.

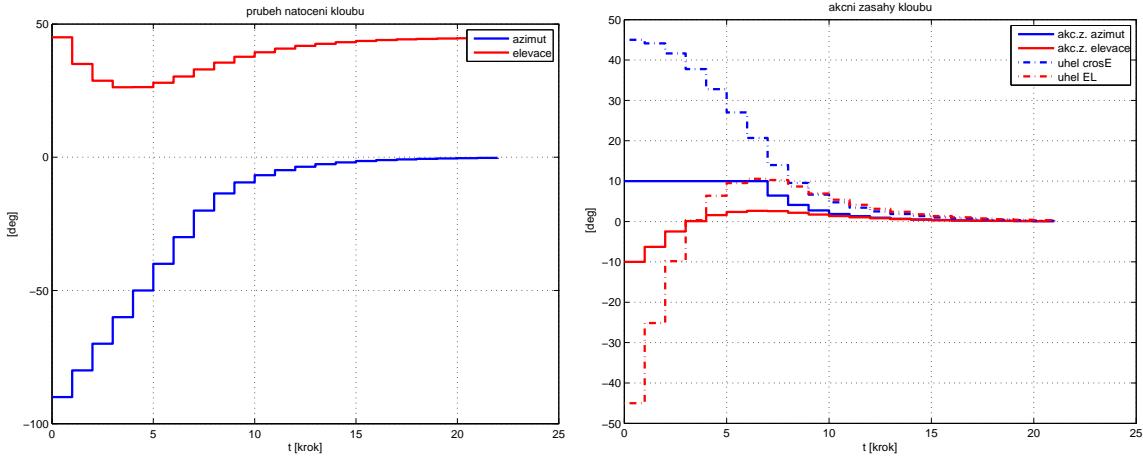
Pro testování obou algoritmů jsem sestavil matlabskou funkci `reguluj.m`, jejíž výstup můžete vidět na obr. 4.4. Tato funkce provádí simulaci regulátoru polohy s využitím výše zmíněné f-ce `position_regul`. Funkce však zanedbává veškerou dynamiku základny i pohonů. Její výstup je třeba brát pouze symbolicky. Vstupem funkce je quaternion popisující naklopení letadla, úhly jednotlivých kloubů stabilizované základny ve stupních, a cílová souřadnice. Obrázek byl vytvořen následujícím příkazem.

```
reguluj(-90,45,1000,euler2quaternion(0,0,0),[1000 0])
```

Letadlo je tedy v rovnomořném přímočarém letu – Eulerovy úhly jsou nulové. Výška letu je 1000 m. Azimut základny je  $90^\circ$  doleva a elevace  $45^\circ$ . Cílová souřadnice je nastavena 1000 m dopředu (v grafu červené kolečko nahore). Červený křížek značí bod, na který jsou kamery namířeny v počátku simulace.



Obrázek 4.4. Rovina země s body, které říkají kam byla kamera v čase namířena. Počátek souřadnic – bod  $[0,0]$  je poloha letadla.



Obrázek 4.5. Vlevo, průběh natočení jednotlivých kloubů. Vpravo, výstupy f-ce `position_regul` a omezené akční zásahy – úhlové rychlosti.

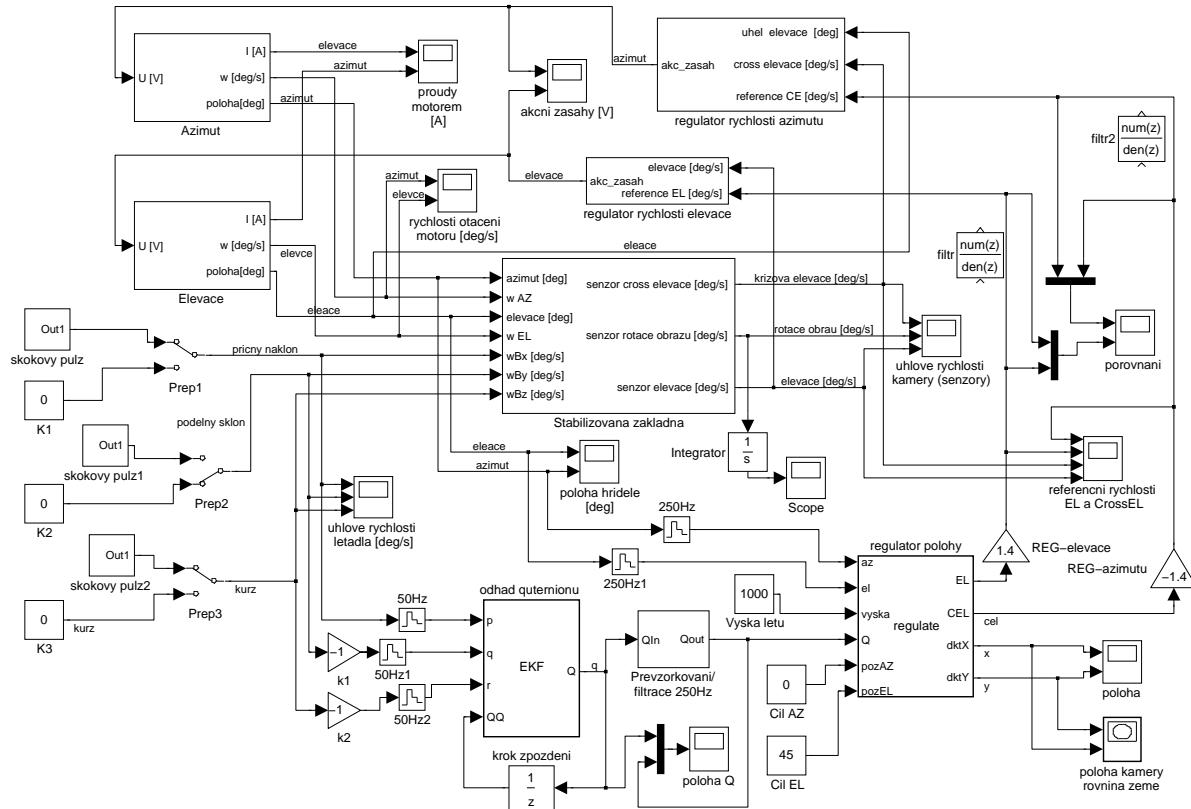
Jak můžete vidět na obr. 4.4 algoritmus navádí základnu vždy ve směru gradientu k cílové souřadnici. Nejedná se tedy, jak by někdo mohl čekat, o nejkratší cestu přes jednotlivé klouby robotického zařízení. Nejkratší cesta přes jednotlivé klouby by totiž vyžadovala vypnutí vnitřní rychlostní zpětné vazby.

Hlavní cíl tohoto algoritmu je tak spíše míněn ne jako rychlé zamíření na zadaný cíl, ale spíše jako polohová zpětná vazba, která upravuje nedokonalosti rychlostní zpětné vazby v případě, kdy je již nějaký cíl zaměřen. Pro rychlé přestavování kloubů je opravdu nejjednodušší využít lokálních polohových smyček pro každý kloub zvlášť a to i za cenu, že během přestavování kloubů nebude v činnosti rychlostní zpětná vazba.

#### 4.4.2 Model v simulinku

Funkce `reguluj` provádí simulaci polohového regulátoru základny, avšak úplně zanedbává dynamiku základny a pohonů. Proto jsem vyvinul model stabilizované základny v režimu „stabilizace optické osy – slepé směrování“ v Simulinku. V modelu jsem již uvažoval dynamiku pohonů, momenty setrvačnosti konstrukce i omezení akčních zásahů. Stavové rovnice motoru viz. (2.1). Výtisk obrazovky modelu můžete vidět na obr. 4.6, případně ve větším měřítku v příloze D této diplomové práce. V modelu je možné sledovat průběhy proudů motorem, měřené úhlové rychlosti senzorů, akční zásahy či aktuální polohu hřídelů. Je možné nastavit úhlové rychlosti definující pohyb letadla a tak použít např. přímo naměřená data z letu. Dále model umožňuje nastavit počáteční natočení kloubů základny a souřadnice cíle (ve formě azimutu a elevace), který má být zaměřen.

Zdrojové soubory modelu najeznete na přiloženém CD. Počáteční podmínky simulace jsou nastaveny tak, že kamera je nasměrována  $90^\circ$  v azimutálním kloubu a  $45^\circ$  elevačním kloubu. Vstupem simulace je rotační pohyb okolo osy letu letadla (příčný sklon). Nejprve po dobu 1 s rychlostí  $20^\circ/\text{s}$  jedním směrem a poté stejnou dobu směrem opačným. Pohyb okolo ostatních os je nastaven jako nulový. Model je nutné spouštět příkazem `kin`, který provádí počáteční nastavení (je možno změnit pro vlastní potřebu).



Obrázek 4.6. Výtisk obrazovky simulinkového modelu základny v režimu „stabilizace optické osy – slepé směrování“. Stejný obrázek ve větším rozlišení naleznete v příloze D této diplomové práce.

Model se skládá z několika bloků jejichž funkce je následující:

**Stabilizovaná základna** – realizuje transformace (3.3) vnějších úhlových rychlostí na vnitřní úhlové rychlosti gyroskopů (elevace, příčné elevace, příp. rotace obrazu). Blok obsahuje 3 vstupy úhlových rychlostí letadla (wBx, wBy, wBz), 2 vstupy aktuálního natočení kloubů základny a 2 vstupy úhlových rychlostí kloubů základny. Výstupem jsou úhlové rychlosti gyroskopů umístěných na kamerách. Jako vstupní úhlové rychlosti pohyby letadla mohou být použity např. data reálného letu.

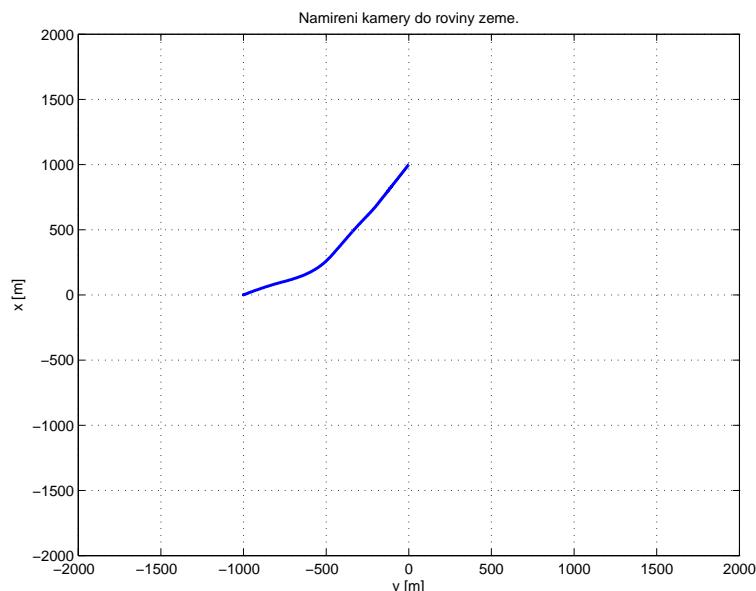
**Azimut, Elevace** – modely chování motorů. Při tvorbě modelů jsem využil dokumentace výrobce, která obsahuje důkladný popis všech parametrů motoru viz. kap. 2.4.

**Regulátory elevace a azimutu** – jsou P/PI regulátory inerciální úhlové rychlosti popsané v kap. 4.3. Regulátory se snaží držet referenční úhlovou rychlosť pohybu okolo elevační a příčné elevační osy. Referenční rychlosť je dána nadřazeným regulátorem polohy.

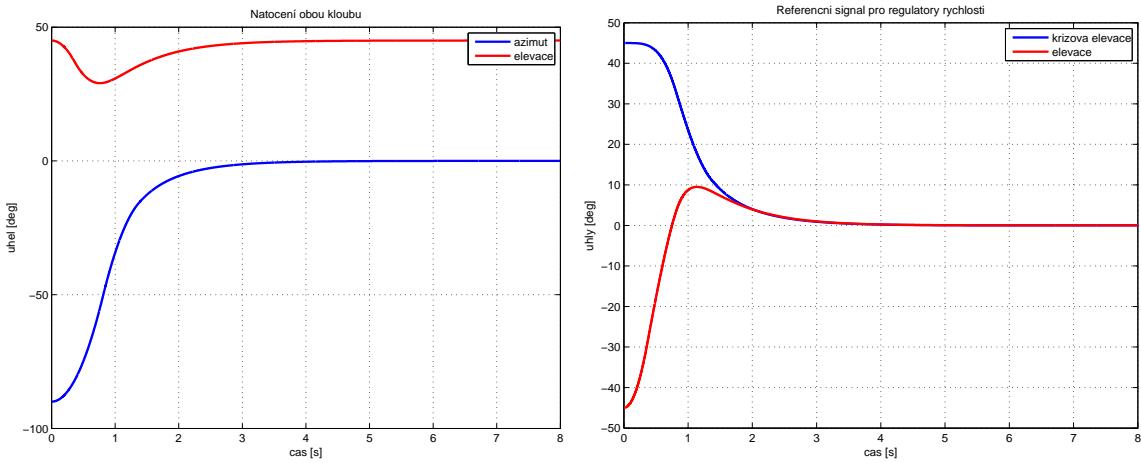
**Odhad quaternionu** – inerciální jednotka – z naměřených dat úhlových rychlostí, zrychlení, aj. se snaží získat co nejpřesnější údaj o poloze letadla (pevné části základny). Pro jednoduchost však v tomto bloku počítám s tím, že měřené úhlové rychlosti pohybu letadla neobsahují ofsety. Odhad tak degraduje na čistou integraci úhlových rychlostí do quaternionu. Podrobnější popis úkolu inerciální jednotky naleznete v následující podkapitole. 4.4.3.

**Regulátor polohy** – na základně aktuální polohy letadla (pevné části základny), nařízení obou kloubů a požadované souřadnice cíle nastavuje referenční úhlové rychlosti pro regulátory rychlosti. Realizuje algoritmus polohového regulátoru slepého směrování popsaného v předešlé podkapitole 4.4.

Na následujícím obr. 4.7 můžete vidět průběh navádění na zadaný cíl se stejnými podmínkami jako na obr. 4.4. jen s tím rozdílem, že je uvažována dynamika stabilizované základny a čas.



Obrázek 4.7. Stejná situace jako na obr. 4.4, avšak nyní již s uvažováním dynamiky základny. Graf je vykreslen pro možnost porovnání s obr. 4.4. Tzn. že nejsou uvažovány výše zmíněné poruchy pohybu letadla.



Obrázek 4.8. Stejná situace jako na obr. 4.5, avšak nyní již s uvažováním dynamiky základny. Vlevo, průběh natočení obou kloubů. Vpravo, vypočtené úhly elevace a příčné elevace pro regulátory úhlové rychlosti. Graf je vykreslen pro možnost porovnání s obr. 4.5, nejsou tedy uvažovány výše zmíněné poruchy pohybu letadla a úhly elevace jsou vykresleny s opačnými znaménky oproti průběhům v příloze D.

Průběhy simulace i s uvažováním pohybu letadla zmíněného v úvodu této podkapitoly najeznete bez komentářů v příloze D této diplomové práce. Kromě polohy kloubů a akčních zásahů obsahují simulace v příloze i např. průběhy proudů motory a průběhy signálů senzory.

#### 4.4.3 Redukovaná inerciální jednotka s EKF (rozšířený Kalmanův filtr)

Pro realizaci zmíněného algoritmu slepého směrování je důležité mít k dispozici inerciální jednotku, která umí odhadnout polohové úhly letadla. Situací kdy jsou senzory umístěny na palubě letadla MANTA se zabývala po teoretické stránce diplomová práce [7]. Motivací této podkapitoly je úkol odhadu polohových úhlů pevné části stabilizované základny, který by umožnil činnost algoritmu slepého směrování optické osy i během laboratorního testování hlavice. V této situaci totiž pohyb pevné části základny nesplňuje dynamické rovnice pohybu letadla. Rozhodnul jsem proto navrhnout a také realizovat redukovanou inerciální jednotku, která je schopna odhadnout polohové úhly objektu vůči zemi (poloha v prostoru odhadována není).

V poslední době velmi klesla cena a dostupnost MEMS gyroskopů a akcelerometrů. Měl jsem tu možnost setkat se s kombinovaným tříosým gyroskopem/akcelerometrem ADIS 16350 od firmy Analog Devices. Senzor umí pracovat na vzorkovací frekvenci až 255Hz, data jsou vyčítána přes rozhraní SPI.

Hlavní nevýhodou použití MEMS gyroskopů obecně je jejich nemalý offset. Vzorek gyroskopu, který jsem měl k dispozici (viz. obr. 4.9) dosahuje offsetu až  $0,8^\circ/s$ , navíc se hodnota offsetu mění v závislosti na poloze (natočení) gyroskopu v prostoru a také v závislosti na teplotě a čase (viz. obr. 4.10). Integrujeme-li tak měřenou úhlovou rychlosť

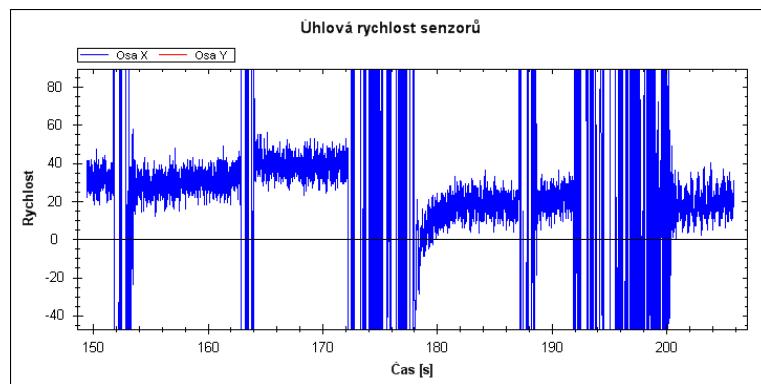
bez jakékoli kompenzace ofsetu dojde za necelé čtvrt hodiny k chybě určení polohy celých  $360^\circ$ .

Při hledání v odborné literatuře jsem narazil na několik zajímavých článků [11, 5] jak se s problémem ofsetu gyroskopů vyrovnat kompenzací měřené rychlosti pomocí měření vektoru gravitačního zrychlení, případně měřením magnetického pole země. Jednoduše řečeno, pokud opravdu dochází k nějaké rotaci, kterou gyroskopy měří, musí se vše projevit i měřením gravitačního zrychlení resp. magnetického pole. Oba zmíněné články popisují jak účinně skládat dohromady údaje ze všech senzorů pomocí rozšířeného Kalmanova filtru (angl. extended Kalman filter – EKF).

