ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA DOPRAVNÍ

Bohuslav Vladyka

NÁVRH ANTI-KOLIZNÍHO SYSTÉMU

Bakalářská práce

2021

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta dopravní
děkan
Konviktská 20, 110 00 Praha 1

K621 ..... **Ústav letecké dopravy**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Bohuslav Vladyka

Kód studijního programu a studijní obor studenta:

B 3710 – LED – Letecká doprava

Název tématu (česky): **Návrh anti-kolizního systému**

Název tématu (anglicky): Design of Anti-collision system

Zásady pro vypracování

Při zpracování bakalářské práce se řiďte následujícími pokyny:

- Cílem práce je vytvořit anti-kolizní systém použitelný pro létání multikoptéry kolem tělesa, které je umístěné ve volném prostoru, s následnou validací funkcí systému v simulačním prostředí Gazebo.
- Vypracujte analýzu současného stavu využití bezpilotních letadel ve vnitřních prostorách se zaměřením na používané anti-kolizní systémy.
- Navrhněte anti-kolizní systém, který bude schopný detekce překážky v prostoru a následně validujte jeho funkčnost.
- Na základě používaných dat navrhněte set senzorů, kterými bude nutné vybavit UAS pro aplikaci navrhovaného systému.
- Formulujte závěry práce.

Rozsah grafických prací: podle pokynů vedoucího práce

Rozsah průvodní zprávy: minimálně 35 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury:
Lentin Joseph and Jonathan Cacace. 2018. Mastering ROS for Robotics Programming - Second Edition: Design, build, and simulate complex robots using the Robot Operating System (2nd. ed.). Packt Publishing.

Regtien, P. P., & Dertien, E. (2018). Sensors for mechatronics. Elsevier.

Vedoucí bakalářské práce: **Ing. Stanislav Kušmírek**

Datum zadání bakalářské práce: **9. října 2020**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání bakalářské práce: **9. srpna 2021**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia

L. S.

doc. Ing. Jakub Kraus, Ph.D.
vedoucí
Ústavu letecké dopravy

doc. Ing. Pavel Hrubeš, Ph.D.
děkan fakulty

Potvrzuji převzetí zadání bakalářské práce.

Bohuslav Vladyka
jméno a podpis studenta

V Praze dne 9. října 2020

Poděkování

Na tomto místě bych rád poděkoval Ing. Stanislavu Kušmírekovi za odborné vedení práce a za cenné rady, které mi byly poskytovány po celou dobu psaní bakalářské práce. Dále bych chtěl poděkovat panu Payamu Farajiani za nasměrování při potížích s instalací systému. V neposlední řadě je mou milou povinností poděkovatcelé mé rodině a blízkým přátelům za podporu, které se mi dostávalo během celé doby studia.

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci, která byla zpracována v rámci závěru studia na ČVUT v Praze Fakultě dopravní.

Tímto prohlašuji, že předložená práce byla vypracována samostatně a veškeré použité zdroje jsou uvedeny v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 9. srpna 2021 .....

podpis

Abstrakt

V posledních letech se rapidně rozšiřují možnosti využití bezpilotních prostředků. S rostoucím trendem vzniká větší důraz na bezpečnost v provozu. Podchycení bezpečnosti je nutné realizovat již během vývoje. Tato práce věnuje pozornost vývoji anti-kolizního systému bezpilotního prostředku. Validace probíhá v simulačním prostředí Gazebo. Výzkumná část tvoří instalaci náležitého softwaru pro detekci překážek, výběr senzoru, úprava PID kontroleru s následnou validací a pohyb bezpilotního prostředku v simulátoru. Důrazem výzkumu je detekce překážek, nikoliv tvorba samotných objektů. V průběhu detekce je dynamicky upravována vzdálenost blížícího se bezpilotního prostředku k překážce. Analýza signálu je zpracována do simulinkového modelu PID kontroléru.

Klíčová slova: detekce překážky, bezpilotní prostředek, plánování letu

Abstract

In the last few years the possible ways of using unmanned aerial vehicles have been rapidly expanding. With the growing trend, there is a bigger emphasis placed on operational safety. It is necessary to provide safety already during development. This work pays attention to the development of an anti-collision system of an unmanned aerial vehicle. Validation takes place in the Gazebo simulation environment. The research part creates the installation of appropriate software for obstacles detection, sensor selection, modification of the PID controller with following validation and movement of the unmanned vehicle in the simulation. The emphasis of the research is the detection of obstacle, not the creation of the objects themselves. During detection, the distance of the approaching unmanned vehicle to the obstacle is dynamically managed. The signal analysis is processed into a simulation model of the PID controller.

Keywords: obstacle detection, unmanned aerial vehicle, flight planning

Obsah

Seznam použitých zkratek 7

Úvod 8

1. Analýza současného stavu 9

1.1. Definice UAV a legislativní omezení 10

1.2. Využitelnost UAV 13

1.2.1. Využití ve venkovních prostorách 14

1.2.2. Využití ve vnitřních prostorách 17

1.2.3. Speciální prostor 19

1.3. Vnitřní poziční systémy 20

1.3.1. Vyvíjené technologie 22

1.3.2. Řízení a kontrola 23

1.3.3. PID kontrolér 24

1.4. Testovací prostředí a implementace testů letu 26

2. Použité prostředky a metody 28

2.1. LINUX 28

2.2. ROS 29

2.3. Gazebo 31

2.4. MAVLink 32

2.5. Pixhawk 4 a kódování v Pythonu 33

3. Výsledky a zpracování 34

3.1. Instalace systému 34

3.2. Výběr senzoru 39

3.3. Úprava hodnot PID kontroléru – ROS 40

3.4. Validace hodnot PID kontroléru – Simulink 41

3.5. Umístění bezpilotního prostředku do programu Gazebo 43

3.6. Konfigurace celého bezpilotního prostředku 50

3.7. Shrnutí kapitoly 51

Závěr 53

Použité zdroje 55

Seznam obrázků 63

Seznam tabulek 64

Seznam použitých zkratek

GPS	Global Positioning System
UAV	Unmanned Aerial Vehicle
RPAS	Remotely Piloted Aircraft Systems
GNSS	Global Navigation Satellite System
IoT	Internet of Things
ROS	Robot Operating System
UWB	Ultra Wide Band
GNC	Guidance, Navigation, and Control
GCS	Ground Control Station
ATIS	Automatic Terminal Information Service
IMU	Inertial Measurement Unit
ÚCL	Ústav Civilního Letectví
PSO	Particle Swarm Optimization
WLAN	Wireless Local Area Network

Úvod

Provoz bezpilotních prostředků prochází obdobím, kdy je začleňován do mnoha různých oborů. Bezpečnostní složky využívají bezpilotních prostředků na místech, kde je špatně dostupný terén, při sledování ilegální činnosti či při monitoringu dálničních koridorů. V civilním sektoru jsou využívány například v zemědělství, ale také při kontrole technického stavu objektů. Zvyšující se dostupnost bezpilotních systémů láká zejména širokou veřejnost pro možnost pořízení snímků a natáčení pomocí zabudované kamery. Vyšší využitelnost bezpilotních prostředků si žádá větší pozornost již během vývoje, primárně kvůli zvýšení rizika kolize. Nedílnou součástí během vývoje je důraz na softwarový vývoj anti-kolizního systému bezpilotního prostředku. Ten tvoří řídící jednotku bezpilotního prostředku.

Motivací pro vypracování této bakalářské práce je rychle rostoucí trend vývoje technologií bezpilotních prostředků s absencí vývojových metodik pro vytvoření systému, který dokáže včas detekovat blížící se překážku. Absence je způsobena tím, že rychlý růst neodpovídá potřebě využití bezpilotních prostředků. Kromě venkovního využívání bezpilotních prostředků, vstupují na scénu bezpilotní systémy s vnitřním použitím. Indoor používání sebou přináší vyšší zásady pro dodržení bezpečnosti, ale rovněž absenci družicové navigace. Tento faktor demonstruje směr výzkumu, kterým by se měl nadále ubírat. Tento pohled tkví ve vývoji anti-kolizních systémů uvnitř objektů.

Cílem této bakalářské práce je vytvoření anti-kolizního systému, který bude umožňovat včasnou detekci překážky ve volném prostoru. První část se zabývá analýzou současného stavu, jaké jsou provozní možnosti a co vše je možné testovat. V následující části je popsána instalace systému ROS, výběr senzoru, modelování signálu PID kontroléru a následná validace signálu v Gazebo. Validace je provedena detekcí překážky ve vnitřních prostorech. Překážka je vložena do grafického rozhraní Gazebo. Na konci této části je navržena sada senzorů a jejich umístění.

1. Analýza současného stavu

Technologický pokrok vede k rozsáhlé implementaci bezpilotních prostředků v široké oblasti působnosti. Lze tvrdit, že uplatnění bezpilotních prostředků můžeme nalézt v různých oborech. Vzrůstající použitelnost plyne z jednoduché obslužnosti prostředků, protože není potřebné využívat lidského zdroje přímo v kokpitu bezpilotního prostředku. Pilot operuje v kontrolních pozemních stanicích, odkud ovládá pohyb bezpilotního prostředku. Ovládání bezpilotního prostředku je omezené pouze na technické parametry. V současném provozu je počítáno s využitím bezpilotního prostředku pro vojenský dohled, služby SAR, letecké snímkování či konstrukční práce. Americké ministerstvo obrany však došlo k závěrům, že malá bezpilotní letadla mají obrovskou poruchovost, která je vyčíslená na 10^{-2} selhání za jednu letovou hodinu. Takováto poruchovost neodpovídá požadavku federálního leteckého úřadu Spojených států (FAA), který ji stanovil na 10^{-9} selhání. Tenhle důvod je jeden z mnoha důvodů, proč státní agentury a výzkumníci upustili od používání bezpilotních prostředků. Selhání odpovědnosti, spolehlivosti a obava ze ztráty soukromí nastává hlavně při letu na dlouhé vzdálenosti. Při dlouhých vzdálenostech se zvyšuje pravděpodobnost, že signál bude ztracen. Mimo pokrytí signálu z pozemních stanic se stává bezpilotní prostředek neřiditelným a může dojít i k následné havárii. Je zapotřebí brát v úvahu i zmocnění bezpilotního prostředku neoprávněnou osobou či teroristický útok. V takových případech je těžko identifikovatelný bezpilotní letoun, protože radar těžko zaznamenává malé plochy. [1, 2, 3]

Velké množství UAV je zejména používáno k civilním účelům. Takové systémy nemají detekci poruch nebo černé skříňky. Cílem pro mnohé výzkumy je vytvoření systému, který dokáže dokonale trasovat leteckou cestu UAV, sledovat letecké podmínky, detekovat překážky a vyhnout se jim. Detekce překážek a následné zabránění kolize je klíčovým tématem této práce. Vývoj klade důraz na zásah autopilotních podpůrných systémů v případě selhání operátora. Než se začneme zabývat použitím a technologiemi, je důležité si definovat bezpilotní prostředek, tedy UAV.

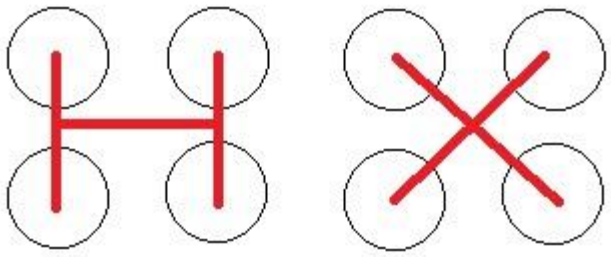
1.1. Definice UAV a legislativní omezení

Definice podle FAA popisuje UAV – Unmanned Air Vehicle – jako letecký prostředek bez lidské posádky. V české terminologii je bezpilotní letoun definován podle normy ČSN 31 0001 jako letadlo způsobilé létat bez pilota, které je za letu řízené automatickým zařízením nebo dálkově ze země. [4]

Dle obou definic, řízení dopravního prostředku probíhá dálkově nebo autonomně. Mezi speciální funkce bezpilotního prostředku z výzkumů patří režim návratu domů, senzory pro přiblížení a senzory pro zabezpečení proti selhání. Autonomní řízení se skládá z předem naprogramovaných algoritmů a letových plánů. UAV je možné brát i jako pojem, ve kterém je zakomponována konstrukce, velikost či specifické letecké vlastnosti. Integrace aktivních prvků systému funguje s propracovaným IPS. Bezpilotní prostředek je komplexní systém, který je složen z hardwaru a softwaru. Elektronické zlepšení povoluje vývoj navigace a kontrolních systémů. Hlavní komponenty jsou rozděleny do tří kategorií:

- letová plošina,
    - drak
    - senzorové vybavení
    - navigační systém
    - kontrolní systém
    - energetické systémy včetně zdroje
- pozemní stanice,
- komunikační systém.

Součástí letové plošiny je zahrnutý drak, navigační a senzorický systém, kontrolní systémy, systém zdroje a distribuce energie. Konstrukce draku je hlavní strukturou bezpilotního prostředku. Mezi nejpoužívanějšími typy konfigurace rotorů je konstrukce typu „H“ nebo „X“ (Obrázek 1). [3, 5]



Obrázek 1: Konstrukční typy bezpilotních prostředků [58]

Tyto konstrukce jsou navrženy tak, aby během letu odolávaly momentům síly. V konstrukci musí být zamezeno účinkům deformace během letu a také vibracím v celé konstrukci. Pro výrobní účely se používá hliník nebo uhlíková vlákna. Celková struktura draku bere v úvahu hmotnost, zejména vůči výkonu a palubním systémům. Počet vycházejících ramen závisí na tom, jaké je zapotřebí mít užitečné zatížení a počet motorů. Díky užitečnému zatížení, které je součástí draku, je umožněno létání a také schopnost nést různá zařízení a senzory. K pohánění hnacích jednotek se využívá energetický systém. V dnešní době se nejčastěji setkáváme s energetickým systémem, který využívá lithium-polymerových baterií. Kromě pohánění hnacích sil se energetický systém využívá k napájení senzorů a zařízení zavěšených na bezpilotním prostředku. V případě poškození nebo vyčerpanosti energetického systému nedochází k plnohodnotné obsluze senzorů, ale také dochází ke zhoršení komunikace s pozemní stanicí. Operátor z tohoto místa kontroluje a analyzuje pohyb bezpilotního prostředku. [5]

Komunikace je zajištěna skrz komunikační systém. Tato komunikace probíhá bezdrátovým přenosem dat. Při komunikaci se zprostředkovává navigace bezpilotního prostředku. Navigační systém je vybaven autopilotem, který umožňuje provádět autonomní nebo poloautonomní lety. Pro autonomní lety je důležité testování anti-kolizních systémů, které dokážou zabránit kolizím včas. Testování probíhá při reálném provozu, ale také v prvopočátku. V prvopočátku je vhodné si toto prostředí definovat pomocí softwarových možností. To může vést ke snížení celkových škod. Důležitým aspektem pro vytvoření optimálních metod je znalost odvětví, kde se může využít testování pomocí softwaru. [5, 2]

Pozemní stanice jsou součástí bezpilotních systémů. Stanice tvoří řídící objekty na zemském povrchu a jsou osazeny vysílačem a přijímačem. Pomocí vysílače a přijímače je zprostředkována komunikace mezi bezpilotním prostředkem a operátorem. Kromě vysílače a přijímače zahrnuje stanice také počítač, telemetrická zařízení a ovládací panely datových přenosů. Mobilní bezdrátová stanice zprostředkovává dálkové ovládání. Softwarové rozhraní mobilních stanic může poskytovat, kromě stavových parametrů baterie, také obrazový přenos z kamery. [5]

Klasifikace bezpilotních prostředku

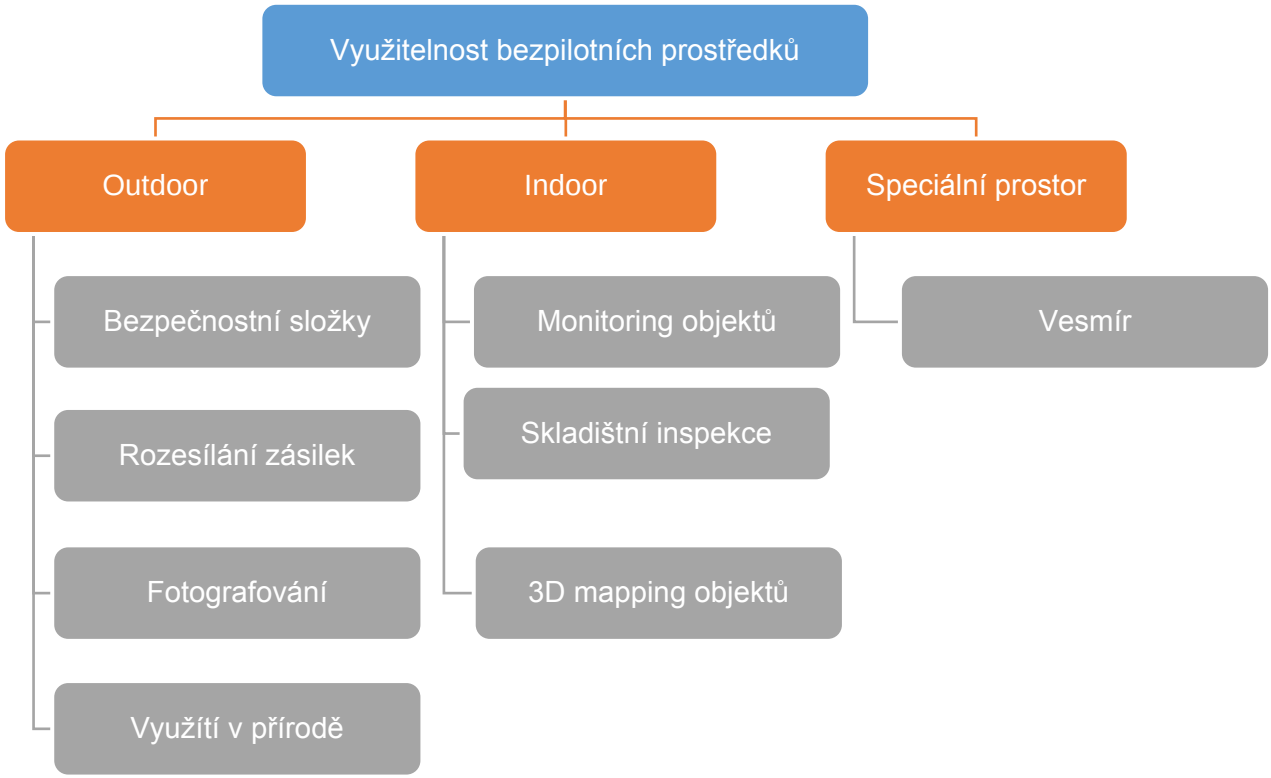
Bezpilotní prostředky jsou klasifikovány z různých hledisek. Jako hlavní rozdělení je definováno dle využití na vojenské a civilní. Rozdělení dle používání má dost široký záběr, tedy je nutné rozdělit bezpilotní prostředky trochu podrobněji. Pro další dělení je důležitá analýza klasifikačních aspektů. Mezi klasifikační aspekty patří hmotnost, doba letu, celkový dolet a rychlost. Po přehodnocení klasifikačních aspektů je možné zpracování podle výkonnostních charakteristik. Evropská vyhláška 2019/247, která nabrala platnosti 30. 12. 2020, stanovuje rámcová opatření pro bezpečný provoz bezpilotních prostředků. Vyhláška přijímá přístup založený na riziku a nerozlišuje účel létání. Bere v úvahu právě výkonnostní charakteristiky bezpilotních prostředků. Legislativní rozdělení zahrnuje tři kategorie – otevřená, specifická, certifikovaná. Kategorie, u které není vyžadováno předchozí povolení příslušného úřadu, ani prohlášení provozovatele UAV před uskutečněním provozu. Tato kategorie však podléhá pravidlům pro létání ve vzdušném prostoru. Nesmí dojít k žádnému omezení vyplývajícím z uspořádání vzdušného provozu. Průkazné znalosti, z chování ve vzdušném prostoru pro piloty bezpilotních prostředků v otevřené kategorii, jsou součástí online testu. Online test je zajištěn a zpracován úřadem civilního letectví. Do otevřené kategorie spadají stroje do 25 kg, které mají omezení pro letištní plochy a zakázané zóny, jejich maximální dosažitelná rychlost je 19 m/s. Vzdušný pohyb je omezen do maximální dosažitelné výšky 120 m, létání probíhá pouze na dohled operátora, jehož minimální věková hranice musí být alespoň 16 let. Pokud operátor nedosahuje věkové hranice, musí létat pouze v doprovodu registrovaného pilota. Otevřená kategorie je dělena do dvou subkategorií. Subkategorie C se týká fyzických vlastností bezpilotního prostředku a subkategorie A specifikuje provozní omezení.

Specifická kategorie se týká pilotů, kteří překračují hranice otevřené kategorie. Pro splnění podmínek této kategorie je důležité si zpracovat analýzu celkových rizik letu – SORA a dále je zapotřebí podat analýzu na patřičný úřad. Pro operace ve specifické kategorii musí dojít i ke schválení oprávnění patřičným úřadem, v případě České republiky je toto opatření vydané Úřadem pro civilní letectví. Součástí tohoto oprávnění je pojištění bezpilotního prostředku, scénáře provozu, identifikační štítek a testování letu na ÚCL. Praktické zkoušky zprostředkované ÚCL může splnit každý pilot, který dovršil 16 let. V provádění letů mimo Českou republiku je nutné si zajistit povolení i od členského státu. [6, 7]

Jako nejrizikovější kategorie, která počítá s masivním nárůstem bezpilotních prostředků do provozu, je certifikovaná kategorie. Certifikovaná kategorie klade důraz na nejvyšší možné zajištění bezpečnosti. Pokládá požadavky na provozovatele a piloty, směřuje k požadavkům malých a ultralehkých letadel. Kategorie vyžaduje osvědčení pro techniku i personál letadla. Kompletní osvědčení je bráno jako proces certifikace. Proces certifikace je zahájen již během výroby bezpilotního prostředku. Pilot včetně personálu taktéž prochází procesem certifikace. Celý proces certifikace je zajištěn příslušným státním orgánem. V případě České republiky je to Úřad pro civilní letectví. [7]

1.2. Využitelnost UAV

Rozdělení bezpilotních prostředků je možné na základě mnoha parametrů. Jedním ze základních parametrů dělení je podle oblasti, ve které se bezpilotní prostředek pohybuje. Oblasti působnosti, kde se může bezpilotní prostředek pohybovat, se dají rozdělit do tří kategorií. Kategorie jsou definovány jako outdoor, indoor a speciální prostor (Obrázek 2).



