



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUT OF SOLID MECHANICS, MECHATRONICS AND
BIOMECHANICS

REALIZACE ALGORITMU AUTOPILOTA PRO BEZPILOTNÍ LETADLO SKYDOG

IMPLEMENTATION OF AUTOPILOT ALGORITHM FOR AUTONOMOUS AIRCRAFT SKYDOG

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. JAKUB NĚMEČEK

VEDOUCÍ PRÁCE
SUPERVISOR

DOC. ING. JIŘÍ KREJSA, PH.D.

BRNO 2014

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav mechaniky těles, mechatroniky a biomechaniky

Akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Jakub Němeček

který/která studuje v **magisterském navazujícím studijním programu**

obor: **Mechatronika (3906T001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Návrh autopilota bezpilotního letounu

v anglickém jazyce:

Design of fixed-wing UAV autopilot

Stručná charakteristika problematiky úkolu:

Při ovládání moderních letadel se stále častěji využívají systémy, které pilotům usnadňují práci nebo jejich práci zcela nahrazují. Na palubě těchto letadel často není potřeba lidské posádky. Tímto způsobem můžeme automatizovat letové procesy nebo minimalizovat vliv lidského faktoru při těchto procesech (např.: stabilizace rovného letu v letové hladině, koordinovaná zatáčka v letové hladině, přistání a vzletnutí a další). Letadlo se pohybuje autonomně pod dohledem osoby na pozemní základně. Plně bezpilotní letadla jsou určena převážně pro sledování daného prostoru nebo pro účely vývoje.

Ovládání a kontrola těchto letadel je založena na výpočtových algoritmech autopilota. Tyto jsou nejdříve otestovány na modelu letadla ve vhodném simulačním prostředí a poté implementovány do řídicí elektroniky letadla.

Při návrhu autopilota je vhodné využít simulační prostředí s modelem letadla. Pro tento účel poslouží model letadla SkyDog. Tato práce je součástí projektu SkyDog zadaného firmou Honeywell Aerospace.

Cíle diplomové práce:

1. Přehled užívaných algoritmů autopilotů
2. Výběr vhodného algoritmu pro implementaci v bezpilotním letadle SkyDog
3. Vytvoření modelu vybraného algoritmu ve vhodném simulačním prostředí
4. Ověření funkčnosti autopilota na výpočtovém modelu letadla SkyDog

Seznam odborné literatury:

Reg Austin, Unmanned Aircraft Systems: UAVS Design, Development and Deployment, John Wiley & Sons, 2011

R.S.Christiansen: Design of an autopilot for small unmanned aerial vehicles, Brigham Young University, 2004

Vedoucí diplomové práce: doc. Ing. Jiří Krejsa, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2013/2014.

V Brně, dne 19.11.2013

L.S.

prof. Ing. Jindřich Petruška, CSc.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc., dr. h. c.
Děkan fakulty

Abstrakt

Práce přináší přehled užívaných řešení autopilotů pro malá bezpilotní letadla a analýzu jednotlivých jeho částí. Součástí práce je implementace algoritmu autopilota v prostředí Matlab/Simulink s cílem ovládat výpočetní model letadla. Práce popisuje vyšší vrstvy řízení a navigace letadla, konkrétně systém plánování trajektorie.

Výsledný algoritmus je navržen jako sledování referenčního bodu pohybujícího se mezi letadlem a zadaným bodem zájmu. Součástí algoritmu je řešení speciálních poloh bodů a chybového hlášení.

Abstract

This thesis describe approaches for autopilots for small unmanned aerial vehicles. Part of thesis is implementation of autopilot in Matlab/Simulink. The goal is to control computation model of the aircraft. Main part of thesis describes high layer of control and navigation with focus on trajectory planning.

Final algorithm is based on following the reference point between the aircraft and waypoint. Algorithm solves special positions of waypoints and error flag.

Klíčová slova

algoritmus autopilota, řízení letadel, modelování v prostředí Matlab/Simulink

Keywords

algorithm of autopilot systems, aircraft control, modeling in Matlab/Simulink

Bibliografická citace

Bc. NĚMEČEK, J. Realizace algoritmu autopilota pro bezpilotní letadlo skydog. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014. 79 s. Vedoucí diplomové práce doc. Ing. Jiří Krejsa, Ph.D.

Čestné prohlášení

Prohlašuji, že jsem diplomovou práci na téma *Realizace algoritmu autopilota pro bezpilotní letadlo SkyDog* vypracoval samostatně s použitím odborné literatury a pramenů uvedených v seznamu, který tvoří přílohu této práce.

21.5.2014

.....
Bc. Jakub Němeček

Poděkování

Děkuji doc. Ing. Jiřímu Krejsovi, Ph.D. za vedení, rady a poznámky při vypracování diplomové práce.

Dále svým spolupracovníkům Bc. Libor Příleský, Bc. Vojtěch Kuchař a Bc. David Kraus, kteří se mnou pracovali na dalších částech projektu SkyDog.

V neposlední řadě rodině a blízkým za podporu a motivaci ke studiu.

Obsah

1 Úvod do řízení a navigace letadla.....	6
1.1 Projekt SkyDog.....	6
2 Přehled užívaných řešení autopilota.....	7
2.1 Vektorové pole rychlostí.....	7
2.2 Využití inverzní dynamiky.....	11
2.2.1 Dynamické rovnice letadla.....	11
2.3 Rozložení na podélný a příčný směr.....	14
2.4 Jednoduché adaptivní řízení	15
2.4.1 Výpočet laterálního zrychlení a žádaného úhlu náklonu.....	16
2.4.2 Výpočet žádané výšky letu a úhlu sklonu.....	18
2.4.3 Výpočet polohy referenčního bodu P	19
2.4.4 Princip detekce průletu bodem zájmu.....	20
3 Žádané funkce autopilota letadla SkyDog.....	22
3.1 Přímý let na bod.....	22
3.2 Speciální poloha bodů.....	22
3.3 Automatické přepínání bodů zájmu	22
3.4 Online aktualizace bodů zájmu.....	23
3.5 Reakce na aktuální polohu letadla.....	23
3.6 Chybové stavy.....	23
3.7 Nastavitelné parametry algoritmu.....	23
4 Vybraná metoda řešení.....	24
4.1 Řešení řízení závislé na poloze letadla.....	24
4.1.1 Poloha referenčního bodu.....	24
4.1.2 Pohyb kolem domácí stanice.....	25
4.2 Řešení speciálních poloh bodů.....	25
4.2.1 Nadměrná změna výšky.....	25
4.2.2 Poloha domácí stanice.....	26
4.2.3 Dva body přesně nad sebou.....	26
4.3 Řešení chybových stavů.....	26
4.3.1 Možnosti vybavení chybového stavu.....	27
4.4 Proměnné parametry.....	27
4.4.1 Vzdálenost referenčního bodu před letadlem.....	27
4.4.2 Poloměr kruhové trajektorie.....	27
4.4.3 Dovolená odchylka od bodu.....	27
4.5 Popis navrženého algoritmu.....	27
5 Realizace autopilota v Matlab/Simulink.....	30
5.1 Vstupní data.....	30
5.1.1 Přepočet GPS na vzdálenosti od domácí stanice.....	32
5.2 Součásti řídicí vrstvy autopilota.....	32
5.3 Výstupní hodnoty.....	33
6 Ověření funkčnosti řešení v simulačním prostředí.....	35
6.1 Použitý model letadla k ověření.....	35
6.2 Letové manévry.....	35
6.2.1 Přímá trajektorie s pravoúhlou zatáčkou.....	35
6.2.2 Speciální polohy bodů.....	37
6.2.3 Reakce na přepnutí z manuálního modu.....	38
6.2.4 Vybavení chyby.....	39
6.2.5 Změna parametrů.....	41
6.2.5.1 Změna L.....	41

6.2.5.2 Změna R.....	43
6.2.5.3 Změna „threshold“.....	46
7 Zhodnocení výsledných dat	49
7.1 Výsledky simulace letu v Matlab/Simulink.....	49
7.1.1 Přímá trajektorie s pravoúhlou zatáčkou.....	49
7.1.2 Kruhová trajektorie kolem bodu.....	51
7.1.3 Změna parametrů algoritmu.....	52
7.1.3.1 Chování při změně vzdálenosti referenčního bodu - L.....	52
7.1.3.2 Chování při změně poloměru krouživé trajektorie - R.....	57
7.1.3.3 Chování při změně hodnoty threshold.....	59
7.2 Výsledky hardware in the loop simulace.....	62
8 Závěr.....	64
8.1 Možné další vylepšení autopilota.....	64
9 Použitá literatura.....	65
10 Seznam ilustrací a tabulek.....	67
11 Seznam značek a symbolů.....	69
12 Přílohy.....	72
12.1 Příloha A – geometrie letadla.....	72
12.2 Příloha B – celkové schema autonomního letadla.....	73

1 Úvod do řízení a navigace letadla

Snaha člověka zdokonalit automatické systémy zasáhla veškeré odvětví činnosti lidí od jednoduchých manuálních úkonů přes složitější procesy, řídicí a navigační algoritmy a v neposlední řadě automatické řízení mobilních prostředků. Příležitost sestavit stroj, který se pohybuje samostatně, byla pro člověka velká výzva, stejně jako samotná konstrukce stroje plně řízeného člověkem. Jedná se nejen o pohyb po Zemi, ale také ve vzduchu.

Při pohledu do historie je zřejmé, že samotné letadlo je poměrně nový vynález. Od prvních pokusů pohybovat se nad Zemí (bratři Wrightové) lidé postoupili k vývoji rychlejších strojů schopných přepravovat několik lidí či náklad. Účel přepravy není zdaleka posledním využitím letadel. V 1. polovině 20. století zaznamenalo letectví vývoj pro zásobovací a vojenské účely.

Dále se v letadlech začaly objevovat první mechanické systémy podpory pilota, které postupně přešly v elektronické. Tyto systémy nejdříve pomáhaly pilotům zjednodušit práci a postupně začaly jejich činnost nahrazovat v přehledných a předvídatelných situacích letu. Takto vznikly první algoritmy autopilotů využívající informace ze senzorů a zasahující do řízení pomocí aktuátorů řídicích prvků. S vývojem těchto systémů se dostáváme do fáze, kdy na palubě letadla není potřeba posádka, tedy bezpilotních strojů.

Základem pro návrh řízení bezpilotního letadla je pochopení a osvojení dynamiky letadel a způsob řízení bez podpory autopilotních systémů. Po pochopení těchto základů lze přemýšlet o vývoji letadla, které je schopné letět pouze s využitím senzorických systémů a řídicích prvků. S rozvojem senzorických a výpočetních systémů jsme schopni letadlo ovládat na dálku nebo nechat letadlo manévrovat autonomně podle nadřazených příkazů člověka.