Hlavním cílem tohoto úkolu, jak již bylo částečně zmíněno v úvodu kapitoly, je navrhnut a implementovat v jazyce C algoritmus rozšířeného Kalmanova filtru, který by byl schopný kompenzovat účinky ofsetu gyroskopů při jejich integraci v reálném čase v procesoru LPC 2119. Realizace rozšířeného kalmanova filtru v jazyce C je pak spojena s úkoly implementace maticového násobení a inverze pozitivně definitní matice [17].



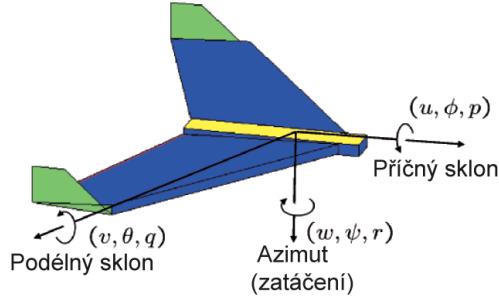
Obrázek 4.9. Vlevo deska s kombinovaným tříosým gyroskopem/akcelerometrem ADIS 16350. Vpravo propojení s deskou SPIBoard.



Obrázek 4.10. Na obrázku můžete vidět jak se mění ofset gyroskopu ADIS 16350 v závislosti na poloze senzoru. Senzor je postupně položen v klidu na stole v šesti různých polohách a ustálená hodnota ofsetu je pokaždé jiná. Jednotka úhlové rychlosti závislé osy grafu je daná rozlišením senzoru. Hodnota ofsetu odpovídá asi  $0,5^\circ/\text{s}$ .

#### 4.4.3.1 Měřené veličiny, polovodičové gyroskopy a akcelerometry

Pro realizaci algoritmu musí být senzor ADIS 16350 umístěn tak, aby gyroskopy měřily jednotlivé úhlové rychlosti  $p, q, r$  podle obr. 4.11.



Obrázek 4.11. Označení měřených rotací okolo jednotlivých os letadla (objektu).

Výstup MEMS gyroskopu popisuje podle [14] následující rovnice

$$y_{gx} = k_x \cdot \omega_{gx} + \eta_{gx} + B_{gx},$$

kde  $x$  značí gyroskop v x-té ose,  $y_{gx}$  značí měřenou výstupní rychlosť gyroskopu,  $k_x$  převodní konstantu z úhlové rychlosti,  $\omega_{gx}$  skutečnou úhlovou rychlosť x-tého senzoru,  $\eta_{gx}$  šum senzoru s nulovou střední hodnotou a  $B_{gx}$  offset senzoru. Jelikož převodní konstanta  $k_x = 1$  je u každého senzoru známá, případě je možné ji změřit, uvažují pro jednoduchost  $k_x = 1$ .

Výstup MEMS akcelerometru charakterizuje následující rovnice

$$y_{ax} = k_x \cdot a_x + \eta_{ax} + B_{ax}, \quad (4.4)$$

kde  $x$  značí gyroskop v x-té ose,  $y_{ax}$  značí měřený výstup x-tého akcelerometru,  $k_x$  převodní konstantu ze zrychlení,  $a_x$  je skutečné zrychlení. Pro jednoduchost opět uvažují převodní konstantu  $k_x = 1$ , avšak offset akcelerometru uvažují nulový  $B_{ax} = 0$ . Offset akcelerometrů není tedy odhadován ani kompenzován. To vede na nepříznivé vlivy v případě jeho přítomnosti.

V případě akcelerometru je možné se podívat podrobněji na jednotlivé složky skutečného zrychlení. Pro mě je užitečná jen složka gravitačního zrychlení a ostatní složky (vnější zrychlení) musí považovat za nulové, i když tomu tak v mnoha případech být nemusí, nebo je kompenzovat (odstředivé zrychlení). Rozpis jednotlivých složek zrychlení měřeného akcelerometry ukazuje rovnice 4.5.

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \frac{1}{m} \cdot (F - F_{gravity}) = \dot{v} + \omega \times v - \frac{1}{m} \cdot F_{gravity} \quad (4.5)$$

Vektorový součin  $\omega \times v$  charakterizuje odstředivé zrychlení. Jsou-li tedy gyroskopy oproti akcelerometrům posunuty, vzniká zde dodatečná chyba, která může být při znalosti

tohoto posunutí kompenzována pomocí rovnice 4.6. S využitím vztahu pro odstředivé zrychlení  $a = \omega^2 r$  je možné celou rovnici zapsat po složkách také ve tvaru

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \dot{u} - (q^2 + r^2) \cdot X_{A_x} + g \cdot \sin \theta \\ \dot{v} - (p^2 + r^2) \cdot X_{A_y} - g \cdot \cos \theta \sin \phi \\ \dot{w} - (p^2 + q^2) \cdot X_{A_z} - g \cdot \cos \theta \cos \phi \end{bmatrix}. \quad (4.6)$$

V této rovnici jsou použity z důvodu snazší interpretace Eulerovy úhly  $(\phi, \theta, \psi)$  namísto quaternionů použitých modelu. Výsledná rovnice již s quaterniony je součástí rovnice 4.10. Ta však zanedbává výše zmíněné odstředivé zrychlení a samozřejmě účinky vnějších zrychlení, které do modelu vstupují jako poruchy – šum senzoru. Účinky odstředivého zrychlení jsem v dalším postupu neuvažoval, protože u použitého kombinovaného senzoru ADIS 16350 jsou akcelerometry i gyroskopy umístěny ve stejném bodě (uvnitř senzoru) a vzájemné odstředivé zrychlení se tak neprojeví.

#### 4.4.3.2 Reprezentace polohy - quaterniony vs. Eulerovy úhly

Měřené úhlové rychlosti okolo tří os tělesa (značí se  $p, q, r$  viz obr. 4.11) je možné „integrovat“ buď pomocí rovnice (4.8) do Eulerových úhlů nebo je možné využít quaternionů [1]. Pro laické přiblížení je quaternion čtyřprvkový vektor s jednotkovou absolutní hodnotou, jehož tři složky definují osu rotace a zbylá složka definuje úhel rotace okolo této osy. Ve skutečnosti se však nejedná o vektory, protože quaterniony definují operace pravého i levého dělení, které vektory nemají. Pro bližší pochopení quaternionů doporučuji webové stránky serveru *Euclidean Space* [1]. Pro pochopení principu modelu však není nutné znát detaily quaternionovské algebry.

V případě použití quaternionů k popisu polohy ukazuje stavová rovnice (4.7) vývoj aktuální polohy jako integraci úhlových rychlostí  $p, q, r$  (viz. rovnici 1.3-36 v [13]).

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (4.7)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (4.8)$$

Z porovnání obou možností (rovnic 4.7 a 4.8) plyne jednoznačná výhoda reprezentace pomocí quaternionů. Rovnice popisující vývoj Eulerových úhlů je značně složitá, navíc obsahuje dokonce i podíly. Pro určité hodnoty Eulerových úhlů je tedy dokonce singulární. V případě použití rozšířeného Kalmanova filtru je nutné určit derivace pravých stavových rovnic pro sestavení jakobiánu. Derivace pravé strany rovnice 4.8 je velmi složitá a dosazování naměřených hodnot do jakobiánu je v případě realizace výsledného algoritmu v jednočipovém počítači téměř nepředstavitelné.

#### 4.4.3.3 Stavový model použitý pro odhadování

Při sestavování stavového modelu jsem se inspiroval v [11] a [5]. Pro začátek jsem se rozhodl pro použití tří (jednoosých) gyroskopů a jednoho (tříosého) akcelerometru. Rovnice popisující chování akcelerometrů a gyroskopů jsou výstupními rovnicemi modelu. Model tedy neobsahuje žádné vstupy, je založený na rovnici (4.7). Kromě těchto čtyř stavů jsou stavovými proměnnými ještě 3 ofsety gyroskopů ( $B_p, B_q, B_r$ ) a úhlové rychlosti  $p, q, r$ . Ofset gyroskopu jsem modeloval jako tzv. náhodnou procházku [18]. Úhlové rychlosti ( $p, q, r$ ) jsem modeloval jako systém prvního řádu s časovou konstantou  $\tau$  buzený bílým šumem procesu. Tím určuji, že úhlové rychlosti předmětu se nemohou měnit skokově, ale nanejvýš s danou časovou konstantou. Tento jednoduchý model používám z důvodu toho, že není k dispozici žádný model pohybu předmětu. Výstupní rovnice se skládají z dvou trojic rovnic popisující chování dvou typů senzorů. První trojice výstupních rovnic jsou rovnice měření akcelerometry. Tyto rovnice udávají průmět gravitačního zrychlení do jednotlivých akcelerometrů v závislosti na aktuální poloze (stav quaternionů). Druhá trojice výstupních rovnic popisuje chování gyroskopů. Celkově vypadá spojitý nelineární stavový model následujícím způsobem.

**Stavová rovnice:**

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{B}_p \\ \dot{B}_q \\ \dot{B}_r \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(-p \cdot q_1 - q \cdot q_2 - r \cdot q_3) \\ \frac{1}{2}(+p \cdot q_0 + r \cdot q_2 - q \cdot q_3) \\ \frac{1}{2}(+q \cdot q_0 - r \cdot q_1 + p \cdot q_3) \\ \frac{1}{2}(+r \cdot q_0 + q \cdot q_1 - p \cdot q_2) \\ 0 \\ 0 \\ 0 \\ -\frac{1}{\tau}p \\ -\frac{1}{\tau}q \\ -\frac{1}{\tau}r \end{bmatrix} \quad (4.9)$$

**Výstupní rovnice:**

$$\begin{bmatrix} A_x \\ A_y \\ A_z \\ p_m \\ q_m \\ r_m \end{bmatrix} = \begin{bmatrix} 2 \cdot g \cdot (q_1 \cdot q_3 - q_0 \cdot q_2) \\ 2 \cdot g \cdot (q_2 \cdot q_3 + q_0 \cdot q_1) \\ g \cdot (q_0^2 - q_1^2 - q_2^2 + q_3^2) \\ p + B_p \\ q + B_q \\ r + B_r \end{bmatrix} \quad (4.10)$$

Pro návrh algoritmu rozšířeného Kalmanova filtru je nutné nejprve spojitý model popsaný rovnicemi (4.9) a (4.10) diskretizovat a poté určit jakobiány stavové a výstupní rovnice v obecném pracovním bodě. Celý postup algoritmu Kalmanova filtru je v [18]. Diskretizaci jsem provedl Eulerovou diskretizační metodou. Tedy všechny derivace byly nahrazeny podle vztahu

$$\dot{x}(t) \approx \frac{x(t + T_s) - x(t)}{T_s}.$$

Vzorkovací periodu  $T_s$  jsem zvolil 50 Hz, což je těsně pod výpočetní rychlosť zvládnutelnou v procesoru LPC 2119.

#### 4.4.3.4 Algoritmus rozšířeného Kalmanova filtru

Algoritmus kalmanova filtru v případě lineárního modelu je možné realizovat jako algebraickou rekurzi nad kovariační maticí a střední hodnotou stavu [8, 18]. Abychom mohli operace algebraické rekurze použít i pro nelinaérní systémy, je potřeba nelinaérní systém linearizovat v aktuálním pracovním bodě a pro výpočty algebraické rekurze použít parametry tohoto linearizovaného modelu.

V každém kroku rozšířeného Kalmanova filtru je tedy nutné provést následující výpočet.

$$\begin{aligned}
 \mathbf{C}(t) &= \left. \frac{\partial \mathbf{g}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right|_{x=x(t|t-1), u=u(t)} \\
 \mathbf{L}(t) &= \mathbf{P}(t|t-1) \mathbf{C}(t)^T \left( \mathbf{C}(t) \mathbf{P}(t|t-1) \mathbf{C}^T + \mathbf{R} \right)^{-1} \\
 \hat{\mathbf{x}}(t|t) &= \hat{\mathbf{x}}(t|t-1) + \mathbf{L}(t) \left( \mathbf{y}(t) - \mathbf{g}(\hat{\mathbf{x}}(t|t-1), \mathbf{u}(t)) \right) \\
 \mathbf{P}(t|t) &= \mathbf{P}(t|t-1) - \mathbf{L}(t) \left( \mathbf{C}(t) \mathbf{P}(t|t-1) \mathbf{C}^T + \mathbf{R} \right) \mathbf{L}^T \\
 \hat{\mathbf{x}}(t+1|t) &= \mathbf{f}(\hat{\mathbf{x}}(t|t), \mathbf{u}(t)) \\
 \mathbf{A}(t) &= \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right|_{x=\hat{\mathbf{x}}(t|t), u=u(t)} \\
 \mathbf{P}(t|t+1) &= \mathbf{A}(t) \mathbf{P}(t|t) \mathbf{A}^T + \mathbf{Q},
 \end{aligned} \tag{4.11}$$

$\mathbf{P}(t)$	je kovariační maticí stavu	$\mathbf{g}(\mathbf{x}, \mathbf{u})$	je nelinaérní výstupní funkce
$\mathbf{Q}$	je kovariační maticí šumu procesů	$\mathbf{f}(\mathbf{x}, \mathbf{u})$	je nelinaérní stavová funkce
$\mathbf{R}$	je kovariační maticí šumu měření	$\hat{\mathbf{x}}(t)$	je odhad stavu
$\mathbf{L}(t)$	je Kalmanovo zesílení	$\mathbf{u}(t)$	je měřený vstup
$\mathbf{C}(t)$	je linearizovaná maticí výstupu	$\mathbf{y}(t)$	je měřený výstup
$\mathbf{A}(t)$	je linearizovaná stavová maticí		

Realizaci tohoto výpočtu v jazyce Matlab naleznete v příloze A této diplomové práce. Algoritmus jsem testoval nejprve na uměle vytvořených datech [21], po té i na reálných datech naměřených senzorem. Po úspěšných výsledcích algoritmu jsem se rozhodl algoritmus implementovat v jazyce C v procesoru LPC 2119, abych tak ověřil možnost použití algoritmu při zpracování v reálném čase.

#### 4.4.3.5 Realizace algoritmu v jednočipovém počítači

Pro realizaci algoritmu v jazyce C bylo nejprve nutné napsat algoritmy pro maticové operace. Jedná se o operace násobení matic a násobení matic s transpozicí. Násobení matic s transpozicí je výpočetně méně náročné než provádět nejprve transpozici a pak samotné násobení. Dalsí operací je operace výpočtu inverze matice, resp. výpočtu rovnice (4.11). Rovnice (4.11) se dá po substituci

$$\mathbf{A} = \mathbf{C}(t) \mathbf{P}(t|t-1) \mathbf{C}^T + \mathbf{R} \quad \text{a} \quad \mathbf{B} = \mathbf{P}(t|t-1) \mathbf{C}(t)^T$$

zapsat jako  $\mathbf{L} = \mathbf{B} \cdot \mathbf{A}^{-1}$  což je totéž jako  $\mathbf{L} \cdot \mathbf{A} = \mathbf{B}$ . Přitom matice  $\mathbf{A}$  je pozitivně definitní, protože vznikla přenásobením pozitivně definitní matice  $\mathbf{P}$  kvadrátem matice  $\mathbf{C}$  a přičtením pozitivně definitní matice  $\mathbf{R}$ . K výpočtu Kalmanova zesílení je tedy možné využít Choleskyho faktorizace [17] matice  $\mathbf{A}$ . Pro implementaci Choleskyho faktorizace jsem využil algoritmů v [2].

Důkladně zdokumentovaný zdrojový kód provádějící tento algoritmus v jazyce C můžete nalézt na webových stránkách projektu stabilizované základny v sekci *software* [20]. Program nejprve provede veškerá potřebná nastavení periférií, časovačů a CAN sběrnice. Poté s frekvencí 50 Hz vzorkuje hodnoty zrychlení a úhlových rychlostí senzoru ADIS 16350 a provádí výpočet algebraické rekurze. Na konci každého cyklu program odešle aktuální odhad parametrů pomocí CAN sběrnice do desktopového počítače, kde je možné je prostřednictvím uživatelského grafického rozhraní zobrazit či uložit pro další zpracování např. v Matlabu (pozn. GUI bude podrobněji popsáno v kapitole 5.3.).

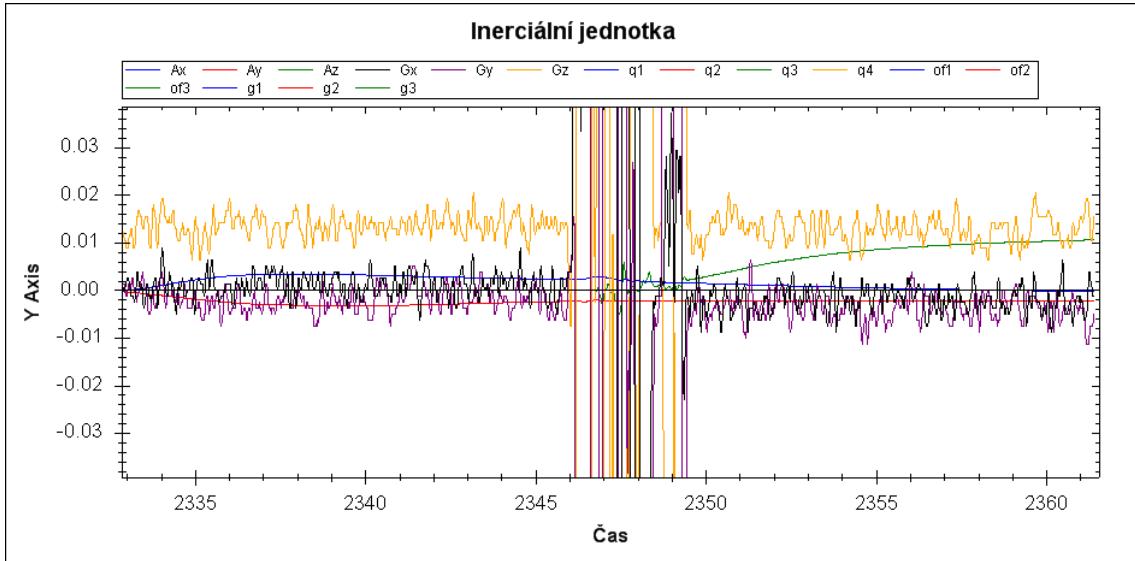
#### 4.4.3.6 Výsledky

Počáteční stav je vždy nastavený na  $[q_0, q_1, q_2, q_3]^T = [1, 0, 0, 0]^T$  což odpovídá nulovým Eulerovým úhlům (přímočarý let s nulovým azimutem). Pro přepočty quaternionů na Eulerovy úhly a zpět je výhodné použít Java appletu [1]. Data grafů v této podkapitole byla všechna vytvořena činností Kalmanova filtru v reálném čase v procesoru.