Obrázek 2: Kategorizace bezpilotních prostředků

1.2.1. Využití ve venkovních prostorách

Prioritním prostředím pro využívání bezpilotních prostředků je exteriér. Dle legislativních nařízení EU 2019/945 a EU 2019/947 jsou stanoveny základní požadavky na projektování a výrobu bezpilotních systémů v rámci vzdušného prostoru. Rychlý vývoj v rozvoji bezpilotních prostředků, které jsou řízeny autonomně, umožňuje implementaci bezpilotních systémů do velkého množství odvětví. Podle certifikované kategorie, která počítá s masivním nárůstem bezpilotních prostředků v následujících letech, bude možné využívat bezpilotní prostředky i k transportu osob. Vývoj musí počítat s rozsáhlejším testováním bezpilotních prostředků i mimo provoz v reálném čase. Evoluční změny, které přichází skrz nová odvětví, patří zejména armádnímu průmyslu. Používání bezpilotní prostředků je využíváno ve venkovních prostorách a za podmínek, které jasně definují, že bezpilotní prostředek nesmí omezit jiné účastníky vzdušného prostoru. [6, 7, 8]

Bezpečnostní složky

Z hlediska vývoje bezpilotních prostředků je nejdůležitějším odvětvím armáda. Armádní použití bezpilotních prostředků patří mezi nejvýznamnější. V počátcích éry bezpilotních prostředků se USA a Velká Británie staly průkopníky ve vývoji. První bezpilotní prostředek sloužil ke špionážním misím. Vývoj v současnosti cílí na vybavení. Dnešní bezpilotní prostředky řízené armádou jsou osazeny například termovizí, laserovým zaměřovačem i nástroji pro provedení náletů. Nejznámějším bezpilotním prostředkem, využívaný armádou, je bezpochyby MQ – 9 Reaper (Obrázek 3). Jedná se o bezpilotní letoun, který je 11 metrů dlouhý a má dolet až 1852 kilometrů. Bezpilotní prostředek Reaper vede nezjistitelné letové operace ve výšce až 15 kilometrů. Zásadní vývojovou myšlenkou armádních bezpilotních letounů je provedení mise a následný návrat zpět na základnu. Za touto myšlenkou stojí testovací lety, které jsou v prvopočátku zprostředkovány pomocí počítačových simulací.

Obrázek 3: MQ - 9 Reaper [59]

Počítačové testování se zabývá nejen testováním jednoho bezpilotního prostředku. Existují strategické plány, kde je vhodné využití kvantity. Studie rojů statických bezpilotních prostředků začaly v 80. letech minulého století. S výzkumem dynamického roje bezpilotních prostředků se začalo spekulovat až počátkem 90. let, ale nejvíce populární se roje staly až před pár lety. Taktické rojové plány vedou k efektivitě a nižšímu riziku, že mise nebude splněna. Díky pokročilým metodám lze využít softwaru a výpočetního výkonu k rojovému letu. Rojový let zahrnuje několik výhod, když se jeden bezpilotní prostředek ztratí, tak zbytek roje je nadále schopen vykonat misi. Centrálně je tedy možné řídit a konfigurovat let bezpilotních prostředků ve formaci. U testování rojového letu je klíčová implementace testování z důvodu zvýšení efektivity a snížení potenciálních rizik nárazů mezi jednotlivými bezpilotními prostředky. Strategie plyne z přírodních vzorců letu roje ptactva či mobility mravenčí kolonie. [9, 10]

Záchranné složky, jako jsou hasiči nebo zdravotníci, využívají bezpilotních prostředků v oblastech, kde není bezpečné povolávat lidi do akce nebo v oblastech, které jsou špatně přístupné i pro helikoptéry. Bezpilotní prostředky jsou schopny zasahovat v kritických oblastech, jako jsou například lavinové oblasti, oblasti se sopečnou činnosti či v hornatém terénu. Příkladem nasazení bezpilotního prostředku ve velmi nepříznivých podmínkách je K-MAX, který byl navržen společností Kaman a bez problému dokáže nést i 2,5 tuny nákladu. K-MAX je nasazován na velké lesní požáry, kde využívá payload jako obrovské nádrže na vodu. Kromě záchranných misí, mohou bezpilotní prostředky transportovat zdravotnický materiál v obtížně přístupných oblastech nebo k velmi rychlému dodání orgánů pro transplantaci. Dodáváním lidských

orgánů se zabývala univerzita v Marylandu. Aby bylo možné provádět dodávky s orgány, musí být systém testován pomocí softwaru, kde jsme schopni změřit dobu letu. Softwarové testování klade důraz na přesnost a rychlost. [11]

Rozesílání zásilek

Odvětví, které je aktuálně v průběhu testování, bere zřetel na rychlost doručování potravin, balíků či jiných specifických zásilek. Bezpilotní prostředky určené k doručování jsou vhodné pro koncové doručení přímo v oblasti. Doručovací multikoptéry nesou označení “last mile”. Využití je tedy vztaženo na velmi krátké vzdálenosti, kde je předpoklad udržitelnosti teploty potravin. Myšlenka doručování uvažuje nad tím, že by bezpilotní prostředek vzlétl pouze z lokálních skladů a obchodů. Jedná se tedy o alternativní řešení přepravy zásilek, které zajistí rychlé dodání (Obrázek 4). V současnosti může nést bezpilotní prostředek až 25 kg poštovní balík. Takové množství je podle vědeckých studií dostačující pro běžný nákup. Technologie doručování jsou testovány společnostmi jako je například Amazon, Walmart a Google. Doručovací bezpilotní prostředky jsou součástí legislativního ustanovení certifikované kategorie. [1,13]

Obrázek 4: Doručovací bezpilotní prostředek Amazonu [60]

Fotografování

Provozování bezpilotního prostředku s kamerovým systém je v této době zcela běžné. Malé bezpilotní prostředky umožňují pořizování kvalitních video záznamů i pro amatérskou sekci fotografů. Společnost DJI se věnuje výrobou bezpilotních

prostředků s osazením kamery. Kromě vyvíjení bezpilotních prostředků se DJI stará o vývoj testovacího prostředí. Simulace počítá s vnějšími vlivy, které je možné si vyzkoušet. Důležitým aspektem pro fotografické bezpilotní prostředky je testovací vývojové prostředí, které zamezuje vzniku reálných škod, protože během testování softwarem je možné odladit chyby PID kontroléru. [12]

Využití v přírodě

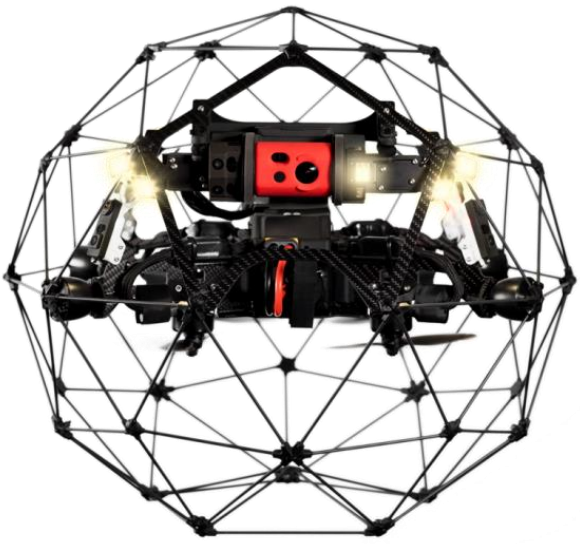
Alternativním a levnějším řešením pro ochranu divoké zvěře je aplikace bezpilotních prostředků do pozorování fauny. Sledování zvířecí populace na zemi je téměř nemožné. Zvířata dokážou posoudit nebezpečí a mohou se chovat jinak, než pokud jim nehrozí nebezpečí. Alternativní řešení přináší pozorování zvěře z oblohy. Díky této technologii je možné sledovat pohyby skupin zvířat. Obzvlášť v Africe a na asijském kontinentu jsou bezpilotní prostředky využívány jako dokonalé nástroje proti pytláctví. Další využití je při zalesňování, kdy bezpilotní prostředek provádí nálety v oblastech po rozsáhlých lesních požárech a rozsévá semena nových stromů. Využití bezpilotních prostředků nalézá svoje uplatnění taktéž při pořizování snímků a vytváření 3D mapování. 3D mapování klade důraz na výkonnou 4K kameru, značnou výdrž baterie a autonomní let. Autonomní let zajišťuje přesnější letové cesty prostřednictvím aplikace Litchi. Testování bezpilotních prostředků v přírodě se jednoznačně zabývá i testováním ve vývojovém prostředí. Tyto testy zamezují vzniku případných chyb. [14]

1.2.2. Využití ve vnitřních prostorách

Rychlý rozvoj bezpilotních prostředků se schopností autonomního řízení a prostorové orientace umožňuje aplikace v průmyslových oblastech, jako je například právě monitoring objektů, inspekce skladišť a 3D mappingu objektů. Důrazem, kladený na vývoj, indoorového bezpilotního prostředku je schopnost nést senzory dle vykonávaného úkolu. Senzory jsou akustické, vizuální, biologické či chemické. Kromě senzorů je důležité věnovat se testovacímu prostředí během vytváření komplexního systému. Díky testovacímu prostředí zamezíme kolizím před jejich vznikem. Aktuálně běžící systémy, které jsou uzpůsobené pro indoorové využití, jsou pouze proprietárně vyvíjeny. Nelze je tedy brát jako sjednocené. Úkolem výzkumu je definice přesnosti pozice. Přesnost pozice musí být v řádech centimetrů, spíše však k milimetrům. Taková přesnost nás vede k myšlence, že parametry družicové navigace jsou velice nepřesné pro určení polohy uvnitř budovy. [15, 16]

Monitoring objektů

Jak již je výše zmíněno, vývoj indoor bezpilotních prostředků klade důraz na schopnost nést senzory podle zadané úlohy. Ke zvýšení výkonu a efektivity byl navržen vědci design, který napomáhá k optimalizaci. Projekt Aeryon Scout byl vyvinut roku 2013 a zabývá se využitím bezpilotních prostředků v oblast důlních pracích. Aeryon Scout používá pro orientaci bodové mračno v trojrozměrném prostoru. U Aeryonu byla baterie přesunuta na vrchní část bezpilotního prostředku, tento krok vedl, ke zvýšení kapacity užitečného zatížení. Aeryon byl vybaven GPS, sonarovým systémem, tlakoměrem, teplotním senzorem, tříosým magnetometrem a gyroskopem. Bezpilotní prostředek byl připojen k pozemní základně pomocí rádia s dosahem až tří kilometrů. Hlavní myšlenkou Aeryonu spočívá v používání bezdrátové komunikace mezi horníky pomocí aplikace SkyHelp. Dalším výzkumem v důlních prostorách je Elios 2 (Obrázek 5), který je zároveň proprietárním řešením. Elios 2 je vytvořen firmou Flyablity a pomocí něho jsou prováděny důlní inspekce. [17, 18]



Obrázek 5: ELIOS 2 - důlní bezpilotní prostředek [18]

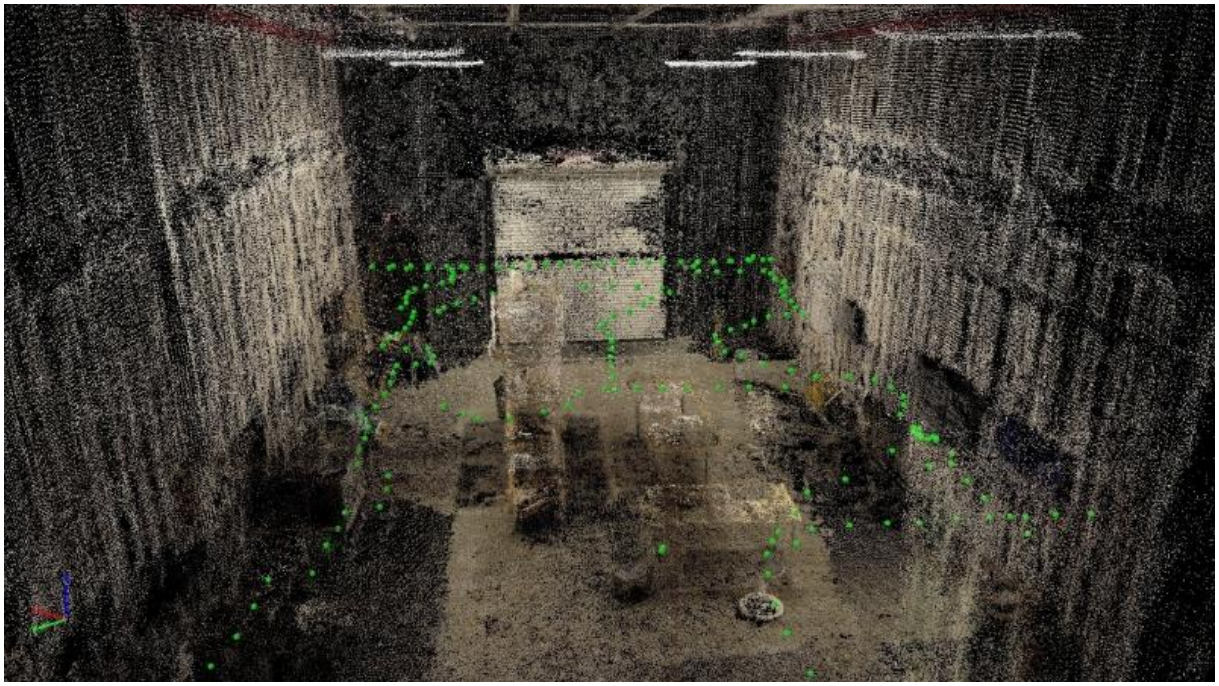
Skladištní inspekce

Kontrola skladových zásob pomocí bezpilotních prostředků má hned několik výhod oproti manuální kontrole. Výhody skladištní inspekce pomocí bezpilotních prostředků šetří čas pracovníků, ale také šetří finanční náklady na vysokozdvižné vozíky a zamezuje riziku úrazu. Většina aktuálně vyvíjených bezpilotních prostředků však není vhodná pro použití. Důvodem je provoz bezpilotních prostředků v nepřehledném prostředí, který je náročný i pro vyškolené piloty. Řešením inventarizačních úkolů,

je právě plně autonomní systém. Venkovní prostor umožňuje autonomnímu systému plynulý pohyb bezpilotního prostředku, ale pro vnitřní pohyb je provoz velmi náročný, jelikož bezpilotní prostředek musí být nejen schopen skvěle manévrovat v úzkých a dlouhých uličkách mezi regály ale také zároveň musí být schopen identifikovat čárový kód. Součástí výzkumu byl využit rozšířený Kalmanův filtr, který dokáže obsluhovat senzory s nízkým nákladem. Než však došlo k testování ve skladu, bylo nutné definovat vhodné vstupní podmínky pro PID kontrolér. [19, 20]

3D Mapping Objektů

Fotogrammetrie je používána zejména ve venkovních prostorech. Technologie jsou využitelné i uvnitř objekt. Technologie využívá generované bodové mračno k definování celého prostoru v bodech. Ve spolupráci Flyability a Pix4D byl přidán senzor fotogrammetrie na bezpilotní prostředek Elios 2 (Obrázek 6). Tento krok otevírá komplexní přístup pro modelování 3D vnitřních prostorů pomocí fotogrammetrie. Hlavní proces zpracování 3D modelů probíhá v Pix4Dmapperu. Pix4Dmapper se využívá jako industriální nástroj na vytváření prostředí. [18]



Obrázek 6: Pořízení 3D mapping fotografie z Elios 2 [18]

1.2.3. Speciální prostor

Vesmír

Nejzásadnější dopad na vývoj bezpilotních prostředků má pohyb bezpilotního prostředku ve vesmíru. Vesmírné odvětví patří i mezi nejutajovanější odvětví

využitelnosti. Vývojem se zabývá NASA a americká Air Force, kteří tajně testují bezpilotní prostředky pro kosmické lety. Vesmírný průmysl je nákladný po finanční stránce, je tedy nutné vyvíjet i testovací prostředí, než bude samotný let umožněn. Kosmický let Boeingu X-37, který se podobá miniaturnímu raketoplánu, tvoří rekord v nejdelším letu bezpilotního letadla. Tento rekord je přes 719 dní. Letectvo Spojených států amerických naznačuje, že hlavní cíle pro X-37B spočívají v opakovaně využitelných technologiích kosmických lodí. Využité technologie X-37B byly pouze pro oblet Země, existují však projekty, které se zabývaly i letem na okolní tělesa. Jeden z projektů SpaceX se zabýval konceptem návratů vzorků z Marsu. Tento koncept se postupem času změnil, ale základní myšlenka zůstala stejná. A to využít kapsli Red Dragon k testování bezpilotních technologií na Marsu. Roku 2017 byl vývoj zrušen z důvodu zastavení vývoje propulzního přistání ve prospěch optimálnější techniky pro přistání. Budoucnost průzkumu vesmíru je tedy kladena na vývoj přistávacích systémů, které umožní bezpečně přistát na povrchu. Během přistání nesmí dojít k poškození kosmického prostředku. Co se týče výzkumných prací na okolních planetách, tak za zmínku stojí i vesmírný bezpilotní prostředek Dragonfly, který umožňuje vertikální vzlet a přistání na měsících planet. Projekt Dragonfly je vyvíjen pro přistání na Titanu, což je měsíc Saturnu, a následném průzkumu biologie v půdě. Dragonfly má v přistávacím podvozku zabudované vrtáky, díky kterým dokáže vrtat do povrchu a dostat se tak přes hydrokarbonové podlaží k zmrzlé vodě. [21, 22]

1.3. Vnitřní poziční systémy

Díky výše zmíněné kategorizaci je důležité zhodnotit výhody a nevýhody prostředí, ve kterém se bezpilotní prostředek pohybuje. Velkým nedostatkem vnitřních prostorů je právě absence navigačních zařízení. Absence navigace klade důraz na bezpečnost vůči okolním objektům a lidem. Proto se cíl bakalářské práce se zaměřuje na definování indoor testovacího prostředí, které bere v potaz pohyb ve vnitřních prostorech. Menší prostory a potřeba zajištění nepoškození majetku vyžadují precizní přesnost. Řádově se jedná o centimetry, to je o dva řády vyšší přesnost, než je schopna zajistit družicová navigace. S odepřením družicové navigace vznikají problémy, které se dotýkají definicí rizik. Rizika, spojena s vnitřním prostorem, jsou odlišného charakteru než při pohybu v externích prostorech. Proto se tomuto tématu věnuje velké množství studií, založených na pokrokových technologiích.