Práce je založena na zpracování dat ze senzorů a vstupních hodnot zadaných obsluhou letadla. Cílem je vypočítat hodnoty sloužící jako vstupní data do regulátorů žádaných stavů letadla takové, aby se letadlo pohybovalo po zadané trajektorii.

Toto vedlo náš team k projektu SkyDog.

1.1 Projekt SkyDog

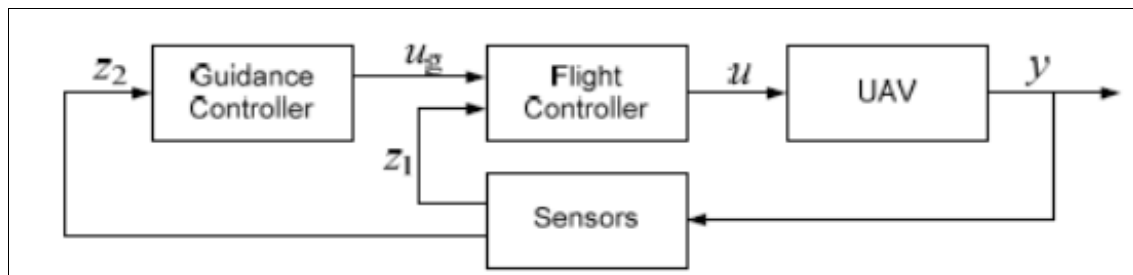
Tato práce je součástí projektu SkyDog, který má původ ve společnosti Honeywell, spol. s r.o. . Projekt se zabývá vývojem testovací platformy autonomního letadla. Výstupem z projektu je letadlo s navrženou elektronikou a možností navrhnout algoritmy autopilota jednoduše v prostředí Simulink. Z tohoto prostředí vygenerovat C kód těchto algoritmů. Výsledný C kód je využit pro řízení letadla dle zadaného seznamu bodů v GPS souřadnicích.

Projekt je natolik komplexní, že byl rozdělen na více částí.

- Simulační modelování bezpilotního letounu – Bc. Libor Přileský (zdroj [8])
- Návrh a implementace SW architektury bezpilotního letounu – Bc. Vojtěch Kuchař (zdroj [9])
- Návrh mechanického a elektrického subsystému bezpilotního letounu – Bc. David Kraus (zdroj [10])
- Realizace algoritmu autopilota pro bezpilotní letadlo Skydog – Bc. Jakub Němeček

2 Přehled užívaných řešení autopilota

Většina popsaných algoritmů se řídí následujícím schematem



Ilustrace 1: celkové jednoduché schema letadla s autopilotem (zdroj [2])

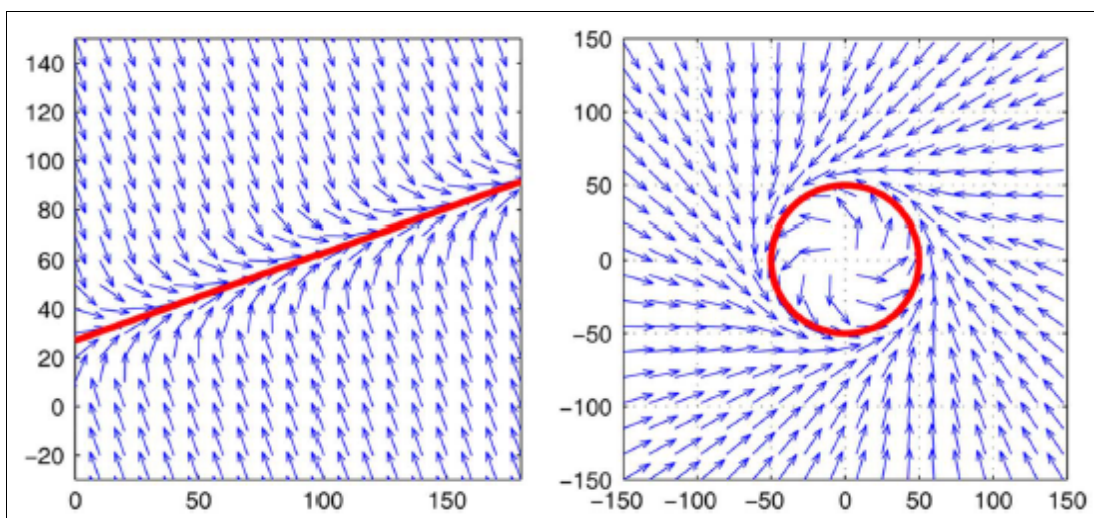
Systém samostatného řízení je rozdělen na více funkcí. K aktuátorům (UAV) je přiveden kontrolní signál (u) ze systému Flight Controller. Tento systém obsahuje regulátory pro jednotlivé ovládací prvky. Regulátory reagují na příkazy ze systému Guidance controller, který reprezentuje nadřazené řízení. Tento systém zpracovává příkazy obsluhy z pozemní stanice, porovnává je s daty ze systému navigačních senzorů (z_2) a na základě těchto hodnot počítá žádané hodnoty pro regulátory dle navrženého algoritmu. UAV je v tomto projektu letadlo s horním křídlem a vrtulí vpředu. Práce se zabývá vnitřním algoritmem systému Guidance controller.

Letadlo, stejně jako automobil, je neholonomní soustava a podle toho je třeba navrhovat řízení. Řízení těchto soustav je založeno na řízení rychlostí. Základní rychlosti letadla jsou rychlost vzhledem k Zemi, rychlost proudění vzduchu kolem křídel a rychlost okolního prostředí (vítr). Vítr také vstupuje do modelu letadla jako rušivý vliv, který v *Ilustraci 1* není zobrazen.

V dalších podkapitolách jsou popsány možné přístupy k výpočtu žádaných hodnot pro regulátory (u_g).

2.1 Vektorové pole rychlostí

Kolem žádané trajektorie se vytvoří vektorové pole žádaných směrů letu v okolí trajektorie. Trajektorie je rozložena na jednoduché části typu rovný usek nebo oblouk. Na základě polohy vzhledem k této trajektorii je počítán úhel letového kurzu, který je vstupem do regulátorů jako žádaná hodnota.



Ilustrace 2: zobrazení navrženého vektorového pole pro přímý let a pro kruhovou trajektorii (zdroj [1])

Při výpočtu složek rychlosti letadla v osách souřadného systému spojeného se Zemí uvažujeme závislost na proudění vzduchu kolem křídel letadla a vektoru větru (zobrazeno v Ilustrace 3):

$$\begin{aligned}\dot{x} &= V_a \cos \psi + W_x \\ \dot{y} &= V_a \sin \psi + W_y\end{aligned}\quad (1)$$

Kde: $\dot{x}, \dot{y}$ - složky rychlosti v souřadném systému Země (m/s)
 V_a - proudění vzduchu kolem letadla (airspeed; m/s)
 ψ - tzv. heading – nastavený kurz letu bez započítání vlivu větru (rad)
 W - vektor směru větru (rozdělen do složek v jednotlivých osách; m/s)

Při zpracování dat je vhodné přepočítat tyto rovnice na rychlost vzhledem k Zemi (GPS speed) a kurz letu (azimuth).

$$V_{ai} + W_i = V_{gi}\quad (2)$$

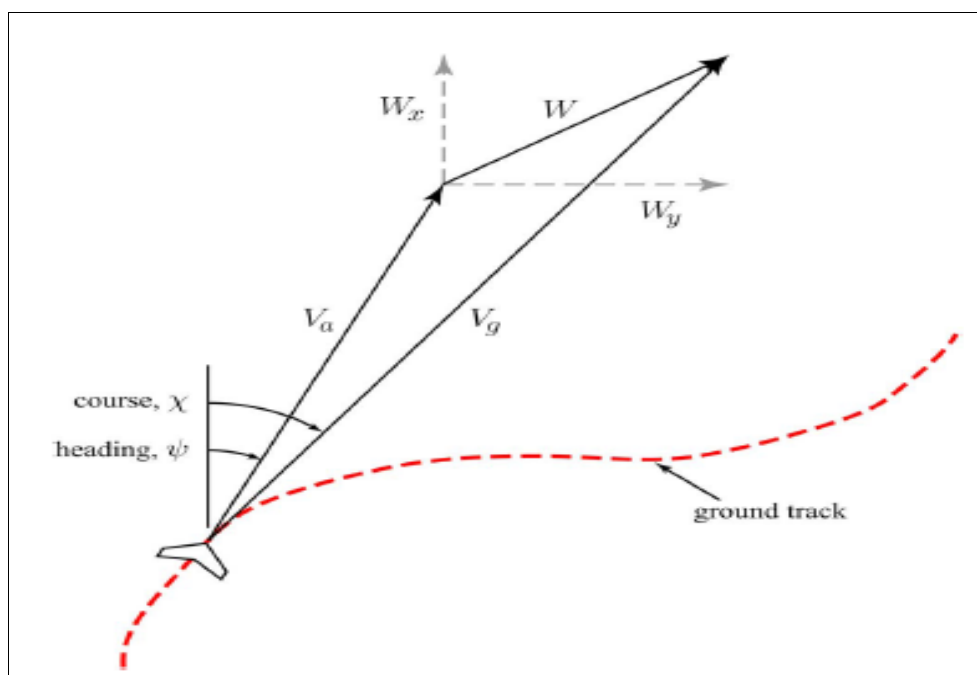
Kde: i - index osy (x,y; -)
 V_{ai} - složka vektoru proudění ve zvolené ose (m/s)
 W_i - složka vektoru větru ve zvolené ose (m/s)
 V_{gi} - složka vektoru rychlosti vzhledem k Zemi ve zvolené ose (m/s)

Při využití tohoto přepočtu získáme rovnice složek rychlosti letadla.

