



**FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI**

**KATEDRA
KYBERNETIKY**

Bakalářská práce

Sledování trajektorie dronem

Jan Vrba



**FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI**

**KATEDRA
KYBERNETIKY**

Bakalářská práce

Sledování trajektorie dronem

Jan Vrba

Vedoucí práce

Ing. Zdeněk Bouček

© Jan Vrba, 2023.

Všechna práva vyhrazena. Žádná část tohoto dokumentu nesmí být reprodukována ani rozšiřována jakoukoli formou, elektronicky či mechanicky, fotokopírováním, nahráváním nebo jiným způsobem, nebo uložena v systému pro ukládání a vyhledávání informací bez písemného souhlasu držitelů autorských práv.

Citace v seznamu literatury:

VRBA, Jan. *Sledování trajektorie dronem*. Plzeň, 2023. Bakalářská práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra kybernetiky. Vedoucí práce Ing. Zdeněk Bouček.

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jan VRBA**
Osobní číslo: **A19B0333P**
Studijní program: **B0714A150005 Kybernetika a řídicí technika**
Specializace: **Automatické řízení a robotika**
Téma práce: **Sledování trajektorie dronem**
Zadávací katedra: **Katedra kybernetiky**

Zásady pro vypracování

- Seznámení se s chováním kvadrotorového dronu.
- Seznámení se s problémem sledování trajektorie.
- Návrh řídicího algoritmu pro sledování trajektorie.
- Otestování algoritmu v simulačním prostředí.



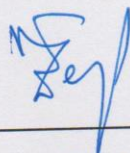
Rozsah bakalářské práce: **30-40 stránek A4**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

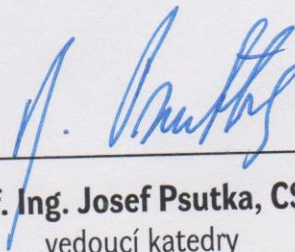
- Kamel, M., Burri, M., & Siegwart, R. (2017). Linear vs Nonlinear MPC for Trajectory Tracking Applied to Rotary Wing Micro Aerial Vehicles. *IFAC-PapersOnLine*, 50(1), 3463–3469.
- Nguyen, H., Kamel, M., Alexis, K., & Siegwart, R. (2021). Model Predictive Control for Micro Aerial Vehicles: A Survey. *2021 European Control Conference, ECC 2021*, 1556–1563.
- Borrelli, F., Bemporad, A., & Morari, M. (2017). *Predictive Control for Linear and Hybrid Systems* (draft Febr). Cambridge University Press.
- Martin Klaučo and Michal Kvasnica. MPC-Based Reference Governors. *Advances in Industrial Control*. Springer International Publishing, Cham, 2019.

Vedoucí bakalářské práce: **Ing. Zdeněk Bouček**
Výzkumný program 1

Datum zadání bakalářské práce: **17. října 2022**
Termín odevzdání bakalářské práce: **22. května 2023**



Doc. Ing. Miloš Železný, Ph.D.
děkan



Prof. Ing. Josef Pšutka, CSc.
vedoucí katedry

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného akademického titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Západočeská univerzita v Plzni má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Plzni dne 10. srpna 2023

.....

Jan Vrba

V textu jsou použity názvy produktů, technologií, služeb, aplikací, společností apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Abstrakt

Bakalářská práce popisuje problém sledování trajektorie dronem. Dron, kterého trajektorie byla řízena, je multikoptéra o čtyřech vrtulích, jinak řečeno kvadrokoptéra. Model Predictive Control (MPC), které kvadrokoptéru řídí, našel optimální strategii řízení pro dvě definované trajektorie. MPC a kvadrokoptéra byly implementovány a simulovány v aplikaci Matlab/Simulink.

Dokument popisuje vytvoření nelineárního regulátoru MPC na základě modelu dynamického systému dronu. MPC byl následně otestován při sledování referenčních trajektorií. Funkčnost sledování trajektorie pomocí MPC byla ověřena v simulaci za ideálních podmínek, ale i při působení poruchy. V závěru byly okomentovány výsledky simulací. Dodatek pojednává o způsobu ovládání letu kvadrokoptéry v simulátoru PX4/Gazebo.

Abstract

The bachelor's thesis addresses the problem of trajectory tracking by a drone. The drone, whose trajectory was controlled, is a quadcopter, a multicopter with four rotors. Model Predictive Control (MPC), which governs the quadcopter, found an optimal control strategy for two predefined trajectories. Both MPC and the quadcopter were implemented and simulated using the Matlab/Simulink application.

The document describes the development of a nonlinear MPC controller based on the dynamic model of the drone's system. The MPC was subsequently tested for tracking reference trajectories. The functionality of trajectory tracking using MPC was verified in simulations under ideal conditions and in the presence of disturbances. In conclusion, the simulation results are discussed. The appendix discusses the method of flight control for the quadcopter in the PX4/Gazebo simulator.

Klíčová slova

Sledování trajektorie • Kvadrokoptéra • Dron • Prediktivní řízení • Trajectory tracking • Quadcopter • Drone • Model predictive control

Poděkování

Na tomto místě bych rád poděkoval svému vedoucímu, panu Ing. Zdeňku Boučkovi, za jeho trpělivost s tempem vytváření této práce, za jeho přínosné poznatky a pomoc při řešení problémů týkajících se této Bakalářské práce. Také bych rád poděkoval všem profesorům a učitelům, jejichž předměty jsem absolvoval a které mi pomohly k lepšímu pochopení a uchopení této Bakalářské práce.

Obsah

1	Úvod	3
2	Kvadroptéra	5
2.1	Konstrukce s více rotory	5
2.2	Systém řízení letu	6
2.3	Aerodynamická stabilita	8
2.4	Vstupy do řízení	8
3	Návrh modelu kvadroptéry	11
3.1	Souřadnicový systém	11
3.1.1	Souřadnicový systém s pevnou vazbou v zemi	11
3.1.2	Souřadnicový systém s pevnou vazbou na dron	12
3.2	Orientace kvadroptéry	13
3.3	Model kvadroptéry	14
3.3.1	Model efektivity řízení	16
3.3.2	Kinematický model tuhého tělesa	18
3.3.3	Dynamický model tuhého tělesa	18
4	Sledování trajektorie kvadroptéry pomocí MPC	21
4.1	Sledování trajektorie pomocí MPC	21
4.2	Implementace MPC	24
4.2.1	Omezení	27
5	Implementace a výsledky	29
5.1	Implementace v Matlab/Simulinku	29
5.1.1	Matlab System	30
5.2	MPC výsledky	32
5.2.1	Trajektorie ve tvaru lemniskáty	32
5.2.2	Trajektorie ve tvaru spirály	36
6	Závěr	41

A Simulace dronu v PX4/Gazebo	43
A.1 Sledování trajektorie pomocí PID regulátorů	43
A.2 Simulace kvadrokoptéry v Gazebo	45
A.2.1 Letecké režimy	47
A.2.2 Hlavní algoritmus	47
A.3 Výsledky simulátoru Gazebo	48
A.4 Závěr dodatku	50
Bibliografie	51
Seznam obrázků	53

Svět bezpilotního letectví zaznamenává stále obrovský rozmach. Jedno z klíčových témat v oblasti autonomního letectví je sledování trajektorie dronem, jelikož přímo souvisí se stabilitou letu a efektivním pohybem dronu ve vzduchu.

Tato bakalářská práce se zabývá především návrhem prediktivního regulátoru (MPC – Model Predictive Control), který dokáže generovat akční zásahy tak, aby dron mohl hladce sledovat předem stanovenou trajektorii dronu. Základem MPC je již podle názvu model systému, v tomto případě systém dronu.

MPC zvládá generovat akční zásahy i s ohledem na omezení rozsahu. Dodržení omezení je důležité z několika důvodů. Dron má fyzická omezení daná konstrukcí dronu a rotory, jako například maximální otáčky rotorů. Dronu se dají zavést i bezpečnostní úhly. Při naklánění nesmí dron přesáhnout hraniční úhel, aby nehrozilo překlopení. Samotná úloha sledování trajektorie může mít omezení, která chceme dodržovat, jako například rychlost, jakou má dron létat. Hlavní výhoda MPC spočívá v tom, že dokáže předpovídat na základě modelu chování dronu na krátkém horizontu, proto je MPC vhodné pro úlohu sledování trajektorie. V práci využijeme nelineární model dronu diskretní v čase.

V kapitole 2 vysvětlíme základní informace o multikoptérech a jejich řízení. Popíšeme jejich konstrukci, základní systémy, typy řízení a způsob stabilizace. Následně v kapitole 3 navrhne model kvadroptéry. Popíšeme použité souřadnicové systémy a orientaci dronu. V kapitole 4 budeme popisovat metodu sledování trajektorie pomocí MPC. Popíšeme konkrétní použité vztahy, parametry a omezení pro řízení. V kapitole 5 se věnujeme popisu implementace MPC v Matlabu a prezentování výsledků průletu dvou trajektorií. Rozebereme rozdíly mezi referenční a řízenou trajektorií s poruchou a bez ní. V závěru 6 zhodnotíme výsledky a splnění cílů této bakalářské práce.

V dodatku této práce A pojednáme o simulaci letu v simulátoru PX4/Gazebo. Simulátor Gazebo podporuje simulaci různých typů autonomních vozidel. Prozkoumáme možnosti simulátoru, režim řízení a regulátor pro sledování trajektorie obsažený v PX4. V sekci A.1 vysvětlíme metodu sledování trajektorie pomocí PID regulátorů, které řídí let dronu v simulátoru. V sekci A.2 popíšeme použité aplikace,

balíčky a rozhraní, použité letecké režimy a implementaci algoritmu pro let v simulátoru PX4/Gazebo. V sekci A.3 Ukážeme let kvadrokoptéry v simulátoru a jak se liší od předchozích trajektorií v hlavní části této práce. V závěru dodatku A.4 zhodnotíme dosažený cíl a navrhne cíl možného budoucího pokračování na této práci.

V této kapitole uvedeme základní informace a principy týkající se kvadrokoptéry a jejího fungování. Dostatečný přehled o problematice fungování a jeho letu je důležitý při tvorbě matematického modelu dronu, který bude vytvořen v kapitole 3 a při návrhu řídicího algoritmu. Pro tento model je důležité vhodně definovat konstrukci kvadroptéry a konfiguraci rotorů. Popíšeme tedy, co znamená návrh dronu s více rotory, jaké jsou výhody takových konstrukcí a jaké konfigurace se nejčastěji využívají. Popíšeme systém řízení letu, který je součástí dronů se systémem PX4, jeho roli při letu dronu a jeho nejdůležitější aspekty. Vysvětlíme, co znamená aerodynamická stabilita a co přispívá ke stabilnímu letu multikoptéry. Nakonec se zmíníme o nejčastějších vstupech, které se používají pro řízení dronu.

2.1 Konstrukce s více rotory

Kvadrokoptéra je typem letounu, které je vybaven čtyřmi rotory umístěnými na jeho ramenech. Tato konstrukce se liší od tradičních letounů, která mají obvykle jedno nebo dvě pevná křídla, nebo vrtulníků s jedním hlavním rotorem. Jednorotorové vrtulníky se ovládají na principu naklánění lopatek a změnou otáček, zatímco multikoptéry mají fixní rotory a řídí se pouze změnou otáček.

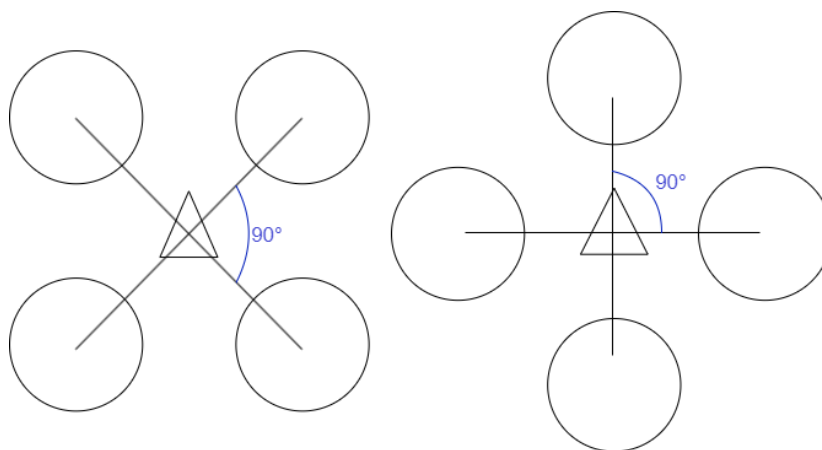
Existence čtyř rotorů u kvadrokoptéry přináší několik výhod. Jednou z nich je schopnost vertikálního vzletu a přistání. Rotory generují směrem nahoru sílu, známou jako tah (thrust), který vyrovnává váhu letadla a umožňuje mu stoupat do vzduchu a přistávat s lehkostí. Díky tomu může kvadrokoptéra vzlétnout z jakéhokoli místa a dokázat zůstat ve vzduchu na místě, aniž by potřebovala rozsáhlou vzletovou a přistávací dráhu.

Další výhodou konstrukce s více rotory je vyšší stabilita a manévrovatelnost. Díky čtyřem rotorům může kvadrokoptéra dosahovat rychlých a přesných změn směru pohybu. Ovládání jednotlivých rotorů umožňuje regulovat tah na jednotlivých stranách a tím měnit směr a náklon kvadrokoptéry. Tímto způsobem lze provádět různé manévry, jako je pohyb dopředu, dozadu, do stran a rotace kolem osy.

Návrh s více rotory také přispívá ke zvýšení redundance a bezpečnosti. Multi-koptéry s osmi nebo šesti rotory jsou schopny udržet stabilitu a létat i v případě selhání jednoho z rotorů. Ostatní rotory převzou zátěž a umožní letounu pokračovat v letu nebo provést bezpečné přistání. Kvadroptéra potřebuje speciální regulátor, který řídí zbylé funkční rotory [1].

Pro návrh s více rotory je důležité určit vhodné uspořádání rotorů [2]. Uspořádání rotorů znamená, jak jsou jednotlivé rotory rozloženy vzhledem ke středu dronu a k ostatním rotorům. Konfigurace rotorů je důležitá z různých důvodů. Například pokud je potřeba umístit na dron kameru, pokud chceme mít osy x a y zarovnané s rameny rotorů, nebo pokud má nějaký motor slabší výkon než ostatní. Nejčastější konfigurací je "+"-konfigurace, kde jsou čtyři rotory umístěny ve tvaru kříže ve stejných směrech, kam vedou kladné a záporné směry os x a y . Druhá nejčastější je "x"-konfigurace, kde jsou přední a zadní rotory diagonálně zarovnané a osa x leží mezi dvěma předními rotory.

Simulované kvadroptéry v této práci využívají "x"-konfigurace. Důvodem jsou směry os kvadroptéry. Pro rotaci podle os x a y má kvadroptéra k dispozici vždy dva přilehlé protichůdné rotory namísto jednoho v případě "+"-konfigurace. To může vést ke zvýšené pohyblivosti dronu. Protichůdné rotory znamenají, že rotory na jedné diagonále se točí po směru hodinových ručiček a rotory na druhé diagonále proti směru hodinových ručiček. Vizualní reprezentace konfigurací je na obrázku 2.1.



