

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA ELEKTROTECHNICKÁ  
KATEDRA KYBERNETIKY



## BAKALÁŘSKÁ PRÁCE

Návrh řízení pro inerciálně stabilizovanou  
kamerovou hlavici

Praha, 2012

Autor: Petr Marek

Autor: Petr Marek  
Studijní program kybernetika a robotika (bakalářský),  
Obor robotika

Vedoucí: Ing. Zdeněk Hurák, Ph.D.  
Katedra kybernetiky, Fakulta elektrotechnická,  
České vysoké učení technické v Praze

Oponent: Ing. Michal Sojka, Ph.D.  
Katedra kybernetiky, Fakulta elektrotechnická,  
České vysoké učení technické v Praze

Rok: 2012

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

**Student:** Petr Marek

**Studijní program:** Kybernetika a robotika (bakalářský)

**Obor:** Robotika

**Název tématu:** Návrh řízení pro inerciálně stabilizovanou kamerovou hlavici

### Pokyny pro vypracování:


1. Seznamte se s existujícími prototypovými inerciálně stabilizovanými kamerovými platformami vyvinutými na pracovišti vedoucího. Věnujte stejnou pozornost struktuře HW i SW.
2. Seznamte se rovněž s pracovními postupy pro modifikaci kódu regulátorů, počínaje tvorbou kódu přes kompilaci a nahrání kódu do cílového procesoru, spuštění regulace a záznam dat, až po spravování verzí kódu se systémem Git. Demonstrujte modifikaci některých z existujících regulátorů přeladěním konstant u PID regulátoru.
3. Pro stabilizovanou platformu s přímými momentovými motory nahradte stávající zastaralé grafické uživatelské rozhraní nově vyvinutým (a používaným u novějších platforem).
4. Pro stabilizovanou platformu s přímými momentovými motory proveďte úpravy nezbytné pro převedení vývojového procesu ze systému Linux na Windows s vývojovým prostředím Eclipse.
5. Podle pokynů vedoucího navrhněte dílčí zlepšení kódu regulátorů (například pokročilá kompenzace tření či zpoždění).

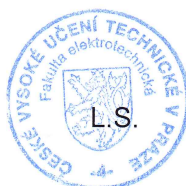
### Seznam odborné literatury:

- [1] Řezáč, M.: Návrh řízení pro systém stabilizace optické osy pro bezpilotní letoun. Diplomová práce, ČVUT, 2008.
- [2] Žoha, J.: Elektronika pro systém stabilizace optické osy kamerového systému. Diplomová práce, ČVUT, 2008.

**Vedoucí bakalářské práce:** Ing. Zdeněk Hurák, Ph.D.

**Platnost zadání:** do konce zimního semestru 2012/2013

  
prof. Ing. Vladimír Mařík, DrSc.  
vedoucí katedry



  
prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 9. 1. 2012



## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 25.5.2012



---

Petr Marek

## Poděkování

Děkuji především vedoucímu bakalářské práce Ing. Zdeňkovi Hurákovi, Ph.D. za vedení této bakalářské práce a také všem ostatním kolegům, především Ing. Martinu Řezáčovi, Ing. Jaroslavu Žohovi a Bc. Janu Saláškoví, za ochotu a cenné rady.

Dále bych rád poděkoval své rodině a blízkým přátelům za jejich podporu během celé doby mého studia.

## Abstrakt

Cílem této bakalářské práce je demonstrovat kompletní zvládnutí vývojového procesu a provést nezbytné úpravy v existujícím softwaru pro stabilizovanou základnu pro kamerový systém, aby bylo možné využít její veškerý potenciál, například pro různé výzkumné a výukové aktivity. Je potřeba nahradit zastaralé grafické uživatelské rozhraní novým, ale také zpřehlednit a modifikovat zdrojové kódy a provést implementaci nových regulátorů. Dalším z úkolů je převést celý vývojový proces ze systému Linux na Windows s vývojovým prostředím Eclipse. Samozřejmostí je prostudování si materiálů k již existujícím prototypovým inerciálně stabilizovaným kamerovým platformám. Aktualizací zdrojových kódů stabilizované platformy a souvisejících aplikací dojde ke sjednocení stabilizované základny s ostatními nově vyvíjenými. Výsledkem bakalářské práce jsou funkční kódy k inerciálně stabilizované základně pro kamerový systém. Zveřejněna v této práci je však pouze jejich HTML dokumentace, která je přiložena na CD. Bakalářská práce by měla posloužit jako ucelený blok informací týkající se kamerové hlavičky a proto je zde uveden i stručný popis o mechanické konstrukci a elektronice.

## Abstract

The objective of this bachelor thesis is a demonstration of understanding of the development process and implementation of requiring software adjustments for the stabilized camera platform and at the same time to use it to extend its possibilities to its full potential – for instance to extend feasibility of its usage for research and in education. Outdated graphical user interface is being replaced, source codes clarified and modified and new controllers implemented. Other task is to transfer the development process from Linux to Windows via Eclipse development environment. As research fundamentals are used prototypes of the existing inertially stabilized camera platforms. By updating the source codes and other applications, unification of the platform with other ones currently being developed is achieved. The result of this thesis is a set of functional codes for the inertially stabilized camera platform. Its documentation, which is enclosed to this thesis on CD, is presented only in HTML form. This thesis endeavours to provide comprehensive information relating to camera platforms. A summary on mechanical construction and electronics is included as well.



# Obsah

Seznam obrázků	vii
<b>1 Úvod</b>	<b>1</b>
1.1 Historie projektu . . . . .	1
1.2 Současný stav projektu . . . . .	2
1.3 Motivace . . . . .	2
1.4 Cíl bakalářské práce . . . . .	3
1.5 Shrnutí výsledků . . . . .	4
<b>2 Mechanické uspořádání</b>	<b>5</b>
2.1 Konstrukce . . . . .	5
2.1.1 Tip-Tilt . . . . .	6
2.1.2 El-Az . . . . .	6
2.2 Terminologie . . . . .	8
<b>3 Elektronika</b>	<b>9</b>
3.1 Použitá koncepce . . . . .	9
3.2 Sběrnice, hardware, senzory . . . . .	11
3.2.1 Sběrnice . . . . .	11
3.2.2 Hardware a senzory . . . . .	12
<b>4 Řídicí systém pro inerciální stabilizaci</b>	<b>15</b>
4.1 Stabilizace obrazu . . . . .	15
4.2 Diskrétní PID regulátory . . . . .	17
4.3 Dynamika závěsu a motoru . . . . .	19
4.4 Kaskádní regulace . . . . .	21
4.5 Režimy řízení . . . . .	23
4.5.1 Režim stop - STOP . . . . .	23

4.5.2	Stabilizace optické osy - LOS . . . . .	23
4.5.3	Inicializace - INIT . . . . .	23
4.5.4	Ovládání joystickem - JOY . . . . .	24
4.5.5	Řízení kloubových úhlů - IRC . . . . .	25
4.5.6	Kalibrace - CALIB . . . . .	26
<b>5</b>	<b>Práce s kódy</b>	<b>27</b>
5.1	Tvorba programu . . . . .	27
5.2	Kompilace kódu . . . . .	29
5.2.1	Operační systém linuxového typu . . . . .	29
5.2.2	Windows s vývojovým prostředím Eclipse . . . . .	29
5.2.2.1	Návod pod Windows . . . . .	30
5.3	Správa kódu . . . . .	31
5.3.1	Návod na GIT . . . . .	32
<b>6</b>	<b>Grafické uživatelské rozhraní</b>	<b>35</b>
6.1	Úvod do GUI . . . . .	35
6.2	Záměna původního GUI za nové . . . . .	36
6.3	Ovládání a možnosti nového GUI . . . . .	37
<b>7</b>	<b>Provedené experimenty</b>	<b>43</b>
7.1	Zapojení a příprava na experimenty . . . . .	43
7.2	Průběh a výsledky experimentů . . . . .	44
7.2.1	Stabilizace elevační osy základny . . . . .	44
7.2.2	Stabilizace příčné elevační osy základny . . . . .	45
<b>8</b>	<b>Závěr</b>	<b>47</b>
8.1	Shrnutí . . . . .	47
8.2	Návrh na další práci . . . . .	47
<b>A</b>	<b>Fotografie</b>	<b>I</b>
<b>B</b>	<b>Obsah přiloženého CD</b>	<b>III</b>

# Seznam obrázků

1.1	Inerciálně stabilizovaná kamerová hlavice H240. . . . .	1
1.2	Stabilizovaná základna Svíce120 vlevo a Svíce250 vpravo . . . . .	2
2.1	Konstrukce typu Tip-Tilt, převzato z [7]. . . . .	6
2.2	Konstrukce typu El-Az, převzato z [7]. . . . .	7
2.3	Realizovaná konstrukce, převzato z [7]. . . . .	7
2.4	Terminologie - části stabilizované základny, převzato z [3] a upraveno. . .	8
3.1	Blokové schéma zapojení elektroniky, převzato z [4]. . . . .	10
3.2	RGB kamera, převzato z [3]. . . . .	12
3.3	IR kamera, převzato z [3]. . . . .	12
3.4	Laserový dálkoměr, převzato z [3]. . . . .	12
3.5	Kluzný kroužek, převzato z [3]. . . . .	13
3.6	Inkrementální senzor, převzato z [3]. . . . .	13
3.7	Přímý momentový motor - pro ilustraci, převzato z [3]. . . . .	13
4.1	Souřadné soustavy v základně, převzato z článku [2]. . . . .	16
4.2	PID regulátor, který obsahuje anti-windup. . . . .	18
4.3	Nastavení parametrů regulátorů, filtrů a konstant pomocí nového grafického uživatelského rozhraní. . . . .	19
4.4	Model dynamiky v ose elevace, včetně dynamiky motoru, kde výstupem je inerciální úhlová rychlost elevace $\omega_{Ey}$ . Dále $T_m$ je užitečný krouticí moment motoru, $\psi$ je úhel v azimutu a p, q, jsou složky vektoru inerciální úhlové rychlosti nosiče (rychlost klopení, klonění), ostatní veličiny byly uvedeny u rovnic (4.8) a (4.9). . . . .	20

4.5	Model dynamiky v ose příčné elevace, včetně dynamiky motoru, kde $\omega_{Az}$ , $\omega_{Ez}$ jsou inerciální úhlové rychlosti azimutu a příčné elevace, $T_m$ užitečný krouticí moment motoru, $\psi$ a $\theta$ je úhel v azimutu a elevaci. Dále p, q, r jsou složky vektoru inerciální úhlové rychlosti nosiče (rychlost klopení, klonění, zatačení), ostatní veličiny byly uvedeny u rovnic (4.8) a (4.9). . . . .	21
4.6	Kaskádní regulace motoru. . . . .	22
4.7	Principiální schéma režimu LOS. . . . .	23
4.8	Principiální schéma režimu INIT. . . . .	24
4.9	Principiální schéma režimu JOY. . . . .	24
4.10	Ukázka nastavených regulátorů v režimu JOY. . . . .	24
4.11	Principiální schéma režimu IRC. . . . .	25
4.12	Ukázka nastavených regulátorů v režimu IRC. . . . .	25
5.1	Ukázka možností verzovacího systému GIT. . . . .	32
5.2	Stahování aktuální verze z GITu. . . . .	33
5.3	Nahrávání verze na GIT. . . . .	33
5.4	Přepínání mezi větvemi, vytvoření nové větve, spojení větví. . . . .	34
6.1	Struktura nového GUI - pro ilustraci . . . . .	36
6.2	Základní obrazovka GUI. . . . .	38
6.3	Vizualizační model stabilizované základny. . . . .	39
6.4	Indikace přetížení motorů, nastavení grafu, nahrávání naměřených dat. . . . .	39
6.5	Reference joysticku, nastavení kloubových souřadnic či offsetu gyroskopů. . . . .	40
6.6	Nastavení parametrů regulátorů, filtrů a konstant. . . . .	41
7.1	Přípevněná inerciální jednotka k pevné části. . . . .	43
7.2	Dosažené výsledky při stabilizaci elevační osy. . . . .	44
7.3	Dosažené výsledky při stabilizaci příčné elevační osy (při použití pravouhého souřadného systému byly kamery vykloněny o $-60^\circ$ od příčné osy nosiče). . . . .	45
A.1	Pracovní plocha. . . . .	I
A.2	Stabilizovaná základna uchycená k podstavci. . . . .	II
A.3	Detail stabilizované základny. . . . .	II

# Kapitola 1

## Úvod

### 1.1 Historie projektu

Tato práce je součástí většího projektu či spíše série projektů, které začaly v roce 2006, kdy dvě pracoviště z ČVUT FEL, katedra řídicí techniky a katedra kybernetiky, začaly spolupracovat s VTÚLaPVO Praha (dále jen VTÚL - Výzkumný technický ústav letectva a protivzdušné obrany) a firmou ESSA Praha s.r.o. na vývoji inerciálně stabilizované kamerové hlavice pro pilotované i bezpilotní letouny. V roce 2008 dokončil tým vývojářů prototyp kamerové hlavice pod názvem H240<sup>1</sup>, kterého se týká tato bakalářská práce. Jedná se o hlavici s dvouosým principem a její značení vyjadřuje průměr hlavice 240 mm. Cílem celého systému, který je tvořen vnitřní rychlou inerciální zpětnovazební smyčkou a vnější pomalou vizuální zpětnou vazbou, je sledování pevného či pohyblivého objektu a zároveň potlačování dopadu nežádoucích pohybů letadla na nasměrování optické osy.



Obrázek 1.1: Inerciálně stabilizovaná kamerová hlavice H240.

---

<sup>1</sup>Stabilizovaná kamerová platforma H240, původně Eagle1

## 1.2 Současný stav projektu

V současné době je k dispozici několik prototypových inerciálně stabilizovaných platform vyvinutých a vyvíjených na katedře řídicí techniky a katedře kybernetiky. Fakticky se jedná o typy pod názvem Svíce120<sup>2</sup> a Svíce250<sup>3</sup>, které jsou dále vyvíjeny. Z názvů vyplývá, že se jedná o novou řadu inerciálně stabilizovaných kamerových hlavic oproti původní hlavici H240. V prvním případě se jedná o malou hlavici s průměrem 120 mm, obsahující dvouosý princip s běžnými motory s převodem, a v druhém případě o větší hlavici s průměrem 250 mm, obsahující dva vnitřní a dva vnější závěsy, tedy celkem čtyři, namísto dříve použitých dvou u hlavice H240.



