VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

BEZPEČNÉ ŘÍZENÍ LETU DRONA S VYUŽITÍM MĚŘENÍ ZE SONARŮ

SAFE FLIGHT-CONTROL OF DRONE BASED ON SONAR MEASUREMENTS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE KRISZTIÁN BENKŐ
AUTHOR

VEDOUCÍ PRÁCE Ing. VÍTĚZSLAV BERAN, Ph.D.
SUPERVISOR

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Benkő Krisztián**
Obor: Informační technologie
Téma: **Bezpečné řízení letu drona s využitím měření ze sonarů**
Safe Flight-Control of Drone Based on Sonar Measurements
Kategorie: Uživatelská rozhraní

Pokyny:

1. Prostudujte problematiku uživatelských rozhraní pro řízení bezpilotních letounů. Seznamte se s postupy měření a modelování prostředí z hloubkových dat a řízením drona.
2. Navrhněte systém, který bude s využitím měření sadou sonarů a dalších zařízení rekonstruovat velmi hrubý model prostředí v okolí drona. Tento model využijte k návrhu nového bezpečného způsobu řízení drona ve vnitřním prostředí.
3. Navržené rozhraní implementujte s využitím moderních řídicích zařízení a existujících knihoven (ROS, PCL apod.).
4. Proveďte testy bezpečnosti a stability výsledného řešení v různých situacích. Výsledky diskutujte.
5. Vytvořte plakát a krátké demonstrační video reprezentující Vaše řešení.

Literatura:

- H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Boston, 2005
- D. Stone, C. Jarrett, M. Woodroffe, S. Minocha. *User Interface Design and Evaluation*, Morgan Kaufmann, ISBN-10: 0120884364, 2005
- Dále dle pokynu vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a částečně bod 3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Beran Vítězslav, Ing., Ph.D.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2015
Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Bakalárska práca sa zaoberá návrhom, implementáciou a testovaním protinárazového systému pre bezpečné riadenie kvadrokoptéry. Systém je implementovaný pomocou ROS frameworku, v ktorom sa spracovávajú vstupné údaje z klávesnice a sonarov, potom sa vykoná výpočet a výstupom je ovládanie motorov. Kvadrokoptéra sa dokáže bezpečne pohybovať vo vnútornom prostredí. Dôraz sa kládol na rýchlosť výpočtu pohybu a testovanie systému. Dron je možné ovládať pomocou klávesnice.

Abstract

Bachaleor's thesis describes design, implementation and testing of anti-crash system for safe flight-control of quadcopter. The system is implemented using ROS framework, where input data from sonars and keyboard are processed, calculation is then performed, output of which is controlling motors. Quadcopter is able to move safely in indoor environment. The emphasis was on speed of calculation and system testing. The quadcopter can be controlled by keyboard.

Kľúčové slová

dron, kvadrokoptéra, ROS, bezpečné riadenie, sonar, PID

Keywords

drone, quadcopter, ROS, safe-control, sonar, PID

Citácia

BENKŐ, Krisztián. *Bezpečné řízení letu drona s využitím měření ze sonarů.* Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Beran Vítězslav.

Bezpečné řízení letu drona s využitím měření ze sonarů

Prehlásenie

Prehlasujem, že som túto bakalárskú prácu vypracoval samostatne pod vedením pána Ing. Vítězslava Berana, Ph.D.. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

. . . . . .
Krisztián Benkő
17. mája 2016

Poďakovanie

Rád by som sa poďakoval vedúcemu práce pánovi Ing. Vítězslavovi Beranovi, Ph.D. za odbornú pomoc, konzultácie a cenné rady pri riešení bakalárskej práce.

© Krisztián Benkő, 2016.
Táto práca vznikla ako školské dielo na FIT VUT v Brně. Práca je chránená autorským zákonom a jej využitie bez poskytnutia oprávnenia autorom je nezákonné, s výnimkou zákonne definovaných prípadov.

Obsah

1 Úvod **2**

2 Teória **3**
2.1 AscTec Pelican . . . 3
2.2 Robotický operačný systém . . . 6
2.3 Ultrazvukové sonary . . . 7
2.4 Filtrácia hodnôt zo sonarov . . . 9
2.5 PID regulátor . . . 10
2.6 Existujúce riešenia . . . 13

3 Návrh systému **15**
3.1 Analýza zadania . . . 15
3.2 Vstup údajov do systému . . . 15
3.3 Filtrovanie hodnôt zo sonarov . . . 16
3.4 Vyhýbanie sa prekážkam . . . 16
3.5 Letový mód udržania stabilnej výšky . . . 17

4 Realizácia **21**
4.1 ROS balík $safety_flight_control$ . . . 21
4.2 Spojazdnenie kvadrokoptéry a práca so sonarmi . . . 21
4.3 PID regulátor . . . 24
4.4 Protinárazový systém . . . 24

5 Testovanie **26**
5.1 Zabezpečenie a nastavenie pred lietaním . . . 26
5.2 Testovanie protinárazového systému . . . 27
5.3 Vyhodnotenie testov . . . 30

6 Záver **32**

Literatúra **33**

Prílohy **34**
Zoznam príloh . . . 35

A Obsah CD **36**

Kapitola 1

Úvod

Cieľom tejto práce je dosiahnuť úspešnú detekciu prekážok pomocou ultrazvukových sonarov a následné úspešné vyhýbanie sa prekážkam, s čím sa zabezpečí pilotovi kvadrokoptéry bezpečné riadenie bez možnosti havárie zavinením narazenia do prekážky.

Dron je lietadlo bez posádky, ktoré môže byť ovládané autonómnym systémom pomocou nainštalovaného programu alebo môže byť ovládané na diaľku pilotom. Jedným typom dronu je kvadrokoptéra. Kvadrokoptéra je rotorový lietajúci objekt, ktorý používa dva páry opačne otáčajúcich sa rotorov, ktoré poskytujú vzlietnutie a ovládanie. Motory používajú samostatnú reguláciu rýchlosti každého rotora pre dosiahnutie kontroly nad kvadrokoptérou. Reguláciou rýchlosti každého rotora je možné špecificky generovať požadovaný pohyb.

Ovládanie kvadrokoptéry nie je zďaleka triviálne a vyžaduje to istý stupeň zručnosti. Systém bezpečného riadenia kvadrokoptéry by mal slúžiť jak pre začiatočníkov, tak aj pre pokročilých, ktorí využívajú kvadrokoptéru napríklad v tesných uzavretých priestoroch. Pre vývoj systému sa využije školský model kvadrokoptéry AscTec Pelican, na ktorom boli v predchádzajúcom projekte osadené ultrazvukové sonary. Pomocou nich sa budú získavať vzdialenosti od prekážok. Odmeraná vzdialenosť pomocou sonaru nemusí byť vždy skutočná vzdialenosť, keďže signál sa môže od daného predmetu napríklad zle odraziť alebo môže vplývať na sonar rušenie z okolia. Pre tieto prípady bude potrebné vykonať výpočty a filtráciu. V ďalšom kroku sa bude riešiť ovládanie motorov, kde sa bude brať do úvahy veľkosť a hmotnosť kvadrokoptéry pre zistenie zotrvačnej sily a potrebného priestoru. Pri implementácii sa použije robotický operačný systém, ktorý značne uľahčí prácu.

Čitateľ sa v kapitole 2 dozvie informácie o kvadrokoptére Tyra, sonaroch, ktoré boli na nej umiestnené a potrebných nástrojoch pre vytvorenie protinárazového systému. Popisujú sa aj existujúce riešenia, v ktorých sa už protinárazový systém využíva. V kapitole 3 sa dekomponuje protinárazový systém na menšie časti a vypracuje sa detailný návrh. V kapitole 4 sa nachádza štruktúra programu s popisom jednotlivých uzlov a čitateľ sa dočíta aj o problémoch pri spojazdnení Tyry. V poslednej kapotile 5 je popísaný priebeh testovania funkčnosti zabezpečeného lietania a vyhodnotenie testov.

Kapitola 2

Teória

V tejto kapitole sa čitateľ zoznámi s potrebnými informáciami pre pochopenie problematiky. Popisujú sa tu nástroje, ktoré budú využívané pri návrhu a realizácii systému na školskom modeli kvadrokoptéry, ktorá sa volá Tyra[1]. Nakoniec sa predstavia existujúce riešenia kvadrokoptér, ktoré obsahujú systém bezpečného riadenia a vyhýbanie sa prekážkam.

2.1 AscTec Pelican

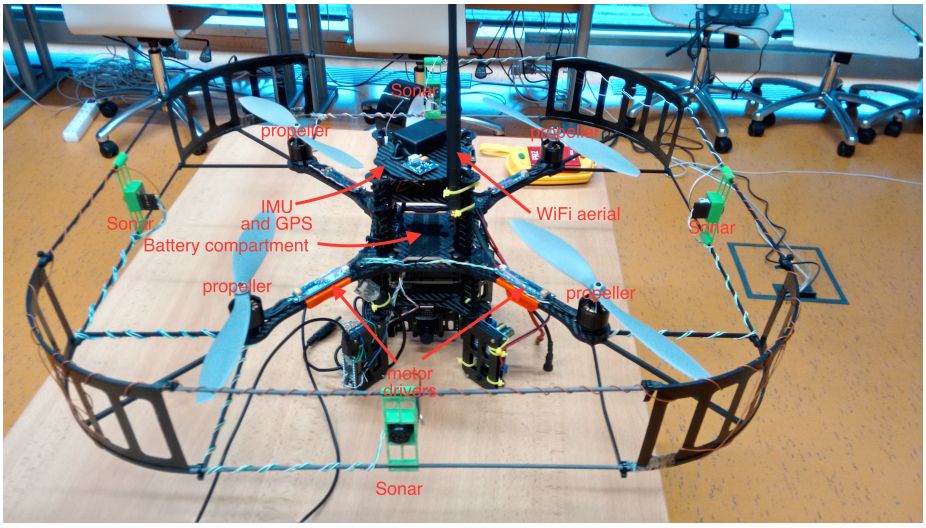
Bezpečnostný systém pre riadenie letu bude navrhnutý pre kvadrokoptéru AscTec Pelican[2]. AscTec Pelican patrí medzi drony, ktoré majú podporou pre výskum. Ponúka dostatok priestoru pre zaťaženie dronu a dostatok rozhraní pre užitočné komponenty. Keďže má AscTec Pelican modulárnu konštrukciu, tak sa rozloženie prvkov dá aj meniť. Ďalej ponúka spoľahlivosť a bezpečnosť letového robota. Tým sa javí byť tento dron ideálny pre školské projekty.

Rozloženie prvkov na Tyre (zdola hore):

- nosič kamery, ultrazvukový sonar
- Atomboard
- ramená s motormi, v strede ramien sa nachádza batéria
- vo výške vrtúľ sa nachádzajú osadené ultrazvukové sonary
- AutoPilot a riadenie napájania
- GPS modul, magnetometer, ultrazvukový sonar

Atomboard
Základná doska s operačným systémom Ubuntu 12.04 LTS, na ktorom je osadený procesor Intel Atom a je formátu Pico-ITX. Táto základná doska bola navrhnutá spoločnosťou Advantech ako najmenšia, ktorá obsahuje procesor Intel Atom a jeho rozmery sú zároveň maximálne pre osadenie na AscTec Pelican. Konkrétny model, ktorý je použitý na Tyre sa volá MIO-2261. Tento model bol uprednostnený pred modelom od firmy Ascending Technologies kvôli cene. Vlasnosťami sa takmer vyrovnáva drahšej verzii. Procesor Intel Atom je dvojjadrový a jeho spotrebou sa radí do kategórie nízkopríkonových. S pamäťou 4GB

[1] http://www.fit.vutbr.cz/research/groups/robo/eqp.php.cs
[2] http://www.asctec.de/en/uav-uas-drones-rpas-roav/asctec-pelican/



Obr. 2.1: Kvadrokoptéra Tyra[3].

DDR3 1333MHz a úložným priestorom 30GB SSD mSata má Tyra výborný výpočetný výkon a je aj komerčne dostupný. Doska umožňuje pripojiť periferné zariadenia cez rozhrania. Obsahuje napríklad USB porty, VGA výstup, Audio I/O, Wi-Fi modul, Ethernet, GPIO a I2C. Cez USB sa pripája napríklad Arduino, ktoré je potrebné pre spracovanie údajov zo sonarov. Wi-Fi modul slúži napríklad na vzdialené pripojenie k Tyre.

Význam Atomboardu spočíva vo vývoji nových funkcionalít na architektúre x86. Podporuje systém ROS, ktorý značne uľahčuje programovanie a prepájanie jednotlivých zariadení.

AutoPilot

AutoPilot sa pripája pomocou USB UART prevodníka na Atomboard. Cieľom dosky AutoPilot je udržať stabilitu drona, zjednodušiť a zabezpečiť jeho riadenie. Je to hlavný riadiaci obvod drona a obsahuje dva ARM7 procesory, ktorými sú Low Level Processor (ďalej len LLP) a High Level Processor (ďalej len HLP).

