

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ŘÍZENÍ A STABILIZACE POLOHY MALÉHO LÉTAJÍCÍHO DRONA

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ADRIÁN MARTINEC

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ŘÍZENÍ A STABILIZACE POLOHY MALÉHO LÉTAJÍCÍHO DRONA

CONTROL AND POSITION STABILIZATION OF A SMALL FLYING DRONE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ADRIÁN MARTINEC

VEDOUcí PRÁCE

SUPERVISOR

Ing. STRNADEL JOSEF, Ph.D.

BRNO 2015

Abstrakt

Tato práce řeší stavbu a zprovoznění dronu v podobě letadla, řízeného autopilotem. Zvolený problém jsem vyřešil pomocí teorie letu a regulátorů z oblasti automatizace. V práci jsem vytvořil fungující řešení, ale kvůli nehodě nebylo možné provést závěrečné testy. Přínosem této práce je zjištění a objasnění fungování různých součástí využívajících se při stavbě dronů, objasnění funkce regulátorů a vytvořených knihoven na tuto problematiku.

Abstract

The goal of this thesis is to create a drone in a form of a plane, driven by autopilot. I have solved the selected issues using a theory of flight and regulators from an automatization area. The text describes a functional solution. Unfortunately because of an accident, no final tests were performed. A contribution of this work is in exploring a functionality of various parts that are used with a drone building process. That includes a use of regulators and creation of libraries driving those parts.

Klíčová slova

bezpilotní letadlo, dron, autopilot, pwm, letadlo, arduino, raspberry pi, gyroskop, akcelerometr, barometr, kompas, pid regulátor, psd regulátor

Keywords

drone, autopilot, pwm, aircraft, airplane, arduino, raspberry pi, gyroscope, accelerometer, barometer, compass, pid controller, psd controller

Citace

Adrián Martinec: Řízení a stabilizace polohy malého létajícího drona, bakalářská práce, Brno, FIT VUT v Brně, 2015

Řízení a stabilizace polohy malého létajícího drona

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Josefa Strnadela, Ph.D. . Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Adrián Martinec
20. května 2015

Poděkování

Chcel by som sa poďakovať svojmu vedúcemu práce, Ing. Josefovi Strnadelovi, ktorý sa ujal mojej témy a neustále mi bol k dispozícii pre účely konzultácii ohľadom danej problematiky a usmerňovaním ku jej úspešnému riešeniu.

© Adrián Martinec, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Prehľad dronov a problematika riadenia letu	4
2.1	Rozdelenie dronov	4
2.2	Riadenie letu lietadiel	5
2.2.1	Súradnicové sústavy	5
2.2.2	Ovládacie plochy	8
2.2.3	Pohyb lietadla	9
3	Metódy stabilizácie a riadenia	10
3.1	Automatizácia	10
3.1.1	Stabilita regulačného obvodu	12
3.1.2	PID regulátor	12
3.1.3	PSD regulátor	13
3.2	Stabilita	14
3.2.1	Priečna stabilizácia	14
3.2.2	Pozdĺžna stabilizácia	15
3.2.3	Smerová stabilizácia	16
3.3	Riadenie letu	16
4	Realizačné prostriedky	17
4.1	Dron	17
4.2	Platformy	18
4.2.1	Arduino Leonardo	18
4.2.2	Raspberry Pi A+	19
4.2.3	10 Degrees of freedom	20
4.3	Komunikačné rozhrania	21
5	Implementácia	22
5.1	Elektrické napájanie a prepojenie modulov	22
5.1.1	Protokol	23
5.2	Arduino Leonardo	24
5.3	Raspberry Pi	25
6	Testovanie	27

7 Záver	28
7.1 Výsledky testovania	28
7.2 Zvolené riešenie	28
7.3 Rozvoj	28
A Obsah CD	33

Kapitola 1

Úvod

V dnešnej dobe sa stále posúva svet informačných technológií dopredu neuveriteľnou rýchlosťou. Ich smerovanie a zamerania zasahujú už do väčšiny ostatných odborov. Jednou z takýchto oblastí je aj robotika. Robotika sa snaží nahradiť ľudský faktor v oblastiach, ktoré je možné automatizovať. Roboty v továrňach nahrádzajú ľudskú prácu, a tým vylepšujú napríklad kvantitu, kvalitu a cenu výroby. Stroje musia pracovať autonómne a poprípadе ešte aj v skupine na výrobnnej linke, kde každý má svoju úlohu. Medzi roboty samozrejme patrí aj dron.

Slovo dron pochádza z anglického jazyka a výraz označuje bezpilotný lietajúci objekt alebo loď, schopný samostatnej navigácie bez zásahu človeka a teda sa dokáže správať autonómne. Existuje ešte jeden výklad, ktorý pozmeňuje definíciu na vzdialené riadenie zo zeme človekom. Samozrejme nič nebráni kombinácií týchto prístupov.

Táto práca sa venuje problematike riadeniu drona lietadla od základov. Použitý dron predstavuje malé lietadlo. Prvá časť práce je o dronoch ako takých, a o uvedení do problematiky spojenej s ich riadením. Samotné riadenie a stabilizovanie je rozobrané v tretej kapitole. Vybrané prostriedky na realizáciu sa nachádzajú v štvrtej kapitole práce. Ďalej nasleduje implementácia a popis rôznych problémov s ňou spojených. Jedná sa o autonómne pracujúci stroj a preto prácu uzatvára testovanie drona v reálnych prostrediach. Na záver sú zhrnuté všetky získané poznatky počas tvorby projektu.

Kapitola 2

Prehľad dronov a problematika riadenia letu

Lietajúce drony sa zvyknú označovať skratkou z angličtiny unmanned aerial vehicle UAV. Pôvodne drony vznikli vo vojenskej sfére, kde sú technologicky najviac pokročilé. Dostupné informácie o dronoch nie sú veľmi hodnotné, pretože tento smer je veľmi mladý a spoločnosti vyvíjajúce svoje drony chránia dôležité detaily o nich. Napriek tomu je možné nájsť rôzne drony a ich využitie, či už existujúcich alebo iba plánovaných.

2.1 Rozdelenie dronov

Oficiálne rozdelenie dronov neexistuje, no logicky sa dajú rozdeliť podľa niektorých hlavných črt:

Typ - lietadlo, vrtuľník, loď alebo rôzna N-koptéra

Pohon - elektrický, spaľovací, tryskový, hybridný[22]

Využitie - vojenské, pošta, fotografovanie a mnohé ďalšie

Ovládanie - manuálne vzdialené, automatické



Obrázok 2.1: Drony: Phantom 2 vision určený pre fotografovanie (vľavo hore) [13], bojový dron Reaper (vpravo hore) [23], zásielkový Amazon dron (vľavo dole) [27] a hybridný dron (vpravo dole) od Amazon, Google a DHL [22]

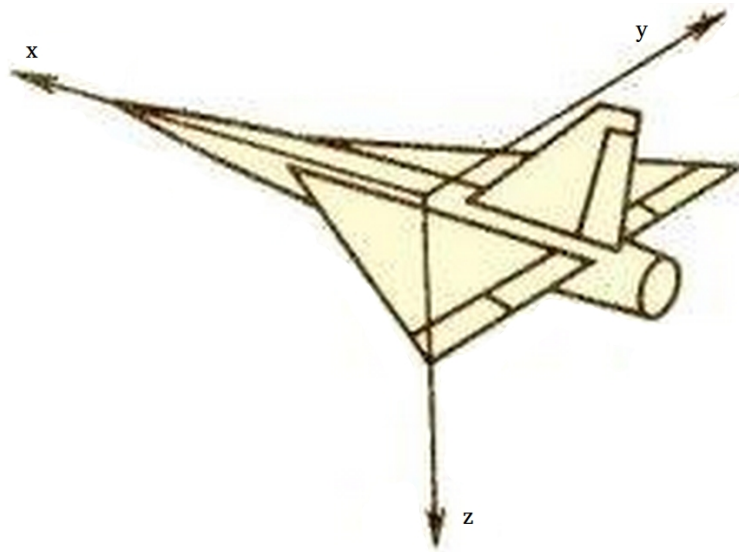
2.2 Riadenie letu lietadiel

Riadenie letu zahŕňa základné aerodynamické kormidlá a ostatné prostriedky k ovládaniu letu podľa požadovanej trajektórie, napríklad výška letu, smer a uhol stúpania, s určitými kinematickými parametrami, napríklad osové zložky rýchlosti, zrýchlenie, uhlové rýchlosti. Základné kormidlá pre riadenie letu je výškové kormidlo, smerové kormidlo a krídelká. Ostatné prostriedky sú vztlakové mechanizmy, vyvažovacie plôšky, brzdiace klapky alebo štíty, ale aj ťah pohonnej jednotky a ďalšie systémy vytvárajúce sily na lietadlo za účelom jeho riadenia. Podľa konštrukcie systému k ovládaniu základných kormidiel a ostatných prostriedkov pre riadenie letu, možno riadenie rozdeliť na mechanické, elektrické, hydraulické a prípadne kombinované. Podľa spôsobu začlenenia pilota do riadenia môže byť lietadlo riadené priamo pilotom, teda priamy zásah pilota do výchyliek kormidiel a ďalších prostriedkov riadenia na základe informácií z letových prístrojov, alebo automatické, kde zásahy prevádza autopilot na základe počítačom vyhodnotených letových údajov a stavu riadiacich systémov pilotom zadaných parametrov letu. Automatické riadenie má rôzne formy automatizácie od kurzových autopilotov až po systémy riadenia letu vrátane automatického pristátia [26].

2.2.1 Súradnicové sústavy

Zvyklosti v používaní súradnicových sústav sa v priebehu vývoja lietadiel takisto vyvíjali a ustálili na dvoch formátoch rozlišujúcich sa v označení a orientácii súradnicových osí. V Českej republike sú sústavy letectva normalizované normou ČSN 310201, ktorá sa opiera

a je kompatibilná s aktuálnymi medzinárodnými normami ISO 1151-1 a ISO 1151-2 2.2 [33].



