Diplomová práce

České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Návrh řízení pro hybridní VTOL bezpilotní prostředek

Bc. Martin Král

Vedoucí práce: Ing. Milan Rollo Ph.D.
Obor: Kybernetika a robotika
Leden 2018

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Martin K r á l

Studijní program: Kybernetika a robotika

Obor: Robotika

Název tématu: Návrh řízení pro hybridní VTOL bezpilotní prostředek

Pokyny pro vypracování:

1. Seznamte se s konceptem hybridních VTOL bezpilotních prostředků, popište jejich výhody a nevýhody ve srovnání s klasickými konstrukcemi.
2. Zpracujte přehled komerčních i open source/hardware autopilotů podporujících let bezpilotních prostředků v hybridním režimu, porovnejte jejich parametry.
3. Vytvořte model Vámi zvoleného hybridního bezpilotního prostředku v prostředí Matlab/Simulink.
4. Zvolte vhodnou jednotku autopilota umožňující řízení Vámi zvoleného bezpilotního prostředku.
5. Identifikujte vhodnou konfiguraci parametrů řídicích smyček autopilota s důrazem na přechodovou fázi mezi vertikálním a horizontálním letem.
6. Ověřte navrženou konfiguraci v software-in-the-loop a hardware-in-the-loop simulacích.

Seznam odborné literatury:

[1] Marques, P., Da Ronch, A.: Advanced UAV Aerodynamics, Flight Stability and Control: Novel Concepts, Theory and Applications, John Wiley & Sons, 2017.
[2] Gopalakrishnan, E. M.: Quadcoptor flight mechanics model and control algorithms, diplomová práce, ČVUT v Praze, 2016.
[3] Matušek, L.: Proposal for UAV with the concept of bicopter allowing vertical take-off and landing, bakalářská práce, ČVUT v Praze, 2016.

Vedoucí diplomové práce: Ing. Milan Rollo, Ph.D.

Platnost zadání: do konce zimního semestru 2018/2019

L.S.

prof. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 23. 5. 2017

Poděkování

Děkuji vedoucímu této práce Ing. Milanu Rollovi Ph.D. za věcné podněty a cenné rady. Děkuji také mojí rodině, přátelům a především mé přítelkyni za ochotné naslouchání a morální podporu.

Prohlášení

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

V Praze dne 9. ledna 2018

Podpis autora práce

Abstrakt

Práce se zabývá vytvořením matematického modelu bezpilotního VTOL prostředku pro účely odladění konfiguračních parametrů autopilotu. V rámci práce byla vytvořena simulace v prostředí Matlab & Simulink, umožňující integraci řídicích algoritmů již existujícího autopilotu ArduPilot v software-in-the-loop simulaci. Část práce se zároveň zabývá identifikací letových parametrů konkrétního letounu, ale celá simulace je navržena převážně parametricky, díky čemuž lze jednoduše přepracovat i pro popis zcela odlišného stroje.

Klíčová slova: VTOL, UAV, ArduPilot, Matlab, Simulink, Software-in-the-loop, simulace

Vedoucí práce: Ing. Milan Rollo Ph.D. Katedra počítačů

Abstract

This thesis aims to create a mathematic model of a VTOL unmanned aerial vehicle for the purpose of debugging the autopilot's configuration parameters. As part of the work, the Matlab & Simulink simulation was created, enabling the integration of the controlling algorithms of ArduPilot, an existent autopilot, in a software-in-the-loop simulation. The work also studies the identification of flight parameters of a specific airplane, but the whole simulation is predominantly designed parametrically, which allows easy remodelling for the purposes of the identification of an entirely different vehicle.

Keywords: VTOL, UAV, ArduPilot, Matlab, Simulink, Software-in-the-loop, simulation

Title translation: Control Design of Hybrid VTOL UAV

Obsah

1 Úvod **1**

2 Rešerše **3**

2.1 Srovnání jednotlivých konstrukcí UAV . . . 3
2.1.1 Multikoptéra . . . 3
2.1.2 Konvenční letoun . . . 5
2.1.3 Hybridní konstrukce . . . 6
2.2 Typy hybridních konstrukcí . . . 6
2.2.1 Tail-sitter . . . 6
2.2.2 Quad/Hexa/Octa-plane . . . 7
2.2.3 Tilt-rotor . . . 8
2.3 Software řízení . . . 8
2.3.1 PX4 Pro . . . 10
2.3.2 ArduPilot . . . 10
2.3.3 LibrePilot . . . 10
2.3.4 AirRails . . . 10
2.4 Řídicí jednotky . . . 11
2.4.1 Pixhawk . . . 11
2.4.2 Beagle Bone Blue . . . 12
2.4.3 APM . . . 12
2.4.4 NAVIO2 . . . 12
2.4.5 Intel Aero . . . 12
2.4.6 Qualcomm Snapdragon Flight Kit . . . 12
2.5 SITL a HIL simulace . . . 13
2.5.1 Matlab & Simulink . . . 14

3 Teoretická část **15**

3.1 Souřadné systémy . . . 15
3.1.1 Zeměpisná souřadná soustava 15
3.1.2 Letadlová souřadná soustava 16
3.1.3 Lokální zeměpisná souřadná soustava (NED) . . . 16
3.1.4 Aerodynamická souřadná soustava . . . 16
3.1.5 Transformace mezi souřadnými soustavami . . . 17
3.2 Mechanika letounu . . . 18
3.2.1 Tenzor setrvačnosti . . . 19
3.2.2 Gravitační síla . . . 20
3.2.3 Aerodynamické síly a momenty 20
3.2.4 Motorové síly a momenty . . . 21

4 Simulace **23**

4.1 Simulovaný prostředek . . . 23
4.1.1 Pohon a elektronika . . . 24
4.2 Model letové dynamiky . . . 24
4.2.1 Motorové síly a momenty . . . 24
4.2.2 Aerodynamické síly a momenty 27
4.2.3 Gravitační síla . . . 32
4.3 Identifikace parametrů simulovaného prostředku . . . 32
4.3.1 Aerodynamické součinitele . . 32
4.3.2 Moment setrvačnosti . . . 36
4.3.3 Parametry motorů a serv . . . 38
4.4 Vizualizace . . . 39
4.5 Propojení FDM s ArduPilot SITL 40
4.5.1 Formát dat . . . 40
4.5.2 Virtuální senzory . . . 42
4.5.3 Spojení . . . 42
4.5.4 Časování a obnovovací frekvence . . . 43
4.5.5 Hardware-in-the-loop simulace 43
4.5.6 Ovládání . . . 44
4.5.7 Konfigurace a ladění . . . 44
4.6 Systémové požadavky a postup zprovoznění . . . 45

5 Experimenty **47**

5.1 Ověření modelu letové dynamiky 47
5.2 Ověření ovládacích prvků a funkčnosti SITL . . . 47
5.3 Ladění parametrů řízení . . . 48

6 Závěr **49**

Reference **51**

A Význam zkratek **53**

B Obsah CD **55**

Obrázky

2.1 Teoretická závislost maximálního letového času na kapacitě pohonného akumulátoru UAV DJI Matrice 600 (dle dat dostupných z [8]) 4
2.2 Příklad konstrukce typu tail sitter se dvěma motory (převzato z [23]) . 7
2.3 Amazon Prime Air - Příklad konstrukce typu octa-plane (převzato z [2]) 8
2.4 E-Flite Convergence - Příklad konstrukce typu tilt-rotor Y3 (převzato z [10]) 9
2.5 BirdsEyeView FireFLY6 - Příklad konstrukce typu tilt-rotor Y6 (převzato z [12]) 9

3.1 Zeměpisné souřadné soustavy - převzato z [11] (upraveno) 15
3.2 Letadlová souřadná soustava . . . 16
3.3 Aerodynamická souřadná soustava 17

4.1 Základní schéma simulace 23
4.2 Schéma simulace modelu letové dynamiky 25
4.3 Simulace otáček motoru 26
4.4 Charakteristika koeficientu statického tahu a výkonu v závislosti na otáčkách vrtule 8,5x6 26
4.5 Charakteristika koeficientu dynamického tahu a výkonu v závislosti na otáčkách vrtule 8,5x6 27
4.6 Schéma systému pro simulaci dynamického tahu motoru 28
4.7 Blok výpočtu celkových sil a momentů pohonu 28
4.8 Blok výpočtu aerodynamických sil a momentů 29
4.9 Výpočet celkových aerodynamických součinitelů 30
4.10 Blok interpolace základních aerodynamických koeficientů 31
4.11 Systém simulace výchylky serva 31
4.12 Model letounu v prostředí XFLR5 32
4.13 Závislost součinitele vztlaku na úhlu náběhu 33
4.14 Závislost součinitele odporu na úhlu náběhu 34
4.15 Závislost součinitele klopivého momentu na úhlu náběhu 34
4.16 Závislost součinitelů na úhlu vybočení při úhlu náběhu 5° 35
4.17 Závislost součinitele stáčivého momentu C_n na úhlu vybočení při úhlu náběhu $\alpha = 0°$ 35
4.18 Závislost součinitele boční síly C_Y na úhlu vybočení při úhlu náběhu $\alpha = 0°$ 36
4.19 Ovlivnění součinitele klonivého momentu C_l při maximální výchylce křidélek v závislosti na úhlu náběhu α 37
4.20 Schéma měření momentu setrvačnosti 37
4.21 Odezva na rychlou změnu řídicích povelů motoru 38
4.22 Odezva na jednotkový skok řídicích povelů serva 39
4.23 Ukázka vizualizace 39
4.24 Blok vizualizace 40
4.25 Schéma propojení simulace s dalšími komponentami 41
4.26 Schéma zajišťující propojení s instancí emulátoru ArduPilot SITL 42

5.1 Zálet plně autonomní mise v prostředí GCS Mission Planner s využitím ArduPilot SITL 48

Kapitola 1

Úvod

Bezpilotní létající prostředek (angl. Unmanned Aerial Vehicle, zkráceně UAV), je letadlo bez posádky, většinou schopné nést užitečnou zátěž. Často se používají jako nosiče nejrůznějších typů kamer při filmování, mapování nebo revizích. Také jsou běžně využívány pro armádní účely. Jejich hlavní výhodou je, že nepotřebují vysoce kvalifikovanou obsluhu, protože jsou většinou řízeny především autopilotem s předem nastavenými instrukcemi. Díky tomu, že nepotřebují žádnou posádku, mohou být menší, případně dosahovat delšího doletu nebo disponovat větším prostorem pro užitečnou zátěž.

V posledních letech jsou UAV s elektrickým pohonem na velikém vzestupu. Vděčí za to především poklesu cen jednotlivých komponent, nízkým provozním nákladům, nízké hmotnosti a malým rozměrům. Díky tomu jsou často použitelné i v oblastech, kam by se konvenční prostředky s posádkou nedostaly. Nicméně jejich použití má také svá omezení. V případě multikoptéry je tímto omezením vysoká spotřeba a v případě konvenčního letounu zase nutnost vzletové dráhy, či katapultu a stálého dopředného pohybu ve vzduchu.

Nízká energetická hustota současných akumulátorů mnohdy elektrickým UAV, obzvlášť multikoptérám, nedovoluje dosáhnout potřebných letových časů. Energetická hustota lithium-polymerových článků je $100 - 265 Wh/kg$. Oproti tomu benzín má energetickou hustotu $12,8 kWh/kg$ [1]. Nízká energetická hustota vede k tomu, že navyšování kapacity pohonných akumulátorů rychle navyšuje hmotnost celého stroje a postupně přestává být další navyšování kapacity efektivní. Nejen kvůli tomu, ale například i z finančních důvodů může být vhodné zvýšení efektivity stroje místo navyšování kapacity pohonného akumulátoru.

Určitým řešením pro skloubení výhod a potlačení některých nevýhod konvenčních konstrukcí mohou být hybridní konstrukce s možností vertikálního startu i přistání a horizontálního letu ve vztlakovém režimu.

Pro tento typ letounů je však klíčová správnost nastavení parametrů regulačních smyček. Jsou zde totiž rovnou tři režimy, které je nutné správně nastavit a odladit, aby nedošlo k poškození samotného letounu, cizího majetku nebo dokonce zdraví. Vhodným prostředkem pro bezpečné odladění veškerého nastavení může být dostatečně věrohodná simulace, umožňující pozorování

[1] Efektivita spalovacích motorů (25 % - 50 %) je zhruba poloviční oproti elektromotorům (až 96%). Reálně tedy lze získat z 1 kg benzínu maximálně cca 6 kWh užitečné energie

chování letounu v různých režimech.

Tato práce se zabývá vytvořením obecného matematického modelu hybridního VTOL prostředku, jeho implementací v prostředí Matlab & Simulink a částečnou identifikací parametrů konkrétního letounu pro účely ověření vytvořeného modelu. Dále se práce zabývá vytvořením spojení mezi tímto modelem a již existujícím autopilotem ArduPilot, zastávajícím funkci řízení, pro ověření nastavení jeho řídicích parametrů.

V následující kapitole je provedeno porovnání hybridní VTOL koncepce s konvenčními letouny a srovnání různých konstrukcí VTOL prostředků. Je zde také provedena rešerše dostupného hardwaru a softwaru autopilotů, určeného pro řízení malých UAV, a dostupných simulátorů, umožňujících propojení s uvedenými autopiloty. V další kapitole je proveden teoretický rozbor zadané úlohy a jsou zde definovány základní vztahy a pojmy. Čtvrtá kapitola se zabývá implementací celého modelu, včetně identifikace parametrů navrhovaného letounu a vytváření rozhraní pro komunikaci autopilotem. V další kapitole jsou popsány experimenty a výsledky této práce. Poslední kapitola obsahuje závěrečné zhodnocení, přínosy a možnosti dalšího rozvoje.

Kapitola 2

Rešerše

2.1 Srovnání jednotlivých konstrukcí UAV

Běžně používané UAV prostředky lze rozdělit do tří skupin: multikoptéry, konvenční letouny s pevným křídlem a hybridní VTOL prostředky.

2.1.1 Multikoptéra

Nejčastěji používaným typem UAV jsou v současnosti multikoptéry. Tyto UAV nabízejí robustní konstrukci, díky které je většina strojů použitelná téměř za jakéhokoliv počasí, zároveň dokáží udržovat pozici bez nutnosti dopředného pohybu a vzlétat i přistávat vertikálně. Zásadním problémem všech strojů konstrukce multikoptéry je však značná neefektivita a s tím související krátký maximální letový čas. Při nasazení na delší mise je pak zapotřebí mít několik sad pohonných akumulátorů, což zvyšuje pořizovací náklady stroje a samotná nutnost výměny akumulátoru pak snižuje efektivní využitelnost letového času stroje.

Efektivita multikoptér je závislá především na použitých pohonných jednotkách a vrtulích. Aby letoun (multikoptéra nebo VTOL prostředek) v režimu visení byl letu-schopný, musí jeho pohon dosáhnout minimálně takového statického tahu, který se rovná velikosti na něj působící gravitační síly F_G. V praxi je však nutné, aby motory byly schopné dosáhnout tahu vyššího kvůli vyšším nárokům při stoupání, manévrování, či případnou užitečnou zátěž. Vhodný výkon pro bezpečný stabilizovaný let je 50-60 % maximálního výkonu, aby motory disponovaly dostatečnou výkonovou rezervou.

Statický tah motoru závisí na použité vrtuli, dodávaném výkonu a efektivitě celého ústrojí. Teoretická závislost tahu na dodané energii je dána vztahem [5]

$$P = \sqrt{\frac{F^3}{2\pi \cdot r^2 \cdot \rho_{air}}}, \tag{2.1}$$

kde P je potřebný výkon pro požadovaný tah F, při poloměru vrtule r a hustotě vzduchu ρ_{air}.

Úpravou získáme vztah pro výpočet statického tahu při známém dodávaném

výkonu:

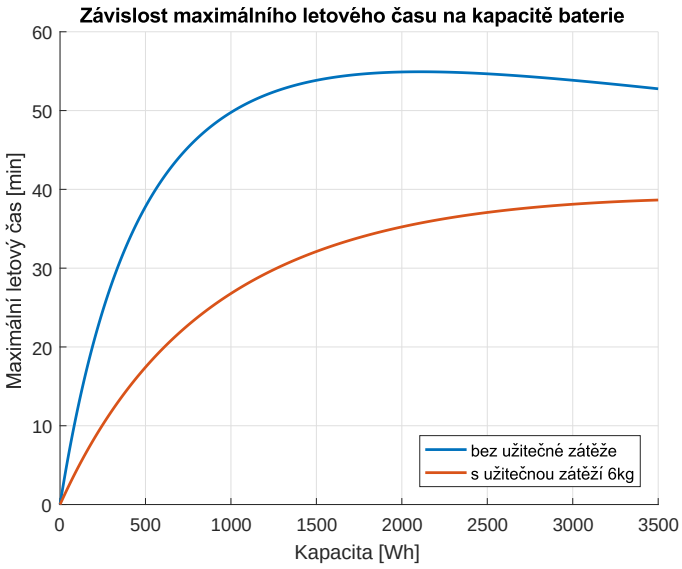
$$F = \sqrt[3]{P^2 \cdot 2\pi \cdot r^2 \cdot \rho_{air}}, \tag{2.2}$$

Například motor T-Motor U8 KV170 s vrtulí 28 x 9,2"[26] dosahuje efektivity až $14,9g/W$ při 50% výkonu (příkon $122W$). Teoretický maximální tah při příkonu $122W$ s vrtulí 28"je dle vztahu 2.2 $F \approx 24,3N$, což dává efektivitu tahu cca $20,3g/W$. Výsledná účinnost pohonu pak může být až 73%.

Maximální letový čas multikoptéry lze odhadnout ze vztahu

$$time_f = \frac{C_b}{M}\eta_m = \frac{m_b \cdot \rho_e}{M}\eta_m, \tag{2.3}$$

kde C_b je kapacita pohonných akumulátorů, M letová hmotnost stroje, m_b hmotnost pohonných akumulátorů, ρ_e energetická hustota použitých akumulátorů a η_m poměr tahu a spotřeby energie. Předpokládejme, že pohonné akumulátory strojů v této výkonnostní kategorii tvoří polovinu hmotnosti stroje bez užitečné zátěže. Dosazením výše uvedených parametrů ($\frac{m_b}{M} = \frac{1}{2}$, $\rho_e = 0.20Wh/g$, $\eta_m = 14,9g/W$) získáme teoretický maximální letový čas pro tento stroj cca 89 minut. Tento údaj je však nutno chápat jako horní hranici pro danou soustavu, protože motory nepracují stále v optimálních otáčkách, ale neustále přizpůsobují výkon s ohledem na stabilizaci a ovládání, což vede ke zvýšení spotřeby. Navíc jsou z akumulátorů napájena další zařízení (řídicí jednotka, přijímač, vysílač telemetrie a jiné).