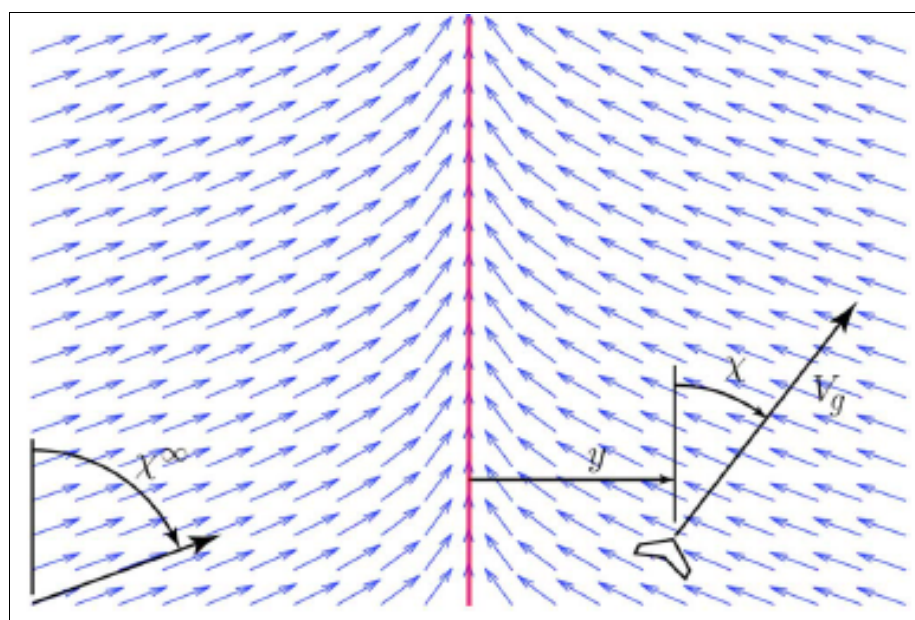
$$\begin{aligned}\dot{x} &= V_{gx} \cos \chi \\ \dot{y} &= V_{gy} \sin \chi\end{aligned}\quad (3)$$

Kde: χ - azimuth, skutečný směr letu vzhledem k zemskému severu (rad)

Trajektorie se rozdělí na rovné a obloukové úseky a vypočte se žádaný kurz v místě kde se nachází letadlo. Výpočet žádaných kurzů se tímto dělí na dvě části. Kurz vypočtený kolem rovné a kolem obloukové trajektorie. Při řešení oblouku musíme rozlišit vzdálenost letadla od středu oblouku (menší nebo větší než poloměr oblouku).



Ilustrace 3: grafické zobrazení jednotlivých vektorů rychlostí a úhlů pro výpočet (zdroj [1])



Ilustrace 4: grafické zobrazení pro výpočet pole kolem rovné trajektorie (zdroj [1])

Pro rovnou trajektorii:

$$\chi^d(y) = -\chi^\infty \frac{2}{\pi} \operatorname{atan}(ky)$$

$$\chi^d(y) \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \quad (4)$$

Kde: χ^d - žádané hodnoty kurzu závislé na y (rad)

χ^∞ - žádaná hodnota v nekonečné vzdálenosti od trajektorie (rad)

y - vzdálenost od žádané trajektorie (m)

k - konstanta určující přechod od kurzu v nekonečné a nulové vzdálenosti od trajektorie (-)

Pro obloukovou trajektorii:

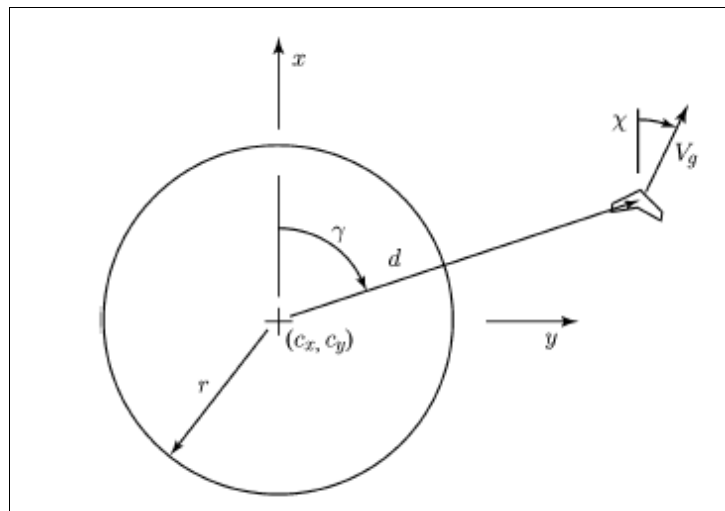
$$\chi^d(d) = \gamma - \frac{\pi}{2} - \operatorname{atan}(k(d-r)) \quad (5)$$

Kde: d - vzdálenost letadla od středu oblouku (m)

r - poloměr oblouku trajektorie (m)

γ - úhlová pozice vzhledem ke spojnici bodu C a zemského severu (rad)

k - konstanta určující přechod od kurzu v nekonečné a nulové vzdálenosti od trajektorie (-)



Ilustrace 5: zobrazení polohy letadla vzhledem k obloukové trajektorii (zdroj [1])

Zhodnocení metody

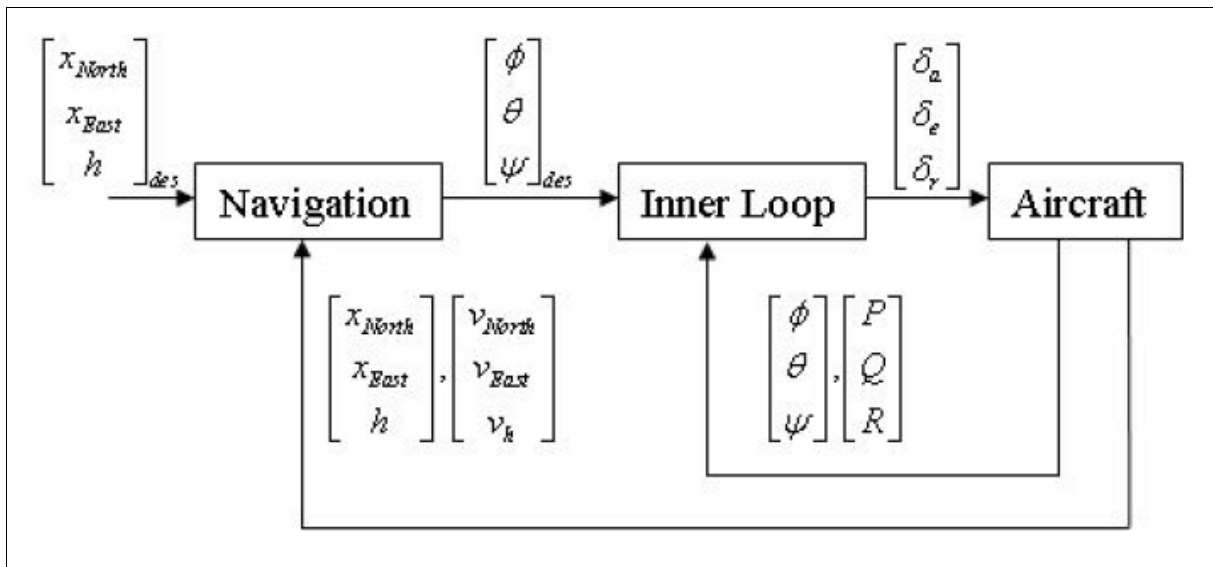
Problém s touto metodou nastává při složitější trajektorii. Je možné každou křivku rozdělit na jednoduché rovné či obloukové části, ale je nutné detekovat, zda se letadlo nachází v oblasti plánování dle rovné nebo obloukové části.

Dalšími výpočty lze žádanou trajektorii optimalizovat dle volených parametrů (např.: nejkratší dráha, přesnost dosažení bodu, atd.).

Celkový výpočet je uveden ve zdroji [1].

2.2 Využití inverzní dynamiky

Návrh řízení na základě inverzní dynamiky letadla je založen na popisu letadla pomocí dynamických rovnic.



Ilustrace 6: schema řízení metodou inverzní dynamiky (zdroj [3])

V Ilustrace 6 je vidět rozdělení řízení na vnitřní smyčky (Inner Loop - regulátory) a vnější smyčku navigace. Vstupem do navigace je vektor hodnot určující bod v trajektorii. Bod je popsán hodnotou severní šířky (x_{North}), východní délky (x_{East}) a výšky letu (h). Tyto hodnoty jsou v bloku „Navigation“ porovnány s naměřenou hodnotou aktuální pozice letadla. Do navigace vstupují také měřené hodnoty rychlostí v severním, východním směru a rychlost změny výšky letu.

Z těchto dat jsou algoritmem navigace spočítány žádané hodnoty eulerových úhlů roll (náklon) Φ , pitch (sklon) Θ a yaw (odklon) Ψ . Tyto žádané hodnoty jsou porovnány s naměřenými hodnotami. Nastavením regulátorů pro eulerovy úhly se počítá žádaná výchylka na řídicích plochách letadla.

2.2.1 Dynamické rovnice letadla

Silové rovnice:

$$\begin{aligned}\dot{U} &= RV - QW - g \sin \theta + (X_A + X_T)/m \\ \dot{V} &= -RU + PW + g \sin \phi \cos \theta + (Y_A + Y_T)/m \\ \dot{W} &= QU - PV + g \cos \phi \cos \theta + (Z_A + Z_T)/m\end{aligned}\tag{6}$$

Kinematické rovnice:

$$\begin{aligned}\dot{\phi} &= P + \tan \theta (Q \sin \theta + R \cos \theta) \\ \dot{\theta} &= Q \cos \phi - R \sin \phi \\ \dot{\psi} &= (Q \sin \phi + R \cos \phi) / \cos \theta\end{aligned}\tag{7}$$

Momentové rovnice:

$$\begin{aligned}
\dot{P} &= \frac{1}{I_{xx}} (l - Q I_{zz} R + R I_{yy} Q) \\
\dot{Q} &= \frac{1}{I_{yy}} (-m + R I_{xx} P - P I_{zz} R) \\
\dot{R} &= \frac{1}{I_{zz}} (n - P I_{yy} Q + Q I_{xx} P)
\end{aligned} \tag{8}$$

Rovnice navigace:

$$\begin{bmatrix} \dot{p}_N \\ \dot{p}_E \\ \dot{h} \end{bmatrix} = \begin{bmatrix} c \phi c \psi & -c \phi s \psi + s \phi s \theta c \psi & s \phi s \psi + c \phi s \theta c \psi \\ c \phi s \psi & c \phi c \psi + s \phi s \theta s \psi & -s \phi c \psi + c \phi s \theta s \psi \\ s \theta & -s \phi c \theta & -c \phi c \theta \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{9}$$

Význam symbolů v rovnicích 6 - 9:

$[U, V, W]'$	m/s	Rychlosti v souřadném systému letadla
$[X_A, Y_A, Z_A]'$	N	Aerodynamické síly v souřadném systému letadla
$[X_T, Y_T, Z_T]'$	N	Tahové síly letadla (tah, vztlak)
$[P, Q, R]'$	rad/s	Úhlové rychlosti kolem os souřadného systému letadla
$[I_{xx}, I_{yy}, I_{zz}]'$	kg m ²	Momenty setrvačnosti letadla
$[l, m, n]'$	Nm	Aerodynamické momenty k souřadnému systému letadla
$[\phi, \theta, \psi]'$	rad	Eulerovy úhly kolem os souřadného systému letadla
$[p_N, p_E, h]'$	m	Vektor určující polohu letadla vzhledem k Zemi
g	m/s ²	Gravitační zrychlení
m	kg	Hmotnost

V rovnicích navigace jsou symboly „c“ a „s“ zkrácené označení pro goniometrické funkce cos a sin.