Obrázek 1.2: Stabilizovaná základna Svíce120 vlevo a Svíce250 vpravo

## 1.3 Motivace

Jak bylo již zmíněno inerciálně stabilizovaná kamerová hlavice H240 byla první prototyp a od té doby vznikla nová řada kamerových hlavic. Stabilizovaná základna H240 obsahuje dva přímé momentové motory v azimutální a elevační ose, které dostatečně potlačují kymácivé rušivé pohyby nosiče, ale pro dosažení rychlé odezvy na požadovanou změnu nasměrování optické osy by bylo potřeba mít asi dvakrát nebo třikrát silnější motory, ale to by byla několikanásobně vyšší cenová kategorie. S přihlédnutím k ceně potřebných přímých momentových motorů novější stabilizovaná platforma Svíce250 obsahuje dva vnější závěsy tvořené běžnými motory, které kvůli velkým otáčkám a malému krouticímu momentu potřebují mechanický převod, který zhoršuje možnost hmotové stabilizace. Při

---

<sup>2</sup>Stabilizovaná kamerová platforma S120

<sup>3</sup>Stabilizovaná kamerová platforma S250

hmotové stabilizaci by se optická osa při pohybu nosiče snažila setrvat v klidu a krouticí moment způsobený nosičem by byl nulový, ale to pouze za předpokladu žádného tření, které ale u mechanického převodu není možné. Funkčnost hmotové stabilizace je zde navržena použitím dalších dvou vnitřních závěsů, i když jen s omezeným úhlovým rozsahem, což způsobí, že stabilizovaná hmota drží svou původní orientaci i při vypnutém elektrickém napětí na motorech. Přesto všechno, i když se hlavice H240 dál nevyvíjí, by byla škoda nevyužít její skrytý potenciál pro různé výzkumné a výukové aktivity. Proto je vhodné provést aktualizaci zdrojových kódů stabilizované platformy, souvisejících aplikací a využít tak možnost sjednotit hlavici H240 s ostatními vyvíjenými hlavicemi. Toto vše přináší neméně zajímavé příležitosti pro zapojení dalších členů, kteří by se výhledově mohli podílet i na aktuálních projektech.

## 1.4 Cíl bakalářské práce

Cílem bakalářské práce je demonstrovat zvládnutí vývojového procesu a provést nezbytné úpravy v existujícím softwaru pro stabilizovanou základnu pro kamerový systém a dalších uživatelských aplikacích, aby byly využity veškeré schopnosti stabilizované základny, které nabízí.

- Nezbytnou součástí této bakalářské práce je proniknutí do koncepce existujících stabilizovaných základen, jak z pohledu mechaniky a hardwaru, ale i softwaru.
- Rovněž je nutné porozumět tvorbě kódů pro inerciálně stabilizovanou kamerovou hlavici H240 v jazyce C pod systémem Linux, jejich způsobu kompilace pomocí GCC<sup>4</sup> kompilátoru a nahrávání kódů do cílového procesoru. Následně po vytvoření, zkompilování a nahrání kódů do stabilizované platformy, je možné spustit samotnou regulaci pomocí nově navržených PID regulátorů s možností změny jednotlivých parametrů téměř v reálném čase pomocí grafického uživatelského rozhraní.
- Samozřejmostí je, že v dalších kapitolách se dočtete o modifikacích regulátorů, o kaskádním řízení, ale také o jejich faktickém použití v různých režimech, jako je například stabilizace optické osy (LOS), řízení joystickem (JOY), nastavení kloubových souřadnic motorů (IRC) či režim kalibrace offsetů gyroskopů (CALIB).

---

<sup>4</sup>GNU Compiler Collection - sada překladačů

- Dalším z cílů je převedení celého vývojového procesu ze systému Linux na Windows s vývojovým prostředím Eclipse. Hlavním důvodem je, že veškeré aplikace vytvořené pro tuto platformu jsou udělané pod systém Windows, ať už se jedná o ovládací systém neboli operátor či grafické uživatelské rozhraní, a proto byl zvolen tento postup, aby bylo možné vše programovat, řídit a ovládat pomocí jednoho počítače s jedním operačním systémem.
- Jedním z bodů bakalářské práce je také u stabilizované platformy s přímými momentovými motory nahradit výše již několikrát zmíněné původní grafické uživatelské rozhraní (GUI) novým. Podnětem k této změně bylo přání, aby se veškeré existující inerciálně stabilizované platformy, nové i staré, daly ovládat a nastavovat pomocí stejného GUI.
- S postupem přibývajících verzí kódů vznikla potřeba jejich přehledného spravování, za tímto účelem jsem začal využívat systém GIT<sup>5</sup>, který byl na katedře řídicí techniky v mé pracovní skupině AA4CC používán. Posloužil k přehlednosti verzí kódů a též k jejich zálohování.

## 1.5 Shrnutí výsledků

Výsledkem bakalářské práce jsou funkční kódy k inerciálně stabilizované základně H240 pro kamerový systém. Zveřejněna v této práci je však pouze jejich HTML dokumentace, která je přiložena na CD. V dokumentaci je stručně popsán projekt, funkce řídicích desek a programy, které jsou zveřejněny pouze formou hlavičkových souborů. Zdrojové kódy řídicích desek byly mnou vytvořeny, ale vycházely z původních programů, a nyní mají podobnou strukturu jako stávající nejnověji vyvíjené platformy.

Dále je popsáno používané, mnou a kolegy upravované, grafické uživatelské prostředí, které slouží k ovládání a odladování programů nových inerciálně stabilizovaných platform a nyní i hlavice H240.

Aby práce posloužila jako ucelený blok informací týkající se kamerové hlavice, je zde uveden i stručný popis mechanické konstrukce a elektroniky.

---

<sup>5</sup>GIT je distribuovaný systém správy verzí



# Kapitola 2

## Mechanické uspořádání

Kapitola popisuje různé mechanické konstrukce stabilizované základny, upozorňuje na jejich výhody a nevýhody. Dále rozebírá dvě v praxi nejrozšířenější mechanická uspořádání a na závěr pojednává o realizované konstrukci a její terminologii.

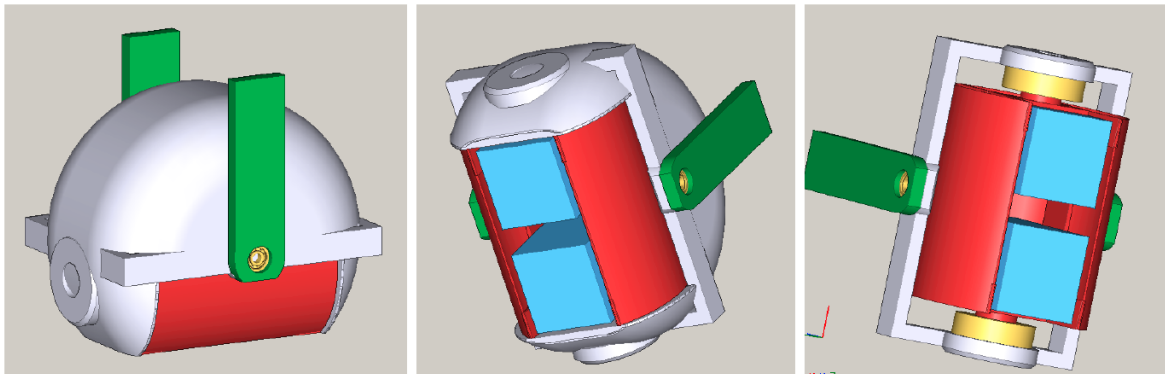
### 2.1 Konstrukce

Při úvaze o mechanické konstrukci stabilizované základny bylo zadavatelem projektu a jeho řešiteli vybráno takové řešení, které uvažuje pohyb celého kamerového systému, kterým dochází k potlačení nežádoucích pohybů nosiče na nasměrování optické osy, na kterém je stabilizovaná základna umístěna. Dané řešení sebou přináší své výhody, ale i nevýhody. Jednou z nevýhod může být například řízení pohybu větší hmotnosti a naopak výhodou je snadná konstrukce oproti kamerovému systému s pohyblivým odrazným prvkem, který by směřoval obraz. Jestliže rozšíříme naši úvahu o informaci, že uvažujeme variantu pouze se dvěma stupni volnosti, tak se nám tu nabízí dvě v praxi nejrozšířenější konfigurace týkající se umístění kloubů základny. Asi nejvíce komerčně dodávané platformy mají konfiguraci El-Az, která je z hlediska mechanického návrhu nejjednodušší. V případě umístění stabilizované základny na letounu, může být jeden motor zapuštěn v trupu letadla. Pro vojenské použití, které bylo původně uvažováno jako dominantní, je hlavním požadavkem pozorování pozemních objektů na velké vzdálenosti od letounu, se kterým není problém. Ale při civilním použití (policie, záchranáři atd.) se předpokládá sledování spíše objektů na kratší vzdálenost s možností sledování scény přímo pod stabilizovanou základnou, kde ovšem vzniká problém se singularitou. Druhou variantou může

být konfigurace Tip-Tilt, která má však mnohem větší mechanickou konstrukci.

### 2.1.1 Tip-Tilt

Konstrukce Tip-Tilt představuje otáčení vnějšího a vnitřního závěsu okolo dvou na sebe kolmých vodorovných os.

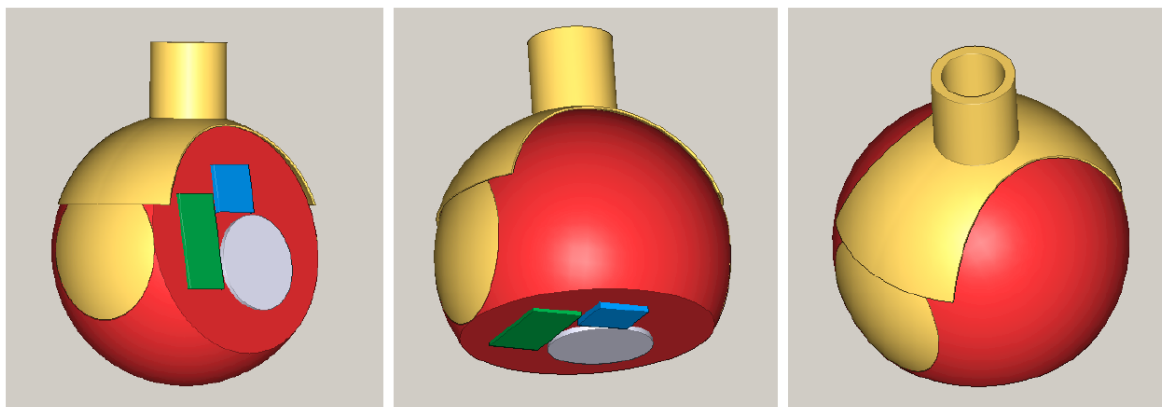


Obrázek 2.1: Konstrukce typu Tip-Tilt, převzato z [7].

Nevýhodou této varianty je větší a složitější konstrukce, při požadavku stejných rozsahů pohledů jako u následující varianty El-Az (rotace azimutálního kloubu o  $n \times 360^\circ$  a naklopení elevačního kloubu o úhel až  $110^\circ$ ). Pokud budeme chtít sledovat scénu, která odpovídá rovině letadla, nastane zde problém tzv. singulárního bodu, jelikož osa obrazu bude shodná s řízenou osou a tím dojde ke ztrátě jednoho stupně volnosti. Naopak výhodou Tip-Tilt konstrukce je neomezené sledování scény přímo pod stabilizovanou základnou.

### 2.1.2 El-Az

U konstrukce El-Az se jedná o otáčení vnějšího závěsu okolo svislé osy a vnitřního závěsu okolo vodorovné osy. Daná konstrukce má řadu výhod, například velký rozsah úhlů, menší konstrukční rozměry a při uvažovaném kroužení letadla nad cílovou pozicí nedojde k rotaci obrazu. Velkou nevýhodou se však stává sledování scény přímo pod stabilizovanou základnou, jelikož je zde shodná osa obrazu s řízenou osou azimutálního kloubu, opět tedy dojde ke ztrátě jednoho stupně volnosti.



Obrázek 2.2: Konstrukce typu El-Az, převzato z [7].

Stabilizace obrazu v tomto singulárním bodě a jeho okolí je znemožněna, protože i pro malý pohyb obrazu vzniká potřeba velkého krouticího momentu. Řešením pro eliminaci singulárního bodu se zde nabízí použití dalšího stupně volnosti, který by umožnil i stabilizaci obrazu přímo pod sebou. Této možnosti je právě využito u novější inerciálně stabilizované kamerové platformy Svice250, která byla zmíněna v kapitole 1. Dalším možným řešením by bylo umístění inerciálně stabilizované platformy ve směru osy letadla, singulární bod by se přesunul, odpadl by problém se sledováním scény pod letadlem, ale vznikl by stejný problém, jako u konstrukce Tip-Tilt.

Zadavatelem LOM s.p., o.z. VTÚLaPVO Praha byla vybrána právě mechanická konstrukce El-Az, která byla zhotovena konstruktérem Ing. Milanem Bartošem z firmy ESSA Praha s.r.o.

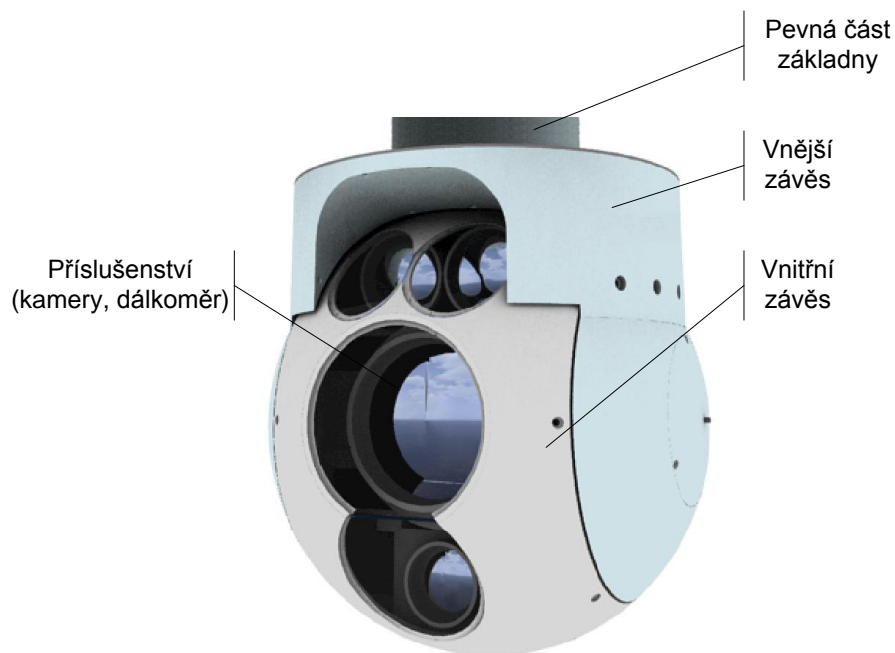


Obrázek 2.3: Realizovaná konstrukce, převzato z [7].

## 2.2 Terminologie

Pro ujasnění veškeré používané terminologie je zde uveden přehled nejpoužívanějších pojmů týkajících se mechanické konstrukce.

- Stabilizovaná základna (hlavice, základna, platforma) - celková konstrukce s veškerým vybavením.
- Pevná část základny - část základny pevně upevněné k nosiči (letadlu, autu).
- Vnější závěs (vnější část) - část základny, která se otáčí okolo azimutální osy oproti pevné části základny.
- Vnitřní závěs (vnitřní část) - část základny, která se otáčí okolo elevační osy oproti vnější závěsu.



Obrázek 2.4: Terminologie - části stabilizované základny, převzato z [3] a upraveno.