Možnosti ovládania Tyry:

- režim, v ktorom sa dron riadi povelmi z RC ovládača, ktoré sa spracujú v LLP
- režim, v ktorom sa z počítača posielajú príkazy do LLP cez sériové rozhranie
- režim, v ktorom riadi dron program vykonávaný z HLP
- režim, v ktorom povely spracováva počítač a následne ich posiela do HLP a ten ich ďalej predá do LLP

Low Level Processor

LLP zabezpečuje stabilné riadenie. Spracováva údaje zo senzorov, vykonáva fúziu údajov a spracovávanie všetkých informácií s frekvenciou aktualizácie 1kHz. Pracuje s tromi letovými módmi:

1. Manuálny mód

[3] http://merlin.fit.vutbr.cz/wiki/index.php/RoboLab_Tyra

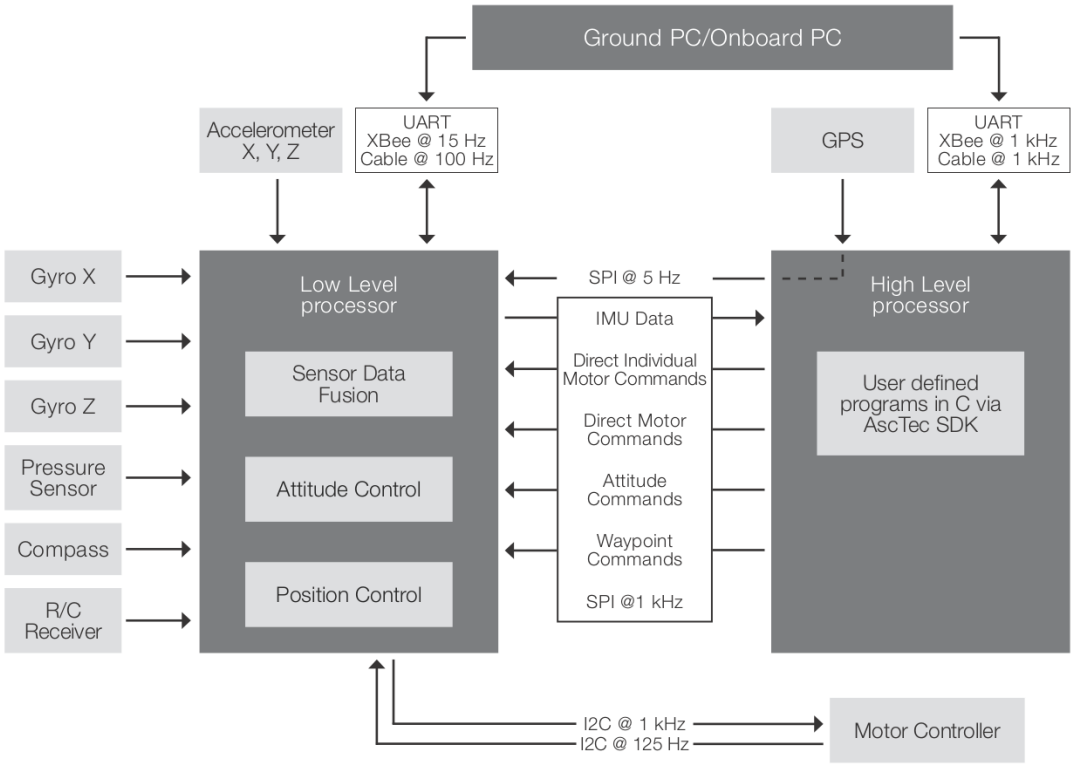
2. Height mód

3. GPS mód

Jednotlivé módy sa prepínajú na ovládači. Pri manuálnom pohybe sa aktivuje len kontrola pohybu a pokyny sa príjmajú z RC ovládača. Pri height móde (výškový mód) sa aj kontrola výšky aktivuje a ide o vypočítanie letovej výšky a udržanie tejto výšky. Pri navigácii pomocou GPS súradníc sa vďaka GPS modulu môže kotrolovať aj pozícia. V prípade, keď Tyra stratí signál alebo sa dostane do kritickej situácie, tak sa zapne záchranný mód, ktorý prevezme riadenie nad dronom. Možností riešenia kritických situácií je niekoľko. Napríklad v prípade straty spojenia dron zahájí klesanie 1 m/s a pristane. V GPS móde sa počká 5 sekúnd na aktuálnu pozíciu, a až potom začne klesať. Ďalšia záchranná procedúra vráti dron na miesto vzletu.

High Level Processor

HLP umožňuje ovládať systém pomocou kódu. Tento kód môže byť užívateľsky nahraný. Pri poruche môže užívateľ hocikedy prepnúť stav ovládania na LLP, kde sa náchadzajú overené algoritmy a teda funguje to ako zabezpečovací systém pri testovaní napríklad. Procesory LL a HL komunikujú na frekvencii 1kHz, čo je dostatočne rýchle na to, aby mohli byť poskytnuté všetky informácie zo senzorov v HLP. Takisto sa môžu predávať informácie do LLP, čo sa týka ovládania napríklad. HLP umožňuje pripojiť ďalšie zariadenia cez svoje vlastné rozhrania a takisto ich dokáže aj napájať s 5V alebo 12V.



Obr. 2.2: Schéma dosky Autopilot a Atomboard[4].

[4] http://wiki.asctec.de/display/AR/AscTec+AutoPilot

2.2 Robotický operačný systém

Robotický operačný systém (ďalej len ROS) predstavuje otvorený (open source) systém, ktorý zahŕňa množstvo nástrojov a balíčkov. Je štrukturalizovaný tak, aby čo najviac zefektívnil programovanie a výslednú činnosť daného robota, prípadne nejakého motorického alebo senzorického subsystému. Vytvoriť stabilný a odolný všeobecný softvér na programovanie a riadenie robotov je skutočne ťažké. ROS toto všetko poskytuje a takisto poskytuje aj simulátor a nástroje pre vizualizáciu. Existujú dve hlavné grafické prostredia - Rviz a Gazebo, kde možno ľahko simulovať pohyby robotov, správanie senzorov alebo ovládať jednotlivé, používateľom naprogramované motorické alebo senzorické subsystémy. Cieľom ROSu je zdieľanie a znovupoužiteľnosť zdrojového kódu, všetky v ňom vytvorené knižnice a balíčky sú priamo dostupné na webových stránkach ROSu. Komunikácia v systéme ROS prebieha v podobe peer-to-peer siete, kde procesy predstavujú uzly (anglicky node), ktoré medzi sebou komunikujú posielaním správ (anglicky message) cez témy (anglicky topic) [4].

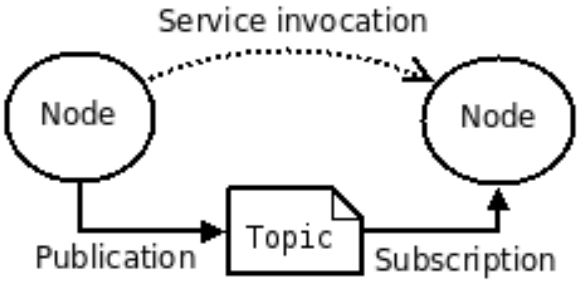
Node
- je proces, ktorý vykonáva výpočet. Uzly sú spojené do grafu a komunikujú navzájom pomocou tém. Robotický systém zvyčajne pozostáva z viacerých uzlov, čo zabezpečuje kontrolu robota. Uzol Master zabezpečuje komunikáciu medzi dvoma uzlami.

Message
- uzly komunikujú medzi sebou pomocou publikovaním správ pre posielanie a odoberaním správ pre príjem. Pri implementácii správy predstavujú údajové štruktúry.

Topic
- témy sú pomenované zbernice, cez ktoré uzly vymieňajú správy. Uzly publikujú na dané témy a pri odbere sa potrebujú zaregistrovať pre odber správ.

Služby (anglicky services)
- v prípade, keď nestačí jednosmerná komunikácia sa využíva komunikácia typu otázka-odpoveď. Služby sú pomenované.

Komunikácia medzi uzlami a témami je vysvetlená na obrázku 2.3. Bodkovaná čiara predstavuje inkováciu služby. Plná čiara smerujúca z Node do Topic značí publikáciu na túto tému a plná čiara smerujúca z Topic do Node značí odoberanie správ uzlom z témy.



Obr. 2.3: Princíp komunikácie uzlov v ROS[5].

[5] http://wiki.ros.org/custom/images/wiki/ROS_basic_concepts.png

Rosbag je súčasťou systému ROS a umožňuje zaznamenávať vybrané témy do bag súboru. Následným prehrávaním tohto súboru sa simuluje stav nahranej témy a dané uzly z neho môžu odoberať správy. Slúži to najmä na testovanie, ktoré takto značne uľahčuje.

Na Tyre je nainštalovaná verzia Hydro a pomocou ROS balíkov sa posielajú príkazy do AutoPilotu. Je priamo vytvorená platforma pre komunikáciu s AscTec Pelican.

AscTec Mav Framework

AscTec Mav Framework je súbor ovládačov a ďalších nástrojov, ktoré umožňujú získavanie údajov a ovládanie dronu pomocou autopilota. Vďaka tomuto frameworku vieme HLP programovať a prepojiť s LLP.

Dôležité časti AscTec Mav Framework:

- asctec_hl_comm - balík obsahuje definíciu ROS správ
- asctec_hl_firmware - firmware pre HLP na komunikáciu s autopilotom, ktorý musí byť nahraný v HLP pre umožnenie komunikácie
- asctec_hl_interface - rozhranie pre HLP na komunikáciu s autopilotom, obsahuje skripty pre ladenie a monitorovanie
- asctec_hl_gps - balík, pomocou ktorého sa posielajú GPS súradnice

Balík asctec_hl_interface obsahuje dôležitý uzol hl_interface, ktorý pracuje s HLP a ponúka nám tieto témy:

- fcu/control - príjma príkazy pre ovládanie
- fcu/pose - príjma informácie o pozícii drona
- fcu/state - príjma informácie o stave drona
- fcu/ekf_state_in - príjma aktualizácie od ethzasl_sensor_fusion[6]

2.3 Ultrazvukové sonary

Ultrazvukový sonar pracuje na princípe generovania vysokofrekvenčných zvukových vĺn, ktoré sa po odraze od prekážky vracajú späť do sonaru. Vyhodnocovacia elektronika senzora vypočíta časový interval medzi vysielaným a prijímaným signálom, čím vie určiť vzdialenosť od objektu. Táto technológia sa využíva v ultrazvukových sonaroch, ktoré boli implementované na Tyre.

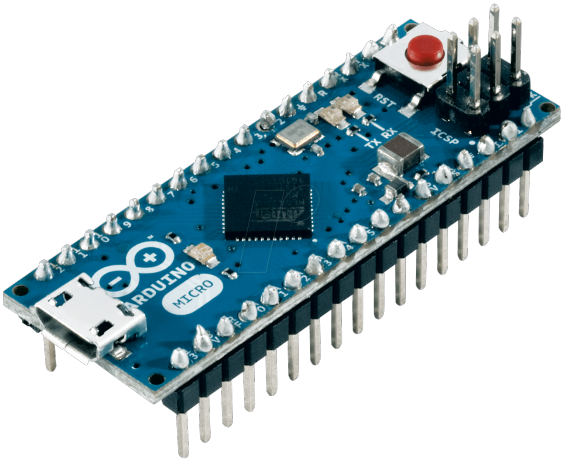
Konkrétny vybraný typ sonarov sa volá MB1220 XL[7]. Tento typ sonarov je vhodný na vnútorné aj vonkajšie merania a predmety rozoznáva v rozsahu 20 až 765 cm. Nameraná hodnota sa dá odmerať meraním dĺžky impulzu alebo hodnoty napätia. Ďalšou dôležitou vlastnosťou sonarov pre tento projekt je, že jeden sonar vykoná desať meraní za sekundu. So sonarmi sa pracuje pomocou mikrokontroléru Arduino Micro 2.4, kde je pre možnosti pripojenia sonarov 12 digitálnych pinov a 6 analógových pinov. To pre zapojenie 14 sonarov nestačí, keďže každý sonar potrebuje 2 piny. Zaviedlo sa teda riešenie, že sonary v rohoch dronu budú zapojené zreťazene a sonary na bokoch, hore a dole budú zapojené jednotlivo,

[6] http://wiki.ros.org/ethzasl_sensor_fusion
[7] http://www.maxbotix.com/documents/XL-MaxSonar-EZ_Datasheet.pdf

keďže je potrebná vyššia frekvencia meraní a navzájom sa zvukové vlny sonarov nerušia [5]. Sonary zapojené zreťazene sa prepínajú multiplexovo - pri dokončení merania na jednom sonare sa vygeneruje štartovací impulz pre meranie na ďaľšom sonare, takto sa zabezpečí to aby sa navzájom nerušili ani sonary zapojené digitálne. Toto ovplyvnilo frekvenciu meraní, ktorá sa značne znížila. V tabuľke 2.1 je zaznamenané koľkokrát sú sonary schopné merať za sekundu.

	Frekvencia [Hz]
Analógovo zapojené	4,6
Digitálne zapojené	1,2

Tabuľka 2.1: Frekvencie meraní rozdielne zapojených sonarov.