Obrázek 2.1: Konfigurace "x" a "+"

2.2 Systém řízení letu

Systém řízení letu (FCS – Flight Control System) udržuje stabilitu, ovladatelnost a kontrolu během letu. Sestává se z různých komponent a technologií, které zajišťují

přesné a rychlé ovládání dronu [3]. Nejdůležitější součásti systému řízení letu jsou:

- **Řídící jednotka:** Řídící jednotka přijímá vstupy od pilota nebo autonomního systému a vypočítává potřebné řídicí příkazy pro regulaci pohybu dronu. Dále také zpracovává data ze senzorů a spouští řídicí algoritmy pro stabilizaci kvadrokoptéry, úpravu rychlosti motorů a správu letových stavů, jako je výška, orientace a poloha.
- **Senzory:** Systém řízení letu spoléhá na různé senzory pro sběr aktuálních dat o stavu kvadrokoptéry a jejím okolí [4]. K běžně používaným senzorům patří inerciální měřicí jednotka (IMU), která měří lineární a úhlovou rychlost kvadrokoptéry, ale i rychlost rotace a magnetické pole, na jejichž základě lze odhadnout její orientaci a rychlost. Barometr měří atmosférický tlak a odhaduje výšku kvadrokoptéry nad úrovní moře nebo nad zemí. GPS přijímače poskytují informace o poloze a rychlosti, což umožňuje dronu udržovat polohu. Dron bývá také vybaven senzorem optického toku, který pomocí kamery sleduje zem nebo rysy pod kvadrokoptérou, umožňující přesné udržování polohy a ovládání výšky, zejména v nízkých výškách.
- **Elektronické regulátory otáček:** Každý motor kvadrokoptéry je připojen k elektronickému regulátoru otáček. Regulátor otáček převádí řídicí signály z řídicí jednotky na elektrické signály, které upravují rychlost a směr otáčení motoru. Řídící jednotka může pomocí nezávislého ovládání rychlosti motorů dosáhnout stability, ovládat náklon a provádět manévry.
- **Komunikační a řídicí protokoly:** Systém řízení letu využívá komunikačních protokolů k zajištění plynulé interakce mezi řídicí jednotkou, senzory a dalšími periferními zařízeními. Běžné protokoly zahrnují PWM (Pulse Width Modulation) a PPM (Pulse Position Modulation) pro přenos řídicích signálů mezi řídicí jednotkou a regulátory otáček. Kromě toho, sběrnice jako I2C (Inter-Integrated Circuit) a rozhraní SPI (Serial Peripheral Interface) umožňují komunikaci s dalšími senzory a moduly.
- **Řídící algoritmy a režimy:** Řídící jednotka využívá řídicí algoritmy pro interpretaci dat ze senzorů, výpočet řídicích příkazů a stabilizaci kvadrokoptéry. Tyto algoritmy zahrnují Proporcionálně-Integračně-Derivační (PID) regulaci, která upravuje rychlosti motorů na základě chyby mezi požadovanou (referenční) a skutečnou orientací, a algoritmy pro fúzi měření ze senzorů za účelem odhadu stavu. Systém řízení letu nabízí různé režimy letu, jako je manuální režim, udržování výšky, GPS-asistované režimy a autonomní režimy, což umožňuje různorodé ovládání v závislosti na požadavcích pilota a schopnostech dronu.

2.3 Aerodynamická stabilita

Aerodynamická stabilita znamená schopnost dronu udržovat požadovanou polohu a odolávat rušivým vlivům ve vzduchu. Dosáhnutí stabilního letu je zásadní pro bezpečný a ovladatelný let. Pro dosáhnutí aerodynamické stability je důležité mít správný design, rozložení hmotnosti, konfigurace rotorů a inteligentních řídicích systémů. Správná kombinace je důležitá pro stabilní let dronu, proto si zde vysvětlíme pojmy jako těžiště a střed vztlaku.

Těžiště je bod, ve kterém tíhová síla působí na kvadrokoptéru. Pro stabilní let by mělo být těžiště správně umístěno obvykle blízko geometrického středu kvadrokoptéry. Dobře vyvážené těžiště zajišťuje, že kvadrokoptéra reaguje předvídatelně na řídicí vstupy a udržuje požadovanou orientaci. V případě dronu, který jsme uvažovali v této bakalářské práci je těžiště v geometrickém středu.

Střed vztlaku je bod, ve kterém působí celková vztlaková síla generovaná rotory kvadrokoptéry. Střed vztlaku bývá zarovnán s těžištěm, což pomáhá dosáhnout stability. Když se dron naklání nebo se ohýbá, rozložení vztlaku mezi rotory se mění, což vytváří kroutící moment, který se snaží obnovit stabilitu kvadrokoptéry.

2.4 Vstupy do řízení

Pro ovládání kvadrokoptéry používáme různé řídicí vstupy, které umožňují pilotovi nebo autonomnímu systému ovládat pohyb a chování dronu. Hlavními vstupy pro řízení jsou:

- **Řízení rychlosti motorů:** Rychlost motorů ovlivňuje vztlak a tím i výšku a stabilitu kvadrokoptéry. Zvýšení rychlosti motorů zvyšuje vztlak a zvedá kvadrokoptéru výše, zatímco snížení rychlosti motorů snižuje vztlak a snižuje výšku. Rovnováha mezi rychlostmi motorů je důležitá pro udržování stability během letu.
- **Řízení náklonu:** Náklon kvadrokoptéry (Tilt control) je dosažen nerovnoměrným zvýšením nebo snížením rychlosti motorů na jednotlivých stranách. Zvýšení rychlosti motorů na jedné straně způsobí naklonění kvadrokoptéry směrem k této straně. Tento mechanismus umožňuje změnu směru a otočení dronu.
- **Řízení otáčení:** Otáčení kvadrokoptéry (Yaw control) kolem svislé osy z je dosaženo změnou rychlosti motorů na protilehlých stranách. Zvýšení rychlosti motoru na jedné straně a současné snížení rychlosti na protilehlé straně způsobuje otáčení kvadrokoptéry.

- **Řízení výšky:** Úprava celkového tahu generovaného rotory umožňuje kvadrokoptěre stoupat nebo klesat. Zvýšení tahu způsobuje stoupání, zatímco snížení tahu způsobuje klesání.

Návrh modelu kvadrokoptéry

3

V této části popíšeme souřadnicové systémy a definujeme orientaci kvadrokoptéry. Určíme, jakým způsobem orientaci určujeme a jak z ní vypočítáme rotační matici pro přechod z jednoho souřadnicového systému do druhého. Nakonec popíšeme hlavní části modelování multikoptéry, určíme předpoklady, které budeme při modelování dodržovat a navrhne model kvadrokoptéry. Kapitola 3 je popsána na základě [3].

3.1 Souřadnicový systém

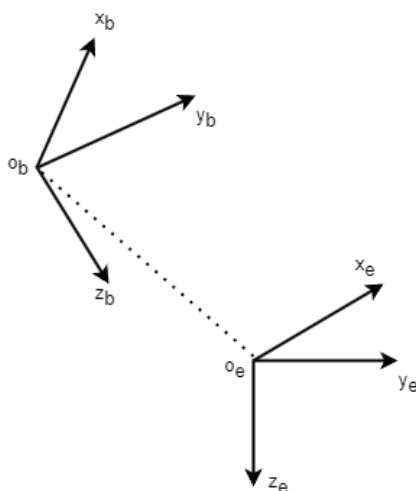
Kvůli popisu pohybu a modelování kvadrokoptéry nejprve zavedeme souřadnicové systémy. Pohyb dronu je popsán dvěma souřadnicovými systémy. Jedním je fixní systém s pevnou vazbou na zem, označovaný E . Druhým je fixní systém s pevnou vazbou na dron, označovaný B . S pomocí těchto souřadnicových systémů můžeme naměřené nebo vypočítané parametry jednoho systému zobrazit v souřadnicích druhého systému.

3.1.1 Souřadnicový systém s pevnou vazbou v zemi

Earth-Fixed Coordinate Frame (EFCF) je souřadnicový systém používán pro popisování dynamických stavů kvadrokoptéry relativních k zemskému povrchu a pro určení pozice ve třídímním prostoru. Systém je postaven tak, že nemusíme přímo reflektovat zakřivení zemského povrchu. Jako počátek souřadnicového systému se nejčastěji užívá počáteční pozice dronu. Osa x_e směřuje horizontálním směrem nejčastěji na sever, osa z_e směřuje vertikálně směrem do země a osa y_e je určena podle pravidla pravé ruky. Například v případě, že osa x_e směřuje na sever a osa z_e do země, potom bude osa y_e směřovat na východ. Tomuto rozložení se říká NED (north-east-down).

3.1.2 Souřadnicový systém s pevnou vazbou na dron

Aircraft-Body Coordinate Frame (ABCF) má počátek o_b v místě težiště kvadrokoptéry. Osa x_b směřuje dopředu ve směru natočení dronu. Osa z_b směřuje kolmo od osy x_b směrem dolů pod dron. Osu y_e určíme podle pravidla pravé ruky. Vztah mezi ABCF a EFCF je zobrazen na obrázku 3.1.



Obrázek 3.1: Souřadnicové systémy

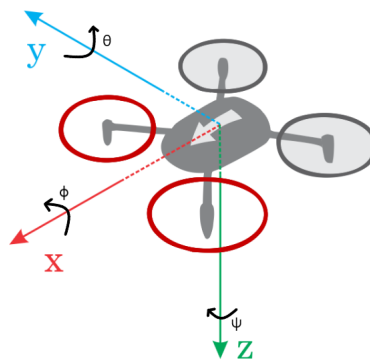
Definujeme také jednotkové vektory pro jednotlivé osy EFCF

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (3.1)$$

Pro ABCF jednotkové vektory jednotlivých os splňují následující vztah

$${}^b b_1 = e_1, \quad {}^b b_2 = e_2, \quad {}^b b_3 = e_3.$$

Na obrázku 3.2 vidíme, jakým směrem od kvadrokoptéry směřují všechny osy a jakou má konfiguraci rotorů.



Obrázek 3.2: ABCF souřadnicový systém, zdroj [5]

3.2 Orientace kvadroptéry

Tato sekce vysvětluje popis orientace a natočení kvadroptéry pomocí Eulerových úhlů a jejich spojení s maticemi rotace.

Popis pomocí Eulerových úhlů je hojně využívaný popis orientace dronu, který často bývá používán i při řízení. Na základě Eulerova teoremu lze rotaci tuhého tělesa kolem jednoho pevného bodu chápat jako složení několika konečných rotací kolem tohoto pevného bodu. V tomto případě to znamená, že můžeme dosáhnout souřadnicového systému ABCF pomocí třech elementárních rotací souřadného systému EFCF kolem jednoho pevného bodu. Proto může být finální rotační matice reprezentována jako součin tří elementárních rotačních matic

$${}^b R = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi). \quad (3.2)$$

- **Roll** (ϕ) představuje úhel natočení kolem osy x . Rozmezí možných hodnot je interval $[-\pi, \pi]$.
- **Pitch** (θ) představuje úhel natočení kolem osy y . Rozmezí možných hodnot je interval $[-\pi, \pi]$.
- **Yaw** (ψ) představuje úhel natočení kolem osy z . Rozmezí možných hodnot je interval $[-\pi, \pi]$.

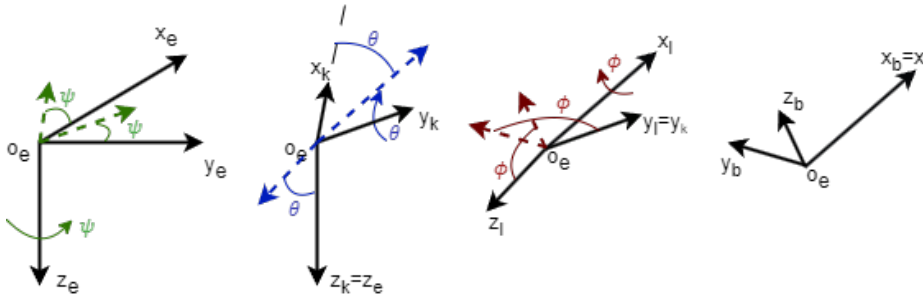
Jednotlivé elementární rotační matice podle každé osy jsou

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.3)$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \quad (3.4)$$

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix}. \quad (3.5)$$

Následující obrázek 3.3 ukazuje přechod z EFCF do ABCF pomocí postupných rotací os podle Eulerových úhlů.



Obrázek 3.3: Přechod z EFCF do ABCF

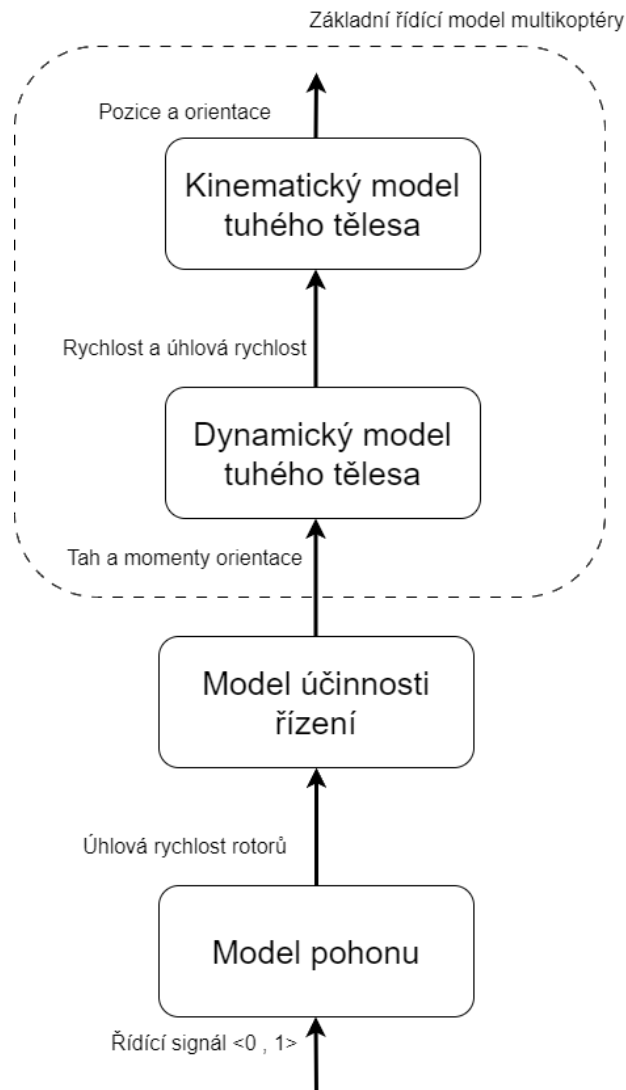
Pokud je úhlová rychlost těla dronu označena jako $\omega = [{}^b\omega_x \ {}^b\omega_y \ {}^b\omega_z]^T$, poté vztah mezi úhlovou rychlostí těla dronu a úhlovou rychlostí jednotlivých os vyjádříme jako

$${}^b\omega = \dot{\psi} \cdot {}^b z_e + \dot{\theta} \cdot {}^b y_k + \dot{\phi} \cdot {}^b x_l. \quad (3.6)$$

3.3 Model kvadrokoptéry

V této sekci si popíšeme čtyři hlavní části, ze kterých se skládá model kvadrokoptéry a samotný postup pro vytvoření modelu pro účely sledování trajektorie. Model jsme využili následně v sekci 4 při vytváření prediktivního regulátoru a v sekci 5 při implementaci systému a regulátoru do Matlabu. Popsanou architekturu hlavních částí můžeme vidět na obrázku 3.4.