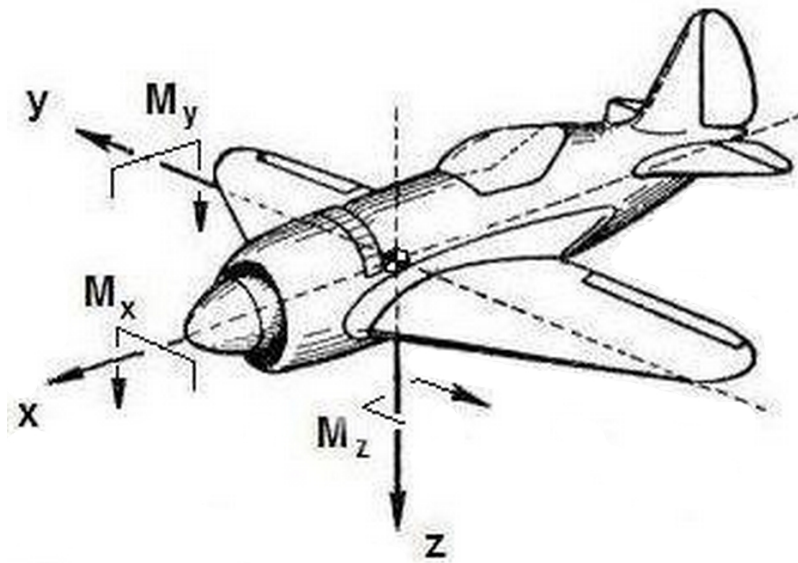
Obrázok 2.2: Platný systém podľa ISO[33]

Lietadlová súradnicová sústava je pevne zviazaná s lietadlom 2.3. Jej začiatok sa nachádza v ťažisku lietadla a os x , ďalej pozdĺžna os lietadla, je približne rovnobežná s pozdĺžnou osou trupu. Kladný zmysel osi x smeruje ku predku lietadla. Os z , ďalej zvislá os lietadla, zvierá s osou x pravý uhol a jej kladný smer smeruje pod lietadlo. Osi x a z určujú rovinu súmernosti lietadla. Os y , ďalej bočná os lietadla, je kolmá na rovinu súmernosti lietadla a jej kladný zmysel mieri na pravé krídlo. Na obrázku 2.3 sú označené aj kladné zmysly momentov, ktoré vyvolávajú rotačný pohyb okolo jednotlivých osí. Otáčanie okolo osi x sa volá klonenie, okolo osi y klopenie a rotácia okolo osi z zatáčanie. Z toho vyplývajú aj názvy momentov, ktoré ich vyvolávajú.

M_x - moment klonivý

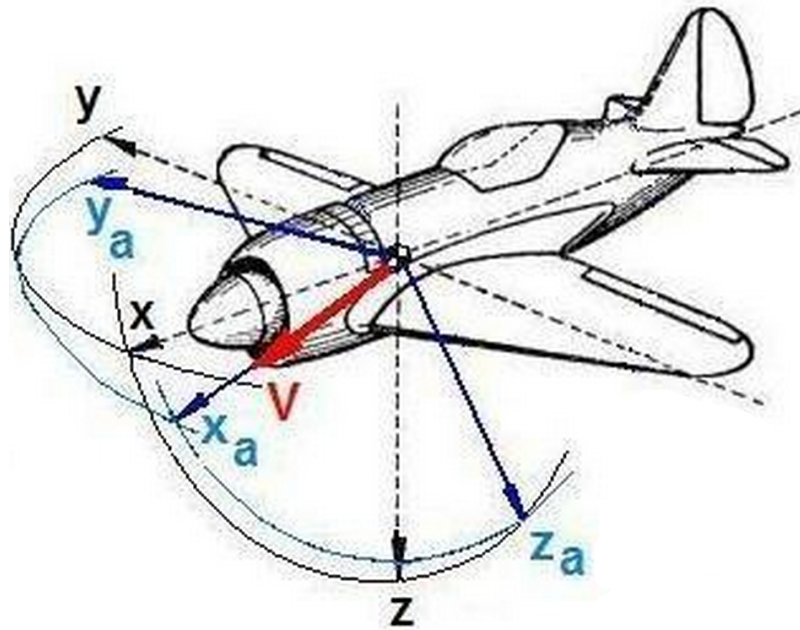
M_y - moment klopivý

M_z - moment zatáčavý



Obrázok 2.3: Lietadlová súradnicová sústava[33]

Aerodynamická súradnicová sústava nie je pevne zviazaná s lietadlom, má s ním zviazaný iba začiatok, ktorý je umiestnený v ťažisku lietadla 2.4. Kladný smer osi x_a je určený vektorom V okamžitej rýchlosti lietadla, ktorý nemusí byť kolineárny s pozdĺžnou osou lietadla. Os z_a je kolmá na os x_a a vždy leží v rovine súmernosti lietadla. Os y_a je kolmá na rovinu x_a a z_a a smeruje v kladnom smere vpravo od tejto roviny pri pohľade v smere rýchlosti. K osiam tejto sústavy sa vzťahujú aerodynamické sily. Kladné zmysly aerodynamických síl sú opačné ako zmysly príslušných aerodynamických osí. Odporová sila pôsobí v smere a v opačnom zmysle k osi x_a . Túto os nazývame odporovou osou. Vztlaková sila pôsobí v smere a v opačnom zmysle k osi z_a . Hovoríme o vztlakovej osi. Priečna aerodynamická sila v opačnom zmysle osi y_a , teda priečnej osi.



Obrázok 2.4: Aerodynamická súradnicová sústava[33]

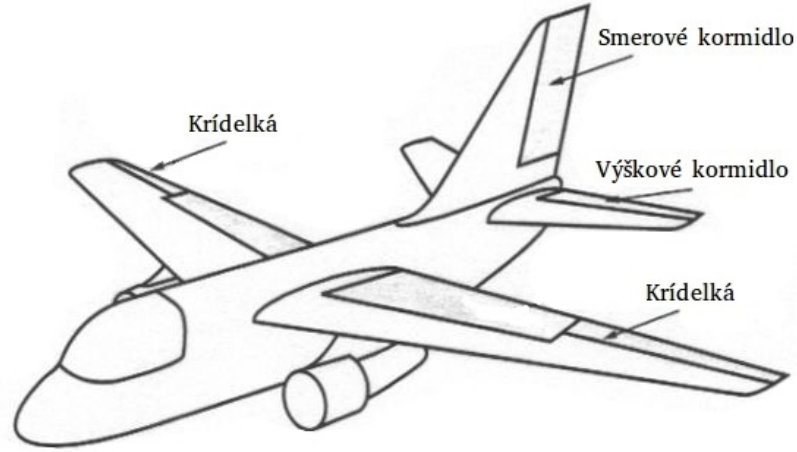
2.2.2 Ovládacie plochy

Ovládacie plochy tvoria primárne riadenie lietadla. Ovládacie plochy sú krídelka, výškové kormidlo, smerové kormidlo 2.5 [26].

Výškové kormidlo je aerodynamické kormidlo k vyvolaniu klopivých momentov k ťažisku lietadla pomocou prírastku alebo úbytku vztlaku na vychýlenom kormidle umiestnenom na dostatočnom ramene pred alebo za ťažiskom. Zaisťuje potrebné momenty pre udržanie ustálených letových režimov, napríklad horizontálny let a stúpanie, alebo preprechody medzi letovými režimami.

Smerové kormidlo umožňuje zatačovať, a tým meniť kurz lietadla. Pri lete v kludnej atmosfére je v priamočiari lete smerové kormidlo nevychýlené, lebo vzhľadom k symetričnosti lietadla nevzniká pri priamom lete zatačavý moment od aerodynamických síl. Smerové kormidlo sa vychýľuje iba pri lete v zatáčke, kvôli letu správnou zatáčkou s výchylkami krídelok, alebo pri lete s bočným vetrom. Pri jednomotorových lietadlách s vrtuľovým pohonom môže byť použitá výchylka smerového kormidla k eliminácii zatačavého momentu od šikmo ofukovaných zvislých zadných plôch stáčajúcim sa vrtuľovým prúdom.

Krídelká alebo kormidlá priečneho riadenia sú aerodynamické prostriedky pre ovládanie klopenia, teda otáčanie okolo pozdĺžnej osi lietadla. Klonivý moment od krídelok je vyvolaný opačnými výchylkami ľavého a pravého krídelka. Vychýlené krídelko smerom dole vyvoláva na polkrídle prírastok vztlaku, opačne vychýlené na druhom polkrídle naopak pokles vztlaku. Takto vzniknuté rozdielne sily vytvárajú klonivý moment.



Obrázok 2.5: Ovládacie plochy lietadla[37]

2.2.3 Pohyb lietadla

Matematický popis pohybu lietadla vyjadrujú 3 silové rovnice 2.1, 2.2 a 2.3, kde v_x , v_y a v_z sú zložky vektoru rýchlosti, ktorý je zviazaný s pozdĺžnou osou aerodynamyckej sústavy, X je odporová sila, Z je vztlaková sila a Y je bočná sila [30].

$$F_x = X - mg \sin \theta = m \left(\frac{dv_x}{dt} + \omega_y v_z - \omega_z v_y \right) \quad (2.1)$$

$$F_y = Y + mg \cos \theta \sin \varphi = m \left(\frac{dv_y}{dt} + \omega_z v_x - \omega_x v_z \right) \quad (2.2)$$

$$F_z = Z + g \cos \theta \cos \varphi = m \left(\frac{dv_z}{dt} + \omega_x v_y - \omega_y v_x \right) \quad (2.3)$$

Ďalej popisujú pohyb 3 momentové rovnice 2.4, 2.5 a 2.6. Tieto momenty pôsobia okolo ťažiská lietadla, sú vyvolané aerodynamickými silami a neobsahujú gravitačné zložky.

$$M_x = I_x \frac{d\omega_x}{dt} - \omega_y \omega_z (I_z - I_y) - (\omega_x \omega_y + \dot{\omega}_z) I_{zx} \quad (2.4)$$

$$M_y = I_y \frac{d\omega_y}{dt} - \omega_x \omega_z (I_z - I_x) - (\omega_z^2 - \omega_x^2) I_{zy} \quad (2.5)$$

$$M_z = I_z \frac{d\omega_z}{dt} - \omega_x \omega_y (I_x - I_y) - (\dot{\omega}_x - \omega_y \omega_z) I_{zx} \quad (2.6)$$

Eulerové kinematické rovnice 2.7, 2.8 a 2.9, dávajú do súvislosti uhlové rýchlosti v telesovej sústave a derivácii Eulerových uhlov.