Na základě žádaného kurzu letu lze pomocí takto popsané dynamiky spočítat žádané hodnoty Eulerových úhlů, které tvoří žádané vstupy do regulátorů.

$$\begin{aligned}
\chi &= \text{atan}\left(\frac{V_{g, East}}{V_{g, North}}\right) \\
\chi_{des} &= \text{atan}\left(\frac{\chi_{EastD} - \chi_{East}}{\chi_{NorthD} - \chi_{North}}\right)
\end{aligned} \tag{10}$$

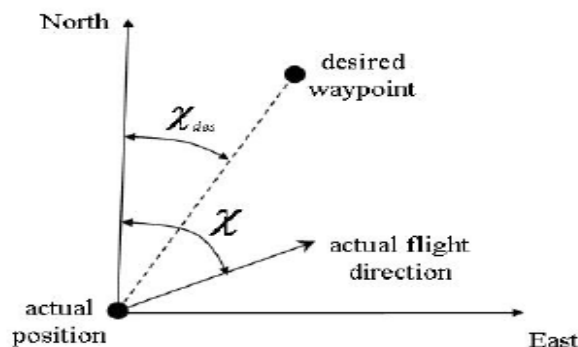
Kde: V_g - rychlost vzhledem k Zemi ve směru zeměpisných os (sever, východ; m/s)

$\chi_{East}, \chi_{North}$ - aktuální pozice letadla ve složkách dle zeměpisných os (m)

$\chi_{EastD}, \chi_{NorthD}$ - poloha žádaného bodu ve složkách dle zeměpisných os (m)

Po sestavení inverzních kinematických a momentových rovnic a doplnění o výpočet aerodynamických sil můžeme přímo spočítat hodnoty výchylek ovládacích ploch.

Aerodynamické síly a momenty se spočítají z geometrie letadla a okolního prostředí pomocí rovnic:



Ilustrace 7: pozice letadla a žádaného bodu (zdroj [3])

Aerodynamické momenty:

$$\begin{bmatrix} l \\ m \\ n \end{bmatrix} = \rho \frac{V_a^2}{2} \begin{bmatrix} S b C_l(\beta, \delta_a, \delta_r, P, R) \\ S c C_m(\alpha, \beta, \delta_e, Q) \\ S b C_n(\beta, \delta_a, \delta_r, P, R) \end{bmatrix} \quad (11)$$

Aerodynamické síly:

$$\begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix} = C_{BW} \begin{bmatrix} C_D(C_L, \delta_e, \delta_a, \delta_r) \\ C_Y(\beta, \delta_a, \delta_r, P, R) \\ C_L(\alpha, \delta_e, Q) \end{bmatrix} \frac{S \rho V_A^2}{2} \quad (12)$$

Transformační matice (ze souř. systému aerodynamického do souř. systému letadla):

$$C_{BW} = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \cos \beta & \cos \alpha \end{bmatrix} \quad (13)$$

Význam symbolů v rovnicích 11 - 13:

Symbol		Význam
$[l, m, n]'$	N m	Aerodynamické momenty v souřadném systému letadla
$[X_A, Y_A, Z_A]'$	N	Aerodynamické síly v souřadném systému letadla
$[P, Q, R]'$	rad/s	Úhlové rychlosti kolem os souřadného systému letadla
$[C_l, C_m, C_n]'$	-	Roll, pitch, yaw momentové koeficienty
$[C_D, C_Y, C_L]'$	-	Drag, side, lift silové koeficienty
$[\delta_a, \delta_e, \delta_r]$	rad	Výchylky ovládacích ploch (a – ailerons, e – elevator, r – rudder)
V_A	m/s	Rychlost letadla vzhledem k okolnímu prostředí (airspeed)
ρ	kg/m ³	Hustota okolního prostředí
S	m ²	Plocha křídel
b, c	m	Rozpětí křídel, Délka trupu
α, β	rad	Úhel náběhu (angle of attack), Úhel skluzu (backslide angle)

Zhodnocení metody

Tato metoda vychází ze znalosti geometrie a aerodynamických koeficientů letadla. Tyto hodnoty pro SkyDog jsou k dispozici, ale pro návrh řízení mohou být nepřesné. Metoda je výpočetně náročná.

Kompletní výpočet uveden ve zdroji [3].

2.3 Rozložení na podélný a příčný směr

Tímto způsobem se rozloží složitější problém řízení ve 3D prostoru na dva rovinné problémy. Dynamiku letadla lze popsat zvlášť v podélném (longitudálním) a příčném (laterálním) směru.

Longitudální pohyb je řešen rychlostí a sklonem letadla (pitch). Tento pohyb je ovládán výškovkou (elevator) a tahem motoru (thrust). Při velké odchylce od žádané výšky řeší pohyb regulátor tahu motoru a sklon letadla je vytrimován pro efektivní a bezpečné stoupání či klesání. Při přiblížení k žádané výšce je tato výška kontrolována změnou úhlu sklonu.

Laterální pohyb je řešen náklonem (roll) a odklonem letadla (yaw). Tento pohyb je ovládán křídélky (aileron) a směrovým kormidlem (rudder). Laterální pohyb letadla je řízen na základě příkazů obsluhy. Tyto příkazy mají formu bodů v GPS souřadnicích, kterými má letadlo proletět.

Z aktuální polohy letadla a zadaných bodů zájmu (waypoints) se vypočítají hodnoty žádaných veličin pro regulátory. Ty lze navrhnout více způsoby. Lze řídit kurz letu, výšku a rychlost nebo úhel sklonu (roll angle), výšku a rychlost. Oba způsoby reprezentují výše popsany princip rozdělení na dvě podúlohy.

Tato metoda nepopisuje přímo algoritmus řízení a navigace, ale je vhodná k pochopení principu těchto algoritmů.

Body mohou být zadány různými způsoby. Základní z nich jsou popsány v *Tabulce 1.* (tabulka dle zdroje [4])

Zhodnocení metody

Princip popsany v této kapitole je vhodný k řízení a navigaci letadla SkyDog. Metoda není matematicky podložena, jde jen o princip přístupu k řízení a navigaci.

Tato metoda vychází ze zdroje [4].

Příkaz	Zadání
Let' na XY	Body jsou zadány v metrech na východ a na sever vzhledem k domácí stanici a výška bodu v m.n.m.
Let' na GPS	Body jsou zadány ve stupních zeměpisné délky a šířky a výška bodu v m.n.m.
Kruh okolo XY	Je zadán bod dle „Let' na XY“ a radius s jakým má letadlo kroužit kolem tohoto bodu v metrech.

Kruh okolo GPS	Je zadán bod dle „Let’ na GPS“ a radius s jakým má letadlo kroužit kolem tohoto bodu v metrech.
Přistaň na XY	Přistávací sekvence s cílem v bodě. Bod je zadán vzdáleností severně a východně od domácí stanice v metrech, výškou m.n.m a nastavenou sestupnou trajektorií (hlavně žádaná výška a rychlost).
Vzlétni	Vzletová sekvence z aktuální pozice letadla. Vzletová sekvence je zadána příkazy pro rovné vzletnutí do stanovené výšky a poté převezme kontrolu navigace na další bod v seznamu.
Opakuj	Příkaz pro opakovaný průlet stanovenými body v seznamu.

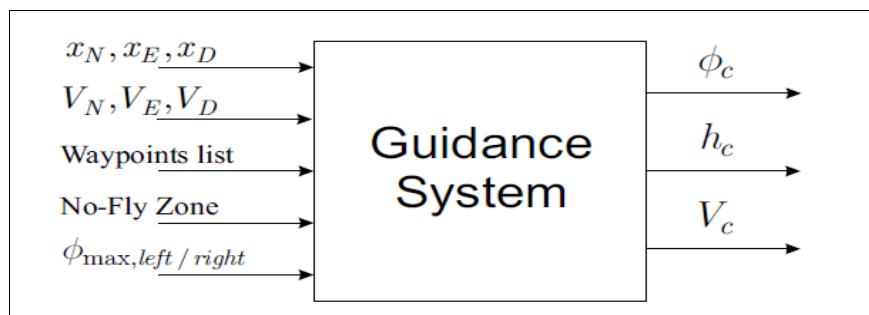
Tabulka 1: Způsob zadávání bodů

2.4 Jednoduché adaptivní řízení

Metoda je založena na nelineárním řízení laterálního pohybu. Řídící hodnota laterálního pohybu je zde úhel náklonu (roll angle), který je počítán pomocí laterální akcelerace dle pohybu po kruhové trajektorii.

Vstupy do řízení jsou rychlost letadla (V), aktuální poloha letadla (x), seznam bodů (Waypoints list), zadaná bezletová zona (No-Fly Zone) a maximální hodnoty úhlu náklonu ($\theta_{\max}$). Hodnoty rychlosti a polohy jsou rozloženy do jednotlivých os dle zeměpisné šířky (N), délky (E) a výšky (D) od hladiny moře (m.n.m.). Algoritmus počítá žádaný úhel náklonu (θ_c), žádanou letovou výšku (h_c) a rychlost (V_c).

Hlavní výpočet spočívá ve zpracování směru rychlosti letadla (V_g) a žádané trajektorie. Na základě těchto hodnot se spočte žádané laterální zrychlení, které se převede na žádaný úhel náklonu (roll angle).



Ilustrace 8: zobrazení vstupů a výstupů řízení (zdroj [5])

2.4.1 Výpočet laterálního zrychlení a žádaného úhlu náklonu

Laterální zrychlení dle pohybu po kružnici:

$$a_l = \frac{V_g^2}{R} \quad (14)$$

Kde: a_l - žádané laterální zrychlení (m/s^2)

V_g - celková rychlost letadla vzhledem k zemi (m/s)

R - žádaný poloměr letu (m)

Při pohledu na *Ilustraci 9* je pro laterální zrychlení důležitý vektor L_1 reprezentující polohu referenčního bodu vzhledem k souřadnému systému spojenému s letadlem.