Zaměříme se na modelování modelu efektivity řízení, kinematického a dynamického modelu. pro modelování není uvažován model pohonu, který by v realitě generoval tah a točivé momenty rotorů. Kinematický a dynamický model dohromady tvoří základní model chování multikoptér.

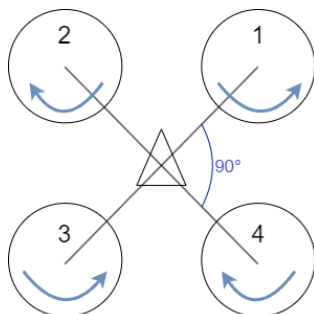


Obrázek 3.4: Architektura modelování multikoptéry

Před modelováním si určíme **předpoklady**, které budeme při modelování dodržovat.

- Kvadroptéra má tuhé tělo
- Hmotnost a momenty inercie jsou konstantní
- Těžiště kvadroptéry a její geometrický střed jsou v jednom bodě (Dron je symetrický).
- Na kvadroptéru působí pouze gravitace a tah rotorů. Gravitace působí v kladném směru osy z_e a tah v záporném směru osy z_b .

- Liché rotory se točí proti směru hodinových ručiček, sudé rotory po směru hodinových ručiček (viz obrázek 3.5).



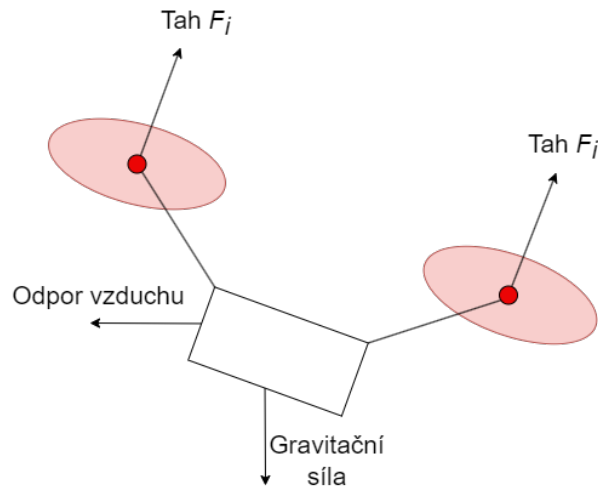
Obrázek 3.5: Kvadroptéra a směr otáčení rotorů

3.3.1 Model efektivity řízení

V této subsekcí mimo [3] využíváme vzorce i z [6] na vyznačených místech.

Model efektivity řízení popisuje, jak se změny v řízení, jako jsou otáčky rotorů nebo úhly natočení, promítají v pohybu kvadroptéry. Pro vstup se používá úhlová rychlost rotorů. Na výstupu se určuje tah F_i a točivý moment M_i působící na jednotlivé rotory a točivý moment τ_B okolo os x_b , y_b a z_b .

Tah (Thrust) je síla směřující v negativním směru osy z_b , kterou kvadroptéra generuje a umožňuje jí pohybovat se vzduchem ve specifickém směru (viz obrázek 3.6). Kolektivní působení všech čtyř rotorů vytváří tah celého dronu. Ovládním rychlosti rotorů kolektivně může kvadroptéra generovat tahovou sílu v požadovaném směru. Například pro pohyb vpřed se zvýší rychlost zadních rotorů a rychlost předních rotorů se sníží. Toto nerovnovážné rozložení tahu vytváří pohyb vpřed. Úpravou relativních rychlostí rotorů může kvadroptéra měnit směr pohybu.



Obrázek 3.6: Tah multikoptéry

Točivý moment je moment síly, který se vytváří rotací rotorů a působí proti otáčení kvadrokoptéry. Důležité je zachovat vyvážený točivý moment rotorů pro stabilitu letu kvadrokoptéry. Při nerovnováze točivého momentu by se kvadrokoptéra začala otáčet v nežádoucím směru.

Dle předpokladu na dron nepůsobí vítr, poté je tah jednotlivých rotorů

$$F_i = k\omega_i^2, \quad (3.7)$$

kde k je konstanta tahu a ω_i^2 je druhá mocnina úhlové rychlosti daného rotoru [6].

Celkový tah je spočítán jako suma tahů všech rotorů

$$f = \sum_{i=1}^4 |F_i| = k \sum_{i=1}^4 \omega_i^2. \quad (3.8)$$

Podobným způsobem se určuje točivý moment každého rotoru

$$M_i = b\omega_i^2. \quad (3.9)$$

kde b je konstanta momentu určená experimenty [6].

Točivý moment okolo os x , y a z se popisuje různými způsoby podle toho, zda má kvadrokoptéra “+” nebo “x”-konfiguraci. V případě “x”-konfigurace je moment popsán jako

$$\tau_B = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} lk \left(\frac{\sqrt{2}}{2} \omega_1^2 - \frac{\sqrt{2}}{2} \omega_2^2 - \frac{\sqrt{2}}{2} \omega_3^2 + \frac{\sqrt{2}}{2} \omega_4^2 \right) \\ lk \left(\frac{\sqrt{2}}{2} \omega_1^2 + \frac{\sqrt{2}}{2} \omega_2^2 - \frac{\sqrt{2}}{2} \omega_3^2 - \frac{\sqrt{2}}{2} \omega_4^2 \right) \\ b(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix}, \quad (3.10)$$

kde l značí vzdálenost rotorů od těžiště kvadrokoptéry.

3.3.2 Kinematický model tuhého tělesa

Kinematika neuvažuje hmotnost ani sílu působící na dron. Kinematický model popisuje pohyb kvadrokoptéry na základě její polohy, orientace, rychlosti a úhlové rychlosti. Tento model předpokládá, že kvadrokoptéra je tuhé těleso a neexistuje deformace ani pohyb vnitřních částí dronu. Jako vstup se pro tento model používá rychlost a úhlová rychlost. Výstupem je pozice a orientace dronu.

Vektor, který popisuje pozici kvadrokoptéry v jejím těžišti se značí ${}^e p \in \mathbb{R}$

$${}^e \dot{p} = {}^e v, \quad (3.11)$$

kde vektor ${}^e v \in \mathbb{R}$ reprezentuje rychlost kvadrokoptéry.

Rozepsáním rovnice (3.6) dostáváme

$$\dot{\Theta} = W^{-1} \cdot {}^b \omega, \quad (3.12)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan(\theta)\sin(\phi) & \tan(\theta)\cos(\phi) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \begin{bmatrix} {}^b \omega_x \\ {}^b \omega_y \\ {}^b \omega_z \end{bmatrix}.$$

kde $\dot{\Theta}$ je úhlová rychlost v souřadnicích $ABCF$. Vztah (3.12) lze využít k přepočtu úhlové rychlosti těla dronu ${}^b \omega$ na derivaci orientace $[\phi, \theta, \psi]$.

3.3.3 Dynamický model tuhého tělesa

Dynamický model [6] [7] popisuje pohyb a chování kvadrokoptéry z hlediska sil a momentů, včetně gravitace, aerodynamických sil a momentů generovaných rotory. V tomto modelu jsou zahrnuty inerciální a kinematické vlastnosti dronu, jako je hmotnost, setrvačnost a geometrie tělesa.

Rotační pohyb [7] kvadrokoptéry jsme popsali Eulerovo rovnicí rotace

$$I\dot{\Theta} + \Theta \times I\Theta = \tau_B, \quad (3.13)$$

kde I je inerciální matice kvadrokoptéry. τ je točivý moment na osách $ABCF$ a Θ je vektor orientace $[\phi, \theta, \psi]^T$.

Díky symetrické konstrukci kvadrokoptéry je inerciální matice I zredukována na diagonální matici

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}.$$

Doplněním do rovnice (3.13) vznikne

$$\begin{aligned} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \\ = \begin{bmatrix} lk \left(\frac{\sqrt{2}}{2} \omega_1^2 - \frac{\sqrt{2}}{2} \omega_2^2 - \frac{\sqrt{2}}{2} \omega_3^2 + \frac{\sqrt{2}}{2} \omega_4^2 \right) \\ lk \left(\frac{\sqrt{2}}{2} \omega_1^2 + \frac{\sqrt{2}}{2} \omega_2^2 - \frac{\sqrt{2}}{2} \omega_3^2 - \frac{\sqrt{2}}{2} \omega_4^2 \right) \\ b(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix}, \end{aligned}$$

$$\begin{bmatrix} I_{xx} \ddot{\phi} \\ I_{yy} \ddot{\theta} \\ I_{zz} \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_{xx} \dot{\phi} \\ I_{yy} \dot{\theta} \\ I_{zz} \dot{\psi} \end{bmatrix} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}.$$

Po úpravě vznikne vzorec

$$\ddot{\phi} = \frac{(I_{yy} - I_{zz})\dot{\theta}\dot{\psi}}{I_{xx}} + \frac{\tau_x}{I_{xx}}, \quad (3.14)$$

$$\ddot{\theta} = \frac{(I_{zz} - I_{xx})\dot{\phi}\dot{\psi}}{I_{yy}} + \frac{\tau_y}{I_{yy}}, \quad (3.15)$$

$$\ddot{\psi} = \frac{(I_{xx} - I_{yy})\dot{\phi}\dot{\theta}}{I_{zz}} + \frac{\tau_z}{I_{zz}}. \quad (3.16)$$

Úpravou rovnice (3.12) dostáváme

$${}^b\omega = W\dot{\Theta}, \quad (3.17)$$

$$\begin{bmatrix} {}^b\omega_x \\ {}^b\omega_y \\ {}^b\omega_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \cos(\theta)\sin(\phi) \\ 0 & -\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}.$$

Pokud řídíme systém okolo rovnovážného bodu pouze malými změnami orientačních úhlů roll (ϕ) a pitch (θ), matice W bude jednotková matice $I_{3 \times 3}$ a bude platit vztah

$${}^b\omega = \dot{\Theta}. \quad (3.18)$$

Předpokládáme, že silové působení na dron zahrnuje pouze gravitaci a tah. Ostatní jevy (například vítr) jsou zanedbány. Pro zrychlení platí následující rovnice

$${}^e\dot{v} = ge_3 - \frac{f}{m} b_3. \quad (3.19)$$

$g \in \mathbb{R}_+$ reprezentuje gravitační zrychlení, $f \in \mathbb{R}_+ \cup \{0\}$ znamená celkový tah působící na dron a m je hmotnost kvadrokoptéry.

3. Návrh modelu kvadrokoptéry

g je kladné kvůli směru osy z , která směřuje dolů pod dron. Ze stejného důvodu je f záporné. Pro získání vektoru ${}^e b_3$ v souřadnicích $ABCF$ musíme vynásobit vektor e_3 rotační maticí R

$${}^e \dot{v} = g e_3 - \frac{f}{m} R e_3. \quad (3.20)$$

Spojením rovnic (3.11, 3.20, 3.14, 3.15, 3.16) získáme základní model kvadrokoptéry

$$\begin{aligned} {}^e \dot{p} &= {}^e v, \\ {}^e \dot{v} &= g e_3 - \frac{f}{m} R e_3, \\ \ddot{\phi} &= \frac{(I_{yy} - I_{zz}) \dot{\theta} \dot{\psi}}{I_{xx}} + \frac{\tau_x}{I_{xx}}, \\ \ddot{\theta} &= \frac{(I_{zz} - I_{xx}) \dot{\phi} \dot{\psi}}{I_{yy}} + \frac{\tau_y}{I_{yy}}, \\ \ddot{\psi} &= \frac{(I_{xx} - I_{yy}) \dot{\phi} \dot{\theta}}{I_{zz}} + \frac{\tau_z}{I_{zz}}. \end{aligned} \quad (3.21)$$

Tento model používá oba souřadnicové systémy $EFCF$ a $ABCF$. Pozice p a rychlost v kvadrokoptéry je popisována v $EFCF$. Tento způsob je přehlednější při určování polohy dronu v prostoru. $ABCF$ je používán pro popis tahu f , točivých momentů τ_B a orientace Θ . Tah má vždy stejný směr jako záporný směr osy z_b .

Kvadrokoptéra má 6 stupňů volnosti, 3 stupně pro translaci a 3 pro rotaci, ale pouze 4 motory na ovládání, to znamená, že kvadrokoptéra je “under-actuated” (nedořízený) systém [6].

Sledování trajektorie kvadrokoptéry pomocí MPC

4

V této sekci se budeme zabývat metodou pro sledování trajektorie multikoptérou pomocí MPC. Vysvětlíme funkci prediktivního regulátoru, jakým způsobem sleduje trajektorii a postup při určování řídicí proměnné. Poté upřesníme parametry použité pro modelování systému kvadrokoptéry.

Popíšeme softwarové aplikace, balíčky a rozhraní použité při vytváření systému a regulátorů pro sledování trajektorie. Přesně určíme váhovou funkci, která je součástí optimalizačního problému, který musí MPC řešit. Definujeme stav a řídicí proměnné kvadrokoptéry, váhové matice Q_l , Q_N a R , a určíme počáteční podmínky pro obě trajektorie. Specifikujeme jaké referenční trajektorie budou použity pro úlohu sledování. Spočítáme omezení pro řízení a určíme omezení pro stavy kvadrokoptéry.

4.1 Sledování trajektorie pomocí MPC

MPC je metoda pro řízení "multi-input-multi-output" systémů [7]. MPC predikuje budoucí chování systému pomocí matematického modelu a podle toho určuje řídicí proměnné. Jinými slovy, cílem MPC je najít optimální strategii řízení na postupném horizontu. Ta je hledána minimalizací váhové funkce $J(x_k, u_k^{k+N}, N)$ s ohledem na omezení systému a modelu chování. Váhová funkce MPC má pro sledování trajektorie kvadrokoptéry tvar

$$J(x_k, u_k^{k+N}, N) = (x_N - x_{ref,N})^T Q_N (x_N - x_{ref,N}) + \sum_{l=k}^{k+N-1} [(x_l - x_{ref,l})^T Q_l (x_l - x_{ref,l}) + (u_l - u_{ref,l})^T R_l (u_l - u_{ref,l})], \quad (4.1)$$

kde x_k jsou predikované stavy kvadrokoptéry, $x_{ref,k}$ jsou referenční stavy, u_k jsou řídicí proměnné a $u_{ref,k}$ jsou referenční hodnoty řídicích proměnných. Q a R jsou

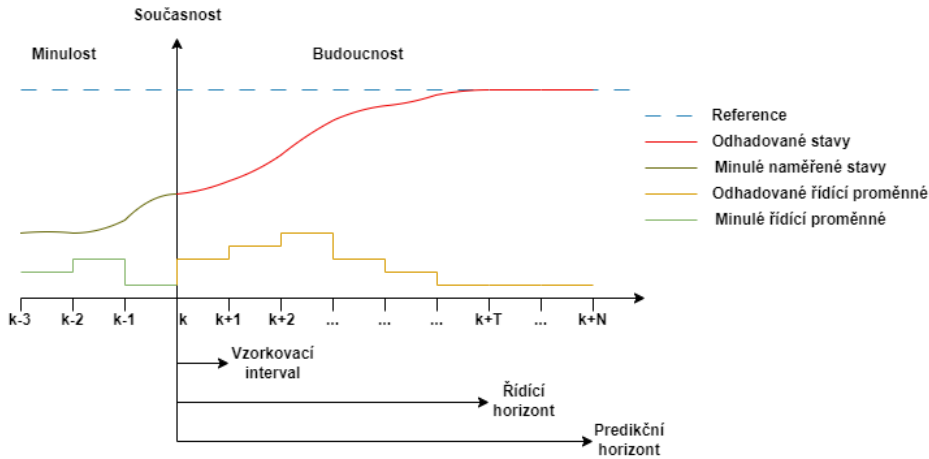
4. Sledování trajektorie kvadrokoptéry pomocí MPC

diagonální semidefinitní váhové matice, které přiřazují váhu jednotlivým stavům a řídicím proměnným.