# Kapitola 3

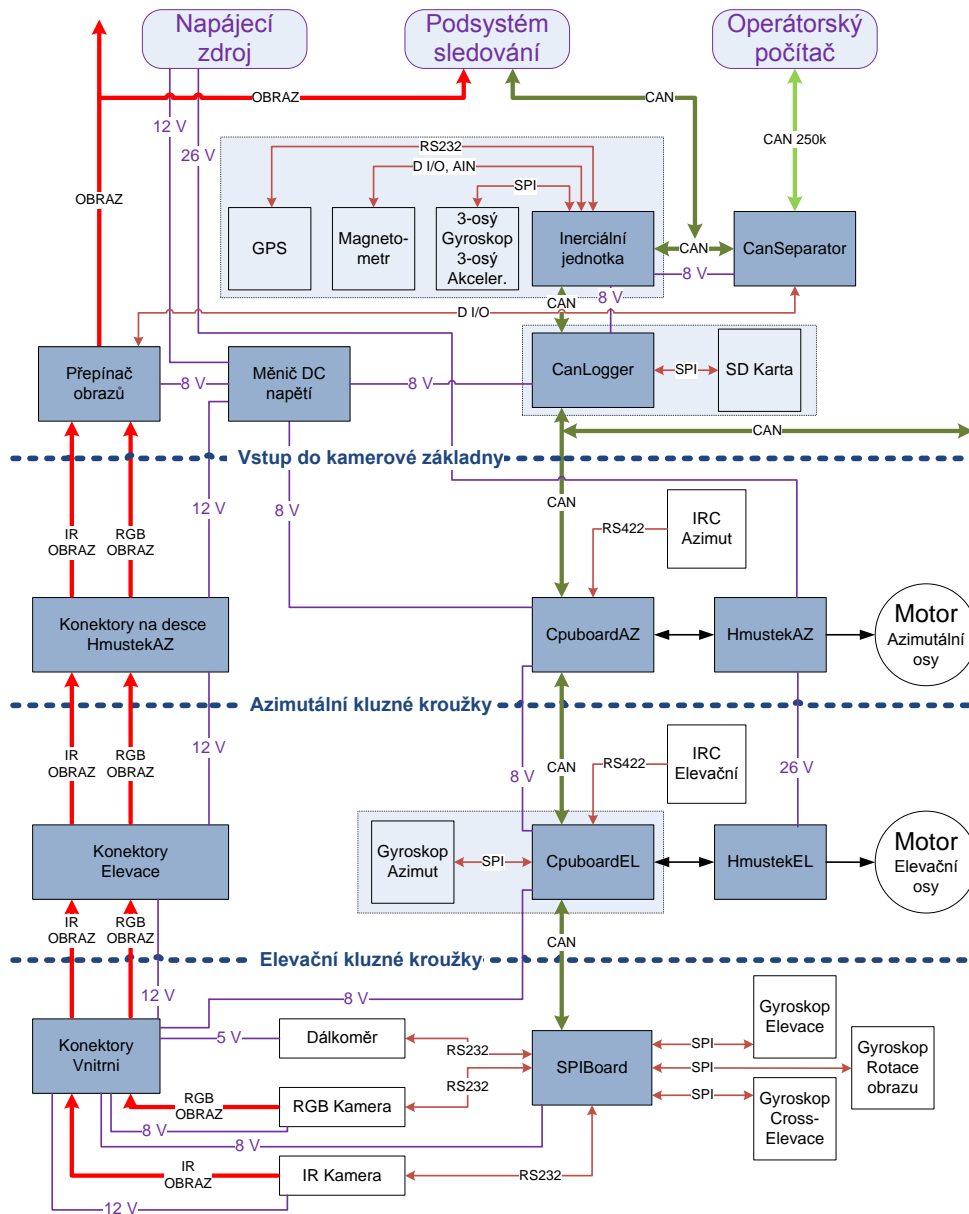
## Elektronika

Kapitola seznamuje s blokovým schématem zapojení elektroniky inerciálně stabilizované základny a přibližuje použití jednotlivých elektronických zařízení a součástek. Podrobnější informace týkající se elektroniky naleznete v [3, 7]. Cílem této kapitoly je, aby budoucí uživatel či pokračující student v práci na stabilizované základně, měl základní přehled o použité elektronice, která je neméně důležitá jako mechanická konstrukce. Veškerý návrh elektroniky byl navrhnout a zrealizován Ing. Jaroslavem Žohou a Ing. Martinem Řezáčem.

### 3.1 Použitá koncepce

V této podkapitole si ukážeme, jak vypadá celková koncepce elektroniky stabilizované platformy. Veškeré zde uvedené informace slouží k vytvoření si představy o provázanosti celé navržené struktury systému.

Na obrázku 3.1 vidíme celkovou elektronickou koncepci rozdělenou do čtyř částí. Díky zvolené mechanické konstrukci bylo potřeba zajistit průchod elektrických signálů skrze osy otáčení, toho bylo docíleno pomocí kluzných kroužků, které nám dovolují propojení 24 vodičů se špičkovým zatížením maximálně 2 A. Dále na zmiňovaném obrázku vidíme propojení jednotlivých bloků pomocí sběrnic, signálů a také distribuci napětí.



Obrázek 3.1: Blokové schéma zapojení elektroniky, převzato z [4].

Vnitřní závěs obsluhuje řídicí deska SPIBoard, která má za úkol měřit úhlové rychlosti pomocí tří jednoosých MEMS gyroskopů (gyroskop elevace, příčné elevace, rotace obrazu) a ovládat laserový dálkoměr, optickou a infračervenou kameru.

Vnější závěs je tvořen řídicí deskou s názvem CpuBoardEL, která má na starosti zjišťování absolutní polohy elevačního motoru pomocí inkrementálního senzoru a řízení napětí na motoru v elevační ose, ale též měří úhlovou rychlost jednoosým gyroskopem

(gyroskop azimut).

Pevná část, která je řízena deskou pod názvem CpuBoardAZ, zajišťuje vyčítání absolutní polohy azimutálního motoru a řízení napětí na motoru v azimutální ose.

Čtvrtá část (nosič - letadlo, auto) musí být vybavena napájecím zdrojem, nadřazeným systémem zpracovávajícím obraz, inerciální jednotkou a operátorským počítačem. Dále je mezi pevnou částí a nosičem umístěn CanSeparator, který slouží k oddělení zpráv a úpravě rychlostí přenosu vnitřního a vnějšího CANu, jelikož vnitřní CAN používá rychlost 1 Mbit/s, ale vnější CAN, který slouží k ovládní hlavičky operátorským počítačem, pouze 250 kbit/s. Obsahuje také CANLogger, který umožňuje ukládání veškeré komunikace po sběrnici CAN ve formě textových souborů na SD kartu.

## 3.2 Sběrnice, hardware, senzory

### 3.2.1 Sběrnice

Jak je vidět na obrázku 3.1, používá se zde hned několik typů sběrnic a proto si je popíšeme a vysvětlíme k čemu se používají.

- Sběrnice RS232 a RS422 - klasická sériová linka. Používá se pro programování procesorů, pro komunikaci s dálkoměrem, kamerami a pro zjišťování absolutní polohy inkrementálního čidla.
- Sběrnice CAN (Controller Area Network) - jde o průmyslovou sběrnici realizovanou jedinou datovou linkou, na které se rozlišují dvě logické hodnoty (dominantní a recesivní úroveň). Zařízení se připojují do sběrnice topologie. Sběrnice CAN je použita k veškeré vnitřní komunikaci mezi řídicími deskami, ale i k vnější komunikaci s operátorem.
- Sběrnice SPI (Serial Peripheral Interface) - jedná se o sériové periferní plně duplexní rozhraní řízené systémem master/slave. Pro komunikaci používá čtyři vodiče (SCK - hodinový signál ze zařízení master, MISO - datový signál do zařízení master, MOSI - datový signál do zařízení slave, CS - výběrový signál sloužící k výběru zařízení slave). Ve stabilizované základně slouží ke komunikaci mezi řídicí deskou a gyroskopy.

### 3.2.2 Hardware a senzory

- RGB kamera (SONY FCB-IX) - analogový kompozitní výstup obrazu



Obrázek 3.2: RGB kamera, převzato z [3].

- IR kamera (Miricle 110K) - analogový výstup obrazu infračervené kamery



Obrázek 3.3: IR kamera, převzato z [3].

- Laserový dálkoměr (Vectronix LRF42) - měří vzdálenost od cíle.



Obrázek 3.4: Laserový dálkoměr, převzato z [3].

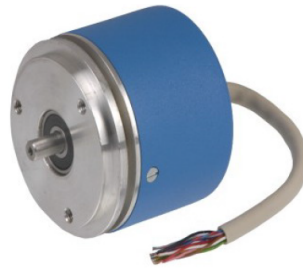


- Kluzné kroužky (od firmy MOOG) - slouží k převedení elektrických signálů v osách otáčení.



Obrázek 3.5: Kluzný kroužek, převzato z [3].

- Inkrementální senzory (ESSA) - se používají k určení absolutní polohy natočení os základny.



Obrázek 3.6: Inkrementální senzor, převzato z [3].

- Jednoosé gyroskopy (ADIS16255) - slouží k měření úhlové rychlosti rotace obrazu, elevace, příčné elevace a azimutu.
- Stejnoseměrné komutátorové přímé momentové motory (D-2910-G) - obstarávají natočení řízených os, elevační a azimutální.



Obrázek 3.7: Přímý momentový motor - pro ilustraci, převzato z [3].

- H-můstky (L6201PS) - jsou určeny k řízení přímých momentových motorů pomocí pulsní šířkové modulace.
- Procesory (LPC2119) - jsou umístěné na všech řídicích deskách a zajišťují veškerý výpočetní výkon a řízení.
- Programovatelné obvody (CPLD Xilinx XC9572XL) - jsou využity ke zpracování signálu z inkrementálních senzorů s velkým počtem pulsů na jednu otáčku.
- Stabilizátory napětí, operační zesilovače, oscilátory atd.

# Kapitola 4

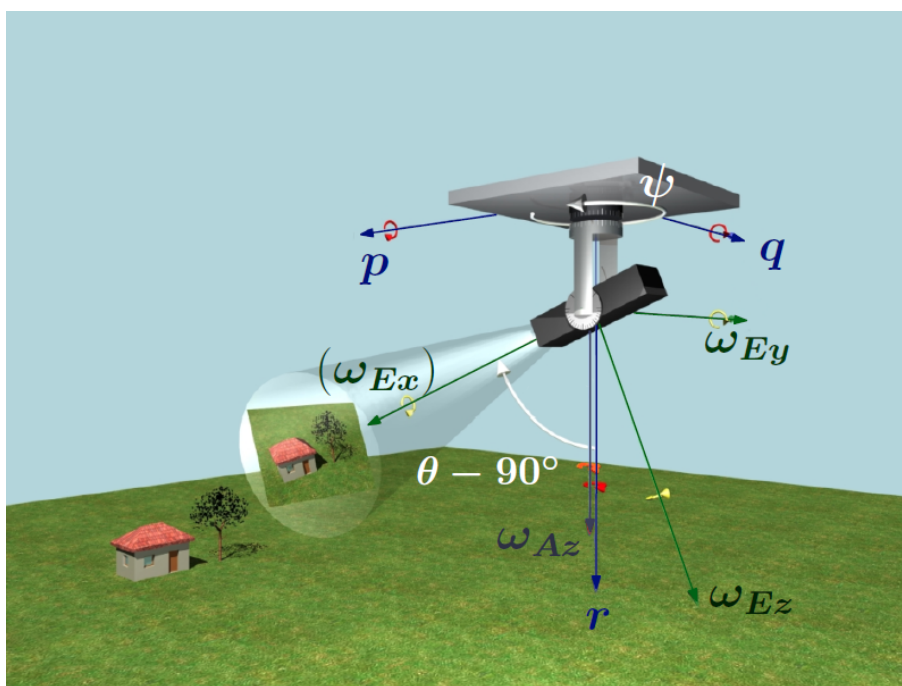
## Řídicí systém pro inerciální stabilizaci

Kapitola řídicí systém pro inerciální stabilizaci Vás provede možnostmi stabilizace obrazu, problematikou singularity, způsobem řešení nových PID regulátorů a kaskádní regulací. Bude se také zabývat dynamikou závěsu a motoru. A na konci této kapitoly si představíme množství režimů řízení, ve kterých může stabilizovaná základna pracovat.

### 4.1 Stabilizace obrazu

Při požadavku na pevné sledování zvoleného objektu je nutné provádět tzv. stabilizaci obrazu, aby se objekt nacházel uprostřed zorného pole neboli ve středu obrazu z kamery, což reprezentuje stabilizace optické osy (angl. Line of sight stabilizace, zkráceně LOS). Ve stabilizované základně je použita metoda přímé LOS stabilizace, podrobněji v [6], která znamená, že senzory úhlové rychlosti jsou umístěny ve vnitřním závěsu a přímo měří pohyb kamery ve třech osách (elevace, příčné elevace, rotace obrazu). Ovšem stabilizovaná platforma má pouze dva stupně volnosti a proto není možné stabilizovat všechny tři osy, v našem případě tedy neprovádíme stabilizaci rotace obrazu. Při stabilizaci obrazu opět narážíme na problém s polohou singulárního bodu, který je způsoben mechanickou konstrukcí, podrobnosti v kapitole 2.1. Částečným řešením tohoto nedostatku je přepínání mezi dvěma režimy v závislosti na blízkosti singulárního bodu. Jestliže se pozorovaný objekt nevyskytuje v blízkosti singulárního bodu, můžeme použít tzv. *stabilizaci optické osy*, která spočívá v měření úhlové rychlosti v ose elevace, příčné elevace a jejich

následné regulace na nulovou hodnotu. Druhým režimem je tzv. *stabilizace elevace-azimut*, která dovoluje stabilizaci obrazu v okolí singulárního bodu, jelikož stabilizujeme osy elevace a azimutu. Dochází zde k menší rotaci obrazu, ale nedochází k úplně přesnému zamíření na daný objekt. Výběr pozemního objektu určeného ke sledování je možné provést manuálně ve snímaném obraze scény nebo zadáním GPS souřadnic. Následné sledování zajišťují algoritmy počítačového rozpoznávání obrazu, vyvinuté na katedře kybernetiky. Úloha zadávání referenčního směru pohledu kamery pomocí GPS souřadnic bez použití dálkoměru je v současnosti řešena jinými členy týmu.



Obrázek 4.1: Souřadné soustavy v základně, převzato z článku [2].

Na obrázku 4.1 vidíme vyznačené úhly ( $\psi, \theta$  je úhel v azimutu a elevaci), inerciální úhlové rychlosti ( $\omega_{Ex}, \omega_{Ey}, \omega_{Ez}$  a  $\omega_{Az}$  jsou úhlové rychlosti rotace obrazu, elevace, příčné elevace a azimutu) a složky vektoru inerciální úhlové rychlosti nosiče ( $p, q, r$  jsou rychlost klopení, klonění, zatáčení).

## 4.2 Diskrétní PID regulátory

Podkapitola diskrétní PID regulátory čerpá z knihy [1]. Většina procesů je v dnešní době řízena pomocí PID regulátorů, které mohou být dány například tímto tvarem rovnice:

$$u(t) = K \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau - T_d \frac{dy(t)}{dt} \right) \quad (4.1)$$

kde chyba  $e$  je rozdíl mezi referenčním signálem a výstupním signálem  $y$ . Akční zásah je označen písmenem  $u$ .  $K$  je proporcionální zesílení regulátoru,  $T_i$  integrační časová konstanta a  $T_d$  derivační časová konstanta.

Regulátor s čistou derivační složkou je fyzikálně nerealizovatelný. Jestliže na vstup regulátoru bude v čase nula působit skoková změna  $e(t)$ , na výstupu se v tomto okamžiku objeví Diracova funkce (derivace skokové změny je v okamžiku skoku nekonečná), která je samozřejmě fyzikálně nerealizovatelná. A proto čistá derivační složka nemůže být implementována, protože by mohlo dojít k velkému zesílení měřeného šumu. Řešením je aproximace derivační složky, která po Laplaceově transformaci vypadá takto.

$$sT_d \approx \frac{sT_d}{1 + \frac{sT_d}{N}} \quad (4.2)$$

Přenosová funkce se tak na nízkých frekvencích přibližně rovná a na vysokých frekvencích je omezena do  $N$ , které je v intervalu od 3 do 20.