Délku L_1 lze spočítat z geometrie letu:

$$\begin{aligned} L_1 &= 2 R \sin \theta_a \\ L_1 &= 2 R \cos \gamma \end{aligned} \quad (15)$$

Na *Ilustraci 9* lze vidět, že:

$$\gamma = \frac{\pi}{2} - \eta \quad (16)$$

Potom lze napsat:

$$\begin{aligned} L_1 &= 2 R \sin \eta; \\ R &= \frac{L_1}{2 \sin \eta} \end{aligned} \quad (17)$$

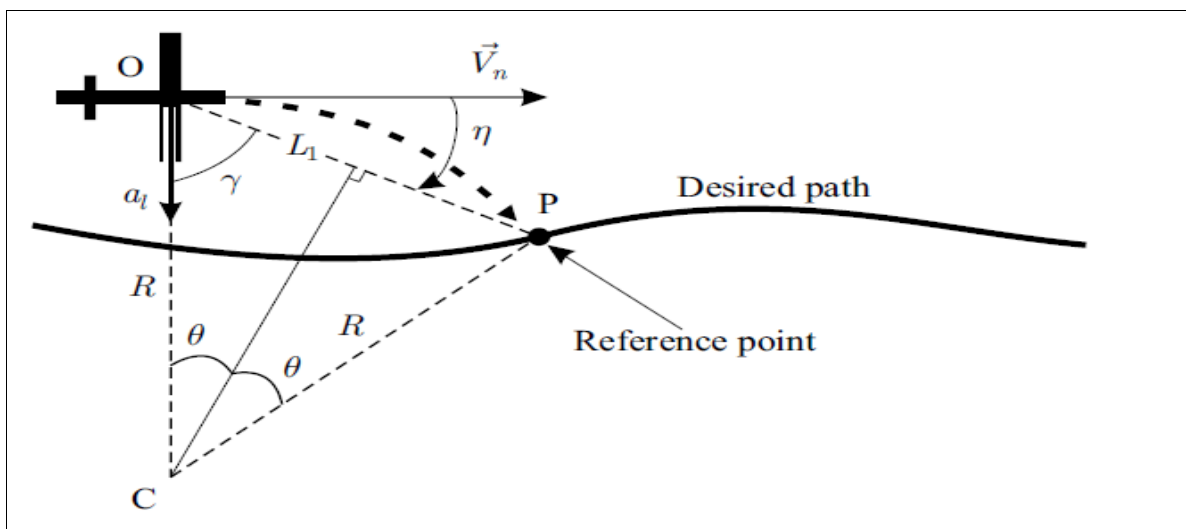
Kde: η - odchylka skutečného směru od žádaného směru letu (rad)

γ - úhel mezi směrem laterálního zrychlení a žádaným směrem letu (rad)

θ_a - úhel mezi vektorem laterálního zrychlení a kolmicí na L_1 v bodě C (rad)

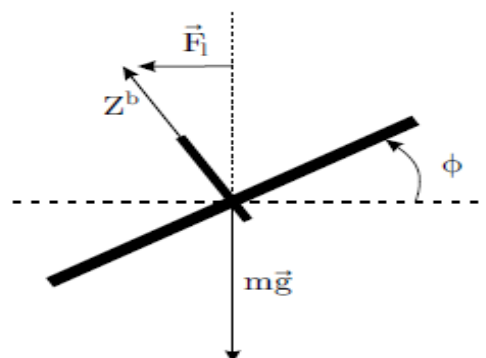
Po dosazení rovnice (17) do rovnice (14) dostaneme výsledný vztah pro a_l :

$$a_l = \frac{2 V_g^2}{L_1} \sin \eta \quad (18)$$



Ilustrace 9: zobrazení principu výpočtu laterálního zrychlení (zdroj [5])

Z kinematiky letadla, zobrazené na *Ilustraci 10*, můžeme z laterálního zrychlení a sil působících na letadlo určit žádaný úhel náklonu (roll angle) dle rovnic silové rovnováhy.



Ilustrace 10: letadlo při změně kurzu (zdroj [5])

Pro přímý let v letové hladině platí:

$$Z_b = m g \quad (19)$$

Kde: Z_b - vztlková síla (N)
 m - hmotnost letadla (kg)
 g - gravitační zrychlení Země (m/s^2)

Při nenulovém úhlu náklonu vzniká laterální síla:

$$F_l = Z_b \sin \theta \quad (20)$$

Kde: F_l - síla vzniklá změnou vztlkových sil na křídla v laterálním směru (N)
 θ - úhel náklonu (pitch angle; rad)

Již Newton pravil, že síla je rovna zrychlení násobené hmotou. Tedy laterální zrychlení (a_l) je rovno laterální síle (F_l) podělené hmotností letadla (m).

Po dosazení rovnic (19, 20) do rce (18) dostaneme:

$$a_l = \frac{F_l}{m} = \frac{Z_b \sin \theta}{m} = \frac{m g \sin \theta}{m} = g \sin \theta \quad (21)$$

Při předpokladu malé odchylky úhlu náklonu od nulové pozice lze rovnicí (21) linearizovat.

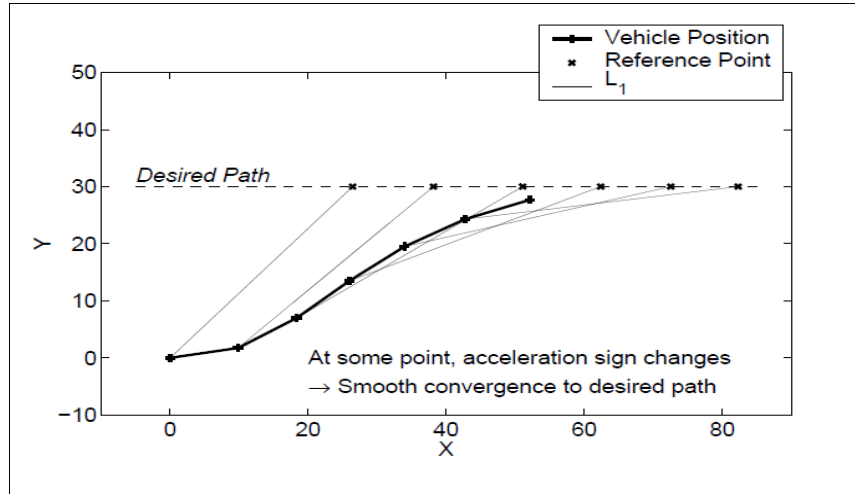
Linearizací funkce sin získáme rovnicí:

$$a_l = g \theta \quad (22)$$

Z této rovnice můžeme úpravou a dosazením rovnice pro laterální zrychlení (18) získat finální vztah pro výpočet žádaného úhlu náklonu:

$$\theta = \frac{a_l}{g} = \frac{2 V_g^2 \sin \eta}{L_1 g} \quad (23)$$

Využitím tohoto výpočtu trajektorie letadla bude konvergovat k žádané trajektorii mezi zadanými body.



Ilustrace 11: průběh konvergence k žádané trajektorii (zdroj [6])

2.4.2 Výpočet žádané výšky letu a úhlu sklonu

Výška letu h_c je počítána z polohy bodů, mezi kterými se letadlo nachází.

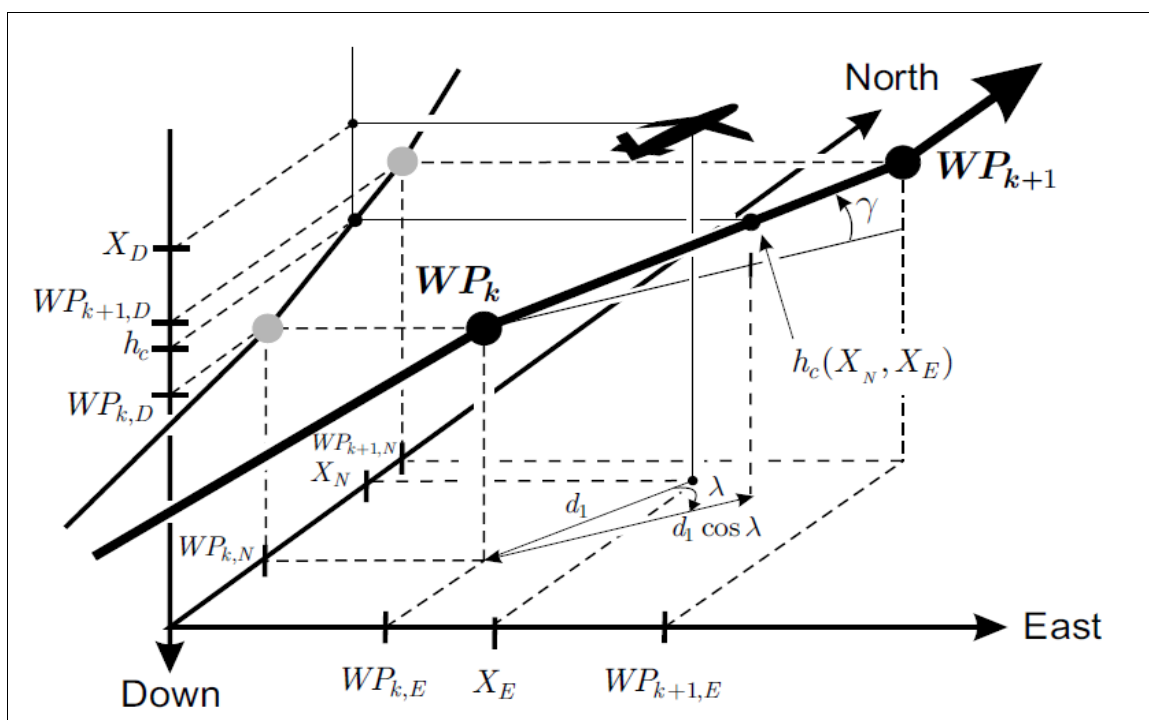
$$h_c = W P_{k,D} + \gamma d_1 \cos \lambda \quad (24)$$

Kde: $W P_{k,D}$ – výška posledního bodu, kterým letadlo proletělo (m)
 γ – úhel stoupání mezi body (rad)
 d_1 – vzdálenost letadla od posledního bodu promítnutá do roviny N-E (m)
 λ – odchylka polohy letadla od ideální trajektorie mezi body (rad)

Úhel γ se vypočte dle polohy dvou po sobě následujících bodů v seznamu:

$$\gamma = \text{atan} \left(\frac{W P_{k+1,D} - W P_{k,D}}{\sqrt{(W P_{k+1,N} - W P_{k,N})^2 + (W P_{k+1,E} - W P_{k,E})^2}} \right) \quad (25)$$

Kde: $W P_{k,D}$ $W P_{k+1,D}$ – nadmořská výška bodů (m)
 $W P_{k,N}$ $W P_{k+1,N}$ - severní šířka bodů (°)
 $W P_{k,E}$ $W P_{k+1,E}$ - východní délka bodů (°)



Ilustrace 12: zobrazení situace pro výpočet žádané výšky letu (zdroj [5])

2.4.3 Výpočet polohy referenčního bodu P

Rovnice (14 – 25) popisují laterální a longitudální pohyb letadla s výpočtem žádaného úhlu náklonu. Tato hodnota je závislá na vzdálenosti letadla od referenčního bodu na trajektorii a kolmé vzdálenosti letadla od trajektorie. Je potřeba definovat referenční bod (P).