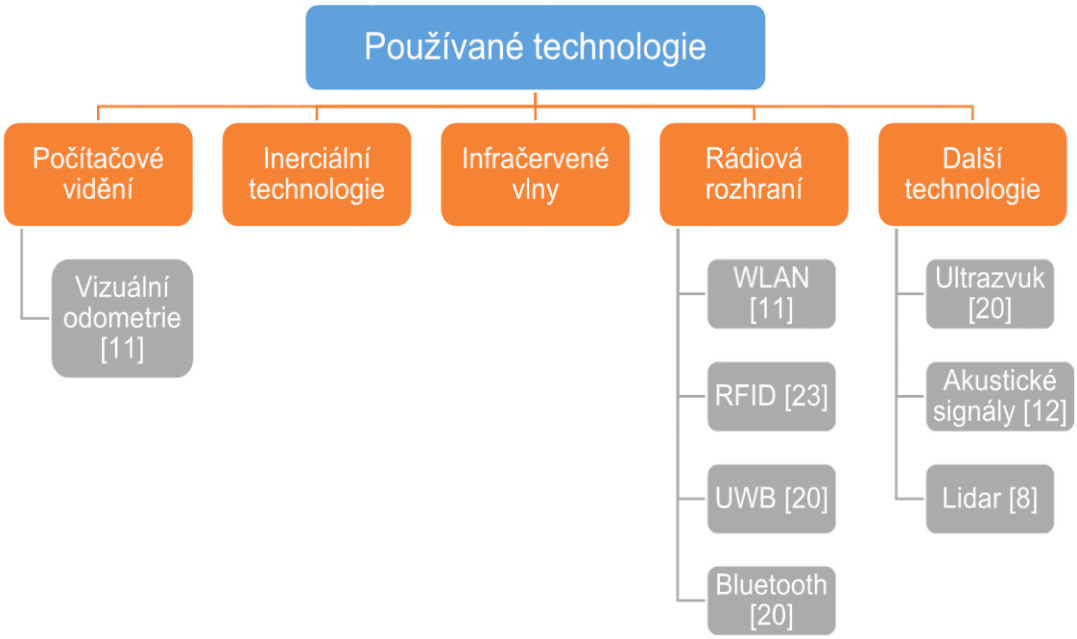
Výzkumy se zabývají autonomní navigací, které vedou k testování a vyhodnocování pomocí RGB-D kamer nebo jiných technologií. [17, 23]

Předchůdcem navigačních technologií uvnitř objektů byly aktivní snímací senzory. Mezi tyto senzory patří například sonar či laserové dálkoměry. Nástupními technologiemi byly inerciální pasivní systémy, které přitahují pozornost díky větším rychlostem procesoru a rozšířenými systémy s více procesory. V současnosti je vnitřní navigační systém prováděn komerčními metodami. Komerční metody jsou využitelné pro ovládání bezpilotního prostředku i pro sledování trajektorie letu. Témata, která se týkají trajektorie letu, jsou častými předměty technologických konferencí. Důležitým aspektem pro vytvoření navigačního systému je pohyb v reálném čase, protože pohyb v hale není regulovaným prostředím. Prostředí uvnitř haly je plné objektů, kterým jsou například výkladní regály, části vzduchotechniky a nábytek. Jeden z výzkumů se zabývá pohybem bezpilotního prostředku v muzejním objektu. Muzeum zvyšuje šanci bezpilotnímu prostředku střetnout se s předmětem, který má atypický tvar. Tedy není možné nadefinovat přesný model objektu. Kombinací atypických předmětů a stísněného prostoru vede k vývoji preciznějších systémů, které umožňují bezpečný pohyb uvnitř budov. V rámci tohoto výzkumu byl vytvořen návrh bezpilotního prostředku včetně polohovacího systému uvnitř budovy. [20, 23, 24]

Světová akademie věd se zabývala implikací koncových uživatelů do celého procesu návrhu metodou designového myšlení. Tato metoda se stále častěji objevuje ve výuce a zabývá se nejen zadáním úkolu a nalezením jeho řešení ale i celým procesem hledání optimální varianty. Touto metodou byla získána informace prostřednictvím fokusních skupin. Fokusní skupiny byly vytvořeny z 13 sektorů zaměřených na kreativní ekonomii. Díky těmto skupinám byly zjištěny potřeby a názory. Dále byl analyzován výstup softwarem pro kvalitativní analýzu. Analyzovaná data byla použita při návrhu AiRT bezpilotního prostředku. Finální fáze výzkumu vedla k debatě, jaké další senzory by měly být součástí bezpilotního prostředku. Koncept byl založen na bezdrátové komunikaci UWB, ta se pro vnitřní využití zdá být jako rozumným řešením pro lokální poziční systémy. [25]

1.3.1. Vyvíjené technologie

Lokální poziční systém se v posledních letech stává hlavní součástí výzkumu, hlavně při umožnění operací, jako je autonomní navigace uvnitř budov nebo při letech v omezeném prostředí. Vyvíjené systémy se skládají z ad-hoc zařízení, které splňují požadavky ke scénářům – daným činnostem. LPS přitahuje velký zájem díky kompromisu přesnosti s dostupností a nákladů. Studie potvrzuje, že přesná definice objektů v prostředí je nutností pro vývoj pohybu bezpilotních prostředků uvnitř budov. Výzkum klade důraz na potlačení omezení přesnosti z družicové navigace a vede k návrhu senzorické sítě, která je nezávislá na GNSS signálu. S tímto východiskem vyplývá na povrch, že je potřeba definovat základní použitelné technologie (Obrázek 7) jako je výše zmíněný UWB. [11, 20, 23]



Obrázek 7: Schéma používaných technologií

Bezpilotní systémy jsou komplexně pojaté systémy, které jsou závislé na hardwarovém, ale zejména na softwarovém řešení. Vylepšení elektroniky umožňuje pokročilejší prvky v navigačních a řídicích systémech. Pokročilejší technologie umožňují větší a stabilnější pokrytí. Aktuálně vyplývá na povrch, že využitelnost infračervených vln, RFID tagů, UWB a ultrazvuku má menší využití s porovnáním s kamerovým systémem. Pro zajištění rychlé a efektivní komunikace je poskytování 5G signálu a IoT. Rádiové technologie vedou k možnosti odhadu polohy za předpokladu nižších provozních nákladů. Nižší provozní náklady vedou k zajímavým řešením, co se týče chybovosti bezpilotních systémů v uzavřených

prostorech. Pro lepší přehlednost jsou technologie rádiového rozhraní chybovosti zpracované do tabulky (Tabulka 1). [16, 26, 24]

Tabulka 1: Souhrn chybovosti technologií rádio frekvence

	Technologie	Chybovost v metrech
RFID	RSSI	1-5
UWB	TDoA/ToA	0,01 - 1
WiFi	RSSI	5 - 15
Bluetooth	RSSI	1 - 5

1.3.2. Řízení a kontrola

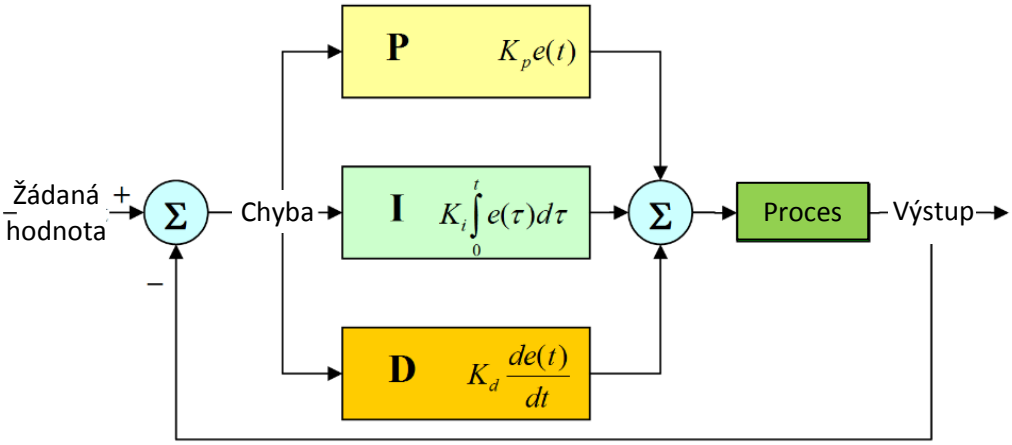
Podle posledních výzkumů za uplynulých 20 let se spousta studií zaměřila na navádění, navigaci a řízení, což má za následek vývoj nových technologií a metod. Někteří z vědců se pokusili přezkoumat podmnožinu GNC. Studie, kterou vedli vědci na univerzitě v Melbourne, se zabývala řízením letů, zatímco další tým, pod vedením Kapoora, zkoumala využitelnost funkcí autopilota. Komplexní australská studie týkající se GNC byla nedávno provedena Faridem Kendoulem. Kendoul poskytl přehled systémů GNC ke zvýšení autonomní schopnosti. GNC je prováděno třemi způsoby – rádiové ovládání, pozemní video stanice a autopilot. Rádiové ovládání, je metoda založena na bázi rádiového systému, který zahrnuje přijímač a vysílač. Vysílání umožňuje přesnost dat pomocí elektromagnetických vln. Dálkové ovládání je složeno z rádiového vysílače, který obsahuje několik rádiových frekvencí a pilot vysílá z pozemní stanice pokyn bezpilotnímu prostředku. U řídicích systémů je dosah vysílače daleko komplexnější a pokrytí je vyšší než u ručního dálkového ovládání. [27, 48]

Pro navigační systémy, ovládané z pozemní video stanice, je bezpilotní prostředek osazen kamerou. Kamera provádí záznam obrazu, ten je dále odesílán do pozemní stanice. Přijímací antény uvnitř pozemní stanice vyhodnocují obraz pomocí analýzy výstupních vln. Ve špatně přístupných oblastech je vhodné využít i zesilovače signálu, které výrazně usnadní příjem datového toku. Díky mnoha výzkumům je možné využít velké množství technologií pro sběr obrazových informací například pomocí ultrazvukových senzorů, barevných, tepelných či infračervených kamer. Důležitou technologií je počítačové vidění, umožňující sběr dat bez potíží. Obraz je zpracování pro navigaci, stabilizaci a další. Pokud je bezpilotní prostředek vychýlen z dosahu

video vysílače, ztrácí se přenos signálů a dochází k systémové nestabilitě. Optimálním řešením pro bezpilotní systémy jsou autopiloti. Autopilot vyhodnocuje z předem definovaných scénářů a je tedy vhodný pro vykonávání misí nejen na krátkých vzdálenostech. Autopilot má předem stanovený letový plán, podle kterého plní úkol. V současné době existuje více druhů autopilotů. Například micropilot má jedinečně funkce a je provozu schopen ve výškách 12 km. Zároveň může ovládat až 24 servomotorů. [28, 29]

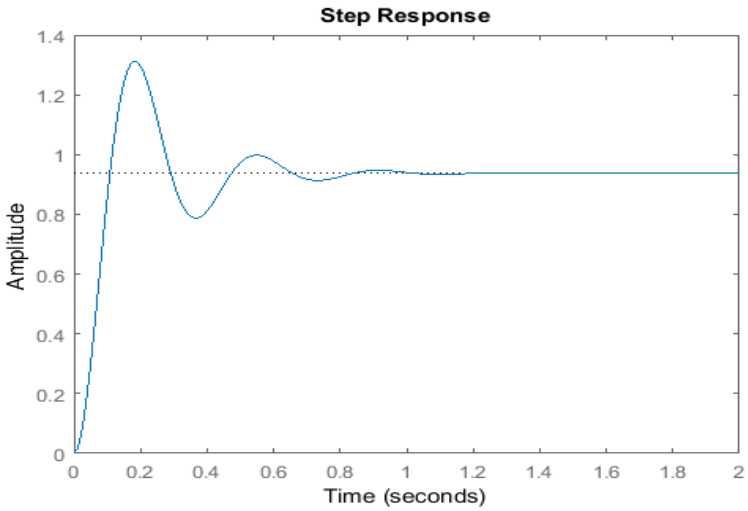
1.3.3. PID kontrolér

PID kontrolér je regulátor, který je používán ve spojitých systémech. Regulátor smyčky je aplikován v automatizaci k řízení výstupu a k přivedení požadované hodnoty. Regulátor je řízen pomocí třech složek (Obrázek 8) a to proporciální, integrační a derivační. Tyto tři složky regulují smyčku procesu. První složka, proporciální, vynásobí konstantou odchylku, která vstupuje jako rozdíl požadované a změřené hodnoty. Integrační složka zpracuje odchylku tak, že ji vynásobí konstantou a přičte ke své složce. Derivační složka vezme rychlost změny odchylky a vynásobí ji svou konstantou. Tyto tři složky se navzájem paralelně kombinují za účelem nalezení jejich optimálních poměrů. Základní podstata celého regulátoru funguje tak, že nepřetržitě sleduje chybovou hodnotu a pomocí ní provádí korekci, která se projeví na výstupní hodnotě. [30, 31, 61]



Obrázek 8: Schématický model PID kontroleru [61]

PID kontrolér při správně vyladěném regulačním obvodu zajišťuje vyšší přesnost regulace oproti regulaci ON/OFF. Příkladem v praxi je například topení. Regulátor ON/OFF vypíná topení až po dosažení žádané hodnoty. U PID kontroléru regulace topení je vypnuta ještě před dosažením žádané hodnoty. Příklad regulace topení je uváděn pouze pro upřesnění funkcionality regulace ON/OFF a PID kontroléru. Ilustrace funkcionality PID kontroléru je ilustrována pomocí grafického zpracování v programu Simulink, který je součástí Matlabu (Obrázek 9). Základním principem je práce se spojitými signály. Homeostáze systému nastává tehdy, kdy PID kontrolér cyklicky upravuje odchylku mezi chybovou hodnotou a požadovanou hodnotou. Homeostáze tvoří přímku kolmou na osu y a rovnoběžnou s požadovanou hodnotou. [32]



Obrázek 9: Ustálený signál aplikace PID kontroleru [62]

Model slouží k návrhu stabilního a přesného kontroléru. Vývoj PID kontroléru je vysvětlen v rámci malajsijské univerzity ve spolupráci s univerzitou v Nottinghamu. Princip PID kontroléru je demonstrován na výše zmíněném schématu. V počátku je definována žádaná hodnota, ke které se PID kontrolér pomocí cyklů přibližuje. PID kontrolér stále pracuje s rozdílem chyby a žádané hodnoty. [31, 32]

Regulace teoretického modelu PID kontroléru vede k jednoduchému popisu docílení stavu homeostáze. U modelu bezpilotního prostředku je získání tohoto stavu složitějším tématem. Tato regulace nevychází pouze z jednoho vektoru jako tomu je například u regulace tempomatu. Bezpilotní prostředek se pohybuje kolem tří os (vybočení, klonění a klopení). Pohyby kolem třech os je třeba sladit tak, aby se dostaly

do homeostáze. Do takového stavu se dostanou pomocí nastavení čtyř vstupních sil, které demonstrují jednotlivé motory. Vstupní síly si lze představit jako jednotlivé tahy na každém rotoru. Pohyb vpřed bezpilotního prostředku je konfigurán tak, že se zvýší rychlost na předních rotorech a současně se sníží rychlost na zadních rotorech. Vývoj klade důraz na rychlejší odezvu PID kontroléru. Pomocí rychlé odezvy je umožněn následně rekurzivní algoritmus pro regulování pohybu. Veškeré fáze výpočtu týkající se chyb sledování jsou zjednodušeny. Dalším kritériem, na kterém závisí funkcionalita výběru kontroléru, je způsob řízení bezpilotního prostředku. Způsob řízení se dělí na režimový a bezrežimový. Pro režimový kontrolér jsou důležité nezávislé řadiče pro každý stav. Poté nadřazená řídící jednotka rozhodne, jak budou jednotlivé řadiče interagovat. U bezrežimových PID kontrolérů ovládá všechny stavy jeden kontrolér. Přijatá strategie kontrolérem je shrnuta do dvou subsystému. První subsystém se týká ovládáním polohy, zatímco ten druhý se zabývá výškovou kontrolou. Studie bere v potaz, že režimový návrh systému může zanedbávat gyroskopické efekty. [31, 33]

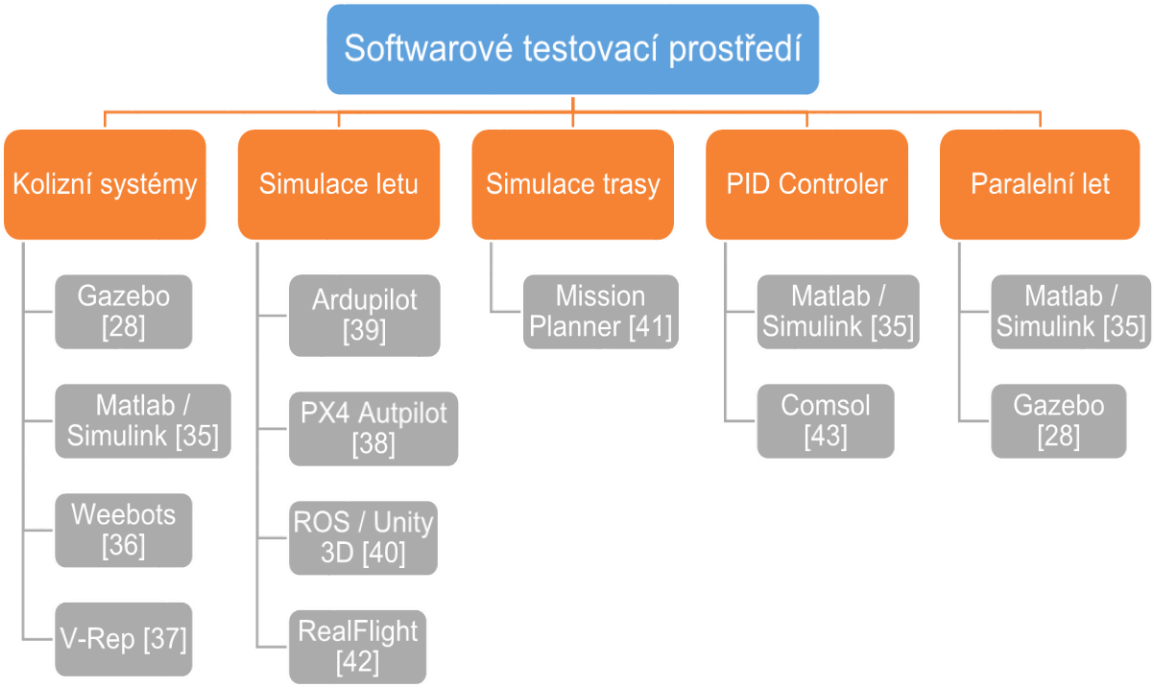
1.4. Testovací prostředí a implementace testů letu

Testování a validace je jeden z prvních kroků celého vývoje. Testování bezpilotních prostředků s rotačním křídlem je klíčové pro zajištění vyšší bezpečnosti. Pro bezpilotní prostředky platí, že mají mnoho výhod; například vertikální vzlet a přistání, schopnost vznášet se nebo schopnost manévrovatelnosti v úzkém prostoru. Užší prostor si vyžaduje rychlejší odezvu ze snímacích senzorů. Rychlejší odezva vede k zajištění vyšší přesnosti, díky které je umožněn optimální bezrizikový pohyb. Pohyb ve stísněných prostorech vyžaduje řídící platformu, která přímo podléhá úkonu letové mise. Každá letová mise s sebou nese množství rizikových situací. Pro zabránění nehod v rizikových situacích je nutné správně nakonfigurovat řízení letu. Pro konfiguraci jsou využívané testovací platformy, které mají základ v gyroskopických kruzích. Simulace se realizuje před letem a slouží k nastavení řídících parametrů. Testy mohou obsahovat nadmořské výšky, sklony, klopení či zatáčení. Zkušební předletové procesy jsou prováděny za kratší dobu. Touto problematikou se zabývala také studie, která vytvořila návrh grafického rozhraní pro odesílání a příjem dat ze senzorů. Komunikační moduly rozesílaly data drátovým či bezdrátovým spojením během předletového testu. Pomocí těchto testů bylo možné získat analýzu hodnot otáček motoru nebo rychlosti. Provedení analýzy a simulace pomocí grafického

rozhraní zjednodušuje předletovou zkoušku. Navrhované rozhraní této studie umožnilo zasílání požadovaných hodnot na řídící desku bezpilotního prostředku. [34]

Softwarové testování klade důraz na preciznější předcházení rizikových situací. U bezpilotních prostředků existují limitní hodnoty pro trajektorii dokonce i tak, pro výkony motorů. Definice limitních hodnot napomáhá k určení hranice na pokraji manévrovatelnosti. Výzkumy směřují k vytvoření testovacího prostředí, které dokáže včas vyhodnotit kritické situace. Trendem je vytvořit uživatelsky přívětivé prostředí, které dokáže dokonale znázornit rizikové oblasti ještě před letem. Softwarové výzkumy vychází z komplexních nástrojů matematické analýzy. Následná validace je prováděna analýzou, která je znázorněna do grafů. Díky softwarové analýze lze identifikovat a definovat výstražný systém, který vede ke zlepšení spolehlivosti bezpilotního prostředku. Spolehlivost je důležitým parametrem pro pohyb v obydlených oblastech ale zejména ve vnitřních prostorech. Studie si dávají za cíl zvednout spolehlivost bezpilotních prostředků, která je aktuálně způsobena vysokou mírou selhání v provozu. Mnoho selhání pohybu bezpilotního prostředku je závislé na pozdní reakci obsluhy, která neidentifikovala chybu dostatečně včas. Systémy jsou navrhovány tak, aby dokázaly predikovat včasné upozornění na chyby v bezpilotním prostředku. Manévrovat může operátor, avšak automatizace se zdá vhodnějším řešením v krizových situacích. [34, 35]

V rámci softwarového testování lze testovat a odladit patřičné chyby. Generování proměnných chyb je ve vzdušném prostoru mnoho. Chyby se týkají vnějších i vnitřních vlivů bezpilotního prostředku v daném prostředí. Testovací prostředí simuluje, jak bude bezpilotní prostředek reagovat při různých zátěžích. Kategorie testování sdružuje testování kolizních systémů, simulaci letu, simulaci trasy, PID kontroléru a paralelního letu. Každé testované odvětví má své vývojové prostředí (Obrázek 10). Tato bakalářská práce se zaměřuje na kolizní systémy zprostředkované skrz Gazebo. [17]



Obrázek 10: Rozdělení softwarového testovacího prostředí

2. Použité prostředky a metody

U návrhu testovacího softwarového prostředí bezpilotního prostředku je důležité zvolit správné postupy testování včetně výběru správného testovacího prostředí. Jak je již vidět z výše uvedeného schématu, testovat lze téměř veškeré parametry pro let. Cílem testovacích procedur je nalezení a zajištění optimálních výstupních hodnot. Pro softwarovou validaci indoor bezpilotních prostředků existuje celá řada technologií, které lze použít. Pro účely téhle práce bude vytvořené testovací prostředí v systému ROS propojené s grafickým rozhraním Gazebo. V této části je nastíněný popis jednotlivých nástrojů, se kterými se můžeme setkat. Jsou zde vysvětleny softwarové nástroje jako je například ROS, Gazebo či Linux. Tato kapitola rovněž obsahuje bližší specifikace podpůrných programů a knihoven, které zajišťují celkovou funkcionalitu.