Použijeme Laplaceovu transformaci, kde  $R$  je referenční signál a  $Y$  výstupní signál.

$$U(s) = K \left( R(s) - Y(s) + \frac{1}{sT_i} (R(s) - Y(s)) - \frac{sT_d}{1 + \frac{sT_d}{N}} Y(s) \right) \quad (4.3)$$

Po provedení diskretizace metodou Tustin dostaneme:

$$p_{kh} = K e_{kh} \quad (4.4)$$

$$i_{kh+h} = i_{kh} + \frac{Kh}{T_i} e_{kh} \quad (4.5)$$

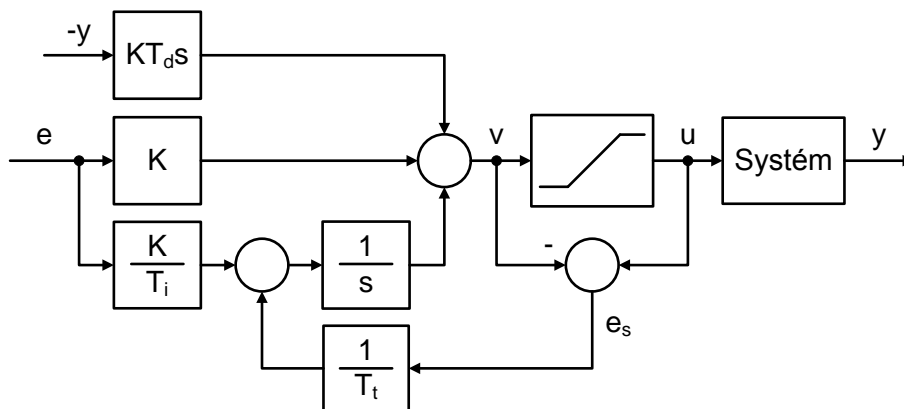
$$d_{kh} = \frac{T_d}{T_d + Nh} d_{kh-h} - \frac{KT_d N}{T_d + Nh} (y_{kh} - y_{kh-h}) \quad (4.6)$$

kde  $h$  je vzorkovací frekvence,  $k$  je krok a parametry  $p$ ,  $i$ ,  $d$  označují proporcionální, integrační a derivační část regulátoru.

Akční zásah je pak roven:

$$u_{kh} = p_{kh} + i_{kh} + d_{kh} \quad (4.7)$$

Spojením regulátoru s integrační složkou a motoru, obsahujícím saturaci, může dojít k nežádoucímu účinku zvanému integrační windup. Příkladem může být motor, který se dostane do saturace v kladných otáčkách, v tomto okamžiku dojde v průběhu času k velkému nárůstu integrační složky. Když ovšem po určitém čase změním referenci tak, aby se motor otáčel v záporných otáčkách, motor se bude kvůli velké integrační složce stále otáčet ještě nějaký čas v kladných otáčkách. Řešením je použít tzv. anti-windup, jehož citlivost je možné nastavit časovou konstantou  $T_t$ .



Obrázek 4.2: PID regulátor, který obsahuje anti-windup.

Výsledný PID regulátor, který byl použit, má následující parametry  $K$ ,  $T_i$ ,  $T_d$ ,  $T_t$ ,  $N$ ,  $u_{low}$ ,  $u_{high}$ . Parametr  $N$  bývá často nastaven na výchozí hodnotu, například 10. Konstanta  $T_t$  souvisí s parametrem  $T_i$ , v některých aplikacích se musí rovnat a v jiných může být rovna 0,1 až 0,5 krát  $T_i$ . Parametry saturace  $u_{low}$ ,  $u_{high}$  by měli být nastaveny v blízkosti skutečné saturace.

Doporučuji nahlédnout do příložené dokumentace zdrojových kódů na příloženém CD, přesněji na knihovnu `lib_pid.c`, která byla vytvořena Ing. Martinem Řezáčem na základě ukázky kódu v knize [1].

Ukázka přímého nastavování PID regulátoru pomocí nového grafického uživatelského rozhraní, obrázek 4.3, namísto původně napevno nastavených konstant P, I, D přímo v kódu. Stejným způsobem je možné nastavovat i filtry a konstanty, více v kapitole 6.

Obrázek 4.3: Nastavení parametrů regulátorů, filtrů a konstant pomocí nového grafického uživatelského rozhraní.

Další výhodou nových PID regulátorů je například možnost omezení na rychlost změny, kterým jde nastavit strmost akčního zásahu. Přináší nám šanci zvolit si o jak velkou hodnotu může v jednom kroku vzrůst akční zásah oproti minulému. Tento vliv je patrný na obrázku 4.12.

Na závěr této podkapitoly bych rád konstatoval, že se úspěšně podařilo zapracovat nový druh PID regulátoru na místo původního neflexibilního regulátoru.

## 4.3 Dynamika závěsu a motoru

Tabulka 4.1: Parametry motoru D-2910-G-1ENS, viz specifikace [8].

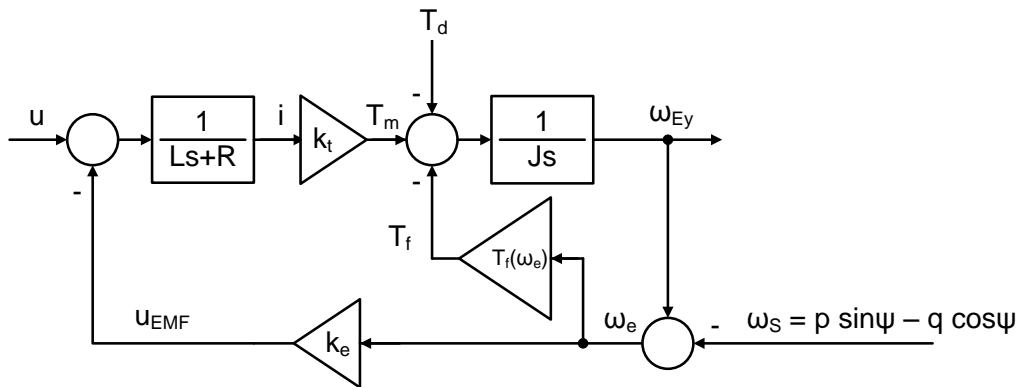
Parametr	Hodnota	Jednotka
Špičkový moment	$T_{\text{peak}}=0,42$	[N m]
Napájecí napětí	$V_p=26,00$	[V]
Špičkový proud	$I_p=1,45$	[A]
Motorová konstanta	$k_t=0,29$	[N m/A]
Konstanta indukovaného napětí	$k_e=0,29$	[V/rad/s]
Odpor vinutí	$R=18,00$	[ $\Omega$ ]
Indukčnost vinutí	$L=6,00 \cdot 10^{-3}$	[H]
Moment setrvačnosti	$J_m=8,66 \cdot 10^{-4}$	[N m <sup>2</sup> ]
Třecí moment	$T_f=0,85 \cdot 10^{-2}$	[N m]

Souřadné soustavy v základně odpovídají obrázku 4.1. Model dynamiky elevačního závěsu, včetně dynamiky motoru, obsahuje mechanickou a elektrickou část, které můžeme popsat následujícími rovnicemi uvedenými v prezentaci [9].

$$\underbrace{(J_m + I_{Ey})}_{\text{celk. mom setr.}} \cdot \dot{\omega}_{Ey}(t) = k_t i(t) - T_f(\omega_{Ey}(t) - \omega_s(t)) - T_d(t) \quad (4.8)$$

$$L \frac{di(t)}{dt} + Ri(t) + k_e \cdot (\omega_{Ey}(t) - \omega_s(t)) = u(t) \quad (4.9)$$

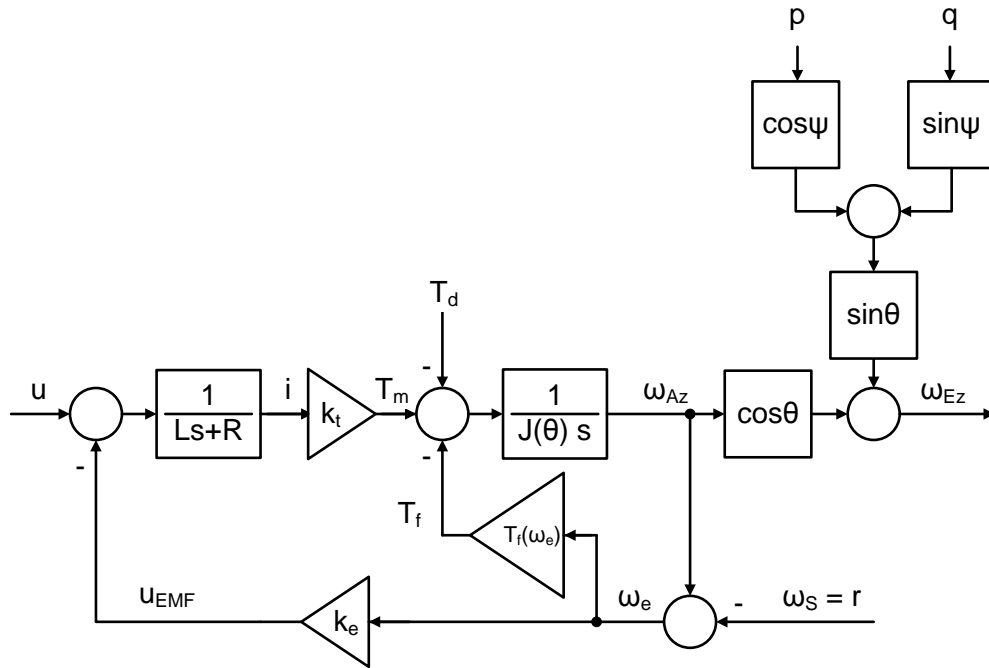
kde  $\omega_{Ey}$  je inerciální úhlová rychlost v elevaci,  $\omega_s$  představuje průmět úhlové rychlosti základny do osy statoru. Dále  $T_f$  je nežádoucí krouticí moment indukovaný třením a  $T_d$  je nežádoucí krouticí moment v důsledku odporu vzduchu.  $I_{Ey}$  je prvek matice momentu setrvačnosti,  $u$ ,  $i$  odpovídají napětí a proudu.



Obrázek 4.4: Model dynamiky v ose elevace, včetně dynamiky motoru, kde výstupem je inerciální úhlová rychlost elevace  $\omega_{Ey}$ . Dále  $T_m$  je užitečný krouticí moment motoru,  $\psi$  je úhel v azimutu a  $p$ ,  $q$ , jsou složky vektoru inerciální úhlové rychlosti nosiče (rychlost klopení, klonění), ostatní veličiny byly uvedeny u rovnic (4.8) a (4.9).

**Poznámka:** Pozor, v trojrozměrném prostoru úhlové rychlosti nejsou derivací úhlových poloh! Podrobnější informace v článku [5]. □

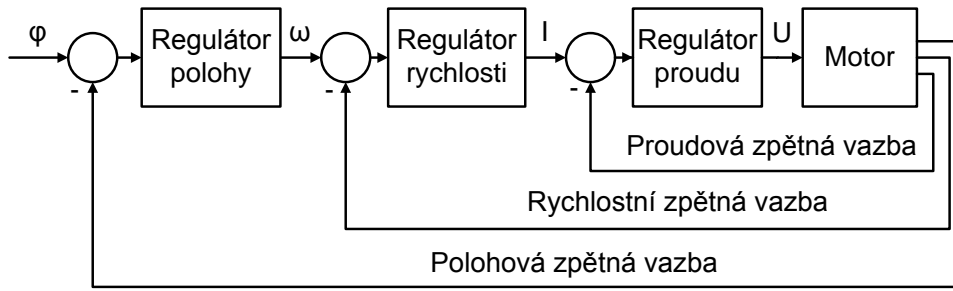




Obrázek 4.5: Model dynamiky v ose příčné elevace, včetně dynamiky motoru, kde  $\omega_{Az}$ ,  $\omega_{Ez}$  jsou inerciální úhlové rychlosti azimutu a příčné elevace,  $T_m$  užitečný kroučící moment motoru,  $\psi$  a  $\theta$  je úhel v azimutu a elevaci. Dále  $p$ ,  $q$ ,  $r$  jsou složky vektoru inerciální úhlové rychlosti nosiče (rychlost klopení, klonění, zatáčení), ostatní veličiny byly uvedeny u rovnic (4.8) a (4.9).

## 4.4 Kaskádní regulace

Kaskádní regulace se velice často používá pro řízení motorů. Obecnou výhodou kaskádního řízení je, že poruchové vlivy, které zasáhnou regulovanou veličinu podřazeného regulátoru, jsou tímto regulátorem kompenzovány dříve než zasáhnou hlavní regulovanou veličinu, ale také nelinearity v podřazené regulační smyčce jsou zvládnuty přímo v ní a nezasahují hlavní regulační smyčce. Další výhodou této regulace je možnost navržení jednotlivých regulátorů zvlášť. Motory bývají řízeny třemi kaskádně řazenými regulátory proudu, rychlosti a polohy dle obrázku 4.6.



Obrázek 4.6: Kaskádní regulace motoru.

Jestliže je v hardwaru zabudované měření proudu motorem, je i velice vhodné regulátor proudu zrealizovat z hlediska proudové a výkonové ochrany motoru. Dále také krouticí moment motoru  $T_m$  je přímo úměrný proudu přes motorovou konstantu  $k_t$ , ale v případě řízení motoru napětím je velikost proudu ovlivněna elektrickou konstantou  $k_e$  a zpožděna indukčností motoru, viz tabulka 4.1 a obrázky 4.4, 4.5.

Rychlostní zpětná vazba je realizována pomocí senzorů úhlové rychlosti. Regulátory rychlosti elevace a azimutu jsou si podobné, ale regulátor v azimutu v režimu stabilizace optické osy obsahuje konstantu  $\frac{1}{\cos(\theta)}$ , kde  $\theta$  je úhel elevačního kloubu, aby došlo k vykrácení nelinearity, podrobněji v článku [5]. S rostoucím úhlem elevace se totiž přibližujeme k singulárnímu bodu a dochází k růstu rychlosti azimutálního kloubu a stabilizovaná základna začíná být nestabilní. Řešením je přepnutí na stabilizaci elevace-azimut popsaném v kapitole 4.1.

Vstupem polohového regulátoru může být posun středu obrazu v pixelech, případně v jiných režimech řízení, kapitola 4.5, absolutní poloha. Pohyb v osách elevace a příčné elevace odpovídá posunu obrazu ve svislé a vodorovné ose. Jedná se tedy pouze o přepočtení vzdálenosti v pixelech aktuálním přiblížením kamery na úhel opravy, který je vynásoben konstantou  $P$  regulátoru. Požadavky na úhlové opravy ze zpracování obrazu se posílají přibližně každých 0,5 až 1 s, a proto vnitřní rychlostní smyčka zaznamenává pohyb kamer pomocí gyroskopů mezi příchozími opravami úhlů. Po příchodu následujících úhlových oprav se k nim přičte údaj o zaznamenaném pohybu kamer a tím se vytvoří aktuální údaj úhlových oprav.