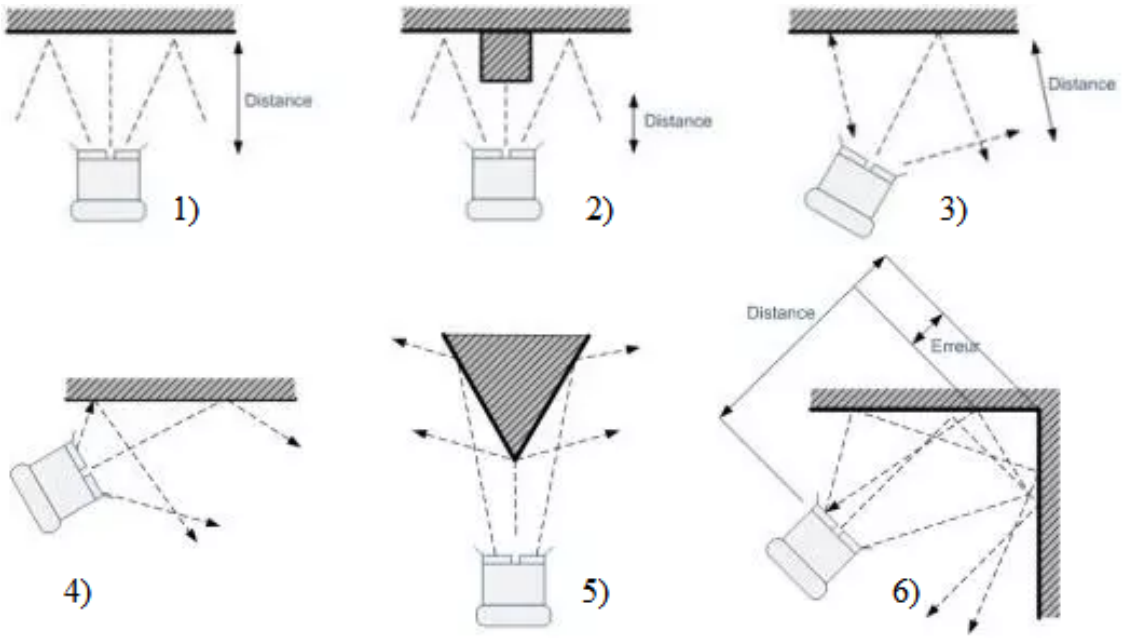
Obr. 2.4: Arduino Micro[8].

Sonary museli byť rozmiestnené a spúšťané tak, aby zvukové vlny sonarov neboli rušené. Zvukové vlny môžu byť rušené napríklad prúdom vzduchu, ktorý vytvárajú motory pri lietaní, vibráciami rámu kvadrokoptéry alebo výkyvmi napätia. Pre meranie vzdialenosti nad a pod kvadrokoptérou bola najlepšia voľba umiestniť sonary do stredu kvadrokoptéry. Ostatné sonary boli umiestnené na kostre kvadrokoptéry tak, aby sa ich zvukové vlny neprekrývali. Nevýhodou tohto typu sonarov je, že nepodporujú pripojenie cez I2C zbernicu a teda na zapojenie sonarov bolo potrebných viac vodičov. Vďaka pripojeniu cez I2C zbernicu by boli sonary stabilnejšie. V našom prípade museli byť sonary zapojené sériovo, lebo v opačnom prípade by sa muselo použiť veľké množstvo káblov. Pri sériovom zapojení však vzniká problém nárazového vysokého odberu napätia, čo znamená výkyvy napätia a v konečnom dôsledku nevalidné namerané hodnoty. Riešením bolo pridanie RC filtrov pre elimináciu krátkych výkyvov napätia.

Na nasledujúcom obrázku 2.5 sú znazornené špecifické prípady, v ktorých sonary vedia vykonať správne meranie, ale aj obrázky kde sonary zlyhávajú kvôli zle odrazenej zvukovej vlny. V prvom prípade bude nameraná hodnota vždy správna. V druhom prípade budú vo väčšine prípadov tiež správne namerané hodnoty, ale môže sa zvuková vlna odraziť aj od pozadia prekážky. V treťom prípade a v ďalších prípadoch sa vo väčšine prípadov nenamerajú

[8] https://cdn-reichelt.de/bilder/web/xxl_ws/A300/FRANZIS_65192-5.png

správne vzdialenosti.



Obr. 2.5: Znázornenie špecifických situácií meraní sonarov. Zdroj:[10]

2.4 Filtrácia hodnôt zo sonarov

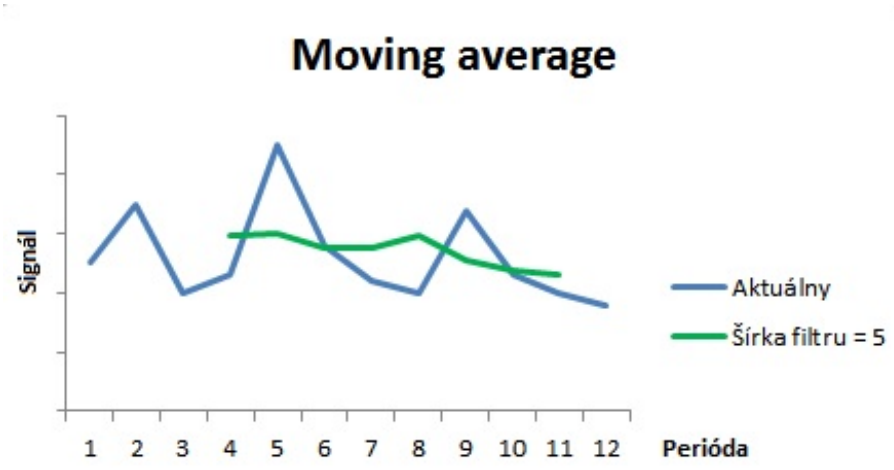
Kĺzavý priemer

Kĺzavý priemer (anglicky moving average) je softvérová technológia pre vyhladzovanie signálu, ktorá sa skladá z rovnomerne vzdialených bodov [7]. Pole nameraných hodnôt sa prevedie na vyhladené hodnoty. Šírka filtra sa udáva nepárnym číslom 2N+1. Čím väčšia je šírka filtra, tým efektívnejší je vyhladzovací účinok. Táto operácia je znázornená na animácií 2.6. V tomto príklade je šírka filtra 5. Čierne štvorčeky znázorňujú namerané hodnoty a červený obdĺžník znázorňuje hodnoty, ktoré sa budú spriemerovať. Spriemerovanou hodnotou je zelený štvorček. Pri ďalšom priemerovaní sa obdĺžnik posunie o jeden krok dopredu a hodnoty sa zase spriemerujú a vytvorí sa nová vyhladená hodnota, a tak ďalej.

Pomer signálu k šumu sa dá vylepšiť zvýšením šírky pásma alebo viacnásobným filtrovaním. Výsledky tejto techniky sú pôsobivé kvôli nadmernému filtrovaniu. Tento algoritmus nevyhovuje v prípade vysokých špičiek.

Mediánový filter

Pri spracovaní signálu je často žiaduce, aby bolo možné znížiť šum v obraze alebo v signále. Mediánový filter je nelineárna metóda vyhladzovania, ktorá sa používa na potlačenie impulzívneho šumu [7]. Redukcia šumu touto metódou je typické predspracovanie signálu pre zlepšenie výsledkov pri neskoršom spracovaní. Hlavná myšlienka algoritmu tohto filtru spočíva v posúvaní "okna" nad vstupnými hodnotami signálu. "Okno" určuje koľko hodnôt



Obr. 2.6: Animácia výpočtu kĺzavého priemeru[9].

sa bude naraz spracovávať. Aktuálne hodnoty z "okna" sa najprv zoradia od najmenšej po najväčšiu hodnotu a potom sa tieto hodnoty nahradia strednou hodnotou (mediánom). Postup je približne rovnaký ako pri kĺzavom priemere.

2.5 PID regulátor

PID regulátor je proporcionálny, integračný a derivačný regulátor, a patrí medzi najjednoduchšie lineárne regulátory [1].

Tieto regulátory sa konštruujú, preto aby riadenie procesu nevyžadovalo pozornosť a manuálne zásahy. Regulátor automaticky mení akčnú veličinu u(t) tak, aby regulovaná veličina y(t) mala požadovanú hodnotu w(t), kde t je čas. Regulovaný obvod je tvorený zo vstupu regulátora, kde je privedená skutočná hodnota (výstup riadenej sústavy) y(t) a požadovaná hodnota w(t), a z výstupu regulátoru.

Požadovaná hodnota regulovanej veličiny predstavuje ten bod, v ktorom by sa mala nachádzať skutočná hodnota regulovanej veličiny. Chyba a regulačná odchýlka je definovaná ako rozdiel medzi požadovanou, a skutočnou hodnotou regulovanej veličiny.

$$e(t) = w(t) - y(t) \tag{2.1}$$

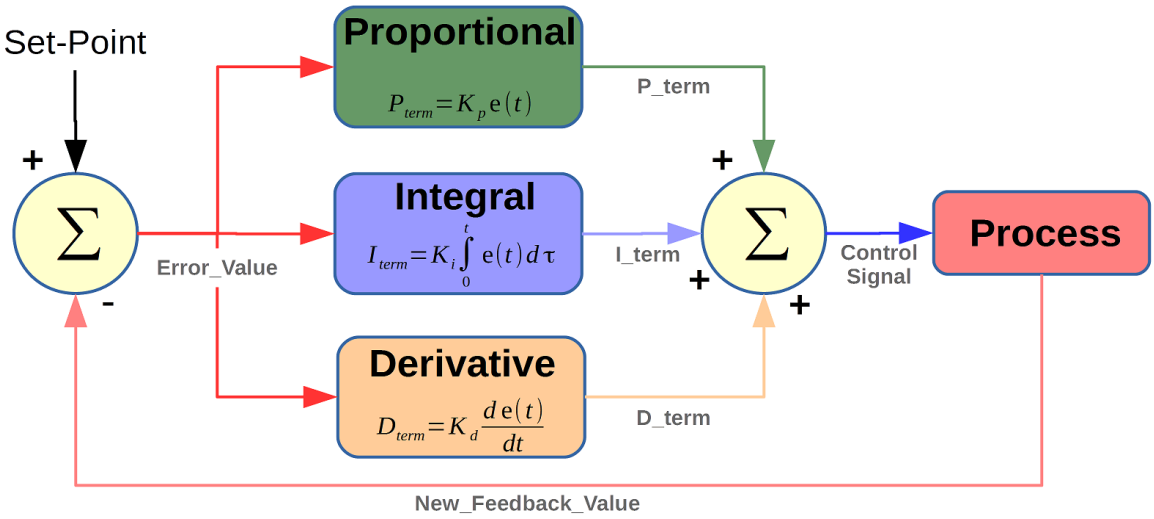
Príčinou nenulovej hodnoty e(t) teda môžu byť poruchová zmena regulovanej veličiny alebo cielená zmena jej žiadanej hodnoty. V závislosti na regulačnej odchýlke potom regulátor mení akčnú veličinu.

Výsledné chovanie PID regulátoru ovplyvňujú zložky: proporcionálne, derivačné a integračné. Tieto tri zložky sa môžu nastaviť prostredníctvom konfigurovateľných konštánt - parametrov PID regulátoru, vo voliteľnom pomere.

- P - proporcionálna zložka - obyčajný zosilňovač s nastaviteľným zosilnením
- I - integračná zložka - integrátor s nastaviteľným koeficientom
- D - derivačná zložka - derivátor s nastaviteľným koeficientom

[9] http://www.chem.uoa.gr/applets/appletsmooth/Images/Animation.gif

Zložky P, I a D majú rôzny dopad na finálny systém, a musia byť vyladené na konkrétnom riadiacom systéme. Cieľom riadiacich systémov je reagovanie, čo najrýchlejšie na zmeny, bez značného prekročenia a oscilácie okolo nastavenej hodnoty. Bloková schéma riadiaceho systému PID je zobrazená nižšie na obrázku 2.7.



Obr. 2.7: Bloková schéma riadiaceho systému PID[10].

Proporcionálne chovanie

Pri proporcionálnom chovaní je akčný zásah regulátoru úmerný regulačnej odchýlke podľa vzťahu:

$$u(t) = K_p e(t) \tag{2.2}$$

Použitie samotného proporcionálneho regulátoru so sústavami, u ktorých regulovaná veličina reaguje na akčný zásah konštantnej veľkosti ustálením sa na novej hodnote, vedie k vzniku trvalej regulačnej odchýlky. Zväčšením zosilnenia K_p sa dá trvalú regulačnú odchýlku zmenšiť. Vzniká však nebezpečie, že dôjde k tzv. nestabilite regulačného obvodu, tj. k stavu, kedy regulovaná veličina neobmedzene narastá až k dorazu alebo poškodeniu zariadenia. K odstráneniu trvalej regulačnej odchýlky sa do činnosti regulátorov zvyčajne pridáva integračná zložka.

Integračné chovanie

Pri integračnom chovaní je akčný zásah úmerný dobe, po ktorú existuje regulačná odchýlka, teda po pridaní integračnej zložky vzorec vyzerá nasledovne:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) d\tau \tag{2.3}$$

[10] http://i0.wp.com/coder-tronics.com/wp-content/uploads/2014/05/PID-Block-dia1.png

Keď je zapojený regulátor s proporcionálnou aj integračnou zložkou, tak trvalá regulačná odchýlka zmizne. Je eliminovaná integračným chovaním regulátoru, pri ktorom regulátor neustále mení akčnú veličinu, pokiaľ sa mu nepodarí dosiahnuť nulové regulačné odchýlky. Zväčšovaním podielu integračnej zložky (zmenšovaním koeficientu) kmitavosť regulácie obecne rastie. Do istej miery sa dá zmierniť pridaním derivačnej zložky.

Derivačné chovanie

Derivačná zložka slúži ku kompenzácii oneskorenia spôsobeného integračným chovaním regulátoru. Pri derivačnom chovaní sa výstup z regulátoru vytvára jako úmerný rýchlosti zmeny regulačnej odchýlky.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) d\tau + K_d \frac{de(t)}{dt} \tag{2.4}$$

Derivačné chovanie môže s predstihom kompenzovať zmeny regulovanej veličiny, a preto sa využíva k tlmeniu zákmitov regulácie. Ak po zmene žiadanej alebo skutočnej hodnoty regulovanej veličiny sa začne regulovaná veličina znovu približovať k žiadanej hodnote, tak derivačná zložka preventívne zmení zosilnenie do opačného smeru. Derivačná zložka regulátoru posúva fázu akčného zásahu vpred, a tým môže stabilizovať regulačnú slučku.