2.1. LINUX

Pro účely vývojového prostředí v rámci pokročilé správy počítače se využívá open source operační systém Linux. Jádro Linuxu umožňuje spouštění více programů najednou a každý program se může skládat z více procesů. Chod více procesů se v současné době vyznačuje spíše pod pojmem multitasking. Multitasking je již několik let běžnou funkcí desktopů. Velkou výhodou systémů Linux jsou svobodně

šiřitelné licence v podobě distribucí, které je možné nainstalovat nebo používat bez instalace. Strategie Linuxu je v dostupnosti distribuovaných licencí, které lze velmi volně využít, distribuovat i upravovat. Mnohé distribuce poskytují vizuálně orientované nástroje, které usnadňují konfiguraci. Tyto nástroje mohou být velice užitečné zvlášť pro začátečníky s Linuxem. Velkou nevýhodou těchto nástrojů můžou být skryté podrobné informace o tom, co se děje, když jsou prováděny změny. Pro každou distribuci jsou vizuální nástroje proprietární, a tudíž se mohou lišit u různých distribucí. Vizuální nástroje v případě poruchy úplně nepomohou při řešení problému. V tuto chvíli je zajištěno, že problematické oblasti je možné řešit s uživateli, kteří již v minulosti tento problém řešili. Z toho vyplývá obrovská výhoda Linuxové implementace, která je zakomponována právě v otevřených licencích. [44, 45]

2.2. ROS

Ros Operating System, dále jen ROS, je open source robotický systém. Jeden z hlavních cílů ROSu je zrychlování vývoje robotických aplikací pomocí vytváření frameworků. Systém poskytuje standardy pro vývoj robotického softwaru, který lze implementovat na jakémkoli robotickém zařízení. Velkou výhodou ROSu je používání již vytvořených frameworků. Používání frameworků vede k efektivnímu přístupu pro vývoj robotů. ROS je middleware, který funguje na již existujícím operačním systému. Middleware zodpovídá za zpracování komunikace mezi jednotlivými programy v distribuovaném systému. Software ROSu se dále dělí na dvě části – jádro a sadu knihoven. Jádro je právě middleware, který zajišťuje komunikace. Oproti tomu sada knihoven je zásobník frameworků, které fungují na bázi plug & play. Tyto knihovny obsahují například PID control loop, řadiče pro servomotory nebo plánování pohybu. Vývoj ROSu je zejména ve dvou jazycích – C++ a Python. Obě jazykové varianty jsou velmi využívané ve vývoji robotických aplikací. Programování v ROSu umožňuje vytvářet programy pro základové desky jako je například Arduino nebo Tiva-C LaunchPad. [40, 46]

Základním principem ROSu je činnost, která umožňuje velké množství spuštěných procesů. Tyto procesy mezi sebou komunikují a předávají si data. Toto koncepční řešení se skládá z Nodes, Master, Messages, Services, Topics a Bags (Obrázek 11).

Uzel (node)

Uzly jsou interpretovány jako jednotlivé procesy a jsou nejmenší částí celku. Lze je využít jako účelové programy s funkcí. Každý uzel se stará o jednu činnost. Můžeme tedy říci, že jeden uzel se stará například o zpracování dat ze senzorů, další uzel řídí otáčky servomotorů či tvoří monitoring pro stav baterie. Uzel představuje zdrojový kód, který je psaný v Pythonu a C++. Uzly jsou sloučeny do balíčků a společně tvoří celek. Tento celek může být publikován uživatelům, kteří ho mohou dále využít.

Master

Master je prostředníkem mezi jednotlivými uzly. Lze říci, že zprostředkovává komunikaci mezi uzly, které jsou jmenovitě registrovány a vzájemně se dokázaly dohledat ve struktuře. Další úlohou masteru je přenos informací o případných změnách struktury. Tato změna může prezentovat například přidání nového uzlu, tedy zajišťuje dynamiku systému. V rámci systému ROS je masterem roscore. Roscore zprospředkovává komunikaci mezi jednotlivými uzly. Kromě komunikace umožňuje uzlům aby se vzájemně lokalizovaly.

Zpráva (message)

Zprostředkování komunikace funguje tak, že si jednotlivé uzly zasílají zprávy, které představují jednoduchou datovou strukturu. Jsou složeny z řetězců znaků, reálných čísel, čísel s plovoucí řadovou čárkou, algebraických zápisů i logických funkcí. Příkladem může být zpráva, která představuje hodnotu stavu akumulátoru v procentech.

Téma (topic)

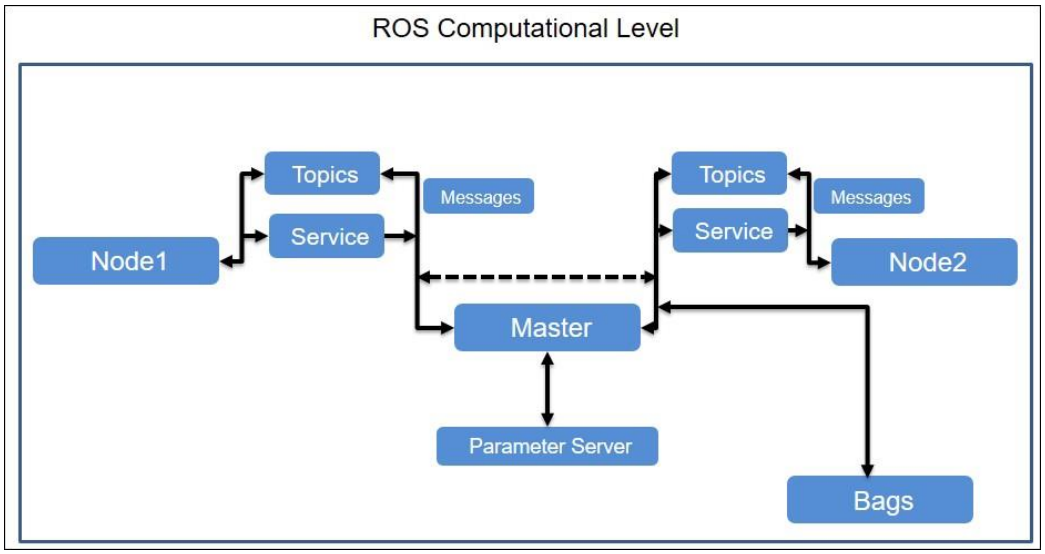
Témata identifikují zasílané zprávy podle názvu či struktury. Na základě určeného tématu dokáže uzel odeslat či naopak přijímat pouze data, o která má zájem. Uzly se dále dělí na takzvané odesílatele a přijímače. Neexistuje žádný limit pro jedno téma. Pro každé téma může více odesílatelů i přijímačů.

Služba (service)

Služba může být zaměněna s tématem, jelikož má stejnou funkci pro identifikaci příchozích i odchozích zpráv. Hlavní rozdíl spočívá v zaslání zprávy pouze na vyžádání určitým uzlem. Poté jiný uzel odpovídá na poskytnutou žádanou zprávu.

Taška (bag)

Nástroj systému ROS, který umožňuje ukládání a zpětné nahrávání zaslaných dat prostřednictvím zpráv. Přináší tak možnost nahrání dat ze senzorů určených pro vzdálenost, do souborů, kde má uživatel přístup ke kompletním nasnímaným hodnotám. S těmito hodnotami lze dále pracovat. [46]

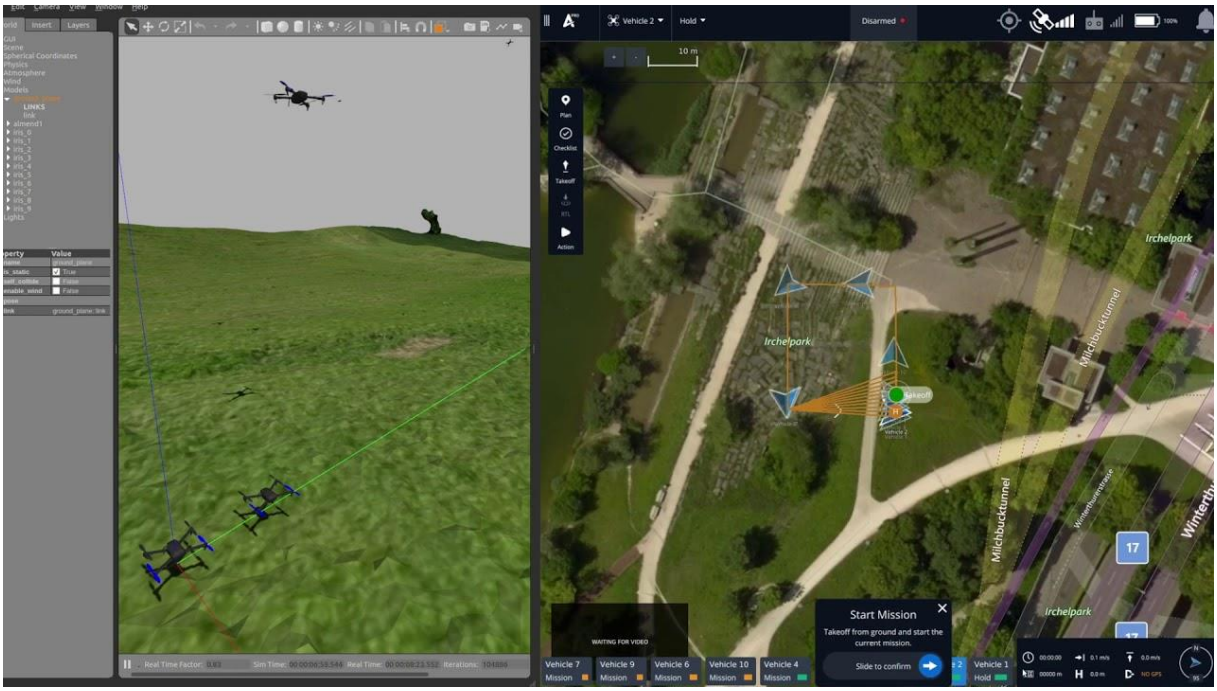


Obrázek 11: Komunikační úrovně ROSu [63]

2.3. Gazebo

Gazebo je open source 3D simulátor, který je schopen být spuštěn na jakémkoliv počítačovém systému. Komplexní komunikace je nejlépe poskytována operačním systémem Linux. 3D simulátor integruje ODE fyzický engine, OpenGL renderování. Kromě integrace podpůrných nástrojů nabízí nástroje pro podporu senzorů při simulaci. Robotický simulátor umožňuje rychlé testování algoritmů, vytváření nových designů robotických modelů, provádění degresivní testování, simulování populace robotů ve složitých prostředí a v neposlední řadě řeší problémy s Gazebo „živou“ komunitou. Gazebo je vytvořené jako nezávislý projekt, je velmi využívaným nástrojem v oblasti vývojového prostředí pro robotické systémy. Program využívá vysoce účinných enginů, jako je například výše zmíněný ODE, Dart, Bullet a Simbody. Tyto enginy zajišťují reálnější fyzikální zákony. Grafické rozhraní Gazebo poskytuje

velmi jednoduché ovládání. Práce v Gazebo je realizována pomocí nástrojové lišty. Tato lišta zajišťuje jednoduché operace s objekty. Gazebo slouží jako nástroj pro vizuální výstup z ROSu (Obrázek 12). [47]



Obrázek 12: Gazebo s implementací bezpilotního prostředku Iris [64]

2.4. MAVLink

Micro Air Vehicle Link, zkráceně MAVLink je komunikační protokol pro malé bezpilotní prostředky. Komunikační protokol vznikl pod rukama Lorenze Meiera v roce 2009. MAVLink definuje způsob, jak má být zpráva strukturována a styl serializace v aplikační vrstvě. Zprávy jsou poté předány do nižších vrstev protokolu TCP/IP. Velkou výhodou MAVLinku je podpora různých typů medií po transportní vrstvě. Tato výhoda je zajištěna díky lehké struktuře. MAVLink zajišťuje přenos skrz WiFi, Ethernet či sériové telemetrické kanály s nižší šířkou pásma. Frekvence nižší než 1GHz umožňuje dosáhnout rozsahů pro dálkovou komunikaci bezpilotních systémů. Citlivost signálu závisí na prostředí, úrovni hluku a nastavení antény. Nejjednodušším zprostředkováním je použití sítí WiFi nebo Ethernetu, kde probíhá streamové vysílání MAVLink zpráv prostřednictvím IP adresy. V takovém případě autopilot podporuje UDP i TCP protokol. Kombinace těchto dvou protokolů zajišťuje spolehlivější spojení mezi bezpilotním prostředkem a pozemní stanicí. Protokol je využívaný pro orientaci prostředku, protože vychází ze zaznamenaných dat jako je například GPS poloha a rychlost. [49]

2.5. Pixhawk 4 a kódování v Pythonu

Pixhawk 4 autopilot je open-source autopilotní systém, který je orientován na autonomní bezpilotní prostředky. Autopilotní systém PX4 je součástí projektu Dronecode. Z toho plyne bezproblémová integrace s dalšími nástroji z tohoto projektu, jako je například komunikační protokol MAVLink nebo řídící stanice QGroundControl. Autopilot nabízí širokou flexibilní sadu nástrojů pro vývojáře bezpilotních prostředků. Poskytování této sady dopomáhá ke sdílení technologií a vytváří tak celosvětovou podporu ve vývoji bezpilotních prostředků. Cílem této podpory je zajištění softwarového zásobníku PX4 založeného na Linuxu. Modulární systém PX4 spojuje hardwarové komponenty se softwarovým rozhraním. Testování autopilota je specifické ve výběru senzorů. Po výběru senzorů je autopilot schopen detekce a následně ovládat jednotlivé motory. Řídící smyčka sbírá data ze senzorů s deterministickými intervaly. Z těchto hodnot pochází aktualizované nastavení pohybu. [49]

V kořenovém adresáři PX4 lze nalézt konfiguraci jednotlivých základních desek. V těchto konfiguracích lze upravovat parametry senzorů, které jsou nastaveny jako výchozí. Kromě toho lze importovat senzory, které nejsou přímo svázané se základní deskou. Import senzorů probíhá z repozitáře GitHub. Mimo konfigurovatelných ovladačů PX4 je možné naimportovat rozšiřující framework. V našem případě to je kolizní vyhýbání – avoidance. Tento framework poskytuje rozšíření, díky kterému je možné oblétávat detekované překážky. Pro oblétávání překážek jsou zde složky s integračními testy, které umožňují vyzkoušení jednotlivých misí. V případě potřeby je možné tyto soubory modifikovat a programovat pomocí kódu. Kódové programování probíhá v C++, ale zejména v Pythonu. [49, 50]

Python je vysokoúrovňový skriptovací programovací jazyk, který nabízí dynamickou kontrolu datových typů a zprostředkovává podporu například objektově orientovaného či imperativního programování. Rostoucí trend užívání Pythonu vzrůstá zejména díky open-source, který nabízí zdarma instalační balíky pro veškeré operační systémy. Většina distribucí Linux nabízí instalaci Pythonu jako základní součást instalace. Hlavní vlastností tohoto jazyku je jednoduchost z hlediska učení. Je považován za jeden z nejvhodnějších programovacích jazyků pro začátečníky. Díky modulárním instalačním balíčkům je vysoká produktivnost Pythonu z hlediska rychlosti psaní nových programů. Kód působí velmi jednoduchou stručností zápisu a je tak vhodný

k vytváření programů z hlediska rychlosti psaní. Stručnost kódu se týká těch nejjednodušších programů, ale také aplikací velmi rozsáhlých projektů. [49, 50]

3. Výsledky a zpracování

Jednotlivé kroky jsou postaveny tak, aby nejprve byla získána orientace vývojového prostředí. Kromě orientace ve vývojovém prostředí je osvojena znalost senzorů. Spojením těchto dovedností je využito k získání hodnot PID kontroléru, které byly následně namodelovány v Simulinku. Výsledné chování bezpilotního prostředku je demonstrováno přímo v grafickém prostředí Gazebo. Jednotlivé kroky celého postupu jsou popsány níže.

3.1. Instalace systému

Na začátku této kapitoly je zcela zásadní vybrat si správnou verzi operačního systému Ubuntu. Rozhodnutí závisí i na volbě live nebo desktop distribuce. Optimálním řešením pro docílení sledovaných výsledků práce byla vybrána verze Ubuntu 18.04, která je instalována jako desktopová. Tato verze je zcela klíčová pro zajištění stability a kompatibilty systému. Verze Ubuntu 18.04 umožňuje bezproblémovou instalaci systému ROS Melodic. Součástí instalace jsou zahrnuty i doplňující a rozšiřující balíčky knihoven, které otvírají nové možnosti pro celkovou funkcionalitu. Postup je demonstrován v klíčových krocích. [51]

Orientace v rozhraní Linux je provedena zejména skrz příkazový řádek. Prvním příkazem pro instalaci ROS systému je příkaz *wget* pro získání souborů pomocí přenosových protokolů. ROS Melodic je importován z webového prostředí na lokální počítač. Import cílového souboru je umístěn do složky, odkud je zavolán příkaz. Výchozím nastavením tohoto příkazu je kořenová složka počítače. Dále je importovaný soubor spuštěn příkazem *bash*.

```
bash ubuntu_sim_ros_Melodic.sh (1)
```

Příkazem je provedena instalace ROS Melodic. Zvolený typ instalace obsahuje kromě ROS Melodic také instalaci MAVROS, Gazebo9 a je vybudováno pracovní prostředí catkin_ws. Následně je provedena aktualizace, díky které nedochází k potížím při dalších krocích. Jako další operací je zvolen příkaz pro import složky včetně následné vytvoření kopie. Při tomto postupu již není nutné spouštět instalaci pomocí

```
git clone https://github.com/PX4/Firmware.git (2)
```

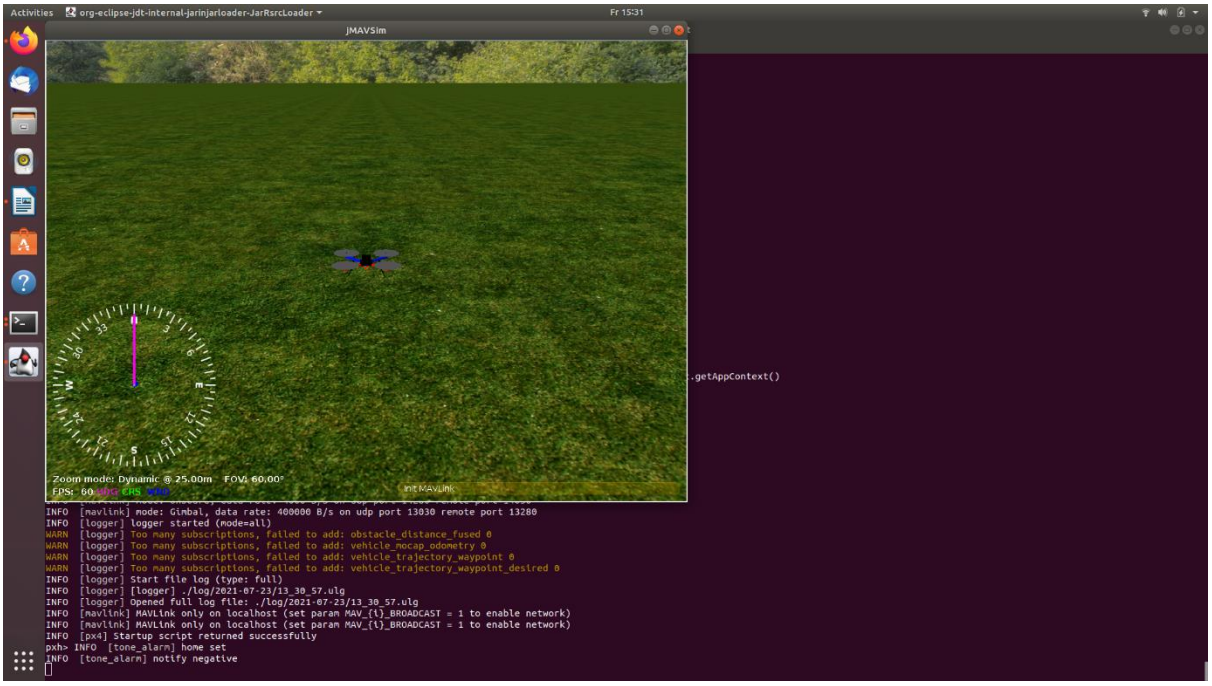
příkazu *bash*. Importovaná kopie složky je vytvořena v místě zavolání příkazu. Pro práci a volání programů uvnitř složky je zapotřebí se přesunout do této složky.

Po přesunu do složky je možné spustit příkazy, které otevřou grafické rozhraní jmavsim. Pokud se tomu tak nestane, je důležité doinstalovat moduly Pythonu 3 včetně

```
make px4_sitl jmavsim                    (3)
```

instalačního balíčku *python3-pip.* Díky tomuto balíčku je zprostředkována možnost instalace, která stáhne jednotlivé moduly a následně je nakonfiguruje. Tyto moduly řeší z velké části problém se spuštěním příkazu *make*.