Bod P je dle Ilustrace 13 reprezentován polohou v N-E rovině, závisle na vzdálenosti d_1 a úhlech χ a λ . Úhel λ lze spočítat na základě odchylky spojnice bodů od zeměpisného severního směru ψ_{seg} a odečtením χ od této odchylky.

$$\psi_{seg(k)} = \text{atan} \left(\frac{WP_{k+1,E} - WP_{k,E}}{WP_{k+1,N} - WP_{k,N}} \right); \in [-\pi, \pi] \quad (26)$$

Úhel odchylky letadla od ideální trajektorie:

$$\lambda = |\psi_{seg(k)}| - |\chi| \quad (27)$$

Kde: χ - úhel polohy letadla vzhledem k zeměpisnému severu od bodu WP_k (rad)

ψ_{seg} - úhel mezi spojnici se zeměpisným severem a bodem WP_{k+1} od WP_k (rad)

$$\chi = \text{atan} \left(\frac{X_E - WP_{k,E}}{X_N - WP_{k,N}} \right); \in [-\pi, \pi] \quad (28)$$

Vzdálenost d_1 určuje aktuální vzdálenost letadla od bodu WP_k v rovině N-E:

$$d_1 = \sqrt{(X_N - WP_{k,N})^2 + (X_E - WP_{k,E})^2} \quad (29)$$

Kde: X - GPS souřadnice aktuální polohy letadla rozložené do zeměpisných os (°)

Na základě těchto hodnot je poloha bodu P definována jako součet kolmého průměru polohy letadla na ideální trajektorii sečtený s částí ΔP :

$$[WP_k, P] = d_1 \cos \lambda + \sqrt{L_1^2 + d_1^2 \sin^2 \lambda} \quad (30)$$

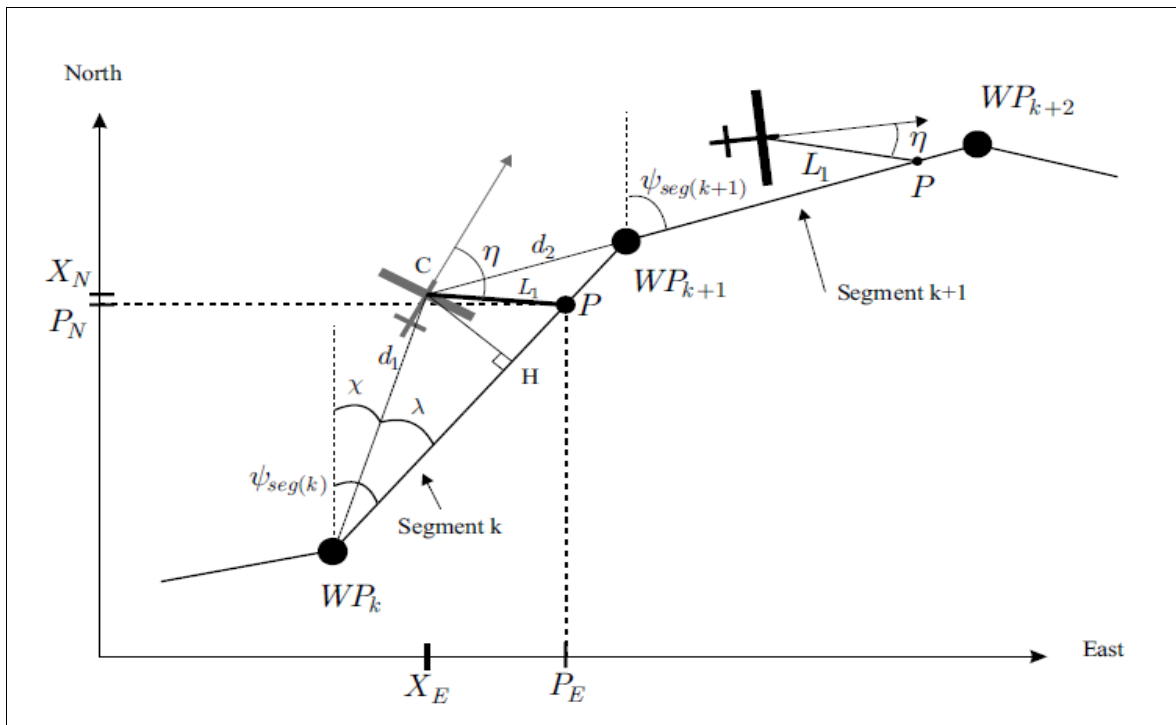
Kde: $[WP_k, H] = d_1 \cos \lambda$ - vzdálenost kolmého průmětu polohy letadla na trajektorii

$[H, P] = \Delta P = \sqrt{L_1^2 + d_1^2 \sin^2 \lambda}$ - vzdálenost od průmětu polohy letadla k bodu P

Poloha referenčního bodu P se rozloží do zeměpisných os (N-E roviny):

$$P_N = WP_{k,N} + [WP_k, P] \cos \psi_{seg(k)} \quad (31)$$

$$P_E = WP_{k,E} + [WP_k, P] \sin \psi_{seg(k)}$$



Ilustrace 13: zobrazení trajektorie mezi body (zdroj [5])

2.4.4 Princip detekce průletu bodem zájmu

Detekce je reprezentována dvěmi podmínkami.

První podmínka porovnává hodnoty vzdálenosti letadla od následujícího bodu (d_2) se vzdáleností od bodu P (L_1). V případě, že letadlo bude od bodu WP_k vzdáleno více než od referenčního bodu P tak stále letí k bodu WP_{k+1} .

Druhá podmínka porovnává vzdálenost kolmého průměru aktuální polohy letadla na ideální trajektorii s celkovou vzdáleností bodů. V případě, že vzdálenost $[WP_k, H]$ bude menší než $[WP_k, WP_{k+1}]$, tak letadlo stále letí k bodu WP_{k+1} .

$$\begin{aligned} d_2 &\geq L_1 \\ d_1 \cos \lambda &< |WP_k, WP_{k+1}| \end{aligned} \quad (32)$$

Při porušení alespoň jedné podmínky proběhne přepnutí do dalšího segmentu trajektorie, tedy inkrementace indexu waypointů k na $k+1$. Letadlo poletí na další bod v seznamu.

Zhodnocení metody

Tato metoda jednoduše popisuje řízení a navigaci letadla. Pro letadlo SkyDog je tato metoda vhodná. Neřeší situaci, když letadlo bude ve velké vzdálenosti od ideální trajektorie.

Řešení bezletové zóny

Ve zdroji [5] je popsáno i řešení zadání a oblétání bezletové zóny. V práci se tímto problémem nezabývám - tuto problematiku tedy nebudu rozepisovat.

3 Žádané funkce autopilota letadla SkyDog

Hlavní výpočet algoritmu je navržen na řešení klasického plánování letu bez omezení kinematikou a dynamikou letadla nebo uspořádáním bodů zájmu. V reálných situacích mohou nastat situace, které není klasický přístup schopný řešit. Proto při samotném zpracování reálného řídicího algoritmu je potřeba zaměřit pozornost i na tyto situace a přizpůsobit řídicí systém.

3.1 Přímý let na bod

klasické plánování trajektorie je řešeno jako neomezený přímý let na zvolený bod. Každý systém pro řízení letadla musí být schopný toto zpracovat. Přístup k tomuto problému je popsán v každé metodě řešení autopilota.

Pro tento problém je nejlépe popsán řešení v kapitole 2.4, kdy na základě výpočtu referenčního bodu a odchylky směru rychlosti od směru letadla k tomuto bodu počítá algoritmus žádanou hodnotu úhlu náklonu, který drží letadlo na zvolené trajektorii.

3.2 Speciální poloha bodů

Při zadávání waypointů může nastat několik typů nestandardních situací, které letadlo není schopno z konstrukčních či jiných důvodů splnit pouze klasickou metodou přímého letu. Je potřeba tyto situace odhalit a vyřešit samostatným algoritmem.

Jde o body v pozicích dle tabulky 2.

Poloha bodů	Problém
Body blízko sebe s velkým rozdílem výšky	Letadlo není schopno dostatečného stoupání (resp. klesání), aby stihlo splnit žádanou výšku letu.
Body přímo nad sebou	Klasický algoritmus není schopný tento problém řešit.
Poloha domácí stanice	Letadlo má v paměti polohu domácí stanice pro potřeby bezpečnosti.

Tabulka 2: Přehled speciálních poloh bodů

3.3 Automatické přepínání bodů zájmu

Při autonomním letu je žádoucí, aby na základě zpracování dat z použitých senzorů řídicí systém rozpoznal dosažení aktuálně žádaného waypointu a pokračoval v trajektorii na následující bod v zadaném letovém plánu.

Tento problém je vhodně popsán v kapitole 2.4.4. Zde jsou sestaveny podmínky (rovnice 32) pro detekci průletu bodem. Letadlo z důvodu přesnosti řízení a vlivu atmosferických jevů nemusí bod proletět přesně. Je nastavena hodnota tzv. „threshold“, která udává okolí kolem bodu vyhodnocené jako průlet bodem.

3.4 Online aktualizace bodů zájmu

Obsluha bezpilotního prostředku má vždy kontrolu nad průběhem letu. Tedy je nutné zajistit reakci letadla na žádanou změnu letového plánu. Letadlo na podnět z pozemní stanice reaguje na změnu žádaných bodů v trajektorii.

3.5 Reakce na aktuální polohu letadla

Letadlo je schopno letět v plně autonomním, semi-automatickém nebo manuálním modu. Proto řídicí algoritmus musí být navržen tak, aby při přepnutí z manuálního, resp. semi-automatického, modu počítal správné žádané hodnoty pro regulátory. Řídicí systém vždy musí být připravený na to, že se letadlo může nacházet v libovolné pozici vzhledem k bodu, ke kterému se má dostat. Z tohoto důvodu je potřeba změnit výpočet referenčního bodu, který je popsán v kapitole 2.4.3.

3.6 Chybové stavy

Při vybavení kteréhokoliv chybového stavu letadlo okamžitě zamíří na domácí stanici. Chybový stav lze resetovat pouze přepnutím do semi-automatického nebo manuálního modu letu. Po tomto resetu a přepnutí do autonomního modu se letadlo rozletí opět k bodu, který je první na seznamu. Chybové stavy jsou rozepsány v kapitole 4.3.

3.7 Nastavitelné parametry algoritmu

Některé parametry algoritmu lze měnit z pozemní stanice. Jsou to parametry, které pouze upravují přesnost dosažení žádaného bodu nebo mění některé řídicí hodnoty. Tyto parametry nezasahují do principu navrženého algoritmu.

4 Vybraná metoda řešení

Po analýze požadavků na autopilota jsem zvolil metodu popsanou v kapitole 2.4 – Jednoduché adaptivní řízení. Tato metoda řeší většinu z popsaných požadavků, ale je potřeba ji upravit a doplnit.