Je potrebné si uvedomiť, že základom správnej funkcie PID regulátoru sú optimálne nastavené hodnoty koeficientov. Optimálna regulácia závisí na miere znalosti regulovaného procesu. Obecne povedané, pre kvalitné riadenie je potrebné voliť čo najväčšie dynamické zosilnenie regulátoru so snahou priblížiť sa čo najviac k medze stability regulačného obvodu.

Nastavenie PID regulátora

Existujú metódy, ktoré uľahčujú nastavenie správnych konštánt jednotlivých zložiek PID regulátora:

Stabilizačná metóda

Táto metóda je najjednoduchšia, lebo nevyžaduje žiadne meracie prístroje [9].

1. Uveďte regulátor do polohy automatickej regulácie a odpojte integračné, a derivačné zložky.
2. Zvýšte zosilnenie K_p, až regulácia začne byť "nekľudná". Potom znížte zosilnenie na polovicu.
3. Pokiaľ chcete používať integračnú časť, žnižujte koeficient integračnej časti K_i, až regulácia začne byť "nekľudná". Potom zvýšte hodnotu na dvojnásobok.
4. Pokiaľ chcete použiť derivačnú zložku, zvyšujte koeficient K_d, až regulácia začne byť "nekľudná". Znížte potom hodnotu na polovicu.

Ziegler-Nicholsová metóda

Veľmi známa metóda, ktorá je kritizovaná, že je veľmi agresívna [11].

1. Uveďte regulátor do polohy automatickej regulácie a odpojte integračné, a derivačné zložky.

2. Zvýšte zosilnenie K_p do tej miery, až systém začne sám kmitať. Zaznamenajte si zosilnenie K_p a čas periódy pre vlastnú osciláciu.

3. Nastavte koeficienty v PID regulátori podľa nasledujúcej tabuľky:

Regulátor	K_p	K_i	K_d
P	$0.5K_{max}$		
PI	$0.45K_{max}$	$0.85T_0$	
PID	$0.6K_{max}$	$0.5T_0$	$0.125T_0$

Tabuľka 2.2: Ziegler-Nicholsova metóda: K_{max} je zosilnenie, kedy sa rozkmitá P-regulátor. T_0 je čas periódy kmitov.

2.6 Existujúce riešenia

V súčasnej dobe sú drony veľmi rozšírené a stretávame sa s nimi takmer všade. Drony sú dostupné pre bežných ľudí, ktorí ich chcú využívať len pre zábavu, až po drony využívané vo vojenskej technike. Jednotlivé drony majú rôzne stupne inteligencie. V tomto projekte budú predstavené existujúce riešenia dronov, ktorých stupeň autonómnosti zabezpečuje bezpečné riadenie a teda zabráni poškodeniu dronov pri lietaní úspešným vyhýbaním sa prekážkam.

Amazon package delivery quadcopter

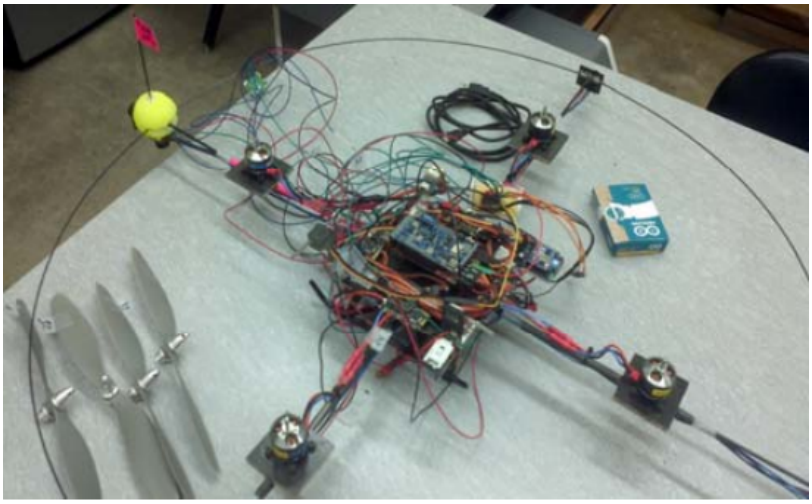
Kvadkoptéra na doručovanie balíkov slúžiaca pre jeden z najväčších internetových obchodov, ktorý sa volá Amazon [2]. Ide vlastne o protokol, ktorý bude nasadený v dronoch a jeho úlohou bude doručiť menšie balíky domov k zákazníkom. Pri doručovaní budú musieť drony častokrát prechádzať rušným mestom a dokonalé vyhýbanie sa prekážkam je nevyhnutné. Narozdiel od nášho dronu sa na tomto drone využívajú laserové senzory na bokoch a v rohoch. Na meranie priestoru nad a pod kvadrokoptérou využívajú ultrazvukové sonary. Navigácia dronu neprebieha riadením pilota ale navigáciou na určité miesto pomocou GPS súradníc. Na vyhýbanie sa prekážkam používajú fuzzy logiku pre ich nelineárny systém namiesto matematických funkcií. Hľavným cieľom fuzzy logiky nie je definovať presnú požadovanú vzdialenosť na obíjdenie prekážky, ale jednoducho nenaraziť do prekážky. Ďalším dôležitým cieľom fuzzy kontroléru je udržať kvadrokoptéru v stabilnej polohe po obíjdení prekážky. Využíva sa na to gyrometer.

Zabránenie narazeniu pomocou fuzzy logiky

Kvadrokoptére budú zadané GPS súradnice pre zanesenie balíku a cestou môže naraziť na prekážky. Použitím fuzzy logiky bude kvadrokoptéra schopná zabrániť narazeniu pomocou lacných senzorov a menej výkonným procesorom. Potrebné bude pokryť veľkú časť prostredia, ale keďže to bude rýchlo sa meniace prostredie, tak mapovanie okolia by bolo náročné. Fuzzy logika ponúka jednoduchú metódu pre riešenie nelineárnych systémov. Kvadrokoptéra sa bude vzďalovať od blížiacich sa prekážok. Po zabránení narazeniu sa kvadrokoptéra preorientuje a bude pokračovať na svojej ceste. Podľa toho, ktorý senzor zdetekuje prekážku, fuzzy kontrolér zvolí vstupy a výstupy. Vstupy budú senzory a osy kvadrokoptéry, výstupy budú motory. Prepojené vstupy a výstupy budú na základe fuzzy pravidiel.

Protinárazový systém pre kvadrotor UAV

Na kalifornskej univerzite v Pomone prebieha výskum bezpilotných vzdušných prostriedkov vzhľadom na ich potenciál k všestrannému použitiu [8]. Cieľom tohto výskumu je vyvinúť kvadrokoptéru, ktorá bude ľahká, nebude drahá, bude vytvorená z komponent, ktoré spotrebujú málo energie, a ktorá dokáže zmapovať a pohybovať sa v neznámom prostredí. Preskúmali sa rôzne možnosti nákupu jednotlivých komponent a vybrali sa komponenty, ktoré najviac vyhovovali požiadavkám. Poskladaná kvadrokoptéra je zobrazená na obrázku 2.8.



Obr. 2.8: Poskladaná kvadrokoptéra. Zdroj: [8]

Prekážky sa zaznamenávajú pomocou osadených ultrazvukových sonarov. Celkovo sa ich osadilo 6 kusov - 4 na bočné strany, 1 hore a 1 dole. Protinárazový systém a mapovanie sú rozdelené do menších častí.

V prvej časti protinárazového systému sa využíva sonar smerujúci vpred. Pri narazení na prekážku bude kvadrokoptéra schopná spomaliť a vzdialiť sa od prekážky.

V druhej časti protinárazového systému sa využívajú sonary smerujúce do ľavej a pravej strany pre vycentrovanie kvadrokoptéry medzi dvoma prekážkami pomocou push algoritmu. Hlavná myšlienka push algoritmu spočíva vo výbere uhlu pretáčania sa na základe vzdialenosti od prekážok. Ďalší algoritmus, ktorý sa využíva v tomto projekte je Rapidly Exploring Random Trees algoritmus[11]. Pomocou tohto algoritmu sa nielen zabráni narazeniu, ale naplánuje sa aj cesta pre obíjdenie prekážky.

Po kalibrovaní sonarov sa v skúšobnom teste prvej časti systému nastavil bezpečný odstup od prekážky - 30cm a uhol naklonenia pre vzdialenie od prekážky - 5 stupňov. Protinárazový systém prvej časti bol úspešný, ale do budúcna budú potrebné sonary s väčším rozsahom pre skoršie detekovanie prekážky.

V skúšobnom teste druhej časti systému sa zablokoval jeden sonar a stupne pre pretáčanie, a naklonenie neboli vyrovnané. Budúca práca na tomto výskume bude teda obsahovať aj opravenie a dokončenie druhej časti.

[11] http://coecsl.ece.illinois.edu/ge423/spring13/RickRekoskeAvoid/rrt.html

Kapitola 3

Návrh systému

Na základe zadania a predchádzajúcej kapitoly, v ktorej boli popísané teoretické základy bude v tejto kapitole vypracovaný návrh systému. Čitateľ sa v tejto kapitole dozvie detaily návrhu systému, ktorý zabezpečí bezpečné riadenie kvadrokoptéry.

3.1 Analýza zadania

Zadaním tejto práce je vytvoriť systém, ktorý zabezpečí bezpečnosť kvadrokoptéry aby pri riadení pilotom vo vnútornom prostredí neprišlo k nárazu do prekážky. Tento systém by mal byť aj prenositeľný na iné kvadrokoptéry pri detailných zmenách v softvéri v rámci prispôsobenia hardvéru. Kvadrokoptéry patria medzi ťažko ovládateľné drony a mnohé z nich skončia rozbité, kvôli slabým skúsenostiam pilota s riadením dronu alebo zavinením cudzieho predmetu. Osadenie kvadrokoptéry ultrazvukovými sonarmi sa javí ako vyhovujúce riešenie pre jej zabezpečenie.

Cieľom tejto práce je pokračovať na predchádzajúcej práci, v ktorej boli osadené sonary a vytvorené základné uzly v ROS pre spracovanie údajov z Arduina, na ktoré sú pripojené sonary. Pomocou sonarov a už vytvorených uzlov sa vytvorí bezpečnostný systém riadenia kvadrokoptéry vytvorením nových uzlov v ROS. V konečnom dôsledku by mal pilot kvadrokoptéry dokázať letieť bez možnosti poškodenia kvadrokoptéry, bez rozdielu na pilotové schopnosti. Tyra by mala umožniť ovládanie cez diaľkový ovládač a klávesnicu.

Podstata návrhu systému by mala spočívať vo výbere algoritmov, ktoré vyžadujú málo meraní a zároveň sú schopné vykonať výpočty v krátkej časovej dobe. Vzdialenosti od prekážok sú získavané zo 14 sonarov a keďže nie je možné vykonávať merania na všetkých sonaroch naraz, tak to spomalí objavovanie prekážok. Sonary v istých prípadoch zlyhávajú a to môže spôsobiť skreslenú hodnotu vzdialenosti, kvôli čomu bude potrebné merania zo sonarov filtrovať. Komunikáciu medzi ROS uzlami a samotné výpočty bude treba optimalizovať, aby sa dosiahol najmenší potrebný čas. Následne bude riešené vyhýbanie sa prekážkam pomocou ovládania motorov, kde sa vyberie najvhodnejší algoritmus, ktorý určí do ktorého smeru sa môže daná prekážka obísť.

3.2 Vstup údajov do systému

Vstupnými hodnotami systému budú vzdialenosti od prekážok zo sonarov a príkazy od pilota z klávesnice. Jeden sonar vytvorí za jednu sekundu 10 meraní, avšak v našom prípade,

kde sú niektoré sonary zapojené zreťazene sa frekvencia meraní znižuje a meranie je pomalšie. Sonary sú použiteľné, ale pri ich odskúšaní sa zistilo, že kontakty medzi Arduinom a zapojenými sonarmi nie sú spoľahlivé a občas sa rozpájajú, čo spôsobuje ďalšie spomalenie frekvencie meraní. Kvôli týmto a ďalším rušivým okolnostiam sonarov sa zavedie simulácia sonarov, ktorá umožní a uľahčí vývoj protinárazového systému a testovanie jednotlivých ROS uzlov. Na sonaroch sa vykoná diagnostikovanie problémov, pri ktorom budem snaha opraviť väčšinu problémov, stabilizovať sonar aby spojenie medzi Arduinom a počítačom nevypadávalo, a úprava programu v Arudine, pokiaľ to bude potrebné.

Pre simuláciu hodnôt podľa sonarov sa teda vytvorí uzol, ktorý ich bude publikovať na tému `sonar_data`. Tieto hodnoty budú zodpovedať reálnym hodnotám, hodnoty budú však generované podľa nasimulovaného prostredia. Pri dodržaní formátu správy nebude problém pripojiť na vstup systému rosbag súbor, ktorý obsahuje merania nahrané pomocou sonarov, alebo sonary.

3.3 Filtrovanie hodnôt zo sonarov

Pri meraniach sonarov môže dôjsť k chybám merania zlým odrazom signálu alebo vysokým rušením signálu z okolia. Hodnoty zo sonarov mali byť predspracované v samostatnom uzle, pre dosiahnutie väčšej presnosti. V tomto uzle sa mali najprv hodnoty predspracovať pomocou mediánového filtru, pre odstránenie nevalidných meraní a následne sa nad týmito hodnotami mal počítať kĺzavý priemer. Touto metódou by sa hodnoty vyhladili a dostali by sme presnejší model okolia. Experimentami sa však nepodarilo vytvoriť vhodnú šírku filtra, pretože pri príliš malej šírke filtra už metóda stráca význam a pre väčšiu šírku filtra sonar nevykonal dostatok meraní, a metóda spôsobovala vysokú odozvu, takže táto metóda bude vynechaná.