Obrázek 2.1: Teoretická závislost maximálního letového času na kapacitě pohonného akumulátoru UAV DJI Matrice 600 (dle dat dostupných z [8])

Navyšování kapacity pohonných akumulátorů vede k rychlému nárůstu letové hmotnosti kvůli jejich nízké energetické hustotě. Tím pádem je mimo jiné potřeba vyvinout vyšší výkon pro udržení takového stroje v letu, což má negativní dopad na letový čas. Proto navyšování kapacity pohonných akumulátorů je efektivní pouze do určitého bodu. Na grafu 2.1 je uveden

příklad teoretické závislosti kapacity pohonné baterie na letovém času. V tomto příkladu bylo uvažováno pouze s proměnnou hmotností akumulátoru v závislosti na jeho kapacitě a bylo zanedbáno zvyšování hmotnosti šasi a motorů v důsledku vyššího potřebného výkonů.

V tabulce 2.1 jsou uvedeny některé další motory. Nejefektivnější motory dosahují účinnosti téměř 74 %, ale ani tato hodnota není pro některé aplikace dostačující.

Název motoru / vrtule	příkon [W]	skutečný tah [g]	teoretický tah [g]	účinnost
DYS SE2205 / 5x4.5	176	600	1000	60,0%
Emax RS2205s / 5x4	240	730	1230	59,3%
T-motor U5 / 16x5.4	115	1200	1640	73,2%
T-motor U8 / 28x9.2	122	1820	2471	73,6%
T-motor U10 / 30x10.5	183	2162	3391	63,7%

Tabulka 2.1: Výkon a efektivita různých motorů

2.1.2 Konvenční letoun

Naproti tomu UAV klasické konstrukce s pevným křídlem (Fixed Wing) mohou nabídnout vysokou efektivitu a tím pádem i dlouhé letové časy. Jejich nasazení je však poměrně omezené, zejména kvůli nutnosti vzletové a přistávací dráhy a velké závislosti na povětrnostních podmínkách. Kromě toho existují také legislativní omezení, která nedovolují létání bez přímého vizuálního dohledu operátora. Vzhledem k horší manévrovatelnosti ve srovnání s multikoptérami tato skutečnost může značně zmenšit efektivně využitelný prostor mise.

Při stabilním letu je nutné, aby se vztlaková síla F_L vyrovnala tíhové síle působící na letoun F_G. Vztlaková síla je závislá na tvaru a velikosti křídla, jeho profilu, úhlu náběhu α a vzdušné rychlosti v. Musí tedy platit rovnice

$$F_L = F_G = M \cdot g = \frac{1}{2} \cdot \rho \cdot v^2 \cdot S \cdot C_L, \tag{2.4}$$

kde ρ je hustota vzduchu, S plocha křídla a C_L součinitel vztlaku.

Dále je při stabilním letu nutné, aby se velikost tahu F_T rovnala velikosti odporových sil F_D. Tyto síly lze při stabilizovaném letu vypočítat vzorcem

$$F_T = F_D = \frac{1}{2} \cdot \rho \cdot v^2 \cdot S \cdot C_D, \tag{2.5}$$

kde C_D je součinitel odporu.

Z výše uvedených rovnic vyplývá, že potřebný tah motorů je možné vypočítat ze vztahu

$$F_T = \frac{M \cdot g}{\frac{C_L}{C_D}}, \tag{2.6}$$

přičemž poměr mezi součinitelem vztlaku a odporu $\frac{C_L}{C_D}$ je nazýván klouzavost letounu a lze jej chápat jako vzdálenost, do které je letoun schopen doklouzat z jednotkové výšky bez dodání další energie.

Chceme-li určit potřebný výkon motoru při stabilním letu, musíme znát dynamický tah pohonné jednotky s konkrétní vrtulí při dané rychlosti. Maximální klouzavost se sice nemění s hmotností letounu, ale pro dosažení stejné klouzavosti s více zatíženým letounem je potřeba vyvinout vyšší rychlost, což klade vyšší nároky na pohonnou jednotku, protože dynamický tah při rostoucí rychlosti klesá a pro dosažení stejného tahu je tedy potřeba vyvinout větší rychlost otáčení vrtule.

Letový čas lze odhadnout pomocí vztahu

$$time_f = \frac{C_b}{T_d}\eta_{md} = \frac{C_b}{M \cdot g}\frac{C_l}{C_d}\eta_{md}, \tag{2.7}$$

kde η_{md} je poměr mezi dynamickým tahem a spotřebou energie pohonné jednotky. Pohonná jednotka by měla požadovaného tahu opět dosáhnout s dostatečnou rezervou, neboť při stoupání jsou na ni kladeny vyšší nároky. Nicméně je běžné, že podíl tahu pohonu a hmotnosti letounu je menší, než 1, na rozdíl od multikoptéry a tedy i celkový výkon pohonu může být menší. Navíc není pohon využíván pro stabilizaci, a tudíž si konvenční letoun může vystačit i s jediným motorem. Výše uvedené skutečnosti vedou k tomu, že je hmotnost pohonu nižší, než v případě multikoptéry, což vede k dalšímu zvýšení efektivity prostředku.

Při porovnání s rovnicí 2.3 je patrné, že letový čas je pozitivně ovlivněn koeficientem klouzavosti C_l/C_d, který obvykle nabývá hodnot v řádu desítek. Negativní dopad má naopak nahrazení efektivity statického tahu η_m za efektivitu dynamického tahu η_{md}, jejíž hodnota je běžně nižší.

2.1.3 Hybridní konstrukce

Alternativou mohou být hybridní letouny VTOL, kombinující možnost verikálního startu a přistání s vysokou efektivitou klasické konstrukce s pevným křídlem v klouzavém letu. Jejich výhodou je i možnost zastavení v letu, čehož může být také při některých typech misí využíváno.

Hlavní nevýhodou této konstrukce je vyšší hmotnost kvůli požadavku na vyšší statický tah motorů, který v součtu musí s dostatečnou rezervou převyšovat hmotnost letounu, a také kvůli přítomnosti překlápěcího mechanismu (u některých typů konstrukcí však být nemusí). Hybridní letouny tedy většinou dosahují horší efektivity oproti letounům klasické konstrukce. Proti multikoptérám mají zase horší manévrovatelnost, větší rozměry a vyšší hmotnost při stejné užitečné zatížitelnosti. Nicméně VTOL prostředky v misích, při nichž lze dostatečně využít vodorovného letu ve vztlakovém režimu, mohou dosáhnout delších letových časů, než multikoptéry.

2.2 Typy hybridních konstrukcí

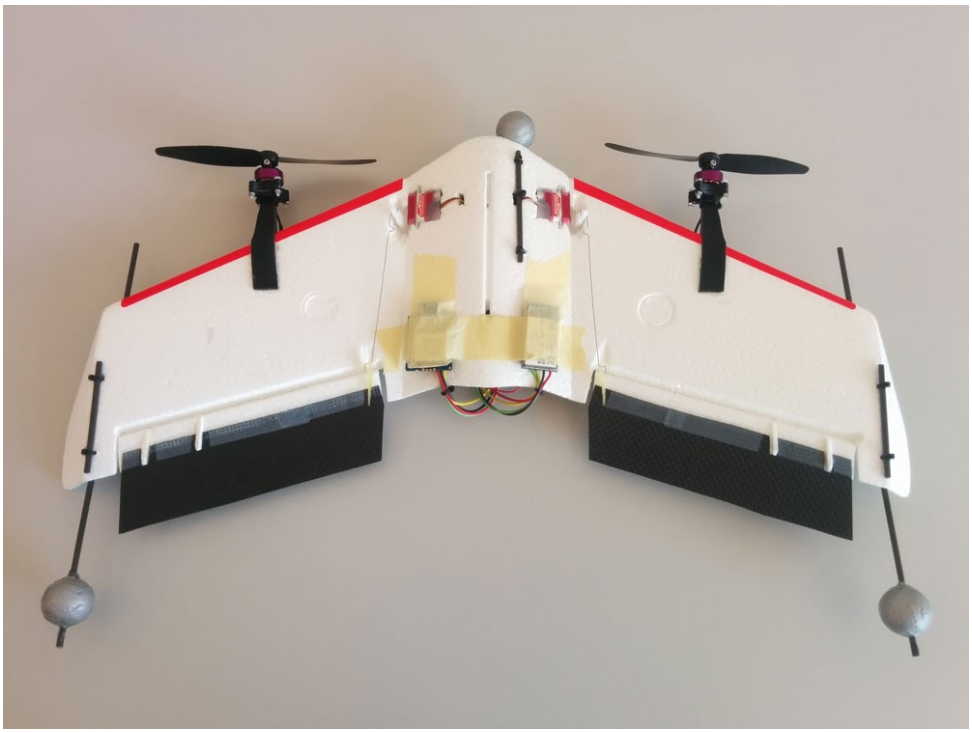
2.2.1 Tail-sitter

Jedná se o technicky nejjednodušší typ konstrukce, kdy letoun může startovat ze zádě (tail) i na ni přistávat. Ve vertikálním režimu je letoun stabilizován tahy

motorů (klonění) i křidélky (klopení a bočení). Přechod do klouzavého letu probíhá klopením, tedy otáčením celého letounu okolo osy Y. Při dopředném letu se pak mění ovládací prvky tak, že křidélka již neslouží pro bočení, nýbrž pro klonění a pro bočení lze využít rozdílný tah motorů. Plocha křidélek musí být dostatečně velká, aby bylo možné manévrovat i při nulové rychlosti, kdy je křídlo ofukováno pouze proudem vzduchu od vrtulí. Ovládání křidélek musí pak být přesné a bez vůlí, protože při vyšších rychlostech je ovládání velice citlivé na změnu výchylky.

Hlavními nevýhodami této konstrukce je horší manévrovatelnost ve vertikálním režimu a překlápění celého letounu při přechodu mezi režimy, což má za následek omezené možnosti aplikace. Výhodou je jednoduchost konstrukce.

Příklad tail-sitteru je na obrázku 2.2.



Obrázek 2.2: Příklad konstrukce typu tail sitter se dvěma motory (převzato z [23])

2.2.2 Quad/Hexa/Octa-plane

Tato konstrukce využívá jinou sadu motorů pro zdvih a jinou pro dopředný let (například 4 a 1). Při vzletové fázi jsou tedy vypnuté motory pro dopředný let a po přechodu zase motory pro zdvih. Pouze při přechodových fázích jsou využívány obě sady motorů. Díky tomu však přechodové fáze probíhají velmi hladce. Nevýhoda spočívá především ve vysoké hmotnosti letounu z důvodu velkého počtu motorů, z nichž je většinu času část z nich nevyužívaná.

Příklad konstrukce typu Quad-plane je na obrázku 2.3.

Obrázek 2.3: Amazon Prime Air - Příklad konstrukce typu octa-plane (převzato z [2])

2.2.3 Tilt-rotor

Tento typ konstrukce využívá alespoň část motorů pro zdvih i dopředný let. Mezi těmito režimy jsou motory otočeny a tím je přizpůsobeno směřování tahu - dolů v případě režimu visení, či dopředu při vodorovném letu. Výhodou této konstrukce je, že nedochází k otáčení trupu jako u tail-sitteru a zároveň není potřeba nadbytečných motorů. Oproti tomu musí prostředek tohoto typu disponovat dostatečně robustním otočným mechanismem motorů.

U tohoto typu konstrukce je možný větší počet konfigurací. Konfigurace Y3 (na obrázku 2.4) používá minimálně dva nezávislé sklopné mechanismy, které aktivně kompenzují momenty vrtulí. Bez to by nebylo možné docílit boční stability v režimu visení. Proti tomu například Y6 (na obrázku 2.5) může mít pouze jeden sklopný mechanismus, protože využívá k dosažení boční stability protiběžné vrtule. Podobně i u konfigurace se čtyřmi, šesti, případně osmi motory postačuje jeden sklopný mechanismus.

2.3 Software řízení

Software řízení je soubor algoritmů běžících na palubní výpočetní jednotce, starajících se především o stabilizaci letounu a jeho řízení. To může probíhat na základě povelů operátora nebo podle předdefinované mise, pokud software řízení obsahuje i autopilot. K úplnému ovládání a konfigurování softwaru autopilota je většinou potřeba software pozemní stanice (GCS).

Obrázek 2.4: E-Flite Convergence - Příklad konstrukce typu tilt-rotor Y3 (převzato z [10])

Obrázek 2.5: BirdsEyeView FireFLY6 - Příklad konstrukce typu tilt-rotor Y6 (převzato z [12])

2.3.1 PX4 Pro

PX4 Pro je open-source autopilot orientovaný především na modely letadel a multikoptér, ale je použitelný i jako řídicí jednotka pro modely aut, lodí a dalších prostředků. Projekt byl zahájen v roce 2009 na Spolkové vysoké technické škole v Curychu a společně s firmware autopilotu byla vyvíjena i řídicí jednotka PX4FMU a kontrolér PX4IO. Z těchto jednotek následně pod záštitou společností 3DRobotics a DIYDrones vznikla jednotka Pixhawk.

PX4 Pro podporuje všechny druhy multikoptér a klasických letadel i všechny výše uvedené typy hybridních konstrukcí v základním nastavení. Protože tento autopilot ke komunikaci s pozemní stanicí využívá protokol MAVLink (Micro Air Vehicle Link), lze pro pozemní stanici použít jakýkoliv software, podporující tento protokol. V případě PX4 Pro se však doporučuje používat software QGroundControl.

Autopilot PX4 Pro je vyvíjen s licencí BSD 3-Clause, která umožňuje jeho modifikaci a využití pro komerční účely bez nutnosti zveřejnění zdrojových kódů [21].

2.3.2 ArduPilot

ArduPilot je open-source projekt autopilotu, umožňující ovládat multikoptéry a helikoptéry, letadla s klasickou konstrukcí a VTOL mechanismy, ale i lodě, ponorky, vozidla, či sledovače antény. Tento autopilot je úzce spjatý s autopilotem PX4 Pro, ale jeho komunita je podstatně širší a tudíž lze předpokládat větší míru odladěnosti software [3].

Nabízí široké možnosti software-in-the-loop simulací, ale hardware-in-the-loop simulace jsou v současnosti podporovány již pouze v omezené míře.

Stejně jako PX4 Pro i ArduPilot ke komunikaci s pozemní stanicí využívá protokol MAVLink. U tohoto autopilotu je doporučováno používání softwaru GCS MissionPlanner nebo APM Planner.

2.3.3 LibrePilot

Jedná se o open-source projekt, který vzešel z projektu OpenPilot. Jeho podpora hardware je však velice omezená, neboť je určen výhradně pro speciální jednotky OpenPilot. Navíc podpora VTOL mechanismů u tohoto autopilotu je pouze experimentální [19].

2.3.4 AirRails

AirRails je closed-source projekt, speciálně zaměřený na VTOL prostředky (podporuje všechny výše uvedené), ale umožňuje i řízení klasických konstrukcí (multikoptéra, či letadlo). Je použitelný na jednotkách Pixhawk. Oproti firmware PX4 Pro obsahuje drobná vylepšení.

Spolu s firmware autopilotu je nabízen i software pro Android tablet jako pozemní stanici. Tento software umožňuje uživetelsky přívětivé plánování

misí s ohledem na VTOL prostředky, 3D vizualizaci letu, přehledné zobrazení letových údajů a další užitečné funkce [1].

2.4 Řídicí jednotky

Řídicí jednotka reprezentuje platformu, na níž běží software řízení. Kromě výpočetní jednotky se vstupními a výstupními rozhraními většinou obsahuje sadu senzorů (běžně tříosý akcelerometr, tříosý gyroskopický senzor, tříosý kompas, barometr) a přípravu pro připojení dalších (především globální družicový polohový systém). Některé jednotky mohou kromě běžných senzorů disponovat rozhraními pro další senzory (lidar, kamera s rozpoznáním obrazu, sonar a dalšími).

2.4.1 Pixhawk

Jedná se o open-hardware jednotku, vzniklou sloučením řídicí jednotky PX4FMU a desky PX4IO do jediné desky. Díky tomu je jednotka více kompaktní a navíc disponuje dalšími vylepšeními v podobě většího počtu portů, či větší pamětí. Tato jednotka byla navržena pro fungování s firmware autopilotu PX4 Pro, ale podporuje ji i firmware ArduPilot, či AirRails [21].

Specifikace

- Procesor 32-bit ARM Cortex M4 core
- Paměť: 168 Mhz/256 KB RAM/2 MB Flash
- Senzory: 2x gyroskopický, 2x akcelerometr, barometr, magnetometr (kompas)
- 5 sériových portů (UART), S-bus/ppm vstup, I2C, SPI, 2x CAN, USB
- 8 PWM výstupů pro elektronické regulátory otáček (ESC), 6 pro přídavná zařízení (např. spouštěč kamery, ovládání podvozku, či přepínání režimů letu u VTOL mechanismů)

Pixhawk 2

Jedná se o modernizaci původní jednotky. Oproti svému předchůdci jednotka disponuje trojitou redundancí pro IMU (Inertial Meassurement Unit - jednotka zodpovědná za měření působících sil, obsahující akcelerometr, gyroskopický senzor a magnetometr), dvěma porty sběrnice I2C a CAN, pěti UART porty a S-Bus výstupem pro serva.

Pixhawk 3 pro

Tato jednotka obsahuje výkonnější procesor oproti první a druhé verzi. Naproti tomu nedisponuje trojitou redundancí senzorů, nýbrž pouze dvojitou.

2.4.2 Beagle Bone Blue

Beagle Bone Blue je open-hardware výkonný linuxový počítač s procesorem ARM Cortex A8 1 GHz, 512 MB RAM a 4 GB eMMC, navržený pro real-time aplikace. Přímo na desce je obsažena jednotka IMU, která však nemá žádnou redundanci a pouze 8 PWM výstupů. Dále disponuje pěti UART konektory, čtyřmi GPIO (General-Purpose Input Output) a výstupy sběrnice I2C i CAN. Tato jednotka podporuje pouze firmware ArduPilot [6].