MPC potřebuje k predikci model systému (3.21), který je definován jako

$$x_{l+1} = f(x_l, u_l), \quad l = k, k + 1, \dots, k + N - 1 \quad (4.2)$$

kde $f \in \mathbb{R}^n \times \mathbb{R}^m$ je model kvadrokoptéry (3.21), x_l je vektor stavu systému a u_l je řídicí vektor systému. MPC následně vytváří predikční horizont N (viz obrázek 4.1), což je časové období, ve kterém se předpokládá, že bude systém reagovat na řídicí signály. MPC poté dokáže přes délku horizontu N minimalizovat váhovou funkci $J(x_k, u_k^{k+N}, N)$ a určit sekvenci řídicích vstupů u_k^{k+N} . Ze sekvence řídicích vstupů se většinou využívá pouze první element, protože se proces neustále opakuje pro aktuální referenci a stav systému.



Obrázek 4.1: MPC řídicí a predikční horizont

Prediktivní řízení dokáže takto přesně sledovat požadovanou trajektorii a zohledňovat u toho počáteční podmínky stavu x_0 a omezení kvadrokoptéry

$$\begin{aligned} u_{min} &\leq u_i \leq u_{max}, \\ x_{min} &\leq x_i \leq x_{max}, \quad i = k, k + 1, \dots, k + N, \end{aligned} \quad (4.3)$$

kde u_{min} a u_{max} jsou omezení řídicí proměnné, což mohou být například maximální a minimální otáčky rotorů. x_{min} a x_{max} jsou omezení týkající se stavu systému. Může jít například o omezení prostoru, ve kterém se smí dron pohybovat, nebo maximální rychlost, kterou může létat.

Váhová funkce $J(x_k, u_k^{k+N}, N)$, model systému (4.2), počáteční podmínky x_k a

omezení kvadroptéry (4.3) tvoří dohromady problém optimálního řízení (OCP) [8]

$$\begin{aligned}
 & \min_u J(x_k, u_k^{k+N}, N), \\
 & \text{s.t. } x_{k+1} = f(x_k, u_k), \\
 & \quad x_k = x_0, \quad k = 0, \\
 & \quad u_{min} \leq u_i \leq u_{max}, \\
 & \quad x_{min} \leq x_i \leq x_{max}.
 \end{aligned} \tag{4.4}$$

K řešení OCP se často využívá převedení na úlohu nelineárního programování (NLP).

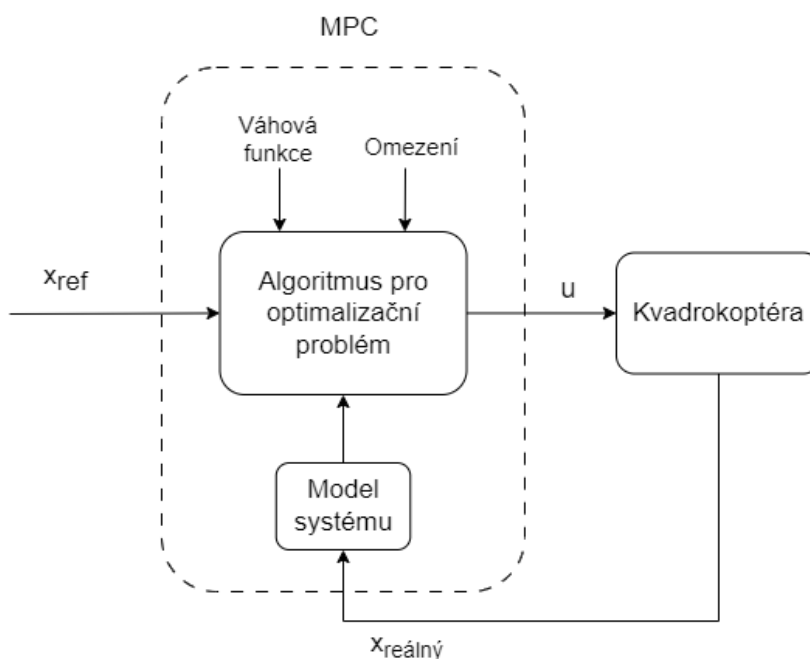
NLP je optimalizační problém, kde je cílem nalézt vektor rozhodovacích proměnných $w = [w_1, w_2, \dots, w_n]$, který minimalizuje funkci $\Phi(w)$ při daných omezení $g(w)$ [9]. Úloha NLP může být popsána jako

$$\begin{aligned}
 & \min_w \Phi(w), \\
 & \text{s.t. } g_1(w) \leq 0, \\
 & \quad g_2(w) = 0,
 \end{aligned} \tag{4.5}$$

kde w je rozhodovací proměnná, obvykle řídicích veličin u . V případě této práce obsahuje w řídicí veličiny u a i stavy systému x . Omezení nerovnosti $g_1(w)$ může například popisovat překážky v prostoru nebo omezení na stav a řízení. Pomocí omezení rovnosti $g_2(w)$ lze popsat rozdíl mezi dynamikou systému a predikcí stavu danou rovnicí (4.2).

Dále popíšeme princip MPC pro řízení kvadroptéry [10] (viz obrázek 4.2). Nejprve jsme v sekci 3.3 vytvořili matematický model kvadroptéry (3.21). Poté se definuje referenční trajektorie, kterou má kvadroptéra sledovat. Formulace se optimalizační problém, jehož řešením je nalezení optimální strategie řízení pro minimalizaci kritéria optimality MPC. Formulace OCP, kritéria optimality a omezení se provádí pouze jednou před prvním použitím regulátoru.

MPC následně předpovídá budoucí stav kvadroptéry na základě aktuálního stavu, řídicích signálů a matematického modelu. Predikce se provádí pro určitý řídicí horizont, ve kterém se očekává reakce kvadroptéry. Z optimální strategie řízení je využit pouze první člen. Poté je optimalizační problém řešen znovu pro aktualizovaný stav, referenci a predikční horizont. Opakovaným řešením OCP je uzavřena zpětná vazba, díky níž je narozdíl od klasického optimálního řízení robustní.



Obrázek 4.2: Model predictive control (MPC) diagram

Výhody implementace MPC může být, že díky zahrnutí matematického modelu systému dokáže MPC zohledňovat požadovanou dynamiku a omezení. MPC dokáže předvídat budoucí chování dronu a dokáže se přizpůsobit změnám v podmínkách a požadavcích na řízení.

4.2 Implementace MPC

Parametry kvadrokoptéry, které budou využity v modelu, který je součástí MPC, jsme převzali z [6] a jsou popsány v tabulce 4.1.

Název	Symbol	Hodnota
Délka ramen	l	0.243 m
Hmotnost dronu	m	1.587 kg
Koeficient tahu	k	$4.0687 \cdot 10^{-7} \text{ N/rpm}^2$
Koeficient točivého momentu	b	$8.4367 \cdot 10^{-9} \text{ Nm/rpm}^2$
Hodnota inercie na ose x	I_{xx}	0.0213 kgm ²
Hodnota inercie na ose y	I_{yy}	0.02217 kgm ²
Hodnota inercie na ose z	I_{zz}	0.0282 kgm ²

Tabulka 4.1: Parametry kvadrokoptéry

Váhové funkci (4.1) jsme pro $u_{ref,k}$ přiřadili podle níže vypočítaného vztahu (4.10) hodnotu otáček rotoru 3093 RPM, kdy se kvadroptéra vznáší ve vzduchu, aniž by klesala nebo stoupala.

Vektor stavu kvadroptéry lze popsat jako $x_l = [x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$, kde $p = [x, y, z]^T$ je poloha, $\Theta = [\phi, \theta, \psi]^T$ je orientace, $v = [\dot{x}, \dot{y}, \dot{z}]^T$ je rychlost a $\dot{\Theta} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ je úhlové zrychlení kvadroptéry. Vektor řízení je složen z druhých mocnin požadovaných úhlových rychlostí jednotlivých rotorů a lze je popsat jako $u_l = [\omega_1^2, \omega_2^2, \omega_3^2, \omega_4^2]$.

Systém má 6 stupňů volnosti, ale jen 4 řídicí proměnné. Musíme tedy určit, které stavy budeme řídit. K tomu slouží matice Q , která přiřadí stavům váhy důležitosti, a zavedení omezení na některé stavy. Váhy jsme přiřadili pozicím $[x, y, z]$ a třem Eulerovým úhlům $[\phi, \theta, \psi]$.

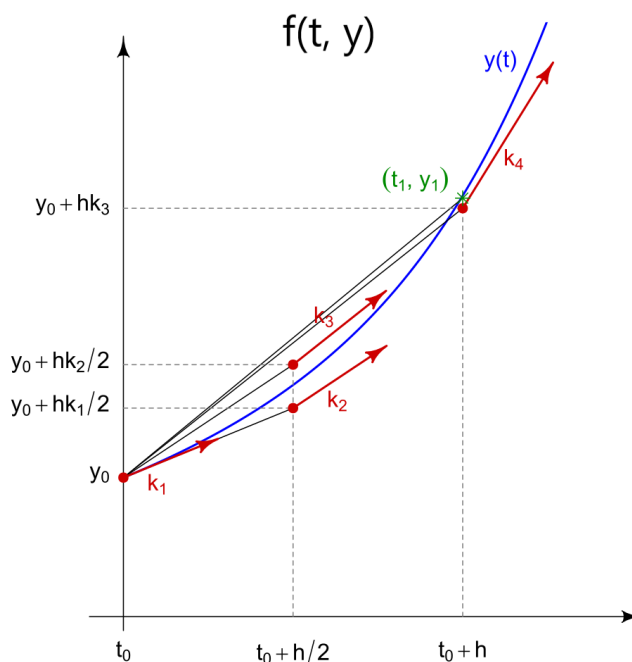
Váhové matice Q a R pro sledování trajektorie jsou definovány jako

$$\begin{aligned} Q_l &= \text{diag} \left(\left[0.25, 0.25, 0.25, \left(\frac{\pi}{2}\right)^{-2}, \left(\frac{\pi}{2}\right)^{-2}, \pi^{-2}, 0, 0, 0, 0, 0, 0 \right] \right), \\ Q_N &= \text{diag} \left(\left[1, 1, 1, \left(\frac{\pi}{2}\right)^{-2}, \left(\frac{\pi}{2}\right)^{-2}, \pi^{-2}, 0, 0, 0, 0, 0, 0 \right] \right), \\ R &= \text{diag} \left(\left[(u_{max} - u_{min})^{-2}, (u_{max} - u_{min})^{-2}, (u_{max} - u_{min})^{-2}, (u_{max} - u_{min})^{-2} \right] \right). \end{aligned} \quad (4.6)$$

Matice Q_l , Q_N a R se určují, aby poskytly MPC informace, které stavy a řídicí proměnné jsou důležitější pro přesné sledování. Matice Q_l a Q_N určují důležitost stavům x_l a x_N , zatímco matice R určuje důležitost přesného sledování řídicích proměnných u_l . MPC neřeší řízení na konci horizontu, proto je dobré přiřadit polohám v matici Q_N větší hodnotu, aby MPC kladlo na konci řízení větší důležitost na řízení do správné polohy.

Dynamický model kvadroptéry (4.2) jsme sestavili z dříve uvedených rovnic (3.21) a jsou reprezentovány v OCP podle dříve popsání vztahu (4.4).

Dynamika kvadroptéry byla diskretizována pomocí Runge-Kuttovi metody čtvrtého řádu se vzorkováním 0.1 s. Pro představu vidíme Runge-Kuttovu metodu 4. řádu pro dynamiku $\dot{y} = f(t, y)$ na obrázku 4.3.



Obrázek 4.3: RK4 pro $\dot{y} = f(t, y)$, zdroj [11]

Počáteční hodnoty stavu x_0 jsme pro obě trajektorie nastavili na nulové s výjimkou pozice na ose z_e , kde se kvadroptéra nachází ve 2.5 m nad zemí $x_0 = [0, 0, -2.5, 0, 0, 0, 0, 0, 0, 0, 0, 0]$. 2.5 m je z důvodu, že počítáme s řízením dronu, který se již vznáší stabilizovaný ve vzduchu.

Referenční trajektorie pro let ve tvaru lemniskáty byla určena jako

$$\begin{aligned} x &= \cos\left(\frac{t}{2} + \frac{\pi}{2}\right) \cdot 10, \\ y &= \left(\frac{\sin\left(2 \cdot \left(\frac{t}{2} + \frac{\pi}{2}\right)\right)}{2}\right) \cdot 10, \\ z &= -2.5. \end{aligned} \tag{4.7}$$

ve tvaru spirály vypadá referenční trajektorie následovně

$$\begin{aligned} x &= 5 \cdot \cos\left(\frac{t}{2}\right) - 5, \\ y &= 5 \cdot \sin\left(\frac{t}{2}\right), \\ z &= -\left(2.5 + \frac{t}{2}\right). \end{aligned} \tag{4.8}$$

4.2.1 Omezení

Pro správné řízení je důležité určit vhodná omezení pro stavy a řízení. Při špatných omezení řízení se může stát, že se bude MPC snažit řídit většími otáčkami, než jaké je rotor schopný vygenerovat. Nebo se naopak bude snažit řídit na příliš nízké otáčky a zastaví tím motory. Omezení na stavy je důležité například proto, aby se dron nemohl pokusit letět "pod zem", nebo aby nebyla orientace natolik velká, aby převrátila dron.

Regulátor jsme doplnili o omezení pro výšku z , aby dron nenarazil do země. To znamená podle souřadnicového systému (viz obrázek 3.2) do kladných hodnot $z \leq 0$. Omezení jsme nastavili také pro orientaci $[\phi, \theta, \psi]$

$$\begin{aligned} -\frac{\pi}{2} &\leq \phi \leq \frac{\pi}{2}, \\ -\frac{\pi}{2} &\leq \theta \leq \frac{\pi}{2}, \\ -\pi &\leq \psi \leq \pi. \end{aligned} \quad (4.9)$$

Poté vypočítáme omezení úhlové rychlosti ω_i pro každý rotor. Maximální úhlová rychlost ω_i byla nalezena 4720 RPM. Nyní spočítáme hodnotu úhlové rychlosti, při které je kvadrokoptéra v rovnovážném stavu, neklesá ani nestoupá. Nejdříve spočítáme gravitační sílu, která táhne dron k zemi. Tu spočítáme pomocí hmotnosti dronu $m = 1.587\text{kg}$ a gravitačního zrychlení $g = 9.81\text{m/s}^2$

$$F = m \cdot g = 1.587 \cdot 9.81 = 15.56847\text{N}.$$

Vypočtená síla je potřebná hodnota tahu v opačném směru osy z_e , aby se dron udržel ve vzduchu

$$f = 15.56847 = k \sum_{i=1}^4 \omega_i^2.$$

Z tabulky 4.1 známe koeficient tahu $k = 4.0687 \cdot 10^{-7}$. Můžeme spočítat hodnotu úhlové rychlosti ω_i pro rovnovážný stav

$$\omega_i = \sqrt{\frac{f}{4 \cdot k}} = \sqrt{\frac{15.56847}{4 \cdot 4.0687 \cdot 10^{-7}}} \approx 3093 \text{ RPM}. \quad (4.10)$$

Určíme spodní omezení pro řízení, které bude menší než rovnovážný stav, ale dost velké, aby se nezastavily motory

$$1000^2 < u_l < 4720^2, \quad l = k, k + 1, \dots, k + N - 1. \quad (4.11)$$

V případě, kdybychom chtěli sledovat omezení pro rotační pohyb τ_B , stačí dosadit horní a spodní hranice úhlových rychlostí ω_i do vzorce (3.10)

$$\begin{bmatrix} \tau_{x_{max}} \\ \tau_{y_{max}} \\ \tau_{z_{max}} \end{bmatrix} = \begin{bmatrix} lk \left(\frac{\sqrt{2}}{2} 4720^2 - \frac{\sqrt{2}}{2} 3000^2 - \frac{\sqrt{2}}{2} 3000^2 + \frac{\sqrt{2}}{2} 4720^2 \right) \\ lk \left(\frac{\sqrt{2}}{2} 4720^2 + \frac{\sqrt{2}}{2} 4720^2 - \frac{\sqrt{2}}{2} 3000^2 - \frac{\sqrt{2}}{2} 3000^2 \right) \\ b(4720^2 - 3000^2 + 4720^2 - 3000^2) \end{bmatrix} \simeq \begin{bmatrix} 1.8566 \\ 1.8566 \\ 0.2241 \end{bmatrix}, \quad (4.12)$$

kde délka ramen l , koeficient tahu k a koeficient točivého momentu b jsou definované v tabulce 4.1.