4.1 Řešení řízení závisle na poloze letadla

Při přepnutí do autonomního letu musí řídicí systém počítat žádané hodnoty pro regulátory z aktuální pozice letadla, která může být velmi odlišná od ideální trajektorie mezi předchozím a následujícím bodem.

Využijeme základní princip algoritmu, který byl popsán v kapitole 2.4, ale polohu referenčního bodu je potřeba počítat jiným způsobem.

4.1.1 Poloha referenčního bodu

Poloha referenčního bodu musí být vázána k aktuální poloze letadla z důvodu přepínání mezi manuálním a autonomním modem. V manuálním modu může letadlo zaletět na místo, které je velmi vzdálené od žádaného bodu v seznamu.

Vzdálenost mezi letadlem a referenčním bodem je konstantní. Toto je jeden z parametrů, které lze měnit z domácí stanice. Poloha bodu je počítána z aktuální pozice letadla vzhledem k aktuálnímu bodu zájmu. Referenční bod bude stále mezi letadlem a bodem zájmu. Poloha ref. bodu je počítána z jednotkového vektoru polohy bodu zájmu od letadla.

Jednotkový vektor polohy bodu zájmu od letadla:

$$d_{2one, N} = \left(\frac{W P_{k+1, N} - U_N}{\sqrt{(W P_{k+1, N} - U_N)^2 + (W P_{k+1, E} - U_E)^2}} \right)$$
$$d_{2one, E} = \left(\frac{W P_{k+1, E} - U_E}{\sqrt{(W P_{k+1, N} - U_N)^2 + (W P_{k+1, E} - U_E)^2}} \right)$$
(33)

Kde: U – poloha letadla vzhledem k domácí stanici, rozloženo do směrů zeměpisných os (m)

Poloha referenčního bodu při klasickém plánování trajektorie:

$$P_N = U_N + L d_{2one, N}$$
$$P_E = U_E + L d_{2one, E}$$
$$P_D = W P_{k+1, D}$$
(34)

Kde: L – volená vzdálenost referenčního bodu od letadla (m)

Dále postupujeme dle kapitoly 2.4. Bod P nebude ležet na ideální trajektorii z důvodů popsaných výše.

4.1.2 Pohyb kolem domácí stanice

Letadlo se kolem domácí stanice pohybuje podle stejného plánování trajektorie jako je tomu při nadměrné změně výšky (popsaný v kapitole 4.2.1). Jediný rozdíl nastává při pokračování letu na další bod v seznamu. V případě, že letadlo stále poletí v autonomním modu, bude kroužit kolem domácí stanice. Letadlo změní letový plán pouze v případě resetování (kap. 3.6).

4.2 Řešení speciálních poloh bodů

4.2.1 Nadměrná změna výšky

Algoritmus je schopný detekovat, zda žádaný úhel stoupání (úhel sklonu, pitch angle) nepřesáhl dovolenou hodnotu danou dynamikou letadla. V případě, že algoritmus detekuje tento stav, vybaví se přepnutím do nestandardního plánování.

Letadlo se pohybuje s omezeným úhlem stoupání dokud se nedostane do blízkosti waypointu. Po dosažení určité vzdálenosti od waypointu začne spirálovitě stoupat kolem navrženého bodu C se zvoleným poloměrem.

Bod C je volen kolmo na segment, ve kterém se letadlo pohybuje ve vzdálenosti voleného poloměru R (m). Kolmici na segment vypočteme z jednotkového vektoru směru segmentu d_{ONE} .

$$\begin{aligned} d_{one,N} &= \left(\frac{W P_{k+1,N} - W P_{k,N}}{\sqrt{(W P_{k+1,N} - W P_{k,N})^2 + (W P_{k+1,E} - W P_{k,E})^2}} \right) \\ d_{one,E} &= \left(\frac{W P_{k+1,E} - W P_{k,E}}{\sqrt{(W P_{k+1,N} - W P_{k,N})^2 + (W P_{k+1,E} - W P_{k,E})^2}} \right) \end{aligned} \quad (35)$$

Pro výpočet kolmice stačí prvky jednotkového vektoru zaměnit a k jednomu z nich přidat opačné znaménko. Rovnice pro střed kruhu C po dosazení rozložené do jednotlivých zeměpisných os:

$$\begin{aligned} C_N &= W P_{k+1,N} - R d_{one,E} \\ C_E &= W P_{k+1,E} + R d_{one,N} \\ C_D &= W P_{k+1,D} \end{aligned} \quad (36)$$

Změna plánování trajektorie se projeví také ve výpočtu polohy referenčního bodu. Bod je určen polohou bodu C a aktuální polohou letadla vzhledem k tomuto bodu. Aktuální poloha letadla je v tomto algoritmu určena odchylkou od zeměpisného severu. Bod P se nachází na trajektorii před letadlem, proto je potřeba spočítat úhel mezi U a P vzhledem k C.

Úhlová odchylka γ_{UP} (rad) mezi U, P závisle na velikosti vektoru L a poloměru traj. R:

$$\gamma_{UP} = 2 \sin^{-1} \left(\frac{L}{2R} \right) \quad (37)$$

Odchylka polohy letadla ϕ_{CU} (rad) vzhledem k bodu C od zeměpisného severu:

$$\phi_{CU} = \text{atan}\left(\frac{CU_E}{CU_N}\right) \quad (38)$$

Kde: CU – vzdálenost letadla v daném směru zeměpisné osy od bodu C (m)

Poloha bodu P s využitím polohy letadla vzhledem k C a odchylky γ_{UP} :

$$\begin{aligned} P_E &= C_E + R \sin(\phi_{CU} + \gamma_{UP}) \\ P_N &= C_N + R \cos(\phi_{CU} + \gamma_{UP}) \\ P_D &= C_D \end{aligned} \quad (39)$$

Dále algoritmus pracuje dle kap. 2.4.

4.2.2 Poloha domácí stanice

Před vzletnutím je potřeba zadat polohu domácí stanice určující plán letu při vybavení jakéhokoli chybového stavu nebo po proletění celého seznamu bodů. Tato pozice je použita jako offset hodnota od nulového bodu GPS, který se nachází na průsečíku rovníku a nultého poledníku.

Domácí stanice je využívána při řešení chybových stavů letadla, při průletu celého seznamu bodů, který má zadaný a také při řešení situace, kdy žádný bod v seznamu nemá zadaný.

4.2.3 Dva body přesně nad sebou

V případě této situace je problém s výpočtem bodu C, kolem kterého má letadlo spirálovitě stoupat či klesat. Jednotkový vektor vzdálenosti bodů (rovnice 35) bude v tomto případě nabývat nulových hodnot. Bod C tedy bude shodný s bodem WP_{k+1} . V případě, že poloměr kružnice, po které má letadlo letět, bude větší než „threshold“, potom by nikdy nebyla splněna podmínka přepnutí bodu zájmu. Letadlo by tedy uvízlo na trajektorii kolem tohoto bodu.

Jednoduchou detekcí této situace a ošetřením hodnot jednotkového vektoru můžeme této situaci předejít. Vypočteme si hodnoty jednotkového vektoru. Při zjištění nulových hodnot v obou členech vektoru tyto hodnoty přepíšeme na hodnoty dle předchozího segmentu trajektorie nebo-li pro $k = k-1$. Tím zajistíme správný výpočet bodu C, který se bude nacházet ve vzdálenosti R od WP_{k+1} . Tím přejdeme principiálně do plánování dle kapitoly 4.2.1 – Nadměrná změna výšky.

4.3 Řešení chybových stavů

Při jakémkoliv chybovém hlášení je letadlo, z důvodu bezpečnosti, naváděno na domácí stanici. Zde si může převzít kontrolu obsluha nebo letadlo začne kroužit v letové hladině dokud nedostane nový povel.

4.3.1 Možnosti vybavení chybového stavu

Chybový stav může být vybaven několika způsoby:

přerušení signálu z vysílačky	Při ztrátě signálu z vysílačky je nebezpečí nekontrolovaného letu. Proto je tato situace ošetřena letem na domácí stanici, což je v našem případě stanoviště obsluhy.
indikace vybití baterie	Při zaznamenání poklesu napětí baterie lze očekávat v nejbližší době zřícení letadla. Když navedeme letadlo na domácí stanici může obsluha s letadlem bezpečně přistát.
jiný stav	Při každém jiném hlášení chyby letadlo změní letový plán a letí na domácí stanici.

4.4 Proměnné parametry

4.4.1 Vzdálenost referenčního bodu před letadlem

Vzdáleností tohoto bodu od letadla (vzdálenost L_1) ovlivňujeme dynamiku řízení letadla v laterálním směru. Pokud vzdálenost U a P zvolíme malou tak letadlo bude prudce reagovat na změny v okolí waypointů. Při zvolení příliš velké vzdálenosti letadlo bude reagovat velmi pomalu.

4.4.2 Poloměr kruhové trajektorie

Kruhová trajektorie je využita při požadavku na nadměrné stoupání či klesání nebo při kroužení kolem domácí stanice. Volený poloměr (parametr R) určuje žádanou kružnici, po které se letadlo má pohybovat. Letadlo je schopné letět po určitém minimálním poloměru. Pokud hodnotu R zvolíme příliš malou potom je jisté, že letadlo nebude schopné tuto trajektorii dodržet.

4.4.3 Dovolená odchylka od bodu

Tímto parametrem (tzv. „threshold“) dovoluujeme uznat průlet se stanovenou odchylkou od skutečné pozice. Toto je z důvodu reálné aplikace ve velmi proměnném prostředí atmosféry. Zde může nastat situace, že letadlo poletí přímo na bod a působením náhlého poryvu větru odnese letadlo o několik metrů. Proto si nastavíme threshold tak, abychom i při této situaci dovolili pokračovat na další bod v seznamu bodů zájmu.