$$\omega_x = \dot{\phi} - \dot{\psi} \sin \theta \quad (2.7)$$

$$\omega_y = \dot{\theta} \cos \phi + \dot{\psi} \cos \theta \sin \phi \quad (2.8)$$

$$\omega_z = \dot{\psi} \cos \theta \cos \phi - \dot{\theta} \sin \phi \quad (2.9)$$

Kapitola 3

Metódy stabilizácie a riadenia

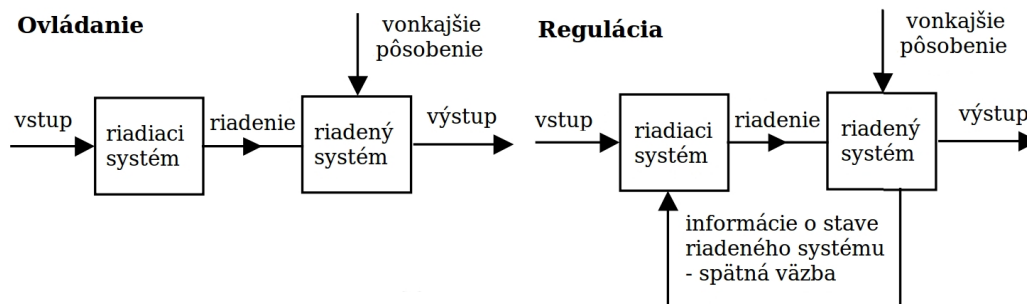
Od stabilizácie očakávam schopnosť udržať lietadlo v jednej letovej hladine a udržiavanie smeru letu. Riadenie letu by malo zvládnuť zmeniť letovú hladinu a takisto zmeniť smer letu. V tejto kapitole najprv rozoberám automatizáciu, ktorá je spojená so všeobecným riešením stabilizácie. Následne sa venujem samotným stabilizačným metódam a systému riadenia letu.

3.1 Automatizácia

”Všude kolem nás vidíme snahu o neustálé zvyšování produktivity práce. Úkolem inženýra v tomto procesu je hledat nové pracovní postupy s minimální spotřebou času a nákladů. Jednotlivé pracovní úkony musí být co nejkratší a nejjednodušší, aby vyžadovaly minimum lidských sil. K tomu všemu musí přispívat především automatizace výrobních procesů.” [36]

”K automatizaci vede snaha člověka osvobodit se nejen od fyzické činnosti, ale i od jednotvárné a unavující činnosti duševní. Činnost člověka přebírají automaty, počítače a prvky umělé inteligence. Tento poměrně složitý proces, při němž lidská řídicí činnost při výrobě i mimo výrobní proces je nahrazována činností různých přístrojů a zařízení je nazývána automatizací.” [36]

Automatizácia sa zaoberá od základu riadením. Riadenie je snaha pôsobiť na riadený objekt tak, aby sa dosiahlo očakávaného výsledku. Riadenie sa rozdeľuje hlavne podľa toho, či je výsledok riadenia spätne kontrolovaný. Základnými formami riadenia sú ovládanie a regulácia. Ovládanie je riadenie bez spätnej väzby a regulácia so spätnou väzbou. Regulácia sa snaží o udržanie určitej fyzikálnej veličiny na hodnote, ktorú očakávame. Porovnaním tejto veličiny a očakávanej hodnoty sa získava odchýlka. Do regulačného procesu sa zasahuje, tak aby sa táto odchýlka odstránila [36].



Obrázok 3.1: Bloková schéma ovládania a regulácie[36]

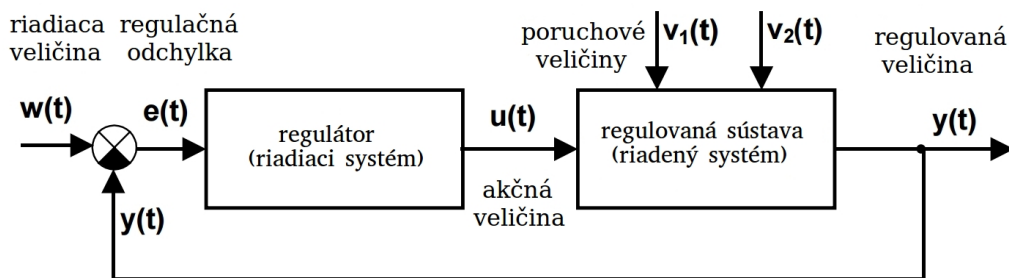
Ďalšie vyššie formy riadenia:

- Optimálne riadenie je s maximálnou účinnosťou alebo v najkratšom čase.
- Adaptívne riadenie je, keď systém dokáže meniť svoje parametre, aby proces riadenia prebehal stále optimálne.
- O umelej inteligencii hovoríme vtedy, ak adaptívny systém je schopný ukladať informácie a neskôr tieto skúsenosti znova použiť. Dostávame učiaci systém, ktorý je najvyššou formou riadenia.

Automatické riadenie sa uskutočňuje niekoľkými spôsobmi, ktoré sa líšia v tom, akým princípom riadiaci systém pôsobí na riadený systém.

- Logické riadenie používa k riadeniu dvojhodnotovú veličinu a teda logickú 1 alebo 0.
- Spojité riadenie nastáva, ak je riadený systém riadený v čase spojitou.
- Diskrétné riadenie nastáva, ak je riadený systém riadený v čase diskretné.
- Fuzzy riadenie je vhodné pre systémy, ktoré nevieme popísať, ale vieme ich riadiť.

Regulácia sa uskutočňuje v regulačnom systéme nazvanom regulačný obvod. Regulačný obvod sa rozdeľuje na dve hlavné časti, a to regulátor a regulovanú sústavu. Regulátor alebo riadiaci systém je zariadenie, ktoré uskutočňuje samotnú reguláciu. Regulovaná sústava je objekt regulácie. Regulovaná veličina je výstupom regulovanej sústavy, ktorá sa reguláciou snaží udržať na požadovanej hodnote. Riadiaca veličina nastavuje hodnotu, ktorú má dosahovať regulovaná veličina. Regulačná odchýlka je vstupom pre regulátor a získava sa rozdielom riadiacej veličiny s regulovanou veličinou. Výstupom regulátora je akčná veličina, ktorá je následne vstupom pre regulovanú sústavu s cieľom zmenšiť regulačnú odchýlku. Na regulovanú sústavu pôsobia poruchové veličiny, ktoré ovplyvňujú regulovanú veličinu a to je dôvodom celej regulácie.



Obrázok 3.2: Bloková schéma regulácie[36]

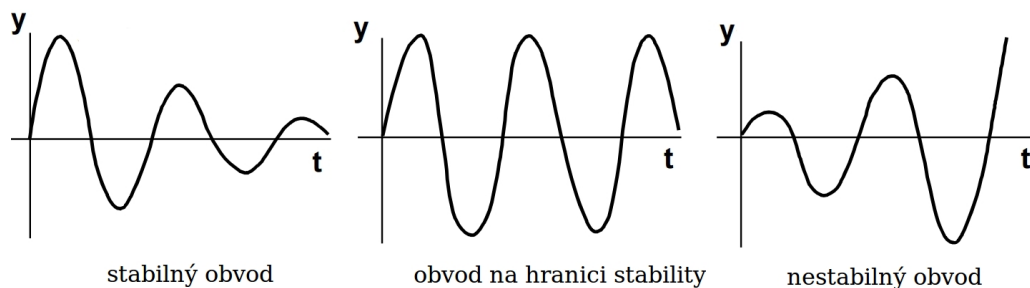
- w - riadiaca veličina
- y - regulovaná veličina
- u - akčná veličina
- $e = w - y$, e - regulačná odchýlka
- v - poruchové veličiny

Pre potreby tejto práce nás ďalej bude zaujímať iba spojité a diskrétno riadenie vo forme optimálneho a adaptívneho riadenia. Pre spojité riadenie sa používa PID regulátor a pre diskrétno PSD regulátor.

3.1.1 Stabilita regulačného obvodu

”Regulační obvod je stabilní, jestliže po svém vychýlení z rovnovážného stavu a odstranění vzruchu, který vychýlení způsobil, je schopen se ustálit v rovnovážném stavu.”[36]

Regulačný obvod je stabilný, ak regulovaná veličina y sa ustáli na hodnote riadiacej veličiny w alebo sa limitne blíži riadiacej veličine w [36].



Obrázok 3.3: Tri druhy stability obvodu[36]

3.1.2 PID regulátor

Regulátor spracováva odchýlku troma spôsobmi. Proporcionálny regulátor, ďalej P regulátor, odchýlku zosilňuje tak, že akčná veličina u je úmerná regulačnej odchýlke e vzorec 3.1 [36].

$$u(t) = r_0 e(t) \quad (3.1)$$

Druhým spôsobom je akčná veličina u úmerná integrálu regulačnej odchýlky e vzorec 3.2. V tomto prípade hovoríme o integračnom, ďalej I regulátore.

$$u(t) = r_{-1} \int e(t) dt \quad (3.2)$$

Posledný prípad je derivačný regulátor, ďalej D regulátor, ktorý nie je možné realizovať, pretože by došlo k rozpojeniu regulačného obvodu v ustálenom stave, ak by akčná veličina u bola úmerná derivácii regulačnej odchýlky e vzorec 3.3.

$$u(t) = r_1 e(t)' \quad (3.3)$$

Kombinovaním týchto základných typov získame ďalšie regulátory. Proporcionalne-integračný 3.4, ďalej PI regulátor, proporcionalne-derivačný regulátor 3.5, ďalej PD regulátor, a proporcionalne-integračne-derivačný regulátor 3.6, ďalej PID regulátor. PID regulátor je všeobecný typ regulátora tvorený troma zložkami P, I a D a ostatné môžeme získať zmenou hodnoty konstant príslušnej zložky r_0 , r_{-1} a r_1 na nulu vo vzorci 3.6. Tento typ regulátora sa používa pre spojité riadenie.

$$u(t) = r_0 e(t) + r_{-1} \int e(t) dt \quad (3.4)$$

$$u(t) = r_0 e(t) + r_1 e(t)' \quad (3.5)$$

$$u(t) = r_0 e(t) + r_{-1} \int e(t) dt + r_1 e(t)' \quad (3.6)$$

Funkcia jednotlivých zložiek PID regulátora [29]:

P regulátor podľa vzorca 3.1, smeruje hodnotu regulovanej veličiny ku hodnote riadiacej veličiny silou, ktorá závisí od regulačnej odchýlky a konstanty r_0 . Na regulovaný objekt neustále vplývajú rôzne poruchové veličiny. Preto sa môže stať napríklad kvôli zotrvačnej sile alebo veľkej akčnej veličine, že regulátor hodnotu riadiacej veličiny minie a dochádza k oscilácii.