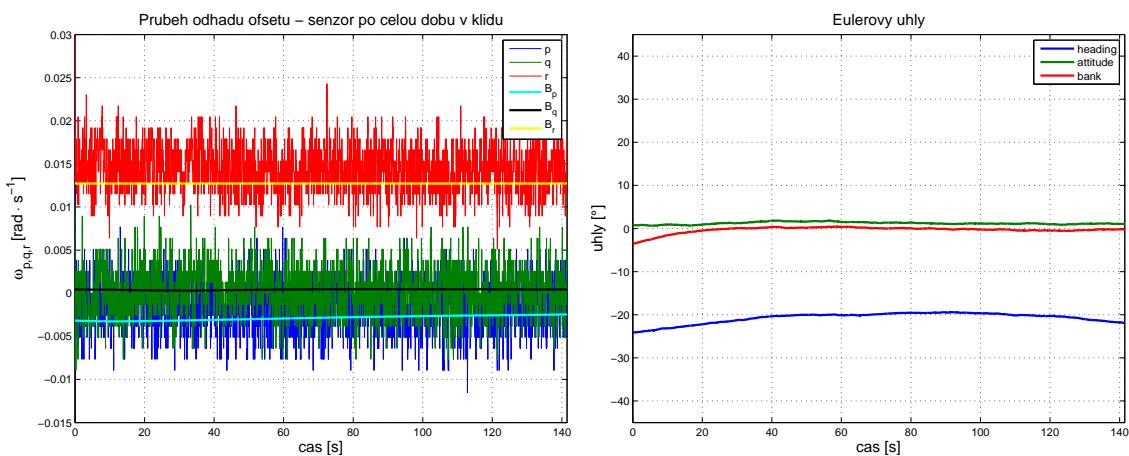
Na obr. 4.12 můžete vidět průběh odhadu ofsetů při začátku činnosti Kalmanova filtru (po resetu počítače). Senzor jsem nejprve nechal ležet v klidu 13 s ve výchozí poloze, potom jsem jej pootočil o asi  $90^\circ$  okolo osy  $p$  a opět jej nechal v klidu ležet na stole. Jelikož algoritmus nemá zádnou apriorní informaci o ofsetech a ni o poloze senzoru, je nejistota daná počáteční kovariační maticí  $P$  nastavena na velkou hodnotu a způsobuje velmi rychlý odhad ofsetů  $B_p$ ,  $B_q$ .

Odhad ofsetu v ose  $B_r$  však není možný dokud nedojde k pootočení senzoru. Pokud totiž dochází k rotaci okolo s gravitačním vektorem rovnoběžné osy, jsou akcelerometry na tuto rotaci necitlivé. Tzn. že akcelerometry jsou necitlivé i na rotaci způsobenou pouze ofsetem. V Kalmanove filtrace vše odpovídá situaci, kdy se snažíme navrhnout filtr na systém s nepozorovatelnou částí. Takto navržený filtr pak odhaduje pouze na základě dostupného modelu [18]. Můj použitý model chování ofsetu je ale náhodná procházka a bez jakéhokoli jiného zásahu do filtru proto bude odhad vypadat jako náhodná procházka. Tuto situaci jsem se proto pokusil vylepšit tím, že na konci datového kroku vynuluji příslušné prvky Kalmanova zesílení, které odpovídají korekci odhadu ofsetu  $B_r$ . Proto na obr. 4.12 vidíte, že odhad ofsetu  $B_r$  (zelená barva) začne probíhat až v čase 2347 s, tj. po pootočení senzoru o  $90^\circ$  tak, že souhlasně vektorem gravitačního zrychlení již není osa  $r$  ale osa  $q$ .

Pokud na začátku každého spuštění filtru provedeme nejprve natočení senzoru do dvou různých vzájemně kolmých os po delší dobu než je asi 10 – 15 s je odhad polohy Kalmanovým filtrem celkem rozumný a krátkodobě v jednotkách minut i relativně přesný a stabilní jak ukazuje obr. 4.13.

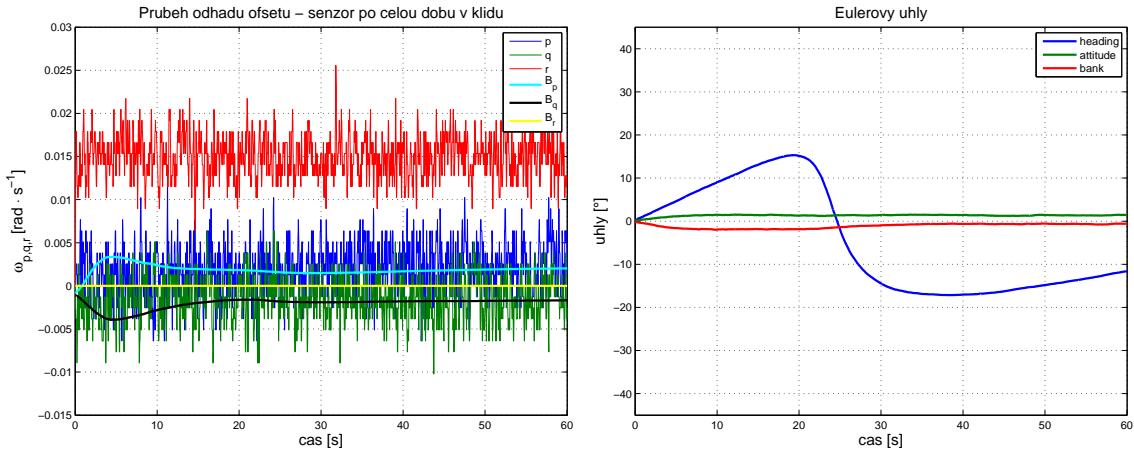


Obrázek 4.12. Úhlové rychlosti  $p, q, r$  (černá, fialová, žlutá) a odhad offsetů  $B_p, B_q, B_r$  (modrá, červená, zelená) po resetu programu procesoru. Odhad offsetu  $B_r$  (zelená barva) je možný až po natočení senzoru do polohy, kdy jsou akcelerometry citlivé na rotaci okolo osy  $r$ . Obrázek byl vytvořen v reálném čase za pomocí grafického rozhranní. Jednotkou závislé osy jsou  $\text{rad} \cdot \text{s}^{-1}$ .



Obrázek 4.13. Dvouminutový průběh odhadu polohy v situaci kdy senzor leží v klidu na stole. Průběh byl zaznamenán již po delší době činnosti filtru v různých polohách. Filtr si tak stačil relativně dobře odhadnout polohu i ofsety jednotlivých gyroskopů. Odhad úhlů attitude a bank (podléný a příčný sklon) je přesný.

Naopak pokud k takovému počátečnímu „vybuzení“ filtru nejdojde, pak není algoritmus schopen přesněji odhadnout azimut (zbylé dvě osy však lze odhadnout dobře) jak ukazuje obr. 4.14.



Obrázek 4.14. Odhad offsetů a polohy po resetu algoritmu. Senzor je umístěn po celou dobu v klidu ve výchozí poloze. Offsety  $B_p$  a  $B_q$  jsou odhadnutý správně stejně jako úhly podélného a příčného sklonu. Offset  $B_r$  není možné odhadnout proto je jeho odhad „zakázán“. Odhad polohy azimutu však „zakázat“ nejde a proto je velmi špatný.

Obrázek 4.15 ukazuje asi 70 s dlouhý průběh odhadu polohy senzoru (předmětu) od resetu algoritmu. Nejprve jsem senzor postupně natočil do všech tří os o úhel cca  $40 - 50^\circ$ . Následně jsem senzor překlopil o více než  $90^\circ$  v příčném sklonu, aby došlo k dobrému odhadu offsetu gyroskopu osy  $r$ . Na závěr jsem senzor umístil do výchozí polohy. Na obrázku můžete vidět porovnání odhadu polohy s využitím veškeré dostupné informace (gyroskopů a akcelerometrů) zpracované rozšířeným Kalmanovým filtrem (obr. 4.15 nahoře) v porovnání se zpracováním pouze úhlových rychlostí gyroskopů a jejich integrování do quaternionů (obr. 4.15 dole). Pro snadnější představu o poloze jsou průběhy quaternionů z levé půlky obrázku převedeny offline v Matlabu do eulerových úhlů.

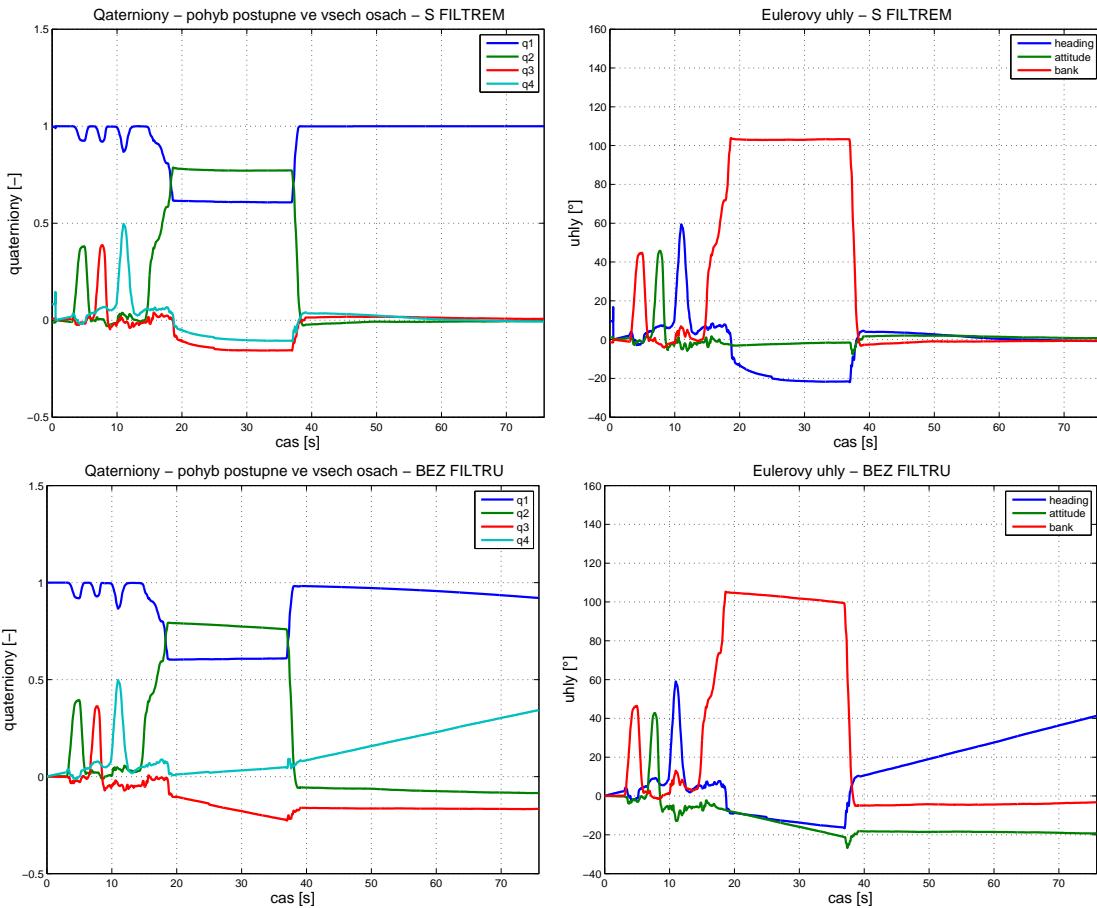
Výsledkem obrázku 4.15 je skutečnost, že použitím filtru dojde k mnohonásobně přesnější informaci o poloze senzoru (předmětu), než je možné dosáhnout pouze integrací úhlových rychlostí senzoru bez jakékoli kalibrace.

#### 4.4.3.7 Působení vnějšího zrychlení

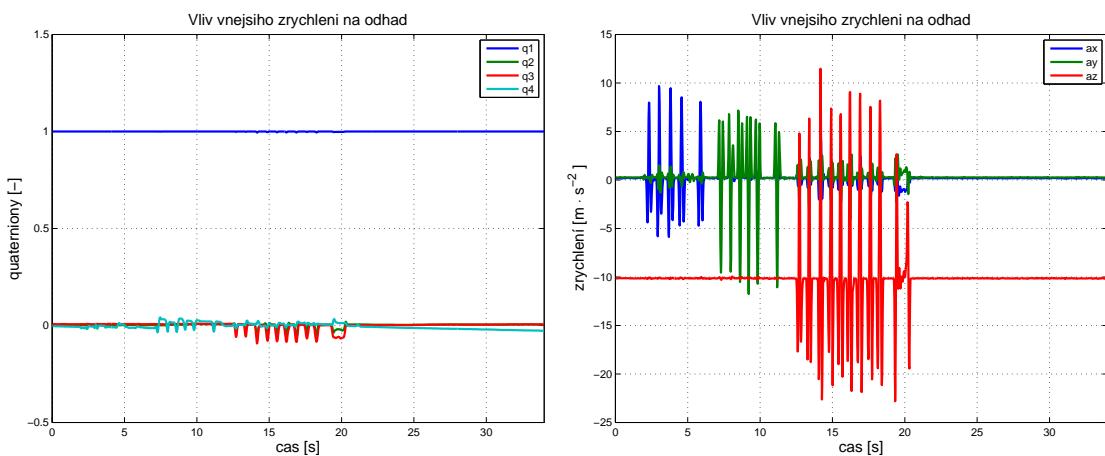
Zajímavým testem navrženého Kalmanova filtru je ověřit chování filtru v situaci, kdy na akcelerometry působí kromě gravitačního zrychlení i vnější zrychlení způsobené pohybem senzoru. Na obr. 4.16 můžete vidět výsledky takového pokusu, kdy jsem senzorem hýbal prudkými pohyby postupně ve všech třech osách. Odhad polohy je velmi dobrý a malé rotace, které obr. 4.16 vlevo ukazuje, jsou skutečné rotace, ke kterým během prudkých pohybů došlo. Hlavní důvod tak dobré odolnosti vůči vnějším zrychlením je poměrně malá důvěřivost algoritmu v měřená zrychlení po uplynutí počáteční doby, kdy byl Kalmanův filtr schopen rychle odhadnout polohu a offsety (v řádu 10 – 20 s). Pokud by k vnějším zrychlením z obr. 4.16 vpravo došlo během prvotní fáze algoritmu, kdy ještě neodezněl počáteční stav s vysokou kovariační maticí  $P$ , výsledky by nebyly tak příznivé.

#### 4.4. SLEPÉ SMĚROVÁNÍ

37



Obrázek 4.15. Nahoře vidíte odhady při použití Kalmanova filtru, dole výsledky pouhé integrace úhlových rychlostí. V levé části pak vidíte odhady polohy v quaternionech. V pravé části jsou quaterniony pro lepší představu převedeny do eulerových úhlů.



Obrázek 4.16. Vliv působení vnějšího zrychlění na odhad quaternionů.

Možností jak zabezpečit, aby i v počáteční fázi filtr nebyl ovlivněn vnějším zrychlením (případně také možností jak dovolit větší důvěřivost algoritmu v měřená zrychlení), je provést před aplikací nových dat akcelerometru v datovém kroku nejprve test, zda je celkové zrychlení měřené akcelerometry rovno v absolutní hodnotě gravitačnímu zrychlení. Pokud totiž ne, značí to že, jsou data znehodnocena působením vnějšího zrychlení. Existuje jen velmi málo nepravděpodobných situací kdy by se objekt pohyboval tak, že by celkové zrychlení měřené akcelerometry bylo v absolutní hodnotě rovno gravitačnímu. Takto je tedy možné provádět neustále predikční krok algoritmu dokud nepřijdou platná data. Je však zřejmé, že tento postup může být použit jen v případě, že měřený objekt je po většinu času v klidu a jen občas se pohybuje (působí na něj vnější zrychlení).

#### 4.4.3.8 Závěr a možná vylepšení

Této kapitole byl popsán navržený algoritmus realizující odhad polohy pomocí MEMS gyroskopů a akcelerometrů rozšířeným Kalmanovým filtrem v jednočipovém procesoru. Navržený algoritmus má svá fyzikální omezení. Prvním omezením je podmínka nulového vnějšího zrychlení, která může být ve spoustě situacích (např. odhad polohových úhlů letadla) velmi kritická a proto tento algoritmus není možné použít na odhad polohových úhlů letadla. Proto pokud to jde je třeba model pohybu objektu více specifikovat (např. u letadla dynamickými rovnicemi pohybu).

Druhým omezením je neschopnost algoritmu odhadnout offset senzoru jehož osa citlivosti je souhlasná s vektorem gravitačního zrychlení. V případě např. letadla v přímočarém letu se jedná o osu azimutu. V této situaci je velmi výhodné doplnit algoritmus o zpracování dalšího senzoru – magnetometru. Magnetometr je schopen měřit vektor magnetického pole Země, který je v předem definovaném místě na Zemi konstantní, podobně jako vektor gravitačního pole Země. Jelikož oba vektory mají jiný směr je možné spojením obou senzorů (akcelerometrů a magnetometrů) mnohem lépe odhadovat ofsety gyroskopů a tím i polohu objektů. Rozšíření modelu o možnost zpracování dat z magnetometru je velmi jednoduché, protože jde v principu o stejný problém jako s akcelerometry. Jen nejde o vektor gravitace, ale o vektor jiný. Magnetometr má pak oproti akcelerometru tu výhodu, že nereaguje na vnější vlivy – vnější zrychlení. Jeho nevýhodou je naopak zase velká citlivost na rušivé magnetické pole (např. od elektromotorů) nebo citlivost na přítomnost kovů v blízkosti senzoru. Závěrem je tedy nutné říci, že díky četným zjednodušením a skutečnosti, že odhad vůbec nepočítá s modelem letounu, měřením vzdušné rychlosti a dokonce ani se zásahy autopilota je nutné výsledný odhad brát částečně s rezervou. Výsledky odhadu jsou však vždy mnohem lepší, než kterých bychom dosáhli pouze integrací měřených úhlových rychlostí nebo pouze spoléháním se na měření vektoru gravitačního pole akcelerometry.

# Kapitola 5

## Implementace řídicího systému

V této bude popsána struktura a detaily navržené elektroniky. Úspěšně se nám podařilo navrhnut několik desek plošných spojů, které jako celek realizují činnost algoritmu rychlostního regulátoru popsaného v kapitole 4. Všechny desky plošných spojů jsou navrženy s ohledem na jejich umístění do stabilizované základny. V druhé polovině této kapitoly je pak popsána softwarová část – programy řídicích a měřicích desek realizující algoritmus rychlostního regulátoru. Po dodání hardware zpracovávající obrazový signál kamery budeme schopni propojit oba podsystémy dohromady a předvést činnost hlavního požadovaného úkolu – režimu *Vizuální zpětné vazby*. O reálných výsledcích implementace algoritmů stabilizace rychlosti se můžete přesvědčit v následující kapitole 6 s názvem *Experimentální ověření*.