Pokud pomocí těchto modulů Pythonu nejsou vyřešeny problémy se spuštěním, je zde ještě jedno řešení, součástí kterého je zahrnuta instalace podpůrných knihoven pro implementaci Java SE. Při správně provedených krocích, které jsou výše zmíněny, by příkaz *build* měl být spuštěn a úspěšně proveden. Výsledkem je otevření grafického prostředí pro testování bezpilotních prostředků (Obrázek 13). Toto prostředí však představuje velice základní možnosti. Součástí tohoto rozhraní nejsou cílové nástroje pro detekci překážek. [50, 51]

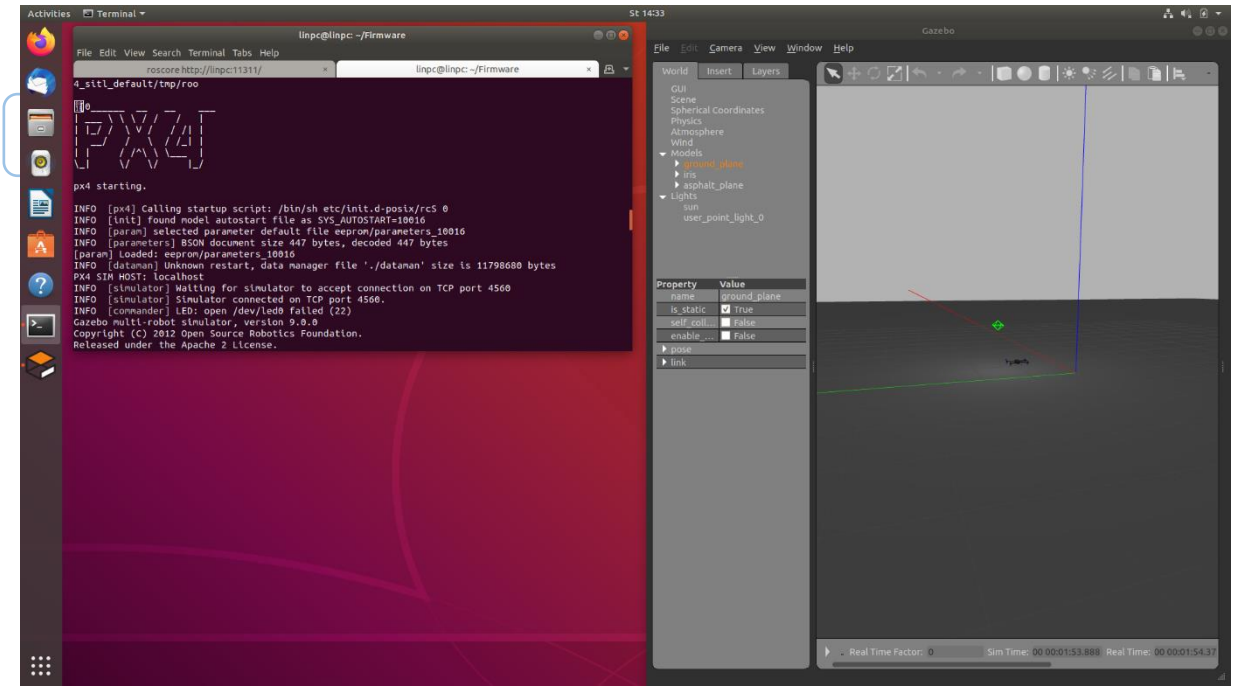


Obrázek 13: Grafické zpracování jmavsim včetně absence kolizního detektoru

Veškeré příkazy jsou prováděny skrz příkazový řádek. Avšak tato instalace nezahrnuje datalinkové připojení. Tudíž například příkaz pro vzlet či přistání v aktuálním nastavení nelze vykonat. Pro plnohodnotnou komunikaci je nutné spustit příkaz *roscore* v novém

terminálu. Roscore zajišťuje komunikaci mezi uzly a programy a je tak nezbytně nutný pro jakoukoliv další aplikaci příkazů, které jsou spojené s balíčky ROSu. To však není zcela vše pro funkční datalinkové spojení. Pro spojení musí být ještě splněna podmínka způsobu komunikace. Výzkum se střetává zejména se dvěma možnostmi pro datalinkové spojení. První způsob je využití příkazu, který vede k přímému spojení skrz protokol UDP. Přímé spojení nedává záruku za doručení informace a tím pádem je nespolehlivé. Bezpečnějším řešením pro zajištění datalinkového spojení skrz přenosový protokol je zprostředkování komunikace pomocí řídící stanice QGroundControl. Před instalací je nutné nainstalovat pluginy gsstreameru a poté zapnout změny pro uživatelská práva. Po zapnutí změn uživatelských práv je nainstalován QGroundControl, který je dále připojen pomocí příkazu *chmod +x*. Po provedení příkazu je otevřena aplikace QGroundControl. Aplikace, která využívá k datalinkovému přenosu mapu. Díky této metodě lze v terminálu, kde je spuštěný jmavsim, zadávat příkazy pro pohyb bezpilotního prostředku, které jsou poté vykonány. [52]

Po úspěšném rozpohybování bezpilotního prostředku je na čase se přesunout a vyzkoušet i prostředí Gazebo. Pro spuštění uživatelského prostředí Gazebo byl zvolen příkaz *make*, který je podobným jako u jmavsim. Příkaz musí být spuštěn v kořenové složce PX4, tedy ve složce Firmware.



Obrázek 14: Výchozí nastavení Gazebo s šedým podkladem a modelem Iris

Při prvním spuštění nejprve je vykonána konfigurace kořenových souborů, poté následuje načtení parametrů s připojením k simulátoru.

Výchozí nastavení Gazebo nemusí být úplně přehledné a model Irisu je zobrazen na tmavé pláni (Obrázek 14). Pro barevné úpravy prostředí je nutné si nakonfigurovat jednotlivé modely v záložce World. Záložka Insert umožňuje vkládání modelů přímo do prostředí. Ve výchozím nastavení je však pouze pár modelů, které nemusí být dostačující pro testování. Proto karta Insert může být doplněna o externí modely, které jsou součástí webového repositáře GitHub. Pro import je proveden příkaz, pomocí kterého jsou nakopírovány modely do kořenové složky. Dalším krokem je nadefinování cesty k modelům. Příkaz pro definici cesty modelům je spuštěn pouze jednou a poté se modely v Gazebo zobrazují stále. Vložené objekty však v aktuálním

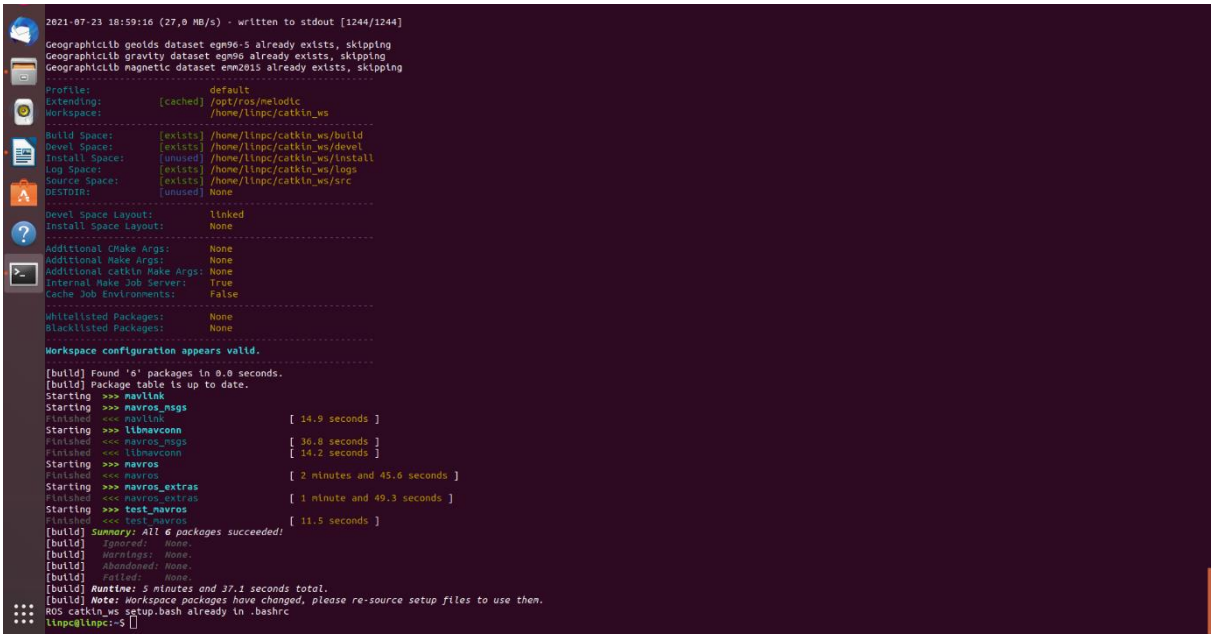
```
cd ~/catkin_ws/src                                    (5)

git clone https://:github.com/PX4/avoidance.git       (6)
```

stavu nejsou detekovány bezpilotním prostředkem a dochází tak ke střetům. Pro detekci překážek je potřeba doinstalovat knihovny pro vyhýbání a detekci. Knihovny jsou k dispozici skrz webový repositář GitHub. Nejprve pro správnou funkcionalitu těchto knihoven je důležité nainstalovat rozšiřující knihovny pro MAVROS a moduly závislosti pro uhýbání. Po instalaci těchto náležitostí následuje přesun do pracovní složky pomocí příkazu *cd*. Po přesunu do této složky provedeme import kopie pro vyhýbání. [51, 56, 57]

Po úspěšném importu je důležitým krokem provést *build* pro vyhýbání. Příkaz *build* nejprve vyhledá makefile, který naskenuje tak, aby zajistil přístup k závislostem. Pokud závislosti nejsou zadány, jsou vyhledány a následně vytvořeny. Tudíž pro kompilaci nových balíčků, které mají být zavolány, je důležité vytvořit build.

```
catkin build -w ~/catkin_ws                           (7)
```

Po úspěšné kompilaci je vygenerován souhrn celého průběhu (Obrázek 15). V případě některých špatně proběhlých kompilací je tato událost vygenerována do textového souboru odkud, lze dohledat, v čem daný problém spočívá.



Obrázek 15: Úspěšné provedení buildu v terminálu

V tuto chvíli je možné spustit Local Planner, díky kterému je umožněna detekce překážek. Vhodným dalším krokem je opětovná aktualizace Linuxu. Pro spuštění simulace Local Planneru je třeba spustit sadu příkazů z kořenové složky PX4. Při spouštění letových plánů vznikají problémy skrz chybějící knihovny pymavlink a analytické nástroje PX4. V tomto případě je vhodné, ještě než bude vývojové prostředí spuštěno, nainstalovat chybějící balíčky včetně Anaconda3, což je platforma Pythonu, která slouží pro vědecké účely. Tato platforma zjednodušuje správu a nasazení balíčků. Instalační soubor Anaconda3 je ve formátu .sh, tedy po stažení do Linuxu je nutné tento soubor zavolat pomocí příkazu bash. Průběh instalace je vykonán přes TUI terminálu. Po dokončení instalace je nutné aktivovat Anaconda3. Úspěšná aktivace programu se projeví tak, že v terminálu přibyde před kořenovou větví text *(base)*. Díky aktivní Anaconda 3 je umožněna instalace px4tools, která musí vycházet z kořenové složky Anaconda 3. [53, 57]

Poslední a nedílnou součástí instalace je vývojařské prostředí pro psaní C++ kódu. Prostředí zprostředkovává lepší orientaci při tvorbě kódu. Při instalaci Anaconda 3 je zmíněné vývojářské prostředí PyCharm ve verzi community. Tato verze je zcela zdarma a volně dostupná. PyCharm v Linuxovém provedení je stáhnutelný jako s příponou .gz. Jedná se o archív, který musí být extrahován pomocí příkazu *tar*.

```
tar xzf pycharm-*.tar.gz -C /opt/
```
(8)

Po extrahování následuje přesun do složky. Zde je spuštěn příkaz sh. Po instalaci vývojářského prostředí je celá instalace kompletní a lze ji považovat za uzavřenou.

3.2. Výběr senzoru

PX4 Autopilot nabízí celou řadu senzorů, které lze v této práci využít. Dle aktuálního zadání byl zvolen senzor Lanbao PSK – CM8JL65 – CC5 (Obrázek 16), který funguje na bázi infračerveného záření. Dosah vzdálenosti je od 0,17 metrů až do 8 metrů. Parametricky se jedná o velmi malé zařízení, jehož váha je 10 g. [54]

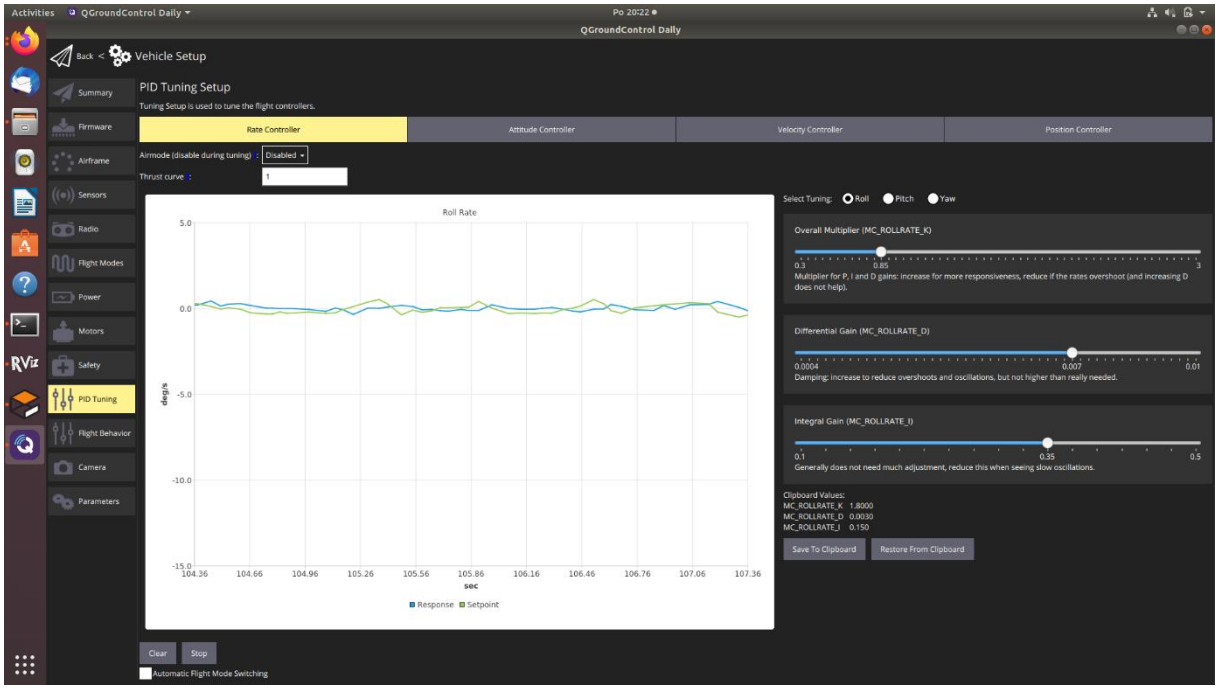
Obrázek 16: Infračervený sensor Lanbao PSK - CM8JL65 - CC5 [54]

Pro konfigurovatelnost tohoto zařízení je třeba přidání ovladače do současné desky PX4. Chybějící ovladač je přidán do souboru default.cmake. Pokud se v tomto souboru nachází ovladač, který má být použit, ale je před ním #, je potřeba odstranit # pro aktivaci. Pokud daný ovladač není součástí definovaných senzorů, je možné provést importování senzoru z repositáře GitHub. Import senzoru je dále přesunut do složky se základní deskou a dopsán do default.cmake. Pro aktivaci nově přidaných senzorů musí být deska aktualizovaná novým buildem pomocí příkazu *make*. Po vytvoření buildu je sensor připraven ke konfiguraci skrz prostředí QGroundControl. U senzoru lze upravit hodnoty pro minimální a maximální dosah senzoru. Pomocí minimálního dosahu je definována limitní hranice. Tato limitní hranice rozhoduje o dalším kroku a pro následující aplikace je tato hodnota nastavena na 1,5 m. U maximálního dosahu je zaznamenáváno, kdy překážka vstupuje do oblasti detekce. U anti-kolizního systému v oblasti užších míst by měla být tato hodnota nastavena maximálně na třech metrech. Takto nízký maximální dosah vymezuje funkčnost senzoru a nedochází tak k neustálým výpočtům vzdálenosti.

Postup po vstupu překážky do oblasti detekce je snímán tak, že hodnota při přibližování se postupně zmenšuje. Součástí výchozího nastavení QGroundControlu je kolizní detekce, která je doprovázena výstražnou hláškou „Collision detection“. Při výstražné hlášce, QGroundControl v režimu „Position“ vyhodnotil překážku jako nezdolatelnou. V takovém případě se zablokuje následný pohyb vpřed. Mód „Position“ zajišťuje pouze manuální ovládání bezpilotního prostředku, proto výstražné hlášení funguje pouze v manuálním režimu. [51, 52]

3.3. Úprava hodnot PID kontroléru – ROS

Pro editaci parametrů je nutné spustit 3D prostředí rviz, ve kterém je umožněn pohyb bezpilotního prostředku a zároveň musí být zajištěn datalinkový přenos. Konfigurace PID kontroléru v ROSu je editována skrz grafické prostředí QGroundControlu (Obrázek 17). Před konfigurací musí být zajištěno, že se bezpilotní prostředek nachází v letu. Zajištěním této podmínky vzniká dostupnost záložky PID tunning, která se součástí konfigurace prostředku. [52]



Obrázek 17: Zobrazení signálu Roll PID kontroleru v QGroundControlu

V této záložce je definováno chování bezpilotního prostředku pomocí táhel. Je zde možné konfigurovat chování rychlosti, výšky, pozice a celkové míry zásahu PID kontroléru. Součástí naší práce je modelování PID kontroléru se všemi třemi složkami. Pro účely téhle práce je zvolenou konfigurací PID kontroléru právě celková míra

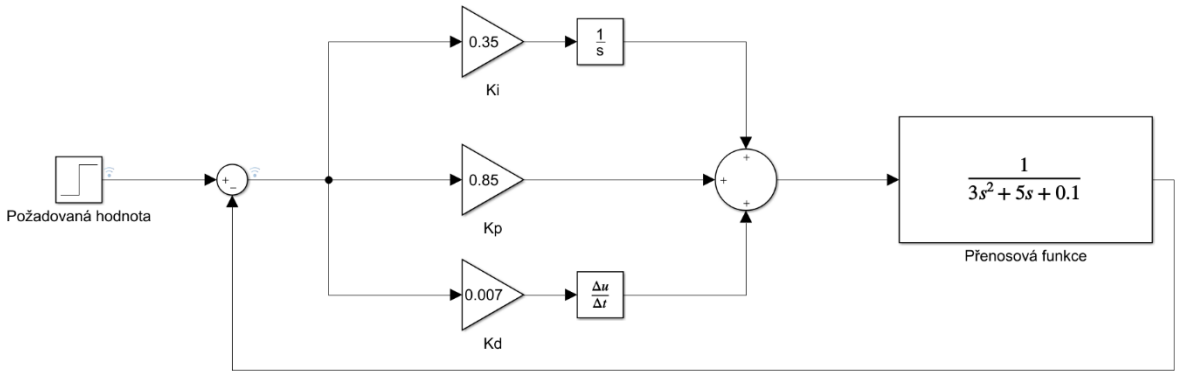
zásahu PID kontroléru. V této konfiguraci lze přizpůsobovat všechna tři táhla, která znázorňují jednotlivé gainy pro proporciální, integrační a derivační složku.

Hodnoty byly nastaveny tak, aby se co nejvíce přiblížily k požadovaným hodnotám vstupu, aby systém působil stabilně. Hodnoty pro následné použití jsou:

- P regulátor – proporcionální 0,85
- I regulátor – integrační 0,007
- D regulátor – derivační 0,35

3.4. Validace hodnot PID kontroléru – Simulink

Pro simulaci a validaci dynamického systému PID kontroléru je zvolen program Simulink. Simulink je součástí interaktivního programového prostředí Matlab. Důvodem pro zpracování PID kontroléru v Simulinku je zejména kvůli intuitivnímu prostředí, práci s maticemi a následnému zpracování do 2D grafu. V případě importu je možné použít import kódu do C++, který lze provést, pokud je zvolena šablona Feedback Controller. Šablona obsahuje objekty již namodelovaného PID kontroléru. V našem případě jsou jednotlivé bloky namodelovány ručně pro lepší demonstraci jednotlivých hodnot (Obrázek 18). Jednotlivé části PID kontroléru jsou popsány níže.



Obrázek 18: Model PID kontroleru v Simulinku

Referenční hodnota

Prvním objektem je referenční požadovaná hodnota. K této hodnotě PID kontrolér upravuje signál tak, aby výsledek odchylky proměnné a požadované hodnoty byl nulový. Pro regulaci je požadovaná hodnota nastavena na 1,5. Tato hodnota vypovídá o ustálení bezpilotního prostředku 1,5 m před překážkou.

PID kontrolér

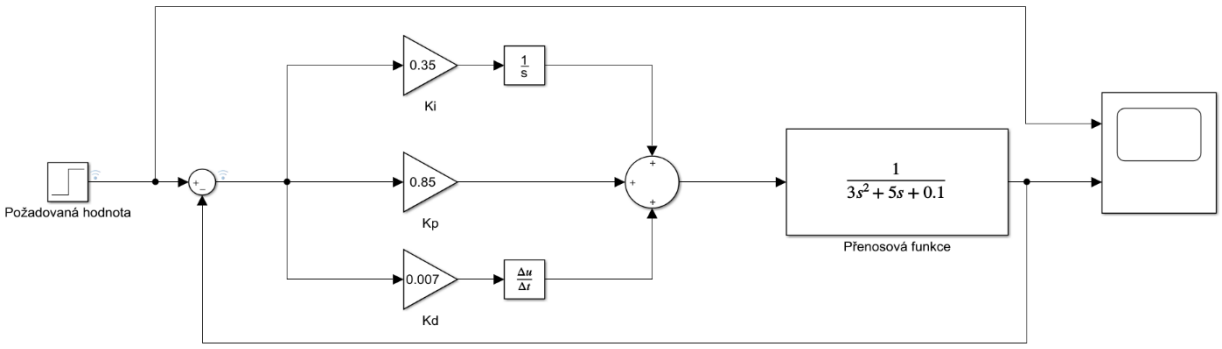
Regulátor je řazen před samotný proces. Do regulátoru vstupuje regulační odchylka a akční veličina (požadovaná hodnota). Jednotlivé složky PID kontroléru byly definovány výše z programu QGroundControl, který funguje ve spolupráci s prostředím Gazebo.