I regulátor na základe 3.2 spočíta všetky chyby od začiatku merania. Takto sa dá docieľť nulová odchýlka v riadení v nekonečnom čase. Znamená to, že ak regulovaný objekt neustále ovplyvňuje poruchová veličina, I regulátor ju vyrovnáva.

D regulátor započítava tendenciu zmeny chyby 3.3. To zaručuje zníženie alebo úplné zamedzenie oscilácie hodnoty regulovanej veličiny okolo hodnoty riadiacej veličiny.

3.1.3 PSD regulátor

Číslicový regulátor, teda pracujúci s diskretnými hodnotami, má rovnakú funkciu ako spojité regulátor a to vstupnú regulačnú odchýlku zosilňovať, integrovať a derivovať. Vychádza sa z PID regulátora nahradením jednotlivých zložiek. Výsledkom je vzorec pre číslicový regulátor 3.7. Presný postup prevodu je možné nájsť v literatúre [36].

$$u(k) = r_0 \left(e(k) + \frac{T}{T_i} \sum_{i=0}^k e(i) + \frac{T_d}{T} (e(k) - e(k-1)) \right) \quad (3.7)$$

Hodnota integrálu sa počíta pomocou sumy a hodnota derivácie cez diferenciu. Nazývajú sa proporcionalne-sumovo-diferenčné regulátory, ďalej PSD regulátory.

3.2 Stabilita

Stabilita je vlastnosť lietadla zachovávať a po poruche obnovovať zvolený režim letu bez zásahu pilota. V tomto zmysle musíme hovoriť skôr o stabilite režimu letu. Hovoriť o stabilite lietadla je nepresné, pretože lietadlo môže vykazovať v niektorých režimoch letu stabilitu a v iných režimoch môže byť nestabilný. O stabilnom lietadle môžeme hovoriť až vtedy, keď preskúmame, že dané lietadlo je stabilné vo všetkých režimoch letu, ktoré je schopné dosiahnuť [35, 26].

Stabilita lietadla sa rozdeľuje na tri časti priečna stabilita, pozdĺžna stabilita a smerová stabilita. Priečna stabilita je schopnosť vyvažovať nežiaducu rotáciu okolo osi x , pozdĺžna okolo osi y a smerová okolo osi z z lietadlovej súradnicovej sústavy. Na stabilitu lietadla má vplyv veľké množstvo činiteľov. Keď zanedbám vonkajšie vplyvy ako sú vzdušné turbulencie, výška letu alebo rýchlosť, dá sa konštatovať, že miera stability je závislá na vzájomnom prepojení geometrického a hmotného usporiadania lietadla.

Geometrické usporiadanie sa dá chápať ako vzájomné postavenie trupu, krídel a vyvažovacích aerodynamických plôch. Lietadlá s pevnými nosnými plochami sa podľa umiestnenia krídel rozdeľujú na tri skupiny, hornoplošné, plnoplošné a stredplošné. Najvyššiu mieru pozdĺžnej a priečnej stability vykazujú lietadlá s hornoplošným usporiadaním, čo je spôsobené polohou ťažiska pod pôsobiskom vztlakovej sily. Ďalej je stabilita ovplyvnená vzájomnou polohou výškových kormidiel a krídel, respektíve dĺžkou ramena, na ktoré pôsobí vyvažovacia sila. V prípade priečnej stability je dôležitým pojmom vzopätie krídel. Ide o uhol, ktorý zvierá krídlo so základnou rovinou lietadla, pričom za kladné vzopätie sa považuje také, keď sú krídla zalomené v zápornom zmysle osi z . Kladné vzopätie zlepšuje priečnu stabilitu až do takej miery, že nie je nutné použiť ovládacie plochy priečného náklonu a pri lete s modelom sa pre zatáčanie používa len smerové kormidlo.

Hmotným usporiadaním sa myslí rozloženie hmoty lietadla okolo osi otáčania a polohy ťažiska oproti výslednej aerodynamickej sile v smere kolmom na rýchlosť letu. Rozloženie hmoty je charakterizované momentom zotrvačnosti. Rýchlosť reakcie lietadla na vonkajší podnet je priamo úmerná jeho veľkosti podľa vzorca 3.8, kde r je vzdialenosť ťažiska i-té časti lietadla od osi, pre ktorú moment počítame a m je jej hmotnosť. Poloha ťažiska lietadla musí byť taká, aby pri vychýlení lietadla z rovnovážnej polohy pôsobil moment k ťažisku v opačnom zmysle ako moment spôsobujúci vychýlenie. Všeobecne platí, že poloha ťažiska musí ležať pred výslednicou vztlakovej sily v smere letu.

$$J = \sum_i^n r_i^2 m_i \quad (3.8)$$

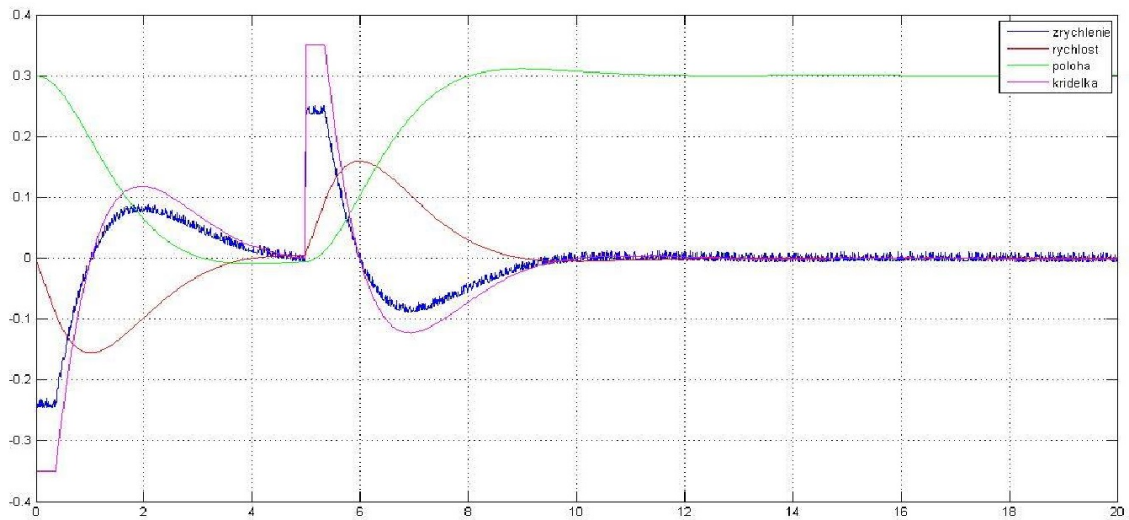
Na stabilitu letu sa dá pozeráť ako na dôsledok stability samotnej konštrukcie lietadla a schopnosti vhodného zásahu do riadenia. Stabilný let je taký, pri ktorom k výraznej zmene smeru letu dochádza iba za účelom vychýlenia ovládacích plôch a po vykonaní manévru sa lietadlo vracia späť do rovnovážneho stavu. Ovládanie výchýliek je v tomto prípade úlohou navrhového autopilota a mieru vychýlenia kormidla na veľkosť požadovanú, prípadne vyrovnávaného uhlového zrýchlenia okolo prečnej, pozdĺžnej a smerovej osi rieši regulácia regulátorom.

3.2.1 Priečna stabilizácia

Riadenie lietadla okolo priečnej osi lietadla je realizované krídelkami. Ich vychýlením vzniká moment k ose x a následne rotácia lietadla. Regulátor má ako riadiacu veličinu žiadaný uhol

natočenia a regulovaná veličina je skutočné natočenie lietadla, v mojom prípade výstupu z gyroskopu [35, 26].

V simulácii overujúcu túto stabilitu nie sú zakomponované žiadne prirodzené tlmenia, ktoré sa v skutočnosti značne prejavujú. Nasledkom toho je chovanie modelového lietadla nestabilné a regulátor je tak preverený pre najhoršie možné varianty letu. Lietadlo v simulácii je považované za hmotný bod. Dynamické vlastnosti sú reprezentované momentom zotrvačnosti a celkovou hmotnosťou lietadla. Program najprv spočíta moment k ťažisku lietadla vzniknutý prírastkom sily na krídle vplyvom vychýlenia krídelka. Vzniknutý moment spôsobí uhlové zrýchlenie. Ďalej sa uvažuje, že dané uhlové zrýchlenie pôsobí po dobu, ktorá je zvolená ako doba medzi jednotlivými výstupmi z gyroskopu. Prírastok sily v spojitosti s uhlom natočenia krídelka je spocítaný ako odhadnutá konštanta * výchylka krídelka. Prírastok je teda lineárne závislý na výchylke, čo priamo neodpovedá realite, ale pre potreby simulácie funkčnosti regulátora je dostačujúci. Simulátor som vytvoril v programe matlab [12] a je v prílohe A.



Obrázok 3.4: Simulácia PSD regulátora priečnej stabilizácie

Prvých 5 sekúnd simulácie je riadiaca veličina nastavená na 0 radiánov a potom sa mení na hodnotu 0,3 radiánov. Poloha, teda regulovaná veličina znázornená zelenou farbou, potvrdzuje funkčnosť regulátora. Zaujímavé sú výchylky krídelok, znázornené fialovou farbou. Taktiež na zrýchlení, ktoré pôsobí na lietadlo, je vidieť poruchové veličiny.

3.2.2 Pozdĺžna stabilizácia

Pozdĺžnu stabilizáciu lietadla ovládajú výškové kormidlá. Pri uhlovom zrýchlení okolo osi y pôsobí vychýlenie výškového kormidla silou na rameno daným vzdialenosťou kormidla od tejto osi proti pôsobenému pohybu [35, 26].