### 5.1 Struktura elektronické části

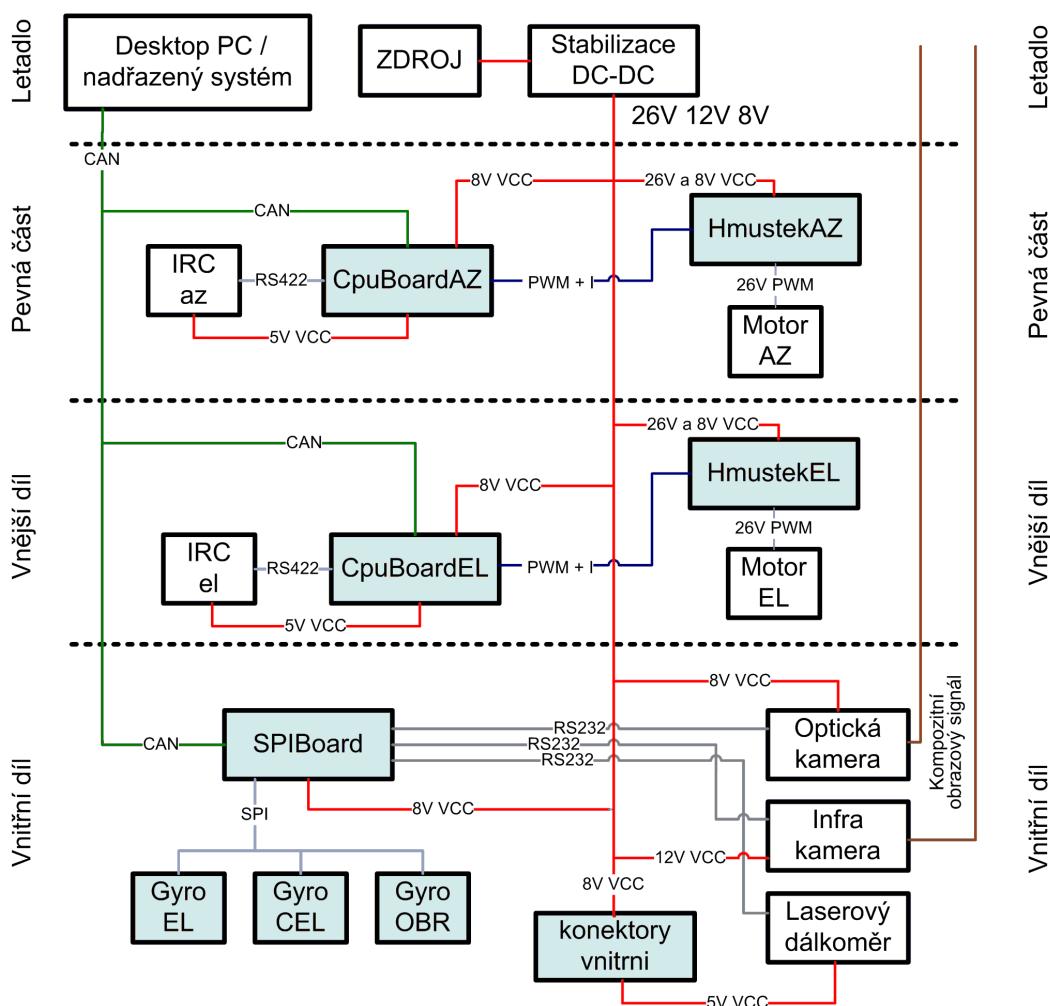
Jelikož kluzné kroužky zmíněné v kapitole 2 umožňují průchod jen 24 vodičů se špičkovým zatížením každého vodiče max 2 A, bylo nutné provést distribuci částí celého systému na různá snadno přístupná místa ve stabilizované základně. Rozdelení také činí větší odolnost celého systému vůči rušení, protože veškerá kabeláž je vedena na co nejkratší vzdálenost.

Rozdelení elektroniky základny nejlépe zobrazuje obr. 5.1. Stabilizovaná základna obsahuje celkem čtyři vzájemně pohybující se místa kde je umístěna elektronika. Tyto místa jsou od sebe na obr. 5.1 oddělena čárkovanou čarou. Dolní dvě čárkované čáry symbolizují kluzné kroužky. Postup ve schématu zdola nahoru odpovídá postupu od vnitřních rámů základny k pevné části spojené s letadlem. Bloky označené světle modrou barvou jsou desky plošných spojů, které jsme navrhli ve spolupráci s kolegou Jaroslavem Žohou. Názvy bloků odpovídají názvům elektroniky (a také zdrojovým souborům v programu Eagle) jak ji můžete nalézt na webových stránkách projektu Stabilizované základny [20].

Červené linky na obr. 5.1 označují rozvod napájení v základně. Jelikož uzly elektroniky vyžadují různá napájecí napětí a různé proudové odběry bylo důležité rozmyslet strukturu napájecí sítě. Struktura se odvíjí hlavně od napájení pohonů (26 V) a napájení kamery (12 V, 8 V a 5 V). Přes jednotlivé kroužky tedy vedeme tři úrovně napájení (26 V, 12 V

a 8 V), napájecí napětí pro laserový dálkoměr je stabilizováno z 8 V na 5 V až ve vnitřním závěsu základny.

Zelené linky na obr. 5.1 značí rozvod CAN sběrnice. Ta propojuje jednotlivé měřicí a řídící uzly. Používaná rychlosť CAN sběrnice je 1 mbit/s. Hnědé linky značí kompozitní obrazový signál kamér, který je bez změny (a dokonce i bez jakéhokoli propojení se společnou zemí elektroniky) veden do letadla, kde bude umístěna jednotka zpracování obrazu.



Obrázek 5.1. Blokové schéma zapojení elektroniky. Světle modré bloky jsou navržené desky plošných spojů.

Blok *Desktop PC / nadřazený systém* v obr. 5.1 značí, že ostatní elektronika zpracování obrazového signálu, elektronika inerciální jednotky a také elektronika polohového regulátoru v režimu slepého směrování bude umístěna v letadle. V případě testování základny v laboratorních podmínkách tento blok symbolizuje desktopový počítač, na kterém běží vizualizační a ovládací software (viz. kap. 5.3).

Jako výpočetní jednotku všech navržených řídících a senzorových jednotek jsem vybral

procesor LPC 2119 od fy. NXP<sup>1</sup> (Phillips). Důvodem výběru tohoto procesoru je hlavně velká zkušenosť s tímto procesorem na katedře řídicí techniky a dále také jeho výhodná cenová dostupnost a dostatečná výpočetní kapacita. Procesor LPC 2119 je 32-bitový procesor s jádrem ARM7 s taktovací frekvencí až 60 MHz. Obsahuje potřebné periferie pro komunikaci po SPI a CAN sběrnici, AD převodník a blok PWM (pulzně šířkové modulace) pro řízení motorů. Všechny programy jsou napsány v jazyce C, k překladu programů využívám operační systém linuxového typu a volně dostupný překladač GCC 4.1.2 za podpory sestavovacího prostředí OMK (OCERA Make System<sup>2</sup>). Procesory programují buď do RAM paměti, nebo do flash paměti v případě finálních programů, pomocí sériové linky a interního bootloaderu procesoru.

### 5.1.1 Elektronika vnitřního závěsu základny

Ve vnitřním závěsu základny je umístěna navržená deska s názvem *SPIBoard*. Tato deska plošných spojů má za úkol měřit aktuální pohyb kamery pomocí dvou jednoosých MEMS gyroskopů ADIS 16255 nutných pro realizaci stabilizace optické osy. Je však osazen také třetí senzor, jehož použití se předpokládá pro zlepšení algoritmu sledovaní při zpracování obrazu kamer. Pro snadné umístění gyroskopů ve vnitřním závěsu základny jsou senzory osazeny každý na své vlastní desce jak ukazuje obr. 2.8. Aktuální změřené úhlové rychlosti předává deska *SPIBoard* na CAN sběrnici. Detailnější popis gyroskopu ADIS 16255 již byl uveden v kapitole 2.



Obrázek 5.2. Elektronika vnitřního závěsu základny. Na obrázku můžete vidět dva ploché vícežilové barevné kably, které připojují k desce SPIboard gyroskopy.

Dalším úkolem desky *SPIBoard* je obsluha obou kamer a laserového dálkoměru po třech sériových linkách RS232 (viz. obr. 5.1). Deska předává obousměrně data ze všech sériových

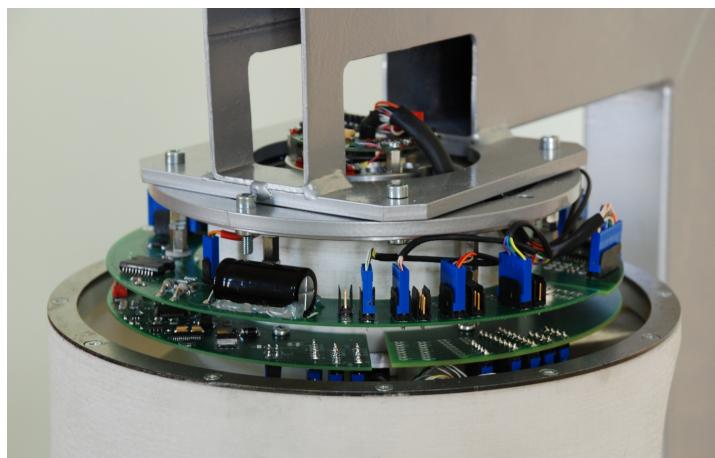
<sup>1</sup><http://www.nxp.com/>

<sup>2</sup><http://rtime.felk.cvut.cz/OMK>

linek podle předem stanovených identifikátorů CAN zprávy. Na obr. 5.2 můžete vidět fotku elektroniky vnitřního závěsu základny. Detailnější popis programu a elektroniky desky *SPIBoard* naleznete v následujících podkapitolách 5.2 a 5.3.

### 5.1.2 Elektronika řídicích jednotek

Desky plošných spojů zodpovědné za řízení pohonů azimutálního a elevačního kloubu nesou název *CPUBoardEL* a *CPUBoardAZ*. Jak můžete vidět na schématu na obr. 5.1 je řídicí deska azimutálního kloubu umístěna v pevné části základny, zatímco řídicí deska elevačního kloubu je umístěna ve vnějším závěsu konstrukce. Obě desky se proto vůči sobě pohybují rotací azimutálním kloubem (viz. obr. 5.3). Jsou však umístěny na takovém místě, že jsou snadno přístupné.



Obrázek 5.3. Detail řídicí elektroniky základny a H-můstku azimutálního motoru. Horní desky plošných spojů jsou umístěny v pevné části základny, dolní ve vnějším závěsu. Úplně nahoře je vidět inkrementální senzor polohy azimutálního kloubu.

Programy desek realizují v kapitole 3 popsáný algoritmus stabilizace optické osy. Obě desky pracují s aktuálními úhlovými rychlostmi předávanými z měřícího uzlu *SPIBoard* po CAN sběrnici a řízením napětí na motorech pomocí PWM (pulzně šírková modulace) se snaží udržet měřené úhlové rychlosti nulové. Jelikož spínaný výkonový signál napětí motoru je nutné vést co nejkratší kabeláží, je elektronika pro výkonové přizpůsobení PWM signálu (tzv. H-můstek) umístěna co nejbliže ke svorkám motorů. Desky H-můstků mají přímočarý název *HmustekEL* a *HmustekAZ*. Desky řídicích jednotek a H-můstků jsou tedy propojeny PWM signálem, a dále také dvojicí vodičů s napětím odpovídajícím aktuálnímu proudu motorem pro realizaci proudové zpětné vazby či ochrany motoru (detailnější popis v následující podkapitole 5.2). K převodu napětí odpovídajícího aktuálnímu proudu motorem je využit interní 10-bitový AD převodník, který je pro tento účel dostatečně rychlý a přesný.

Posledním úkolem, který řídicí jednotky pohonů zastávají, je úkol zpracování signálu inkrementálních čidel a odesílání aktuálního stavu polohy, proudu motorem a aktuálního

akčního zásahu na CAN sběrnici.

## 5.2 Navržený hardware

Nyní uvedu krátký popis jednotlivých navržených desek plošných spojů popsaných v blokovém schématu na obr. 5.1. Všechny desky plošných spojů byly navrženy v programu Eagle pod školní licencí. V návrhu a výrobě celého hardware jsem se velmi významně podílel hlavně po stránce koncepční, po stránce návrhu schématu a výběru součástek. Samotný návrh desek plošných spojů v programu Eagle ve většině případů provedl kolega Jaroslav Žoha [19] (kromě desky HmustekAZ, kterou jsem navrhoval sám). Následné osazení vyrobených desek součástkami opět probíhalo společnými silami, stejně tak jako umístění veškeré navržené elektroniky do základny a propojení kabelází. Mezi hlavní navržené desky plošných spojů patří řídicí jednotky elevačního a azimutálního pohonu (názvy desek jsou *CPUboardEL* a *CPUboardAZ*), desky pro výkonové přizpůsobení PWM signálu procesoru pro motor – tzv. H-můstky (*HmustekEL* a *HmustekAZ*) a deska obsluhující všechny gyroskopy a kamery (*SPIboard*).

### 5.2.1 Řídicí jednotky pohonů – CPUBoardEL a CPUBoardAZ

Při návrhu schématu desky plošných spojů jsem se částečně inspiroval v desce s názvem *SpejblARM* pracující jako řídicí elektronika chodícího robota<sup>3</sup>. Schéma navržené desky *CPUBoardAZ* v programu Eagle naleznete jako druhé schéma v příloze C této diplomové práce. Obě desky plní stejný úkol – řízení pohonu kloubu, měření proudu motorem, měření polohy kloubu a komunikují po CAN sběrnici. Hlavní rozdíl v deskách je různé umístění jednotlivých konektorů tak, aby všechna kabeláž vedla co nejkratší cestu k cíli. Deska *CPUBoardAZ* navíc obsahuje místo pro zatím neosazený gyroskop ADIS 16255, který by mohl sloužit pro realizaci elevace-azimut stabilizace (viz. kap. 3).

Základem obou řídicích jednotek pohonů je již zmíněný procesor LPC 2119. Z důvodu snížení spotřeby a výpočetní nenáročnosti programu řídicí jednotky procesor je taktován na frekvenci 10 MHz. K taktování procesoru však není použit krystal, jak by se dalo očekávat, ale krystalový oscilátor. Tento oscilátor je pak také použit jako zdroj hodinového signálu pro obvod CPLD. Popis programů pro procesor a CPLD naleznete v kap. 5.3.

Pro vyšší odolnost vůči rušení obsahuje inkrementální senzor polohy výstupy s RS422 sběrnicí. Jedná se v podstatě o diferenciální linku. Pro připojení IRC čidla je tedy nejprve nutné zakončit vedení RS422 sběrnice a provést převod úrovní na 3,3 V logiku. K tomuto účelu jsme s výhodou použili převodník MAX3096 od firmy MAXIM.

Procesor LPC 2119 neobsahuje obvody zpracování signálů IRC. Neobsahuje dokonce ani žádné čítače externího signálu. Jedinou možností jak čítat externí signál procesorem by tedy bylo použití externích přerušení. Jelikož však použitá čidla dosahují přesnosti až

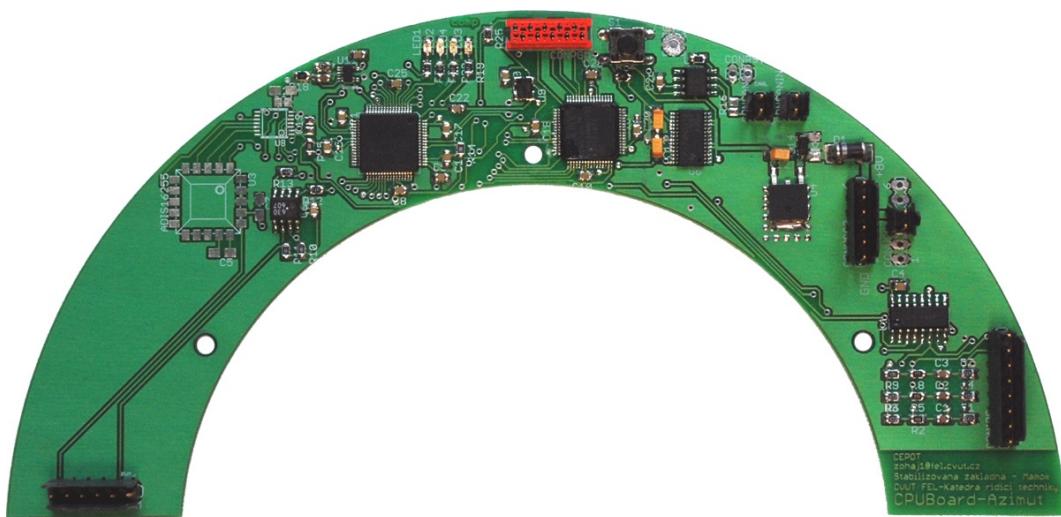
<sup>3</sup><http://dce.felk.cvut.cz/spejbl/>

50000 pulzů na jednu otáčku, došlo by k zbytečné velkému vytížení procesoru. Jediná možnost jak měřit polohu IRC čidla je tedy využít vycítání signálu externě jiným obvodem. Zvolil jsem variantu použití programovatelného obvodu CPLD XC9572XL od firmy Xilinx s 3,3 V napájecí logikou v pouzdru VQ64. Obvod má dostatečnou kapacitu i rychlosť, aby mohl splňovat funkci dekóderu IRC signálu a obsahoval v sobě 16-bitový čítač polohy potřebný pro uchování informace o poloze od 0 do 360°.

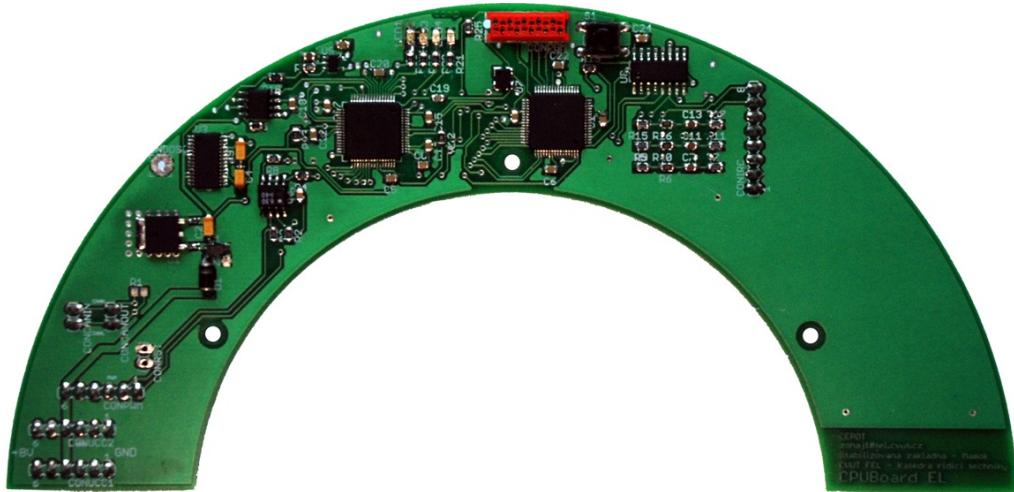
K řízení motoru využíváme vnitřní PWM modulátor procesoru. Tento PWM signál je potom veden do desky H-můstku kde je „výkonově posílen“ a přiveden na svorky motoru. Číslicová část procesoru je od výkonové části H-můstku oddělena pomocí dvojitého optočlenu HCPL0630, který je dostatečně rychlý na to, aby nezpožďoval hrany PWM signálu.