2.4.3 APM

ArduPilotMega, je open-hardware řídicí jednotka, jejímž základem je vývojová deska Arduino Mega. Jak již název napovídá, jednotka je určena především pro firmware ArduPilot. Mezi moderními jednotkami začal však být výkon APM nedostatečný a proto byl vývoj této jednotky ukončen.

2.4.4 NAVIO2

Jedná se o closed-hardware rozšiřující desku k mikropočítači Raspberry PI (verze 2 nebo 3), který rozšiřuje o sadu vstupně-výstupních portů a senzory. Mezi ty patří GNSS přijímač (GPS, Glonass, Beidou, Galileo) a IMU s dvojitou redundancí. Součástí je i barometr s vysokou přesností (rozlišení až 10 cm). Jednotka obsahuje 14 PWM výstupních kanálů pro motory nebo serva, A/D převodník, I2C sběrnici, a UART port. Napájení jednotky může disponovat až trojitou redundancí [20].

2.4.5 Intel Aero

Jedná se o closed-hardware výkonný počítač se čtyřjádrovým procesorem Intel Atom, 4 GB RAM, 32 GB eMMC a s Linuxovým operačním systémem Yocto. Deska obsahuje dvě rozhraní I2C, tři UART, SPI, CAN, 25 GPIO pinů a 5 analogových vstupů. Vysoký výpočetní výkon je možné využít například při zpracování obrazu (například při využití Intel Aero Vision) nebo dat z lidaru [16].

2.4.6 Qualcomm Snapdragon Flight Kit

Jedná se o pokročilý closed-hardware linuxový počítač, obsahující čtyřjádrový procesor Qualcomm Snapdragon 801, pracující na frekvenci 2,26 GHz, grafický procesor Qualcomm Adreno, 2 GB RAM, 32 GB eMMC. Navíc obsahuje dedikovaný kontrolér pro real-time operace. Výkon tohoto počítače dostačuje pro téměř jakékoliv aplikace. Jeho použití však není triviální, například i z důvodu absence PWM výstupů pro regulátory otáček motorů. Mezi autopiloty, podporující tuto desku patří PX4 Pro i ArduPilot [22].

2.5 SITL a HIL simulace

SITL (případně SIL - Software-In-The-Loop) simulace využívá základní část firmware autopilotu, určeného pro řízení simulovaného prostředku, bez nutnosti připojování jakéhokoliv dodatečného hardwaru. Řídicí jednotka, pro kterou je firmware autopilotu určen, je tudíž pouze emulována a jeho senzorické vybavení je simulováno na základě modelu letové dynamiky letounu (FDM - Flight Dynamics Model).

HIL (Hardware-In-the-Loop) simulace využívá pro výpočetní úkony spojené s řízením simulovaného letounu nejen software řízení, ale i výpočetní hardware, určený pro skutečný letoun. Tento postup slouží k ověření hardwarových prostředků. Pro úplné ověření by pro komunikaci mezi jednotkou a simulátorem mělo být využíváno stejných rozhraní a sběrnic, kterých je využíváno pro komunikaci mezi jednotkou aktuátory a senzory při skutečném nasazení. HIL simulace v tomto rozsahu již však dnes postupně přestává být podporována.

V současné době se pro software/hardware-in-the-loop simulace s firmware autopilotu PX4 Pro nebo ArduPilot používají především níže uvedené simulátory.

Výchozí simulace pro ArduPilot

Jedná se o FDM multikoptér, který je implementován přímo v komponentě ArduPilot SITL. Celá software-in-the-loop simulace může běžet v rámci jedné aplikace (nepočítáme-li GCS). Model letounu (s firmware AdruPlane) v této simulaci podporován není. Hlavní předností této simulace je jednoduchost zprovoznění, neboť není přímo vyžadován žádný další software. Nevýhodou pak velká omezenost využití a úprav.

JSBSim

JSBSim [17] je open-source knihovna, umožňující vytvoření a simulaci dynamického modelu letounu. Je například využívána leteckým simulátorem FlightGear [13] jako jeden z výchozích modelů letové dynamiky. Umožňuje vytvoření vlastního modelu pouze definicí jeho parametrů ve formě konfiguračního XML souboru. Vytvořený model letové dynamiky umožňuje vytvoření software-in-the-loop a hardware-in-the-loop simulaci s firmware ArduPilot. Pro tuto komunikaci je využíváno protokolu UDP pro přenos paketů obsahujících informace o dynamice letounu, respektive řídicí povely. Všechny pakety jsou strukturovány dle FGFDM (FlightGear Flight Dynamics Model). JSBSim je implementována v jazyce C++.

JMavSim

JMavSim [4] je open-source knihovna napsaná v programovacím jazyce Java, umožňující vytvoření modelu letové dynamiky pro software-in-the-loop a hardware-in-the-loop simulaci s firmware autopilotu PX4 Pro. Pro komunikaci je využíván protokol UDP a pakety obsahují zprávy dle standardu MAVLink.

DroneKit-SITL

DroneKit - SITL [9] je FDM, vytvořený pro tvorbu software-in-the-loop simulací s autopilotem ArduPilot. Je implementován v programovacím jazyce Python a poskytuje aplikační rozhraní ve formě funkcí, jejichž voláním lze vytvářet různé příkazy, získávat informace o průběhu simulace, či upravovat parametry autopilotu.

Gazebo

Gazebo [14] je soubor open-source softwarových nástrojů pro prostředí ROS (Robot Operating System), určených pro návrh a realistické simulace nejrůznějších typů zařízení z oblasti robotiky. Tento simulátor umožňuje vytvoření FDM a dovoluje propojit jej s firmware ArduPilot i PX4 Pro pomocí protokolu UDP. Přenášená data jsou ve formě struktury dle standardů jazyka C/C++, obsahující předávaná data o FDM, respektive akční zásahy řízení. Tento simulátor je vhodný nejen pro SITL, ale i HIL simulace, umožňuje konfiguraci nejrůznějších parametrů a jako jediný z výše uvedených dokáže simulovat dynamiku VTOL prostředku.

2.5.1 Matlab & Simulink

Rešerše ukázala, že žádný z výše uvedených autopilotů přímo nepodporuje software-in-the-loop ani hardware-in-the-loop simulaci v součinnosti s prostředím Matlab & Simulink. Nebyla nalezena ani žádná běžně dostupná knihovna podporující tuto funkcionalitu.

Kapitola 3

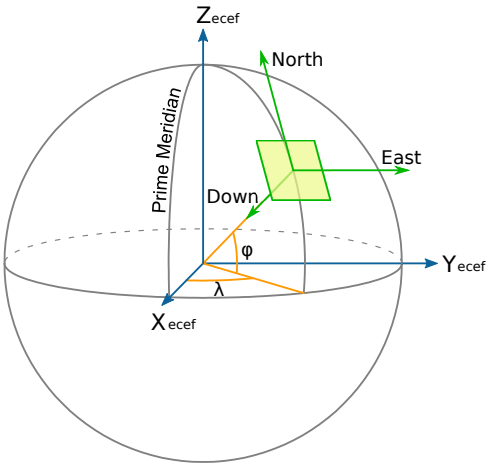
Teoretická část

3.1 Souřadné systémy

Pro vytvoření matematického modelu letounu je vhodné využít následujících souřadných soustav: zeměpisné globální, zeměpisné lokální, letadlové a aerodynamické.

3.1.1 Zeměpisná souřadná soustava

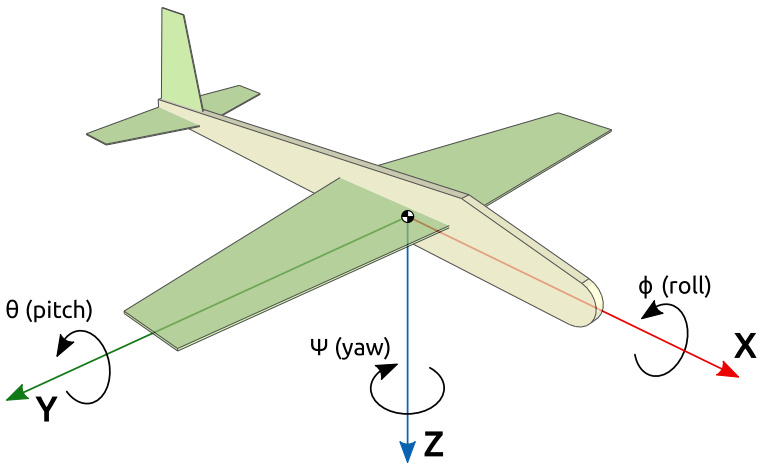
Zeměpisná souřadná soustava je základní soustava pro určení geografické polohy. Tato soustava se podobá soustavě sférické. Její počátek leží ve středu Země. Poloha bodu kdekoliv na zemském povrchu v této soustavě je určena dvojicí úhlů - zeměpisnou délkou a šířkou. Zeměpisná délka (λ, angl. longtitude) je úhel, který svírá nultý poledník a meridián (místní poledník), který hledaným bodem prochází. Zeměpisná šířka (φ, angl. latitude) pak určuje polohu na tomto meridiánu, přičemž na rovníku nabývá hodnoty 0, na Severním pólu 90° a na pólu Jižním -90°. Princip zeměpisné souřadné soustavy je patrný z obrázku 3.1.



Obrázek 3.1: Zeměpisné souřadné soustavy - převzato z [11] (upraveno)

3.1.2 Letadlová souřadná soustava

Letadlová souřadná soustava je pevně svázána s konstrukcí letounu. Jedná se o kartézskou soustavu se středem v těžišti letounu, s osou X směřující podél osy trupu ke špičce letounu, většinou ve směru tětivy profilu hlavního křídla. Dále s osou Y směřující doprava (vzhledem k předpokládanému směru letu) a osou Z dolů pod letoun. Se stoupající výškou tak hodnota Z roste. Letadlová souřadná soustava je znázorněna na obrázku 3.2.



Obrázek 3.2: Letadlová souřadná soustava

Orientace letounu vůči zemi je určena trojicí úhlů (Kardanovy úhly): příčný náklon (ϕ, angl. roll) okolo osy X, podélný sklon (θ, angl. pitch) okolo osy Y, a kurz (ψ, angl. yaw) okolo osy Z. Úhlové rychlosti odpovídající změnám těchto úhlů se nazývají klonění, klopení a bočení.

3.1.3 Lokální zeměpisná souřadná soustava (NED)

Pro zjednodušení transformace mezi souřadnými soustavami, je v této práci pro určení polohy v blízkosti referenčního bodu, používána souřadná soustava North-East-Down. Jedná se o kartézskou soustavu vyplývající z letadlové souřadné soustavy a její střed leží ve zvoleném referenčním bodě. První složka směřuje z referenčního bodu na sever, druhá na východ a třetí do středu Země. Princip lokální zeměpisné souřadné soustavy je patrný z obrázku 3.1.

3.1.4 Aerodynamická souřadná soustava

Jedná se o kartézskou souřadnou soustavu s počátkem v těžišti, která ale jinak není pevně svázaná s konstrukcí letounu. Osa X je totožná s vektorem vzdušné rychlosti. Aerodynamická souřadná soustava (viz obrázek 3.3) je vhodná pro popis sil působících na letoun. Ve směru osy X působí síla odporová, ve směru osy Y boční síla a ve směru osy Z síla vztlaková. Všechny aerodynamické síly mají opačnou orientaci oproti jejich příslušným osám.

Vzdušná rychlost

Vektor vzdušné rychlosti $V_A = \begin{bmatrix} u & v & w \end{bmatrix}^T$ určuje rychlost a směr pohybu letounu vzhledem k okolní vzdušné mase. Složky tohoto vektoru vyjadřují rychlosti v osách letadlové souřadné soustavy.

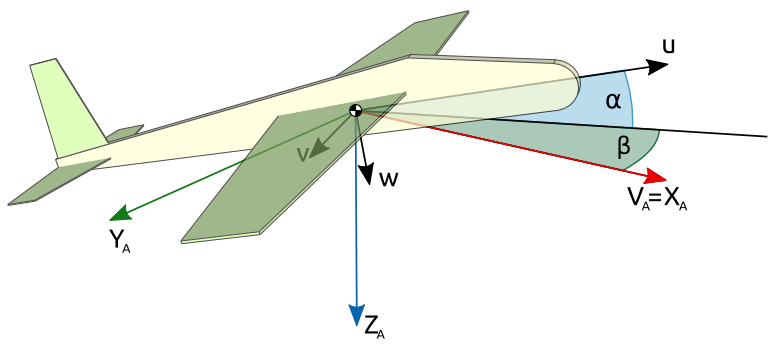
Úhly náběhu a vybočení

S toutou souřadnou soustavou jsou svázány tyto dva důležité úhly. Úhel náběhu α určuje úhel, pod kterým je v rovině XZ letadlové soustavy ofukována zvolená referenční přímka letounu (nejčastěji tětiva profilu hlavního křídla nebo osa X letadlové souřadné soustavy). Tento úhel lze z vektoru vzdušné rychlosti vypočítat jako

$$\alpha = \tan^{-1}\left(\frac{w}{u}\right). \tag{3.1}$$

Úhel vybočení β je úhel, pod kterým je v rovině XY ofukována zvolená referenční přímka letounu (nejčastěji osa trupu nebo osa X letadlové souřadné soustavy). Z vektoru vzdušné rychlosti jej lze vypočítat jako

$$\beta = \sin^{-1}\left(\frac{v}{|V_A|}\right) = \tan^{-1}\left(\frac{v}{\sqrt{u^2+w^2}}\right). \tag{3.2}$$



Obrázek 3.3: Aerodynamická souřadná soustava

3.1.5 Transformace mezi souřadnými soustavami

Pro transformaci mezi kartézskými souřadnými soustavami jsou využívány transformační matice rotace. Těmito maticemi lze zleva násobit vektor v počáteční soustavě, čímž dostaneme vektor v nové souřadné soustavě definované danou transformací. Transformační matice mají následující podobu

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix},$$

$$R_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{3.3}$$

$$R_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

kde θ je úhel otočení okolo dané osy v pravotočivém smyslu. Pro dosažení vícenásobné transformace lze provést zřetězení přinásobením další matice zleva.

Lokální zeměpisná soustava - Letadlová soustava

Pro transformaci orientace z lokální zeměpisné do letadlové soustavy je třeba dodržet pořadí jednotlivých rotací z, y', x" (yaw, pitch, roll). Výslednou transformační matici získáme ze vztahu

$$DCM_{wb} = R_x(\phi)\, R_y(\theta)\, R_z(\psi). \tag{3.4}$$

Tato transformace je využívána například při výpočtu gravitační síly, působící na letoun.

Aerodynamická soustava - Letadlová soustava

Pro transformaci z aerodynamické souřadné soustavy do soustavy letadlové je nutné dodržet pořadí rotace z (úhel β), y' (úhel α).

$$DCM_{ab} = R_y(\alpha)\, R_z(\beta). \tag{3.5}$$

Tato transformace je potřeba například při výpočtu aerodynamických sil působících na letoun. Orientace sil v letadlové soustavě je však přesně opačná a proto je nutné nakonec výsledný vektor sil vynásobit -1.

3.2 Mechanika letounu

Pohybové rovnice letounu vycházejí ze základních principů Newtonovy mechaniky.

$$F = m\, a_b + m\left[\Omega_b \times v_b\right], \tag{3.6}$$

kde F je vektor síly působící na letoun, m hmotnost letounu, a_b vektor zrychlení, v_b vektor lineárních rychlostí a Ω_b vektor úhlových rychlostí.

Momenty jsou definovány jako

$$M = I\,\varepsilon_b + [\Omega_b \times I\,\Omega_b]\,, \tag{3.7}$$

kde I je tenzor setrvačnosti a ε_b úhlové zrychlení.

3.2.1 Tenzor setrvačnosti

Tenzor setrvačnosti definuje momenty setrvačnosti v jednotlivých osách a deviační momenty. Jeho tvar je

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}. \tag{3.8}$$

Na hlavní diagonále leží momenty setrvačnosti, definované jako

$$I = \begin{bmatrix} I_{xx} \\ I_{yy} \\ I_{zz} \end{bmatrix} = \int_M r^2 dm$$

pro spojité rozložení hmoty v tělese, kde je integrace prováděna přes celou hmotnost tělesa M a r značí polohový vektor hmotnostního příspěvku dm, případně

$$I = \begin{bmatrix} I_{xx} \\ I_{yy} \\ I_{zz} \end{bmatrix} = \sum_i r_i^2 m_i$$

pro diskrétní rozložení hmoty, kde m_i je i-tý hmotný bod tělesa a r_i jeho polohový vektor. Zbylé hodnoty se nazývají deviační momenty a lze je vypočítat jako

$$I_{xy} = \int_M x\,y\,dm \quad I_{xz} = \int_M x\,z\,dm \quad I_{yz} = \int_M y\,z\,dm$$

pro spojité rozložení hmoty, respektive

$$I_{xy} = \sum_i x_i\,y_i\,m_i \quad I_{xz} = \sum_i x_i\,z_i\,m_i \quad I_{yz} = \sum_i y_i\,z_i\,m_i$$

pro rozložení diskrétní. Jelikož jsou letouny většinou symetrické podle roviny xz, příspěvky deviačních momentů obsahující složku y se vzájemně vyruší a deviační momenty $I_{xy}, I_{yz}, I_{yx}, I_{zy}$ jsou tedy nulové. Z deviačních momentů se uplatní pouze $I_{xz} = I_{zx}$. Proto platí

$$I = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{zx} & 0 & I_{zz} \end{bmatrix}. \tag{3.9}$$

3.2.2 Gravitační síla

Gravitační síla F_G působí do středu země, tedy pouze ve směru osy z souřadné soustavy north-east-down. Její hodnota závisí na hmotnosti letounu m a gravitačním zrychlení $g \approx 9.81\, ms^{-2}$ jako

$$F_G = m\, g.$$

Pro výpočet jejích účinků na letoun je pak nutné transformovat vektor $\begin{bmatrix} 0 & 0 & F_G \end{bmatrix}^T$ pomocí matice DCM_{wb} ze vztahu 3.4 jako

$$F_{Gb} = DCM_{wb} \begin{bmatrix} 0 \\ 0 \\ F_G \end{bmatrix}. \tag{3.10}$$

3.2.3 Aerodynamické síly a momenty

Jedná se o dynamické síly a momenty projevující se pouze při nenulové vzdušné rychlosti a tím pádem i nenulovém dynamickém tlaku $\bar{q}$, což je základní veličina určující velikost aerodynamických sil a momentů. Dynamický tlak lze vypočítat ze vztahu

$$\bar{q} = \frac{1}{2}\rho v_A^2, \tag{3.11}$$

kde ρ je hustota vzduchu a v_A velikost vzdušné rychlosti.