## 4.5 Režimy řízení

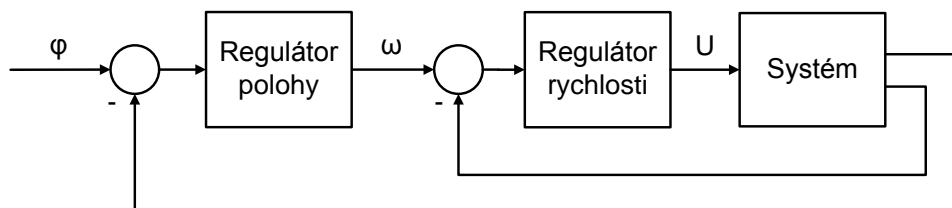
Podkapitola režimy řízení obsahuje seznámení se s jednotlivými módy řízení od blokového schématu zapojení přes ukázkou naladěných regulátorů až k vysvětlení použití a funkcí režimů.

### 4.5.1 Režim stop - STOP

Jestliže se stabilizovaná základna dostane do režimu stop, dojde k okamžitému vypnutí napájení motorů. Což u zde použitých stejnosměrných komutátorových přímo momentových motorů způsobí, že zůstanou uvolněné (volně otočné).

### 4.5.2 Stabilizace optické osy - LOS

Stabilizace optické osy spočívá ve vybrání sledovaného objektu v obraze, který se zobrazuje na operátorské konzoly. Následně pomocí algoritmů zpracování obrazu, vyvinutými na katedře kybernetiky FEL ČVUT, dojde ke sledování zvoleného objektu díky vizuální zpětné vazbě z kamer a rychlostní zpětné vazbě ze sensorů úhlové rychlosti. V případě blízkosti singulárního bodu dojde k přepnutí ze stabilizace optické osy na stabilizaci elevace-azimut. Problematika stabilizace obrazu a řešení kaskádní regulace týkající se režimu LOS jsou více popsány v kapitolách 4.1 a 4.4.

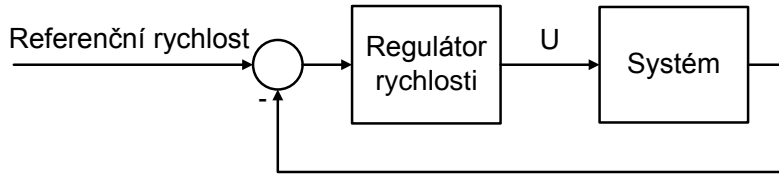


Obrázek 4.7: Principiální schéma režimu LOS.

### 4.5.3 Inicializace - INIT

Inicializace slouží k určení referenčních bodů inkrementálních sensorů. Bez provedení inicializace není provoz stabilizované základny možný, a proto by tento režim měl být prvním krokem při zapnutí hlavice. Spuštění zvoleného módu způsobí pomalé roztáčení

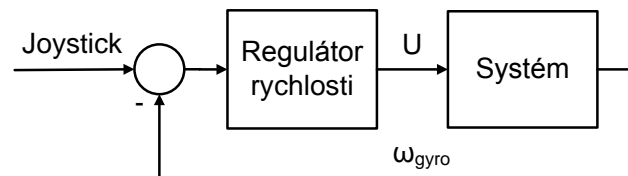
vnitřního a vnějšího závěsu, maximálně na velikost referenční rychlosti, do doby nalezení referenčních bodů a poté přejde stabilizovaná základna do režimu STOP.



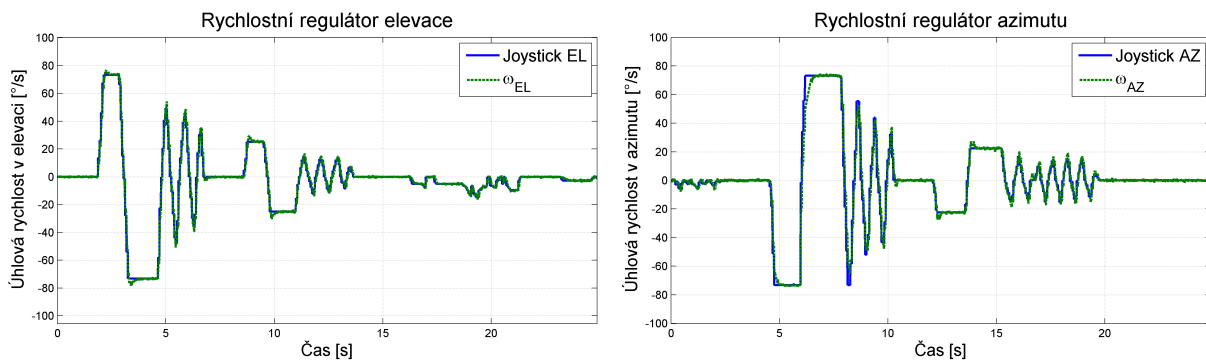
Obrázek 4.8: Principiální schéma režimu INIT.

#### 4.5.4 Ovládání joystickem - JOY

Režim ovládání joystickem je založen na principu sledování žádané reference z joysticku. Zpětná vazba je zde tvořena úhlovými rychlostmi v elevaci a azimutu. Mód je určen k manuálnímu ovládání kamerové hlavičky pomocí joysticku.



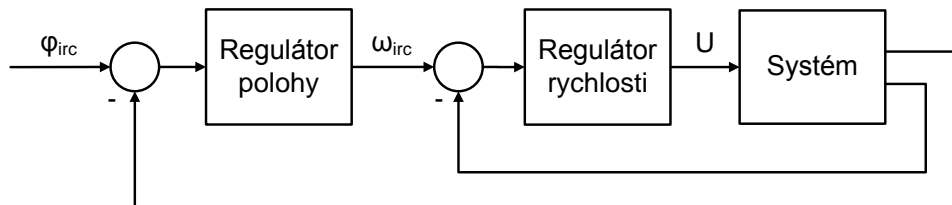
Obrázek 4.9: Principiální schéma režimu JOY.



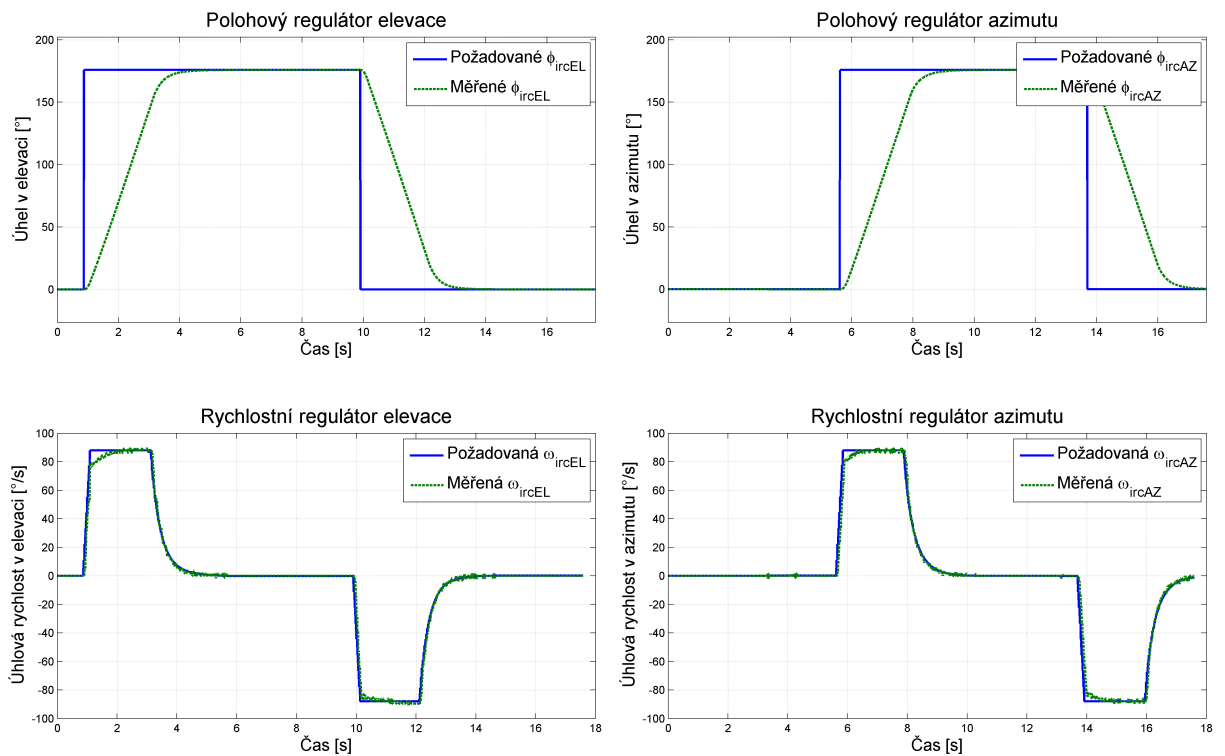
Obrázek 4.10: Ukázka nastavených regulátorů v režimu JOY.

### 4.5.5 Řízení kloubových úhlů - IRC

Stabilizovaná základna má nastavený offset v kloubových souřadnicích, aby poloha absolutní nuly odpovídala vodorovnému pohledu dopředu, za předpokladu zavěšení hlavice svisle dolů. Operátor v daném režimu zvolí hodnotu pro natočení kloubových souřadnic, regulátor polohy vypočítá velikost požadované rychlosti. Tento regulátor navíc obsahuje omezení na rychlost změny, kterým jde nastavit o jak velkou hodnotu může v jednom kroku vzrůst akční zásah oproti minulému, v našem případě požadovaná rychlost, aby nebyl požadavek skokový, ale postupný. Následné nastavení a vypočítání konečného akčního zásahu provede rychlostní regulátor. Výsledkem je natočení hlavice do požadované polohy.



Obrázek 4.11: Principiální schéma režimu IRC.



Obrázek 4.12: Ukázka nastavených regulátorů v režimu IRC.

#### 4.5.6 Kalibrace - CALIB

Režim kalibrace provede tzv. zamčení kloubů elevace a azimutu, to znamená, že zjistí aktuální polohu kloubových souřadnic a na určitý čas tuto polohu drží. V průběhu tohoto času dochází ke kalibraci offsetů gyroskopů. Následně po dokončení kalibrace stabilizovaná platforma přejde do minulého režimu, ve kterém se nacházela před tímto režimem.

# Kapitola 5

## Práce s kódy

Kapitola popisuje vývoj programu od vzniku přes kompilaci až k jeho správě. Vysvětluje provedené změny a implementované novinky v programech řídicích desek stabilizované základny. Informuje uživatele o původním způsobu programování pod systémem linuxového typu, ale také o nově prováděné kompilaci pod systémem Windows s prostředím Eclipse. A na závěr pojednává o způsobu správy kódů.

### 5.1 Tvorba programu

Jedním z prvních úkolů bylo napsat kód v jazyce C pro řídicí desky v elevaci (CpuboardEL) a azimutu (CpuboardAZ). Vycházelo se z již existujících a funkčních kódů, které vznikly na základě projektu a diplomových prací [3, 7]. Později byly upraveny i desky SPI-Board, obsluhující gyroskopy, měřící úhlové rychlosti elevace, příčné elevace a rotace obrazu, a CanSeparator, který odděluje vnitřní a vnější CAN. Nejpodstatnějšími požadavky bylo, aby byl kód přehlednější, aby bylo možné měnit různé parametry za běhu programu pomocí nového grafického uživatelského rozhraní a aby se vývojový proces pod systémem Linux převedl pod Windows.

Dříve byly veškeré konstanty či parametry PID regulátorů a podobně nastaveny na pevně danou hodnotu přímo v programu řídicích desek. Pokud bylo potřeba některý z parametrů změnit, bylo nutné upravit a zkompilovat program, připojit sériovou linku, přepnout procesor do programovatelného režimu, nahrát program, odpojit sériovou linku, přepnout procesor do běžného režimu a následně bylo možné vyzkoušet, zda vše funguje podle našich představ. Je vidět, že tento proces byl velice zdlouhavý. Nyní je

možné díky vytvořené tabulce parametrů, tzv. „constants table“, která je zdefinovaná v `parameters_elevation.h` či `parameters_azimuth.h` a která obsahuje seznam konstant, parametrů PID regulátorů a filtrů příslušných řídicích desek, provádět ukládání dat ze sběrnice CAN do paměti RAM procesoru a odesílání dat na sběrnici CAN po vyžádání z GUI pomocí funkcí vytvořených v knihovně `lib_parameters.c`. S tímto přístupem se ovšem objevil problém, jelikož pokud dojde k resetování procesoru, všechny parametry se vymažou, protože byly uloženy pouze v paměti RAM. Novější stabilizované platformy již mají flash paměť, která umožňuje ukládání dat a jejich obnovení po resetování procesoru, tato bohužel ještě ne. Problém jsem vyřešil ukládáním parametrů do souboru, `nactitabEL.c` či `nactitabAZ.c`, ve formátu jazyka C a jeho přilinkováním společně s ostatními knihovnami při každé kompilaci programu, více o vytváření souboru v související kapitole 6.3 o grafickém uživatelském rozhraní. Následně při opětovném naběhnutí programu, například po resetu procesoru či odpojení napájení, jsou všechny parametry načteny do paměti RAM. Díky takto provedeným změnám je nyní možné měnit za běhu programu jednotlivé parametry do doby, kdy nám budou všechny vyhovovat. Poté provedeme uložení do souboru, přeprogramujeme procesor a od toho okamžiku bude vždy po resetu procesoru použita naposledy zkompilovaná a nahraná konfigurace parametrů.

Použitím nového grafického uživatelského rozhraní, které používají nové základny, vznikla komplikace týkající se komunikace, jelikož komunikační protokol datové sběrnice CAN u použité stabilizované hlavice H240 a novějších Svíce120, Svíce250 nebyly stejné. Ovládání režimů řízení, nastavování kloubových souřadnic, offsetů gyroskopů, autokalibrace atd. odpovídalo starému komunikačnímu protokolu, který již nemohl být použit. Nastudoval jsem odchozí a příchozí zprávy GUI a komunikační protokoly nejnovějších stabilizovaných základen a provedl nezbytné úpravy kódů v hlavici H240, aby bylo vše kompatibilní s novým grafickým uživatelským prostředím a jeho ovládacími prvky.

Pro přehlednost a lepší orientaci v kódu jsem program rozdělil do více menších jednodušších částí, které jsem řešil zvlášť. S tím souvisel i vznik řady knihoven pro UART, CAN, SPI, PWM, časovač, čtení absolutní polohy, filtry, PID regulátory a další, které byly úplně nově vytvořeny nebo vycházely z původních knihoven či kódu. Dále jsem také vytvořil zvláštní adresář, ve kterém jsou uloženy knihovny a soubory, které sdílí řídicí desky.

Implementoval jsem přepočty rozsahů absolutních senzorů polohy a nastavení jejich offsetů pomocí konstant v GUI. Kvůli jiným ovládacím prvkům v novém grafickém uživatelském rozhraní jsem musel upravit manuální nastavování offsetů gyroskopů, ale



i jejich autokalibraci, kterou lze spustit pouze pokud je stabilizovaná základna v klidu.

Mimo jiné jsem změnil vyčítání hodnot z gyroskopu tak, původně řešené pomocí přerušení na sběrnici SPI, aby docházelo ke komunikaci mezi procesorem a gyroskopem i bez čekání na přerušení, které zde bylo zbytečné.

Funkci PID regulátorů a filtrů zajišťují knihovny `lib_pid.c` a `lib_filter.c`, které byly vytvořeny Ing. Martinem Řezáčem podle vzoru v knize [1].