Jelikož napětí úměrné proudu motoru je v 5 V úrovni a interní AD převodník umí převádět napětí jen do úrovně 3,3 V bylo nutné vybavit vstup AD převodníku děličem a operačním zesilovačem s asymetrickým napájením v zapojení sledovače. Dělič napětí je rovnou vybaven i filtračním kapacitorem pro realizaci dolnofrekvenční propusti určené k odfiltrování zvlnění vstupního napětí zapříčiněného jeho původem z PWM signálu.

Pro připojení gyroskopu ADIS 16255 s 5 V logikou k procesoru s 3,3 V logikou vnějších obvodů a +5 V tolerancí bylo nutné převést výstupní 3,3 V signál z procesoru na 5 V úroveň gyroskopu pomocí převodníku MAX3379 od fy. MAXIM. V opačném směru není převod úrovní napětí nutný díky +5 V toleranci vstupních obvodů procesoru.



Obrázek 5.4. Řídicí deska azimutálního kloubu.



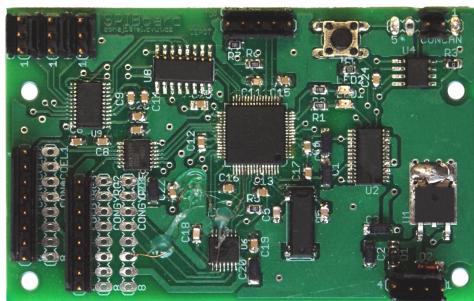
Obrázek 5.5. Řídicí deska elevačního kloubu.

### 5.2.2 Měřicí jednotka – SPIBoard

Schéma navržené desky *SPIBoard* v programu Eagle naleznete jako třetí schéma v příloze C této diplomové práce. Tato deska byla navržena pro dva úkoly. Prvním úkolem desky je obsluha gyroskopů a kamer ve vnitřním závěsu základny. Druhý podřadný úkol desky je využití tohoto hardware pro realizaci algoritmu inerciální jednotky popsaného v kapitole 4.4.3. Z tohoto důvodu je deska vybavena 5 vstupy SPI zařízení, protože se původně na místo kombinovaného senzoru ADIS 16355 počítalo s použitím tří jednoosých gyroskopů a dvou dvouosých akcelerometrů.

SPI výstupy jsou vybaveny stejnými převodníky logického signálu z úrovně 3,3 V na 5 V jako je tomu u předešlé desky *CPUBoardAZ*.

Jedním z úkolů desky je obsluha třech zařízení po sériové lince. Procesor LPC 2119, ale bohužel obsahuje jen dvě sériová rozhraní. Než vybírat nový procesor jeví se jako výhodnější přepínat jeden ze sériových portů procesoru na dvě zařízení – dvě kamery. Operátor totiž s největší pravděpodobností nikdy nebude nastavovat parametry obou kamer současně. K realizaci přepínače sériových linek jsme použili multiplexer HCF4052 pracující s 3,3 V logikou. Jako budiče tří sériových linek jsme použili obvod MAX 3387 od fy. MAXIM. Fotku navržené desky můžete vidět na následujícím obr. 5.6.



Obrázek 5.6. Deska plošných spojů obsluhující gyroskopy a kamery umístěná ve vnitřním závěsu stabilizované základny.

### 5.2.3 Výkonové spínače motorů – HmustekEL a HmustekAZ

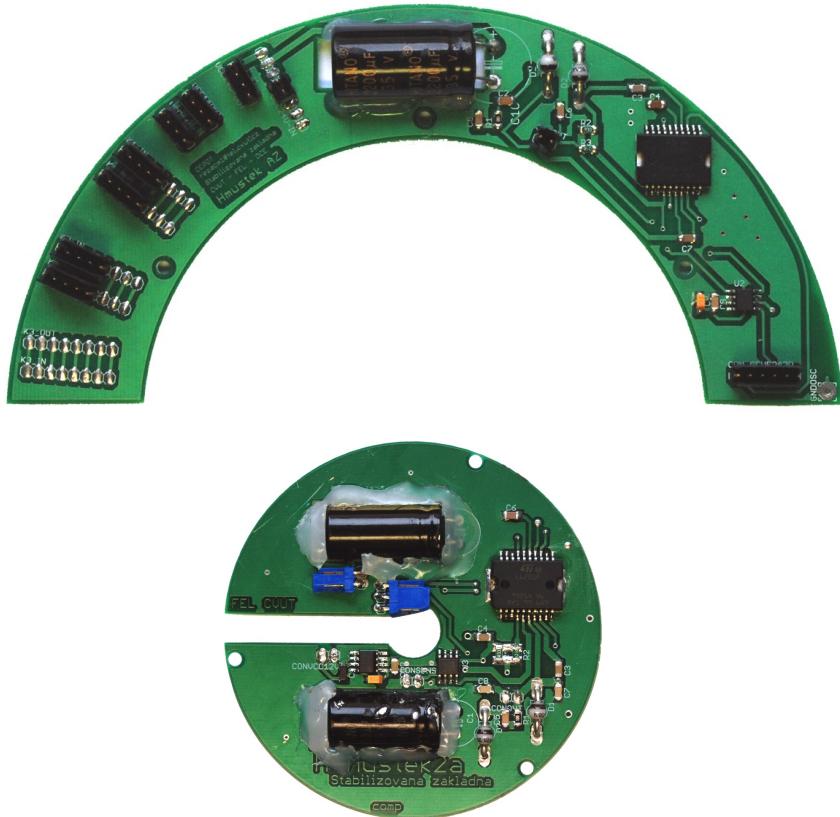
Schéma navržené desky *HmustekEL* v programu Eagle naleznete jako první schéma v příloze C této diplomové práce. Obě desky *HmustekEL* a *HmustekAZ* se od sebe po stránce schématu liší hlavně umístěním propojovacích konektorů pro kabeláž přicházející z letadla a vedoucí do azimutálního kroužku. Jedná se hlavně konektory napájení a konektory obrazového signálu kamer. Tyto konektory na desce elevačního H-můstku (*HmustekEL*) chybí.

Jako výkonový spínač PWM signálu a základ obou desek jsem vybral výkonový integrovaný obvod L6201PS v pouzdru PowerSO<sup>4</sup>. Důvodem výběru byla hlavně jeho snadná dostupnost a také skutečnost, že je typu SMD<sup>5</sup>, což jej umožnilo umístit do stísněného prostoru v místě nad elevačním motorem. H-můstek je schopný spínat napájecí napětí až do úrovně 48 V a proudy až 4 A. V sepnutém stavu mají spínací tranzistory můstku odpor  $R_{DS(ON)} = 0,3\Omega$ . Frekvence vstupního PWM signálu může být až 100 kHz.

Pro možnost proudové ochrany pohonu a případně i proudové zpětné vazby jsou obě navržené desky vybaveny měřením proudu pomocí měření úbytku napětí na rezistoru sériově zapojeného se svorkami motoru. Polarita úbytku napětí na tomto rezistoru je závislá na směru průchodu proudu motorem. Pro zpracování tohoto napětí jsme použili rozdílového operačního zesilovače AD8210 s asymetrickým napájením (0 a 5 V). Při správném zapojení tohoto operačního zesilovače přidává zesilovač k výstupnímu napětí takový offset, že nulovému proudu motorem odpovídá napětí v půlce rozsahu napájecího napětí. Tento offset je velmi výhodný a umožňuje zpracování výstupního napětí operačního zesilovače interním AD převodníkem procesoru.

<sup>4</sup>pěkný popis funkce tohoto obvodu naleznete na webové stránce <http://hw.cz/Teorie-a-praxe/Dokumentace/ART406-Ovladac-stejnosmerneho-motoru-využívající-PWM.html>

<sup>5</sup>Surface Mount Device = součástka s plošnou montáží



Obrázek 5.7. Desky H-můstků elevačního motoru (dole) a azimutálního motoru (nahoře).

## 5.3 Software

V této podkapitole uvedu popis programů, které jsem navrhoval. Jedná se hlavně o program řídicí jednotky pohonů v jazyce C realizující rychlostní smyčku algoritmu stabilizace optické osy. Dále se jedná o program dekódování IRC signálu CPLD obvodem napsaný v jazyce VHDL a program grafického uživatelského rozhraní (dále jen GUI). Tento program slouží k pohodlnému ovládání stabilizované základny, zobrazování aktuálních průběhů regulace a k zobrazování aktuálních odhadů z inerciální jednotky popsane v kap. 4.4.3. Program je napsán v jazyce C#.

### 5.3.1 Program řídicí jednotky motorů – CPUBoardAZ a CPUBoardEL

Program pracuje v roli regulátoru rychlosti elevačního resp. azimutálního motoru (viz. bloky Regulator1 a Regulator2 na obr. 3.4). Detailní dokumentaci zdrojového kódu najeznete na přiloženém CD, případně také na webové stránce projektu [20] v sekci *Software*.

Dokumentace byla vytvořena za pomocí prostředí DoxyGen<sup>6</sup>.

Program je taktován frekvencí 10 MHz. Nejprve je provedeno nastavení veškerých periférií procesoru a CAN sběrnice. Poté program naslouchá zprávám přicházejícím po CAN sběrnici a reaguje na zprávy s identifikátorem (id=303) odpovídajícím nově naměřeným datům senzorové desky (SPIboard). Pokud jsou přijata nová data, provede program výpočet jednoho kroku PI regulátoru rychlosti a následně odešle aktuální hodnoty proudu motoru, polohy inkrementálního čidla a aktuálního akčního zásahu (id=304 pro elevaci a 305 pro azimut).

Program řídicí jednotky azimutálního motoru se přitom liší tím, že pro svou činnost potřebuje nejprve znát aktuální hodnotu natočení elevačního kloubu, proto krok regulátoru azimutálního motoru není taktován příchodem nových dat, ale až příchodem stavové informace regulátoru elevačního kloubu (id=304).

Podle blokového schématu z obr. 3.4 by měly desky CPUBoard realizovat i proudovou zpětnou vazbu. Ta však zatím v programu regulátoru není ještě realizovaná. V pokračování projektu Stabilizované základny se ale počítá se její realizací. Program potom bude pomocí interního časovače s přerušením vzorkovat aktuální hodnotu proudu a nastavovat regulátorem proudu požadované napětí na motor. Výhodou tohoto uspořádání bude skutečnost, že v případě výpadku desky senzorů, která momentálně řídí takt regulátoru rychlosti, nebude na motor nastaven poslední akční zásah regulátoru rychlosti, ale regulátor proudu bude ovládat napětí motoru sám a bude realizovat proudovou a výkonovou ochranu motoru.

Dalším úkolem programu je čekání pomocí externího přerušení na příchod referenčního impulzu inkrementálního čidla. Pokud se tak stane, provede program přečtení chyby z CPLD obvodu a odešle na CAN sběrnici zprávu obsahující hodnotu chyby.

Aby bylo možné sledovat průběhy stavu základny v grafickém rozhraní (viz. kap. 5.3.4) i při vypnuté činnosti regulátoru rychlosti, je program vybaven možností odpojit akční zásah na motor, ale přitom zůstane neustále v činnosti. Toho je dosaženo dalšími zprávami s identifikátory (id=100 a 101), které jsou vysílány na sběrnici na požadavek uživatele prostřednictvím grafického rozhraní.

### 5.3.2 Program měřicí jednotky – SPIBoard

Deska SPIBoard slouží jako měřicí jednotka obsluhující gyroskopy a kamery umístěné ve vnitřním závěsu konstrukce. Detailní dokumentaci zdvojového kódu opět najeznete na přiloženém CD, případně také na webové stránce projektu [20] v sekci *Software*. Dokumentace byla taktéž vytvořena za pomocí prostředí DoxyGen. Po počátečním nastavení všech periférií program pomocí interního časovače s přerušením vzorkuje s frekvencí 250 Hz aktuální úhlové rychlosti senzorů elevace, příčné elevace a předává naměřená data na CAN sběrnici odkud jsou data dále zpracována programy řídicích jednotek obou os.

Dalším úkolem programu je role převodníku RS232 na CAN sběrnici pro 3 sériové linky třech kamer (2 kamer a laserového dálkoměru). Tato část programu je však neustále ve vývoji. Ve finále však bude princip činnosti takový, že operátor pomocí grafického rozhraní zvolí se kterou kamerou chce pracovat. Podle toho bude také nastaven

---

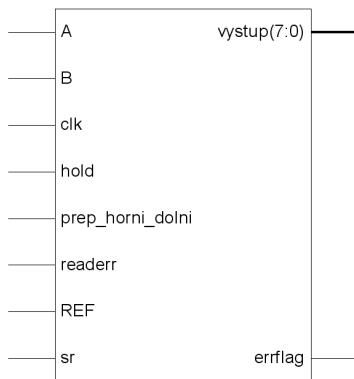
<sup>6</sup><http://www.doxygen.org/>

multiplexer sériové linky (viz. popis desky SPIboard v kap. 5.2) do příslušné polohy. Program bude všechny příchozí zprávy po CAN sběrnici předávat kameře a dálkoměru a naopak všechna příchozí data od kamery a dálkoměru bude přeposílat na CAN sběrnici s příslušnými identifikátory.

### 5.3.3 Zpracování signálu IRC čidel

K tvorbě programu pro obvod CPLD a samotného programování čipu jsem využil volně dostupného prostředí Xilinx ISE<sup>7</sup> s licencí TBL (Time-Based license). Zdrojové kódy programu v jazyce VHDL naleznete na přiloženém CD, případně také na webové stránce projektu stabilizované základny [20].

Na obr. 5.8 můžete vidět schéma hlavní entity (funkčního bloku) programu. Hlavní entita obsahuje vstupy IRC signálu (A,B,REF), vstup hodin (CLK), vstupy výběru horního a dolního byte (U/D), vstup zachycení aktuálního stavu čítače (HOLD), vstup výběru čtení chyby (READERR), 8 bitový výstup polohy (vystup) a výstup signalizace chyby (ERRFLAG).



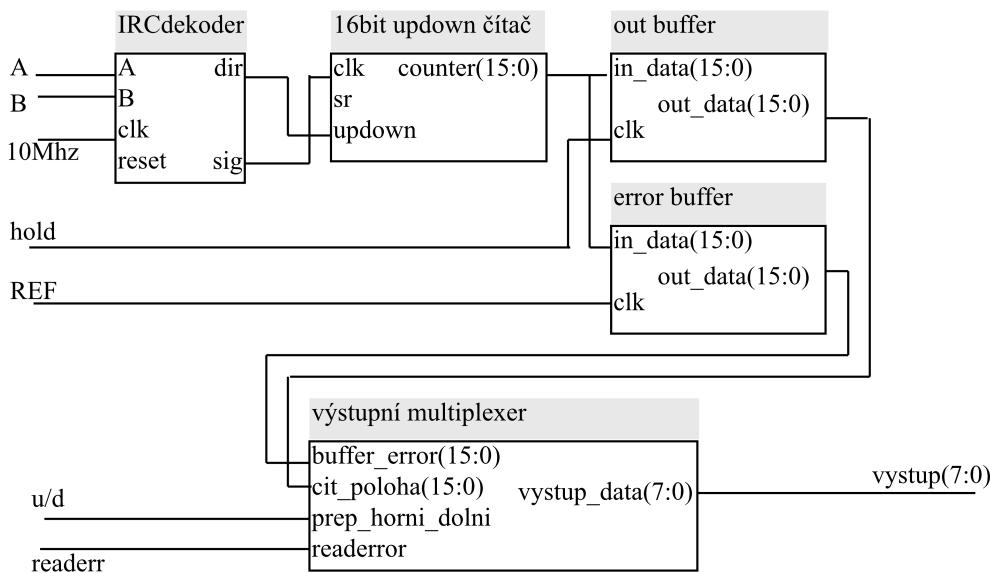
Obrázek 5.8. Schéma hlavní entity

Na obr. 5.9 je blokové schéma vnitřní struktury. Skládá se z těchto bloků:

- **IRCdekodeř** což je samotný dekoder IRC signálů. Jedná se o obyčejný dekóder<sup>8</sup>. Výstupem bloku je směr otáčení (**dir**) a jednotlivé impulzy pro čítač (**sig**). Dekóder je řešen jako synchronní stavový automat řízený signálem hodin **CLK**.
- **16-bit updown čítač** – reverzibilní 16-bit čítač, realizovaný jako čtyři 4-bitové kaskádně zapojené reverzibilní čítače
- **out buffer** je výstupní buffer (16-bit) – jedná se o D klopný obvod řízený náběžnou hranou signálu **HOLD**. Je zde obsažen z důvodu možnosti čtení horního a dolního byte aktuální polohy při současném nezávislém běhu čítače.

<sup>7</sup>[http://www.xilinx.com/ise/logic\\_design\\_prod/webpack.htm](http://www.xilinx.com/ise/logic_design_prod/webpack.htm)

<sup>8</sup>Tzn. ne kvadraturní, který by počítal i hrany signálů A,B. V případě realizace kvadraturního dekóderu bych dosáhl dvojnásobné přesnosti určení polohy, na jejíž uložení by bylo potřeba 17 bitů v registru polohy – z toho plynou další komplikace s přenosem údajů do procesoru.



Obrázek 5.9. Blokové schéma vnitřní struktury dekóderu

- **error buffer** je 16-bit buffer řízený sestupnou hranou signálu **REF** – referenčního pulzu IRC čidla. Při příchodu referenčního pulzu se uloží do tohoto bufferu aktuální stav čítače polohy. Je-li tento stav nulový, nedošlo v době před příchodem referenčního pulzu k žádné chybě dekódování.
- **výstupní multiplexer** na základě stavu vstupů **u/d** a **readerr** vybírá příslušných 8 bitů dat, které nastavuje na 8 bitový výstup **vystup**.

### 5.3.3.1 Sled kroků při čtení dat IRC čidla z obvodu CPLD

Následují postup ukazuje jakým způsobem by měl procesor navázat komunikaci a číst data z CPLD obvodu.