Dalšími základními parametry, které určují velikost aerodynamických sil a momentů jsou aerodynamické součinitele - bezrozměrné veličiny, které charakterizují aerodynamické vlastnosti letounu v určité pozici. Hodnoty těchto součinitelů je nutné buď vypočítat pro konkrétní letoun (například s využitím software XFLR5) nebo experimentálně naměřit v aerodynamickém tunelu.

Síly

Jak již bylo uvedeno výše, pro popis aerodynamických sil je vhodné využít aerodynamickou souřadnou soustavu, neboť velikost těchto sil je silně závislá na úhlu náběhu a úhlu vybočení.

Mezi aerodynamické síly se řadí tyto:

- odporová (F_D) - působí ve směru proudění vzduchu obtékajícího letoun
- boční (F_Y) - působí kolmo do stran vzhledem k proudění vzduchu
- vztlaková (F_L) - působí kolmo vzhůru nebo dolů vzhledem k proudění vzduchu

Ty lze pro nízké rychlosti « 1 Mach vypočítat podle vztahů

$$F_D = \bar{q}\, S\, C_D \quad F_Y = \bar{q}\, S\, C_Y \quad F_L = \bar{q}\, S\, C_L, \tag{3.12}$$

kde S je plocha křídla, C_D součinitel odporu, C_Y součinitel boční síly a C_L součinitel vztlaku. Všechny tyto součinitele mohou být závislé na úhlu náběhu, vybočení a vzdušné rychlosti.

Momenty

Aerodynamické momenty zásadním způsobem ovlivňují stabilitu letounu. Pro stabilní letouny platí, že v případě odchylky od stabilního stavu by momenty měly působit proti odchylce a tím tuto vzniklou odchylku korigovat bez jakéhokoliv zásahu řízení.

Pro popis momentů působících na tělo letounu je vhodné použít letadlovou souřadnou soustavu a uvažovat momenty působící okolo jejích os.

Mezi aerodynamické momenty se řadí tyto:

- klonivý moment (M_l)
- klopivý moment (M_m)
- stáčivý moment (M_n)

Ty lze pro nízké rychlosti « 1 Mach vypočítat podle vztahů

$$M_l = \bar{q}\, S\, b\, C_l \quad M_m = \bar{q}\, S\, \bar{c}\, C_m \quad M_n = \bar{q}\, S\, b\, C_n, \tag{3.13}$$

kde S je plocha křídla, b rozpětí, $\bar{c} = \frac{S}{b}$ střední geometrická tětiva křídla, C_l součinitel klonivého momentu, C_m součinitel klopivého momentu a C_n součinitel stáčivého momentu [15].

K takto vypočteným momentům je nutné přičíst aerodynamický tlumivý moment, který lze pro každou osu zvlášť vypočítat jako

$$M_D = \frac{1}{2}\sigma(\omega)\, \omega^2\, \rho_{air}\, C_{md}, \tag{3.14}$$

kde σ značí funkci signum, ω je úhlová rychlost v dané ose a C_{md} koeficient aerodynamického tlumení v dané ose.

Vliv aktuátorů

Výchylky aktuátorů (běžně křidélka, výškové a směrové kormidlo, případně klapky) ovlivňují hodnoty aerodynamických součinitelů. Hodnota o kterou může daný aktuátor ovlivnit jednotlivé součinitele, je závislá zejména na velikosti jeho výchylky, ale také na úhlu náběhu či úhlu vybočení.

3.2.4 Motorové síly a momenty

Tyto síly a momenty vyvozují motory s vrtulemi. Velikost těchto sil a momentů závisí tedy jak na použité vrtuli, tak na motoru a jeho aktuálním výkonu.

Síly

Motorové síly jsou vyvozovány listy vrtule, které ohýbají proud nabíhajícího vzduchu směrem za sebe, čímž je vytvářen tah. V případě, že rychlost vzduchu, nabíhajícího na vrtuli je nulová, hovoříme o statickém tahu, jinak se jedná o tah dynamický. Velikost tahu lze vypočítat jako

$$F_T = C_T\, \rho n^2 D^4, \tag{3.15}$$

kde ρ je hustota vzduchu, n frekvence otáčení vrtule (počet otáček za sekundu), D průměr vrtule a C_T koeficient tahu, který závisí na konkrétní vrtuli. Koeficient tahu je v případě dynamického tahu závislý také na rychlostním koeficientu

$$J = \frac{v}{n\,D}, \tag{3.16}$$

kde v je rychlost nabíhajícího proudu vzduchu. Pro několik vrtulí byly tyto koeficienty, včetně závislosti na rychlostním koeficientu, naměřeny v [7].

Momenty

Při ustáleném otáčení působí na rotor s vrtulí brzdný moment způsobený odporem vzduchu. Tento moment v důsledku působí na stator pohonného ústrojí a tím i na celý letoun. Velikost tohoto momentu je závislá především na použité vrtuli a rychlosti otáčení. Při zrychlení rotoru navíc působí na stator moment setrvačnosti rotoru s vrtulí. Oba tyto momenty působí pouze v ose otáčení rotoru, ale v opačném směru. Kompenzace těchto momentů je nejčastěji dosahováno přidáním protiběžného rotoru. Tohoto principu je využíváno například pro dosažení boční stability u multikoptér.

Moment vyvozovaný vrtulí lze vypočítat jako

$$M_Q = C_Q\,\rho n^2 D^5, \tag{3.17}$$

kde C_Q je koeficient momentu, který lze vypočítat z koeficientu výkonu C_P jako $C_Q = \frac{C_P}{2\pi}$. I tyto momenty byly naměřeny v [7].

Druhým způsobem vyvození momentu je princip diference tahu, vyplývající přímo z definice momentu síly, kdy na konci ramene působí síla vyvozovaná motorem. Moment lze potom vypočítat podle vztahu

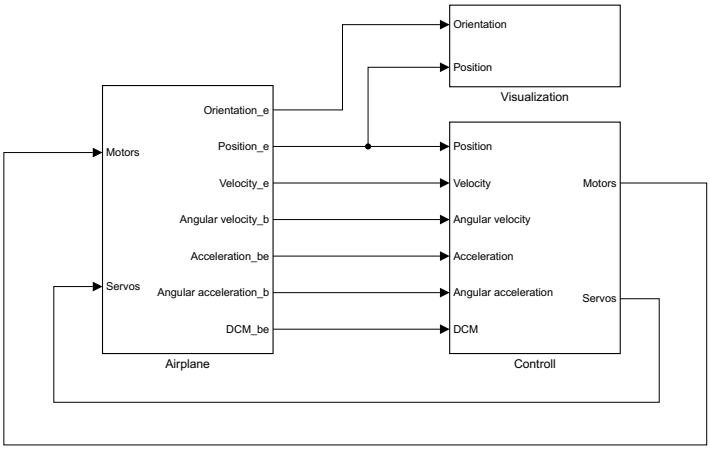
$$M_T = r \times F,$$

kde r je směrový vektor polohy motoru v letadlové soustavě s počátkem v těžišti a F vektor tahu motoru v letadlové soustavě. V případě použití více motorů se jednotlivé vektory momentů sčítají. Tento princip je využíván pro dosažení podélné a příčné stability u multikoptér.

Kapitola 4

Simulace

Pro vytvoření nelineárního modelu letové dynamiky VTOL prostředku bylo využito prostředí Matlab & Simulink s využitím doplňku Aerospace toolbox. Vytvořený model poskytuje rozhraní pro vizualizaci open-source leteckým simulátorem Flight Gear a pro řízení autopilotem pomocí ArduPilot v režimu SITL. Základní schéma simulace je zobrazeno na obrázku 4.1.



Obrázek 4.1: Základní schéma simulace

4.1 Simulovaný prostředek

Simulovaný model, který tato práce popisuje, je do velké míry vytvořen obecně a lze jej tedy využít pro různé konfigurace pouze pozměněním příslušných

parametrů. Teoreticky lze vytvořit model nejen různých VTOL prostředků, ale i konvenčních letounů či multikoptér. Tato práce se však soustředí na popis konkrétního VTOL prostředku.

Tímto prostředkem je letoun kachní koncepce se čtyřmi motory, z nichž 2 jsou překlopné pro režim vodorovného letu. Jedná se tedy o konstrukci typu quadplane-tiltrotor. Tento prostředek bude dosahovat rozpětí 1,3 m a hmotnosti bez užitečné zátěže cca 1,2 kg.

4.1.1 Pohon a elektronika

Jako pohonné jednotky pro tento model jsou plánovány 4 motory T-Motor MT2208 KV1100, ovládané elektronickým regulátorem otáček 4 v 1 Racerstar RS20AX4. Motory budou osazeny vrtulemi 8,5x6". Jako pohonná baterie bude sloužit Turnigy Multistar se třemi lithium-polymerovými články o kapacitě 5000 mAh. Jako řídicí jednotka bude sloužit Pixhawk s firmware ArduPlane.

4.2 Model letové dynamiky

Model letové dynamiky (FDM - Flight Dynamics Model) je vytvořen jako nelineární systém. Mezi vstupy tohoto systému patří ovládání jednotlivých motorů a serv. Výstupem je poloha, orientace, rychlost v zemské i letadlové soustavě, úhlová rychlost, lineární a úhlové zrychlení a transformační matice mezi zemskou a letadlovou souřadnou soustavou. Tento model zároveň musí mít definované počáteční podmínky, mezi které patří poloha, orientace, rychlost a úhlová rychlost.

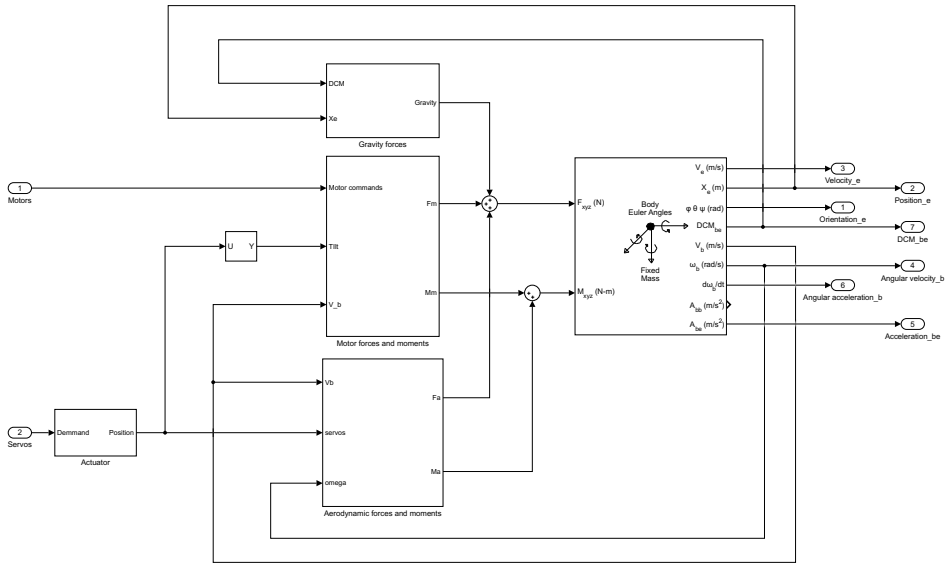
Ovládání motorů a serv je reprezentováno dvěma vektory, jejichž hodnoty jsou v rozsahu 0-1 pro každý motor, respektive servo, kde 0 představuje vypnutý motor, respektive minimální výchylku serva, a 1 maximální výkon motoru, respektive maximální výchylku serva.

Základem modelu letové dynamiky (viz obrázek 4.2) je blok *6DoF*, implementující všechny základní pohybové rovnice v šesti stupních volnosti. Na základě sil a momentů působících na letoun vypočítá polohu, orientaci, rychlost, úhlovou rychlost, zrychlení a další základní veličiny.

Jak bylo popsáno v části 3.2, na letoun působí síly a momenty aerodynamické, motorové a gravitační. Všechny tyto síly a momenty jsou v modelu zahrnuty. Pro lepší přehlednost jsou jednotlivé druhy sil a momentů rozděleny do samostatných bloků.

4.2.1 Motorové síly a momenty

Vstupem do bloku simulace motorových sil a momentů je vektor povelů řízení pro jednotlivé motory, hodnota otočení sklopného mechanismu motorů a vektor vzdušné rychlosti. Výstupem bloku jsou síly a momenty vyvozované pohonem, které působí na letoun v letadlové souřadné soustavě. Síly a momenty jednotlivých motorů jsou nejprve vypočítány zvlášť a následně jsou na základě nakonfigurovaného uspořádání vypočítány celkové síly a momenty.



Obrázek 4.2: Schéma simulace modelu letové dynamiky

Otáčky motoru

Závislost otáček motoru na vstupních povelech byla aproximována lineárním systémem prvního řádu (viz obrázek 4.3). Vstupem systému je vektor povelů pro jednotlivé motory a výstupem jejich úhlová frekvence. Lineární systém reprezentuje zpoždění mezi povelem a reakcí a zároveň simuluje útlum točivého momentu motoru při zvyšujících se otáčkách. Přenosová funkce tohoto lineárního systému je dána vztahem

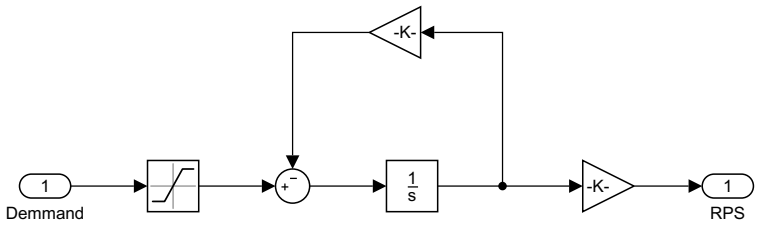
$$G_m(s) = \frac{\frac{f_{max}}{k\,T_m}}{s + \frac{1}{k\,T_m}},$$

kde f_{max} je maximální počet otáček motoru za sekundu, T_m časová konstanta, udávající přibližný čas ustálení otáček motoru při jednotkovém skoku a $k = 0,15$ je empiricky získaná konstanta pro kompenzaci času ustálení.

Sklopný mechanismus motorů

V modelu je uvažován sklopný mechanismus pro dva zadní motory. Způsob provedení simulace však umožňuje sklápět libovolnou kombinaci motorů pouze pozměněním inicializačních parametrů. Simulované sklopení motorů zajišťuje jak změnu směrování tahu vzhledem k letadlové souřadné soustavě, tak i změnu směru působení momentů vyvozovaných rotory.

Simulace sklopného mechanismu umožňuje libovolné spojité otočení pro zajištění maximální věrohodnosti, obzvláště v přechodových fázích. Pro každý



Obrázek 4.3: Simulace otáček motoru

motor lze navíc zvlášť zvolit minimální a maximální výchylku sklopného mechanismu.

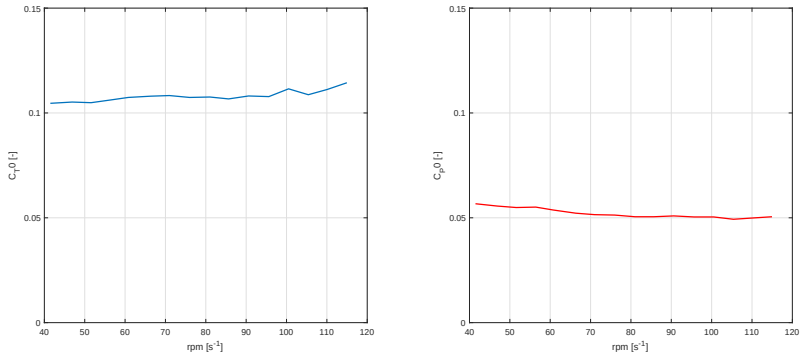
Požadavek na sklopení je určen úrovní příslušného kanálu výstupů emulované řídicí jednotky v rozsahu $0 - 1$. Tento požadavek je podle inicializačních parametrů, určujících minimální a maximální sklopení, přepočítán na úhel sklopení α_{tilt}, přičemž 0 znamená směr tahu vzhůru a $\pi/2$ směr tahu vpřed.

Dynamický tah

Pro výpočet dynamického tahu je nutné určit rychlost vzduchu, nabíhajícího na každou z vrtulí podle vztahu

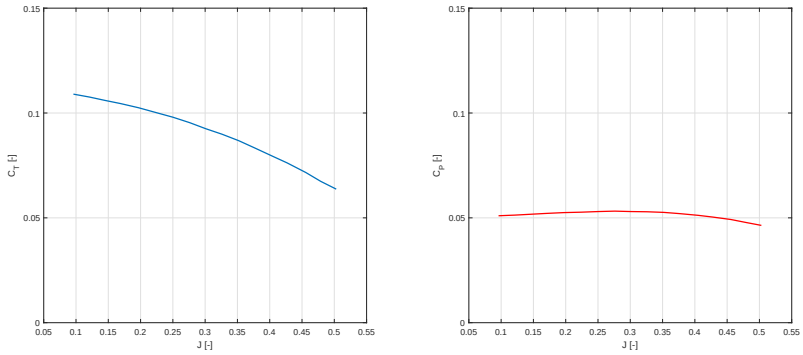
$$v_{prop} = \cos\left(\frac{pi}{2} - \alpha - \alpha_{tilt}\right)\sqrt{u^2 + w^2}.$$

Na základě rychlosti nabíhajícího proudu vzduchu na vrtuli, aktuálních otáček a průměru vrtule je vypočítán rychlostní koeficient J podle vztahu 3.16. Dle hodnoty rychlostního koeficientu je z naměřených dat z [7] interpolací získán koeficient tahu a koeficient momentu. Následně jsou vypočítány síly a momenty vyvozované danou vrtulí dle vztahů 3.15 a 3.15.



Obrázek 4.4: Charakteristika koeficientu statického tahu a výkonu v závislosti na otáčkách vrtule 8,5x6

Pro simulovaný prostředek byla uvažována vrtule 8,5x6. Naměřená statická charakteristika této vrtule z [7] je na grafu 4.4 a dynamická charakteristika



Obrázek 4.5: Charakteristika koeficientu dynamického tahu a výkonu v závislosti na otáčkách vrtule 8,5x6

na grafu 4.5. Z uvedených charakteristik vyplývá, že statické koeficienty jsou na otáčkách jen velmi málo závislé. Hodnota dynamických koeficientů pro rychlostní koeficient $J = 0$ tedy bude uvažována jako střední hodnota koeficientů statických. Blok výpočtu dynamického tahu motorů je na obrázku 4.6.