Túto stabilizáciu je možno regulovať niekoľkými spôsobmi. Jednoduché riešenie je použiť regulátor, ktorý bude regulovať odchylku natočenia okolo pozdĺžnej osi lietadla. Tento spôsob ale nezaručí stabilný let v rovnakej letovej hladine. Pridaním regulátora pre reguláciu výšky a ich skombinovaním je pravdepodobne možné udržiavať stabilný let v nemiacej sa letovej hladine.

3.2.3 Smerová stabilizácia

Smerovú stabilizáciu lietadla ovláda aj smerové kormidlo. Môžeme ho regulovať pomocou riadiacej veličiny a odchýlky natočenia okolo smerovej osi lietadla. Smerové kormidlo sa používa hlavne pre elimináciu poruchových veličín ako je napríklad bočný vietor [35, 26].

3.3 Riadenie letu

Horizontálny let patrí medzi základné letové režimy atmosferických lietadiel. Za horizontálny let sa bežne považuje let v konštantnej výške nad povrchom zeme priamym smerom [35, 26].

Pokiaľ je treba meniť letovú hladinu je potrebné uviesť lietadlo do rotácie pomocou výškových kormidiel a zmeniť natočenie lietadla proti zemi a rýchlosti letu. Vplyvom zmeny uhlu dochádza k zmene vztlakovej sily a lietadlo začne stúpať respektíve klesať, prípadne pomocou kolmej zložky ťahu motora bude tato rýchlosť navyšovaná alebo znižovaná. Pre realizáciu je možné použiť systém dvoch regulátorov popísaný v pozdĺžnej stabilizácii, tak že budeme meniť hodnotu riadiacej veličiny regulátora, ktorý reguluje výšku.

Pre zmenu smeru letu sa dá použiť smerová stabilizácia a to zmenou riadiacej veličiny jej regulátora. Avšak takéto použitie smerového kormidla je preň atypické a väčšinou sa používa len na stabilizáciu kvôli poruchovým veličinám. Ďalšou možnosťou je zmena riadiacej veličiny regulátora priečnej stabilizácie. Lietadlo začne meniť smer a po dosiahnutí vyžadovanej zmeny smeru letu sa riadiaca veličina nastaví späť na stabilizačnú hodnotu.

Riadenie výkonu motora bude realizované regulátorom, ktorého riadiaca veličina bude rýchlosť a regulovaná veličina skutočná rýchlosť lietadla.

Kapitola 4

Realizačné prostriedky

Na dosiahnutie cieľa bolo nutné, aby som si zaistil prístup k rôznym súčiastkam a prípadne si ich upravil. Moje vstupné požiadavky boli letuschopný dron, modul pre autopilota s vysokým výpočtovým výkonom, modul pre riadenie ovládacích plôch a senzory predstavujúce spätnú väzbu pre autopilota. Veľmi dôležitým kritériom pre každú súčiastku, ktorá pre svoju funkciu vyžadovala elektrické napájanie, aby jej energetický odber bol čo najmenší z dôvodu zaistenia čo najväčšej dĺžky letu.

4.1 Dron

Vychádzal som z rc modelu hydroplánu coota od spoločnosti Hobbyking[8]. Model lietadla je z polystyrénu a plastu. Dôvodom zvolenia tohoto typu modelu je spevnená spodná časť lietadla vhodná pre testovanie a veľký vnútorný priestor pre elektroniku. Model je z bazára, čo nakoniec nebol správny krok kvôli poškodeným servám a neidentifikovateľným motorom bez regulátora a bol som nútený všetko vymeniť. Vnútorný priestor nebol nakoniec dostačujúci svojimi rozmermi, tak som model upravil a rozšíril. V neposlednom rade som vymenil aj závesy ovládacích plôch, ktoré predtým nedovoľovali rovnaký uhol natočenia na každú stranu.

Parametre:

- Rozpätie 930 mm
- Dĺžka 680 mm
- Váha 428g
- Motor 11.1 V, 0.8-14 A, synchronný striedavý
- Vrtuľa 17.78x10.16 cm
- Regulátor 11.1 V, 30 A, striedavý
- Servá 0.009 kg, analógové, elektrické
- Akumulátor 11.1 V, 3700mAh



Obrázok 4.1: Moja upravená verzia modelu coota

4.2 Platformy

Od platformiem som potreboval možnosť vzájomnej komunikácie, hardwarovú podporu pwm pre aspoň štyri kanály a minimálne dva senzory, trojosí gyroskop a trojosí akcelerometer. Ďalej som volil na základe ich dostupnosti, ceny, odberu elektrického prúdu a vhodnosti pre samotnú implementáciu, teda veľkosti komunity a dostupných knižníc pre rôzne funkcie. Najvhodnejšie boli výrobky od Arduino [3], Raspberry Pi [14], Adafruit [1] a nový zaujímavý Intel Edison od Intelu [10]. Nakoniec som sa rozhodol pre Arduino Leonardo [2], Raspberry Pi A+ [15] a Arduino modul 10 Degrees of freedom [7]. Ako vývojové prostredie som použil upravený Sublime Text 3 [18]. Pre zjednodušenie vývoja som používal verzovací systém Git [6].

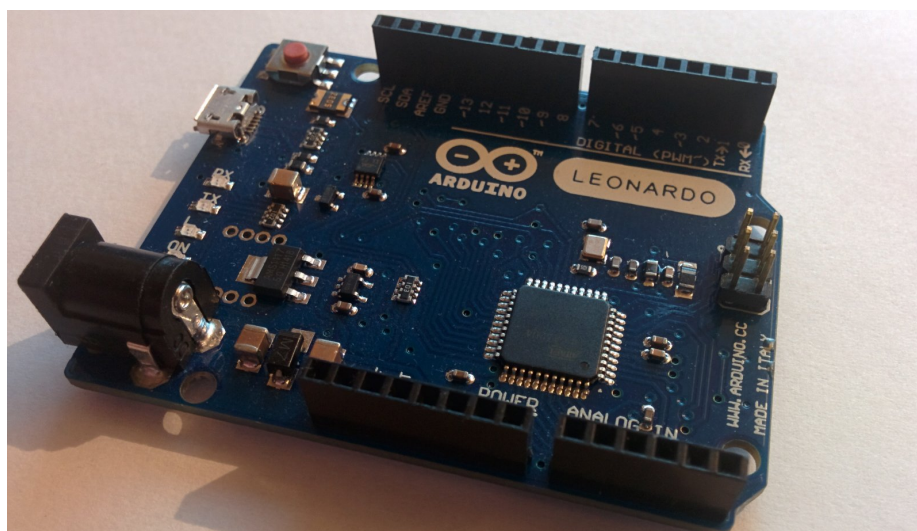
4.2.1 Arduino Leonardo

Parametre:

- Mikrokontrolér ATmega32u4
- Vstupné napätie 7-12 V
- PWM kanály 7
- Pamäť FLASH 32 KB
- Frekvencia 16 MHz

Arduino je otvorená elektronická platforma založená na ľahkom použití hardwaru a softwaru [3]. Arduino Leonardo som vybral, pretože má sedem hardwarovo riešených pwm kanálov s ovládacím rozhraním pre zjednodušenie a urýchlenie vývoja. Taktiež podporuje komunikačné rozhrania I2C, SPI a USB, ktorých rozhrania sa takisto nachádzajú v knižniciach. Keďže sa jedná o dosku z mikrokontrolérom, k vývoju je nutné program preložiť na túto architektúru a následne ho nahráť na mikrokontrolér. Z tohto dôvodu som použil

Arduino IDE, ktoré ma všetky potrebné programy zabudované [5]. Toto vývojové prostredie je vhodné pre malý projekt. Túto skutočnosť som zistil pri kompilovaní programu, keď som skúšal program z viacerými modulmi. Pridávanie modulov je priveľmi komplikované a samotný editor je veľmi jednoduchý. Tento problém rieši PlatformIO [31]. Jedná sa o multiplatformový kompilátor a manažér knižníc, ktorý pre svoju funkciu vyžaduje iba python. To rieši aj presúvanie projektov na rôzne počítače. Je možnosť ho zabudovať do vývojového prostredia alebo používať z príkazového riadka. Podporuje vyše sto dosiek a medzi nimi aj Arduino Leonardo.



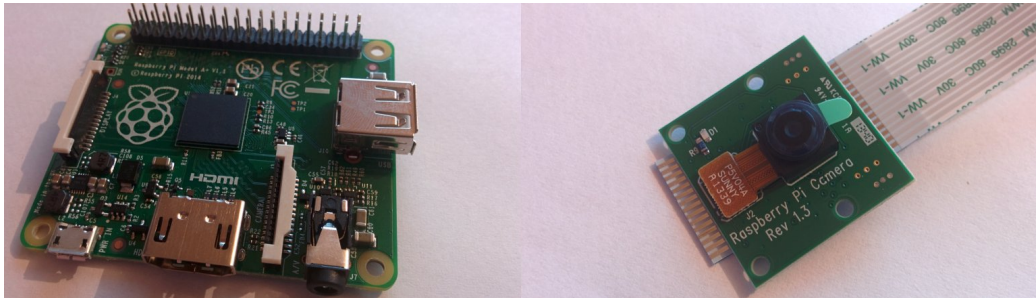
Obrázok 4.2: Arduino Leonardo

4.2.2 Raspberry Pi A+

Parametre:

- Čip BCM2835
- Procesor ARM1176JZ-F
- USB port 2
- HDMI port 1

Raspberry Pi poskytne dostatočný výpočtový výkon pre stabilizačné metódy a riadenie drona. Existuje niekoľko ďalších verzií, ktoré majú napríklad viac usb slotov alebo sieťovú kartu, ale ich energetická náročnosť je priveľmi drahá [16]. Napriek tomu aj tento model má možnosť rozšírenia o modul s Full HD kamerou s podporou H.264, ktorý sa môže neskôr zísť pri testovaní a ďalšom vývoji. V podstate je to veľmi malý počítač s operačným systémom. Ako operačný systém som použil upravenú linuxovú distribúciu Raspbian z operačného systému Debian [20].