Na začátku činnosti je nejprve nutné vynulovat CPLD signálem reset. Dále nadřazený procesor musí počkat na příchod prvního referenčního impulzu (**REF**) pomocí přerušení nastaveného na náběžnou hranu signálu **REF**. Po té je již CPLD obvod plně v činnosti. Při každém příchodu referenčního pulzu je možné přečíst stav pinu **ERRFLAG**, který případě kladné úrovně hlásí, že odchylka při příchodu referenčního pulzu **REF** byla větší než jeden impulz inkrementálního čidla.

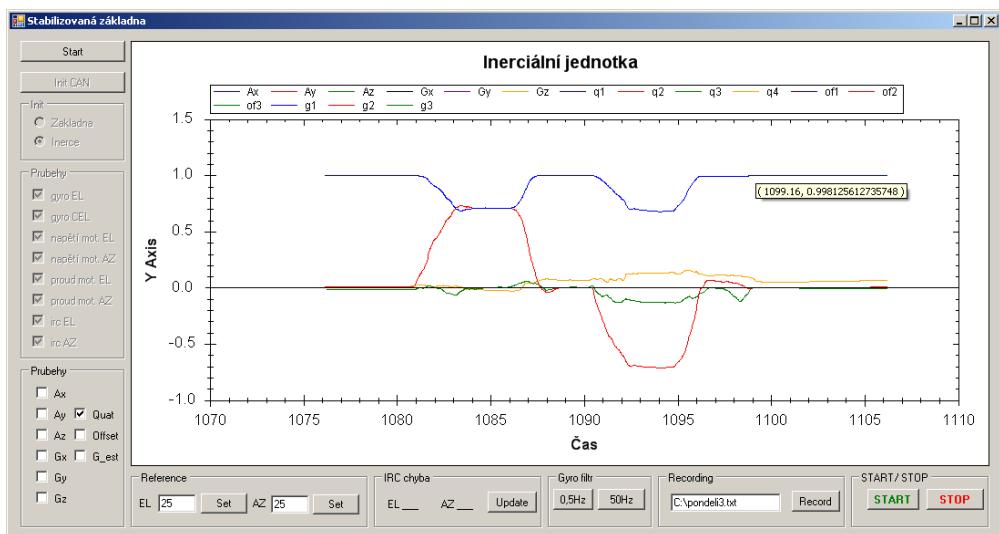
Průběh čtení aktuální polohy probíhá tak, že nejprve musí procesor nastavit kladnou úroveň na vstupu **HOLD** čímž přinutí CPLD uchovat aktuální hodnotu polohy ve výstupním bufferu („**out buffer**“ na obr. 5.9)). Po té procesor pomocí vstupů **updown** a **readerr** ovládá zda chce paralelně číst horních či dolních bitů výstupního bufferu či bufferu chyby (viz. blok „**výstupní multiplexer**“ na obr. 5.9)).

### 5.3.4 Grafické uživatelské rozhraní

Pro pohodlné ovládání stabilizované základny a zobrazování jednotlivých průběhů proudu, natočení kloubu aj. jsem vyvinul grafické uživatelské rozhraní (viz. obr. 5.10). GUI umí pracovat s daty přicházejícími po CAN sběrnici stabilizované základny. Pro připojení CAN sběrnice k desktopovému počítači byla použita PCIExpress karta s dvěma CAN vstupy od fy. Kvaser<sup>9</sup>. Pro vývoj grafického rozhraní jsem se rozhodl využít prostředí C# a to hlavně z důvodu dostatečné podpory tohoto jazyka v oblasti ovladačů CAN karty a také své zkušenosti s tímto jazykem.

GUI umožňuje činnost ve dvou základních režimech, podle toho jaká data zobrazuje. Režim je nutné zvolit ještě před začátkem přijímání dat a během činnosti jej zatím nelze bohužel měnit. Prvním režimem je režim monitorování provozu na sběrnici stabilizované základny. Druhým režimem je režim zobrazování aktuálního odhadu Kalmanova filtru. GUI obsahuje možnost zobrazení chyby polohy při posledním průjezdu IRC čidla referenčním pulzem, také umožňuje nastavovat frekvenci dolnopropustního interního filtru gyroskopů. Dále obsahuje možnost nahrávání všech průběhů do textového souboru ve formátu hodnot oddělených čárkami. Tento formát je snadno čitelný pro většinu aplikací jako jsou Microsoft Excel, OpenOffice Calc či Matlab.

K zobrazení průběhů byla použita knihovna ZedGraph<sup>10</sup> s licencí LGPL, která umožňuje její volné použití i v komerčních projektech, pokud nedojde ke změně zdrojového kódu knihovny. Grafový objekt přidává další funkčnosti grafickému rozhraní jako je možnost přibližování a oddalování v okně grafu a možnost uložení plátna grafu do obrazkových souborů různých typů (nejvýznamnější z nich je formát PNG).



Obrázek 5.10. Grafické uživatelské rozhraní.

<sup>9</sup><http://www.kvaser.com/>

<sup>10</sup><http://www.zedgraph.org/>

# Kapitola 6

## Experimentální ověření

Tato kapitola popisuje výsledky, kterých se mi podařilo dosáhnout při experimentech se stabilizací optické osy základny. Jsou zde zobrazeny průběhy stabilizace při dvou typických průbězích pohybu pevné části (letadla).

### 6.1 Výsledná funkčnost stabilizované základny

Po zapojení veškeré elektroniky popsané v předešlé kapitole a naprogramování všech potřebných procesorů a podpůrných obvodů jsem mohl přistoupit k prvním pokusům se stabilizací základny. Jelikož zatím nemám k dispozici hardware zpracovávající signál z kamery a figurující v celém systému jako polohový regulátor, provedl jsem pro začátek jen několik pokusů s rychlostní stabilizační smyčkou řídicího systému. Jelikož však obě řídicí smyčky v sobě obsahují PI regulátory bez použití antiwindupu, chová se integrační konstanta PI regulátoru jako proporcionální konstanta regulátoru polohy (resp. integrálu měřené rychlosti). Navržený regulátor se tak celkem dobře snaží držet namířenou polohu.

Princip činnosti je tedy následující. Po zapnutí napájení se spustí programy umístěné ve flash pamětech ve všech třech procesorech dvou řídicích a jedné měřicí desky. Po startu jsou zatím oba regulátory kloubů v režimu s odpojeným výstupem (viz. kap. 5), tzn. že všechny průběhy jsou již zobrazeny s reálnou vzorkovací frekvencí v podokně grafu. Pro zahájení stabilizace je nejprve nutné manuálně protočit oba klouby tak, aby byly nalezeny referenční polohy kloubů. Průjezdy referenční polohou diagnostikuje grafické rozhraní zobrazením obsahu registru chyby v dolní části grafického rozhraní. Při prvním průjezdu bude téměř jistě přečtená chyba nenulová. Po průjezdu oběma referenčními pulzy je možné manuálně zamířit požadovaný cíl a tlačítkem start v pravé dolní části GUI povolit připojení akčních zásahů na oba klouby. Stabilizace optické osy je nyní v činnosti.

Jelikož se jedná jen o stabilizaci rychlosti s PI regulátorem, dochází k integraci již několikrát zmíněných offsetů gyroskopů a optická osa „ujíždí“ pomalu z namířeného cíle pryč. Stlačením tlačítka STOP v pravé dolní části grafického rozhraní je možné akční zásahy opět odpojit.



Obrázek 6.1. Laboratorní stůl se stabilizovanou základnou během experimentu.

## 6.2 Zapojení experimentu

Na obr. 6.1 můžete vidět fotku celého uspořádání – stabilizovanou základnu, desktopový počítač, na kterém běží aplikace grafického rozhraní a vpravo malá televize zobrazující signál optické kamery. Abych byl schopen porovnat jakým způsobem se podařilo základně při stabilizaci optické osy eliminovat působení poruchových pohybů, vybavil jsem stojan základny<sup>1</sup> senzorem ADIS 16350 s deskou SPIBoard (oba jsou na vidět na obr. 6.1 na dolní rovné ploše stojanu). Do desky SPIboard jsem pak naprogramoval program realizující odhad polohových úhlů popsaný v kapitole 4.4.3. Desku jsem také připojil ke sběrnici stabilizované základny, takže je možné přijímat současně odhady polohy pevné části základny a měřené úhlové rychlosti vnitřního závěsu základny.

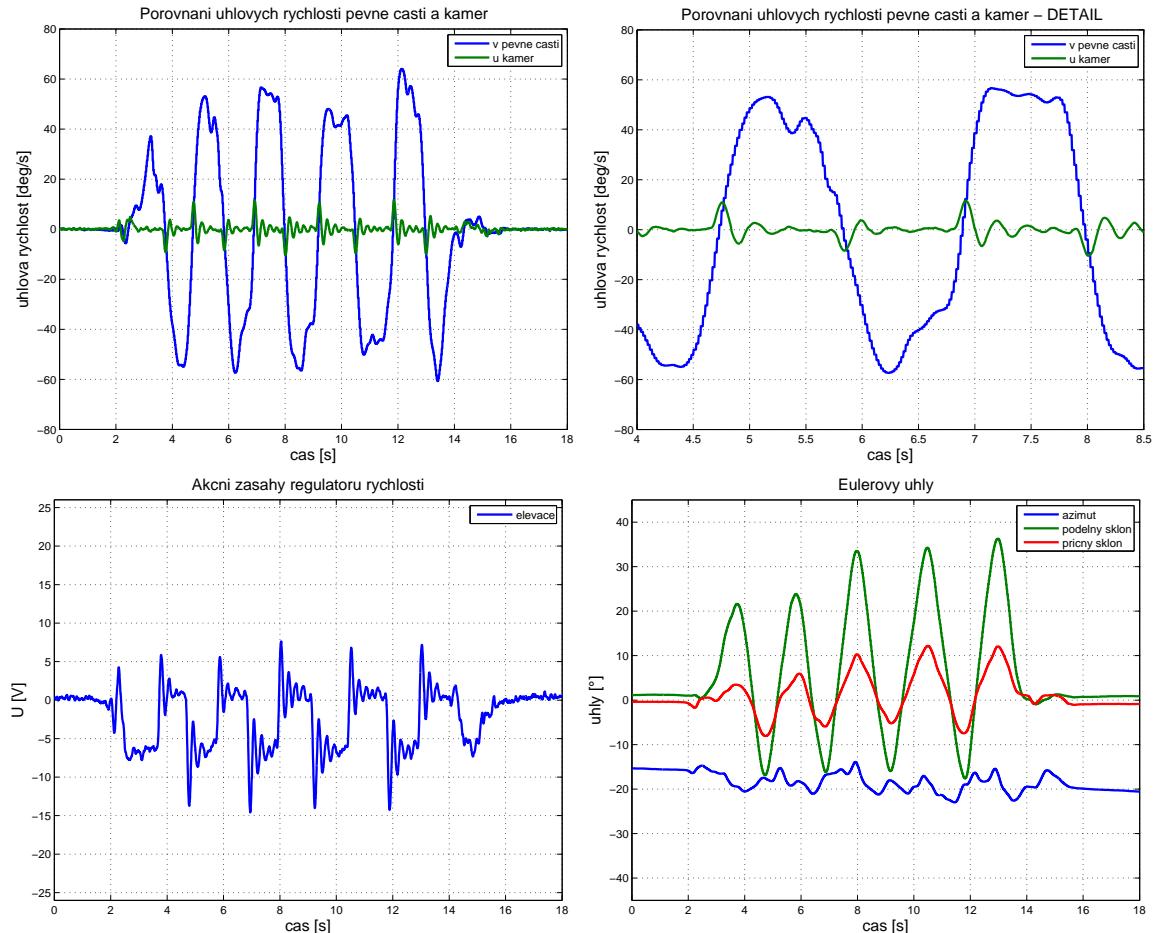
## 6.3 Výsledky

Na obr. 6.2 můžete vidět výsledky prvního pokusu stabilizace. Jako poruchový pohyb jsem vybral kývavý pohyb okolo elevační osy základny. Jelikož úhel azimutálního kloubu byl nulový, odpovídá tento pohyb rotaci okolo příčné osy (podélnému sklonu). Výsledný odhad kalmanovým filtrem přepočtený na Eulerovy úhly můžete vidět na obr. 6.2 vpravo dole<sup>2</sup>. Dva horní obrázky v obr. 6.2 zobrazují porovnání senzorů úhlové rychlosti v osách elevace a rotaci odpovídající stejné ose v souřadné soustavě pevné části (jakoby podélný

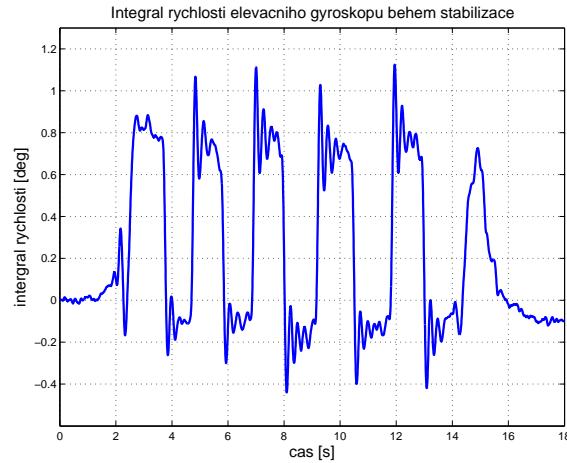
<sup>1</sup>Stojan je vlastně pevně spojen s pevnou částí základny.

<sup>2</sup>Jelikož před zaznamenáním tohoto průběhu jsem nechal nejprve Kalmanův filtr dostatečně dobře odhadnout ofsety, nedochází ani k příliš velké chybě určení polohy azimu tupevné části.

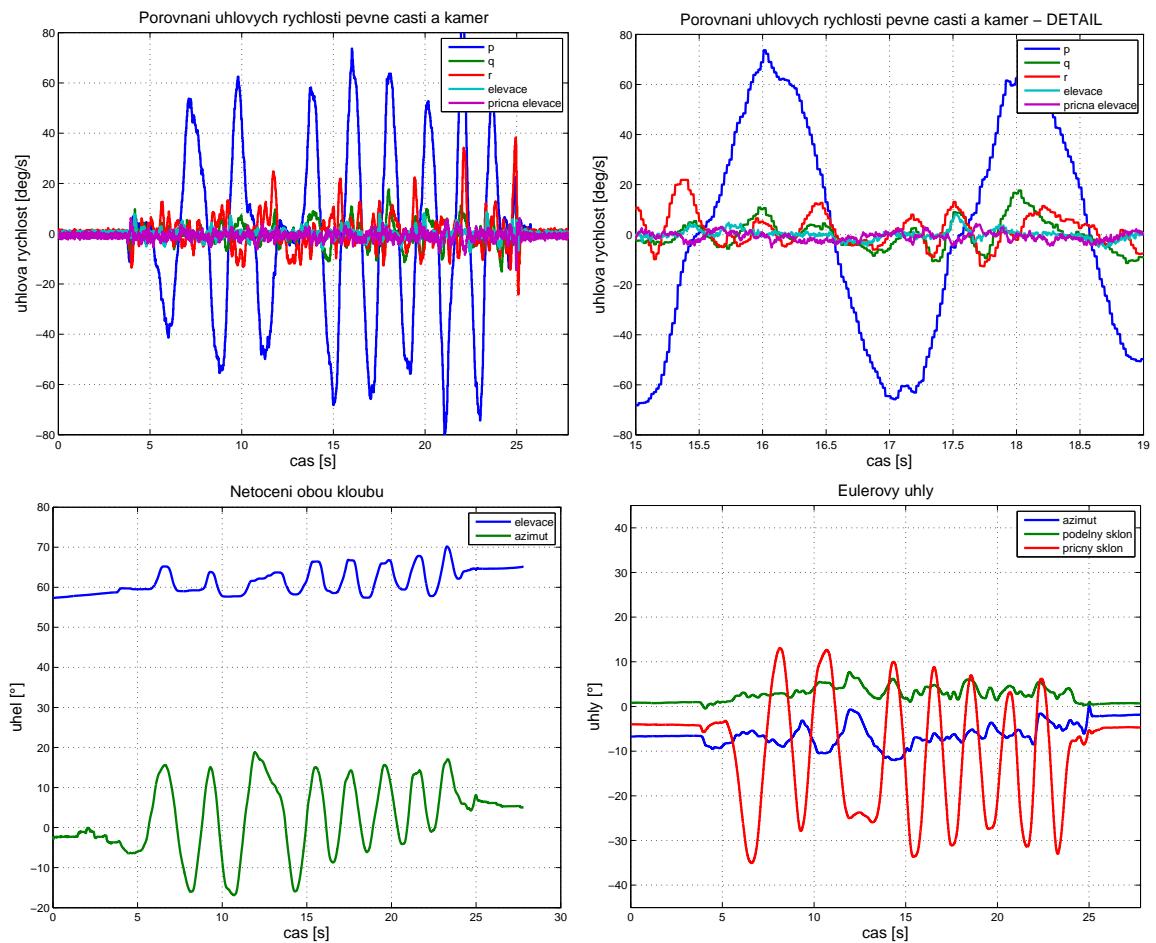
sklon letounu  $q$ ). Jak vidíte došlo k velmi dobrému zatlumení poruchového pohybu. Při výpočtu integrálu (kumulativní sumy) průběhu senzoru elevace jsem zjistil, že docházelo k pohybu v rozsahu necelého  $1^\circ$  (viz. obr. 6.3). V detailu v obr. 6.2 vpravo nahoře si můžete povšimnout jakoby jemně kmitavého děje (zelený průběh). První a druhý překmit je však ve skutečnosti naprosto v pořádku. První překmit značí nedokonalost stabilizace a druhý s opačným znaménkem značí to, jakým způsobem vracela integrační složka PI regulátoru rychlosti polohu na původní místo. Ve skutečnosti k úplnému potlačení regulační odchylky polohy nedošlo z důvodu toho, že P regulátory trpí problémem ustálené regulační odchylky jak můžete vidět na obr. 6.3. Přidávat další integraci pro odstranění této odchylky je však naprosto zbytečné. Tuto odchylku vyrovnaná polohová zpětná vazba realizovaná jako vizuální či slepé směrování. Poslední průběh na obr. 6.2 vlevo dole je průběh akčního zásahu (nastaveného napětí) na elevační motor během regulačního pochodu.



Obrázek 6.2. Dosažené výsledky při poruchovém pohybu, který je souhlasný s osou elevace. Nahoře vidíte porovnání úhlových rychlostí ve vnitřním závěsu a v pevné části. Vpravo dole můžete vidět odhad polohy pevné části Kalmanovým filtrem a v levé dolní části průběhy akčních zásahů na elevační motor.



Obrázek 6.3. Integrál úhlové rychlosti elevačního senzoru během stabilizace.

Obrázek 6.4. Dosažené výsledky při poruchovém pohybu rotace okolo podélné osy letadla (kamery jsou namířeny ve směru letu pod úhlem asi  $30^\circ$  od singulárního bodu). Nahoře vidíte porovnání úhlových rychlostí ve vnitřním závěsu a v pevné části. Vpravo dole můžete vidět odhad polohy pevné části Kalmanovým filtrem a v levé dolní části natočení obou kloubů základny – aby došlo k potlačení poruchového pohybu je nutné pohyb kamer skládat z pohybů oběma kloubů.