Zistilo sa, že filtrácia hodnôt zo sonarov pre získanie presnejšieho modelu okolia sa musí vynechať, kvôli zachovaniu rýchlosti detekovania prekážok. Použije sa obyčajná filtrácia nevalidných meraní, čiže hodnoty, ktoré nepatria do intervalu meracieho rozsahu sonarov.

3.4 Vyhýbanie sa prekážkam

Pri tejto časti návrhu som sa rozhodol pre dekompozíciu problému na menšie problémy. Teda najprv sa navrhne riešenie pre dva sonary a to: sonar pripojený na spodnú časť Tyry a sonar pripojený na hornú časť Tyry. V druhej časti sa pripoja k návrhu ďalšie štyri sonary pripojené na bokoch Tyry. A v poslednej časti sa navrhne riešenie za využitia všetkých sonarov.

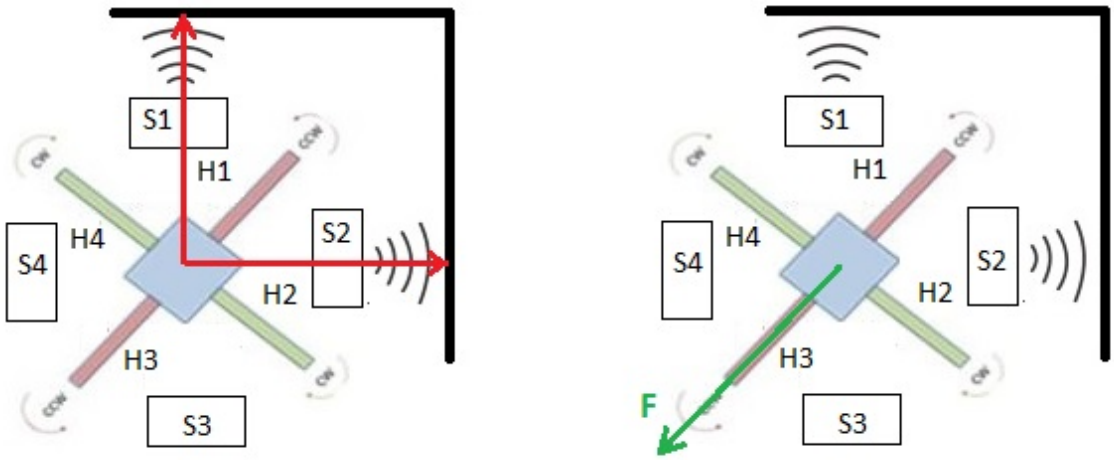
Prvý krok

Využitím sonarov na meranie vzdialenosti medzi kvadrokoptérou a prekážkami, ktoré sú umiestnené priamo pod a nad kvadrokoptérou, sa zabezpečí aby kvadrokoptéra mohla udržovať určitú výšku napríklad 1 meter a nespadnúť. Podrobne je tento postup popísaný nižšie v podkapitole 3.5.

Druhý krok

Pridajú sa sonary pre detekciu prekážok po bokoch kvadrokoptéry. Určí sa bezpečný odstup od prekážok (napríklad 100cm). Sonary umiestnené na bokoch, ktoré namerajú vzdialenosť pod touto hranicou, budú vyhodnoté, že narazili na prekážku a vykoná sa pohyb od prekážky.

Keď detekuje prekážku jeden sonar na ľubovoľnom boku, tak sa na hranách, ktorý tento sonar prepája, zvýši dočasne rýchlosť motoru pre pohyb od prekážky.



Obr. 3.1: Vyhýbanie sa prekážkam pomocou štyroch sonarov.

V príklade na obrázku sonary S1 a S2 detekujú prekážku. Motor na hrane H1 určuje prednú časť kvadrokoptéry. Vykoná sa merací cyklus, ktorého výsledokom je model okolia a určí sa kade môže Tyra odletieť. Pre zabránenie narazeniu do prekážky sa zvýši rýchlosť motoru na hrane H1. Rýchlosť motoru na hrane H1 sa krátkodobo zvýši na najvyššiu možnú rýchlosť pre odklonenie sa od prekážky a zároveň zabránenie narazeniu. Ďalej sa udrží na rýchlosti, ktorá je potrebná pre udržanie výšky.

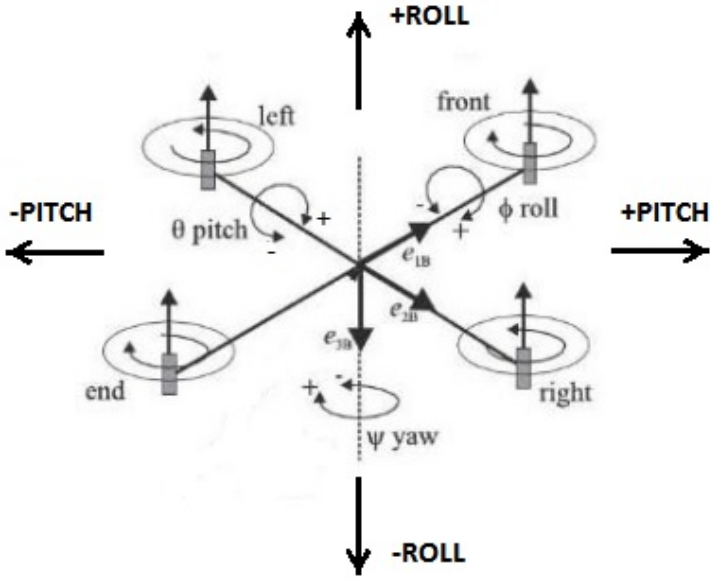
Tretí krok

Pokiaľ sa kvadrokoptéra pohybuje dopredu rohom kvadrokoptéry, podobne ako v predchádzajúcom prípade, tak je pravdepodobnejšie, že na prekážku narazí práve roh kvadrokoptéry namiesto boku. Pre tieto prípady objavovania prekážok už bude potrebné použiť všetky sonary a pohyb bude prebiehať pomocou naklonenia (anglicky pitch) a preklopenia (anglicky roll) kvadrokoptéry od prekážok.

Pre detekovanie prekážky bude stačiť meranie z jedného sonara. Pokiaľ sa zaznamená hodnota, ktorá bude menšia ako zadefinovaná hranica, tak sa pre pohyb kvadrokoptéry zistia merania zo všetkých sonarov. Tým sa vytvorí aktuálny model okolia a tieto hodnoty sa budú ukladať na základe obrázka 3.2. Budú sa ukladať hodnoty pre Pitch a Roll. Zo sonarov, ktoré sa nachádzajú v rohoch sa bude ukladať do Pitch a Roll vždy menšia hodnota. Nakoniec sa vypočíta súčet týchto hodnôt a na základe nich sa určí do ktorého smeru sa vykoná pohyb kvadrokoptéry od prekážky. Pomer medzi Roll a Pitch sa určí pomerom výsledkov súčtov. Vzdialenosti zo sonarov nachádzajúce sa v rohoch budú ovplyvňovať aj rýchlosť motorov.

3.5 Letový mód udržania stabilnej výšky

Mód, v ktorom sa kvadrokoptéra z kľudového stavu začne samostatne zdvíhať zo zeme. Výšku zvyšuje postupne a pomaly sa približuje k vopred určenej vzdialenosti od zeme. Výška od zeme sa bude merať pomocou sonaru a rýchlosť motorov sa bude regulovať pomocou PID regulátoru. Cieľom je aby po dosiahnutí zvolenej výšky mala kvadrokoptéra čo najmenšie



Obr. 3.2: Zobrazenie orientácie kvadrokoptéry.

výkyvy a dokázala spoľahlivo udržať výšku.

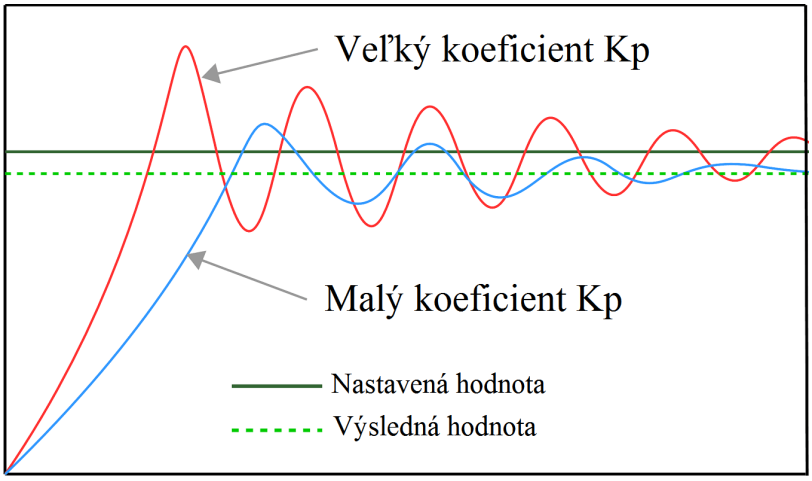
Regulačná odchýlka sa vypočíta z nastavenej výšky, na ktorej chceme kvadrokoptéru udržať a aktuálnej výšky, na ktorej sa kvadrokoptéra aktuálne nachádza. Odchýlku, ktorú týmto spôsobom dostaneme vynásobíme konštantou K_p, aby sme dostali hodnotu, ktorú použijeme na znižovanie a zvyšovanie rýchlosti motorov.

$$odchylka = K_p(nastavena_vyska - aktualna_vyska) \tag{3.1}$$

Akonáhle budú motory zrýchlovať a kvadrokoptéra sa začne zdvíhať zo zeme, tak sa odchýlka začne znižovať. Toto je žiaduce a pri nulovej odchýlke by mala kvadrokoptéra udržať výšku v ideálnom svete. Ale keďže sa lúč sonaru môže rôzne odraziť, tak je potrebné počítať s menšími výkyvmi. Konštanta Kp určuje ako intenzívne by mala byť odozva porovnávaná s odchýlkou. To znamená, že ak je v čase t kvadrokoptéra 30cm ďaleko od požadovanej hodnoty (e(t)=30), tak sa aplikuje taká konštanta K_p, pomocou ktorej sa získa korektná hodnota. Ak by bola nastavená hodnota K_p príliš vysoká, tak by sme požadovanú hodnotu mohli minúť. Rovnovážny stav odchýlky sa bude znižovať s narastajúcim zosilnením, ale takisto sa zvýši tendencia kmitania. Kvadrokoptéra nebude spoľahlivo ovládateľná len pomocou tejto zložky.

Nižšie uvedený obrázok 3.3 znázorňuje priebehy odozvy proporcionálnej zložky pri nastavenom nízkom a vysokom koeficiente K_p.

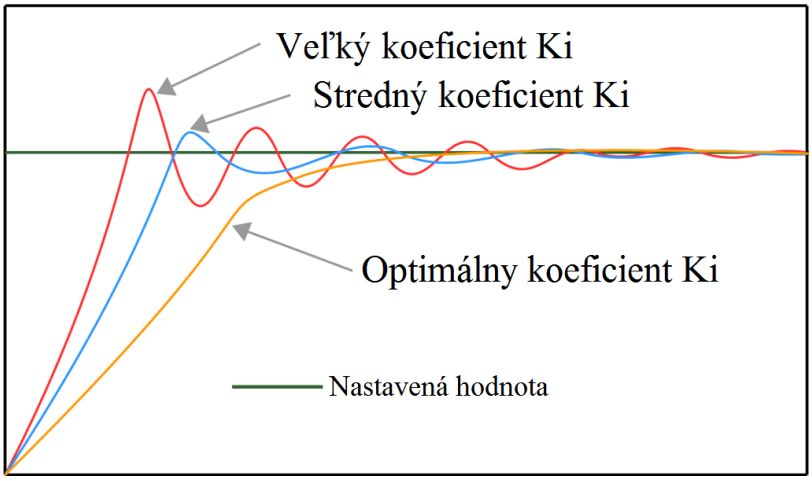
Pomocou integračnej zložky sa vypočíta priemerná odchýlka, ako suma okamžitých odchýlok v priebehu času. Týmto vznikne narastajúca odchýlka, ktorá sa potom vynásobí s koeficientom K_i a takto vzniknutú vyladenú odchýlku, môžeme potom pridať na výstup regulátoru. Toto potom urýchli odozvu na výstupe regulátora smerom k nastavenej výške kvadrokoptéry, odstránením odchýlky, ktorá bola vypočítaná len pomocou proporcionálnej zložky. Pokiaľ bude koeficient K_i veľmi vysoký, tak bude mať systém rýchlejšiu odozvu, ale bude pravdepodobnejšie, že sa presiahne nastavená hodnota, čo by spôsobovalo nežia-



Obr. 3.3: Zobrazenie samotnej proporcionálnej zložky.

duce kmitanie. Ideálny koeficient K_i bude vytvárať odozvu, ktorá je dostatočne rýchla pre systém, s malým alebo žiadnym presahom nastavenej výšky a rýchlou dobou ustálenia na danej hodnote.

Nižšie uvedený obrázok znázorňuje priebehy odozvy integračnej zložky pri vysokom, nízkom a ideálnom koeficiente K_i, kde je už proporcionálna zložka nastavená.



Obr. 3.4: Zobrazenie integračnej zložky.

Vo väčšine prípadov sú regulátory postavené len na zložkách PI. Derivačná zložka sa využíva v systémoch, kde vysoké prekročenie nastavenej hodnoty môže mať nebezpečné následky. Vysoké prekročenie nastavenej výšky by mohlo znamenať narazenie kvadrokoptéry do stropu a následné rozbitie. Diferenciácia sa používa na stanovenie miery zmeny pomocou aktuálnej odchýlky a predchádzajúcej odchýlky. Takže pomocou tejto hodnoty vynásobenej s konštantou K_d, dokážeme predvídať chovanie procesu a zlepšiť dobu ustálenia systému.