Proces – přenosová funkce

Proces je matematické zobrazení daného systému. Rovnice v tomto procesu vychází z automatických robotických vysavačů, které využívají zpětnovazebný proces k ustálení systému před překážkou. Z principu anti-kolizního systému robotického vysavače vystupuje na povrch, že princip anti-kolizního systému pro bezpilotní prostředky používá k ustálení stejný proces. Převzata je tedy přenosová funkce:

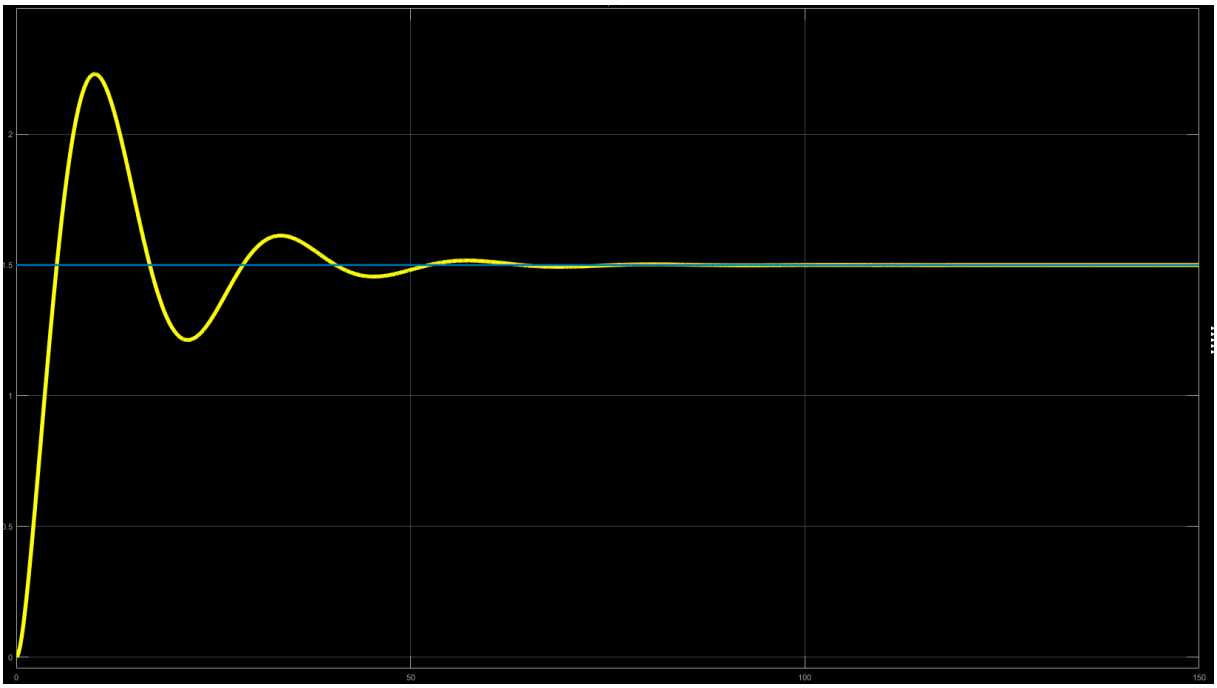
$$\frac{1}{3s^2 + 5s + 0.1}$$

Po rozklíčování jednotlivých bloků je spuštěn model. Pro znázornění výstupních hodnot je k aktuálnímu schématu připojen objekt Scope – náhled (Obrázek 19).



Obrázek 19: Simulinkový model s připojeným náhledem

Díky připojenému bloku Scope, je možné otevřít náhled. Níže zobrazený náhled demonstruje průběh funkce až do ustálení (Obrázek 20). Pro ustálení bylo nastaveno zastavení simulace po 150. Časová hodnota pro zastavení není však uváděna v sekundách, závisí na komplexnosti celého modelu a rychlosti počítače. Funkce vychází z nuly, což není v reálném provedení možné. Po provedení analýzy, při zapojení bloku Scope na místo před vstupem do následujícího bloku, byla odvozena příčina. Za příčinou stojí blok procesu, který je definován pomocí přenosové rovnice, díky které je počátek signálu umístěn do bodu [0, 0]. [55]



Obrázek 20: Simulace průběhu signálu do fáze homeostáze

3.5. Umístění bezpilotního prostředku do programu Gazebo

Po validaci v Simulinkovém prostředí následuje simulace v systému Gazebo. Spuštění vývojového prostředí Gazebo musí proběhnout v kořenové složce PX4. V našem případě z místa */Firmware*. Před spuštěním nesmí být zapomenuto na spuštění příkazů source a export. [49]

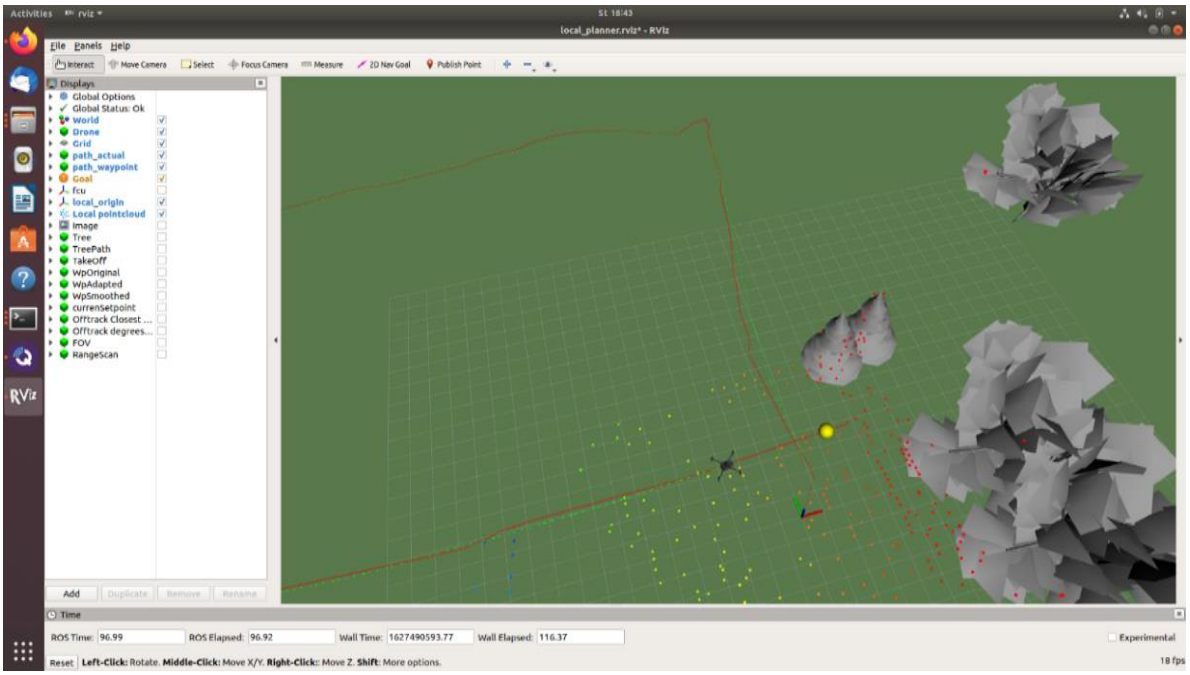
```
source ~/catkin_ws/devel/setup.bash                                    (9)

source Tools/setup_gazebo.bash $(pwd) $(pwd)/build/px4_sitl_default    (10)

export ROS_PACKAGE_PATH=$ROS_PACKAGE_PATH:$(pwd)                       (11)

export ROS_PACKAGE_PATH=$ROS_PACKAGE_PATH:$(pwd)/Tools/sitl_gazebo     (12)
```

Po spuštění výše zmíněných příkazů je možné pustit se do práce s kolizním systémem. Na začátku však je otevřeno nové okno terminálu, kde je spuštěn Roscore. V prvním terminálu je nyní možné zadávat příkazy bez komplikací. Pro spuštění vývojového prostředí zadáme příkaz *roslaunch* včetně parametrů. Tento nástroj zprostředkovává jednoduché spuštění více uzlů lokálně. Není nutné znát kompletně kód. Pohyb v kódu probíhá pomocí klávesy tab, která našeptává možnosti roslaunch parametrů. [51]

```
roslaunch local_planner local_planner_sitl_3cam.launch (13)
```

Po zadání výše zmíněného kódu je v TUI terminálu vygenerováno logo PX4 a následně je spuštěný program rviz včetně grafického rozhraní táhel pro nastavení parametrů. V levé části programu se nachází panel se zobrazenými vlastnostmi, které lze zapínat, vypínat a přidávat. Ve výchozím nastavení jsou zde označeny vlastnosti, které mají být zobrazeny například svět či značení počátku. Takto spuštěný program však umožňuje pouze využití příkazů pro letovou misi. Například přednastavená letová mise, kdy bezpilotní prostředek obletí v bodech čtverec. Letové mise jsou spuštěny skrz příkaz *python*. Příkaz je spustitelným pouze ve složce */Firmware/integrationtests/python_src/px4_it/mavros* a musí být spuštěn v novém okně terminálu. Kde jsou opět nejprve vloženy příkazy pro source a export. Po spuštění příkazu *rosrun* je proveden let bezpilotního prostředku včetně vzletu a přistání (Obrázek 21).

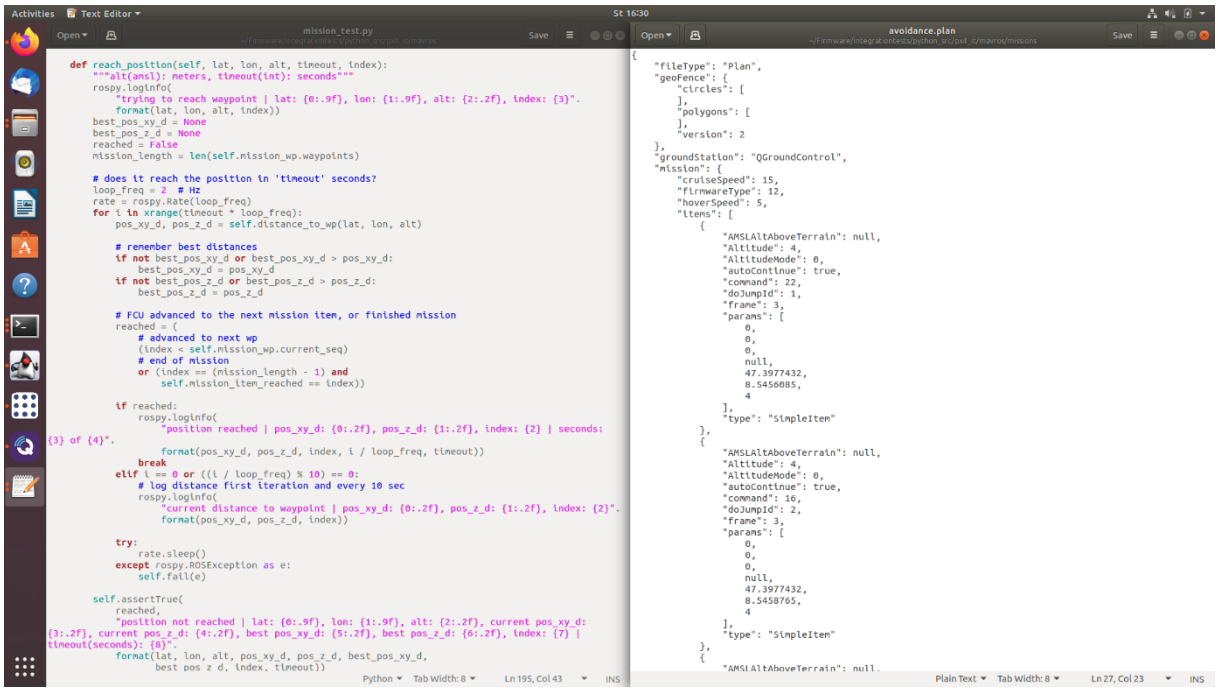
```
rosrun mission_test.py MC_mission_box.plan (14)
```



Obrázek 21: Provedení letové mise – letová mise čtverec

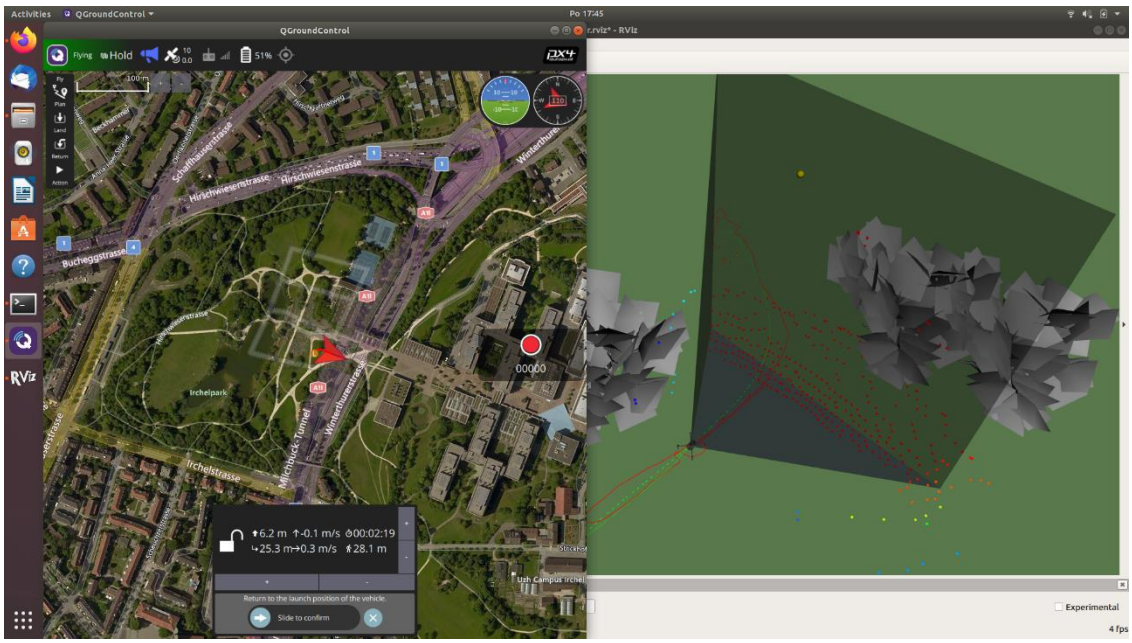
Provedení letové mise je první možností, jak by mohla být provedena implementace anti-kolizního systému bezpilotního prostředku. Toto řešení je rizikové v tom, že existuje velké množství souborů, které jsou mezi sebou provázány. Řešení skrz naprogramování se stává komplikovanějším. Díky vývojovému prostředí

Pythonu je možné dohledat soubory pro zahájení mise a také soubory letových plánů (Obrázek 22).



Obrázek 22: Kódová aplikace při naprogramování letového plánu

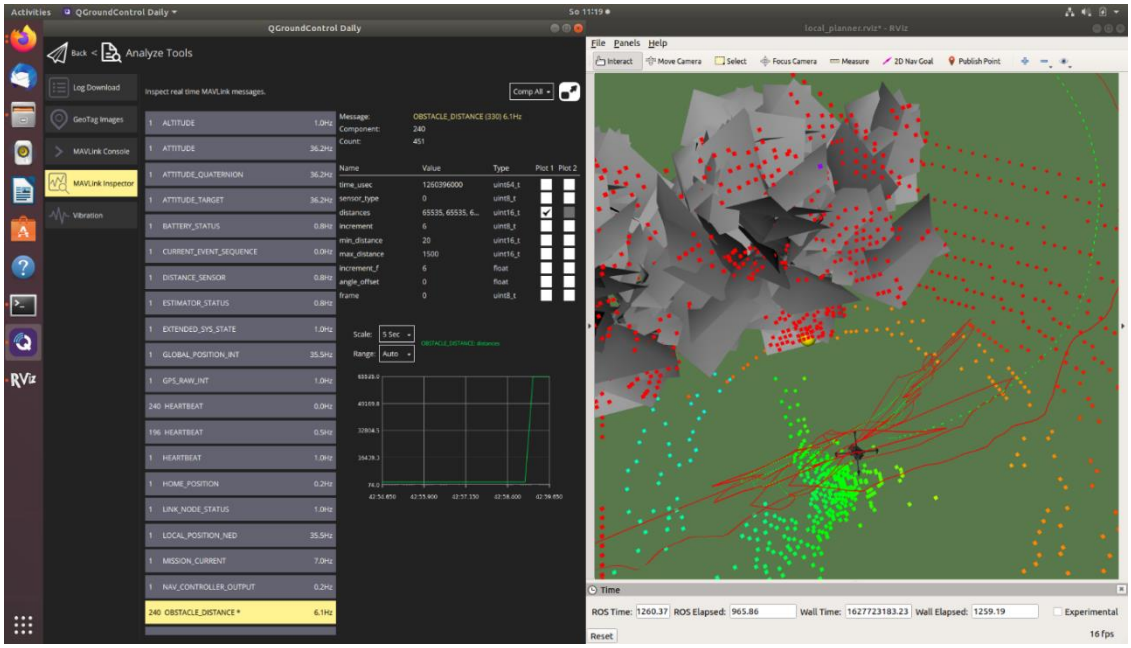
Cílem téhle práce není komplexní naprogramování Local Planneru tak, aby byl použitelný a detekoval správně překážky. Pro další řešení je nutné, aby byl spuštěn program pro datalinkový přenos. Po spuštění QGroundControlu je možné ovládat bezpilotní prostředek skrz softwarový joystick, který je součástí programu. Pro lepší znázornění je zapnut FOV senzoru (Obrázek 23). Díky zornému poli senzoru



Obrázek 23: Řídící stanice a zobrazení detekčního rozsahu v simulátoru rviz

je usnadněna práce a následné pochopení, kde by již měl senzor detekovat střet s překážkou. Kromě zorného pole byly vygenerovány i jednotlivé snímací body. Tyto snímací body demonstrují přímou vzdálenost od překážky až k bezpilotnímu prostředku. [49, 51]

V QGroundControlu lze zadávat příkazy kromě virtuálního joysticku i skrz definování letové trasy. Tato možnost zprostředkovává plně automatické řízení, kdy bezpilotní prostředek dokáže detekovat překážky. Před překážkou v definované vzdálenosti provádí manévr takový, aby se překážce vyhnul. Součástí GQroundControlu jsou analytické nástroje, které umožňují nastavit parametry prostředku a v minimálním množství je konfigurovat. MAVLink inspector umožňuje sledování jednotlivých procesů, které jsou spojené s aktuálním pohybem bezpilotního prostředku (Obrázek 24). [56]



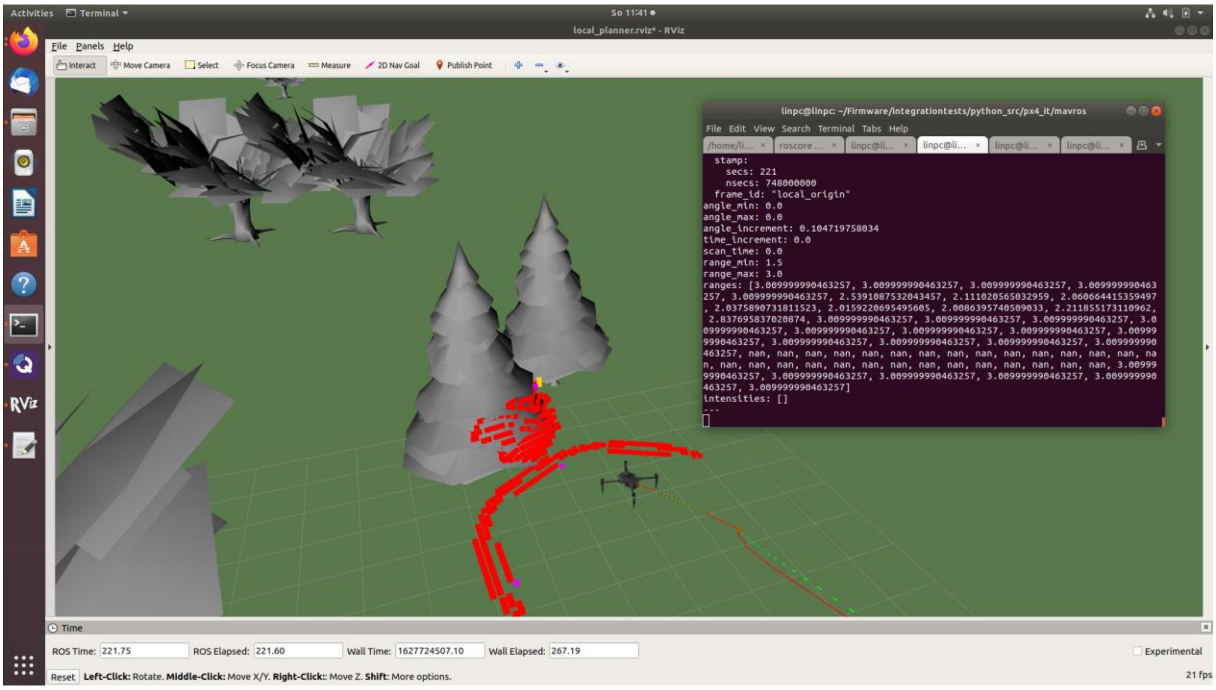
Obrázek 24: MAVLink inspector a dokončení letové mise s cílem v objektu

Pokud se stane, že cílová pozice bezpilotního prostředku je v takové blízkosti překážky, že této pozice není možné dosáhnout. Bezpilotní prostředek provádí lety tam a zpět a získává lepší k pozici k cílovému bodu. Avšak může dojít k zacyklení a poté se terminál včetně QGroundControlu předčasně ukončí. Po této fatální chybě je nutné spustit znovu veškeré programy. V našem případě tedy roscore, roslaunch s Local Plannerem a QGroundControl. Uzly je možné nadále sledovat pomocí ROS

příkazů. Vhodným příkazem pro sledování detekce překážky je příkaz *rostopic*. Příkaz je spuštěn v novém okně terminálu. [52]

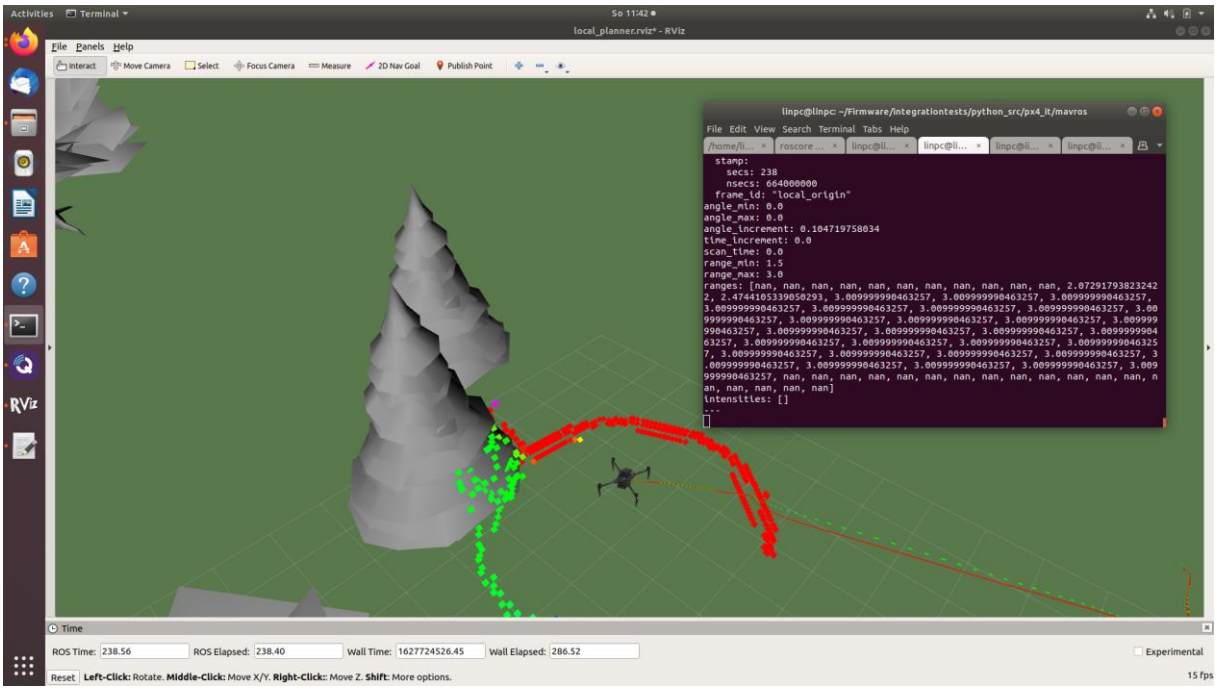
$$\text{rostopic echo /mavros/obstacle/send} \tag{15}$$

Tento příkaz otevírá topic s analýzou překážky, která se nachází v detekčním poli bezpilotního prostředku. Nyní je možná detekce objektů, které jsou prezentovány jako překážky. Detekce je zpracována v TUI terminálu, kde je spuštěn příkaz rostopic (Obrázek 25).