Stejným způsobem získáme i spodní omezení,

$$\begin{bmatrix} \tau_{x_{min}} \\ \tau_{y_{min}} \\ \tau_{z_{min}} \end{bmatrix} = \begin{bmatrix} lk \left(\frac{\sqrt{2}}{2} 3000^2 - \frac{\sqrt{2}}{2} 4720^2 - \frac{\sqrt{2}}{2} 4720^2 + \frac{\sqrt{2}}{2} 3000^2 \right) \\ lk \left(\frac{\sqrt{2}}{2} 3000^2 + \frac{\sqrt{2}}{2} 3000^2 - \frac{\sqrt{2}}{2} 4720^2 - \frac{\sqrt{2}}{2} 4720^2 \right) \\ b(3000^2 - 4720^2 + 3000^2 - 4720^2) \end{bmatrix} \simeq \begin{bmatrix} -1.8566 \\ -1.8566 \\ -0.2241 \end{bmatrix}. \quad (4.13)$$

Implementace a výsledky

5

V této kapitole popíšeme implementaci MPC v Matlab/Simulinku s cílem řídit kvadrokoptéru podél zadané trajektorie. Provedeme simulaci sledování trajektorie. Pro některé scénáře simulujeme i poruchu senzorů akcelerace a úhlové rychlosti. Zobrazíme a vyhodnotíme výsledky řízení pomocí MPC pro dvě referenční trajektorie.

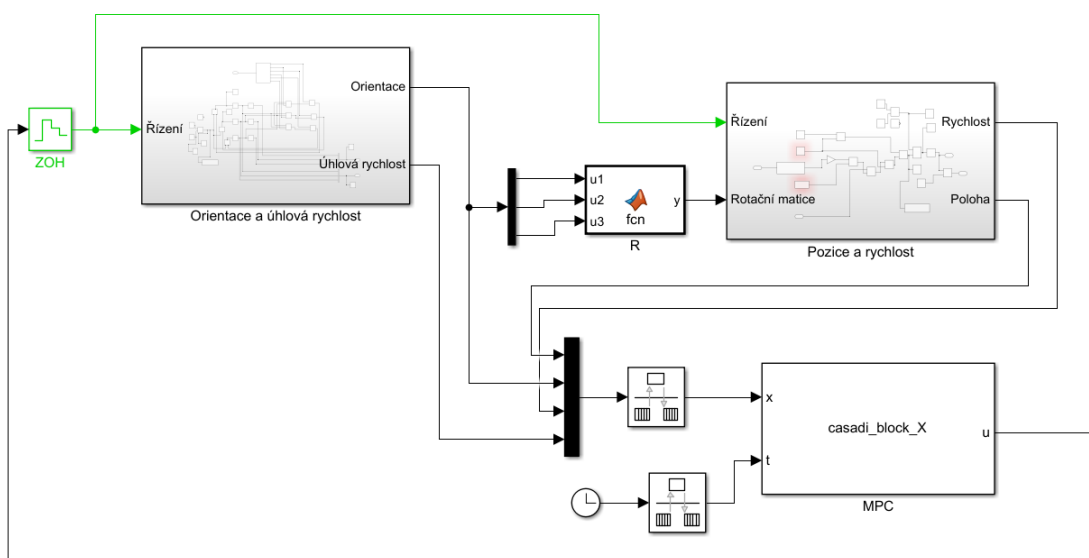
5.1 Implementace v Matlab/Simulinku

MPC jsme implementovali v Matlabu a v Simulinku s použitím balíčku CasADi.

Matlab je výkonný nástroj pro modelování, analýzu a řízení systémů, který poskytuje širokou škálu funkcí a knihoven pro práci s řídicími systémy. Jeho flexibilita a možnost rozšiřitelnosti umožňují vytváření a ladění regulátorů s vysokou přístupností. Matlab tak poskytuje ideální prostředí pro vývoj a testování regulátorů sledování trajektorie dronu.

CasADi [12] je softwarový nástroj vyvinutý pro optimalizaci a matematické modelování. CasADi umožňuje provádět matematické operace nad symbolickými proměnnými. Tímto způsobem uživatelé mohou vytvářet matematické modely systému a optimalizační funkce pro MPC.

Model chování dronu byl vytvořen na základě popisu v sekci 3. Model dronu je implementován v Simulinku a algoritmus MPC je spouštěn pomocí Matlab system bloku (viz obrázek 5.1). Referenční trajektorie je uložena přímo v MPC.



Obrázek 5.1: Model v Simulinku

5.1.1 Matlab System

Základní funkcí balíčku CasADi je možnost vytvářet a počítat se symbolickými proměnnými. Symbolické proměnné využíváme pro definici stavu x , řízení u a parametrů p . Pro definici symbolických proměnných využíváme dva způsoby, které CasADi podporuje. Definici skrze SX symboly a MX symboly. Oba způsoby definují symbolickou proměnnou, ale rozdíl je, jakým způsobem vypisují funkce, ve kterých jsou použity.

Násobení a sčítání v matici probíhá pro SX proměnné v každém elementu matice zvlášť. SX proměnné dokážou provádět pouze elementární operace pro $R \rightarrow R$ nebo $R \times R \rightarrow R$. Elementární operace používané pro tvorbu MX výrazů mohou být více řádkové matice: $R^{n_1 \times m_1} \times \dots \times R^{n_N \times m_N} \rightarrow R^{p_1 \times q_1} \times \dots \times R^{p_M \times q_M}$ Nemůžeme používat obě symboliky v jednom výrazu, ale můžeme pomocí MX symboliky volat funkci složenou z SX proměnných, což jsme využili v algoritmu.

Pro definici stavů x_1, \dots, x_{12} , řízení u_1, \dots, u_4 a parametrů pro stavy p_{x1}, \dots, p_{x12} a pro řízení p_{u1}, \dots, p_{u4} využijeme SX proměnné.

```

1 p_x = SX.sym('p_x', 12);
2 p_u = SX.sym('p_u', 4);
3 x = SX.sym('x', 12);
4 u = SX.sym('u', 4);

```

Stav a řídicí proměnné použijeme pro vytvoření matematických rovnic dynamiky kvadrokoptéry a vložíme je do SX funkce:

```

1 f = Function('f', {x, u}, {xdot});

```


Dynamiku zdiskretizujeme pomocí Runge-Kuttovi metody 4. řádu. Runge-Kuttova metoda 4. řádu vypadá v algoritmu následovně:

```

1 k1 = f(X, U);
2 k2 = f(X + DT/2 * k1, U);
3 k3 = f(X + DT/2 * k2, U);
4 k4 = f(X + DT * k3, U);
5 X = X + DT/6 * (k1 + 2 * k2 + 2 * k3 + k4);

```

DT je velikost kroku. X jsou MX proměnné stavu a U jsou proměnné řízení.

Váhová funkce MPC je složena ze stavů dronu XN , řídicích proměnných U , parametrů, které znázorňují počáteční a referenční hodnoty stavu Px a referenční hodnoty řízení Pu . Referenci Pu nastavíme na spočítané rovnovážné vznášení dronu (4.10):

```

1 L = (XN - Px)' * Q * (XN - Px) + (U - Pu)' * R * (U - Pu);
2 L_end = (XN - Px)' * Q_end * (XN - Px);

```

Váhovou funkci vložíme do MX funkce a voláme jí pro každý okamžik v predikčním horizontu:

```

1 J_f = F2('x0', Xk, 'p', Uk, 'px', Pkx, 'pu', Pku);
2 J = J + J_f.L;

```

Tím vytvoříme minimalizační úlohu, kterou MPC řeší.

MPC se musí převést na NLP podle vzorce (4.5). Rozhodovací proměnná w obsahuje spolu s řídicí proměnnou i stav kvadroptéry $w = [x_0, \dots, x_N, u_0, \dots, u_{N-1}]^T$.

Takto upravené NLP lze popsat jako

$$\begin{aligned}
 \min_w \quad & \Phi(w), \\
 \text{s.t.} \quad & g_2(w) = \begin{bmatrix} X_0 - x_0 \\ f(X_0, U_0) - x_1 \\ \vdots \\ f(X_{N-1}, U_{N-1}) - x_N \end{bmatrix} = 0.
 \end{aligned} \tag{5.1}$$

Nyní můžeme inicializovat solver pro řešení NLP. Solver je funkce, která je určena pro řešení NLP problémů [13]. Existuje více typů solverů (viz obrázek), ale zde používáme solver IPOPT (Interior Point Optimizer), který je volně dostupný v balíčku CasADi. IPOPT solver dokáže řešit NLP s více stupni volnosti a velkým množstvím omezení.

Pro inicializaci NLP solveru musíme definovat NLP problém k řešení. K tomu využijeme dříve vytvořenou minimalizační úlohu (4.1), rozhodovací proměnné w , omezení rovnosti $g(w)$ a parametry p . Následně vytvoříme objekt se solverem:

```
1 prob = struct('f', J, 'x', vertcat(w{:}), ...
2           'g', vertcat(g{:}), 'p', vertcat(p{:}));
3 options = struct('ipopt', struct('print_level', 0), ...
4           'print_time', false);
5 solver = nlpsol('solver', 'ipopt', prob, options);
```

Sestavený solver dokáže vyřešit NLP, pokud mu poskytneme hodnoty rozhodovacích proměnných x_k , počáteční a referenční hodnoty p , horní a spodní limity pro rozhodovací proměnné (ubx, lbx) a omezení rovnosti (ubg, lbg):

```
1 sol = solver('x0', w0, 'lbx', lbw, 'ubx', ubw, ...
2           'lbg', obj.lbg, 'ubg', obj.ubg, 'p', p_ref);
```

Popis MPC kvadrotéry a inicializace solveru je provedena pouze jednou. Solver je poté opakovaně volán pro aktuální stav a referenční trajektorii.

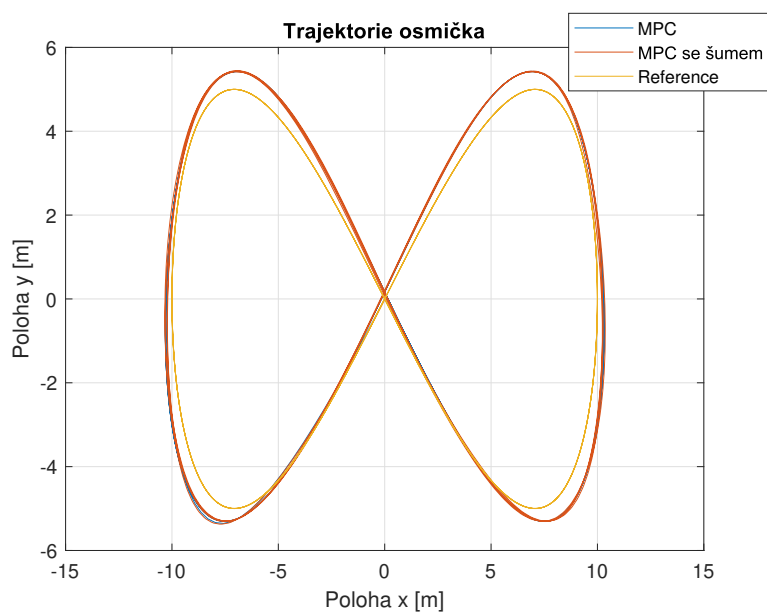
5.2 MPC výsledky

Do systému jsme za účelem otestování MPC zavedli poruchu jako typ šumu. Šum jsme vytvořili pomocí Band-Limited White Noise bloku, který generuje náhodné hodnoty s normální rozdělením, neboli bílý šum. Bílý šum byl pronásoben generátorem pulsů. Bílý šum má parametry spektrální hustoty výkonu $1 W \cdot Hz^{-1}$ a korelační čas se rovná vzorkovacímu času systému. Amplituda generátoru pulsů je rovna 1, perioda pulsů je stejná jako vzorkovací čas systému a šířka pulsů je 5 % velikosti periody. Tuto poruchu jsme zavedli do systému na dvou místech. První je v místě, kde se určí akcelerace (3.20). Druhé místo je po určení úhlového zrychlení (3.14, 3.15, 3.16).

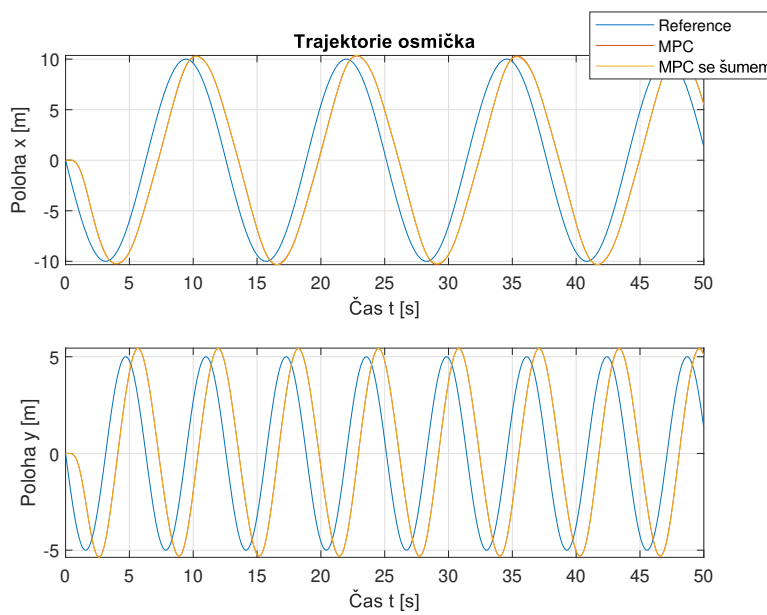
Nelineární MPC není navrženo, aby jakkoliv uvažovalo zavedenou poruchu, ale systém stabilizuje díky zpětné vazbě. Chyba měření se promítá nejvýrazněji v místech, kde by se nacházely senzory akcelerace a úhlové zrychlení. Jiné stavové veličiny jsou poruchou ovlivněny jen minimálně.