Při nulových otáčkách dochází při výpočtu rychlostního koeficientu k dělení nulou a hodnota J se tedy limitně blíží nekonečnu. V simulaci je tedy přidána podmínka pro $n < 0,5s^{-1}$, kdy je hodnota rychlostního koeficientu pevně stanovena na maximální naměřenou hodnotu J. Hodnota obou koeficientů bude tím pádem konečná a při výpočtu tahu a momentu bude vynásobena n. Výsledný tah i moment bude tedy zanedbatelný, což odpovídá předpokladům při takto nízkých otáčkách.

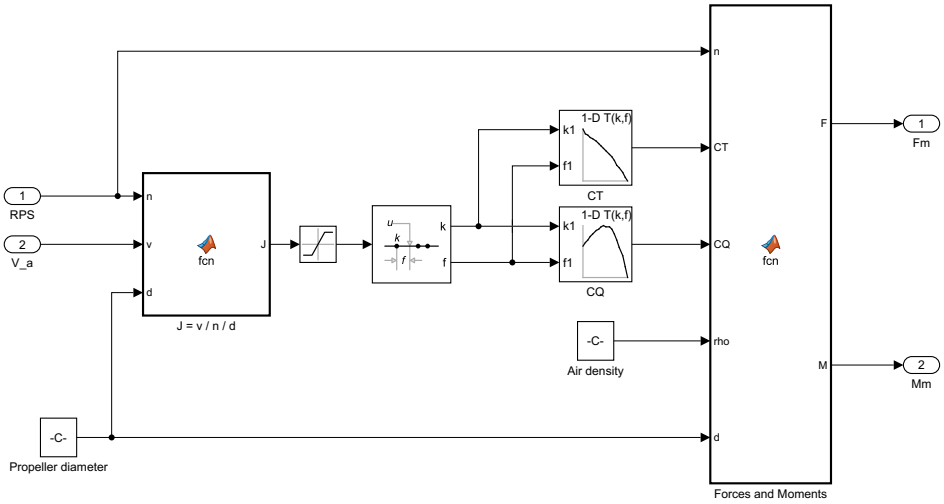
Výpočet celkových sil a momentů pohonu

Výpočet celkových sil a momentů pohonu je patrný ze schématu 4.7. Výsledné síly jsou dány sumou příspěvků sil vyvozovaných jednotlivými motory v příslušných osách. Celkový moment je také dán součtem momentů vyvozovaných jednotlivými motory, ale navíc je přidán moment způsobený diferencí tahu motorů. Ten je vypočítán na základě polohy jednotlivých motorů, která je definována inicializačními parametry. Motorové síly ve směru osy Y nejsou uvažovány.

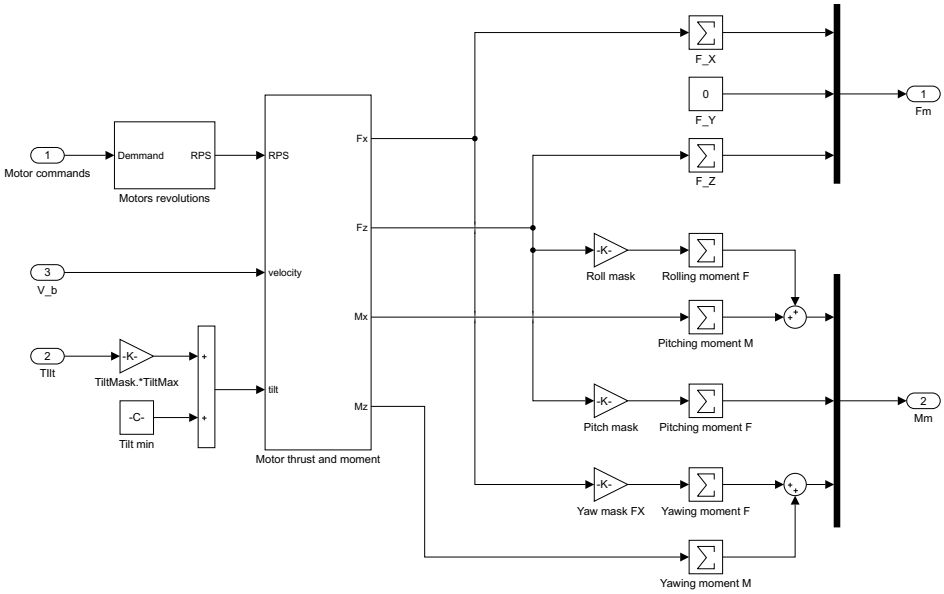
4.2.2 Aerodynamické síly a momenty

Vstupem do tohoto bloku je vektor vzdušné rychlosti, úhlové rychlosti a výchylek jednotlivých serv. Na základě velikosti složek vzdušné rychlosti je vypočítán úhel náběhu a úhel vybočení dle vztahu 3.1 a 3.2. Na základě vzdušné rychlosti je vypočítán dynamický tlak dle vztahu 3.11. Blok výpočtu aerodynamických sil a momentů je znázorněn na obrázku 4.8.

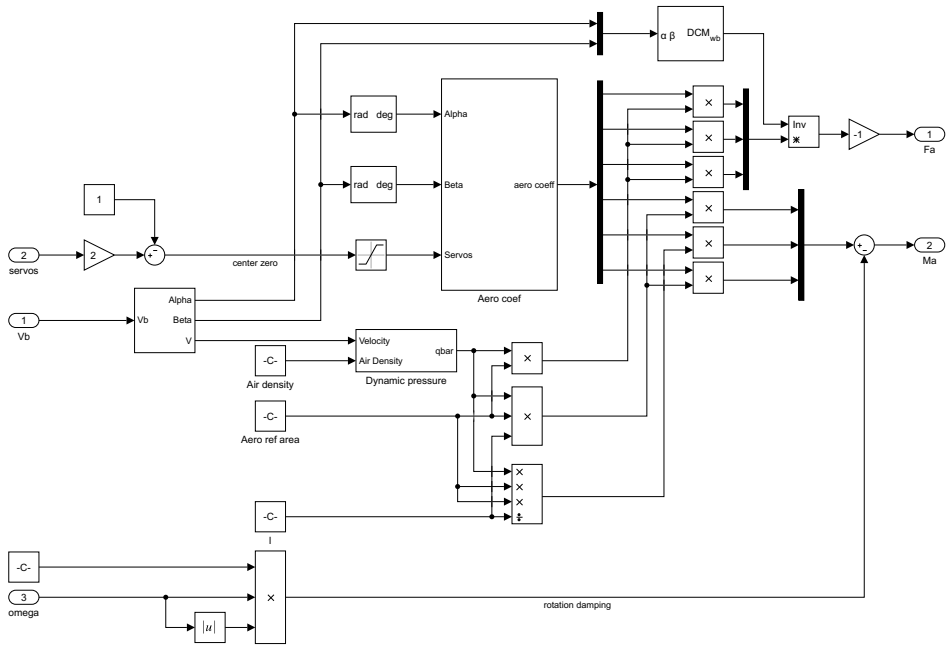
Pro simulaci aerodynamického tlumení otáčivého pohybu je do tohoto bloku navíc přidán brzdný moment, vypočítaný dle vztahu 3.14.



Obrázek 4.6: Schéma systému pro simulaci dynamického tahu motoru



Obrázek 4.7: Blok výpočtu celkových sil a momentů pohonu



Obrázek 4.8: Blok výpočtu aerodynamických sil a momentů

Výpočet aerodynamických součinitelů

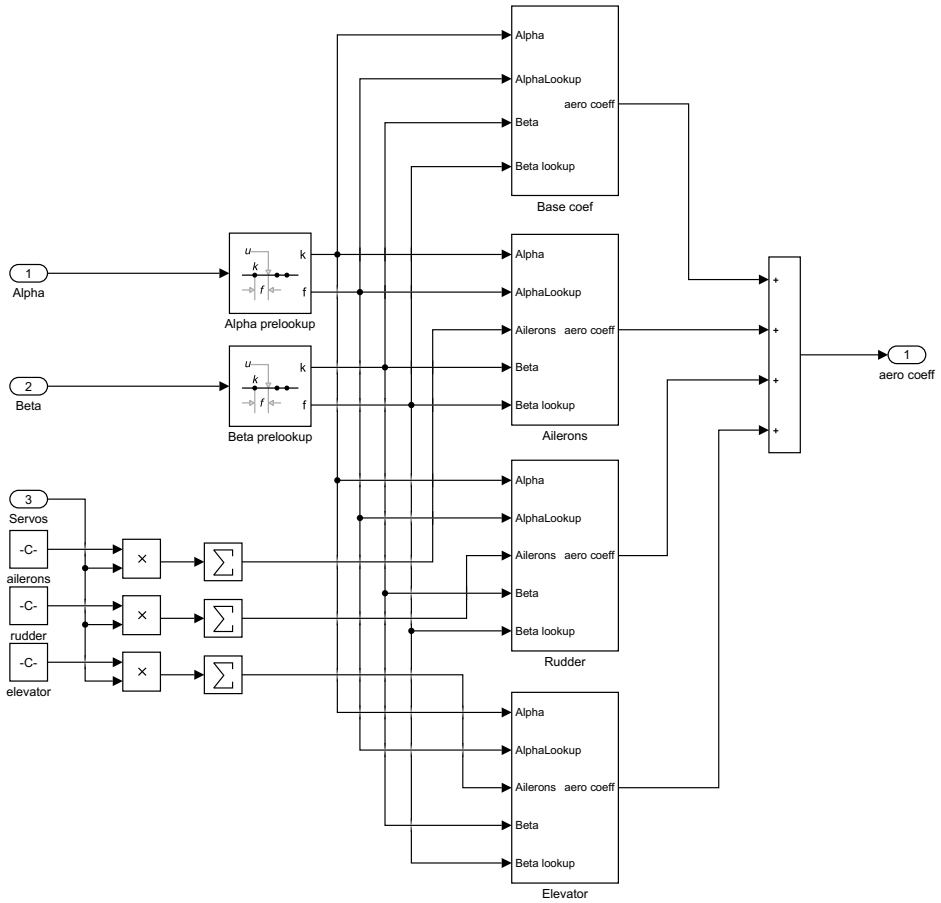
Aerodynamické součinitele lze rozdělit do dvou skupin: základní součinitele, vycházející pouze z úhlu náběhu a vybočení, a součinitele ovlivňované aktuátory, které jsou navíc závislé na výchylce příslušného aktuátoru. Blok zodpovědný za získání výsledných aerodynamických součinitelů je na obrázku 4.9.

Dvojrozměrnou interpolací jsou na základě úhlu náběhu a úhlu vybočení získány hodnoty základních aerodynamických součinitelů. Dvojrozměrná interpolace v tomto případě vyjadřuje závislost na hodnotách obou úhlů zároveň. Blok zodpovědný za získání základních součinitelů je na obrázku 4.10.

Vliv aktuátorů

Reakce na změnu výchylky aktuátoru aerodynamických sil se projeví změnou aerodynamických součinitelů, čímž je dosaženo změny aerodynamických momentů a sil. Mezi aktuátory aerodynamických sil v této práci patří směrové kormidlo, výškové kormidlo a křidélka.

Vliv aktuátoru je v simulaci přímo úměrný výchylce serva a má aditivní vliv na všechny aerodynamické součinitele. Hodnoty přírůstků jednotlivých součinitelů při maximální výchylce lze stanovit individuálně pro všechny aktuátory a jejich hodnota může být závislá na úhlu náběhu i úhlu vybočení. Lze tak simulovat skutečnost, že aktuátory mohou ztratit účinnost například při vysokém úhlu náběhu či vybočení.



Obrázek 4.9: Výpočet celkových aerodynamických součinitelů

Celková hodnota libovolného součinitele C_x je tedy dána vztahem

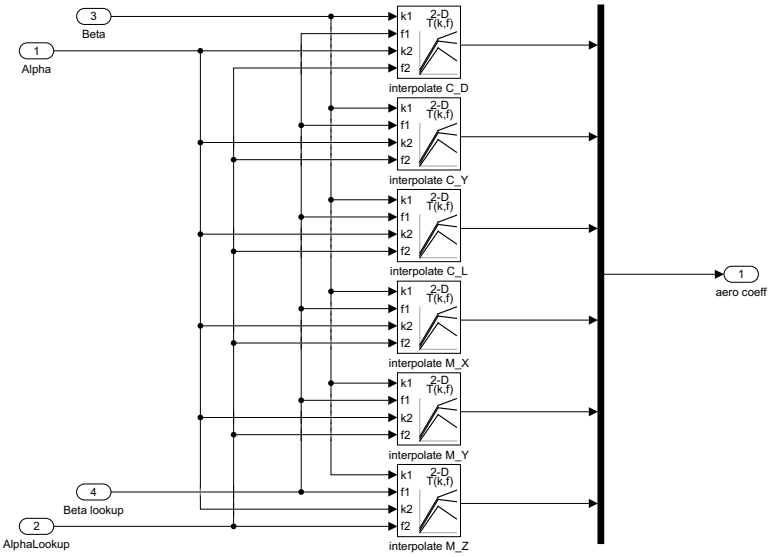
$$C_x = C_{x,base}(\alpha, \beta) + \sum_i \delta_i \, C_{x,i}(\alpha, \beta),$$

kde $C_{x,base}$ je základní hodnota zvoleného součinitele při nulových výchylkách, závislá na úhlu náběhu a vybočení, δ_i je hodnota výchylky i-tého aktuátoru v rozsahu $< -1, 1 >$ a $C_{x,i}$ hodnota daného koeficientu při maximální výchylce i-tého aktuátoru. Hodnota součinitele $C_{x,i}$ je v simulaci interpolována na základě úhlu náběhu a vybočení.

Serva

Pro simulaci dynamiky serv ovládajících kormidla i sklopný mechanismus byla využita aproximace lineárním systémem druhého řádu (viz obrázek 4.11). Přenosová funkce je dána vztahem [25]

$$G_s(s) = \frac{\omega^2}{s^2 + 2\,\zeta\,\omega\,s + \omega^2},$$



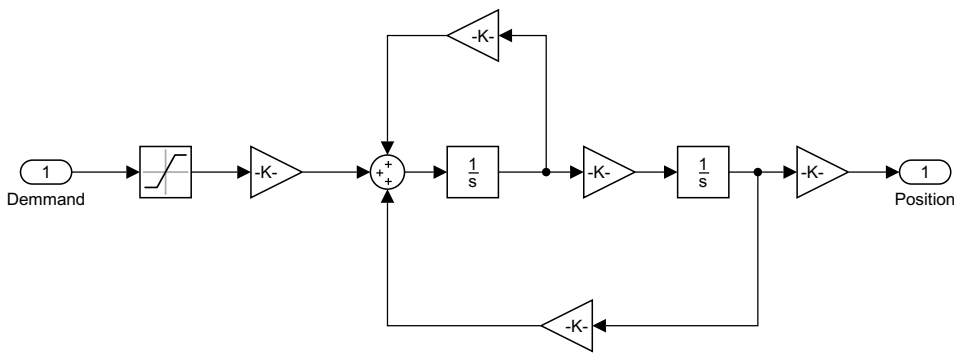
Obrázek 4.10: Blok interpolace základních aerodynamických koeficientů

kde ζ, závislé na podílu překmitu serva k hodnotě ustálené hodnoty δ_o, lze vypočítat jako

$$\zeta = \frac{-\ln(\delta_o)}{\sqrt{\pi^2 + \ln(\delta_o)}},$$

a ω, závislé na času ustálení výchylky T_s, jako

$$\omega = \frac{4}{T_s\,\zeta}.$$



Obrázek 4.11: Systém simulace výchylky serva

4.2.3 Gravitační síla

Gravitační síla působí směrem do středu Země, což je v případě lokální geografické souřadné soustavy ve směru osy z. Pro zohlednění účinků gravitační síly působící na letoun je potřeba provést transformaci do letadlové soustavy podle vztahu 3.10.

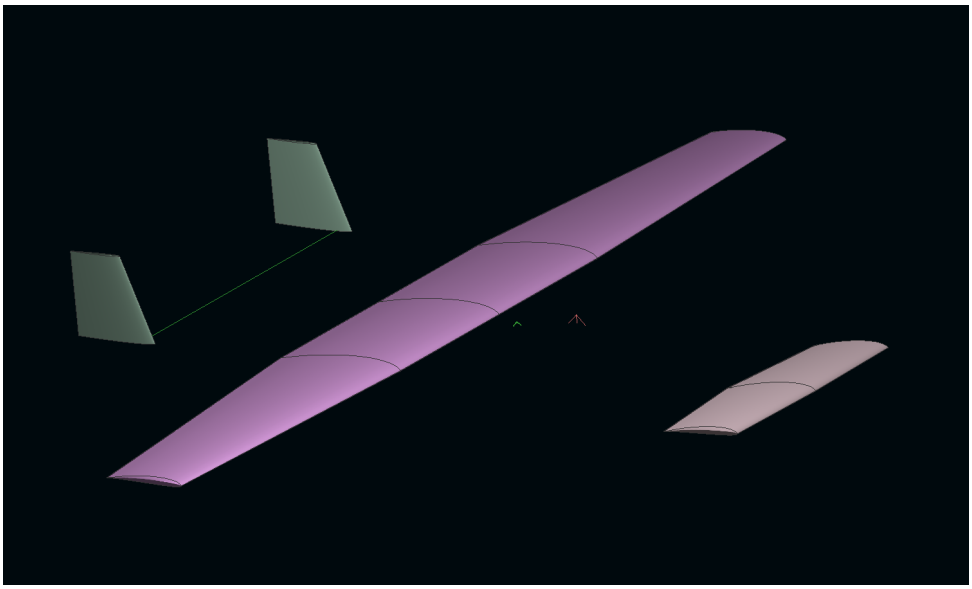
Pro simulaci pevné země je přidána podmínka, že síla působí pouze v kladné relativní výšce. Toho je využíváno především při startu, aby na počátku na model nepůsobily žádné síly. Případ nárazu do země není simulován, protože simulace slouží primárně pro ověření letových parametrů.

4.3 Identifikace parametrů simulovaného prostředku

Aby bylo možné provést přesnou simulaci letové dynamiky daného prostředku, je nutné identifikovat jeho parametry. Jelikož tento prostředek prozatím nebyl vyroben, je většina identifikovaných parametrů získaná teoreticky.