Na obr. 6.4 můžete vidět výsledky druhého pokusu stabilizace, ve kterém jsem jako poruchový pohyb vybral kývavý pohyb okolo podélné osy „letounu“. Jak vidíte na levém dolním obrázku úhel elevace je po celou dobu asi  $60^\circ$  a azimut je nulový. Při rotaci pevné části okolo podélné osy, musí základna tento pohyb tlumit skládáním pohybu z pohybu v obou osách elevace i azimutu. V horní části obrázku opět můžete vidět porovnání měřených úhlových rychlostí pevné části a úhlových rychlostí vnitřního závěsu.

Na webových stránkách projektu [20], případně také na přiloženém CD naleznete několik videozáznamů činnosti stabilizované základny, mezi nimi i oba zmíněné pokusy.

# Kapitola 7

## Závěr

Výsledkem této diplomové práce je funkční prototyp stabilizované základny pro kamerový systém bezpilotního letounu. V práci byly postupně popsány jednotlivé možnosti realizace stabilizované základny, byla podrobně popsána koncepce, výpočetní návrh i realizace řídicího a měřicího hardware umístěného přímo uvnitř stabilizované základny. O funkčnosti vyrobeného prototypu jste se mohli přesvědčit v kap. 6, kde byly představeny výsledky dvou krátkých pokusů se stabilizací optické osy.

Součástí této práce byla také tvorba prezentace projektu stabilizované základny na webových stránkách [20]. Na stránkách naleznete další informace o vývoji projektu, veškeré zdrojové kódy navržených algoritmů a také se zde můžete přesvědčit o výsledcích činnosti stabilizované základny prostřednictvím několika krátkých videí. Jelikož projekt samotný bude ještě pokračovat dále, naleznete na těchto stránkách další informace i po dokončení této diplomové práce.

Činnost algoritmů inerciální stabilizace byla úspěšně ověřena v laboratorních podmínkách. Zůstává tak prostor pro další zlepšování při testování v reálném prostředí v podmínkách, kdy kolem základny proudí vzduch (např. na autě nebo vrtulníku). V této situaci také očekáváme případné komplikace způsobené velkou hmotností osazeného příslušenství, zejména infračervené kamery. Je totiž možné, že azimutální motor základny nebude dostatečně silný na to, aby byl schopný při tak objemném příslušenství odolávat působení proudění vzduchu přes nerovnou část krytu konstrukce. Možným řešením může být zprvu jen úprava tvaru krytu do aerodynamického tvaru, později také výměna stávajícího motoru jiným výkonnějším typem.

Výsledky společné práce s kolegou Jaroslavem Žohou byly představeny v příspěvku s názvem „Line-of-sight stabilization of an airborne camera system“ na studentské konferenci Pegasus AIAA 2008 [6].

# Literatura

- [1] Euclidean Space. [online]. c2008, [cit.2008-05-20].  
URL <http://www.euclideanspace.com/>
- [2] PRESS, W. [et. al.]: *Numerical Recipes in C++*. Cambridge university Press, 2002, ISBN 0-521-75033-4, druhé vydání.
- [3] FRANKLIN, G.F.: *Feedback Control of Dynamic Systems*. USA: Prentice Hall, 2002, ISBN 0-13-149930-0.
- [4] FRANKLIN, G.F.: *Digital Control of Dynamic Systems*. USA: Prentice Hall, 2006, ISBN 0-201-82054-4.
- [5] KIM, A.; GOLNARAGHI, M.F.: A quaternion-based orientation estimation algorithm using an inertial measurement unit. In *Position Location and Navigation Symposium*, 2004.
- [6] ŘEZÁČ, M.; ŽOHA, J.; HANIŠ, T.: Line-of-sight stabilization of an airborne camera system. In *PEGASUS-AIAA Student Conference*, 2008, praha.
- [7] HANIŠ, T.: *Stabilizovaná platforma pro UAV prostředky: zpracování signálu z inerciálních čidel*. Diplomová práce, Praha: České vysoké učení technické. Fakulta elektrotechnická. Katedra řídicí techniky., 2008.
- [8] HAVLENA, V.: *Odhadování a filtrace (doplňkové skriptum)*. Praha : Vydavatelství ČVUT, 2002.
- [9] ŠTECHA, J.; HAVLENA, V.;: *Teorie dynamických systémů*. Praha: Vydavatelství ČVUT, 2005.
- [10] HILKERT, J.,M.: Inertially Stabilized platforms – Concepts and principles. *IEEE control systems magazine*, 2008.
- [11] KUMAR, S.N.; JANN, T.: Estimation of attitudes from a low-cost miniaturized inertial platform using Kalman Filter-based sensor fusion algorithm. In *SADHANA - Academy Proceedings in Engineering Sciences*, Indie: Indian Academy of Sciences.  
URL <http://www.ias.ac.in/sadhana/Pdf2004Apr/Pe1161.pdf>
- [12] KENNEDY, J.P.; KENNEDY, L.R.: Direct Versus Indirect Line of Sight (LOS) Stabilization. In *IEEE Transactions on control systems technology*, 2003.

- [13] STEVENS, B.; LEWIS, F.: *Aircraft control & simulation*. USA: John Wiley & Sons, Inc., 2003, ISBN 0-471-37145-9, URL [http://media.wiley.com/product\\_data/excerpt/59/04713714/0471371459.pdf](http://media.wiley.com/product_data/excerpt/59/04713714/0471371459.pdf).
- [14] RANDAL, W.B: *State Estimation for Micro Air Vehicles*. Německo: Springer-Verlag, 2007, ISBN 978-3-540-72695-1.  
URL <http://www.springerlink.com/content/l5044wq525345107/fulltext.pdf>
- [15] VALENTINE, R.: *Motor control electronics handbook*. USA: McDraw-Hill, 1998, ISBN 0-07-066810-8.
- [16] SPONG, W.,M.; HUTCHINSON, S.; VIDYASAGAR, M.: *Robot Modeling and Control*. USA: John Wiley & Sons, Inc., 2006, ISBN 0-471-64990-8.
- [17] ŠTECHA, J.: *Optimální rozhodování a řízení*. Praha: Vydavatelství ČVUT, 1999.
- [18] HAVLENA, V.; ŠTECHA, J.: *Moderní teorie řízení*. Praha : Vydavatelství ČVUT, 2000, ISBN 80-01-02095-9.
- [19] ŽOHA, J.: *Elektronická stabilizace optické osy kamerového systému pro bezpilotní letoun*. Diplomová práce, Praha: České vysoké učení technické. Fakulta elektrotechnická. Katedra řídicí techniky., 2008.
- [20] ŘEZÁČ, M.: Stabilizovaná základna pro bezpilotní letoun MANTA. [online]. c2008, [cit.2008-05-20].  
URL <http://dce.felk.cvut.cz/mamok/>
- [21] ŘEZÁČ, M.: Odhad polohových úhlů tělesa Kalmanovým filtrem., 2008, [online] semestální práce.  
URL <http://dce.felk.cvut.cz/mamok/doc/kfdoc.pdf>

## Příloha A

### Část zdrojového kódu provádějící algoritmus EKF

Následující zdrojový kód je hlavní smyčka algoritmu rozšířeného Kalmanova filtru (EKF) v Matlabu. Jako vstupní data byla použita reálná data naměřená senzorem ADIS 16350. V algoritmu můžete vidět nastavení jednotlivých časových konstant, kovariačních matic šumu měření a šumu procesů a chyby odhadu. Po každém výpočtu nového odhadu/predikce quaternionu je nutné provést jeho normalizaci tak, aby absolutní hodnota quaternionu zůstala jednotková.

```
0001 %% Extended Kalman filter
0002
0003 p_ = diag([1,1,1,1,1^2,1^2,1^2,1,1,1]);
0004 x_=[1;0;0;0;0;0;0;0;0]; % pocatecni apriorni odhad
0005 % x1..x4 quaterniony, x5..x7 gyro ofsety, x8..x10 gyro odhadu,
0006 x_(1:4)=x_(1:4)/sqrt(x_(1:4)'*x_(1:4)); %normalizace quaternionu
0013
0014 R=diag([200^2,200^2,200^2,1^2,1^2,1^2]);
0015 Q=diag([0,0,0,0,.0001^2,.0001^2,.0001^2,1^2,1^2,1^2]);
0016
0017 y=[ax,ay,az,pp,qq,rr]; % vystupy = senzory
0018 Agx=9.81;
0019
0020 tic
0021 for i = 1:1:length(y)-1,
0022 %C = derivace vystupni rovnice podle stavu, dosazen odhad x_
0023 % pro zjednoduseni napred uvazuji jen Agx=9.81;Agy=0;Agz=0;
0024 C =
0025 [-2*x_(3)*Agx, 2*x_(4)*Agx,-2*x_(1)*Agx, 2*x_(2)*Agx, 0,0,0,0,0,0;
0026 2*x_(2)*Agx, 2*x_(1)*Agx, 2*x_(4)*Agx, 2*x_(3)*Agx, 0,0,0,0,0,0;
0027 2*x_(1)*Agx,-2*x_(2)*Agx,-2*x_(3)*Agx, 2*x_(4)*Agx, 0,0,0,0,0,0;
0028 0, 0, 0, 0, 1, 0, 0, 1, 0, 0;
0029 0, 0, 0, 0, 0, 1, 0, 0, 1, 0;
0030 0, 0, 0, 0, 0, 0, 1, 0, 0, 1;
```

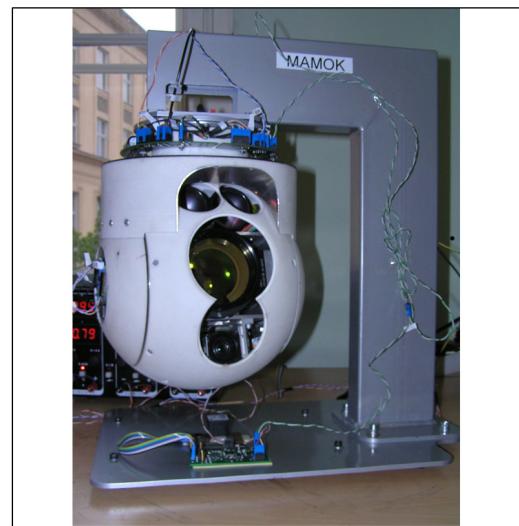
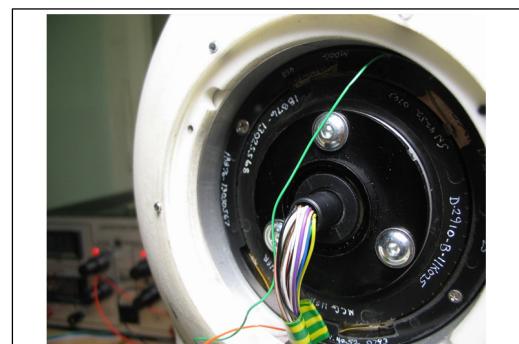
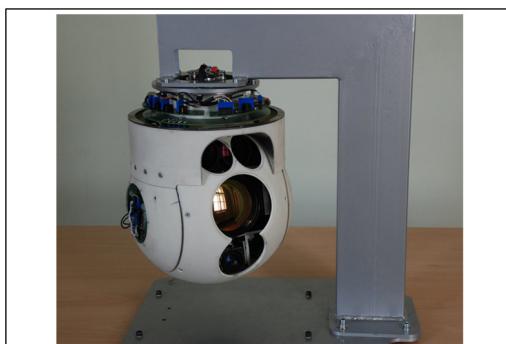
## II PŘÍLOHA A. ČÁST ZDROJOVÉHO KÓDU PROVÁDĚJÍCÍ ALGORITMUS EKF

```
0031           ] ;
0032
0033 % Datovy krok
0034 k = p_*C'*inv(C*p_*C'+R);
0035
0036 e = (y(i,:)'-[2*Agx*(x_(2)*x_(4)-x_(1)*x_(3));
0037                               2*Agx*(x_(1)*x_(2)+x_(3)*x_(4));
0038                               (x_(1)^2-x_(2)^2-x_(3)^2+x_(4)^2)*Agx;
0039                               x_(8)+x_(5);
0040                               x_(9)+x_(6);
0041                               x_(10)+x_(7);
0042 ]);    % nelinearni rovnice
0043
0044 p = p_ - k*(C*p_*C'+R)*k';
0045
0046
0047 absol = sqrt(ax(i)^2 + ay(i)^2 + az(i)^2);
0048 if ((absol-abs(ax(i)))<.5), k(5,:)=0; end
0049 if ((absol-abs(ay(i)))<.5), k(6,:)=0; end
0050 if ((absol-abs(az(i)))<.5), k(7,:)=0; end
0051
0052 x = x_ + k*e;
0053 x(1:4)=x(1:4)/sqrt(x(1:4)'*x(1:4));
0061
0062 % Casovy krok = predikcni
0063 % A = derivace f(x) podle x, dosazen odhad x
0064 A=(T/2*[0 -(x(8)) -(x(9)) -x(10) 0 0 0 -x(2) -x(3) -x(4);
0065                               (x(8)) 0 (x(10)) -(x(9)) 0 0 0 x(1) -x(4) x(3);
0066                               (x(9)) -(x(10)) 0 (x(8)) 0 0 0 x(4) x(1) -x(2);
0067                               (x(10)) (x(9)) -(x(8)) 0 0 0 -x(3) x(2) x(1);
0068                               zeros(6,10)]+diag([1,1,1,1,1,1,.99,.99,.99]));
0069
0070 x_=(T/2*[-x(2)*(x(8))-x(3)*(x(9))-x(4)*(x(10));
0071                               x(1)*(x(8))+x(3)*(x(10))-x(4)*(x(9));
0072                               x(1)*(x(9))-x(2)*(x(10))+x(4)*(x(8));
0073                               x(1)*(x(10))+x(2)*(x(9))-x(3)*(x(8));
0074                               zeros(6,1)]+[x(1:4,1);1*x(5:7,1);.99*x(8:10,1)]);
0075 % nelinearni rovnice
0076 x_(1:4)=x_(1:4)/sqrt(x_(1:4)'*x_(1:4));
0077 p_ = A*p*A' + Q;
0078
0079 end
0080 toc
```



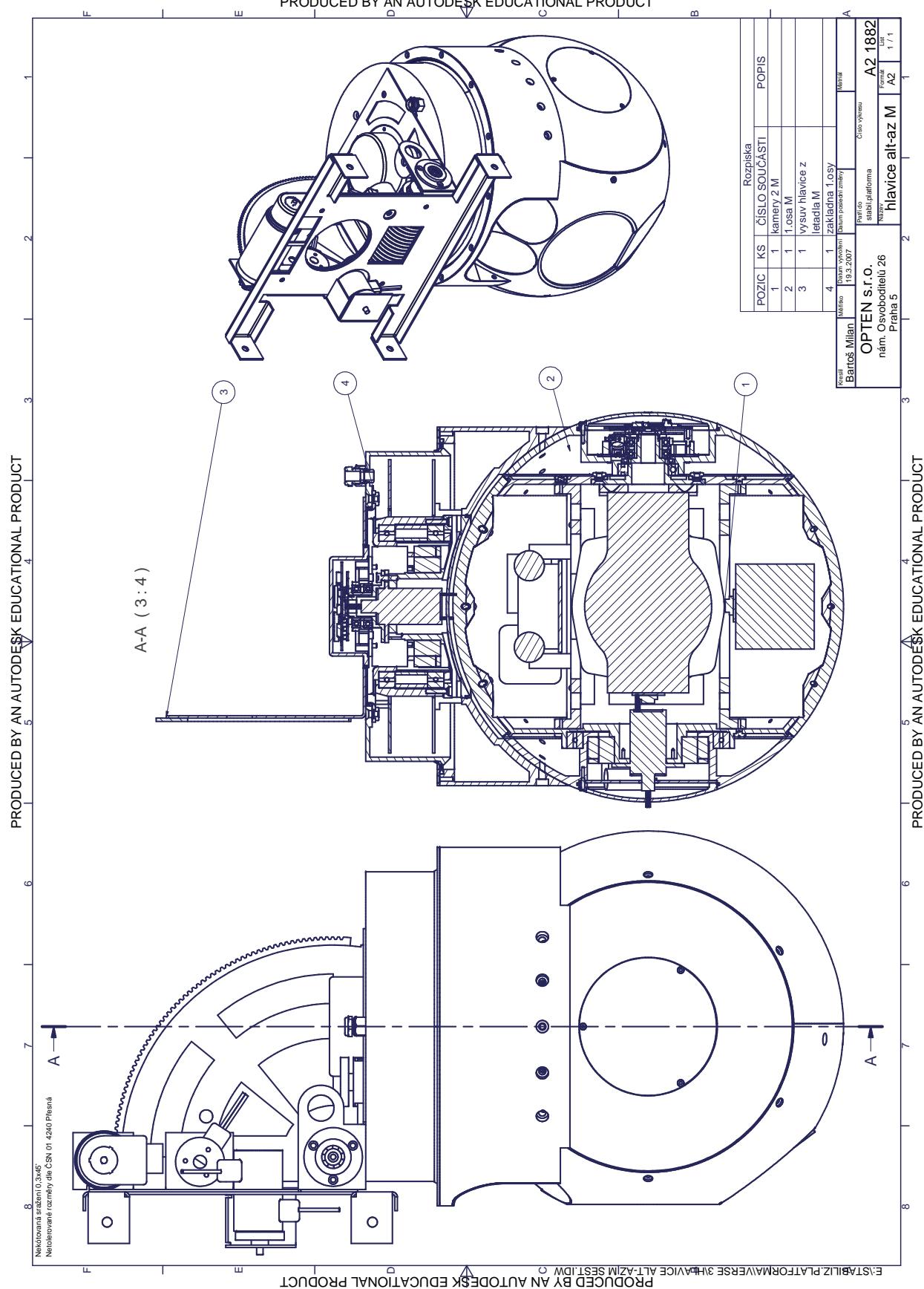
## Příloha B

### Fotky a výkres sestavy základny



V

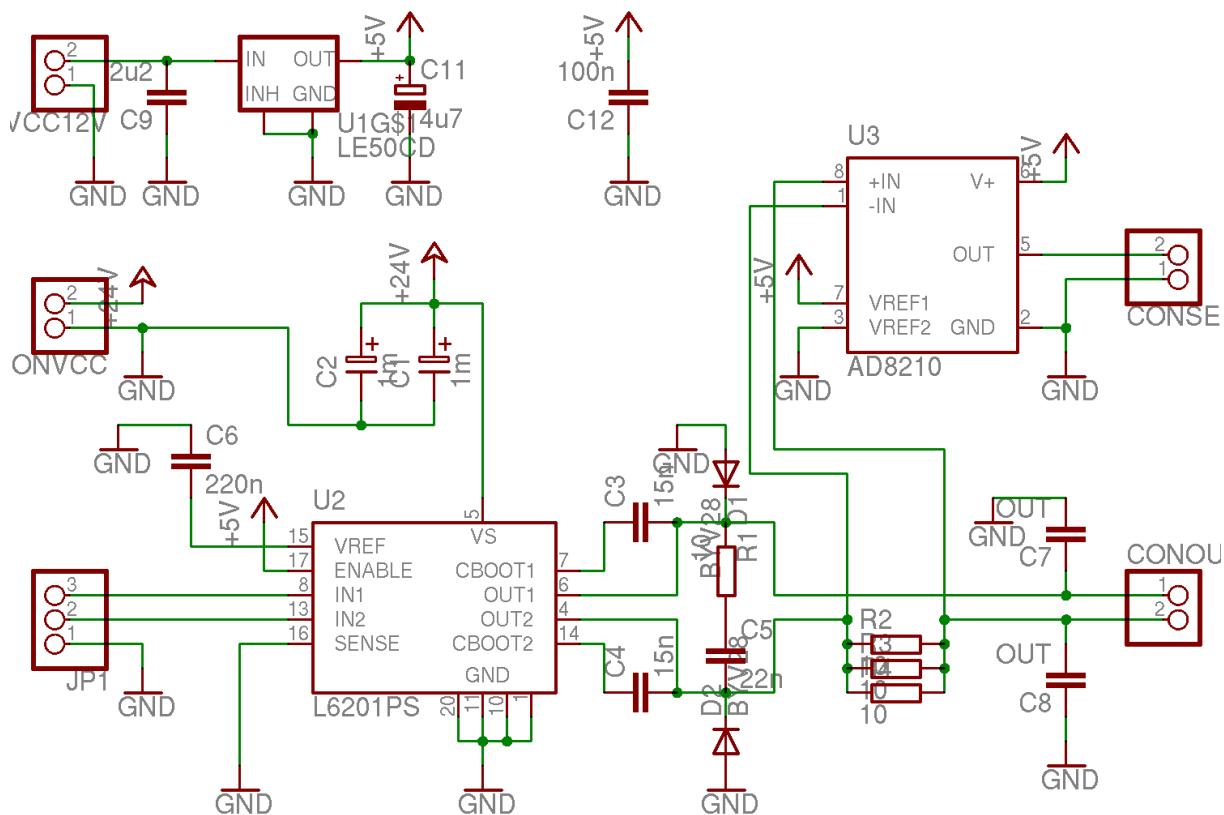
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