Obrázek 25: Detekce překážky v rviz simulátoru včetně zobrazení topicu detekce

Skrz detekci je zobrazena matice vzdáleností, v místě, kde jsou pouze hodnoty nan, je umožněn průlet. V místech, kde jsou hodnoty číselného typu, je známo, že generované bodové mračno infračerveného senzoru detekovalo blížící se překážku. V případě rotace bezpilotního prostředku bude tato matice naplňována dynamicky. Dynamický způsob naplňování značí, že při rotaci jsou vhodnější buňky naplněné hodnotami nan a jiné zase hodnotami pro detekování vzdálenosti překážky (Obrázek 26). [56]

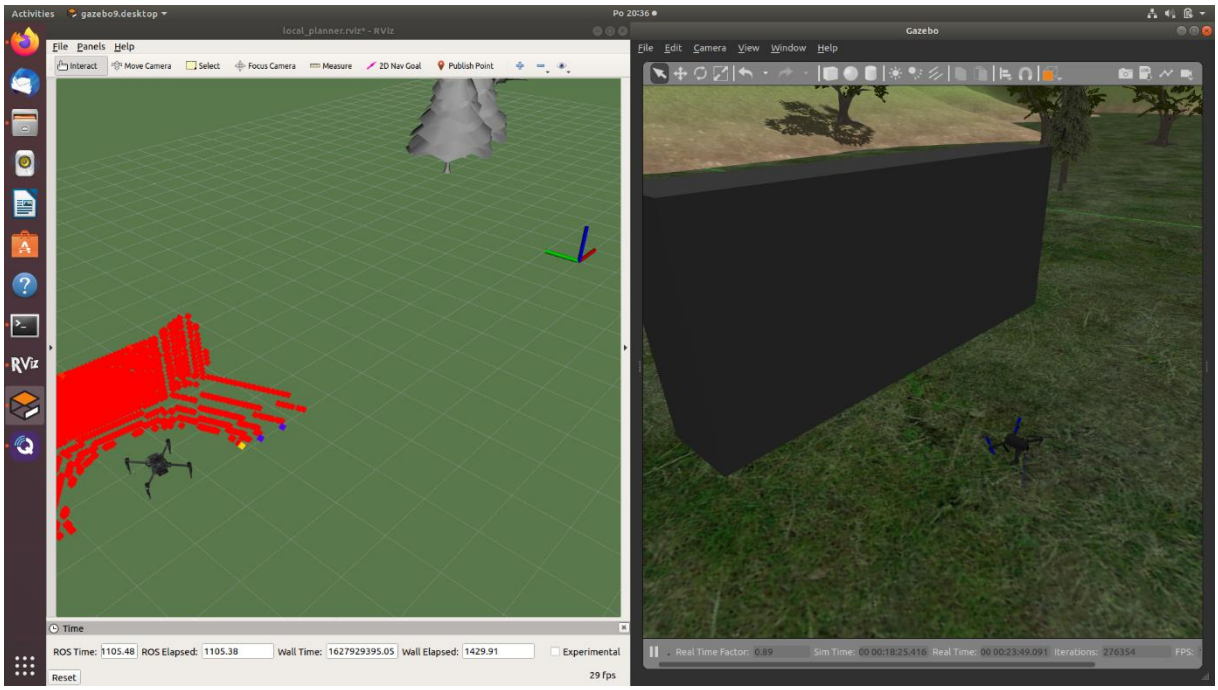


Obrázek 26: Detekce překážky a detekční hodnoty při natočení o 90°

Při jiném natočení jsou hodnoty dynamicky měněny. Z toho vyplývá, že nastavený detektor je nakonfigurován správně. Pro kompletní uskutečnění detekce je důležité vložit předmět, který nebude součástí této mapy. Pro vkládání předmětů je nutné otevřít prostředí ještě skrz *gazebo_ros*. V novém okně terminálu, které je spuštěno v kořenové složce PX4, je spuštěna sada příkazů source a export. Poté je využit další příkaz *roslaunch*.

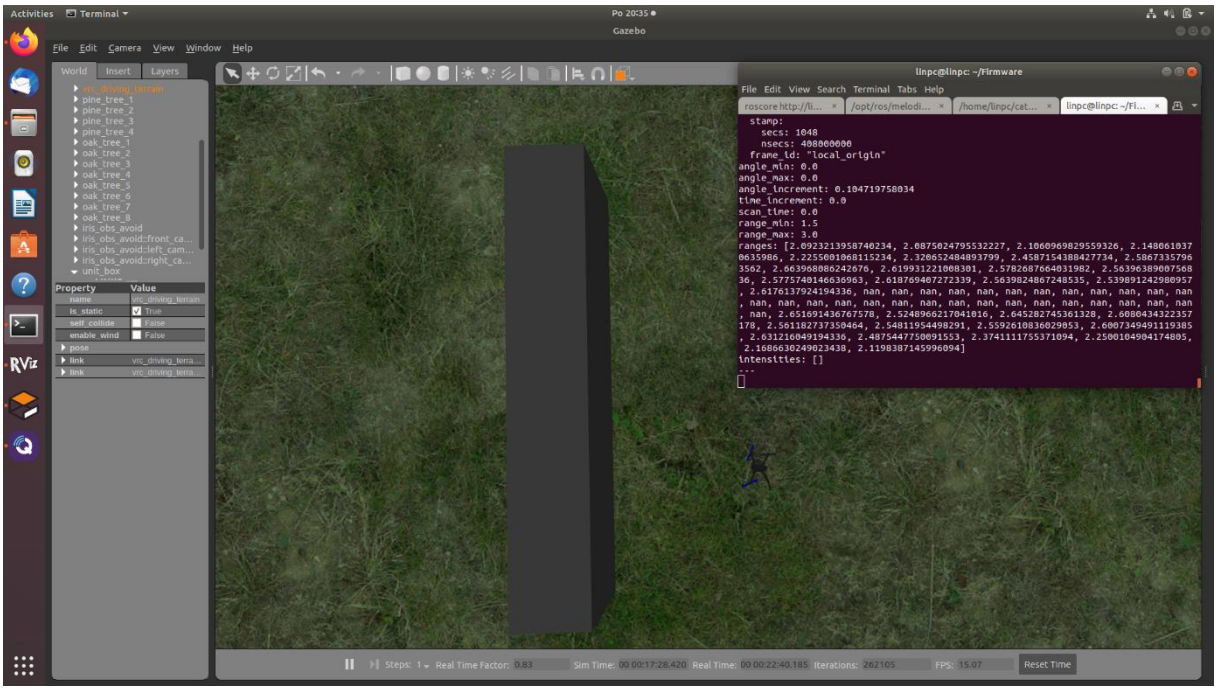
```
roslaunch gazebo_ros empty_world.launch (16)
```

Tento příkaz spustí GUI, ve kterém již není potlačena vykreslovací funkce. Prostředí je zpracováno skrz Gazebo, kde je implementován svět z Local Planneru. V tom prostředí lze vkládat objekty. V tomto prostředí je vložen čtverec s výchozím nastavením, který je upraven do zdi na pozadí je však stále spuštěn program rviz, pro demonstrování detekce překážky (Obrázek 27).



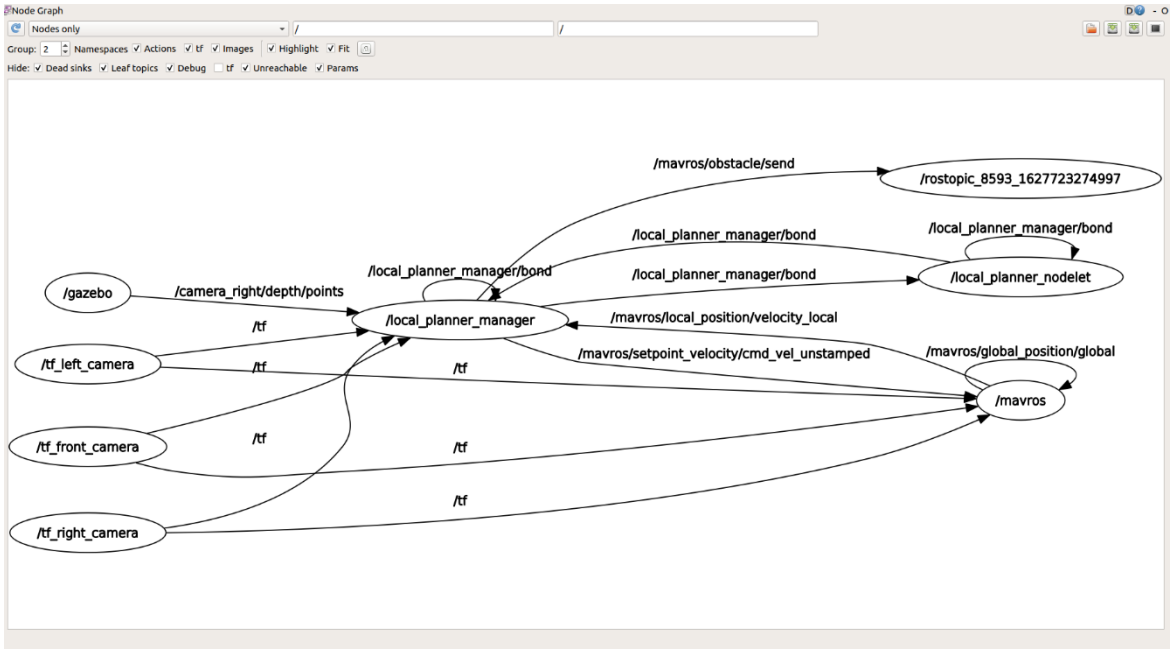
Obrázek 28: Zobrazení překážky pomocí bodového mračna a vizualizace

Detekční systém zachycuje pomocí mračného pole zeď. Pro rozšířenější analýzu detekčního systému prostředku je opět zavolán příkaz *rostopic*. Skrz tento příkaz je zcela zřejmé, že bezpilotní prostředek detekuje překážky v prostředí (Obrázek 28). Při natočení se hodnoty dynamicky mění.



Obrázek 27: Pohled z výšky na model Iris při detekci překážky

Pro souhrn spuštěných uzlů byl vygenerován graf pomocí příkaz *rqt_graph*. Součástí grafu jsou zahrnuty použité uzly (Obrázek 29). Nástroj pro grafickou vizualizaci je používán zejména tehdy, kdy vývoj systému vyžaduje větší množství komponent. Při růstu aplikace je důležité si prohlédnout, který uzel komunikuje s dalším uzlem.



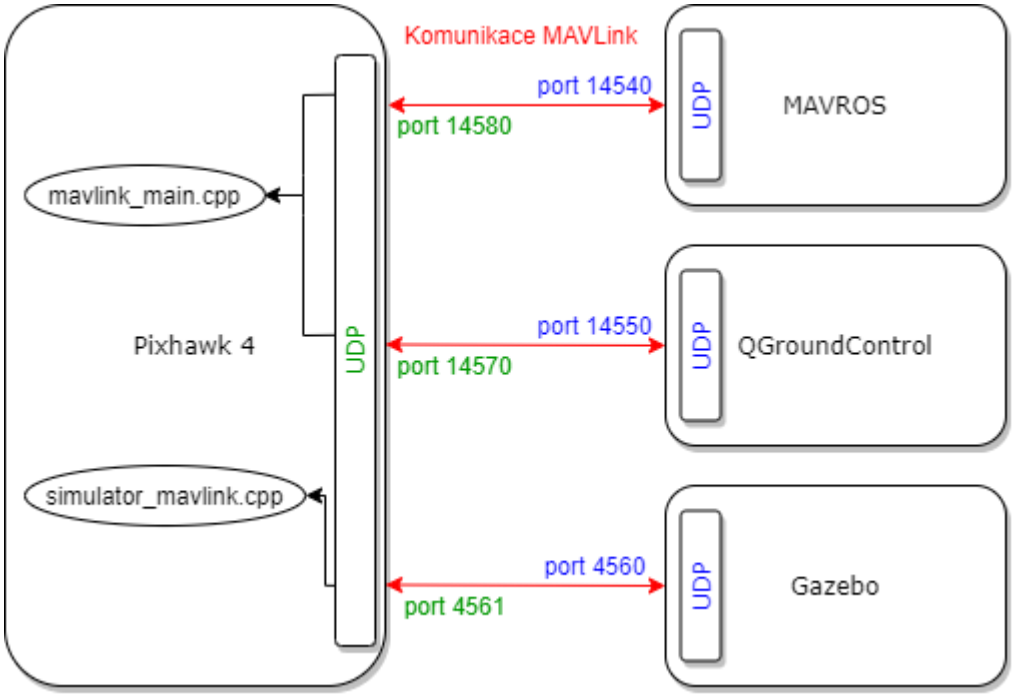
Obrázek 29: Grafická vizualizace spuštěných uzlů včetně znázorněné komunikace

3.6. Konfigurace celého bezpilotního prostředku

Navržení bezpilotního prostředku, který detekuje překážky pomocí senzorů Lanbao PSK – CM8JL65 – CC5, spočívá v umístění senzorů na správné místo. Vzhledem ke konstrukci bezpilotního prostředku je vhodným řešením umístit senzory na každou nohu, která je součástí letové plošiny. Maximální užitečná nosnost bezpilotního prostředku Iris je 400 g. V našem případě by klesla tato využitelná nosnost o 40 gramů, což je $^{1}/_{10}$ této váhy. Takovéto zatížení nezpůsobí nestabilitu systému. Nabízí se ještě druhá možnost instalace radarového senzoru, který bude rotovat. Při tomto řešení však vychází na povrch, že bezpilotní prostředek získá o jeden rotační prvek navíc. Dále frekvence bude nižší než je standardem a může docházet k nestabilitě systému.

3.7. Shrnutí kapitoly

Instalace vychází z metodiky zapojení virtuální základové desky s jednotlivými komponenty. Diagram je tvořen pomocí jednotlivých komponent, které jsou součástí simulačního prostředí. (Obrázek 30) Spojení mezi jednotlivými komponenty je řízeno skrz výchozí nastavení jednotlivých UDP portů. Pro správnou funkcionalitu je zajištěna komunikace MAVLink, která zprostředkovává veškeré přenosy zpráv. Optimální zprostředkování adaptace ROS topiců s přenosem MAVLinkovým zpráv je zajištěno



Obrázek 30: Zapojení virtuální základové desky s jednotlivými komponenty

sadou funkcí a procedur MAVROS. Přičemž MAVROS funguje jako uzel ROSu. Kromě efektivní komunikace musí být dále zajištěna komunikace s řídící stanicí, bez této stanice by nebyl umožněný pohyb bezpilotního prostředku. Instalovaná řídící stanice QGroundControl poskytuje autorizovaný přístup řízení bezpilotního prostředku. Přístup je možné ovládat pomocí nadefinování leteckého plánu či pomocí virtuálního joysticku. Ke kompletnímu testování bezpilotního prostředku je nainstalován program Gazebo, který vede k vizualizaci ve 3D prostoru.

Simulinkové vyjádření signálu PID kontroléru vychází ze získaných hodnot z řídící stanice QGroundControl. Porovnání jednotlivých signálů však není zcela totožné zejména z důvodu, že nebyl importován souhrn dat z QGroundControlu. Porovnání proběhlo pouze okometricky a nebylo tedy provedeno statistické ověření

diskrétních dat z QGroundControl. V Simulinkovém modelu byly převzaty jednotlivé hodnoty gain a byla vytvořena rovnice procesu, která byla totožná pro domácí robotické vysavače. Následná validace těchto dat proběhla v simulačním prostředí Gazebo. Detekce překážky byla znázorněna pomocí topicu pro detekci překážky.

Snímací senzor reagoval správným způsobem na překážku a to tak, že při pohybu od překážky nebo k překážce se jednotlivé hodnoty matice bodového mračna měnily. Při pohybu, kdy se bezpilotní prostředek přiblížil k limitní minimální hranici, byla zablokována funkce vpřed. V tuto chvíli nebylo možné, aby bezpilotní prostředek setrval v pohybu vpřed. Při této pozici byl povolený pohyb, který provedl dílčí pohyb vzad nebo rotaci tak, aby senzor nebyl nadále omezován překážkou.

Závěr

Cílem této bakalářské práce bylo vytvoření návrhu anti-kolizního systému, který dokáže včasně detekovat překážku. Byla vykonána studie, která se opírá o testování kolizní systému v simulačním prostředí Gazebo. Gazebo je však pouze simulátorem, není možné, aby simulační prostředí bylo spuštěno samostatně. Tento fakt znázorňuje, že vývoj anti-kolizního systému musí být navržen v rámci prostředí, které je spojeno s vývojem robotických zařízení. Softwarové prostřední Linux tento krok splňuje, zejména distribuce Ubuntu 18.04, která zajišťuje plnou kompatibilitu s prostředím ROS. Na základě toho byl navržen pracovní postup, který vedl k instalaci ROSu a Gazebo na operačním systému Linux Ubuntu. Průběh celé instalace nesl s sebou problémy v podobě chybějících balíčků. Tyto chyby byly následně vyladěny až do podoby, kdy byla zajištěna plná funkcionalita celého systému. Po instalaci následoval výběr vhodného senzoru pro detekci překážek. Senzor pro měření vzdálenosti, Lanbao PSK – CM8JL65 – CC5, byl zvolen díky menšímu detekčnímu poli. Menší detekční pole zajišťuje nižší spotřebu energie z energetického systému. Během výběru byla zohledněna hmotnost senzoru nosné kapacitě letové plošiny.

Následná validace zapojeného senzoru proběhla v řídící stanici QGroundControl. V řídící stanici bylo provedeno ladění PID kontroléru tak, aby jednotlivé složky PID kontroléru vyvážily stabilitu systému. V následné analýze nebyly hodnoty průběhu signálu statisticky ověřené z důvodu absence exportu dat. Tento problém vedl k vytvoření modelu PID kontroléru v prostředí Matlab/Simulink a vychází z anti-kolizního systému domácích robotických vysavačů. Vzhledem k nezískání diskrétních hodnot z QGroundControlu, byly jednotlivé grafy porovnány pouze okometricky. Průběhy jednotlivých signálů byly ustáleny až do homeostáze. Po ustálení byla provedena validace detektoru v Gazebo.

V simulátoru Gazebo, byla provedena verifikační detekce překážky. Tato detekce vedla k naplnění matice detektoru a při přiblížení do limitní minimální vzdálenosti byl bezpilotní prostředek zablokován o pohyb vpřed. Z toho vyplývá, že následná validace v systému Gazebo byla úspěšná.

Řešení návrhu anti-kolizního systému vede k určitým aspektům, které by mohly být použity jiným způsobem. Například prvním aspektem by mohl být testování v prostředí, které je ze všech stran obestavěné stěnami a zároveň existuje nějaký strop. Při takovém řešení by bylo nutné vybrat senzor, který bude snímat vzdálenost od stropní plochy. Tento pátý senzor, pokud by měl rotující část, by mohl ovlivnit stabilitu bezpilotního prostředku. Kromě přidání dalšího senzoru by přicházelo v úvahu, zda by nebyla celá konstrukce bezpilotního prostředku jiná. Tato konstrukce by mohla mít tlakový senzor, který by detekoval strop. Různé druhy senzorů mohou významně zatížit základovou desku, zejména procesor. Zatížení procesoru je otázkou, která nejspíše není řešitelná skrz softwarové testovací prostředí.

Dále by bylo otázkou, zda vybrané testovací prostředí Gazebo včetně celého systému bylo nejvhodnější cestou. Musíme brát v úvahu, že instalace Linux a přídavných komponentů zabrala mnoho hodin. Během těchto instalací vycházela na povrch nová chybová hlášení, která byla nutná ještě před spuštěním Gazebo simulace.