Súčtom týchto troch hodnôt získame výstup, ktorý bude určovať rýchlosť otáčania motorov. Keď už bude PID regulátor udávať správne hodnoty, tak bude potrebné ešte nastaviť interval, pre validné hodnoty motorov. Intervalom budú reálne čísla od 0 do 1.

Nakoniec sa vytvorí procedúra, ktorej úlohou bude v bezpečí položiť Tyru na zem.

Algoritmus bude použitý z procedúry udržania výšky kvadrokoptéry s rozdielom, že sa kvadrokoptéra bude približovať k zemi namiesto udržania v určitej výške. Rozsah merania sonarov sa začína pri 20 cm. Zhodenie kvadrokoptéry z tejto výšky ešte môže spôsobiť poškodenie, teda kvadrokoptéra sa bude spúšťat z tejto výšky postupne s pomalým krokom.

Kapitola 4

Realizácia

V tejto kapitole je popísaná štruktúra systému a komunikácia medzi uzlami. Uzly, ktoré tvoria hlavnú funkciu protinárazového systému sú detailne popísané. A čitateľ sa dočíta aj o problémoch pri spojazdnení kvadrokoptéry Tyry.

4.1 ROS balík *safety_flight_control*

V rámci mojej práce som vytvoril ROS balík *safety_flight_control* v ROS workspace-i *ros_ws*. V tomto balíku sa nachádzajú prevzaté uzly z predchádzajúcich riešení, ktoré boli mnou modifikované a ďalšie uzly, ktoré boli mnou vytvorené.

Modifikovaný uzol *keyboard_input*
- vytvorený uzol slúži na spracovanie príkazov, ktoré boli zadané užívateľom cez klávesnicu. Tieto príkazy sa potom publikujú na tému `instructions`. Do uzlu boli pridané príkazy pre zapínanie nových letových módov. Bola zmenená regulácia rýchlosti motorov a boli pridané ošetrenia pre nežiaduce prípady.

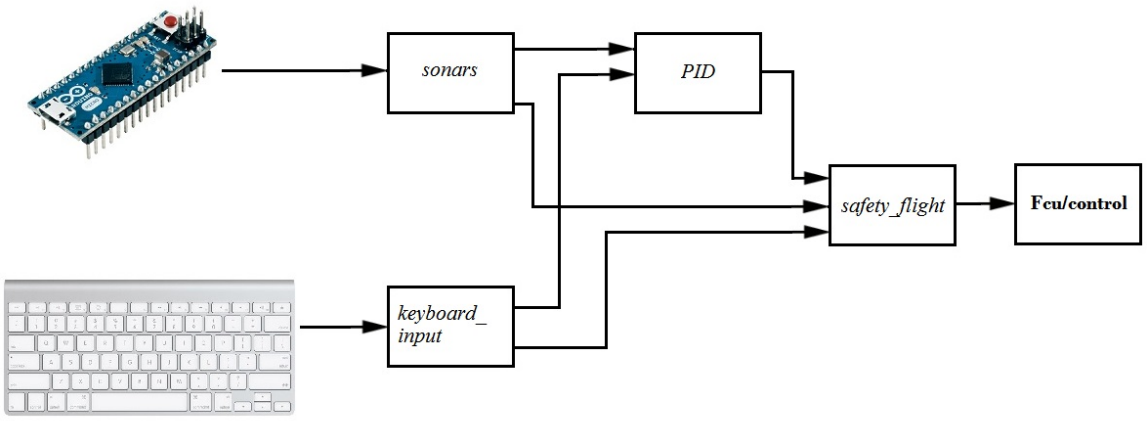
Prebratý uzol *sonars*
- v tomto uzli sa nevykonali žiadne podstatné zmeny. Uzol slúži na zber dát zo sonarov z Arduina a ich publikovanie.

Nový uzol *PID*
- vytvorený uzol, ktorý obsahuje letové módy pre udržanie výšky kvadrokoptéry a pristátie. Odoberajú sa príkazy z uzla *keyboard_input* a vzdialenosti pod, a nad kvadrokoptérou z uzla *sonars*. Regulovaná rýchlosť motorov sa publikuje do hlavného uzla *safety_flight_control*

Nový uzol *safety_flight*
- vytvoril sa uzol safety_flight_control, ktorý slúži ako hlavný program bezpečného riadenia kvadrokoptéry. Spracovávajú sa tu príkazy z klávesnice, hodnoty zo sonarov a údaje z PID regulátora. Výstupom je publikovanie na tému ovládania motorov `Fcu/control`.

4.2 Spojazdnenie kvadrokoptéry a práca so sonarmi

Mojou úlohou bolo zabezpečiť bezpečné riadenie a do ničoho nenaraziť. Teda hlavná práca spočívala vo výbere ideálneho algoritmu a v lietaní s Tyrou, pričom by sa tento algoritmus



Obr. 4.1: Schéma komunikácie ROS uzlov.

skúšal a ladil do dokonalosti, aby sa tento bezpečný systém mohol vytvoriť. Avšak vyskytli sa problémy, ktoré bolo nutné riešiť, aby som mohol začať svoju prácu.

Problémy s kvadrokoptérou

Prvým problémom pri snahe o roztočenie motorov cez ROS bolo chybové hlásenie ohľadom neúspešného spojenia. Ako sa s odstupom času zistilo, problém bol v pripojení AutoPilota pomocou USB do Atomboardu, kde sa toto pripojenie vyhodnotilo ako nepripojené, aj keď je AutoPilot stabilne spojený s Atomboardom. Tento problém zostáva zatiaľ stále nevyriešený, obísť sa tento problém dá tak, že po pripojení batérie sa AutoPilot manuálne odpojí a znovu pripojí. Po nabehnutí systému už potom tento problém nepretrváva.

Ďalší problém s ktorým som sa stretol na začiatku bol ten, že Tyra nedokázala vzlietnuť pokiaľ bola napojená na napájací zdroj namiesto batérie. Vzlietnuť pomocou napájacích káblov by mi značne uľahčilo prácu a ušetrilo čas, dá sa však lietať len pomocou napojenej batérie.

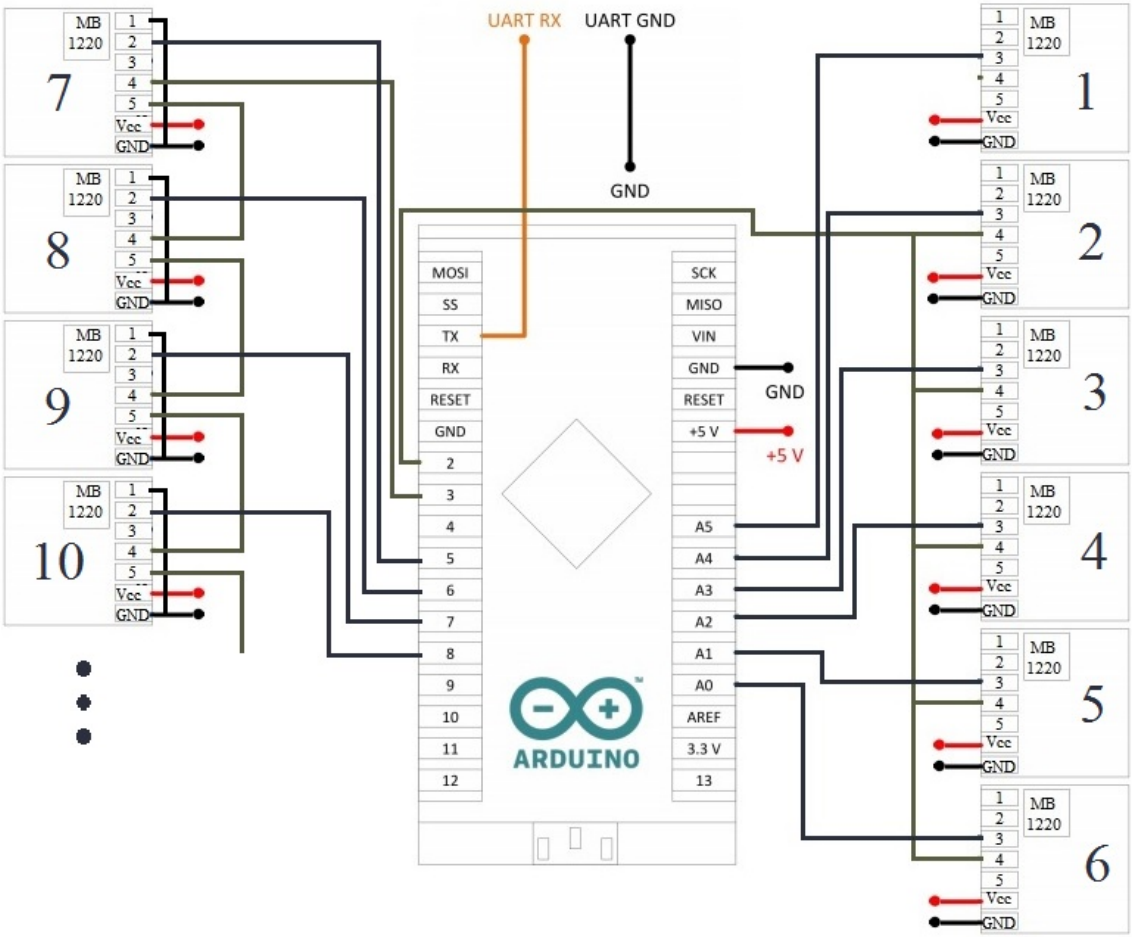
Problémy so sonarmi

Pri prvom stretnutí so sonarmi sa mi nepodarilo zapnúť uzol v systéme ROS, ktorý slúži na spracovávanie údajov z Arduina. Po prešetrení problému som zistil, že chyba bola v slabom pripevnení Arduina na kostru Tyry. Toto spôsobovalo rozpájanie kontaktov a opakované znovupripájanie na novom porte V implementácii sa s týmto nepočítalo a nastavený bol jeden konkrétny port [5]. Po riadnom pripevnení Arduina už tento problém nepretrváva.

Pri spustení ROS uzlu, ktorý slúži na spracovávanie údajov z Arduina sa dostávali na výstup hodnoty, ktoré boli mimo rozsah sonarov. Premeraním jednotlivých sonarov som zistil, že niektoré sonary nemerali vôbec, či už kvôli studeným spojom, rozpojeným kontaktom alebo zlomeným káblom. Diagnostikované chyby vyžadovali opravu. Káble, pri ktorých sa zistila porucha boli vymenené, pri niektorých sonaroch stačilo znovu prispájkovať vodiče, ale boli aj sonary, ktoré sa museli vymeniť za nové.

V programe, ktorý bol nahraný v Arduine sa nachádzala implementácia PID regulátoru, ktorá mierne spomaľovala frekvenciu meraní. Väčším problémom však bolo zistenie, že sonary nie sú zapojené podľa schémy. Zistilo sa to z implementácie programu, ktorá sa nachádzala v Arduine. Podľa schémy a popisu mali byť sonary rozdelené do dvoch skupín -

analógovo a digitálne zapojené sonary. V analógovo zapojených sonaroch sa mali nachádzať sonary, ktoré sa môžu spúšťať naraz a teda sonary bočné, dolný sonar a horný sonar. Spúšťanie merania malo prebiehať pomocou jedného pinu na Arduinu, kde sa mal vygenerovať signál pre merací cyklus. V implementácii boli nastavené dva piny pre spúštanie meracieho cyklu a horný sonar nepatril do skupiny analógovo zapojených sonarov, aj keď bol ešte jeden pin voľný. Horný sonar bol zapojený medzi digitálne zapojenými sonarmi. Implementácia programu bola tomuto prispôsobená, takže sonary merali správne, ale pomaly. Digitálne pripojený sonar vykoná približne 1 meranie za sekundu, kým analógovo zapojený sonar vykoná 3 až 4 merania. Na detekovanie prekážok nad kvadrokoptérou je potrebná čo najvyššia možná frekvencia meraní, čiže bolo potrebné prepojiť sonary podľa pôvodnej schémy a upraviť program v Arduine. Výsledné zapojenie je zobrazené na schéme 4.2. Po zapojení sonarov podľa schémy sa zvýšila frekvencia meraní na 3 až 4 merania za sekundu.



Obr. 4.2: Schéma zapojenia sonarov.

4.3 PID regulátor

Existujúca implementácia

V predchádzajúcej práci sa na Arduino implementoval PID regulátor [5]. Bola využitá knižnica, z ktorej sa použilo ukážkové použitie. Požadovaná hodnota výsledku regulácie bola 25cm, teda tento regulátor sa mal využiť na bezpečné pristátie Tyry. Vstupnou hodnotou bola vzdialenosť od prekážky nameraná spodným sonarom. Keďže sa hodnota potrebná pre regulovanie rýchlosti motorov pre pristátie začínala počítať už pri zapnutí Tyry, tak bolo potrebné resetovanie PID regulátoru v momente potreby pristátia. Toto riešnie sa javilo ako správne, ale nebolo to odskúšané. Po odskúšaní som zistil, že táto implementácia môže fungovať len v teoretickej úrovni.

Implementácia v ROS

Dôvodom implementácie PID regulátora v systéme ROS bola v prvom rade potreba implementovať reguláciu pre udržanie výšky a takisto pristátie. To by vyžadovalo vytvorenie nového PID regulátora v programe v Arduine, čo by viedlo k ďalšiemu spomaleniu meracieho cyklu sonarov, a tiež ku komplikáciám pri komunikovaní s Arduinom zo systému ROS. Po odstránení implementácie PID regulátora z Arduina sa zvýšila frekvencia meraní sonarov. Ďalším dôvodom bola skutočnosť, že motory kvadrokoptéry sa stávajú citlivejšími pri vzlietnutí zo zeme a vyžadujú konštantný interval zmien rýchlosti. Nakoniec implementácia v systéme ROS sa ľahšie a rýchlejšie ladí ako program v Arduine.