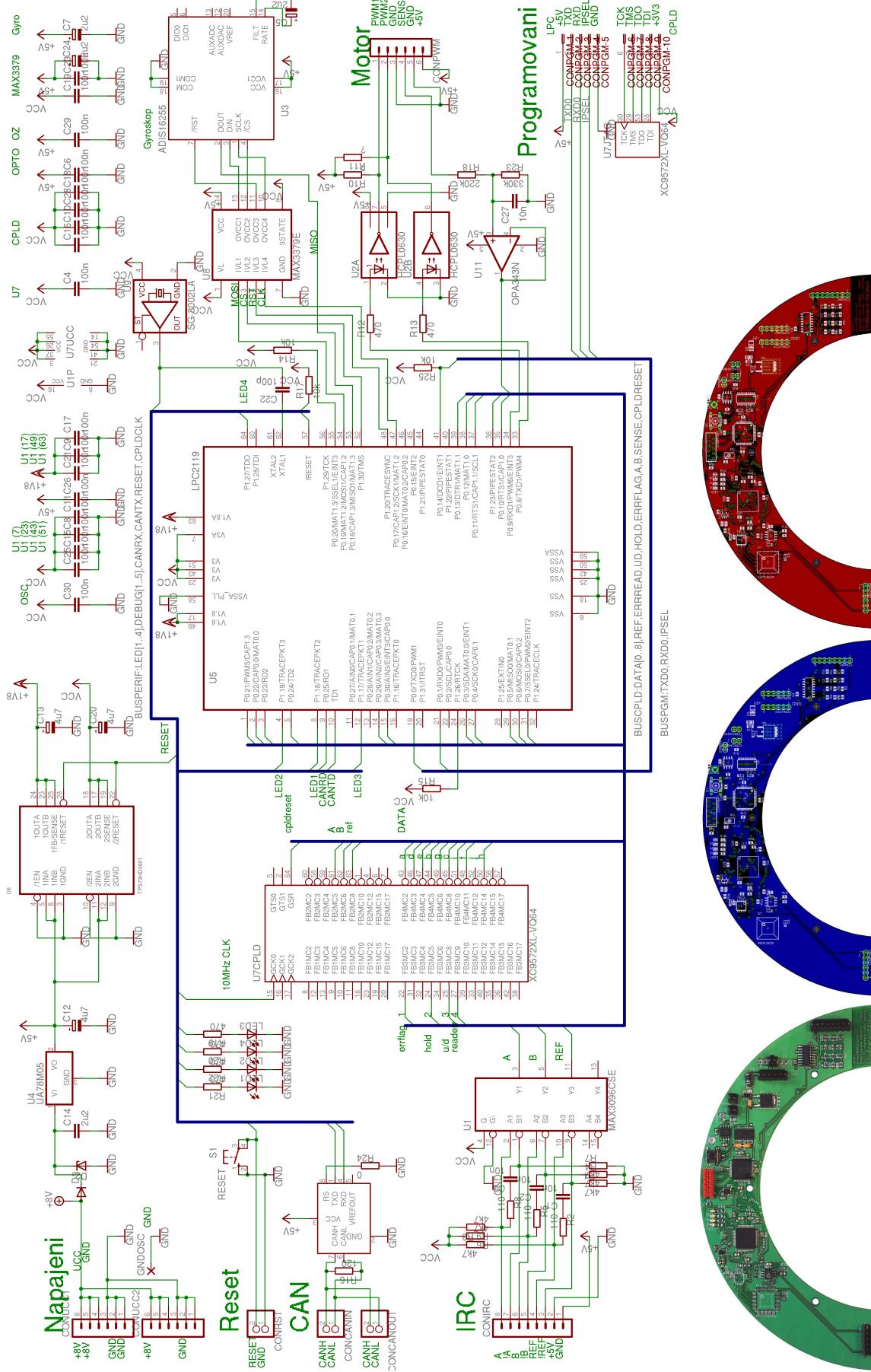


# Příloha C

## Schématika elektroniky

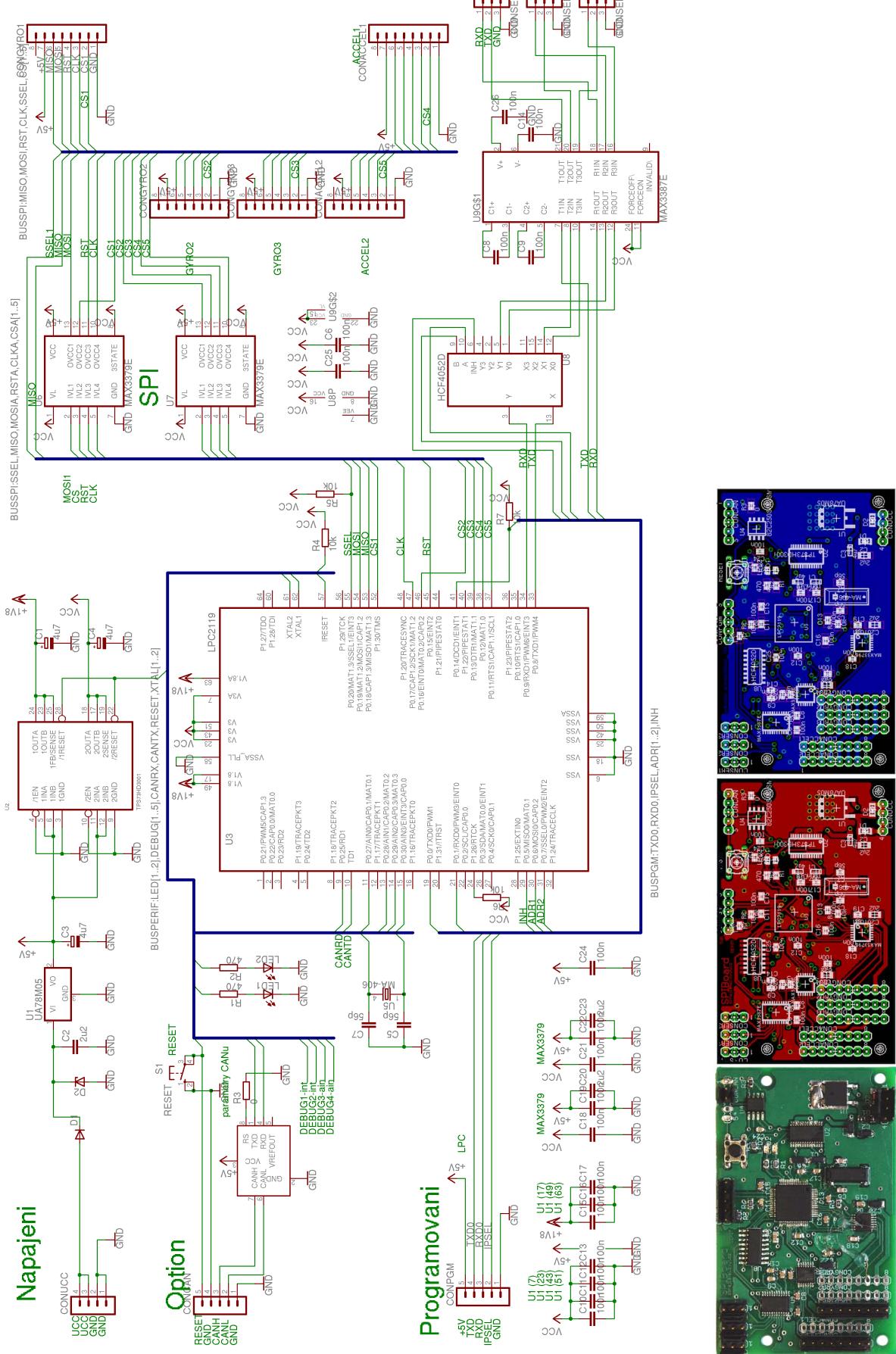
Na následujících stránkách uvádím výtisky obrazovky schémat z programu Eagle. Jedná se postupně o desky H-můstku (HmustekAZ), řídicí desky azimutálního kloubu (Cpubo-ardAZ) a senzorové desky (Spiboard).





VIII

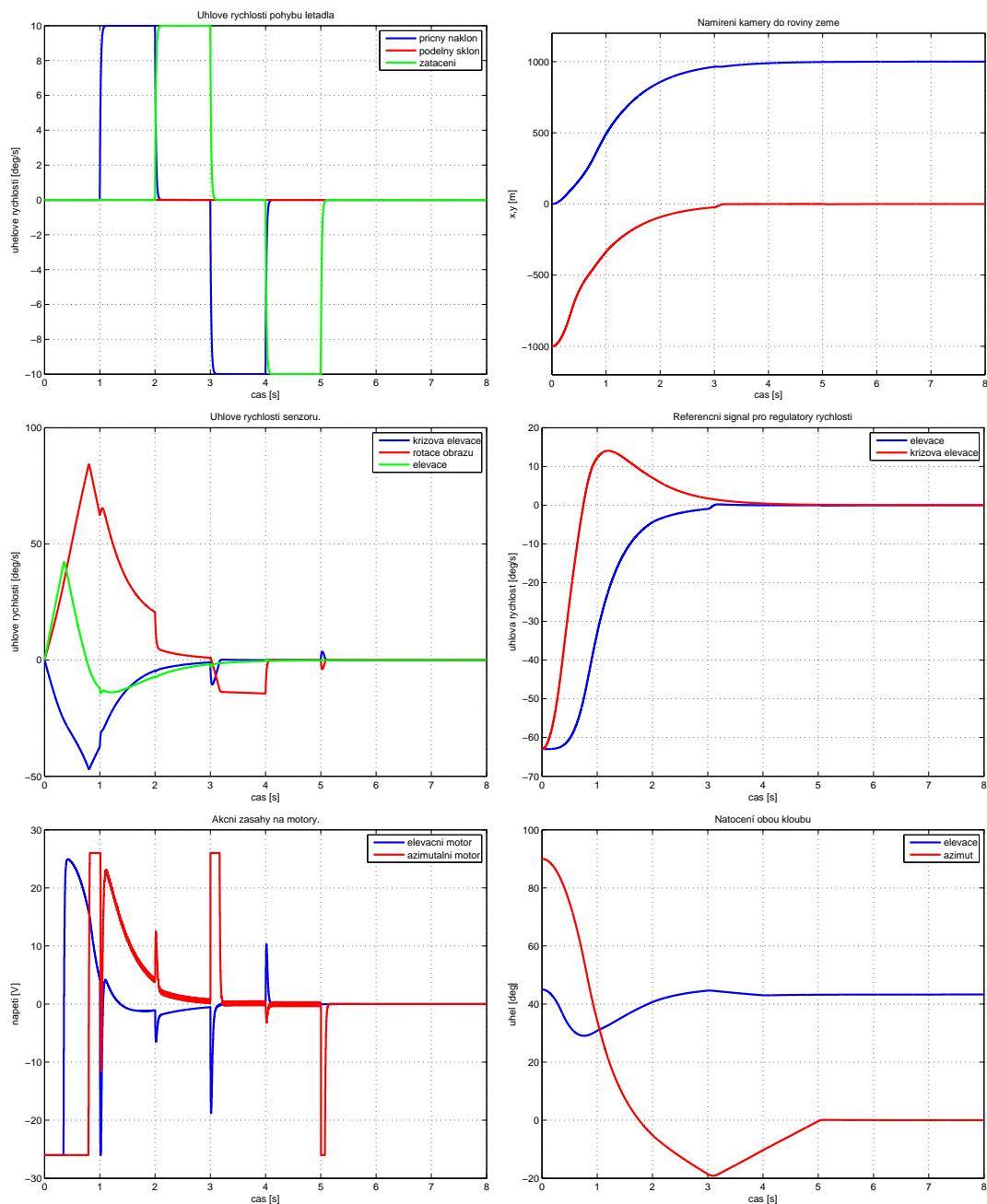
## PŘÍLOHA C. SCHÉMATA ELEKTRONIKY

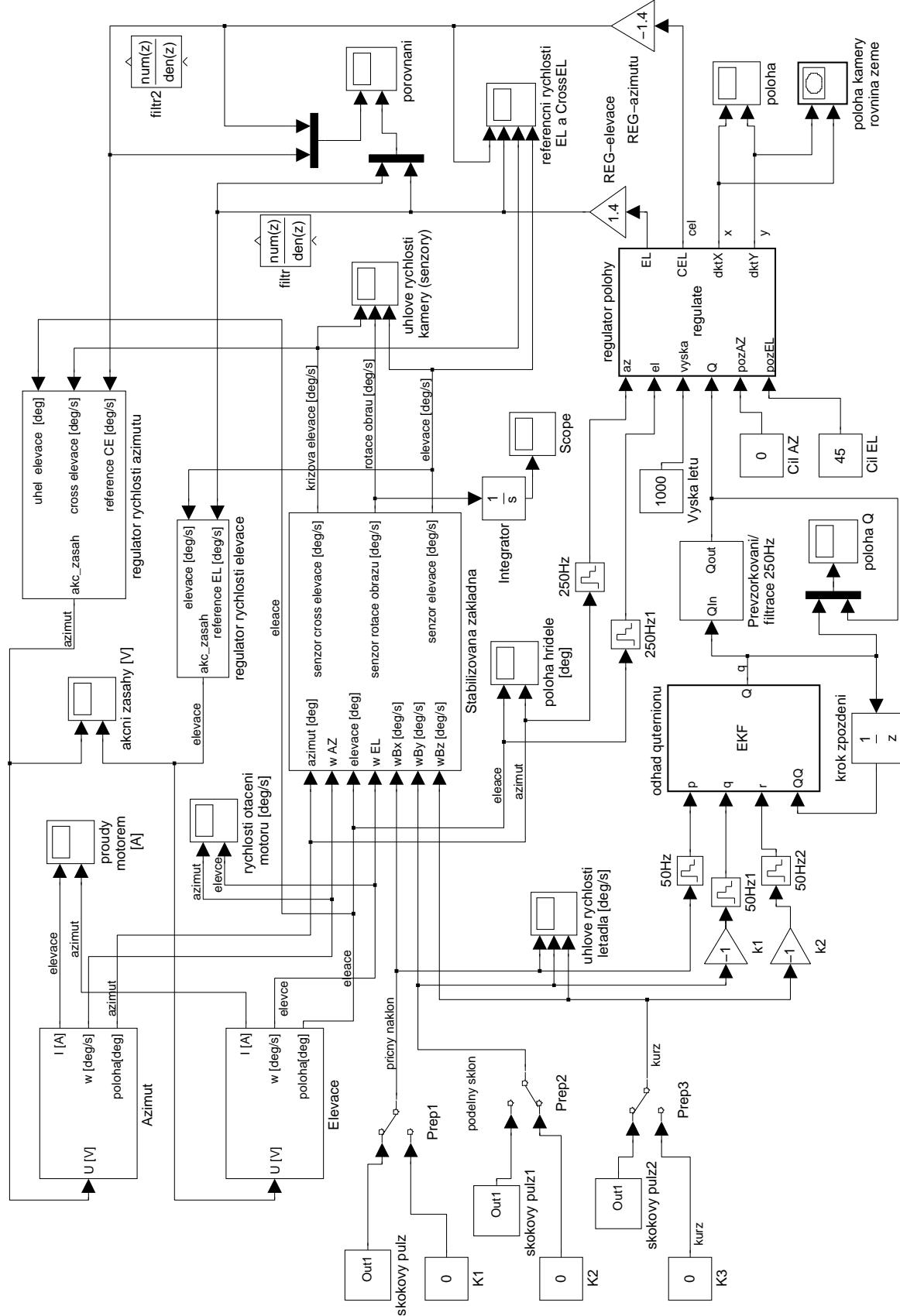




## Příloha D

### Simulinkový model základny





## Příloha E

### Obsah přiloženého CD

- Tento text ve formátu PDF
- Fotky stabilizované základny a video její činnosti
- Simulační modely v prostředí Matlab (redukovaná inerciální jednotka, model základny v simulinku)
- Zdrojové kódy k navrženým programům a jejich HTML dokumentace. Jedná se o programy desek CPUboardEL, CPUboardAZ a SPIboard. Dále pak zdrojové kódy redukované inerciální jednotky, programu CPLD obvodu a grafického rozhraní.