5.2.1 Trajektorie ve tvaru lemniskáty

Referenční trajektorie ve tvaru lemniskáty sledovaná pomocí MPC je generovaná podle vztahů (4.7). Obrázek 5.2 zobrazuje polohy x a y při řízení pomocí MPC. Obrázek 5.3 prezentuje polohy x a y v čase. Na obou obrázcích lze vidět, že řízená trajektorie sleduje správně referenční trajektorii. I přes poměrně velkou poruchu je výsledná trajektorie skoro totožná, což dokazuje robustnost využití MPC. Obrázek 5.3 ukazuje malé časové zpoždění předtím, než dojede řízená trajektorie na referenční. Tuto chybu můžeme pozorovat níže na obrázku 5.6.

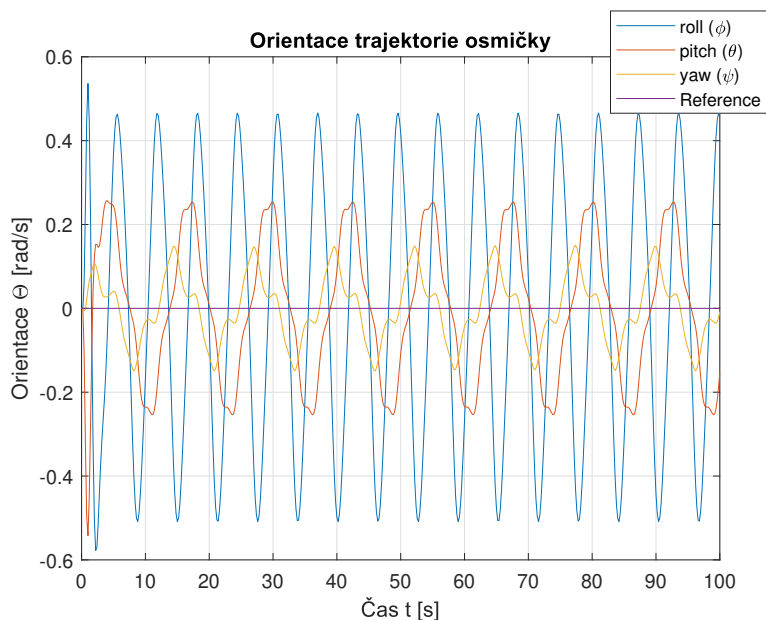


Obrázek 5.2: Poloha p lemniskáty



Obrázek 5.3: Poloha p v čase lemniskáty

Obrázek 5.4 představuje orientaci kvadrokoptéry při letu. Všechny úhly oscilují kolem referenční hodnoty. Největší výchylku v orientaci lze pozorovat na začátku letu, kde úhly ϕ a θ dosahují skoro -0.6 radiánů. Žádný úhel se nepřibližuje k omezení 4.9, a nehrozí tedy překlopení dronu.

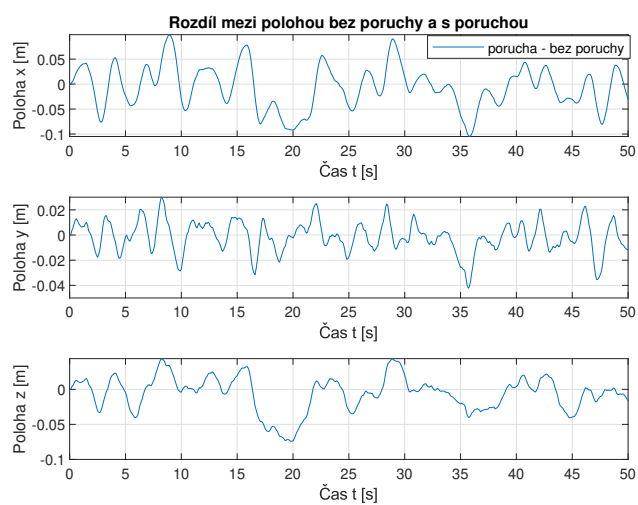


Obrázek 5.4: Orientace Θ lemniskáty

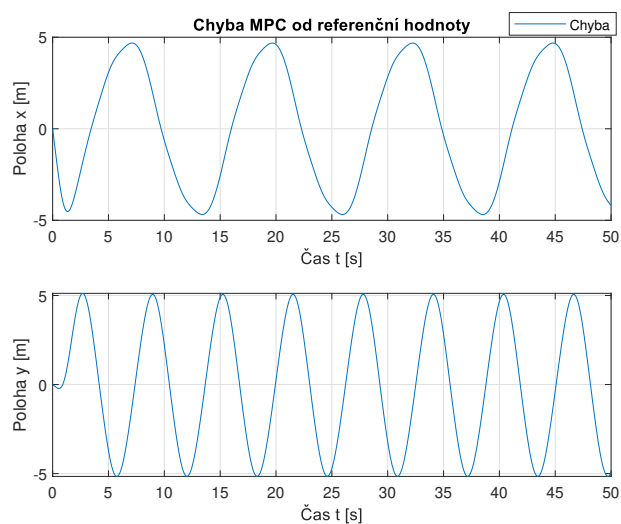
Na obrázku 5.5 můžeme pozorovat rozdíl mezi polohou trajektorie s poruchou a trajektorie bez poruchy. Rozdíl je i přes poruchu minimální a nedosahuje ani 0.1 metru.

Chybu polohy oproti referenční poloze reprezentuje obrázek 5.6. Chyba trajektorie pravidelně dosahuje necelých 5 metrů, což je způsobeno zpožděním, se kterým sleduje MPC referenci a tím, jak rychle referenční hodnota osciluje.

5.2.1. Trajektorie ve tvaru lemniskáty



Obrázek 5.5: Rozdíl mezi trajektorií bez a s poruchou

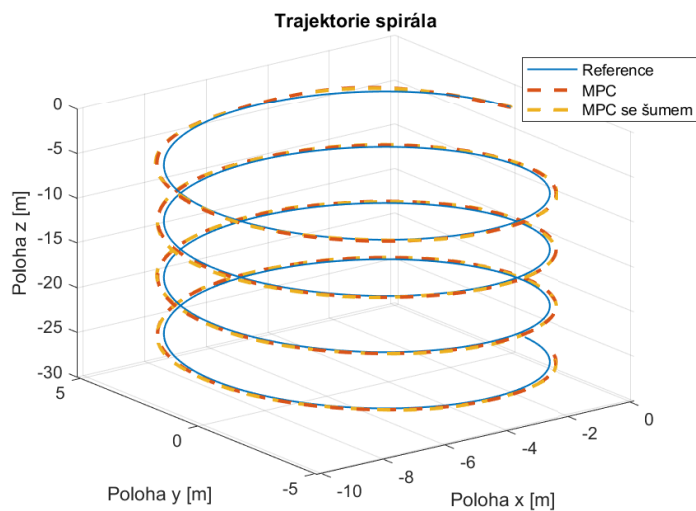


Obrázek 5.6: Chyba řízené trajektorie oproti referenci

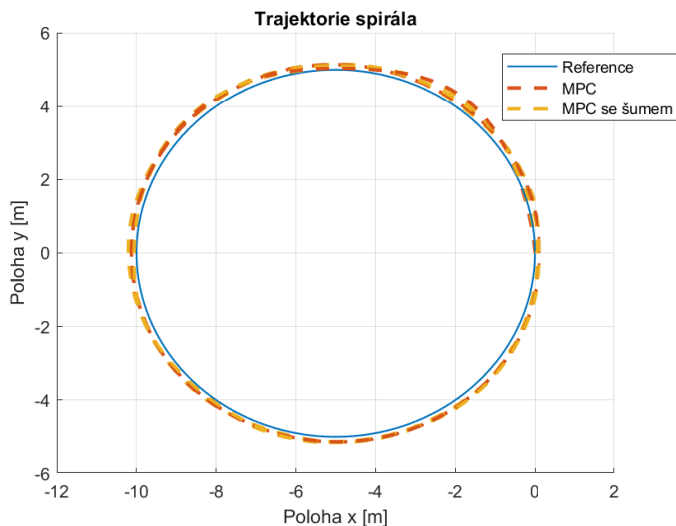
5.2.2 Trajektorie ve tvaru spirály

V této sekci sleduje MPC trajektorii definovanou podle (4.8).

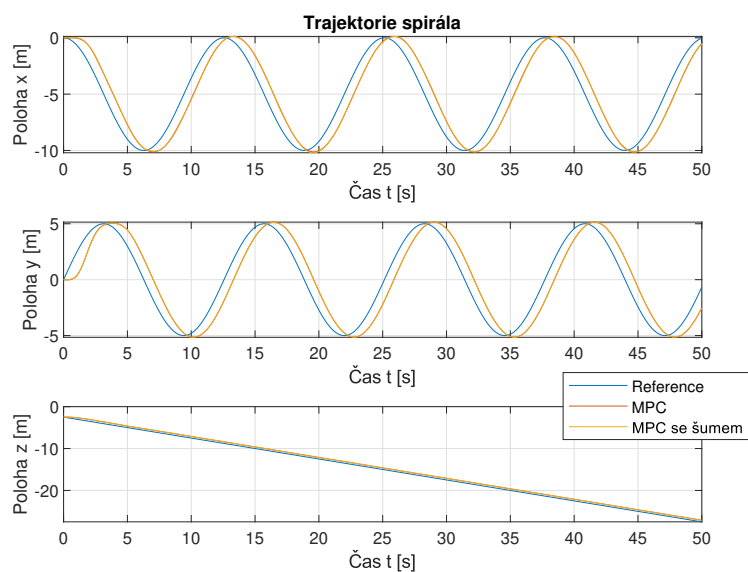
Obrázky 5.7, 5.8 a 5.9 ukazují polohu $[x, y, z]^T$ řízení s poruchou, bez poruchy a referenční trajektorii. Všechny obrázky dokazují, že MPC úspěšně sleduje referenci. Ani zde porucha nijak výrazně neovlivňuje výslednou polohu.



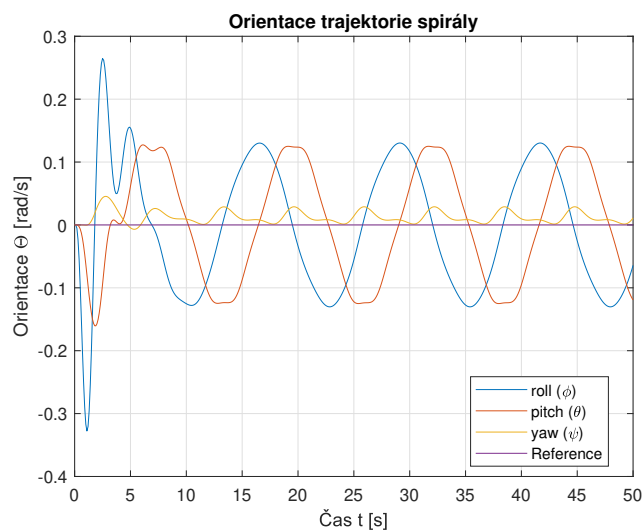
Obrázek 5.7: Poloha p spirály



Obrázek 5.8: Osy x a y spirály

Obrázek 5.9: Poloha p v čase spirály

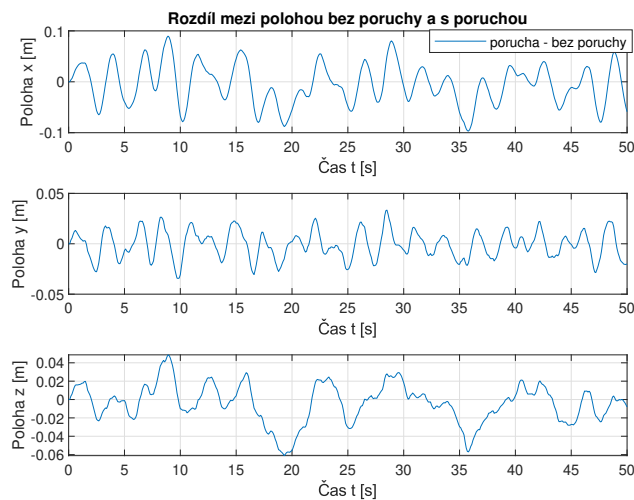
Orientace $[\phi, \theta, \psi]$ splňuje omezení (4.9), jak lze vidět na obrázku 5.10. Úhly ϕ a θ oscilují okolo nulové referenční hodnoty s největší výchylkou na začátku letu. Po srovnání již vykazuje periodickou oscilaci. Úhel ψ se drží reference nejlépe. Vybočuje jen v rámci setin radiánů do kladných hodnot.

Obrázek 5.10: Orientace Θ spirály

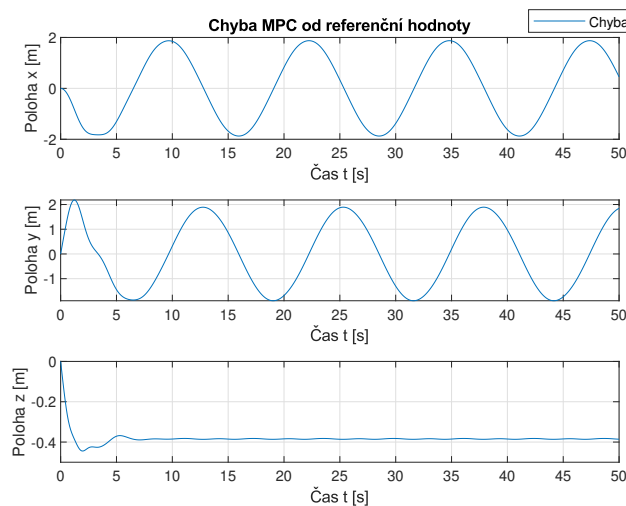
5. Implementace a výsledky

Rozdíl mezi polohou trajektorie s poruchou a bez poruchy, který lze pozorovat na obrázku 5.11, ukazuje velmi malou výchylku. Největší rozdíl v letu je ve směru osy x_e a v rámci 0.1 m.

Obrázek 5.12 prezentuje chybu MPC vůči referenci. V osách x_e a y_e se chyba přibližuje až ke 2 m, což je způsobeno zpožděním MPC při sledování trajektorie a oscilací mezi kladným a záporným směrem. Osa z_e projevuje nejmenší chybu, což je díky jednoduchému lineárnímu pohybu.



Obrázek 5.11: Rozdíl mezi trajektorií bez a s poruchou



Obrázek 5.12: Chyba řízené trajektorie oproti referenci

Simulace řízení sledování trajektorie kvadrokoptéry pomocí MPC proběhly úspěšně. Obě trajektorie jsou sledovány s minimálním vychýlením. Chybu mezi řízenou a referenční trajektorií v čase způsobuje zpoždění řízení a rychlá oscilace referenční hodnoty. největší naklonění měla kvadrokoptéra v počátku řízení, kdy se musela rozletět směrem podle reference. Porucha se v systému promítla nejvíce v místech, kde byla zavedena. MPC ale díky zpětné vazbě dron uřídil. Rozdíl mezi řízenou trajektorií s poruchou a bez poruchy je vždy v řádech setin.