Vytvorený bol nový uzol, ktorý obsahuje nastavenie PID regulátora a letové módy pre udržanie letovej výšky, a mód pristávania. Implementácia PID regulátora je rozdelená na dve časti. V prvej časti sa nastavia počiatočné hodnoty a konštanty jednotlivých zložiek. V druhej časti prebieha výpočet a teda samotné regulovanie. Pre výpočet konštánt som zvolil stabilizačnú metódu. Odpojil som integračnú a derivačnú zložku a začal ladiť proporcionálnu. Konštantu bolo potrebné ladiť s ohľadom na interval, v ktorom motory pracujú. Podľa stabilizačnej metódy som vybral hodnotu Kp, pomocou ktorej budú na výstupe kmitavé hodnoty a následne ju znížil na polovicu. Podobne podľa postupu som riešil aj ostatné zložky. Po prvom skúšobnom lete sa kvadrokoptéra nechovala podľa predstáv. Bolo potrebné ďalej ladiť konštanty a ošetriť integračnú zložku, kvôli veľmi vysokým a nízkym hodnotám. Nakoniec sa zvolil interval výpočtu nasledujúcej hodnoty ohľadom na motory a sonary.

V letovom móde bolo riešené pristávanie pri 25cm nad zemou, kam už sonary nemajú dosah. Tyra pri dosiahnutí tejto výšky konštantnou rýchlosťou začne znižovať rýchlosť motorov, až do kľudovej rýchlosti. Letové módy sa zapínajú na klávesnici.

4.4 Protinárazový systém

Prvé pokusy

Podľa prvotného návrhu sa pomocou meraní zo štrnástich sonarov mal vytvoriť model okolia pri nájdení prekážky a následne vykonať uhýbací manéver, kde smer a sila pohybu mali vyplývať z výpočtu. Na osi Z bolo toto nezrealizovateľné. Pri vzlietnutaní zo zeme sa automaticky zaznamenávala prekážka, čo spôsobovalo nebezpečné zrýchlenie motorov. Pri zapnutí funkcie sonarov až pri lietaní prichádzal problém presného nastavenia rýchlosti motorov pri zaznemnaní prekážky. Tieto okolnosti viedli k použitiu PID regulátora, ktorý je popísaný vyššie. Zostalo zvyšných 12 sonarov, ktoré sú umiestnené na bokoch a v rohoch

Tyry. Znovu bol pôvodný plán vytvoriť algoritmus, ktorý pri detekovaní prekážky vytvorí model okolia, pomocou výpočtu zistí smer a silu pohybu, a pomocou PID regulátoru dostane Tyru na požadované miesto. Veľký šum v meraniach sonarov umiestnených v rohoch, zapríčinený hlavne častým zlým uhlom merania alebo rušením z okolia tento postup nepripúšťa. Kvôli zlej stabilite kvadrokoptéry a skutočnosti, že motory pre navigovanie kvadrokoptéry po osách X a Y sa nastavujú odlišne ako na osi Z, nebol použitý PID regulátor na pohyb kvadrokoptéry od prekážky. Kým na osi Z sa nastaví konštantná rýchlosť, ktorá sa udržuje na osách X a Y sa nastavuje rýchlosť pomocou uhlu naklonenia. Nebolo by možné zaistiť aby sa kvadrokoptéra nepohla o viac ako je potrebné, pomôcť v tomto probléme by mohlo IMU [6], ale je príliš zašumené, a tým pádom by jeho použitie neviedlo k správnemu riešeniu. Ďalším dôvodom prečo sa nemohol použiť PID regulátor je ten, že pri naklonení kvadrokoptéry sa zmení uhol pri odrážaní zvukovej vlny, čo môže viesť ku skresleným vzdialenostiam. Pôvodné spriemerovanie hodnôt zo sonarov pre dosiahnutie reálnej vzdialenosti som bol nútený vynechať kvôli vysokej odozve.

Konečný algoritmus

Hodnoty zo sonarov sa pri vstupe do systému filtrujú a pokiaľ nie sú validné, tak sa tieto merania neberú do úvahy. Ďalej sú sonary rozdelené do dvoch skupín. V prvej skupine sú sonary osadené na bokoch Tyry, na týchto sonaroch sa vykonajú merania s čím sa získa aktuálny model okolia a na základe výpočtu sa vykoná pohyb od prekážky. Pri nameraní prekážky z protichodných strán zostane Tyra na mieste. Ak sa nenameria žiadna prekážka na bočných sonaroch, tak sa model okolia vytvorí pomocou sonarov umietnených v rohoch kvadrokoptéry, ďalej je postup rovnaký. Tento spôsob získavania modelu okolia bol zvolený z dôvodu, že sonary na bokoch Tyry sú zapojené analógovo a teda vykonajú viacero meraní a vďaka ich umiestneniu prekážky zaznamenávajú častejšie pod kolmým uhlom, čo zaručuje presnejšie merania. Naopak sonary umiestnené v rohoch Tyry vykonajú trikrát menej meraní, lebo sú zapojené multiplexne a aby zbytočne nespomaľovali objavovanie prekážok, tak sú použité len vtedy, keď bočné sonary prekážku neobjavia. Prekážka sa vyhodnotí vtedy, keď je nameraná hodnota menšia ako 40cm. Táto hodnota bola zvolená na základe experimentov ako ideálna vo vnútornom prostredí. Pokiaľ sa nenarazí na prekážku ani v jednom z dvoch krokov, tak sa príjmajú príkazy od pilota. Výsledkom je ovládanie motorov, na základe vypočítanej hodnoty. Pre správny pohyb v osách X a Y bolo potrebné vyladiť frekvenciu odosielania príkazov na tému `fcu/control`.

Kapitola 5

Testovanie

Táto kapitola čitateľa oboznámi s detailmi testovania. Popisujú sa tu potrebné zabezpečenia pred lietaním, potrebné nastavenia, vytvorenie prostredia, samotné testy a výsledky testov. Cieľom tohto testovania bolo overiť funkčnosť zabezpečeného lietania.

5.1 Zabezpečenie a nastavenie pred lietaním

Kvôli zlej stabilite kvadrokoptéry, obecne ťažkému ovládaniu dronov a chybám, ktoré sa mohli v implementácii vyskytnúť bolo potrebné Tyru zabezpečiť pred pádmi na zem a následnému rozbitiu. Použila sa teleskopická tyč, na ktorej je Tyra zavesená na povraze. Držanie tejto tyče vyžaduje ďalšiu osobu pri testovaní, ale ochráni to Tyru pred pádmi. Človek, ktorý túto tyč držal musel byť maximálne pozorný a pohyboval sa s tyčou pri lietaní, tak aby sa povraz nezamotal do rotorov. Takisto bol nútený použiť rukavice pre prípad, žeby Tyra vyletela veľmi vysoko a bolo by potrebné sa jej dotknúť ručne pri zapnutých motoroch.

Testovanie bude prebiehať použitím klávesnice pre ovládanie Tyry. Pri lietaní môžu byť na kvadrokoptére pripojené len tie kompontenty, ktoré sú pre lietanie nevyhnutné. Pripojená klávesnica cez kábel a monitor teda nebudú vhodné pre testovanie. Klávesnica môže byť pripojená bezdrôtovo, ale pri testovaní sa oplatí sledovať výpis meraní sonarov kvôli bezpečnosti a dôkladnému testovaniu. Systém ROS umožňuje vzdialené ovládanie robota. Na mobile som vytvoril zdieľaný bod pripojenia, kam som sa pripojil s počítačom, cez ktorý som chcel Tyru ovládať a Tyra sa tam pripojila sama na základe konfigurácie, ktorú má nastavenú v konfiguračnom súbore. Lokálna IP adresa Tyry musela byť zadaná v príkazovom riadku na Tyre:

export ROS_MASTER_URI=http://IP_Adress:11311/
export ROS_IP=IP_Adress

a na počitači:

export ROS_MASTER_URI=http://IP_Adress:11311/

následne sa na Tyre zadal príkaz *roscore* pre spustenie systému ROS a prepojenie robota s počítačom. Toto umožňuje Tyru ovládať a sledovať jednotlivé témy, na ktoré sa na Tyre publikuje. Tento spôsob značne uľahčil testovanie.

Pred lietaním s Tyrou je nutné vykonať tieto kroky v nasledujúcom poradí:

1. pripojiť baterku
2. odpojiť USB kábel, ktorý prepája základnú dosku s AutoPilotom a znovu ho pripojiť
3. zapnúť RC ovládač a AutoPilot

4. prepínač B nastaviť do hornej polohy a roztočiť motory pomocou ovládaču potom ich vypnúť
5. prepínač B nastaviť do dolnej polohy aby príjmalo príkazy z HLP
6. po nabootovaní systému spustiť ROS uzly

Po vykonaní týchto krokov je Tyra pripravená na testovanie.

5.2 Testovanie protinárazového systému

Test reakcie na prichádzajúcu prekážku

V prvom scenári sa vytvorilo prostredie, ktoré je znázornené na obrázku 5.1. V okolí Tyry po vzlietnutí sa nenachádzala žiadna prekážka bližšie ako 1 meter. Úlohou pilota bolo pridávať a uberať plyn, tak aby udržal kvadrokoptéru približne 1 meter nad zemou. Žiadne iné pohyby s kvadrokoptérou. Testovalo sa, či sa Tyra dokáže vzdialiť od blížiacej sa prekážky a ako rýchlo. Vykonali sa 3 testy na príchod prekážky naproti bočnému sonaru (sonar zapojený analógovo) a 3 testy na príchod prekážky naproti rohovému sonaru (sonar zapojený digitálne). Prekážkou v tomto prípade bola osoba, ktorá sa ku kvadrokoptére pomaly približovala.

Obr. 5.1: Vytvorené prostredie bez prekážok.

Číslo testu	Detekovanie prekážky sonarom	Odozva systému	Zabránenie narazeniu
1	áno	120ms	áno
2	áno	110ms	áno
3	áno	90ms	áno

Tabuľka 5.1: Tabuľka výsledkov testov - Reakcia na prichádzajúcu prekážku z boku.

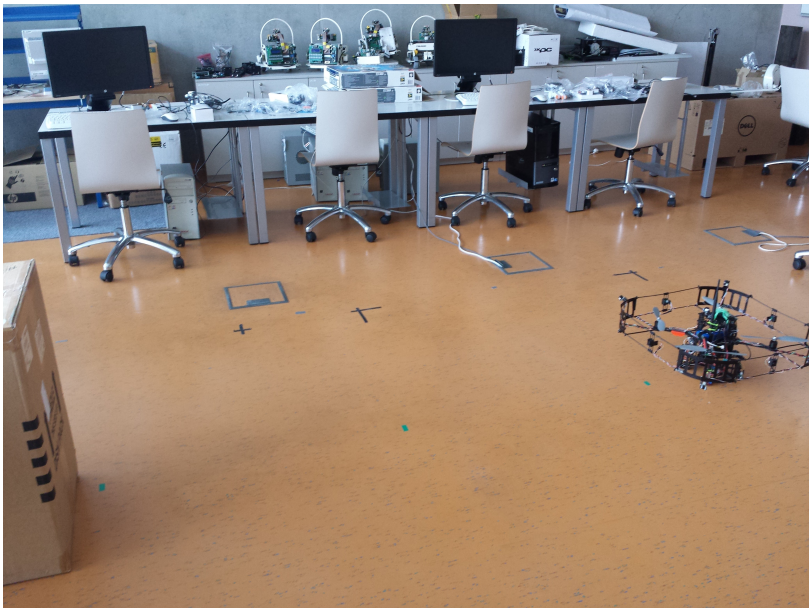
Číslo testu	Detekovanie prekážky sonarom	Odozva systému	Zabránenie narazeniu
1	nie		nie
2	áno	330ms	áno
3	áno	350ms	áno

Tabuľka 5.2: Tabuľka výsledkov testov - Reakcia na prichádzajúcu prekážku z rohu.

Z tabuliek 5.1 a 5.2 s výsledkami testov je jasné, že pokiaľ osoba prichádzala z boku ku kvadrokoptére, tak protinárazový systém zabránil narazeniu v každom prípade. Keď osoba prichádzala z rohu, tak jeden test zlyhal, lebo boli namerané nevalidné hodnoty na oboch sonaroch v danom rohu kvadrokoptéry.

Test zabránenia narazenia pri zrýchlení

V druhom scenári sa vytvorilo nové prostredie, v ktorom sa nachádzala 1 prekážka, viď obrázok 5.2. Testovalo sa, že ak sa umiestni Tyra 2 metre od prekážky, pod jakým uhlom naklonenia pre zrýchlenie stihne Tyra ešte prekážku zaznamenať a vzdialiť sa od nej. Testovanie začínalo na nízkom uhli naklonenia, pri ktorom Tyra dokázala spoľahlivo zabrániť narazeniu. V ďalších testoch sa uhol zvyšoval dovtedy, kým Tyra nebola schopná zabrániť narazeniu. Údaje sú zaznamené v tabuľke.



Obr. 5.2: Vytvorené prostredie s jednou prekážkou.

Testovať sa začalo pri nastavenom uhle naklonenia 0.1rad. Kvadrokoptéra pri tomto naklonení nabrala bezpečnú rýchlosť, pri ktorej dokázala bez problémov zabrániť narazeniu. Podľa hodnôt v tabuľke 5.3 sa uhol naklonenia zvyšoval každým testom, až po hodnotu 0.3rad, kde kvadrokoptéra mierne narazila.