Použité zdroje

[1] YASIN, Jawad N., Sherif A. S. MOHAMED, Mohammad-Hashem HAGHBAYAN, Jukka HEIKKONEN, Hannu TENHUNEN a Juha PLOSILA, 2020. Unmanned Aerial Vehicles (UAVs): Collision Avoidance Systems and Approaches. IEEE Access [online]. 8, 105139–105155. Dostupné z: doi:10.1109/access.2020.3000064

[2] MACRINA, Giusy, Luigi DI PUGLIA PUGLIESE, Francesca GUERRIERO a Gilbert LAPORTE, 2020. Drone-aided routing: A literature review. Transportation Research Part C: Emerging Technologies [online]. 120, 102762. Dostupné z: doi:10.1016/j.trc.2020.102762

[3] PETRITOLI, Enrico, Fabio LECCESE a Lorenzo CIANI, 2018. Reliability and Maintenance Analysis of Unmanned Aerial Vehicles. Sensors [online]. 18(9), 3171. Dostupné z: doi:10.3390/s18093171

[4] ČSN 31 0001. Letectví a kosmonautika - Terminologie. Praha: Český normalizační institut

[5] BURDZIAKOWSKI, Pawel, 2018. UAV Design and Construction for Real Time Photogrammetry and Visual Navigation. In: 2018 Baltic Geodetic Congress (BGC Geomatics): 2018 Baltic Geodetic Congress (BGC Geomatics) [online]. B.m.: IEEE. Dostupné z: doi:10.1109/bgc-geomatics.2018.00076

[6] NAŘÍZENÍ KOMISE V PŘENESENÉ PRAVOMOCI (EU) 2019/945. EUR-lex [online]. [cit. 2020-06- 17]. Dostupné z: https://eur-lex.europa.eu/legalcontent/CS/TXT/PDF/?uri=CELEX:32019R0945&from=C

[7] PROVÁDĚCÍ NAŘÍZENÍ KOMISE (EU) 2019/947. EUR-lex [online]. [cit. 2020-06-17]. Dostupné z: https://eur-lex.europa.eu/legal-content/CS/TXT/PDF/?uri=CELEX:32019R0947&from=IT

[8] GIORDAN, Daniele, Marc S. ADAMS, Irene AICARDI, Maria ALICANDRO, Paolo ALLASIA, Marco BALDO, Pierluigi DE BERARDINIS, Donatella DOMINICI, Danilo GODONE, Peter HOBBS, Veronika LECHNER, Tomasz NIEDZIELSKI, Marco PIRAS, Marianna ROTILIO, Riccardo SALVINI, Valerio SEGOR, Bernadette SOTIER a Fabrizio TROILO, 2020. The use of unmanned aerial vehicles (UAVs) for engineering geology applications. Bulletin of Engineering Geology and the Environment [online]. 79(7), 3437–3481. Dostupné z: doi:10.1007/s10064-020-01766-2

[9] CHEN, Yun, Nuria GONZALEZ-PRELCIC a Robert W. HEATH, 2020. Collision-Free UAV Navigation with a Monocular Camera Using Deep Reinforcement Learning. In: 2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP): 2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP) [online]. B.m.: IEEE. Dostupné z: doi:10.1109/mlsp49062.2020.9231577

[10] PALOSSI, Daniele, Francesco CONTI a Luca BENINI, 2019. An Open Source and Open Hardware Deep Learning-Powered Visual Navigation Engine for Autonomous Nano-UAVs. In: 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS): 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS) [online]. B.m.: IEEE. Dostupné z: doi:10.1109/dcoss.2019.00111

[11] WEI, Fu-Shang, Edward MOORE a Alfred GATES, 2015. An Intermeshing Rotor Helicopter Design and Test. In: 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference: 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference [online]. B.m.: American Institute of Aeronautics and Astronautics. Dostupné z: doi:10.2514/6.2015-0564

[12] HILAL, Ala' Abu a Thabet MISMAR, 2020. Drone Positioning System Based on Sound Signals Detection for Tracking and Photography. In: 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON): 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) [online]. B.m.: IEEE. Dostupné z: doi:10.1109/iemcon51383.2020.9284851

[13] VENNA, Trinadh V S N, Sarosh PATEL a Tarek SOBH, 2020. Application of Image-Based Visual Servoing on Autonomous Drones. In: 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA): 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA) [online]. B.m.: IEEE. Dostupné z: doi:10.1109/iciea48937.2020.9248119

[14] MERIZALDE, Darwin, Wilbert G. AGUILAR a Marco CALDERÓN, 2020. Autonomous Navigation Based on Proportional Controller with GPS Setpoint for UAV in External Environments. In: Smart Innovation, Systems and Technologies [online]. B.m.: Springer Singapore, s. 89–99. Dostupné z: doi:10.1007/978-981-15-4875-8_8

[15] GONZÁLEZ-DESANTOS, L. M., J. MARTÍNEZ-SÁNCHEZ, H. GONZÁLEZ-JORGE a P. ARIAS, 2020. PATH PLANNING FOR INDOOR CONTACT INSPECTION TASKS WITH UAVS. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences [online]. XLIII-B4-2020, 345–351. Dostupné z: doi:10.5194/isprs-archives-xliii-b4-2020-345-2020

[16] Anon., 2020. Smart City IoT: Co-Designing Trustworthy Public Safety Drones for Indoor Flight Missions. In: The 19th European Conference on Cyber Warfare: Proceedings of the 19th European Conference on Cyber Warfare [online]. B.m.: ACPI. Dostupné z: doi:10.34190/ews.20.509

[17] CHEN, J., O. E. MORA a K. C. CLARKE, 2018. ASSESSING THE ACCURACY AND PRECISION OF IMPERFECT POINT CLOUDS FOR 3D INDOOR MAPPING AND MODELING. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences [online]. IV-4/W6, 3–10. Dostupné z: doi:10.5194/isprs-annals-iv-4-w6-3-2018

[18] BROWN, Liam, Robert CLARKE, Ali AKBARI, Ujjar BHANDARI, Sara BERNARDINI, Puneet CHHABRA, Ognjen MARJANOVIC, Thomas RICHARDSON a Simon WATSON, 2020. The Design of Prometheus: A Reconfigurable UAV for Subterranean Mine Inspection. Robotics [online]. 9(4), 95. Dostupné z: doi:10.3390/robotics9040095

[19] LEE, Yi-Sheng a Jih-Gau JUANG, 2019. Color identification for quadcopter flight control and object inspection. Advances in Mechanical Engineering [online]. 11(2), 168781401882255. Dostupné z: doi:10.1177/1687814018822559

[20] GONZALEZ-CASTANO, Francisco Javier, Felipe GIL-CASTINEIRA, David RODRIGUEZ-PEREIRA, Jose Angel REGUEIRO-JANEIRO, Silvia GARCIA-MENDEZ a David CANDAL-VENTUREIRA, 2021. Self-Corrective Sensor Fusion for Drone Positioning in Indoor Facilities. IEEE Access [online]. 9, 2415–2427. Dostupné z: doi:10.1109/access.2020.3048194

[21] VOOSEN, Paul, 2019. NASA to fly drone on Titan. Science [online]. 365(6448), 15.1-15. Dostupné z: doi:10.1126/science.365.6448.15-a

[22] PERKINS, Sid, 2016. Elon Musk's path to Mars begins with Red Dragon—but what science will it do? Science [online]. Dostupné z: doi:10.1126/science.aah7355

[23] LEE, Yi-Sheng a Jih-Gau JUANG, 2019. Color identification for quadcopter flight control and object inspection. Advances in Mechanical Engineering [online]. 11(2), 168781401882255. Dostupné z: doi:10.1177/1687814018822559

[24] CONTI, Massimo, Simone ORCIONI, Francesco GREGORINI, Pietro ANTONELLI, Marco GIAMMARINI, Rocco D'APARO a Federico ROBUFFO, 2019. Performance analysis of an indoor and outdoor real time localization system. In: 2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT): 2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT) [online]. B.m.: IEEE. Dostupné z: doi:10.1109/isce.2019.8901010

[25] SANTAMARINA CAMPOS, DE MIGUEL MOLINA, KRONER, 2018, Development Of An Indoor Drone Designed For The Needs Of The Creative Industries. In: 2018 International Journal of Mechanical and Mechatronics Engineering [online]. B.m.: ResearcheGate. Dostupné z: doi: 10.1999/1307-6892/10009012

[26] TIPANTUÑA-TOPANTA, Giovanny-Javier, Francisco ABAD, Ramón MOLLÁ, Jose-Luis POZA-LUJAN a Juan-Luis POSADAS-YAGÜE, 2018. Intelligent Flight in Indoor Drones. In: Distributed Computing and Artificial Intelligence, 15th International Conference [online]. B.m.: Springer International Publishing, s. 247–254. Dostupné z: doi:10.1007/978-3-319-94649-8_30

[27] KAPOOR, Rohan, Alessandro G. GARDI a Roberto SABATINI, 2019. A Multistatic Ultrasonic Navigation System for GNSS-denied Environments. In: AIAA Scitech 2019 Forum: AIAA Scitech 2019 Forum [online]. B.m.: American Institute of Aeronautics and Astronautics. Dostupné z: doi:10.2514/6.2019-1930

[28] OSKIPER, Taragay, Zhiwei ZHU, Supun SAMARASEKERA a Rakesh KUMAR, 2007. Visual Odometry System Using Multiple Stereo Cameras and Inertial Measurement Unit. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition: 2007 IEEE Conference on Computer Vision and Pattern Recognition [online]. B.m.: IEEE. Dostupné z: doi:10.1109/cvpr.2007.383087

[29] KANG, Hyun-gyu, Byoung-kyun KIM a Wangheon LEE, 2015. Indoor localization of DRON using vision based sensor fusion. In: 2015 15th International Conference on Control, Automation and Systems (ICCAS): 2015 15th International Conference on Control, Automation and Systems (ICCAS) [online]. B.m.: IEEE. Dostupné z: doi:10.1109/iccas.2015.7364757

[30] LI, Jun a Yuntang LI, 2011. Dynamic analysis and PID control for a quadrotor. In: 2011 IEEE International Conference on Mechatronics and Automation (ICMA): 2011 IEEE International Conference on Mechatronics and Automation [online]. B.m.: IEEE. Dostupné z: doi:10.1109/icma.2011.5985724

[31] GOMEZ, Victor, Nicolas GOMEZ, Jorge RODAS, Enrique PAIVA, Maarouf SAAD a Raul GREGOR, 2020. Pareto Optimal PID Tuning for Px4-Based Unmanned Aerial Vehicles by Using a Multi-Objective Particle Swarm Optimization Algorithm. Aerospace [online]. 7(6), 71. Dostupné z: doi:10.3390/aerospace7060071

[32] Anon., 2006. PID control. IEEE Control Systems [online]. 26(1), 30–31. Dostupné z: doi:10.1109/mcs.2006.1580151

[33] NAFEA, Marwan, Abdul Rasyid Mohammad ALI, Jeevananthan BALIAH a Mohamed Sultan MOHAMED ALI, 2018. Metamodel-based Optimization of a PID Controller Parameters for a Coupled-tank System. TELKOMNIKA (Telecommunication Computing Electronics and Control) [online]. 16(4), 1590. Dostupné z: doi:10.12928/telkomnika.v16i4.9069

[34] ZUL AZFAR, A. a D. HAZRY, 2011. Simple GUI design for monitoring of a remotely operated quadrotor unmanned aerial vehicle (UAV). In: its Applications (CSPA): 2011 IEEE 7th International Colloquium on Signal Processing and its Applications [online]. B.m.: IEEE. Dostupné z: doi:10.1109/cspa.2011.5759836

[35] SUNGJU PARK, VISHNU PRASADH SUGUMAR a Vikram Louis MICHAEL, 2019. Modelling and Simulation of a Collision Avoidance System [online]. 2019. B.m.: Unpublished. Dostupné z: doi:10.13140/RG.2.2.32225.

[36] ALMASRI, Marwah M., Abrar M. ALAJLAN a Khaled M. ELLEITHY, 2016. Trajectory Planning and Collision Avoidance Algorithm for Mobile Robotics System. IEEE Sensors Journal [online]. 16(12), 5021–5028. Dostupné z: doi:10.1109/jsen.2016.2553126

[37] OLIVARES-MENDEZ, Miguel A., Somasundar KANNAN a Holger VOOS, 2014. Setting up a testbed for UAV vision based control using V-REP & ROS: A case study on aerial visual inspection. In: 2014 International Conference on Unmanned Aircraft Systems (ICUAS): 2014 International Conference on Unmanned Aircraft Systems (ICUAS) [online]. B.m.: IEEE. Dostupné z: doi:10.1109/icuas.2014.6842285

[38] GARCÍA, Jesús a Jose M. MOLINA, 2020. Simulation in real conditions of navigation and obstacle avoidance with PX4/Gazebo platform. Personal and Ubiquitous Computing [online]. Dostupné z: doi:10.1007/s00779-019-01356-4

[39] YOON, Sugjoon, Dongcho SHIN, Younghoon CHOI a Kyungtae PARK, 2021. Development of a Flexible and Expandable UTM Simulator Based on Open Sources and Platforms. Aerospace [online]. 8(5), 133. Dostupné z: doi:10.3390/aerospace8050133

[40] MENG, Wei, Yuchao HU, Jiaxin LIN, Feng LIN a Rodney TEO, 2015. ROS+unity: An efficient high-fidelity 3D multi-UAV navigation and control simulator in GPS-denied environments. In: IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society: IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society [online]. B.m.: IEEE. Dostupné z: doi:10.1109/iecon.2015.7392488

[41] REUDENBACH, Christoph, 2017. UAV Mission Planner. Unpublished [online]. Dostupné z: doi:10.13140/RG.2.2.24955.82722

[42] GOKTOGAN, A.H., E. NETTLETON, M. RIDLEY a S. SUKKARIEH, nedatováno. Real time Multi-UAV Simulator. In: IEEE International Conference on Robotics and Automation. IEEE ICRA 2003: 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422) [online]. B.m.: IEEE. Dostupné z: doi:10.1109/robot.2003.1242004

[43] WANG, Xiaolong, Hui YUE, Guangliang LIU a Zhao ZHAO, 2011. The Application of COMSOL Multiphysics in Direct Current Method Forward Modeling. Procedia Earth and Planetary Science [online]. 3, 266–272. Dostupné z: doi:10.1016/j.proeps.2011.09.093

[44] ABZUG, Charles, 2004. Linux Operating System [online]. 3. leden 2004. B.m.: John Wiley & Sons, Inc. Dostupné z: doi:10.1002/047148296x.tie108

[45] BARNETT, James, 2015. Linux. In: Drupal 8 for Absolute Beginners [online]. B.m.: Apress, s. 271–292. Dostupné z: doi:10.1007/978-1-4302-6467-5_17

[46] JOSEPH, Lentin, 2018. Programming with ROS. In: Robot Operating System for Absolute Beginners [online]. B.m.: Apress, s. 171–236. Dostupné z: doi:10.1007/978-1-4842-3405-1_5

[47] KOENIG, N. a A. HOWARD, nedatováno. Design and use paradigms for gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat.

No.04CH37566): 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566) [online]. B.m.: IEEE. Dostupné z: doi:10.1109/iros.2004.1389727

[48] KENDOUL, Farid, 2012. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. Journal of Field Robotics [online]. 29(2), 315–378. Dostupné z: doi:10.1002/rob.20414

[49] ANGGRAENI, P., M. MRABET, M. DEFOORT a M. DJEMAI, 2018. Development of a wireless communication platform for multiple-mobile robots using ROS. In: 2018 6th International Conference on Control Engineering & Information Technology (CEIT): 2018 6th International Conference on Control Engineering & Information Technology (CEIT) [online]. B.m.: IEEE. Dostupné z: doi:10.1109/ceit.2018.8751845

[50] EBEID, Emad, Martin SKRIVER a Jie JIN, 2017. A Survey on Open-Source Flight Control Platforms of Unmanned Aerial Vehicle. In: 2017 Euromicro Conference on Digital System Design (DSD): 2017 Euromicro Conference on Digital System Design (DSD) [online]. B.m.: IEEE. Dostupné z: doi:10.1109/dsd.2017.30

[51] PIETRZIK, S a B CHANDRASEKARAN, 2019. Setting up and Using ROS-Kinetic and Gazebo for Educational Robotic Projects and Learning. Journal of Physics: Conference Series [online]. 1207, 012019. Dostupné z: doi:10.1088/1742-6596/1207/1/012019

[52] RAMIREZ-ATENCIA, Cristian a David CAMACHO, 2018. Extending QGroundControl for Automated Mission Planning of UAVs. Sensors [online]. 18(7), 2339. Dostupné z: doi:10.3390/s18072339

[53] MUJUMDAR, Anusha a Radhakant PADHI, 2011. Reactive Collision Avoidance of Using Nonlinear Geometric and Differential Geometric Guidance. Journal of Guidance, Control, and Dynamics [online]. 34(1), 303–311. Dostupné z: doi:10.2514/1.50923

[54] Lanbao PSK - cmjl65 - CC5. Https://docs.px4.io [online]. Curych: Lorenz Meier, 2014 [cit. 2021-7-4]. Dostupné z: https://docs.px4.io/v1.9.0/en/sensor/cm8jl65_ir_distance_sensor.html

[55] Anon., 2018. Matlab Simulink Models. In: Electric Distribution Systems, Second Edition [online]. B.m.: John Wiley & Sons, Inc., s. 583–587. Dostupné z: doi:10.1002/9781119509332.app2

[56] PX4 Avoidance - install. Github.com [online]. California: GitHub, 2007 [cit. 2021-6-7]. Dostupné z: https://github.com/PX4/PX4-Avoidance

[57] ROS - install. Docs.px4.io [online]. Curych: Lorenz Meier, 2014 [cit. 2021-6-7]. Dostupné z: https://docs.px4.io/master/en/simulation/ros_interface.html

[58] Konstrukce bezpilotních prostředků. Imgur.com [online]. Reykjavik: Withheld for Privacy Purposes, 2009 [cit. 2021-8-6]. Dostupné z: https://i.imgur.com/UGWarda.jpg

[59] MQ - Reaper. Gstatic.com [online]. California: Google, 2008 [cit. 2021-8-6]. Dostupné z: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQKz6dkS-AJeM4u-m-MhFDA6zib3HY1ddq2_A&usqp=CAU

[60] Doručovací dron Amazonu. Time.com [online]. New York: MarkMonitor, 1993 [cit. 2021-8-6]. Dostupné z: https://api.time.com/wp-content/uploads/2014/11/amazon-drone-delivery.jpg?w=824&quality=70

[61] NIKUNJ MEHTA, DHARMENDRA CHAUHAN, SAGAR PATEL a SIDDHARTH MISTRY, 2017. Design of HMI Based on PID Control of Temperature. International Journal of Engineering Research and [online]. V6(05). Dostupné z: doi:10.17577/ijertv6is050074

[62] Odezva PID kontroleru. Umich.edu [online]. California: University of Michigan, 1985 [cit. 2021-8-6]. Dostupné z: https://ctms.engin.umich.edu/CTMS/Content/Introduction/Control/PID/html/Introduction_ControlPID_02.png

[63] ROS komunikační diagram. Medium.com [online]. Seattle: Amazon, 1998 [cit. 2021-8-6]. Dostupné z: https://miro.medium.com/max/1400/0*u8BaVmbpkLARxdNS.jpg

[64] Gazebo simulátor. Ytimg.com [online]. California: Google, 2007 [cit. 2021-8-6]. Dostupné z: https://i.ytimg.com/vi/WnHK_x_a17g/maxresdefault.jpg

Seznam obrázků

Obrázek 1: Konstrukční typy bezpilotních prostředků [58] 10
Obrázek 2: Kategorizace bezpilotních prostředků 13
Obrázek 3: MQ - 9 Reaper [59] 15
Obrázek 4: Doručovací bezpilotní prostředek Amazonu [60] 16
Obrázek 5: ELIOS 2 - důlní bezpilotní prostředek [18] 18
Obrázek 6: Pořízení 3D mapping fotografie z Elios 2 [18] 19
Obrázek 7: Schéma používaných technologií 22
Obrázek 8: Schématický model PID kontroleru [61] 24
Obrázek 9: Ustálený signál aplikace PID kontroleru [62] 25
Obrázek 10: Rozdělení softwarového testovacího prostředí 28
Obrázek 11: Komunikační úrovně ROSu [63] 31
Obrázek 12: Gazebo s implementací bezpilotního prostředku Iris [64] 32
Obrázek 13: Grafické zpracování jmavsim včetně absence kolizního detektoru 35
Obrázek 14: Výchozí nastavení Gazebo s šedým podkladem a modelem Iris 36
Obrázek 15: Úspěšné provedení buildu v terminálu 38
Obrázek 16: Infračervený sensor Lanbao PSK - CM8JL65 - CC5 [54] 39
Obrázek 17: Zobrazení signálu Roll PID kontroleru v QGroundControlu 40
Obrázek 18: Model PID kontroleru v Simulinku 41
Obrázek 19: Simulinkový model s připojeným náhledem 42
Obrázek 20: Simulace průběhu signálu do fáze homeostáze 43
Obrázek 21: Provedení letové mise – letová mise čtverec 44
Obrázek 22: Kódová aplikace při naprogramování letového plánu 45
Obrázek 23: Řídící stanice a zobrazení detekčního rozsahu v simulátoru rviz 45
Obrázek 24: MAVLink inspector a dokončení letové mise s cílem v objektu 46
Obrázek 25: Detekce překážky v rviz simulátoru včetně zobrazení topicu detekce 47
Obrázek 26: Detekce překážky a detekční hodnoty při natočení o 90° 48
Obrázek 27: Pohled z výšky na model Iris při detekci překážky 49
Obrázek 28: Zobrazení překážky pomocí bodového mračna a vizualizace 49
Obrázek 29: Grafická vizualizace spuštěných uzlů včetně znázorněné komunikace 50
Obrázek 30: Zapojení virtuální základové desky s jednotlivými komponenty 51

Seznam tabulek

Tabulka 1: Souhrn chybovosti technologií rádio frekvence 23