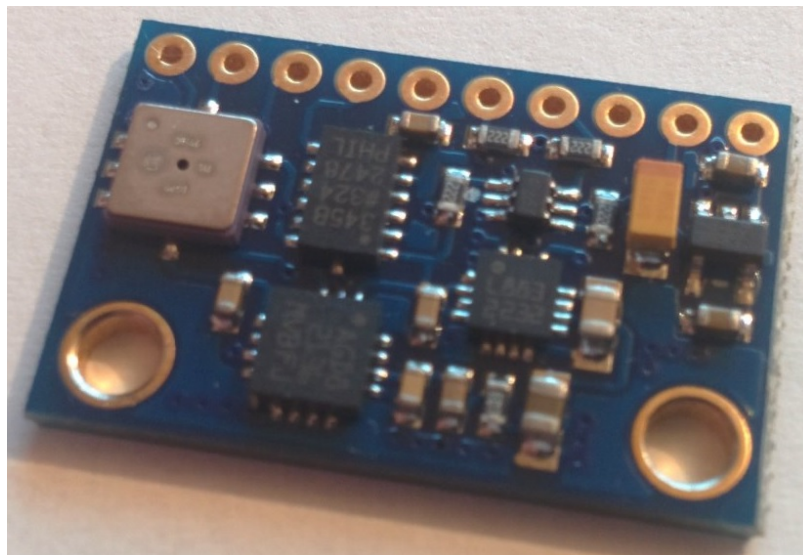
Obrázok 4.3: Raspberry Pi A+ (vľavo) a Raspberry kamera modul (vpravo)

4.2.3 10 Degrees of freedom

Parametre:

- Gyroskop L3G4200D, 3-osí
- Akcelerometer ADXL345, 3-osí
- Kompas HMC5883L, 3-osí
- Barometer BMP085
- Rozhrania SPI, I2C

Modul predstavuje kompaktné riešenie štyroch senzorov na malej doske vhodnej pre moje riešenie. Preklad modulu znamená desať stupňov slobody indikujúci desať rôznych veličín, ktoré môže merať. So senzormi sa dá komunikovať cez komunikačné rozhrania I2C alebo SPI.



Obrázok 4.4: 10 degrees of freedom

4.3 Komunikačné rozhrania

Rozhranie I2C je typu master-slave. Rozhranie I2C vyžaduje pre komunikáciu dva vodiče, ale podporuje až tisíc osem slave zariadení. Taktiež podporuje multi-master systém, no zariadenia riadiace komunikáciu v tom prípade nesmú spolu komunikovať a musia sa striedať. Väčšina zariadení vie komunikovať na frekvencii 100 alebo 400 kHz [9].

Univerzálna seriálová zbernica, ďalej USB, patrí dnes medzi najpoužívanejšie rozhrania. Toto rozhranie umožňuje prenos dát medzi dvoma zariadeniami a napájanie jedného zariadenia druhým.

Kapitola 5

Implementácia

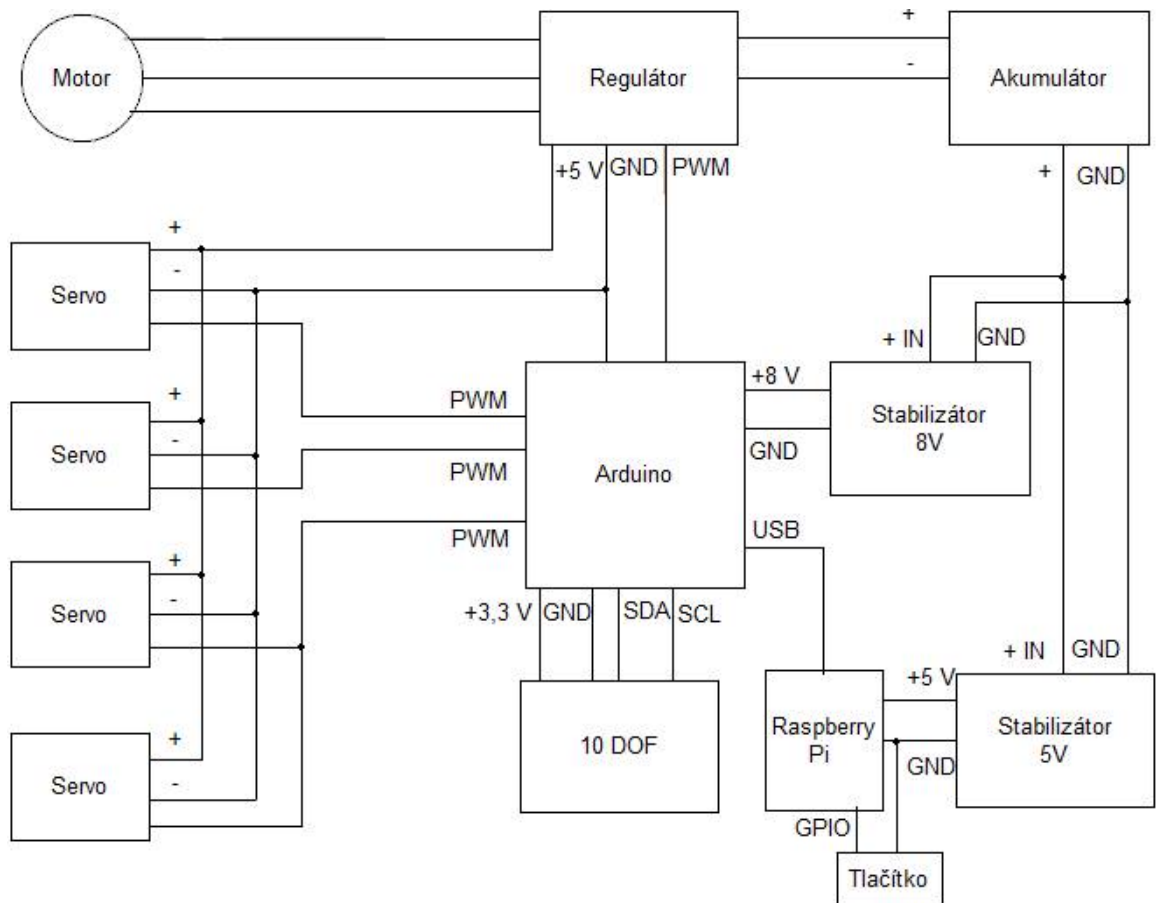
V implementačnej časti som riešil návrh jednotlivých modulov, ich komunikáciu medzi sebou a napájanie počas letu.

Tak isto som potreboval ovládať ovládacie plochy lietadla, ktoré fungujú na princípe elektro serva a to je tiahom spojené s príslušnou ovládacou plochou. Ďalším problémom bol striedavý synchronný elektromotor, ktorý je riadený súčiastkou nazývanou regulátor. Sensory ako gyroskop a akcelerometer sú samozrejme nevyhnutné pre stabilizáciu a let. Tieto problémy som riešil v module Arduino Leonardo.

Riadenie a stabilizácia letu pripadla na modul Raspberry Pi. Tento modul je myslený ako rozhodovací modul.

5.1 Elektrické napájanie a prepojenie modulov

Pri návrhu obvodu som vychádzal z pevne zadaného zapojenia akumulátor-regulátor-motor. Regulátor bol asi najväčším problémom, pretože nemal uvedené maximálne napätie. Po dlhom zisťovaní som sa dozvedel, že v modelárstve lietadiel sa používa na označenie tejto veličiny napríklad tri LI-PO. To znamená že regulátor je určený pre trojčlánkový LI-PO akumulátor. Takto sa dá vypočítať pre rôzne súčiastky, aké je ich maximálne vstupné napätie. $V = \text{počet článkov} * \text{napätie jedného článku}$. Tieto hodnoty sa musia zhodovať inak hrozí poškodenie niektorých súčiastok. Po tomto zistení som mal základ obvodu. Akumulátor dodáva jednosmerný prúd do regulátora, ktorý ho následne mení na striedavý a napája striedavý synchronný elektromotor. Z akumulátora sú ďalej vyvedené vodiče na dva stabilizátory. Stabilizátor stabilizuje napätie na požadované napätie dvoch modulov. Prvý osem voltový stabilizátor je pre napájanie Arduina Leonarda. Druhý päť voltový je určený Raspberry Pi. Z regulátora sú napájané servá, ktorý má pre túto funkciu špeciálny obvod. Zo serv a regulátoru sú vyvedené PWM vodiče pre ovládací PWM signál, ktoré sú zapojené na príslušné PWM piny na Arduine Leonarde. Arduino napája modul 10 DOF. Taktiež s ním komunikuje pomocou rozhrania I2C cez dva vodiče, datový a hodinový. Raspberry pi a Arduino Leonardo komunikujú pomocou USB. Arduino Leonardo môže byť napájané cez USB, čo mohlo spôsobiť problém, ale má zabudovaný systém na prepínanie medzi USB a napájacím pinom. Ak je vstupné napätie v doporučenom rozmedzí, napája sa z napájacieho pinu inak sa prepne na napájanie z USB. Raspberry pi má ešte k dispozícii tlačidlo pre signalizovanie spustenia predprogramovaného letu.



Obrázok 5.1: Systém zapojenia celého systému

5.1.1 Protokol

Pre komunikáciu medzi Arduino Leonardo a Raspberry Pi som vytvoril komunikačný protokol. Tento systém som navrhoval tak aby bol jednoducho rozširovateľný a stále jednoduchý. Pre implementáciu v Arduino Leonardo som použil Arduino knižnicu Serial [4] a pre Raspberry Pi knižnicu PySerial [19]. Počas implementácie som narazil len na jeden problém. Arduino Leonardo pracovalo v malej endianite a Raspberry Pi vo veľkej endianite. To som vyriešil konverziou na komunikačnej strane Raspberry Pi knižnicou struct [17]. Komunikácia prebieha po bajtoch. Niekoľko bajtov idúcich za sebou vytvára paket. Na oboch moduloch je riadiacim systémom seriálový modul, ktorý dekoduje prichádzajúce pakety a kóduje pakety, ktoré chce odoslať.

1. Prvý bajt označuje číslo paketu. Každý modul má uložené číslo odoslaného paketu a prijatého. Týmto spôsobom je možné detekovať stratu paketu a chybu v komunikácii. Toto riešenie nepočíta s možnosťou, že nastane stav, keď sa stratí $k * 256$ paketov.
2. Druhý bajt určuje typ paketu. Napríklad chyba, reštart komunikácie alebo príkaz pre modul.
3. Tretí bajt je akcia. Tento bajt príde, keď sa jedná o požiadavku na modul. Je to informácia, čo sa bude od daného modulu chcieť. Za tretím bajtom prídu ostatné

bajty ak sú potrebné pre vykonanie akcie, ktoré sa dekodujú do potrebných datových typov a akcia sa vykoná.