Neměl bych také zapomenout, že jsem v řídicích deskách zavedl signalizaci pomocí LED diod, zda běží program, časovač, sběrnice CAN či zda stabilizovaná základna již projela referenčním bodem.

Nově je možné provést softwarový reset procesoru či použít druhý časovač, například pro nastavení přesné doby zpoždění či měření času.

## 5.2 Kompilace kódu

### 5.2.1 Operační systém linuxového typu

K překladu prvních verzí programů, které byly napsány v jazyce C, byla použita volně dostupná sada překladačů GCC<sup>1</sup> a sestavovací prostředí OMK (OCERA Make System<sup>2</sup> vytvořené a spravované Ing. Pavlem Píšou, Ph.D. a Ing. Michalem Sojkou, Ph.D.).

### 5.2.2 Windows s vývojovým prostředím Eclipse

Jelikož veškeré aplikace pro tuto platformu byly vytvořené pod systém Windows, ať už se jedná o ovládací systém (operátor) či grafické uživatelské rozhraní, bylo rozhodnuto, že se celkový vývojový proces programování řídicích desek převede ze systému Linux na Windows s vývojovým prostředím Eclipse. Jedním z hlavních důvodů bylo, aby bylo možné vše programovat, řídit a ovládat pomocí jednoho počítače s jedním operačním systémem.

Pro programování pod systémem Windows jsem použil program Eclipse pro jazyk C/C++ a Yagarto, jelikož je používané členy týmu AA4CC pro programování nových stabilizovaných základen. Yagarto má výhodu, že je nativně sestaveno pro Windows a také že je vyvinuto pro Eclipse a je s ním tedy plně kompatibilní.

---

<sup>1</sup>GNU Compiler Collection

<sup>2</sup><http://rttime.felk.cvut.cz/omk/>

Yagarto obsahuje nástroje jako například:

- GCC/G++ – C/C++ kompilátor
- AS – assembler kompilátor
- LD – linker
- GDB – debugger

Zpočátku bylo potřeba zajistit správná nastavení v programu Eclipse, aby se ke kompilaci použilo Yagarto. Poté jsem vytvořil nový projekt v programu Eclipse a prostudoval dostupné příklady v Yagarto a na jejich základě napsal ld-skript, startup a makefile pro procesor LPC2119. Následně jsem vytvořil jednoduchý program, zkompileval ho a otestoval. Po úspěchu jsem postupně začal přepisovat, opravovat a modifikovat kód, který jsem do té doby psal pod systémem Linux. Když už jsem měl práci téměř dokončenou, narazil jsem na problémy s přerušováními, jejich povolováními a maskováními. Problém byl způsoben použitím funkce povolování přerušování z příkladů uvedených v Yagarto, řešením se nakonec, po prostudování dokumentace k LPC2119, stalo nastavení příslušných registrů přerušování procesorů ručně.

### 5.2.2.1 Návod pod Windows

Před vytvořením projektu, jeho kompilací a nahráním do procesoru je nutné nainstalovat potřebné aplikace a ovladače a provést jejich nastavení. Přesný postup naleznete v souboru `winarm.pdf` na přiloženém CD.

#### Vytvoření projektu v programu Eclipse:

1. File - New - C Project
2. Makefile Project - Empty Project - Other toolchain - Finish
3. Project - Properties
4. Builders - Odškrtnout „Scanner Configuration Builder“
5. OK

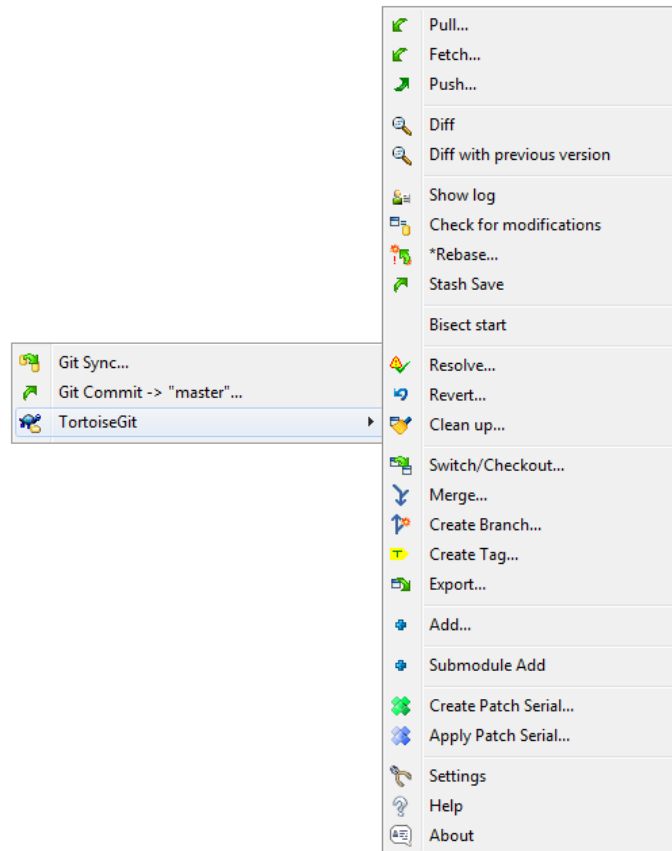
6. Nyní můžeme standardním způsobem psát vlastní kód a nebo podle následujícího postupu provést jeho import
7. File - Import
8. General - Filesystem
9. Import zdrojových kódů - Browse - Zaškrtneme vše, co chceme importovat

#### **Kompilace a nahrání programu do procesoru:**

1. Správně nastavit soubor `Makefile`, doporučuji nahlédnout do příkladů v Yagarto
2. Project - Clean
3. Project - Build Project
4. Označit vytvořený soubor s koncovkou `.hex`
5. Run - External Tools - LPC21ISP Download to Flash
6. Vybrat port - OK
7. Vyčkat na zprávu o úspěšném dokončení nahrávání

## **5.3 Správa kódu**

S postupem času, jak přibývaly nové verze kódů, vznikla potřeba jejich přehledného spravování, ale i zálohování. Za tímto účelem jsem začal využívat systém pro správu verzí GIT, který byl na katedře řídicí techniky v mé pracovní skupině používán. Tento systém pracuje s textovými soubory (zdrojovými kódy programu) a ukládá celou historii jejich vývoje v repozitáři. Mezi hlavní výhody patří možnost vrátit se k předešlým verzím, řešení konfliktu změn provedenými více uživateli ve stejném bloku kódu, dále také umožňuje vývoj v několika nezávislých větvích a jejich vzájemnou synchronizaci či možnost práce s repozitářem bez internetového připojení.

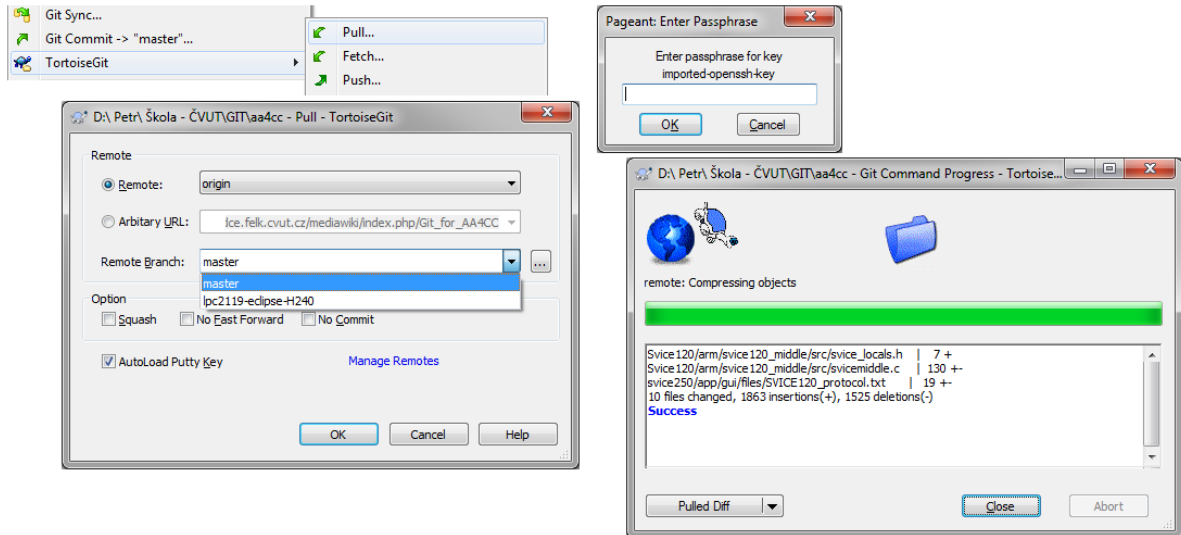


Obrázek 5.1: Ukázka možností verzovacího systému GIT.

### 5.3.1 Návod na GIT

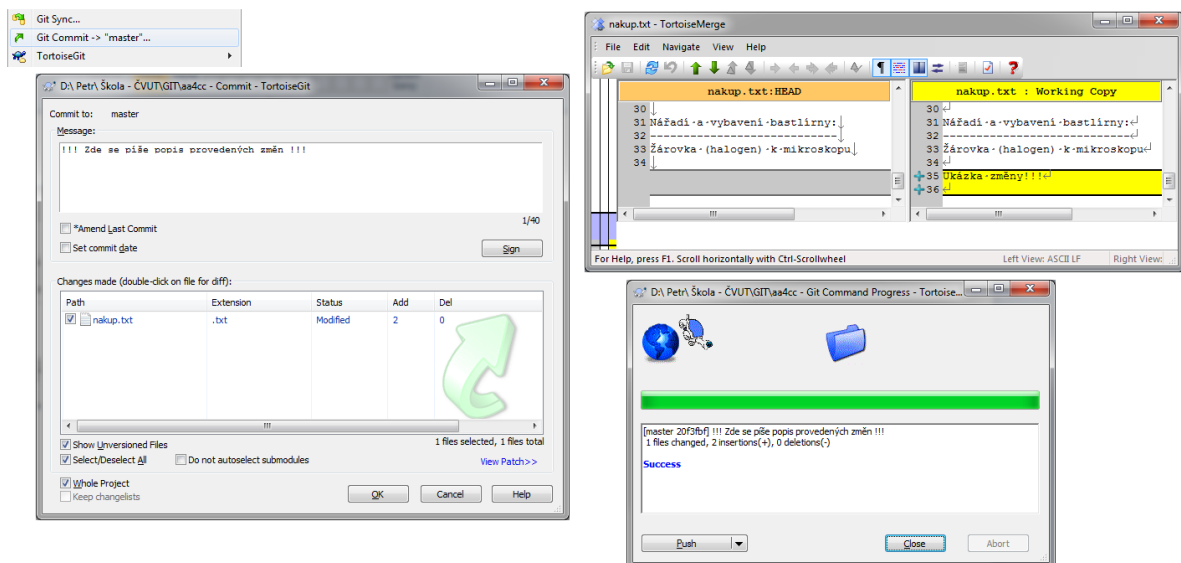
Prvním krokem je stažení a instalace aplikací „GIT client“ a „Tortoise GIT“, podrobnosti o instalaci, nastavení a o vytvoření veřejného a privátního klíče naleznete v souboru `git.pdf` na přiloženém CD. Po úspěšném provedení všech kroků v přiloženém souboru můžeme přistoupit k vysvětlení nejdůležitějších ovládacích prvků „Tortoise GITu“. Přejdeme do adresáře se zkopírovaným repositářem z GITu a podržíme pravé tlačítko myši do doby než se nám zobrazí nabídka jako je na obrázku 5.1.

- **Stahování aktuální verze z GITu:** TortoiseGit - Pull (stáhnout ze serveru) - Remote Branch (výběr větve) - OK - Passphrase (heslo) - OK - CLOSE



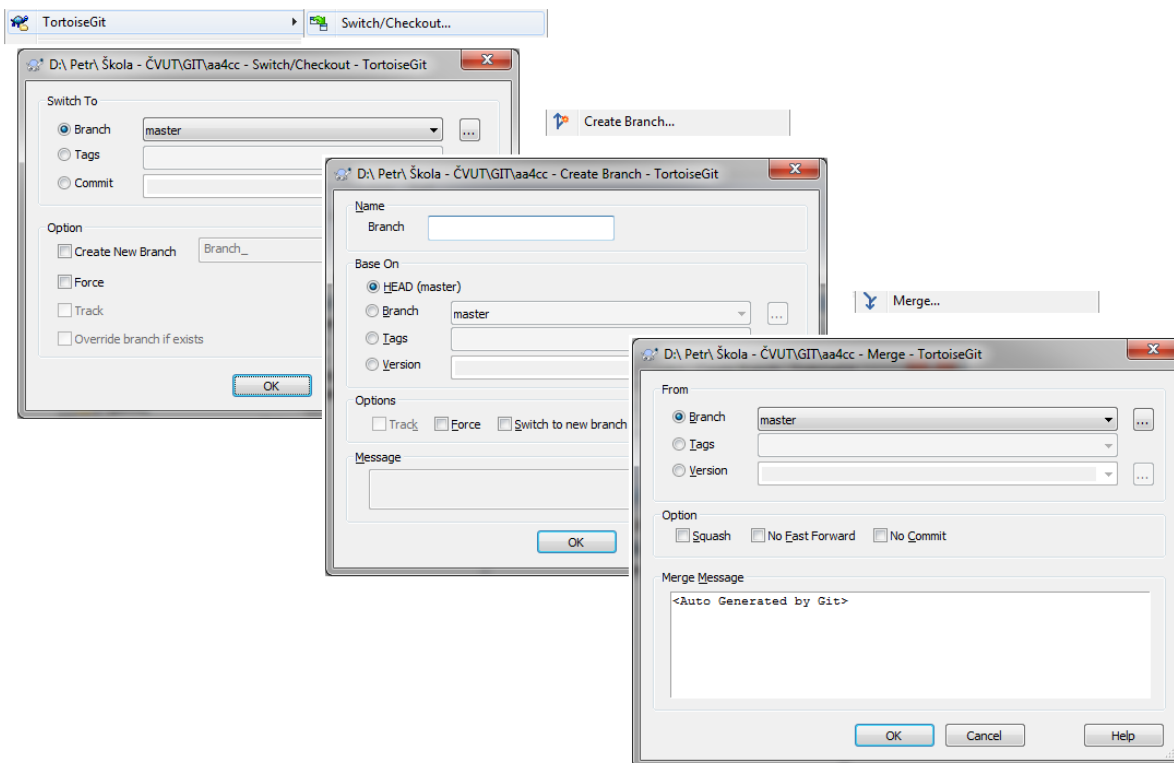
Obrázek 5.2: Stahování aktuální verze z GITu.

- **Nahrávání verze na GIT:** Git Commit - Popis provedených změn a přehled pozměněných souborů - OK - (dvojklikem na souboru se zobrazí rozdíly ve verzích) - Push (nahraj na server) - CLOSE



Obrázek 5.3: Nahrávání verze na GIT.

- Přepínání mezi větvemi, vytvoření nové větve, spojení větví:



Obrázek 5.4: Přepínání mezi větvemi, vytvoření nové větve, spojení větví.