4.3.1 Aerodynamické součinitele

Pro identifikaci aerodynamických součinitelů byl vytvořen model v prostředí XFLR5 (viz obrázek 4.12). Tento software umožňuje výpočet aerodynamických součinitelů za použití panelové metody. Umožňuje dobře určit parametry pro let v obálce stabilního režimu, kdy dochází převážně k laminárnímu proudění, ale není vhodný pro výpočet parametrů při proudění turbulentním, ke kterému dochází při vysokých úhlech náběhu.



Obrázek 4.12: Model letounu v prostředí XFLR5

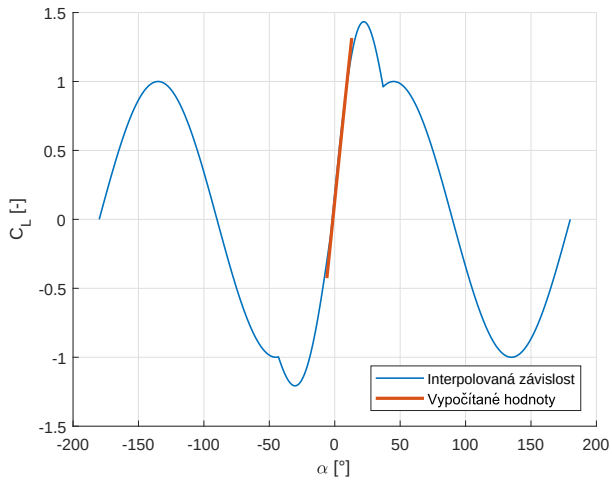
Jelikož ale cílem této práce je simulace prostředku s kolmým startem a

přistáním, je potřeba uvažovat s možnými úhly náběhu -180°až 180°. V případě výše uvedeného modelu byly zjištěny hodnoty koeficientů pro úhly náběhu v rozmezí -10°až 12°. Ideálním způsobem, jak zjistit hodnoty koeficientů pro takto vysoké úhly náběhu je experimentální naměření ve větrném tunelu.

Rychlost letounu při takto vysokých úhlech náběhu však bude zpravidla velmi nízká a generované aerodynamické síly a momenty tedy budou relativně malé. Díky tomuto předpokladu je pak možné spokojit se s méně přesnými hodnotami aerodynamických součinitelů, které lze odhadnout na základě známých parametrů letounu a teoretické závislosti součinitelů pro vysoké úhly náběhu. Problematikou vysokých úhlů náběhu se zabývá [24].

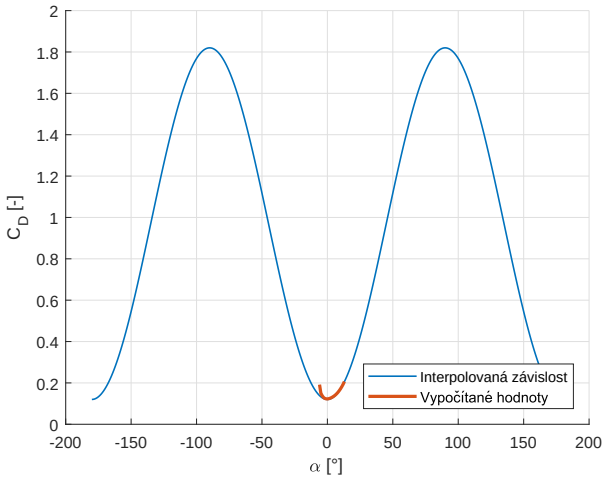
Vytvořený model respektuje profily křídel, které budou použity na skutečném letounu - profil SD7037 pro křídla a kachní plochu a profil NACA 0009 pro svislé ocasní plochy. Tyto profily byly podrobeny analýze pro několik hodnot Reynoldsova čísla - 10 000 až 150 000 s krokem 10 000.

V simulaci jsou však prozatím použity interpolované hodnoty, aby byly pokryty úhly náběhu v celém rozsahu přípustných hodnot. Interpolace probíhala manuálně, pomocí jedné nebo kombinace více sinusovek, případně s využitím Gaussovy funkce tak, aby výsledná hodnota odpovídala hodnotám, získaným z programu XFLR5 a zároveň teoretickým průběhům z [24]. Veškeré závislosti byly tedy určeny pouze hrubě, pro účely dalších experimentů a budou zpřesňovány v budoucím postupu, kdy bude například možné jejich experimentální naměření ve větrném tunelu.



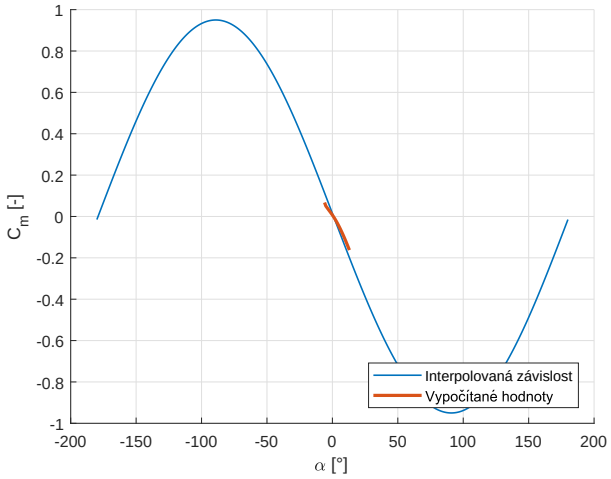
Obrázek 4.13: Závislost součinitele vztlaku na úhlu náběhu

Na obrázku 4.13 je graf závislosti součinitele vztlaku C_L na úhlu náběhu α při úhlu vybočení $\beta = 0°$. Interpolace je tvořena součtem dvojice sinusovek reprezentujících laminární a turbulentní proudění. V bodě maxima a minima dochází k odtržení laminárního proudění a skokově klesá hodnota součinitele vztlaku. Ve skutečnosti navíc existuje určitá hystereze mezi odtržením a opětovným dosažením laminárního proudění, ale tato skutečnost je v simulaci zanedbána.



Obrázek 4.14: Závislost součinitele odporu na úhlu náběhu

Na obrázku 4.14 je graf závislosti součinitele odporové síly C_D na úhlu náběhu α při úhlu vybočení $\beta = 0°$. Interpolace součinitele odporové síly je tvořena jedinou sinusovkou s minimy v bodech $\alpha = 0°$ a $\alpha = \pm 180°$ a maximy v bodech $\alpha = \pm 90°$.

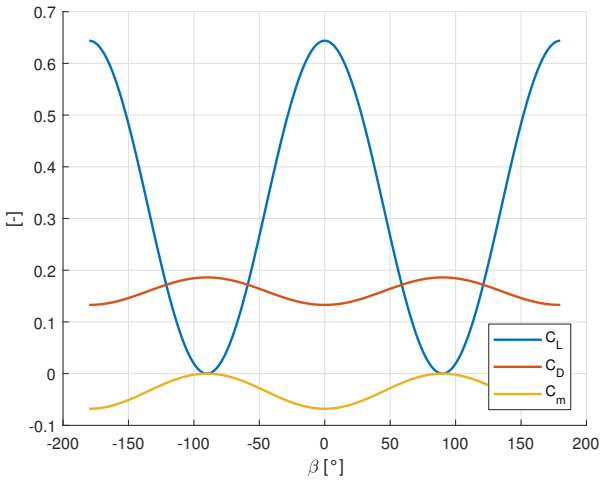


Obrázek 4.15: Závislost součinitele klopivého momentu na úhlu náběhu

Na obrázku 4.15 je graf závislosti součinitele klopivého momentu C_m na úhlu náběhu α při úhlu vybočení $\beta = 0°$. Interpolace součinitele odporové síly je tvořena jedinou sinusovkou s minimem v bodě $\alpha = 90°$ a maximem v bodě $\alpha = -90°$.

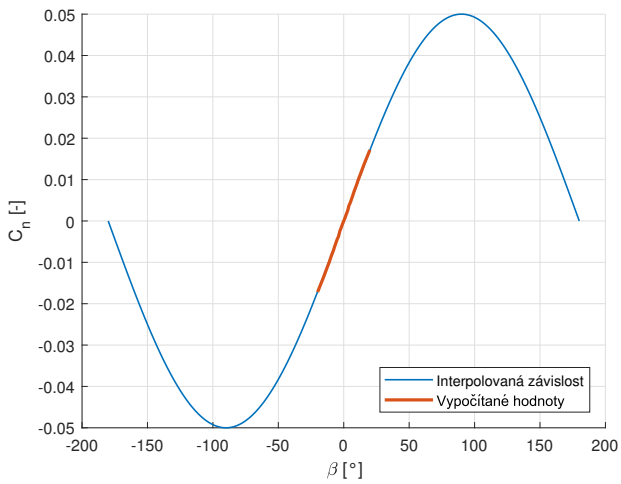
Závislost výše uvedených součinitelů na úhlu vybočení byla určena podle teoretických poznatků. Velikost součinitele vztlaku a klopivého momentu se při zvyšování úhlu vybočení snižuje a naopak velikost součinitele odporu roste. Ovlivnění těchto součinitelů úhlem vybočení bylo pro účely této práce

pouze odhadnuto dle teoretických závislostí. Vliv takto vzniklých sil nebude nijak zásadní, protože při rychlejším letu, v němž by se tyto síly projevily, již dochází pouze k minimálnímu vybočení. Na obrázku 4.16 je příklad ovlivnění hodnot těchto součinitelů úhlem vybočení.



Obrázek 4.16: Závislost součinitelů na úhlu vybočení při úhlu náběhu 5°

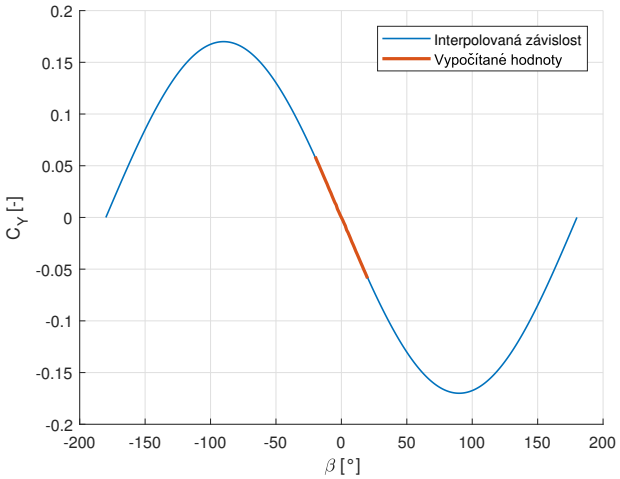
Součinitel stáčivého momentu je závislý především na úhlu vybočení, stejně jako součinitel boční síly. Tyto závislosti jsou patrné z grafů 4.17 a 4.18.



Obrázek 4.17: Závislost součinitele stáčivého momentu C_n na úhlu vybočení při úhlu náběhu $\alpha = 0°$

Vliv ovládacích ploch

Pro identifikaci vlivu ovládacích ploch letounu na aerodynamické součinitele byly vytvořeny 4 další upravené modely v XFLR5: s maximální výchylkou křidélek, směrových kormidel, výškového kormidla nahoru a výškového kormidla



Obrázek 4.18: Závislost součinitele boční síly C_Y na úhlu vybočení při úhlu náběhu $\alpha = 0°$

dolů. Upravené modely slouží pro identifikaci absolutních hodnot součinitelů při maximální výchylce vždy jen jednoho aktuátoru v závislosti na úhlu náběhu a vybočení.

Výchylky křidélek a směrových kormidel jsou symetrické akce podle roviny XZ a tudíž lze předpokládat stejnou reakci v opačném směru při opačné výchylce. V případě výškového kormidla se však reakce může v obou směrech lišit.

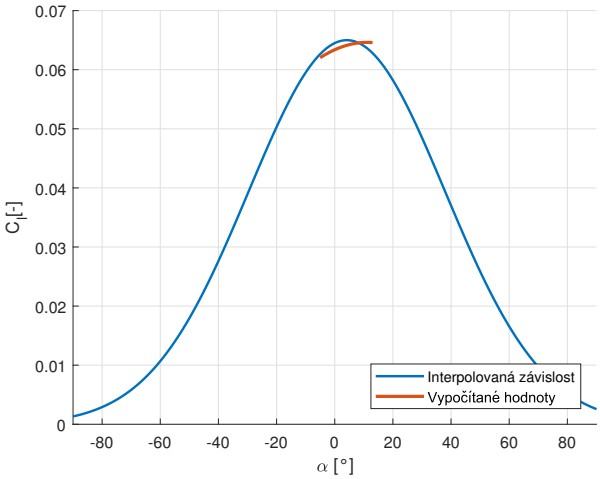
Z analýzy provedené v XFLR5 vyplývá, že relativní změna hodnoty součinitele klopivého momentu se při stejné výchylce výškového kormidla v obou směrech liší pouze nepatrně. Při výchylce nahoru je relativní změna součinitele klopivého momentu $\delta_C m = 0.0258/°$, zatímco při výchylce směrem dolů $\delta_C m = 0.0262/°$. Prozatím byla tedy diference mezi smysly výchylek v simulaci zanedbána.

Na grafu 4.19 je znázorněna závislost součinitele aditivního klonivého momentu na úhlu náběhu při maximální výchylce křidélek. Jelikož je díky symetrii v rovině XZ hodnota součinitele klonivého momentu při nulových výchylkách všech aktuátorů nulová, je aditivní hodnota součinitele klonivého momentu rovna hodnotě absolutní.

Při zkoumání vlivu aktuátorů na součinitele, které při nulových výchylkách nejsou nulové, je pro získání aditivní hodnoty potřeba od naměřené nebo vypočítané absolutní hodnoty daného součinitele odečíst základní hodnotu bez působení aktuátorů.

4.3.2 Moment setrvačnosti

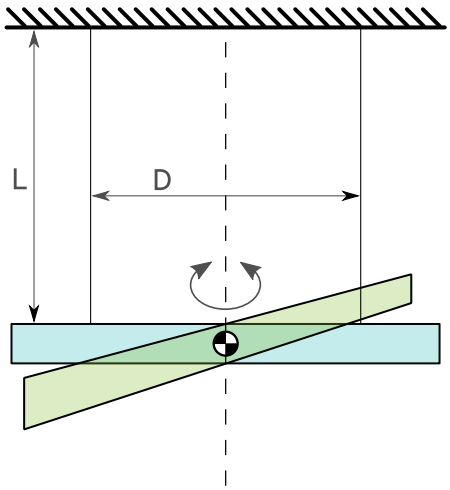
Moment setrvačnosti ve formě tenzoru je získán jako součet momentu setrvačnosti draku letounu, kde je předpokládáno spojité rozložení hmoty, a bodových příspěvků momentu setrvačnosti, které reprezentují hmotnost jednotlivých komponent - baterie, motorů, řídicí jednotky, serv, a podobně.



Obrázek 4.19: Ovlivnění součinitele klonivého momentu C_l při maximální výchylce křidélek v závislosti na úhlu náběhu α

Výsledný tenzor setrvačnosti byl také vypočítán pomocí software XFLR5 a jeho hodnota je

$$I = \begin{bmatrix} 0.0409 & 0 & -0.0038 \\ 0 & 0.0585 & 0 \\ -0.0038 & 0 & 0.0978 \end{bmatrix} kg\, m^2.$$



Obrázek 4.20: Schéma měření momentu setrvačnosti

Pro získání přesnějších výsledků by však bylo vhodné provést experimenty se skutečným letounem. Hodnotu momentu setrvačnosti lze získat například změřením periody T kroutivého kyvu na dvou závěsech délky L, které jsou od sebe ve vzdálenosti $D = 2r$. Osa otáčení přitom musí procházet těžištěm. Schéma měřící aparatury je na obrázku 4.20. Hodnota momentu setrvačnosti

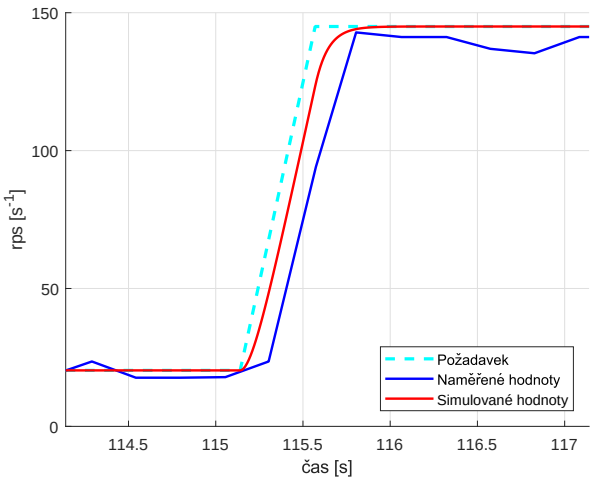
v ose otáčení je pak dána vztahem

$$I = \frac{m\,g\,r^2\,T^2}{4\pi^2\,L}.$$

4.3.3 Parametry motorů a serv

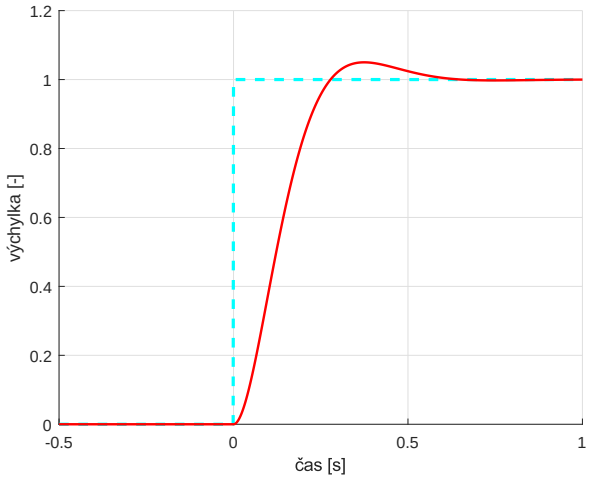
Pro identifikaci časové konstanty motoru pro rozběh z nízkých otáček na plnou rychlost je potřeba provést měření otáček motoru s vrtulí v závislosti na čase při jednotkovém skoku povelů. Na hodnotu konstanty má vliv jak použitý motor s vrtulí, tak regulátor otáček a napájecí napětí. Komponenty, použité pro identifikaci, by tedy měly být totožné s komponentami použitými na simulovaném prostředku.