Číslo paketu	Typ správy	Akcia	Dáta
--------------	------------	-------	------

Obrázok 5.2: Grafické znázornenie paketu

5.2 Arduino Leonardo

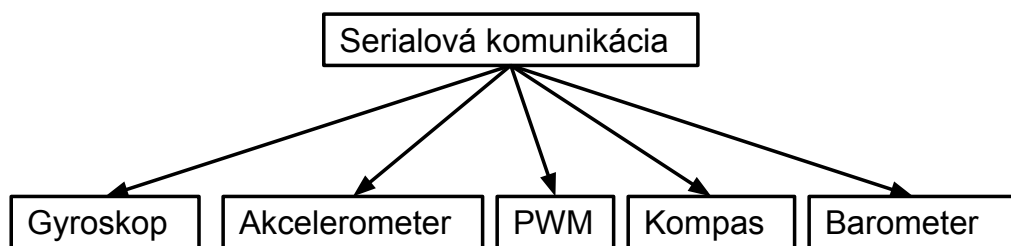
Servo riadiace natočenie ovládacích plôch, podľa informácii zistených u skúsených modelárov je riadené PWM. PWM alebo impulzová šírková modulácia sa používa pri servách na určenie uhla v akom majú byť natočené. PWM signál má periódu a striedu. Signál môže nadobúdať nula alebo päť voltov, teda logických hodnôt nula a jednotka. Strieda sa udáva v percentách a určuje koľko percent času periódy bola nastavená logická jednotka a teda dĺžka pulzu. Po bližšom preštudovaní Leonarda som zistil, že mám k dispozícii sedem, osem bitových pwm kanálov a všetky generované hardwarovo. To znamená, že striedu som mohol nastavovať na 256 hodnôt. Pre servo by to znamenalo citlivosť na menej ako stupeň, čo nebolo úplne ideálne ale dostačujúce. Pri prvom testovaní som zistil, že moja predstava ako funguje servo nie je správna. Preto som servo rozobral a zistil, že servo nemôže byť riadené striedou, ale je riadené reguláciou elektrického výkonu, a k tomu sa PWM dá využiť tiež. Zistil som, že existujú digitálne a analógové servá. Digitálne servá sú programovateľné, výkonnejšie, majú rýchlejšiu reakciu, energeticky náročnejšie a riadia sa striedou. Analógové servá napriek tomu fungujú jednotne. Podľa knižnice od Arduina pre ovládanie serv[11], ktorou som chcel teda nahradiť môj pôvodný nápad, som pochopil presnú funkciu PWM signálu z hľadiska serv ale takisto som si hneď všimol, že knižnica generuje PWM softwarovo pomocou prerušení. Tento prístup je v poriadku, pokiaľ nezáleží na rýchlosti odzvy systému, ale to nie je môj prípad. Knižnicu som teda poriadne preštudoval a narazil na ďalší problém. Pre reguláciu analógového serva PWM signálom je potrebné nastaviť periódu na 20 milisekúnd a uhol natočenia sa určuje pomocou pulzu, ktorý musí byť dlhý minimálne 544 mikrosekúnd a 2400 mikrosekúnd. Ja som mal k dispozícii osem bitové PWM kanály, takže som mohol nastaviť po jednoduchom výpočte 23 fyzických pozícií na serve. Vzhľadom na fakt, že servá v modeloch nevyužívajú celý svoj nastaviteľný rozsah, z čoho pre mňa vyplynulo ešte menej polôh, plánované riešenie nebolo možné. Po preštudovaní mikrokontroléru na Leonarde som zistil, že Arduino nevyužíva ani z ďaleka jeho potenciál [24]. Mikrokontrolér má šesť šesťnásť bitových hardvérovo riešených PWM pinov, ale iba štyri sú vyvedené na dosku Leonardo. Podporuje niekoľko druhov impulzovej šírkovej modulácie, pre rôzne zariadenia. Ja som vybral takzvané rýchle PWM a keďže štyri pwm bol presný počet, ktorý je potrebný pre ovládanie lietadla, a podľa vzorca 5.1 z dokumentácie mikrokontroléra, kde f_{clk} je frekvencia mikrokontroléra, N je delička hodinového signálu a f je požadovaná frekvencia, po dosadení som zistil, že v mojom prípade môže nastavovať šírku pulzu na presnosť pol mikrosekundy, čím som získal pre servo tritisíc sedemsto dvanásť možných nastavení natočenia. Táto presnosť presahuje dokonca citlivosť samotných serv. Rozhodol som sa teda implementovať modul pre ovládanie serv. Modul som sa snažil vytvoriť rozširovateľný, a tak aby ho mohol riadiť systém komunikujúci na USB s Raspberry

Pi.

$$f = \frac{f_{clk}}{N(1 + TOP)} \quad (5.1)$$

Striedavý motor potrebuje regulátor, ktorý je takisto riadený PWM signálom. Za celú dobu práce sa mi nepodarilo odhaliť ako regulátory fungujú. Podľa modelárov je nutné regulátor skalibrovať. Keďže dokumentácia k tejto súčiastke neexistuje a v bežných modeloch s regulátorom komunikuje vysielateľ, pomalými testami sa mi podarilo zistiť, že regulátor sleduje periódu a striedu. Pre nakalibrovanie je nutné po zapojení do napájania byť v takzvaných bezpečných hodnotách. Napríklad držať periódu dvadsať milisekúnd a impulz ako pri serve minimálna hodnota na päťsto štyridsať štyri mikrosekúnd. Keď regulátor zapípa je skalibrovaný a pripravený. Ak pípa pravidelne je chybový stav. Zvyšovaním dĺžky impulzu sa zvyšuje aj rýchlosť otáčok a teda výkon motora. Upravil som pre serva implementovaný PWM modul, tak aby vedel riadiť aj regulátor. Regulátor je riadený v percentách, definujúcich veľkosť výkonu a otáčok motora, a servo v uhloch, definujúcich uhol natočenia serva.

Modul so senzormi obsahuje gyroskop [34], akcelerometer [25], kompas [28] a barometer [32]. Implementoval som všetky moduly ale na riadenie lietadla som potreboval minimálne gyroskop a akcelerometer, preto som sa zamerlal hlavne na ne. Mój gyroskop meria uhlovú rýchlosť v troch osiach a akcelerometer zrýchlenie takisto v troch osiach. Arduino má za úlohu tieto zariadenia inicializovať, riadiť a zbierať dáta. S týmto modulom senzorov komunikuje cez rozhranie I2C. Vďaka Arduino I2C komunikáciu nebolo treba študovať do hĺbky, ale použiť knižnicu Wire [21]. Pri testovaní mi neustále knižnica vracala nesprávne dáta a chybové stavy, ktoré nedávali zmysel. Aby som vyšetril chybu, musel som prejsť krok po kroku celú knižnicu, aby som si overil, že tam nie je chyba. Po prejdení tejto knižnice som došiel k záveru, že problém je v samotnom module so senzormi. Po bližšom preskúmaní som zistil, že nožička jedného pinu nie je v kontakte. Po vyriešení tohoto problému senzory nabehli správne a začali snímať dáta. Seriálová komunikácia s Raspberry Pi sa neustále pýta na nové dáta. Pokiaľ v senzorech sú nové dáta, okamžite sa pošlú Raspberry Pi. Pre moju prácu som zvolil rýchlosť snímania dát 100 Hz. To znamená, že každých desať milisekún sa odošlú ďalšie dáta.

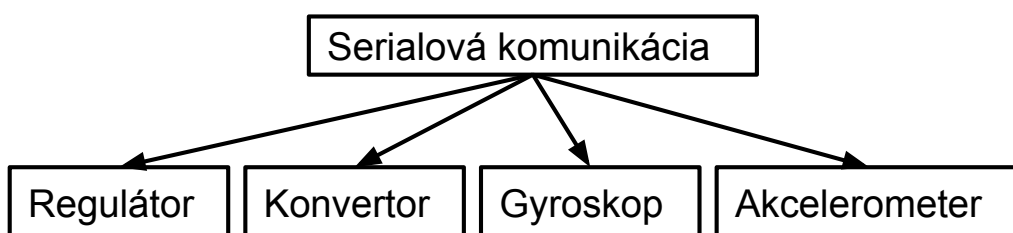


Obrázok 5.3: Grafické znázornenie tried pre Arduino Leonardo

5.3 Raspberry Pi

Tak isto ako pri Arduino Leonardo je centrom systému seriálová komunikácia na USB 5.3. Raspberry Pi môže vydávať príkazy pre PWM a nastavovať tak výchylky serv. Uhly serv, ktoré je možné nastaviť neodpovedajú výchylkám ovládacích plôch, a preto som vytvoril konvertor, ktorého úlohou je tento problém riešiť. Raspberry Pi neustále čaká na dáta

zo senzorov. Tieto chodia v 10 milisekundových intervaloch. Trieda gyroskop dostáva zo skutočného gyroskopu uhlovú rýchlosť a ukladá údaje o uhlovom zrýchlení, uhlovej rýchlosti a aktuálnom uhle natočenia. Neznáme údaje sa dopočítavajú pomocou numerického derivovania a integrovania. Trieda akcelerometer dostáva pravidelne údaje zo skutočného akcelerometru o aktuálnom zrýchlení. Podobne si dopočíta ostatné informácie o aktuálnej rýchlosti a pozícii. Zo zrýchlenia je nutné odstrániť zložku gravitačnú. Pre túto funkciu je potrebné použiť transformáciu zložiek vektora z jednej súradnicovej sústavy do druhej pri ich vzájomnom natočení, preto v tomto mieste aplikujem eulorových kinematických rovníc 2.2.3. V hlavnej smyčke nebeží len komunikácia, ale aj regulátory, ktoré čítajú aktuálne hodnoty uložené v triedach Gyroskop a Akcelerometer. Potom podľa ich aktuálneho nastavenia, teda konštánt proporčionalnej, sumačnej a diferenciálnej zložky, riadia PWM a v konečnom dôsledku ovládacie plochy lietadla, teda lietadlo samotné. Všetky úkony a dáta sa zakaždým ukládajú do pamäte pre potreby analýzy letu.