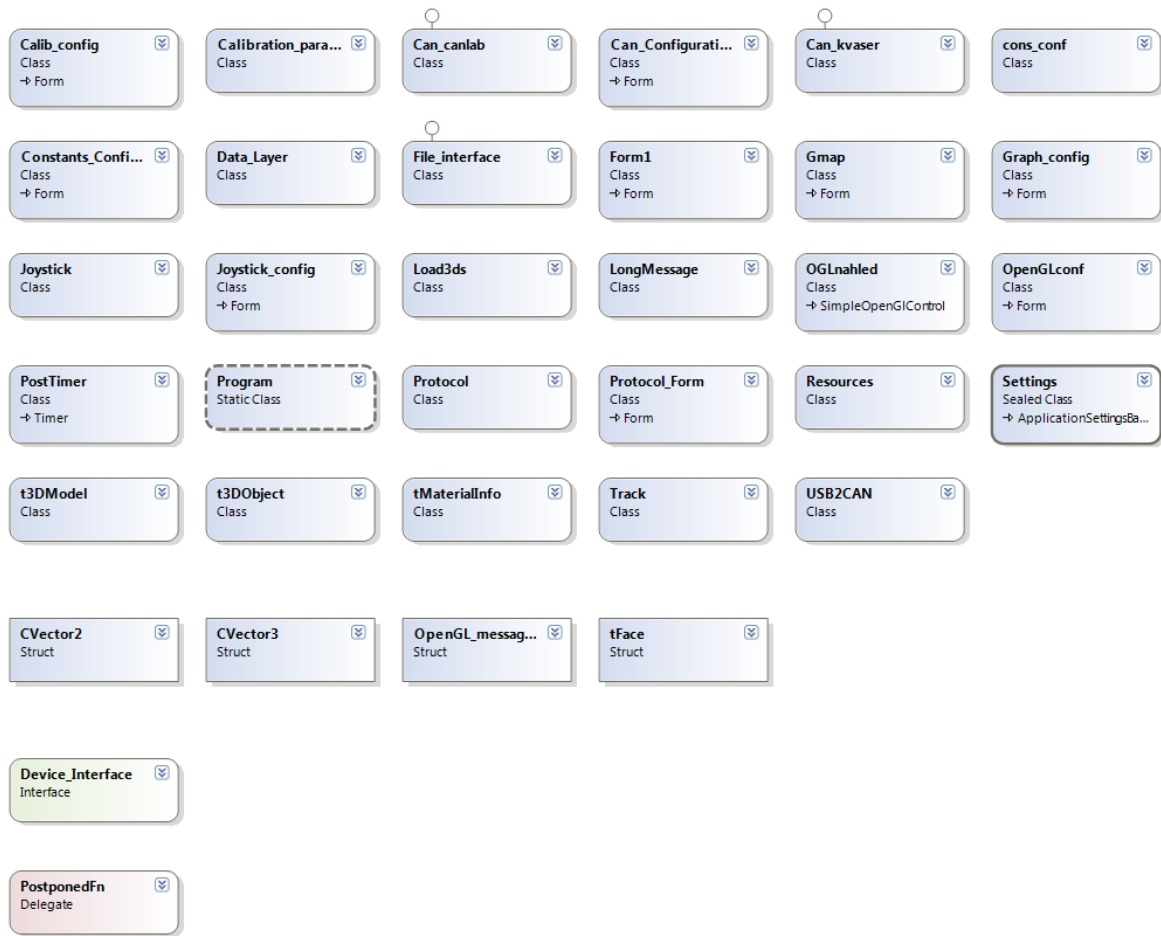
# Kapitola 6

## Grafické uživatelské rozhraní

Cílem této kapitoly je seznámit uživatele s použitým grafickým uživatelským rozhraním (angl. Graphical user interface, zkráceně GUI) a ovládáním stabilizované základny pomocí jeho hlavních komponent. Představuje též dodělané ovládací prvky a také popisuje provedené změny v programech řídicích desek stabilizované platformy souvisejících s novým GUI, některé byly již popsány v kapitole 4.

### 6.1 Úvod do GUI

Na začátek této podkapitoly bych rád uvedl, že existuje původní verze GUI jen pro stabilizovanou základnu H240. Dnes se však pro ovládání všech stabilizovaných platforem používá nové GUI, o kterém se zde dále pojednává. Nové grafické uživatelské prostředí je napsané v jazyce C# a je strukturované do několika tříd, stručný přehled struktury GUI je uveden na obrázku 6.1. Bylo vyvinuto Ing. Martinem Řezáčem a Ing. Jaroslavem Žohou a dále je upravováno celým týmem, pracujícím na stabilizovaných základnách, dle potřeb vyvíjených platforem. Nejaktuálnější verze GUI je k dispozici na GITu, který byl zmíněn v kapitole 5.3.



Obrázek 6.1: Struktura nového GUI - pro ilustraci .

## 6.2 Záměna původního GUI za nové

Původní plán počítal s přepracováním a implementací původního grafického uživatelského rozhraní do nového GUI, které je používáno ve skupině zabývající se problematikou stabilizovaných základen, pomocí nové záložky určené pouze pro stabilizovanou hlavici H240. Toto řešení by bylo velice pracné a v novém GUI by byla jedna část pouze pro stabilizovanou hlavici H240, což by bylo úplně zbytečné. Výhodou tohoto GUI je, že se z něj dají ovládat všechny nově vyvíjené inerciálně stabilizované základny, bez nutnosti vlastních ovládacích prvků pro některou z nich, i další zařízení schopná komunikace po sběrnici CAN. Proto jsem se rozhodl provést pouze několik změn v novém GUI a hlavně uzpůsobit zdrojové kódy hlavice H240 tak, aby byla základna ovladatelná pomocí nového GUI.



Některé změny byly již zmíněny v předešlých kapitolách, ať už se jednalo o nastavování konstant, parametrů PID regulátorů a filtrů, ale i možnost přepínání režimů řízení atd.

## 6.3 Ovládání a možnosti nového GUI

Nyní si ukážeme základní ovládání a možnosti, které přináší nově používané GUI. Též se zaměříme na modifikace, které byly provedeny na straně GUI, ale připomeneme si i s tím související změny v programech řídicích desek.

### Hlavní menu:

#### 1) Program:

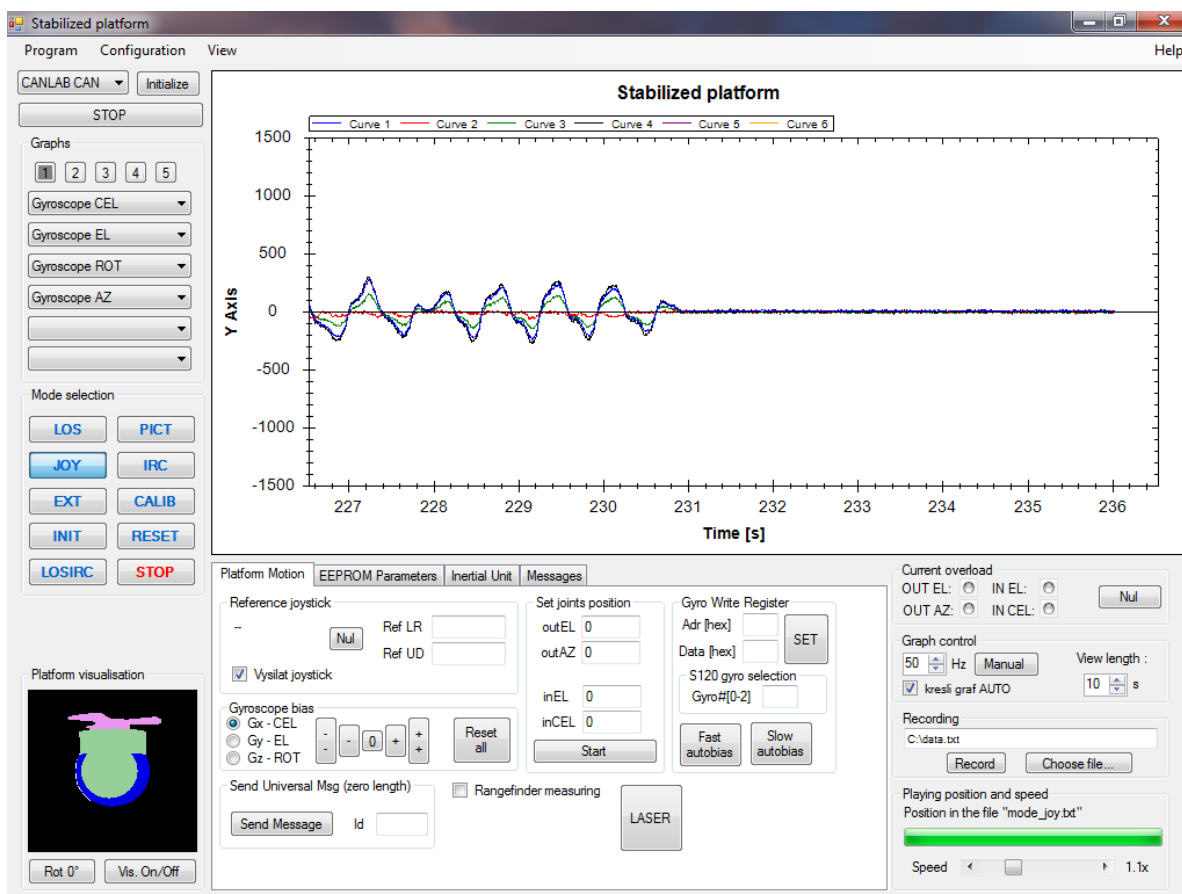
- Exit - ukončit GUI

#### 2) Configuration:

- Protocol - komunikační protokol sběrnice CAN pro interpretaci dat z hlavičky do GUI
- Joystick configuration - výběr joysticku a jeho nastavení
- CAN bus - nastavení přenosové rychlosti a výběr kanálu
- Constants and Filters - seznam konstant, filtrů, PID regulátorů načtených z tabulky konstant („constants table“)
- OpenGL model - nastavení parametrů pro správné zobrazení vizualizačního modelu platformy, obrázek 6.3
- Calibration parameters - nastavení zesílení a offsetů
- Graph setup - doplňkové možnosti pro zobrazení v grafu

#### 3) View:

- GPS Map



Obrázek 6.2: Základní obrazovka GUI.

## Ostatní ovládací prvky:

Dále GUI, obrázek 6.2, obsahuje rolovací nabídku, ve které si lze vybrat použitý CAN analyzátor (Kvaser, CANLab) či přehrávání naměřených dat ze souboru (FILE), následně stačí stisknout tlačítko „Initialize“.

Sekce „Graphs“ obsahuje pět karet, mezi kterými je možné přepínat. Každá z karet umožňuje výběr až šesti proměnných, které budou vykresleny v grafu napravo. K dispozici jsou pouze takové proměnné, které jsou uvedeny v komunikačním protokolu sběrnice CAN pro interpretaci dat z hlavičky do GUI.

Níže se dostáváme do oblasti „Mode selection“, která nám umožňuje přepínání mezi jednotlivými režimy stabilizované základny, které ale musí být implementovány v kódech řídicích desek základny. Režimy, které jsou k dispozici u hlavičky H240, jsou popsány v kapitole 4.5. Modré podsvícení tlačítka signalizuje, ve kterém režimu se stabilizovaná

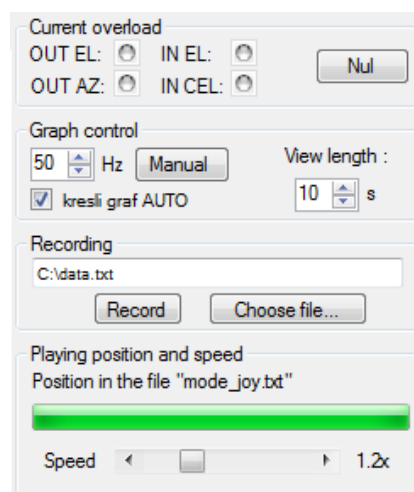
základna právě nachází. Do zdrojových kódů řídicích desek jsem dopsal kód pro softwarový reset procesoru, v souvislosti s existujícím tlačítkem „Reset“ v GUI.

Levý dolní roh základní obrazovky GUI obsahuje vizualizaci stabilizované základny v prostoru, obrázek 6.3, pro správné zobrazení je nutné nastavit parametry OpenGL modelu.



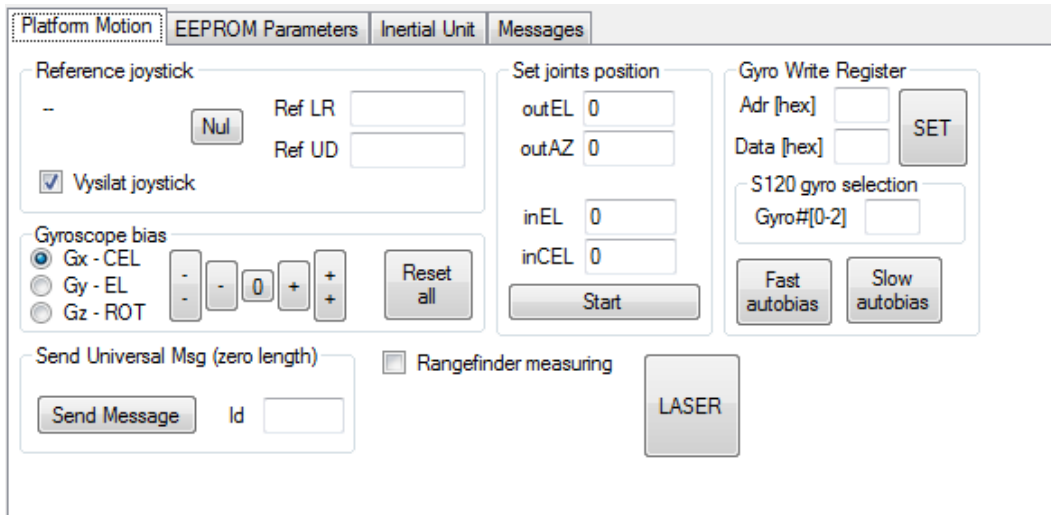
Obrázek 6.3: Vizualizační model stabilizované základny.

Nyní se přesuneme do pravého dolního rohu obrazovky, obrázek 6.4. V sekci „Current overload“ se v případě přetížení některého z motorů zobrazí červená kontrolka, tento motor se okamžitě vypne a jeho opětovné zapnutí lze provést tlačítkem „Nul“. „Graph control“ umožňuje nastavit frekvenci překreslování a délku zobrazovaného okna. A konečně je zde i možné nahrávat a přehrávat naměřená data.



Obrázek 6.4: Indikace přetížení motorů, nastavení grafu, nahrávání naměřených dat.

Na závěr této kapitoly se zaměříme ještě na dvě velice podstatné záložky, první bude záložka „Platform Motion“ na obrázku 6.5 a druhá s názvem „EEPROM Parameters“ na obrázku 6.6.



Obrázek 6.5: Reference joysticku, nastavení kloubových souřadnic či offsetu gyroskopů.

Začneme popisem záložky na obrázku 6.5. Jestliže se stabilizovaná platforma nachází v režimu JOY, můžeme ji zde ovládat pomocí joysticku nebo zadat konstantní hodnoty pro otáčení v osách elevace a azimutu. V sekci „Set joints position“ jde nastavit natočení stabilizované základny do požadovaných úhlů v režimu IRC, úhly se zadávají v rozsahu 0 až 8192, což odpovídá 0° až 360°. Dále v této záložce lze manuálně nastavit offset gyroskopů či odeslat univerzální zprávu na CAN.