Pro měření přechodové charakteristiky byla využita aparatura s optickým snímačem otáček a vývojovým kitem Arduino, sloužícím pro řízení regulátoru otáček motoru a zároveň také pro zachycení a předzpracování dat. Z grafu 4.21 je patrná naměřená charakteristika. Použitá aparatura byla původně určena pro měření statických charakteristik motorů a vrtulí, a proto v době měření neumožňovala vytvoření jednotkového skoku. Zároveň také měřící aparatura nedokázala dosáhnout dostatečně vysoké vzorkovací frekvence. V důsledku toho výsledky nelze považovat za směrodatné, ale spíše za informativní.



Obrázek 4.21: Odezva na rychlou změnu řídicích povelů motoru

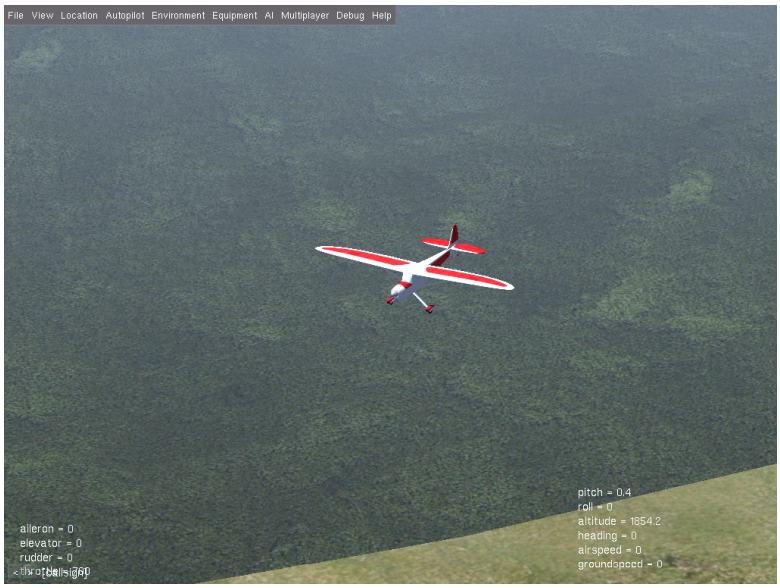
Časovou konstantu serv lze stanovit podobným způsobem - tedy změřením doby, kterou zabere přechod z minimální výchylky serva do maximální. Pro účely této práce byly časové konstanty motorů i serv prozatím pouze odhadnuty na $T_m = 0,5s$. Na grafu 4.22 je znázorněna odezva na jednotkový skok systému, reprezentujícího mechaniku serva.



Obrázek 4.22: Odezva na jednotkový skok řídicích povelů serva

4.4 Vizualizace

Pro 3D vizualizaci (viz obrázek 4.23) byl využit open-source letecký simulátor FlightGear. Ten umožňuje spuštění bez modelu letové dynamiky a zároveň komunikaci přes síťové připojení pomocí protokolu UDP. Díky tomuto spojení lze předávat data ve formě FGFDM paketů, které obsahují informace o letounu, ovládání či prostředí. Pro účely této práce je však toto rozhraní využíváno pouze pro aktualizaci dat o globální poloze a orientaci letounu.



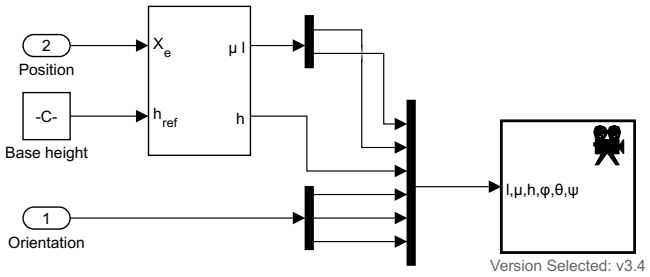
Obrázek 4.23: Ukázka vizualizace

Blok vizualizace (viz obrázek 4.24) zároveň zodpovídá za zpomalování simulace v případě, že by běžela rychleji, než v reálném čase. Ke spuštění

vizualizace je buď potřeba provést konfiguraci leteckého simulátoru FlightGear, nebo jej spustit pomocí skriptu s již přednastavenými parametry. Těmito parametry jsou zejména:

- fdm = null
- native-fdm = socket,in,30,localhost,5502,udp

Dále je vhodné nastavit jako počáteční bod letiště, které je poblíž startovní pozice, aby byla správně vykreslena krajina. Dalšími parametry lze zvolit letoun, čas, viditelnost a další.



Obrázek 4.24: Blok vizualizace

4.5 Propojení FDM s ArduPilot SITL

V současné době neexistuje žádná běžně dostupná knihovna, umožňující vytvoření SITL, případně HIL simulace s firmware ArduPilot v prostředí Matlab & Simulink. Na základě zdrojového kódu autopilotu ArduPilot, jeho komponenty ArduPilot SITL a zdrojových kódů ostatních simulátorů bylo vytvořeno rozhraní, umožňující komunikaci s virtuální řídicí jednotkou ArduPilot. Způsob provedení nevyžaduje žádný zásah do existujícího firmware. Využívá již existující rozhraní pro simulátor Gazebo, a proto celou simulaci bude možné provozovat i s budoucími verzemi autopilotu, pokud tyto budou i nadále podporovat toto rozhraní.

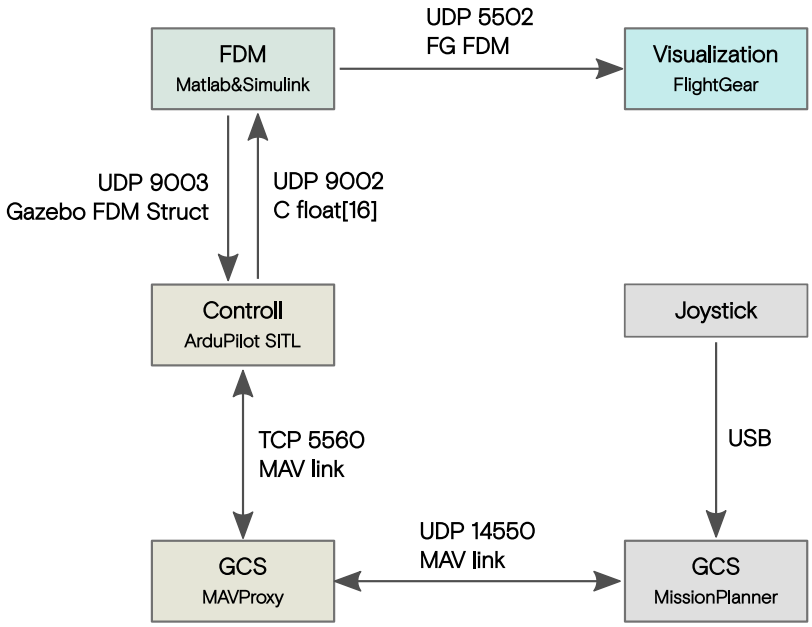
Na obrázku 4.25 je schématicky znázorněn diagram celé software-in-the-loop simulace. Komponenty a principy tohoto schématu jsou popsány níže.

4.5.1 Formát dat

Emulovaná řídicí jednotka ArduPilot v režimu SITL - Gazebo očekává data ve formě struktury dle standardů programovacího jazyka C. Tato data pak emulátor převádí na hodnoty virtuálních senzorů. Pro dosažení realističtější simulace lze také v emulátoru definovat míru zašumění dat z virtuálních senzorů.

Odesílaná struktura obsahuje:

- čas simulace v sekundách,



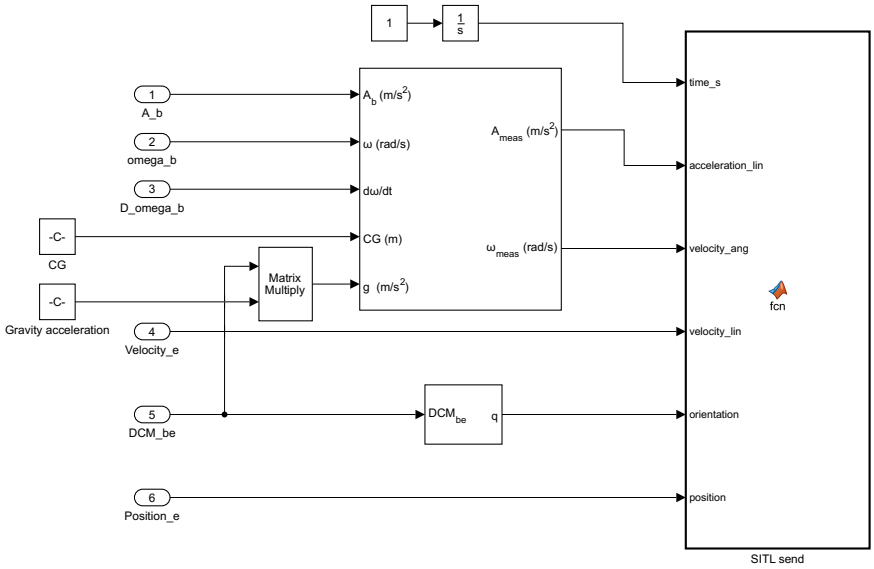
Obrázek 4.25: Schéma propojení simulace s dalšími komponentami

- vektor úhlové rychlosti letounu $\Omega_b\,[rad\,s^{-1}]$,
- vektor akcelerace, působící na letoun $a_b\,[m\,s^{-2}]$,
- orientaci letounu ve formě jednotkového kvaternionu
- vektor lineární rychlosti vůči Zemi v lokální geografické soustavě $v_e\,[m\,s^{-1}]$
- vektor polohy v lokální geografické soustavě $x_e\,[m]$

Na obrázku 4.26 je zobrazeno schéma zajišťující vhodnou úpravu dat z FDM před jejich odesláním. V tomto bloku je proveden výpočet akcelerace, působící na letoun jako součtu gravitačního zrychlení, transformovaného do letadlové souřadné soustavy, a akcelerace letounu vůči zemi rovněž v letadlové souřadné soustavě.

V případě, že by senzorická jednotka nebyla v těžišti letounu, je nutné provést přepočet dat, jelikož budou zrychlení, úhlové zrychlení a úhlové rychlosti vzájemně závislé. Pro tento přepočet slouží blok `6DoF IMU`.

Všechny data odesílaná z FDM jsou datového typu Double podle standardů jazyka C/C++. Výstupem z emulované řídicí jednotky je struktura s jediným prvkem - vektorem polohy 16 výstupů pro motory, serva a další případná zařízení, která mají být ovládána řídicí jednotkou letounu. Tyto výstupy jsou datového typu Float a nabývají hodnot od 0 do 1, kde 0 je minimální výchylka aktuátoru a 1 výchylka maximální.



Obrázek 4.26: Schéma zajišťující propojení s instancí emulátoru ArduPilot SITL

Vektory jsou odesílány i přijímány jako běžné pole hodnot - tedy po sobě jdoucí prvky v paměti s ukazatelem na první prvek. Pro vytvoření správného formátu dat byla využita knihovna Python Struct a její metoda `pack()` pro vytvoření odesílaného paketu, respektive metoda `unpack()` pro rozbalení přijatého paketu.

Aby nemusel být soket po každém odeslání či přijetí paketu zavřen a posléze znovu vytvářen, musí být instance soketu vytvořena ještě před spuštěním simulace. Pracovní prostředí Matlab dále musí zahrnovat funkce `simSend()` a `simReceive()`, které obsahují skripty s výše uvedenými funkcemi `pack()` a `unpack()`.

4.5.2 Virtuální senzory

Data přijatá z FDM jsou převedena na výstupní hodnoty virtuálních senzorů. Čas simulace v sekundách určuje frekvenci zpracování dat, vektor úhlové rychlosti je využit k simulaci tříosého gyroskopického senzoru, vektor akcelerace k simulaci akcelerometru, z kvaternionu orientace jsou vypočítána data pro virtuální tříosý kompas, a vektor rychlosti a polohy v lokální geografické soustavě je využit pro korekce polohy dle virtuálního navigačního systému. Globální poloha je určena při startu instance emulátoru řídicí jednotky a její aktualizace probíhá pouze na základě přijaté odchylky. Tato část je celá implementována v ArduPilot SITL včetně možnosti přidání šumu.

4.5.3 Spojení

Při spojení dynamického modelu s řídicí jednotkou, respektive její emulací, je využíváno protokolu UDP (User Datagram Protocol), který sice nezaručuje

doručení paketů a jejich integritu, ale je možné dosáhnout vyšší přenosové rychlosti, neboť není potřeba tolik výpočetního výkonu na režii přenosu, jako je tomu například v případě protokolu TCP (Transmission Control Protocol).

Toto spojení bylo implementováno pomocí knihovny Python Socket, kterou lze využívat i v prostředí Matlab. Soket pro příjem paketů z ArduPilot emulátoru je inicializován na portu 9002 a soket pro odesílání FDM paketů na portu 9003.

4.5.4 Časování a obnovovací frekvence

Obnovovací frekvence běžných serv a elektronických regulátorů otáček je 50Hz, neboť jsou řízeny pulsně-šířkově modulovaným signálem s šířkou okna 20ms. Simulace tuto skutečnost reflektuje tím, že čtení soketu probíhá pouze v nastaveném intervalu.

Vzorkovací frekvence čtení dat ze senzorů je ve skutečné řídicí jednotce Pixhawk cca 400Hz. S touto frekvencí jsou emulované jednotce posílána simulovaná data.

Pro eliminaci zpoždění mezi modelem letové dynamiky a ArduPilot emulátorem je přijímací soket definován jako blokující. Simulace se tedy při každém vyčkávání na příjem dat z emulátoru na okamžik zastaví. Při každé aktualizaci FDM je jako odpověď odeslán paket s požadovaným nastavením aktuátorů. Frekvence aktualizace požadavku aktuátorů však může být nižší, a proto je třeba přebytečné pakety zahodit, aby nedocházelo ke čtení již zastaralých paketů. To je provedeno dočasným nastavením přijímacího soketu na neblokující a následným opakovaným čtením ze soketu. Po úspěšném vyčištění je soket opět nastaven na blokující a ve chvíli, kdy je očekáván akční zásah, proběhne jeho přečtení.

Simulace neprobíhá v reálném čase, protože její rychlost závisí na výkonu stroje, na kterém probíhá. V experimentech bylo dosaženo cca 2/3 rychlosti vůči reálnému času. Emulátor s dodanými daty nakládá tak, jako by referenčním časem pro výpočet akčních zásahů byl čas simulace, který je posílán jako jeden z parametrů.

4.5.5 Hardware-in-the-loop simulace

Vytvořený model letové dynamiky lze teoreticky bez nutnosti dodatečných úprav použít pro HIL simulaci, podobným způsobem jako pro SITL, protože je využíváno existujícího rozhraní pro simulátor Gazebo na kterém lze HIL simulaci zprovoznit.

V současné době však již řídicí jednotka Pixhawk nepodporuje hardware-in-the-loop simulace v pravém smyslu. Data z FDM jsou do řídicí jednotky posílána přes USB, stejně jako zásahy řízení zpět do simulace. Tato data jsou zpracována stejným způsobem jako v simulaci SITL - je tedy přidán šum a nasimulovány výstupní hodnoty virtuálních senzorů. HIL simulace pak tedy slouží pouze k ověření, zda výpočetní výkon jednotky postačuje na veškeré požadavky.

Výpočetní výkon jednotky Pixhawk však několikanásobně převyšuje maximální požadavky a proto se již v poslední době od HIL simulací upouští. V rámci této práce nemusel být upravován firmware autopilotu, ale používá se běžný firmware ArduPlane, který již byl pečlivě otestován stovkami uživatelů z řad komunity ArduPilot. Jelikož navíc simulace prozatím neběží v reálném čase ale pouze 2/3 rychlostí, nepřinesla by HIL simulace relevantní výsledky. Z těchto důvodů nebyla v rámci této práce hardware-in-the-loop simulace provedena.

4.5.6 Ovládání

Simulovaný letoun je ovládán příkazy z libovolného softwaru GCS, umožňujícího komunikaci dle standardu MAVLink a připojení pomocí TCP či UDP protokolu. Primární GCS je k emulátoru ArduPilot SITL připojována pomocí TCP protokolu na lokální síti (na adrese IP 127.0.0.1 a portu 5560). Touto stanicí může být MAVProxy, poskytující kromě rozhraní pro další případné stanice také další možnosti konfigurace. Preferovanou GCS pak lze připojit pomocí protokolu UDP (běžně na portu 14550, ale může být i na jiném). Druhým způsobem je připojení preferované GCS přímo k emulátoru ArduPilot SITL. V takovém případě musí GCS podporovat připojení pomocí protokolu TCP (například Mission planner nebo APM planner 2).

Díky tomu může být simulovaný letoun ovládán stejným softwarem GCS, jako letoun skutečný, se všemi možnostmi které poskytuje i skutečná řídicí jednotka ve spojení se software GCS, včetně plánování a spouštění misí. Některé GCS navíc umožňují ovládání externím ovladačem. Díky tomu lze simulovaný letoun ovládat například RC soupravou v manuálních režimech.

4.5.7 Konfigurace a ladění

Konfigurace řízení simulovaného letounu pak probíhá stejně jako konfigurace toho skutečného - tedy úpravou konfiguračních parametrů autopilotu pomocí softwaru GCS. Těmito parametry je konfigurována veškerá funkčnost včetně nastavení druhu stroje, funkcí výstupů, či regulačních konstant. V případě hybridní VTOL konstrukce je nutné povolit tento režim a určit výstupy, ovládající případný sklopný mechanismus.

Upravené parametry jsou běžně ukládány do paměti eeprom obsažené v řídicí jednotce. Emulovaná jednotka místo této paměti využívá soubor *eeprom.bin*. Díky tomu není nutné po opětovném spuštění simulace znovu načítat parametry ani uložené mise.

Řízení jednotky ArduPilot obsahuje PID regulátor pro všechny osy v režimu multikoptéry a pro osy roll a pitch v případě letadla. Řízení otáčení podle osy yaw je pro letoun komplikovanější, protože vyžaduje rotaci i podle zbylých os. Ladění PID regulátorů probíhalo podle běžných postupů popsaných například v [18].

4.6 Systémové požadavky a postup zprovoznění

Pro správné fungování simulace je zapotřebí nainstalovaný Matlab & Simulink ve verzi alespoň R2017b. Pro vytvoření software-in-the-loop simulace je potřeba distribuce Pythonu verze 3 (testování probíhalo na verzi 3.6), obsahující knihovny Struct a Socket. V Matlabu je potřeba provést konfiguraci Pythonu. Správnost konfigurace je možné ověřit Matlab příkazem `pyversion`, který vrátí aktuální verzi Pythonu nakonfigurovanou v Matlabu.