Obrázok 5.4: Grafické znázornenie tried pre Raspberry Pi

Kapitola 6

Testovanie

Ako prvé som testoval elektrické napájanie. Po zmeraní všetkých prístrojov som zistil, že obvod funguje bez problémov. Ďalej som testoval ovládacie plochy lietadla a elektromotor. Pri plnej záťaži a zmeraní odberu prúdu multimetrom som zistil, že vďaka kapacite akumulátora je možné letieť okolo 20 minút. Sensory som testoval mimo lietadla a následne v lietadle. Pri montáži do lietadla sa pravdepodobne mechanicky poškodili, pretože pri inicializácii prestávali komunikovať s Arduino. Sensory som nemal ako nahradiť a bez nich nebolo možné pokračovať v testovaní. V prílohe [A](#) je súbor s poslednými záznamami komunikácie a senzormi nameranými hodnotami z vývoja a testovania.

Kapitola 7

Záver

Pri vypracovávaní práce som sa zameril hlavne na praktickú časť, ktorá na koniec skončila neotestovaná. Zo začiatku zadanie vyzeralo triviálne, no postupom času ako som sa dostával čoraz hlbšie do problematiky spojenej s riadením a stabilizáciou lietadiel zistil som, že tomu tak nie je a v práci som sa začal zamotávať. Niektoré problémy, ktoré som si myslel, že som už vyriešil sa znovu objavovali. Ku koncu práce sa mi aj napriek ťažkým problémom podarilo dostať skoro do cieľa. Kvôli nedostatku času sa mi nepodarilo venovať písomnej časti práce toľko, koľko by zasluhovala vzhľadom na čas strávený nad praktickou časťou. Vyriešil som mnohé problémy a získal veľa užitočných skúseností a vedomostí.

7.1 Výsledky testovania

Výsledky testovania boli pozitívne. Navrhnutý obvod funguje správne. Systém ovládania lietadla, teda jeho ovládacích plôch ide plynulo. Regulácia výkonu motora je takisto funkčná. Gyroskop a akcelerometer poskytujú informácie o natočení, rýchlosti, pozícii a ďalšie dopyčované hodnoty, ktoré sú už menej presné kvôli chybám numerických metód a pravidelnému pripočítavaniu chýb.

7.2 Zvolené riešenie

Zvolené riešenie nebolo zlé, ale vzhľadom na náročnosť projektu nebolo vhodné. Arduino Leonardo je platforma pre jednoduché projekty a práca s ním je tiež jednoduchá. Pre moje riešenie som očakával viac. Spoliehal som sa na rôzne knižnice, ale nakoniec som väčšinu práce venoval písaniu vlastných a ich ladením. Celý systém riešenia som veľmi skomplikoval práve vložением Arduino Leonardo medzi Raspberry a senzory. Oveľa vhodnejšie by bolo napojiť senzory priamo na Raspberry Pi a použiť platformu určenú len pre PWM, ktorá by sa tak isto napojila na Raspberry Pi napríklad cez I2C. Tým by sa odstránila zbytočná práca na komunikácii.

7.3 Rozvoj

Ďalším krokom projektu by bola zmena celého systému a zmena niektorých platforiem. Určite je nutné rozšíriť a viac prepracovať knižnice pracujúce so senzormi. Senzory nedávali veľmi presné hodnoty a narastajúce chyby spôsobené sčítaním sú nepoužiteľné pre dlhší let. Pre pokračovanie projektu správnym smerom je nutné pridať ďalšie rôzne senzory a zlepšiť

kvalitu údajov z momentálnych, napríklad pridaním filtrov. Napríklad GPS, ktorou sa môže chyba v pozícii lietadla periodicky opravovať. Senzor pre meranie rýchlosti lietadla proti vetru je v podstate nevyhnutný pre získanie reálnej rýchlosti lietadla, pretože akcelerometer funguje iba keď na lietadlo nepôsobia vonkajšie vplyvy ako napríklad vietor.

Literatúra

- [1] Adafruit Industries. Květen 2015.
URL <https://www.adafruit.com/>
- [2] Arduino - ArduinoBoardLeonardo. Květen 2015.
URL <http://www.arduino.cc/en/Main/ArduinoBoardLeonardo>
- [3] Arduino - Home. Květen 2015.
URL <http://www.arduino.cc/>
- [4] Arduino - Serial. Květen 2015.
URL <http://www.arduino.cc/en/reference/serial>
- [5] Arduino Software. Květen 2015.
URL <http://www.arduino.cc/en/Main/Software>
- [6] Git. Květen 2015.
URL <https://git-scm.com/>
- [7] GY-80 - Multi Sensor Board - 3 Axis Gyro -3 Axis Accelerometer - 3 Axis Magnetometer - Barometer - Thermometer. Květen 2015.
URL <http://selfbuilt.net/shop/gy-80-inertial-management-unit>
- [8] HobbyKing Coota R/C Sea Plane Plug-n-Fly. Květen 2015.
URL <http://www.rcgroups.com/forums/showthread.php?t=1371166>
- [9] I2C. Květen 2015.
URL <https://learn.sparkfun.com/tutorials/i2c>
- [10] Intel Edison. Květen 2015.
URL <https://www.sparkfun.com/products/13024>
- [11] LinkIt-ONE-IDE/Servo.h at master - Seeed-Studio/LinkIt-ONE-IDE - GitHub. Květen 2015.
URL <https://github.com/Seeed-Studio/LinkIt-ONE-IDE/blob/master/hardware/arduino/mtk/libraries/Servo/Servo.h>
- [12] MATLAB - The Language of Technical Computing. Květen 2015.
URL <http://www.mathworks.com/products/matlab/>
- [13] Phantom 2 Vision - Your Flying Camera, Quadcopter Drone for Aerial Photography and Videography. Květen 2015.
URL <http://www.dji.com/product/phantom-2-vision/>

- [14] Raspberry Pi - Teach, Learn, and Make with Raspberry Pi. Květen 2015.
URL <https://www.raspberrypi.org/>
- [15] RASPBERRY PI 1 MODEL A+. Květen 2015.
URL <https://www.raspberrypi.org/products/model-a-plus/>
- [16] Raspberry Pi2 - Power and Performance Measurement. Květen 2015.
URL <http://raspi.tv/2015/raspberry-pi2-power-and-performance-measurement>
- [17] struct - Interpret strings as packed binary data. Květen 2015.
URL <https://docs.python.org/2/library/struct.html>
- [18] Sublime Text 3. Květen 2015.
URL <http://www.sublimetext.com/3>
- [19] Welcome to pySerial s documentation. Květen 2015.
URL <http://pyserial.sourceforge.net/>
- [20] Welcome to Raspbian. Květen 2015.
URL <https://www.raspbian.org/>
- [21] Wire Library. Květen 2015.
URL <http://www.arduino.cc/en/reference/Wire>
- [22] Blackburns, A.: Hybrid power helps drones fly longer. Květen 2015.
URL <http://interestingengineering.com/hybrid-power-helps-drones-fly-longer/>
- [23] Cole, C.; Wright, J.: What are drones? Květen 2015.
URL <http://dronewars.net/aboutdrone/>
- [24] Corporation, A.: ATmega16U4/ATmega32U.
<http://www.atmel.com/devices/atmega32u4.aspx>, 2014.
- [25] Devices, A.: ADXL345.
<http://robotstore.cz/wp-content/uploads/2014/01/ADXL345.pdf>, 2009.
- [26] Draxler, K.; Jiráček, V.; Kulčák, L.; aj.: *AERODYNAMIKA, KONSTRUKCE A SYSTÉMY LETOUNŮ*. Část 1, Akademické nakladatelství CERM, s.r.o., Brno, 2005, ISBN 80-7204-367-6.
- [27] Etherington, D.: Canada Proves Fertile Ground For Amazon Drone Delivery Tests. Květen 2015.
URL <http://techcrunch.com/2015/03/30/canada-proves-fertile-ground-for-amazon-drone-delivery-tests/>
- [28] Honeywell: HMC5883L.
http://robotstore.cz/wp-content/uploads/2014/01/DS_IM130918001.pdf, 2013.
- [29] Hraško, A.: Řízené přistání autonomního drona. 2013.
URL <https://is.cuni.cz/webapps/zzp/download/130114927>

- [30] Humr, J.: Úsporný autopilot. 2010.
URL http://measure.feld.cvut.cz/cs/system/files/files/cs/vyuka/zaverecne_prace/DP_2011_Humr_locked.pdf
- [31] Kravets, I.: PlatformIO. Květen 2015.
URL <http://docs.platformio.org/>
- [32] Sortec, B.: Honeywell.
http://robotstore.cz/wp-content/uploads/2014/01/DS_BMP085.pdf, 2008.
- [33] Slavětínský, D.: Souřadnicové soustavy používané v letectví. Květen 2015.
URL http://www.slavetind.cz/stavba/Mechanika/Sour_soustavy.aspx/
- [34] STMicroelectronics: L3G4200D.
<http://robotstore.cz/wp-content/uploads/2014/02/L3G4200D.pdf>, 2010.
- [35] Teichl, J.: *ZÁKLADY AERODYNAMIKY*. Akademické nakladatelství CERM, s.r.o., Brno, 2003, ISBN 80-7204-316-1.
- [36] Švarc, I.: *ZÁKLADY AUTOMATIZACE*. Květen 2015.
URL <http://autnt.fme.vutbr.cz/svarc/ZakladyAutomatizace.pdf>
- [37] Yoon, J.: Origins of Control Surfaces. Květen 2015.
URL <http://www.aerospaceweb.org/question/history/q0103.shtml>

Príloha A

Obsah CD

- Súbor **result.log** obsahuje posledné záznamy komunikácie a senzormi nameranými hodnotami.
- Súbor **projekt.pdf** obsahuje elektronickú pdf verziu tejto práce.
- Priečink **source** obsahuje zdrojové kódy.
 - Priečink **Arduino** obsahuje zdrojové kódy pre Arduino a platformIO.
 - Priečink **Raspberry Pi** obsahuje zdrojové kódy pre Raspberry Pi.
- Priečink **simulator** obsahuje skript v matlabe pre simuláciu regulátora priečnej stabilizácie lietadla.