Na obrázku 6.6, týkající se druhé zmiňované záložky, vidíme způsob nastavování konstant, parametrů filtrů a PID regulátorů jednotlivých řídicích desek, které jsou zdefinovány v `parameters_elevation.h` či `parameters_azimuth.h`. Pouze stačí z nabídky vybrat příslušnou konstantu, filtr či regulátor a vyplnit všechny dostupné parametry a následně kliknout na tlačítko „SET“, díky kterému se provede uložení do RAM paměti. Případně použít tlačítko „LOAD“ pro zobrazení aktuálně uložené hodnoty v paměti RAM. Tlačítko „STORE“ provádí ukládání do flash paměti, kterou ale nedisponujeme a proto se nic neprovede.

Tlačítko „Backup to file“ slouží k uložení všech parametrů z paměti RAM do souboru a „Restore from file“ slouží k načítání parametrů ze souboru do paměti. Navíc jsem do GUI naprogramoval možnost zaškrtnutí kolonky „Cfile“, která zajistí, že ukládání či

načítání parametrů se provede ve formátu jazyka C. Díky této novince je možné parametry uložit do souboru a následně tento soubor přilinkovat ke kódům řídicích desek, podrobnosti a důvody vytvoření jsou popsány v podkapitole 5.1.

The screenshot shows a software interface with four tabs: Platform Motion, EEPROM Parameters (selected), Inertial Unit, and Messages. The EEPROM Parameters tab contains the following sections:

- Board Selection:** A dropdown menu labeled "Choose the board first:" with "MAMOK\_AZ" selected.
- Backup/Restore:** A section titled "All parameters backup" with buttons for "Backup to file...", "Restore from file...", and a checkbox for "Cfile".
- EEPROM parameters:** A dropdown for "Parameter", a "Data" input field, and buttons for "SET", "STORE", "LOAD", and a checkbox for "HEX".
- FILTER parameters:** A dropdown for "Filter", input fields for "B(z)" and "A(z)", buttons for "SET" and "STORE", and a checkbox for "HEX data".
- PID parameters:** A dropdown for "PID", input fields for "Td", "K", "Ti", "N", "ulow", "uhigh", and "h", and buttons for "SET", "STORE", and "LOAD".

Obrázek 6.6: Nastavení parametrů regulátorů, filtrů a konstant.



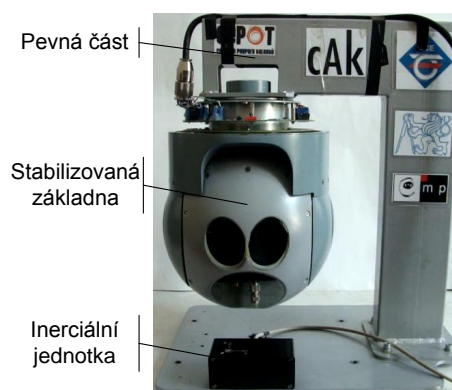
# Kapitola 7

## Provedené experimenty

Tato kapitola představuje provedené experimenty při stabilizaci optické osy a popisuje jejich výsledky. Dále jsou zde zobrazeny naměřené průběhy stabilizace, které byly provedeny pro dva typické pohyby pevné části (letadla).

### 7.1 Zapojení a příprava na experimenty

Prvním krokem bylo provést kompletní zapojení, které spočívalo v propojení stabilizované základny H240 a inerciální jednotky, která měřila úhlové rychlosti pevné části (letadla) pomocí tří gyroskopů, s počítačem pomocí CAN analyzátoru.



Obrázek 7.1: Připevněná inerciální jednotka k pevné části.

Jelikož v experimentu nebylo prováděno testování vnější vizuální zpětné vazby, ale pouze vnitřní rychlostní zpětné vazby, nebylo tedy nutné spouštět podsystém sledování

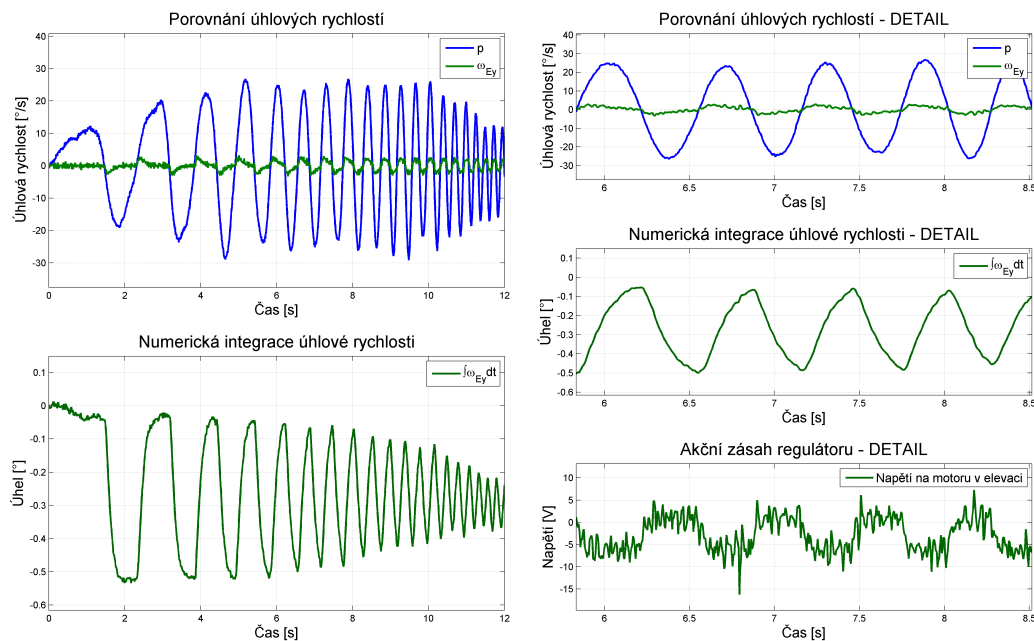
objektů. Následně po zapnutí napájení došlo k naběhnutí dvou řídicích desek (CpuboardAZ, CpuboardEL), jedné měřicí desky (SPIboard) a inerciální jednotky. Po startu byla stabilizovaná základna v režimu STOP, tedy oba rychlostní regulátory kloubů měly odpojený výstup. Dále jak bylo uvedeno v kapitole 4.5.3, byla spuštěna inicializace stabilizované platformy, aby proběhlo určení referenčních bodů inkrementálních senzorů polohy. Až poté byla zapnuta stabilizace optické osy (režim LOS).

## 7.2 Průběh a výsledky experimentů

Veškeré experimenty byly provedeny způsobem, že stabilizovaná základna byla umístěna na stole a následně jsem začal vytvářet poruchový vliv ve formě kývání okolo některé ze zvolených os.

### 7.2.1 Stabilizace elevační osy základny

V prvním pokusu byl jako poruchový vliv použit kývavý pohyb okolo elevační osy základny, který odpovídal rotaci okolo podélné osy nosiče (příčnému sklonu pevné části). Pohyby v ostatních osách byly nulové.



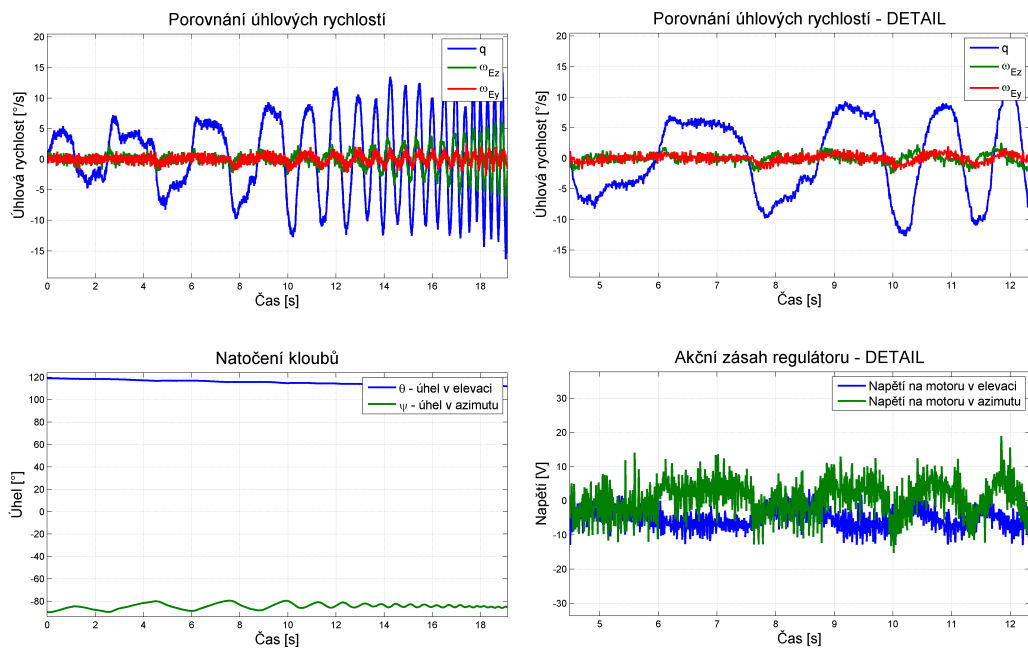
Obrázek 7.2: Dosažené výsledky při stabilizaci elevační osy.



Experiment odpovídal modelu dynamiky v ose elevace zobrazeném na obrázku 4.4. Jako poruchový vliv působila složka (p) vektoru inerciální úhlové rychlosti nosiče, ostatní složky (q, r) byly nulové a úhel v azimutu  $\psi$  byl roven  $-90^\circ$ . Na obrázku 7.2 je vykresleno porovnání úhlových rychlostí v ose elevace ve vnitřním závěsu u kamer a v pevné části, ze kterých je patrné, že došlo k velice dobrému ztlumení poruchového pohybu asi o 20 dB. Při následném výpočtu numerické integrace z úhlové rychlosti  $\omega_{Ey}$ , naměřené gyroskopem v ose elevace, je vidět, že docházelo k pohybu kamer v elevační ose v rozsahu asi  $0,5^\circ$ .

### 7.2.2 Stabilizace příčné elevační osy základny

Ve druhém pokusu byl jako poruchový vliv použit kývavý pohyb okolo příčné osy nosiče (podélnému sklonu pevné části).



Obrázek 7.3: Dosažené výsledky při stabilizaci příčné elevační osy (při použití pravorukého souřadného systému byly kamery vykloněny o  $-60^\circ$  od příčné osy nosiče).

Experiment odpovídal modelu dynamiky v ose příčné elevace zobrazeném na obrázku 4.5. Jako poruchový vliv působila složka (q) vektoru inerciální úhlové rychlosti nosiče. Je použit souřadný systém zobrazený na obrázku 4.1. Na začátku experimentu byl úhel v azimutu  $\psi$  roven  $-90^\circ$  a úhel v elevaci  $\theta$  byl  $120^\circ$ . Na obrázku 7.3 je zobrazeno porovnání

úhlových rychlostí v ose příčné elevace ve vnitřním závěsu u kamer a v pevné části, ze kterých je patrné, že došlo k zatlumení poruchového pohybu asi o 8 dB. Je tam také patrné, že pohyb nosiče (letounu) okolo jeho příčné osy se promítá jak do elevační úhlové rychlosti kamery, tak i do příčné elevační úhlové rychlosti kamery, což potvrzuje, že jsou osy provázané. Pomalé pohyby byly stabilizovány dobře, ale u rychlejších už začínaly být problémy. Dále zde máme vykresleny průběhy úhlů v azimutu, elevaci a akční zásahy regulátorů. Důvodem proč úhel v elevaci malinko ujíždí je, že nebyl dokonale zkalibrován gyroskop v ose elevace.

Na přiloženém CD naleznete několik videozáznamů týkajících se stabilizace optické osy.

# Kapitola 8

## Závěr

### 8.1 Shrnutí

Výsledkem této bakalářské práce jsou funkční zdrojové kódy k inerciálně stabilizované základně H240 pro kamerový systém. Programy byly mnou vytvořeny, ale vycházely z původních programů uvedených v diplomových pracích [3, 7]. Zveřejněna v této práci je však pouze jejich HTML dokumentace, která je přiložena na CD. V dokumentaci je stručně popsán projekt, řídicí desky a vysvětlení jednotlivých funkcí, které jsou zveřejněny pouze formou hlavičkových souborů.

Dále bylo předvedeno nově používané grafické uživatelské rozhraní, které bylo upraveno a přizpůsobeno potřebám kamerové hlavičky H240.

V práci byly postupně popsány mechanická konstrukce stabilizované platformy, elektronická koncepce, souvislosti s řízením kamerové hlavičky, postup tvorby programů řídicích desek, grafické uživatelské rozhraní a také byly předvedeny dva experimenty se stabilizací optické osy.

Na závěr bych rád uvedl, že motivací k provedení veškerých změn a aktualizací na stabilizované základně a GUI, byla možnost oživit dříve vyvinutý systém, aby mohl posloužit k výuce či výzkumu.

### 8.2 Návrh na další práci

Postup další práce na inerciálně stabilizované základně by spočíval ve vyřešení problémů s operátorským rozhraním, aplikace „Operátor H240“, která je nepřehledná, ale především

je nestabilní a neukončuje se standardním způsobem. Doporučil bych provést úpravu kódů stabilizované základny, aby byla kompatibilní s nově používaným operátorským rozhraním u kamerových hlavic Svíce120 a Svíce250.

# Literatura

- [1] ASTROM K. J.; WITTENMARK B. *Computer-Controlled Systems*. 3rd ed. Prentice Hall, USA, 1997, ISBN 7-302-05008-2.
- [2] ŘEZÁČ M. HURÁK Z. *Image-based pointing and tracking for inertially stabilized airborne camera platform*. IEEE Transactions on control systems technology, Vol. 49, No. 99, pp. 1-14, 2010.
- [3] ŽOHA J. Elektronická stabilizace optické osy kamerového systému pro bezpilotní letoun. Diplomová práce, ČVUT, Fakulta elektrotechnická, Katedra řídicí techniky, Praha, 2008.
- [4] ŽOHA J. Dokumentace k elektronickým podsystémům, [technická dokumentace]. Praha: ČVUT, 2008.
- [5] HILKERT J.M. *Inertially stabilized platform technology - Concepts and principles*. IEEE control systems magazine, Vol. 28, No. 1, pp. 26-46, February 2008, ISSN 0272-1708.
- [6] KENNEDY J.P. *Direct Versus Indirect Line of Sight (LOS) Stabilization*. IEEE Transactions on control systems technology, Vol. 11, No. 1, pp. 3-15, January 2003.
- [7] ŘEZÁČ M. Návrh řízení pro systém stabilizace optické osy kamerového systému pro bezpilotní letoun. Diplomová práce, ČVUT, Fakulta elektrotechnická, Katedra řídicí techniky, Praha, 2008.
- [8] MOOG COMPONENTS GROUP [online]. Brush motors: D-2910-g-1ens specificiations, 2008. <http://www.moog.com/literature/MCG/D-2910-G-1ENS.pdf>.
- [9] HURÁK Z. Návrh řídicích algoritmů pro inerciálně stabilizovanou leteckou kamerovou základnu, [prezentace]. Praha: ČVUT, 2011.



# Příloha A

## Fotografie



Obrázek A.1: Pracovní plocha.



Obrázek A.2: Stabilizovaná základna uchycená k podstavci.



Obrázek A.3: Detail stabilizované základny.



# Příloha B

## Obsah přiloženého CD

- Tento text ve formátu PDF.
- Videá činnosti stabilizované základny.
- Návod na nastavení GITu.
- Návod pro nastavení a postupování při programování pod Windows.
- HTML dokumentace zdrojových kódů stabilizované základny.