Číslo testu	Uhol naklonenia	Zabránenie narazeniu
1	0.1rad≐6°	áno
2	0.15rad≐8.5°	áno
3	0.2rad≐11.5°	áno
4	0.25rad≐14°	áno
5	0.3rad≐17°	nie

Tabuľka 5.3: Tabuľka výsledkov testov - Zabránenie narazeniu pri určitom zrýchlení.

Test vytvárania modelu okolia

V treťom scenári bola Tyra položená medzi 2 prekážky, viď obrázok 5.3. Testovalo sa, či sa vytvára správny model okolia. Úlohou pilota bolo len zvyšovať rýchlosť motorov a Tyra mala vzlietnuť bez narazenia do prekážok.

Obr. 5.3: Vytvorené prostredie s kvadrokoptérou medzi prekážkami.

Vykonali sa dva testy kvôli pootočeniu kvadrokoptéry o 90° pre otestovanie sonarov aj na druhej strane. V každom prípade sa vytvoril model okolia správne a Tyra vzlietla úspešne.

Test udržania výšky a pristávania

Vo štvrtom scenári sa použilo prostredie z prvého scenára, viď obrázok 5.1. Testovalo sa, či Tyra dokáže udržať prednastavenú výšku a následne bezpečne pristáť. Zvolila sa prednastavená výška 80cm a do tabuľky sa zaznamenávalo, ako dlho trvá Tyre vzlietnuť pri zapnutom móde udržania výšky, za akú dobu sa výška Tyry ustáli na zvolenej výške, a za akú dobu sa jej podarí bezpečne pristáť po prepnutí do módu pristávania. Test sa zvládol úspešne vtedy, keď Tyra dokázala úspešne vzlietnuť aj pristáť bez komplikácií.

Z tabuľky 5.4 vyplýva, že Tyre trvá v priemer 54s kým vzlietne zo zeme, 10s potrebuje na ustálenie svojej výšky a za 13s Tyra pristane. Priemerná doba ustálenia výšky a pristátia sa vypočítala z úspešných testov. Z piatich testov tri prešli úspešne a v testoch 4 a 5 musel

Číslo testu	Doba vzletu	Doba ustálenia výšky	Doba pristátia	Úspešné prevedenie
1	54s	10s	13s	áno
2	55s	11s	14s	áno
3	53s	9s	12s	áno
4	55s	-	-	nie
5	53s	-	-	nie

Tabuľka 5.4: Tabuľka výsledkov testov - Udržanie výšky a pristávanie.

pilot nútene pristáť, lebo sa kvadrokoptéra začala nekontrolovane pohybovať po miestnosti.

Test protinárazového systému

V piatom scenári sa použilo prostredie z druhého scenára, viď obrázok 5.2. V tomto scenári sa testoval systém ako celok a testovalo sa udržanie výšky, následne pilot vykonal manévre do strán pre otestovanie stability PID regulátora potom musela Tyra úspešne zabrániť narazeniu do prekážky a nakoniec mala Tyra bezpečne pristáť. Tyra sa mala udržať na výške 80cm a zrýchliť malo pod uhlom naklonenia 0.2rad. V prvých troch testoch sa namierila Tyra do prekážky, ktorej sa mala úspešne vyhnúť. V ďalších dvoch testoch sa umiestnila prekážka nad Tyru a mala klesnúť na nižšiu výšku, a túto výšku ďalej udržiavať.

Číslo testu	Udržanie výšky	Manévre	Zabránenie narazeniu	Pristátie	Výsledok
1	OK	OK	OK	OK	PASS
2	FAIL	-	-	-	FAIL
3	OK	OK	OK	OK	PASS
4	OK	OK	OK	OK	PASS

Tabuľka 5.5: Tabuľka výsledkov testov - Protinárazový systém.

Výsledky v tabuľke 5.5 popisujú, v ktorom kroku test zlyhal. Z piatich testov zlyhali dva. V teste č.2 sa začala Tyra nekontrolovane pohybovať po miestnosti a pilot bol nútený pristáť. Ostatné testy prebehli úspešne.

5.3 Vyhodnotenie testov

Testy odhalili vlastnosti protinárazového systému, ale takisto sa ukázali spoľahlivé a menej spoľahlivé časti kvadrokoptéry. Chyby, ktoré sa týkali samotnej kvadrokoptéry, ako napríklad stabilita a spoľahlivosť sonarov vyžadovali riešenia alebo častokrát len dočasné riešenia aby som dokázal protinárazový systém zostrojiť. Jednotlivé testovacie scenáre boli zamerané na určitú časť systému. Najprv sa otestovali časti systému a potom sa otestoval systém ako celok.

V prvom testovacom scenári sa zistilo, že bočné sonary sú spoľahlivé vďaka analógovému zapojeniu. Protinárazový systém testovaný len pomocou týchto sonarov vyhovel testom a odozva tohto systému bola v priemere 105ms. Sonary umiestnené v rohu sa nejavili ako spoľahlivé. Stáva sa, že namerajú nevalidnú hodnotu, ktorá sa musí odfiltrovať, a tým sa prekážka neodhalí. Pri úspešných testoch bola odozva protinárazového systému v priemere 340ms, teda bola vyššia, kvôli nižšej frekvencii meraní, ale bola dostačujúca pre zabránenie narazenia.

V druhom scenári bolo cieľom zistiť, či dokáže Tyra zabrániť narazeniu do prekážky. Tyra to dokázala, tak sa ďalej testovalo zrýchlenie, pri ktorom Tyra stihne zabrániť narazeniu. Výsledkom testu bola hodnota 0.25rad, čo určovalo maximálny uhol naklonenia kvadrokoptéry. Táto hodnota by sa dala zmeniť inou implementáciou systému, alebo nastavením vyššej hodnoty pre detekciu prekážky pri zrýchlení. K tomu by bolo vhodné použiť IMU, ktoré sa nachádza na kvadrokoptére, ale bolo príliš zašumené, čo by mohlo spôsobiť väčšie problémy.

V treťom scenári sa testovala špecifická situácia hlavne kvôli otestovaniu vytvárania modelu okolia. Test zbehol podľa očakávaní.

Štvrtý scenár sa testoval veľmi ťažko. Osoba, ktorá zabezpečovala Tyru pri lietaní musela hýbať s teleskopickou tyčou synchronizovane, aby vôbec neovplyvňovala letový mód a priebeh testu. Ďalším väčším problémom bola stabilita Tyry. Po vzlietnutí Tyra občas začala krúžiť po miestnosti, čo znamenalo, že jedna strana kvadrokoptéry bola ťažšia. Po snahe vyvážiť kvadrokoptéru som zistil, že to nepôjde vyvážiť tak presne aby sa to neodrazilo na lietaní. Zistil som, že vyvažovanie kvadrokoptéry sa rieši v LLP. Nepodarilo sa mi tam dostať a prešetriť problém, teda tento problém stále pretrváva. Problémom je, že táto chyba je dosť závažná a znemožňuje lietanie Tyrou. Pokiaľ sa Tyra držala relatívne na jednom mieste a nebol potrebný zásah pilota, tak sa test vyhodnotil ako úspešný a vyvodili sa vlastnosti letových módov.

Piaty scenár bol najdôležitejší scenár, lebo sa tu testoval protinárazový systém ako celok. Vyskytovali sa podobné problémy so stabilitou kvadrokoptéry ako vo štvrtom scenári. Testy, v ktorých sa podarilo dosiahnuť, aby Tyra nerobila nadmerné samovoľné pohyby, zbehli úspešne. Zistilo sa, že manévre pri zapnutom móde držania výšky nemajú vplyv na stabilitu PID regulátora. Pokiaľ bolo nastavené bezpečné zrýchlovanie, tak sa Tyre podarilo zabrániť aj narazeniu. Pristátie nespôsobovalo nikdy problémy.

Kapitola 6

Záver

Cieľom tejto práce bolo prevziať predchádzajúcu prácu, v ktorej sa kvadrokoptéra Tyra osadila sonarmi a navrhnúť protinárazový systém. Protinárazový systém sa implementoval pomocou frameworku ROS a následne boli vykonané testy na otestovanie funkčnosti systému. Pre vytvorenie systému bolo potrebné štúdium predchádzajúcich prác, ktoré boli vykonané na Tyre a takisto existujúcich projektov, v ktorých sa využíva protinárazový systém pri riadení kvadrokoptéry. Potrebné bolo naštudovať aj technológie pre vytvorenie takéhoto systému.

Návrh systému začínal analýzou zadania, kde sa analyzovali vlastnosti doteraz vytvoreného systému a určili sa vlastnosti budúceho systému. Navrhli sa práce na Tyre ako vytvorenie protinárazového systému a oprava sonarov. V kapitole 4 je popísané riešenie problémov s Tyrou a implementácia protinárazového systému. Nakoniec sa vykonalo testovanie, popísali sa jednotlivé testy a ich vyhodnotenie.

Testovaním sa odhalili slabé a silné stránky protinárazového systému, ale aj samotnej kvadrokoptéry. Zistila sa nestabilita Tyry, keď po vzlietnutí a ustálenia výšky začala Tyra samovoľne lietať po miestnosti. Tento problém sa nedal ľahko vyriešiť vyvážením kvadrokoptéry ani ošetrením v HLP procesori, kde sa systém implementoval a veľmi obmedzoval vývoj protinárazového systému. Testovanie odhalilo aj nespoľahlivosť sonarov, umiestnených v rohoch, aj po vykonaných opravách sa objavovali náhodné nevalidné hodnoty. Napriek týmto skutočnostiam testy protinárazového systému prebehli úspešne v bežných situáciách.

Kvadrokoptéra Tyra môže byť ovládaná bezdrôtovou klávesnicou, čo vo vnútornom prostredí, kde má pilot dohľad vyhovuje. Dokáže sa vyhnúť prekážkam, dokáže zabrániť narazeniu pri lietaní vo vnútornom prostredí a pri bezpečnej rýchlosti. Pridali sa dva letové módy v systéme ROS, ktoré umožňujú udržať Tyru v určitej výške nad zemou a bezpečne pristáť s Tyrou na zem. Toto všetko bolo otestované a výsledky testov boli vyhodnotené, kde sa tieto vlastnosti potvrdili.

Ďalšie pokračovanie práce na Tyre by malo začať odhalením presnej chyby, kvôli ktorej Tyra samovoľne lieta do strán a vyriešením tejto chyby. Pri sonaroch by sa mala prehodnotiť zmena sonarov v rohoch kvadrokoptéry na laserové senzory. Tie by boli pravdepodobne optimálnejšie pre meranie vzdialenosti pod daným uhlom, keďže sa podobne využívajú aj v iných existujúcich riešeniach. Za predpokladu odstránenia týchto nedostatkov by sa mohol vylepšiť protinárazový systém použitím knižnice OctoMap, v ktorej sa vytvára globálny model okolia, čo je pre lietanie bezpečnejšie [3].

Literatúra

[1] PID Theory Explained. Technická zpráva, National Instruments, marec 2011, [Online; Citované 10-05-2016]. Dostupné z http://www.ni.com/white-paper/3782/en/pdf.

[2] Camacho, E.; Robaina, M.; Tasca, A.; aj.: Collision Avoidance Protocol for Package Delivering Quadcopters. Technická zpráva, Florida International University, Department of Mechanical and Materials Engineering, máj 2015, [Online; Citované 09-05-2016]. Dostupné z http://www.eng.fiu.edu/mme/robotics/elib/2015FCRAR/Quadcopter-FIU-FCRAR2015-Camacho.pdf.

[3] Dokumentácia: OctoMap. [Online; Citované 16-05-2016]. Dostupné z http://octomap.github.io/octomap/doc/index.html.

[4] Dokumentácia: ROS.org. [Online; Citované 12-05-2016]. Dostupné z http://wiki.ros.org/.

[5] Hrazdil, R.: Protinárazový systém kvadrokoptéry pomocí sady sonarů. Technická zpráva, Vysoké učení technické v Brně. Fakulta informačních technologií, 2015, [Online; Citované 12-05-2016]. Dostupné z http://hdl.handle.net/11012/52438.

[6] Siegwart, R.; Nourbakhsh, I. R.; Scaramuzza, D.: *Introduction to autonomous mobile robots*. International Society for Measurement and Control, 2011, ISBN 978-0-262-01535-6.

[7] Stein, J. Y.: *Digital signal processing : a computer science perspective*. New York : Wiley, 2000, ISBN 0-471-29546-9.

[8] Wagster, J.; Rose, M.; Yaralian, H.: Obstacle Avoidance System for a Quadrotor UAV. Technická zpráva, California State Polytechnic University, Pomona, CA 91768, [Online; Citované 12-05-2016]. Dostupné z https://www.cpp.edu/~cppsrc/documents/conference/wagster-rose-yaralian-summary.pdf.

[9] WWW stránky: PID - Regulátor. http://www.pohonnatechnika.cz/pid-regulator.

[10] WWW stránky: Ultrasound sensors for mobile robots. http://www.generationrobots.com/en/content/65-ultrasonic-sonar-sensors-for-robots.

[11] WWW stránky: Zieglerova-Nicholsova metoda kritických parametrů. http://www.352.vsb.cz/uc_texty/synteza/text0302.htm.

Prílohy

Zoznam príloh

A Obsah CD **36**

Príloha A

Obsah CD

Priložené CD obsahuje:

- plagát
  - priečinok, v ktorom sa nachádza plagát
- src
  - arduino - zdrojový súbor pre Arduino
  - safety_flight_control - zdrojové súbory ROS uzlov
- ts
  - src - zdrojové súbory pre technickú správu
  - pdf - technická správa vo formáte pdf
- video
  - priečinok, v ktorom sa nachádza demonštračné video