Pro zprovoznění simulace je zapotřebí provést tyto kroky:

1. Provést inicializační skript v Matlabu. Ten vytvoří všechny proměnné, charakteristické pro daný letoun a prostředí a zároveň slouží pro konfiguraci rozhraní mezi FDM a emulátorem ArduPilot SITL.

2. Spustit simulátor FlightGear dle specifikací v kapitole 4.4. Pro jednodušší spuštění s přednastavenými parametry byl vytvořen spouštěcí skript *runfg.bat*.

3. Spustit instanci emulátoru ArduPilot SITL. To lze provést buď přiloženým skriptem pro MS Windows *runsitl.bat*, který spustí instanci ArduPilotu, nebo provedením příkazu `sim\_vehicle.py --model gazebo`, který navíc spustí GCS MAVProxy. Tento příkaz lze spustit v repositáři projektu ArduPilot. Bude tak možné pracovat i s novějšími verzemi autopilotu, než které byly v době vzniku této práce. Tento příkaz však vyžaduje sestavení staženého firmwaru a nainstalovaný software MAVProxy. Vše je detailněji popsáno v [3].

4. Spustit simulaci v prostředí Simulink

5. Spustit GCS a připojit ji buď k instanci emulátoru ArduPilot SITL pomocí protokolu TCP nebo k MAVProxy pomocí UDP.

Pro ukončení simulace je potřeba nejprve zastavit Simulink, jinak absence přijímaných paketů z ArduPilot SITL, na které simulace vyčkává, způsobí zablokování celého Matlabu. Následně je možné zavřít všechna ostatní okna.

Kapitola 5

Experimenty

Pro ověření správnosti návrhu modelu, funkce spojení s ArduPilot SITL a konfiguraci řídících parametrů, byla vytvořená simulace podrobena sérii experimentů.

5.1 Ověření modelu letové dynamiky

Nejprve byla ověřena správnost návrhu simulace s experimenty bez zásahu řízení, kdy byly modelu nastaveny různé počáteční podmínky a byla pozorována jejich odezva. Díky těmto experimentům bylo odhaleno několik chyb v modelu letové dynamiky a v inicializačních parametrech. Díky úplnému vyřazení řízení lze dobře pozorovat přirozené chování letounu a odhalit tak závažné nedostatky modelu.

Mezi tyto experimenty patřilo zejména pozorování chování simulovaného letounu při volném pádu, s různými počátečními rychlostmi a orientací, což sloužilo ke srovnávání projevu modelu s teoretickým projevem v dané situaci.

5.2 Ověření ovládacích prvků a funkčnosti SITL

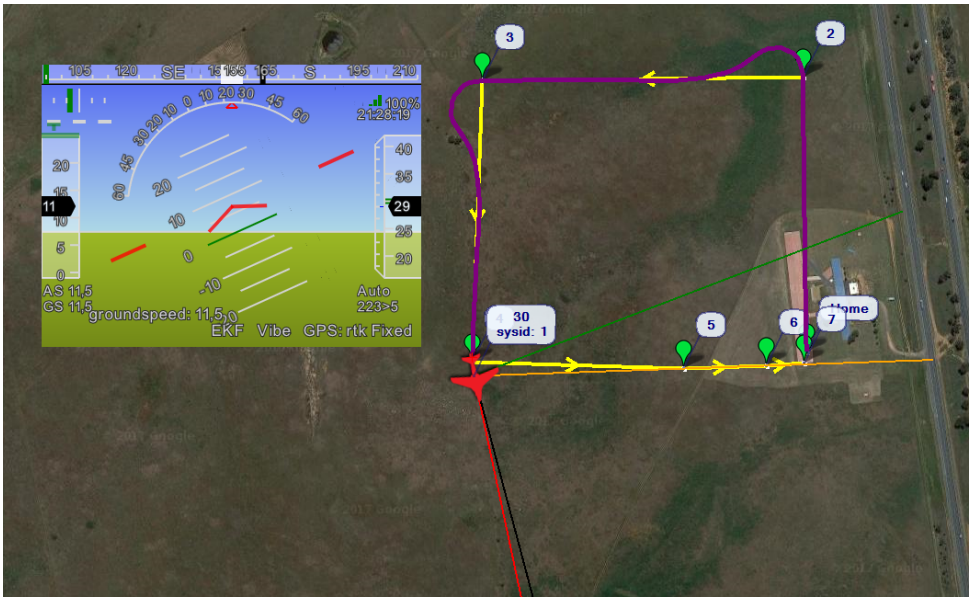
Mezi základní experimenty dále patřil let v plně manuálním režimu pro vyzkoušení základních letových vlastností. Ten probíhal přímým řízením aktuátorů letounu RC soupravou, připojenou k PC. Příkazy z RC soupravy byly při těchto experimentech pomocí software GCS Mission Planner posílány ve formě zpráv protokolu MAVLink do emulátoru řídicí jednotky, připojené k Simulinku výše uvedeným způsobem. V první řadě tyto experimenty sloužily pro testování a ladění spojení modelu letové dynamiky s ArduPilot SITL a za další sloužily pro ověření funkce aktuátorů a jejich ladění.

V těchto experimentech byl simulovaný letoun podroben základním leteckým manévrům pro ověření funkčnosti všech ovládacích ploch a pohonu. Mezi ty patří klopení, klonění a stáčení. Simulovaný letoun byl podroben také experimentům, v nichž bylo pozorováno chování při ztrátě vztlaku.

5.3 Ladění parametrů řízení

Dalším krokem bylo ověření stabilizace emulovanou řídicí jednotkou v režimu visení. Nejprve tedy bylo otestováno řízení v letovém módu *Q_STABILIZE*, v němž je letoun stále ovládán příkazy z RC soupravy, ale je stabilizována jeho orientace v prostoru na základě akčního zásahu, vypočítaného PID regulátory. Jejich parametry byly nastaveny podle [18]. Podobným způsobem proběhlo ladění a otestování stabilizačních režimů vodorovného letu v módu *STABILIZE*.

Dále byly odladěny konstanty pro let v módu *Q_LOITER*, v němž je stabilizována i poloha vůči zemi ve všech třech osách. Při středových polohách ovládacích pák RC soupravy by letoun v tomto režimu měl zůstat na místě bez ohledu na vnější vlivy. Po odladění tohoto režimu již lze otestovat autonomní mód *AUTO*. V tomto režimu je definována posloupnost akcí, které autopilot musí při dané misi provést. Povely těchto akcí jsou vždy definovány dle protokolu MAV Link. Tyto akce mohou, ale nemusí mít definovanou polohu. Nejčastěji používané povely jsou *Q_TAKEOFF* pro vzlet a *Q_LAND* pro přistání v režimu multikoptéry, a příkaz *WAYPOINT* určující bod, v jehož těsném okolí letoun musí proletět. Při tomto povelu dojde automaticky k přepnutí do vodorovného letu. Mise může obsahovat mnoho dalších povelů, umožňujících například běžný vzlet a přistání s definovatelným stoupáním a klesáním, přepnutí mezi horizontálním a vertikálním režimem letu, zavření, či otevření podvozku a podobně. Příklad průběhu naplánované mise v GCS Mission Planner s povely pro vertikální vzlet, pěti průletnými body a vertikální přistání je na obrázku 5.1.



Obrázek 5.1: Zálet plně autonomní mise v prostředí GCS Mission Planner s využitím ArduPilot SITL

Kapitola 6

Závěr

V první části byla provedena rešerše dostupných řídicích jednotek a autopilotů pro malé bezpilotní letouny, s podporou pro hybridní prostředky s možností kolmého startu i přistání a vodorovného letu ve vztlakovém režimu. Byla také provedena rešerše běžně používaných typů těchto hybridních prostředků. V této části jsou také popsány výhody a nevýhody konvenčních i hybridních koncepcí.

Hlavním cílem této práce bylo vytvoření matematického modelu hybridního letounu s kolmým startem a přistáním. Implementace tohoto modelu proběhla v prostředí Matlab & Simulink s využitím některých komponent z rozšíření Aerospace Toolbox. Tento model byl navržen tak, aby byl schopen pracovat i s vysokými úhly náběhu a vybočení (až $\pm 180°$), ke kterým může běžně docházet při vertikálním režimu letu a přechodových fázích. Zároveň model umožňuje libovolné spojité otočení sklopného mechanismu motorů, právě pro simulaci přechodových fází.

Tento model je převážně parametrický a dovoluje tak snadné přepracování pro potřeby simulace zcela jiného letounu. V rámci této práce však byl vytvořen model konkrétního letounu, který je stále ve fázi návrhu, a proto musely být jednotlivé parametry buď vypočítány (například část aerodynamických koeficientů pomocí software XFLR5) nebo odhadnuty na základě teoretických znalostí. Pro nejdůležitější koeficienty, mezi které patří například moment setrvačnosti, byly popsány experimenty vhodné k jejich identifikaci.

Oproti původním plánům nebyla vytvořena vlastní implementace řízení v rámci modelu, simulující řídicí algoritmy plánovaného autopilotu. Místo toho byly tyto řídicí algoritmy využity přímo, čehož bylo dosaženo vytvořením rozhraní pro software-in-the-loop simulaci mezi prostředím Matlab & Simulink a ArduPilot SITL. Díky tomu může být v simulaci využíváno totožných algoritmů řízení, jako v případě skutečného prostředku, což vede k větší věrohodnosti simulace například pro potřeby ověření správnosti naladění parametrů řízení.

Pro software-in-the-loop simulaci je na straně ArduPilot SITL využíváno již existující rozhraní pro fyzikální simulátor Gazebo. Díky tomu je zajištěna použitelnost implementace simulace na straně Matlab & Simulink i pro budoucí verze firmware ArduPilot.

Pro trojrozměrnou vizualizaci letového projevu modelu je využíván letecký

simulátor FlightGear.

V rešerši bylo zjištěno, že běžně používané autopiloty nepodporují plnou hardware-in-the-loop simulaci, kdy je testováno celé zařízení, včetně jeho komunikačních sběrnic. Tyto autopiloty podporují HIL simulace pouze pro ověření hardwarových prostředků výpočetní jednotky. Jelikož však firmware autopilotu nebyl nijak upravován, není potřeba jej ověřovat. Nehledě na to, vytvořená soustava s modelem v prostředí Matlab & Simulink prozatím negarantuje běh v reálném čase a tudíž by HIL simulace nepřinesla žádné směrodatné výsledky. Z těchto důvodů nebyla oproti původním plánům hardware-in-the-loop simulace provedena.

Na závěr bylo provedeno ladění řízení autopilotu ArduPilot - ArduPlane pro konkrétní simulovaný prostředek. Toto ladění probíhalo experimentálně, sledováním odezvy při různých letových režimech, včetně přechodových fází. Na závěr byly provedeny experimenty v plně autonomním režimu. Díky software-in-the-loop simulaci lze všechny parametry takto naladěného virtuálního prostředku beze změny přenést do skutečné řídicí jednotky.

Hlavním přínosem této práce je implementace rozhraní mezi prostředím Matlab & Simulink a emulátorem řídicí jednotky ArduPilot SITL. I přes poptávku mezi členy komunity ArduPilot nebylo totiž dosud publikováno žádné rozhraní, umožňující vytvoření software-in-the-loop simulaci v prostředí Matlab & Simulink ve spojení s firmware ArduPilot. Proto bude zdrojový kód, implementující toto spojení, v nejbližší době publikován na webové službě GitHub, kde bude veřejně dostupný.

Nedostatkem implementovaného rozhraní je, že komunikace neprobíhá dostatečně rychle a proto je celá simulace zpomalena cca na polovinu rychlosti oproti reálnému času. Pro účely hardware-in-the-loop simulace by však bylo vhodné dosáhnout toho, aby simulace běžela v reálném čase. V dalším postupu by tedy bylo vhodné přepsat komunikační funkce, které jsou v současnosti napsané v programovacím jazyce Python, do tzv. s-funkcí, které mohou být implementovány v programovacím jazyce C/C++ a umožňují efektivnější běh celé simulace. Zároveň by bylo vhodné provést rozsáhlejší aerodynamickou analýzu (například ve větrném tunelu) a další experimenty po sestavení reálného letounu, pro lepší identifikaci jeho klíčových parametrů.

Reference

[1] *AirRails*. UAVenture AG. [online], dostupné z: http://uaventure.com/ (navštíveno: 22. 7. 2017).

[2] *Amazon Prime Air*. Amazon.com, Inc. [online], dostupné z: https://www.amazon.com/Amazon-Prime-Air/ (navštíveno: 14. 9. 2017).

[3] *ArduPilot Autopilot Suite*. ArduPilot. [online], dostupné z: http://ardupilot.org/ (navštíveno: 14. 9. 2017).

[4] A. Babushkin. *jMAVSim*. GitHub, Inc. [online], dostupné z: https://github.com/DrTon/jMAVSim (navštíveno: 20. 10. 2017).

[5] S. Baluta. *How much power is needed to hover?* Starlino Electronics. [online], dostupné z: http://www.starlino.com/power2thrust.html (navštíveno: 14. 9. 2017).

[6] *BeagleBone Blue*. BeagleBoard.org Foundation. [online], dostupné z: https://beagleboard.org/blue/ (navštíveno: 1. 8. 2017).

[7] J. B. Brandt a M. S. Selig. "Propeller Performance Data at Low Reynolds Numbers". In: *University of Illinois at Urbana-Champaign, Urbana* (2011). [online], dostupné z: https://m-selig.ae.illinois.edu/.

[8] *DJI Matrice*. DJI. [online], dostupné z: https://www.dji.com/matrice600/ (navštíveno: 20. 8. 2017).

[9] *DroneKit-Python's documentation*. 3D Robotics. [online], dostupné z: http://python.dronekit.io/ (navštíveno: 20. 10. 2017).

[10] *E-Flite Convergence VTOL*. Modelflight. [online], dostupné z: https://www.modelflight.com.au/ (navštíveno: 14. 9. 2017).

[11] *ECEF ENU Longitude Latitude relationships*. Wikimedia Commons. [online], dostupné z: http://commons.wikimedia.org/ (navštíveno: 10. 11. 2017).

[12] *FireFLY6 PRO*. Birds Eye View Aerobotics. [online], dostupné z: https://www.birdseyeview.aero/ (navštíveno: 12. 9. 2017).

[13] *FlightGear flight simulator*. FlightGear. [online], dostupné z: http://www.flightgear.org/ (navštíveno: 1. 9. 2017).

[14] *Gazebo*. Open Source Robotics Foundation. [online], dostupné z: http://gazebosim.org/ (navštíveno: 20. 10. 2017).

[15] M. Hromčík a P. Hospodář. "Systémy řízení letu - poznámky k přednáškám". In: *ČVUT v Praze, Fakulta elektrotechnická* (2017).

[16] *Intel Aero*. Intel Corporation. [online], dostupné z: https://software.intel.com/en-us/aero/ (navštíveno: 1. 8. 2017).

[17] *JSBSim open source flight dynamics model*. Slashdot Media. [online], dostupné z: http://jsbsim.sourceforge.net/ (navštíveno: 12. 10. 2017).

[18] O. Liang. *Quadcopter PID explained*. [online], dostupné z: https://oscarliang.com/quadcopter pid-explained-tuning/ (navštíveno: 1. 12. 2017).

[19] *LibrePilot Documentation*. LibrePilot. [online], dostupné z: https://www.librepilot.org/ (navštíveno: 20. 9. 2017).

[20] *Navio 2*. Emlid Ltd. [online], dostupné z: https://emlid.com/navio/ (navštíveno: 1. 8. 2017).

[21] *PX4*. [online], dostupné z: http://px4.io/ (navštíveno: 22. 7. 2017).

[22] *Qualcomm Snapdragon*. Intrinsyc Technologies Corp. [online], dostupné z: https://www.intrinsyc.com/vertical-development-platforms/qualcomm-snapdragon-flight/ (navštíveno: 1. 8. 2017).

[23] R. Ritz. *Tailsitter flying robot hovers and recovers easily thanks to new algorithm*. RoboHub. 2015. [online], dostupné z: http://robohub.org/.

[24] M. S. Selig. "Modeling Full-Envelope Aerodynamics of Small UAVs in Realtime". In: *University of Illinois at Urbana-Champaign, Urbana* (2010), s. 8–14. [online], dostupné z: https://m-selig.ae.illinois.edu/.

[25] M. Šebek. "Automatické řízení - poznámky k přednáškám". In: *ČVUT v Praze, Fakulta elektrotechnická* (2016).

[26] *T-Motor products*. T-Motor. [online], dostupné z: http://tmotor.com/ (navštíveno: 5. 9. 2017).

Příloha A

Význam zkratek

DoF Degree of Freedom - Stupeň volnosti

ESC Electronic Speed Controller - elektronický regulátor otáček

FDM Flight Dynamics Model - model dynamiky letu

GCS Ground Control Station - pozemní ovládací stanice

GNSS Global Navigation Satelite System - Globální družicový polohový systém

HIL Hardware-In-The-Loop

IMU Inertial Meassurement Unit - jednotka zodpovědná za měření působících sil, obsahující akcelerometr, gyroskopický senzor a magnetometr

MAVLink Micro Air Vehicle Link - komunikační protokol, používaný řídicími jednotkami

PWM Pulse Wide Modulation - pulsně šířková modulace signálu

SITL Software-In-The-Loop

TCP Transmission Control Protocol

UART Universal Asynchronous Receiver-Transmitter - hardware pro asynchronní sériovou komunikaci

UAV Unmanned Aerial Vehicle - bezpilotní letoun

UDP User Datagram Protocol

VTOL Vertical Take-Off and Landing - letoun schopný vodorovného letu ve vztlakovém režimu i kolmého startu a přistání

Příloha B

Obsah CD

```
/kořenový adresář
├── matlab
│   ├── sitl - kořenová složka emulátoru obsahující spustitelné soubory
│   │          ArduPilot SITL a knihovny pro MS Windows
│   ├── init.m - inicializační soubor simulace
│   ├── nonlin_sim.slx - simulační model v prostředí Simulink
│   ├── runfg.bat - skript pro spuštění a konfiguraci FlightGear
│   ├── runsitl.bat - skript pro spuštění a konfiguraci ArduPilot SITL
│   ├── simReceive.m - funkce pro přijetí paketu řízení
│   └── simSend.m - funkce pro odeslání paketu FDM
├── xflr5
│   ├── polars - výsledky aerodynamické analýzy
│   └── vtol.xfl - soubor vytvořených modelů
├── text - zdrojové kódy textu práce a obrázky
└── kralmar3.pdf - text práce
```