V rámci této bakalářské práce jsme se zaměřili na problematiku sledování trajektorie kvadrokoptéry pomocí metody Model Predictive Control (MPC). Naše cíle byly analyzovat a navrhnout efektivní řídicí strategii pro sledování dvou určených trajektorií kvadrokoptéry. Získali jsme ucelený přehled o teoretických základech MPC a jeho aplikaci v oblasti sledování trajektorie. Identifikovali jsme klíčové aspekty a parametry letu kvadrokoptérou, které jsme později využili pro návrh prediktivního řízení. Vytvořili jsme model kvadrokoptéry, který splňoval námi určené předpoklady pro dynamiku a kinematiku. Model kvadrokoptéry jsme využili při vytváření MPC. K modelování systému a vytváření MPC jsme využili prostředí Matlab a Simulink, které jsou díky své robustnosti a přístupnosti vhodné aplikace pro modelování a řízení. Pro vytvoření optimálních řídicích problémů a hledání vhodné strategie řízení jsme využili volně přístupný balíček CasADi, který umožňuje provádět matematické operace se symbolickými proměnnými.

Do modelu byli přidány chyby měření, které by se na reálném dronu mohli projevit jako poruchy na senzorech akcelerace a úhlového zrychlení, abychom ukázali, že MPC dokáže urdit i takový systém. Prediktivní řízení dokázalo úspěšně sledovat obě zadané trajektorie s poruchou i bez ní s minimálním rozdílem. Úhly obou trajektorií potřebovali přibližně prvních 5 sekund na vyrovnání, než začali periodicky oscilovat kolem nulové referenční hodnoty. Poloha řízené trajektorie se od referenční polohy pravidelně vzdalovala, což bylo zapříčiněno časovým zpožděním, se kterým MPC sledovalo referenci. Ve finále jsme dokázali, že MPC je efektivní, stabilní a robustní způsob, kterým můžeme sledovat trajektorii.

V rámci bakalářské práce jsme ještě prozkoumali možnost využití softwaru PX4 a leteckého simulátoru Gazebo. Popsali jsme způsob řízení dronu, obsažené regulátory a otestovali je pro sledování trajektorie. Tato část je celá uvedená v dodatku A.

Ve finále můžeme konstatovat, že MPC je efektivní způsob sledování trajektorie dronu, které otevírá veliké možnosti v oblasti autonomního létání. To je také důvod, proč se dnes hojně využívá pro výzkum a aplikace.

Simulace dronu v PX4/Gazebo



V tomto dodatku prozkoumáme možnosti řízení v simulátoru PX4/Gazebo. Popíšeme metodu sledování trajektorie pomocí PID regulátorů, které jsou v simulátoru používány pro řízení kvadrokoptéry. Popíšeme implementaci algoritmu pro let podél zadaných setpointů. Zobrazíme a vyhodnotíme výsledky letu.

A.1 Sledování trajektorie pomocí PID regulátorů

V této sekci popíšeme obecný způsob sledování trajektorie kvadrokoptéry pomocí PID regulátorů. Následně zobrazíme a popíšeme specifické regulátory, které se používají v simulátoru PX4/Gazebo.

Proporcionální-Integrační-Derivační regulátory je jednoduchý a rozšířený řídicí algoritmus pro sledování trajektorie [14]. Tato metoda porovnává chybu mezi referenčním stavem w_{ref} a skutečným nebo odhadovaným stavem $y(t)$ multikoptéry

$$e(t) = w_{ref} - y(t), \quad (A.1)$$

a poté regulátory a jejich parametry upravují její chování,

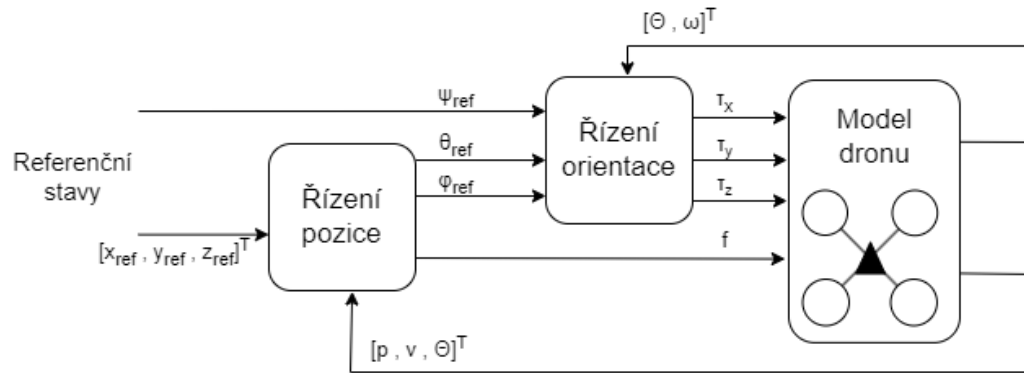
$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}, \quad (A.2)$$

kde $u(t)$ je výstup regulátoru, K_P , K_I a K_D jsou konstanty proporciální, integrační a derivační složky a $e(t)$ je regulační odchylka (A.1).

Proporciální regulátor funguje jako prostý zesilovač. Přenásobuje regulační odchylku konstantou. Problém u P regulátoru je, že nelze získat přesnou požadovanou hodnotu. Proto je v některých případech zapotřebí implementovat i integrační a derivační složku. Integrační složka řeší problém P regulátoru. Projeví se i malá odchylka, ale hrozí rozkmitání. Derivační regulátor zrychluje regulační děj, ale zvyšuje se i citlivost na šum.

Následně popíšeme návrh PID regulátorů pro sledování trajektorie dronu. Řídicí systém se obvykle skládá z dvou P regulátorů a dvou PID regulátorů. Řízení pomocí PID regulátorů je rozděleno na dva okruhy (viz obrázek A.1). Vnější okruh, kde se nachází regulátory polohy a rychlosti, a vnitřní okruh s regulátory orientace a úhlové rychlosti.

Z důvodu nedořízenosti systému zmíněné v sekci 3.3 se mohou řídit jen 4 stupně volnosti. Pohyb v osách X a Y je však výsledkem naklonění dronu v úhlech θ a ϕ , proto se jako vstup do PID regulátorů použijí souřadnice X , Y , Z a orientace ψ . Regulátor generuje referenční hodnotu pro zbylé dva úhly. Regulátor polohy generuje požadovaný tah f na základě odchylky v ose Z a podle referenčních hodnot pro úhly θ_{ref} a ϕ_{ref} .



Obrázek A.1: PID diagram vnitřního a vnějšího okruhu

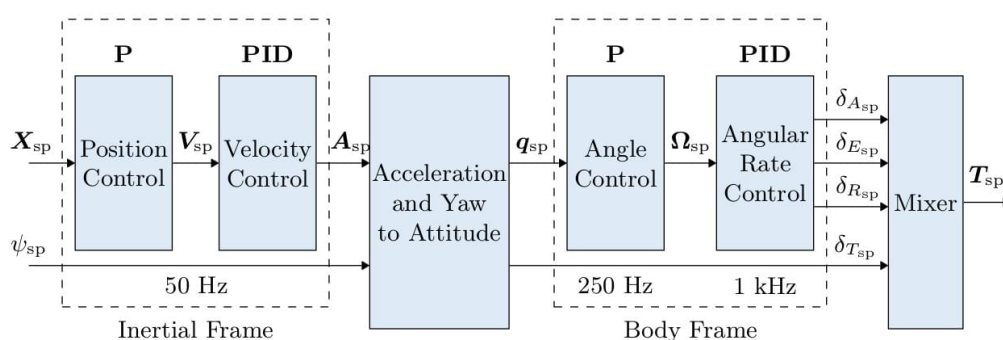
Mezi hlavní výhody PID regulátorů patří jednoduchá implementace, není potřeba vytvářet matematické modely systému, nízká výpočetní náročnost a schopnost rychlé odezvy na změny stavu dronu.

Dále si popíšeme jednotlivé regulátory ve vnitřním a vnějším okruhu, které se nachází v softwaru PX4 (viz obrázek A.2).

- **Position Control:** Proporcionální regulátor polohy, který na vstupu přijímá rozdíl referenčních hodnot souřadnic a odhadovaných hodnot souřadnic $[x_e, y_e, z_e]^T$. Výstup regulátoru jsou setpointy rychlosti (V_{sp}) v souřadnicích EFCF, které se využívají v následujícím regulátoru rychlosti.
- **Velocity Control:** PID regulátor rychlosti, který na vstupu přijímá rychlost z regulátoru polohy (V_{sp}) a odhadovanou hodnotu rychlosti (V) a řídí zrychlení (A_{sp}) dronu v souřadnicích EFCF. Pomocí získané informace o zrychlení (A_{sp}) a pomocí setpointů ψ_{sp} (yaw) dokáže vypočítat tah f a orientaci kvadrokoptéry $\Theta = [\phi, \theta, \psi]^T$.

- **Attitude Control:** P regulátor orientace využívá k popisu kvaterniony $\tilde{q} = q_0, q_1, q_2, q_3$. Regulátor řídí úhlovou rychlost (Ω_{sp}).
- **Angular Rate Control:** Jedná se o tři PID regulátory úhlové rychlosti. Výsledkem jsou aerodynamické úhlové výchylky řídicí plochy (*Aerodynamic control surface angular deflection* δ_{sp}). Pozitivní výchylka generuje negativní moment. Tento regulátor je citlivý na naladění, jelikož ovlivňuje všechny letecké režimy.

Hodnoty tahu a úhlových výchylek se pošlou do „mixéru“, který podle nich nastaví řídicí parametry (např. otáčky rotorů) a řídí let dronu.



Obrázek A.2: Diagram PID a P regulátorů pro sledování trajektorie skrze software PX4, zdroj [15]

A.2 Simulace kvadrokoptéry v Gazebo

V této sekci popíšeme softwarové aplikace, balíčky a rozhraní použité při simulaci dronu. Popíšeme jaký dron jsme použili pro let v simulátoru Gazebo, použité letecké režimy a implementaci hlavního algoritmu pro let v zadaných trajektoriích.

Pro simulace dronu a vytvoření prostředí pro experimenty bylo využito kombinace simulačního prostředí PX4 a simulátoru Gazebo. PX4 je open-source softwarový framework, který poskytuje prostředky pro simulaci a řízení dronů. Spolu s Gazebo, populárním simulačním prostředím pro robotiku a UAV, tvoří PX4-Gazebo dvojici, která umožňuje detailní a realistickou simulaci dronu v různých scénářích. Simulace v PX4-Gazebo zahrnuje reálné fyzikální vlastnosti dronu, jako je aerodynamika, gravitace a interakce s prostředím. Simulační prostředí umožňuje sledování trajektorie v různých podmínkách.

Dále byla pro sledování kvadrokoptéry použita aplikace QGroundControl. Aplikace QGroundControl je open-source softwarová platforma pro plánování, monitorování a řízení autonomních letů pro drony. Slouží jako Ground Control Station,

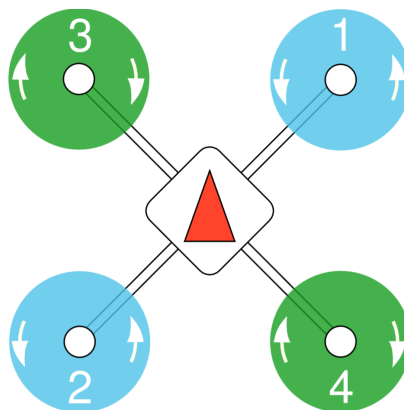
což je uživatelské rozhraní, které umožňuje komunikaci s dronem a poskytuje nástroje pro nastavení, plánování letů, monitorování stavu a vizualizaci dat.

QGroundControl poskytuje uživatelům intuitivní grafické rozhraní, které umožňuje plánování trajektorií, přesné ovládání dronu, sledování jeho polohy a rychlosti, správu úkolů a další funkce. Uživatelé mohou také nastavit parametry letu, sledovat stav baterie a signálů GPS, zobrazovat živé video z kamery dronu a získávat významné informace o jeho letové misi.

Vhodné je zmínit knihovnu MAVSDK, která byla použita pro komunikaci s dronem. MAVSDK (MAVLink SDK) je open-source knihovna, která umožňuje komunikovat s dronem pomocí protokolu MAVLink. MAVLink je komunikační protokol navržený speciálně pro autonomní lety a je široce používán v dronovém průmyslu. MAVSDK poskytuje jednoduché a konzistentní rozhraní API pro programování různých funkcí a řízení dronu z jiných softwarových prostředí nebo aplikací.

MAVSDK byl použit pro programovací jazyk C++. S pomocí MAVSDK mohou vývojáři ovládat a monitorovat dron, plánovat trajektorie, získávat telemetrická data, provádět akce a získávat informace o stavu dronu. Knihovna poskytuje také nástroje pro zachycování a zpracování událostí a poskytuje přístup ke všem důležitým informacím o letu a stavu dronu.

Abychom mohli posílat vyřešenou trajektorii z MPC v Matlabu do simulátoru Gazebo, a chceme, aby let alespoň přiměřeně odpovídal tomu v Matlabu, musíme vybrat dron se stejnou konstrukcí a alespoň podobnými parametry. Proto jsme v simulátoru Gazebo vybrali klasickou symetrickou kvadrokoptéru s těžištěm v geometrickém středu a "x"-konfigurací. Dron jsme vybrali velikostně a váhově podobný tomu namodelovanému v Matlabu 4.1. Určenou kvadrokoptéru můžeme vidět na následujícím obrázku A.3.



Obrázek A.3: Gazebo symetrická kvadrokoptéra, zdroj [16]

Pro průlet dvěma trajektoriemi jsme využili simulátoru PX4/Gazebo a QGroundControl. Algoritmus pro určení pohybu dronu byl napsán v jazyku C++ za pomoci

knihovny MAVSDK.

Pro fungující algoritmus musíme vytvořit dva soubory. Soubor *CMakeLists.txt* pro definici hlavního algoritmu, a soubor s hlavním algoritmem, v tomto případě s názvem *vrba_mpc2.cpp*.

CMakeLists specifikuje název příkladu, příznakový registr pro různé platformy, cíle, kde najít knihovny a hlavičkové soubory, zdrojové soubory k definování a jméno vygenerovaného binárního kódu.

Před popisem samotného programu je důležité vysvětlit letecké režimy použité v algoritmu.

A.2.1 Letecké režimy

Letecké režimy se rozdělují primárně na *manuální* a *autonomní*. V manuálním režimu má uživatel kontrolu nad ovládáním dronu pomocí RC ovládacích páček nebo joysticku. Autonomní režim je plně ovládán autopilotem a nepotřebuje žádného pilota nebo dálkové ovládání.

Manuální režim se poté dělí na módy "*easy*" a "*acrobatic*". "*easy*" mód upravuje ovládání dronu tak, aby se při hýbání páčky pohyboval dron pouze horizontálně. Pohyb je v tomto módu více předvídatelný a nelze dron převrátit. V "*acrobatic*" módu se pro pohyb dronu určuje míra natočení dronu, a je tím pádem možné dron přetočit.

Pro pohyb dronu po předem určené dráze zde používáme pouze autonomní režim a jeho módy.

A.2.2 Hlavní algoritmus

Algoritmus musíme připojit ke kvadrokoptěře. K tomu se využívá komunikační protokol MAVLink. Dron propojíme pomocí portu 14540, který je určený pro PX4 drony a vozidla.

Následně musíme inicializovat pluginy MAVSDK knihovny, které použijeme v algoritmu:

```
1 auto action = Action{system};
2 auto offboard = Offboard{system};
3 auto telemetry = Telemetry{system};
```

Action plugin obsahuje hotové základní úlohy pro let a přípravu k letu, jako je například vzletnutí do určené výšky, přistání nebo kontrola senzorů a zapnutí motorů na minimální otáčky.

Pomocí pluginu **Telemetry** můžeme volat a získávat různé informace týkající se dronu. Může se jednat například o stavech dronu nebo aktivovaném letovém režimu.

Offboard plugin zadává příkazy pro změnu stavu podle zadaných setpointů. Setpointy poskytuje v Offboard módu externí zdroj, který s dronem komunikuje

pomocí protokolu MAVLink. V tomto případě je externí zdroj companion computer a rozhraní MAVSDK.

Trajektorii v simulátoru řídíme skrze stavy rychlosti $[\dot{x}, \dot{y}, \dot{z}]$ a natočení ψ . Hodnoty získáme z Matlabu. MPC nalezne optimální řídicí proměnné, které jsou posílány do modelu kvadrokoptéry. Takto získaná rychlost a natočení jsou vypsané do `.csv` souboru, který se volá v tomto algoritmu. Tyto hodnoty stavu jsou poté pomocí Offboard pluginu posílány jako setpointy pro PID regulátory kvadrokoptéry.

```
1   Offboard::VelocityNedYaw let{};
2   let.north_m_s = (float) dat4[i][0];
3   let.east_m_s = (float) dat4[i][1];
4   let.down_m_s = (float) dat4[i][2];
5   let.yaw_deg = (float) dat4[i][3];
6   offboard.set_velocity_ned(let);
```

Rychlost se zde zadává v NED souřadnicích (North, East, Down), zatímco rychlost vytvořeného dronu počítá rychlost v EFCF souřadnicích. Dron v simulátoru je ale v počátku zarovnán tak, aby osa x_e směřovala na sever a osa y_e na východ, takže se v tomto případě NED a EFCF souřadnice rovnají.

Nemůžeme očekávat identickou trajektorii v modelu v Matlabu a při letu v simulátoru, jelikož modelovaný systém kvadrokoptéry a systém v simulátoru Gazebo nejsou stejné. Trajektorie je v simulátoru posílána do PID regulátorů, které je mohou ještě pozměnit. Cílem je akorát vizuálně proletět vytvořenou trajektorií.

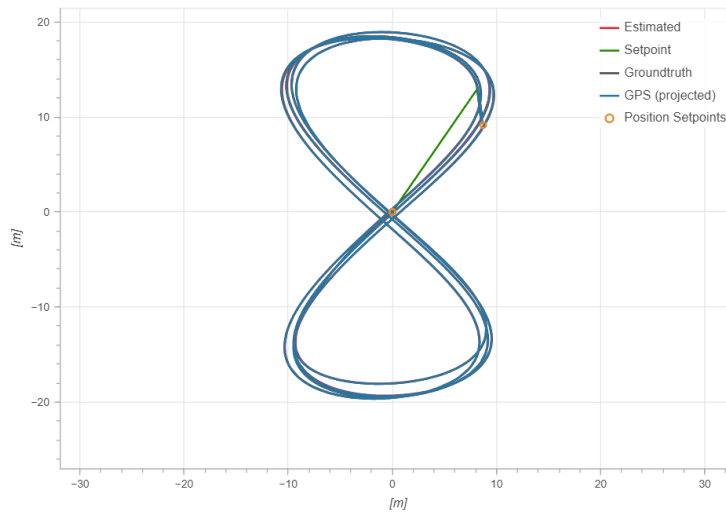
A.3 Výsledky simulátoru Gazebo

Následně předvedeme obě trajektorie prolétnuté v simulátoru PX4/Gazebo.

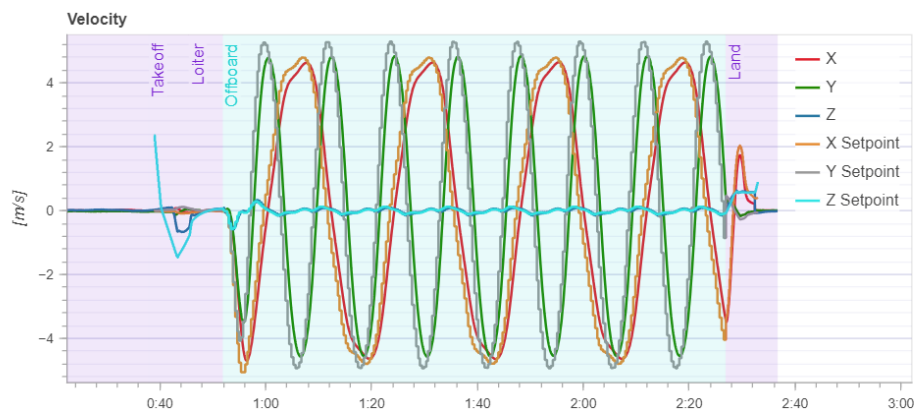
V této části zobrazíme průlet obou trajektorií díky určování setpointů rychlosti. Trajektorii jsme museli nastavovat v delších intervalech, než tomu je u řízení v Matlabu, protože když jsme měnili rychlost po 0.1 s intervalech, program se sekl a dron zůstal stát. Trajektorie je tedy viditelně větší. Stále ale napodobuje tvar, který jsme určili. Cílem letu v tomto simulátoru není správné sledování trajektorie, ale jen ukázat, jakým způsobem lze řídit dron v tomto simulátoru. Simulátor má jiný systém než máme namodelovaný my a obsahuje PID regulátory, které řídí dron podle setpointů, které mu můžeme nastavit.

Pro zobrazení grafů ze simulátoru byl využit software Flight Review od PX4, který ze záznamů letu vygeneroval následující grafy.

Na obrázku A.4 můžeme vidět průlet kvadrokoptéry amplifikovanou trajektorií ve tvaru lemniskáty. Obrázek A.5 ukazuje námi zadávané setpointy rychlosti a reálné hodnoty řízené PID regulátory.



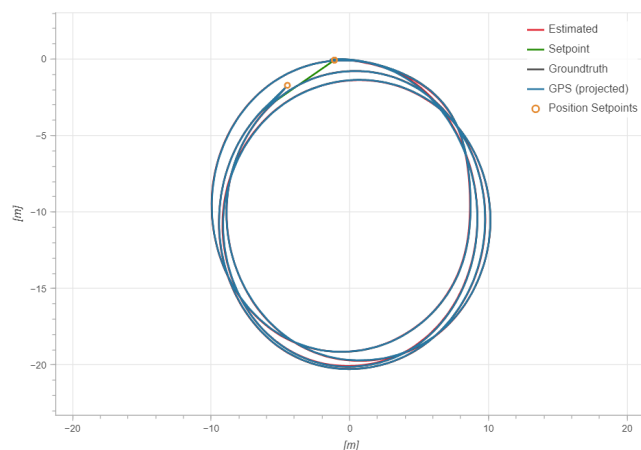
Obrázek A.4: Gazebo simulace první trajektorie



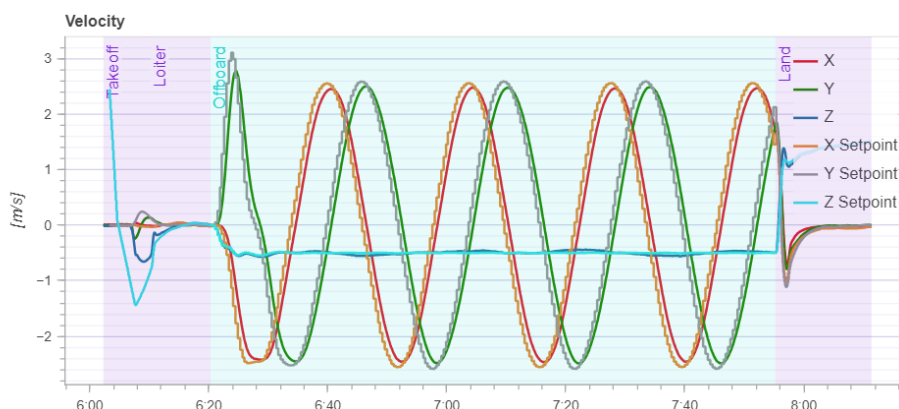
Obrázek A.5: Gazebo simulace rychlost první trajektorie

Druhou trajektorii můžeme sledovat na obrázku A.6. Zde můžeme lépe vidět výchytky které se tvoří při letu. Trajektorie není řízena v průběhu letu, ale je mu dána už předem stanovená rychlost a úhel ψ . PID regulátory už se jen starají, aby sledovali zadanou rychlost, její směr a natočení.

Obrázek A.7 sleduje zadané setpointy a PID řízenou rychlost kvadrokoptéry.



Obrázek A.6: Trajektorie spirála



Obrázek A.7: Gazebo simulace rychlost druhé trajektorie

A.4 Závěr dodatku

Vozidla dostupná v simulátoru Gazebo tvoří složité modely systémů, které jsou řízeny PID regulátory. Proto jsme neřídili přímo zpětnou vazbou let v simulátoru, ale použili jsme již obsažené sledování trajektorie z předchozí části, jejíž rychlost jsme posílali PID regulátorům jako setpointy, které mají sledovat. Pro posílání informací do kvadrokoptéry jsme využili knihovnu MAVSDK a protokol MAVLink. Setpointy jsme posílali ve větších časových intervalech, aby se simulátor nezasekl a nezkolaboval. Proto je výsledná trajektorie větší než ta původní. Simulátor úspěšně sledoval tvar obou trajektorií s menším vychýlením, které je způsobeno tím, že let nebyl řízen zpětnou vazbou.

V budoucnu je možné navrhnout MPC pro přímé řízení letu v simulátoru. Může být navrženo tak, aby posílal řídicí proměnné do zavedených PID regulátorů, nebo je obejít a řídit dron přímo.

Bibliografie

1. KIM, Kyunam; RAHILI, Salar; SHI, Xichen; CHUNG, Soon-Jo; GHARIB, Mor-teza. Controllability and Design of Unmanned Multirotor Aircraft Robust to Rotor Failure. In: 2019. Dostupné z DOI: 10.2514/6.2019-1787.
2. ROBERT NIEMIEC, Farhan Gandhi. A Comparison Between Quadrotor Flight Configurations. In: Rensselaer Polytechnic Institute, 2016.
3. QUAN, Quan. Coordinate System and Attitude Representation. In: *Introduction to Multicopter Design and Control*. Singapore: Springer Singapore, 2017, s. 99–120. ISBN 978-981-10-3382-7. Dostupné z DOI: 10.1007/978-981-10-3382-7_5.
4. LAGHARI, Asif Ali; JUMANI, Awais Khan; LAGHARI, Rashid Ali; NAWAZ, Haque. Unmanned aerial vehicles: A review. *Cognitive Robotics*. 2023, roč. 3, s. 8–22. ISSN 2667-2413. Dostupné z DOI: <https://doi.org/10.1016/j.cogr.2022.12.004>.
5. PX4. *QuadRotorAxes* [online]. [cit. 2023-07-16]. Dostupné z: https://docs.px4.io/v1.13/en/getting_started/px4_basic_concepts.html.
6. ELHESASY, Mohamed et al. Non-Linear Model Predictive Control Using CasADi Package for Trajectory Tracking of Quadrotor. *Energies*. 2023, roč. 16, č. 5. ISSN 1996-1073. Dostupné z DOI: 10.3390/en16052143.
7. AMADI, Chinedu; SMIT, Willie. *Design and Implementation of Model Predictive Control on Pixhawk Flight Controller*. 2018. Dis. pr.
8. KAMEL, Mina; STASTNY, Thomas; ALEXIS, Kostas; SIEGWART, Roland. Model Predictive Control for Trajectory Tracking of Unmanned Aerial Vehicles Using Robot Operating System. In: *Robot Operating System (ROS): The Complete Reference (Volume 2)*. Ed. KOUBAA, Anis. Cham: Springer International Publishing, 2017, s. 3–39. ISBN 978-3-319-54927-9. Dostupné z DOI: 10.1007/978-3-319-54927-9_1.

9. MEHREZ, Mohamed W. *Optimization based Solutions for Control and State Estimation in Dynamical Systems (Implementation to Mobile Robots)* [online]. [cit. 2023-07-18]. Dostupné z: https://github.com/MMehrez/MPC-and-MHE-implementation-in-MATLAB-using-Casadi/blob/master/workshop_github/MPC_MHE_slides.pdf.
10. DMYTRUK, Andriy; SILANO, Giuseppe; BICEGO, Davide; LICEA, Daniel Bonilla; SASKA, Martin. A Perception-Aware NMPC for Vision-Based Target Tracking and Collision Avoidance with a Multi-Rotor UAV. In: *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2022, s. 1668–1673. Dostupné z DOI: 10.1109/ICUAS54217.2022.9836071.
11. WIKIPEDIA. *Runge-Kutta 4th order* [online]. [cit. 2023-07-18]. Dostupné z: https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods.
12. ANDERSSON, Joel A E; GILLIS, Joris; HORN, Greg; RAWLINGS, James B; DIEHL, Moritz. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*. 2018.
13. POKU, Maame; BIEGLER, Lorenz; KELLY, Jeffrey. Nonlinear Optimization with Many Degrees of Freedom in Process Engineering. *Industrial Engineering Chemistry Research - IND ENG CHEM RES*. 2004, roč. 43. Dostupné z DOI: 10.1021/ie0341000.
14. ABDELHAY, Sabir; ZAKRITI, Alia. Modeling of a Quadcopter Trajectory Tracking System Using PID Controller. *Procedia Manufacturing*. 2019, roč. 32, s. 564–571. ISSN 2351-9789. Dostupné z DOI: <https://doi.org/10.1016/j.promfg.2019.02.253>. 12th International Conference Interdisciplinarity in Engineering, INTER-ENG 2018, 4–5 October 2018, Tirgu Mures, Romania.
15. PX4. *QuadRotorDiagram* [online]. [cit. 2023-07-06]. Dostupné z: https://docs.px4.io/main/en/flight_stack/controller_diagrams.html.
16. PX4. *QuadRotorX* [online]. [cit. 2023-07-06]. Dostupné z: https://docs.px4.io/v1.12/en/airframes/airframe_reference.html.

Seznam obrázků

2.1	Konfigurace "x" a "+"	6
3.1	Souřadnicové systémy	12
3.2	ABCF souřadnicový systém, zdroj [5]	13
3.3	Přechod z EFCF do ABCF	14
3.4	Architektura modelování multikoptéry	15
3.5	Kvadroptéra a směr otáčení rotorů	16
3.6	Tah multikoptéry	17
4.1	MPC řídicí a predikční horizont	22
4.2	Model predictive control (MPC) diagram	24
4.3	RK4 pro $\dot{y} = f(t, y)$, zdroj [11]	26
5.1	Model v Simulinku	30
5.2	Poloha p lemniskáty	33
5.3	Poloha p v čase lemniskáty	33
5.4	Orientace Θ lemniskáty	34
5.5	Rozdíl mezi trajektorií bez a s poruchou	35
5.6	Chyba řízené trajektorie oproti referenci	35
5.7	Poloha p spirály	36
5.8	Osy x a y spirály	36
5.9	Poloha p v čase spirály	37
5.10	Orientace Θ spirály	37
5.11	Rozdíl mezi trajektorií bez a s poruchou	38
5.12	Chyba řízené trajektorie oproti referenci	38
A.1	PID diagram vnitřního a vnějšího okruhu	44
A.2	Diagram PID a P regulátorů pro sledování trajektorie skrze software PX4, zdroj [15]	45
A.3	Gazebo symetrická kvadroptéra, zdroj [16]	46
A.4	Gazebo simulace první trajektorie	49
A.5	Gazebo simulace rychlost první trajektorie	49

Seznam obrázků

A.6	Trajektorie spirála	50
A.7	Gazebo simulace rychlost druhé trajektorie	50