4.5 Popis navrženého algoritmu

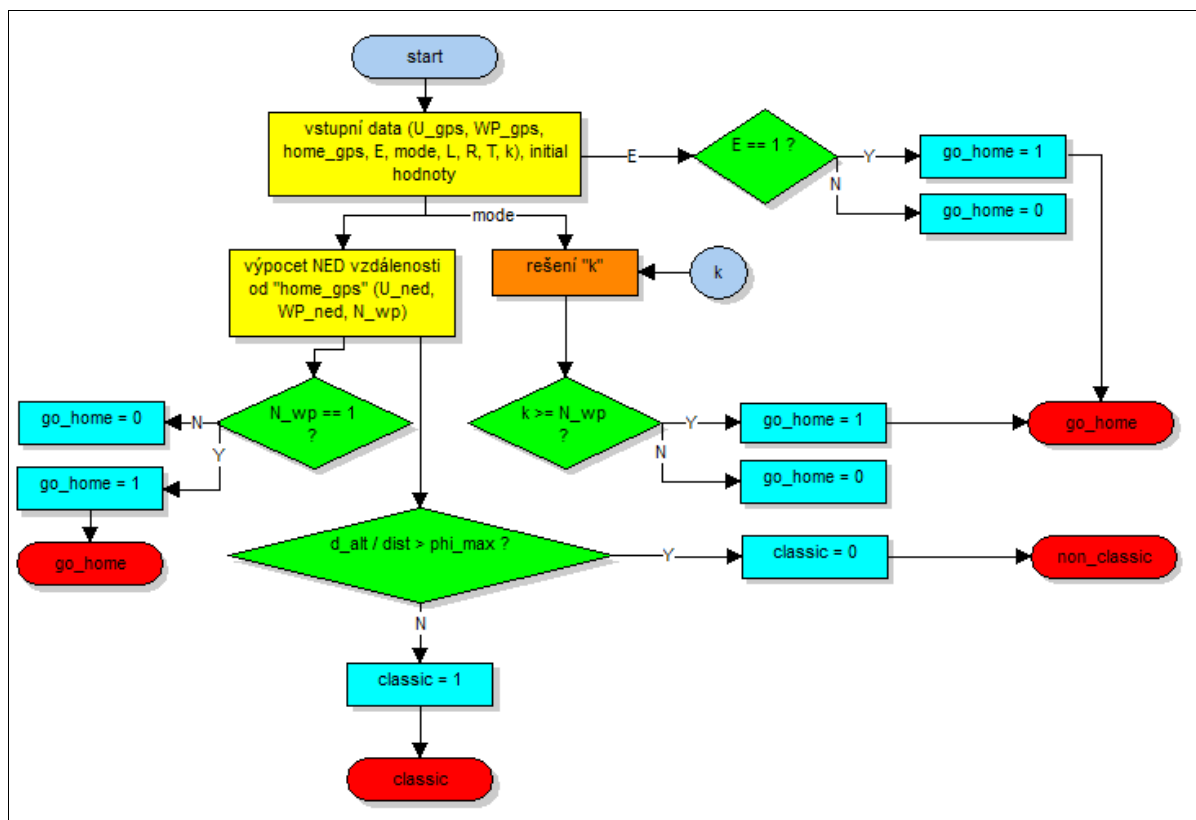
Hlavní výpočet algoritmu byl popsán v kap. 2.4. s drobnou úpravou výpočtu referenčního bodu P popsanou kapitolou 4.1.1 pro klasickou rovnou trajektorii a kapitolou 4.2.1 pro případ kruhové trajektorie. Tento výpočet řeší laterální pohyb letadla.

Při řešení longitudálního pohybu (výška, rychlost) jsem zvolil metodu skokových změn žádané výšky. Při přepnutí aktuálního bodu se do žádané hodnoty výšky zapíše hodnota výšky tohoto bodu. Toto se v simulacích (kap. 6) projevilo jako vyhovující. Rychlost je řešena jako konstantní hodnota.

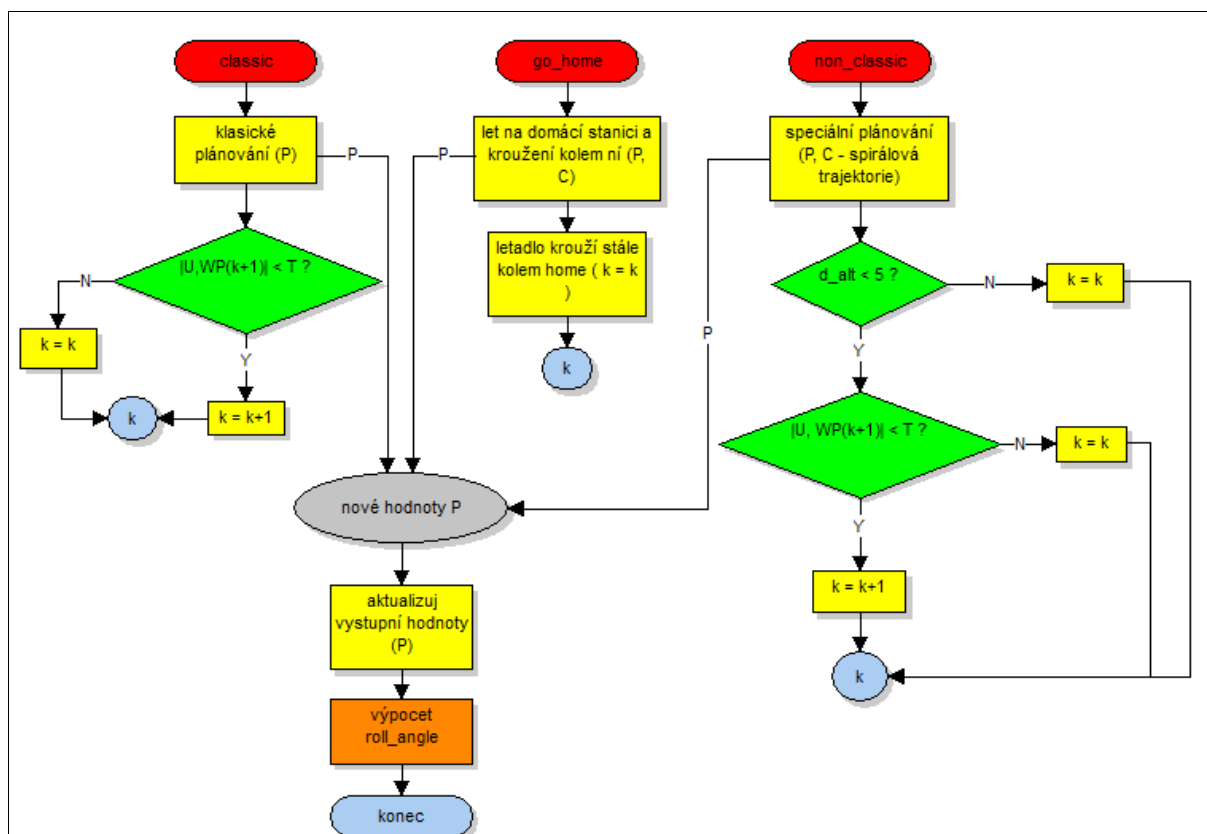
Další částí je řešení kroužení kolem domácí stanice a využití domácí stanice zapsané v paměti autopilota. Tato poloha je využita při vybavení chyby nebo při průletu celého seznamu bodů v paměti. Seznam bodů lze aktualizovat za letu z pozemní stanice.

Grafická interpretace algoritmu

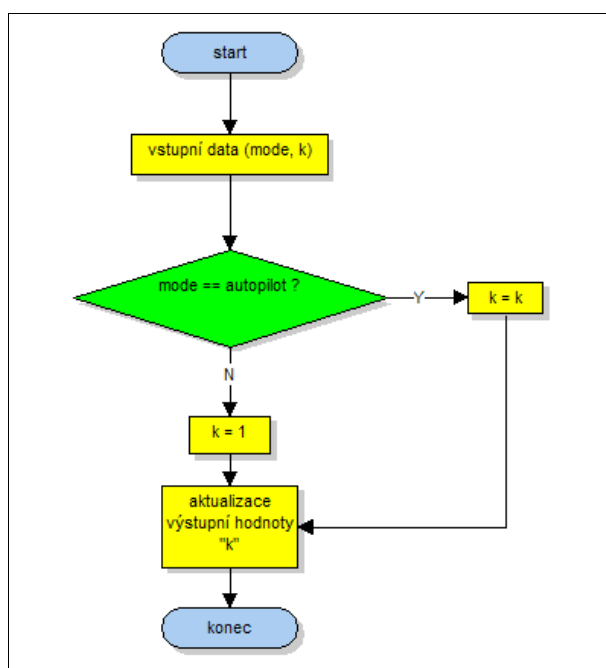
Následující schemata zobrazují graficky sestavený algoritmus řízení letadla. Hlavní část (*Ilustrace 14 a 15*) je výpočet referenčního bodu P a hodnot indexu bodu v seznamu „k“ a úhlu náklonu „roll_angle“. Index „k“ a „roll_angle“ je podrobněji na *Ilustrace 16 a 17*. Řešení indexu bodu „k“ obsahuje dříve zmíněný reset při přepnutí do manuálního modu a zpět do autonomního. Úhel náklonu je počítán z odchylky aktuální rychlosti vzhledem k Zemi a žádané polohy letadla dané referenčním bodem P. Tato odchylka je označena „eta“.



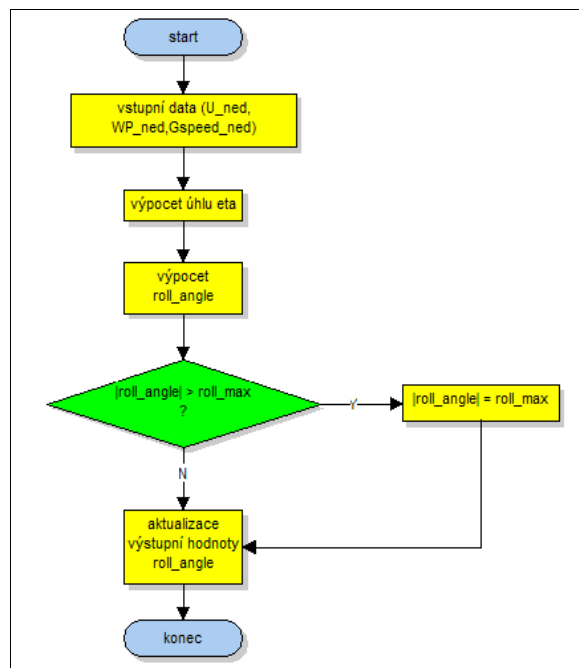
Ilustrace 14: řešení výběru typu plánování



Ilustrace 15: dle typu plánování výpočet P a podmínky pro přepnutí na další bod v seznamu



Ilustrace 17: přepínání z manual do autonomního modu - řešení indexu "k"



Ilustrace 16: výpočet úhlu náklonu (roll angle)

5 Realizace autopilota v Matlab/Simulink

Návrh autopilota probíhal v prostředí Matlab/Simulink. Toto prostředí bylo vybráno na základě předchozích zkušeností s modelováním a prací s ním. Lze tu navrhnout řídicí smyčky, rozhodovací algoritmy a samotný model letadla SkyDog.

5.1 Vstupní data

Nejdůležitější vstupní data jsou aktuální pozice letadla v Gps souřadnicích, jeho rychlost vzhledem k Zemi, seznam waypointů a poloha domácí stanice. Další vstupy jsou vybavení chyby a letový mod.

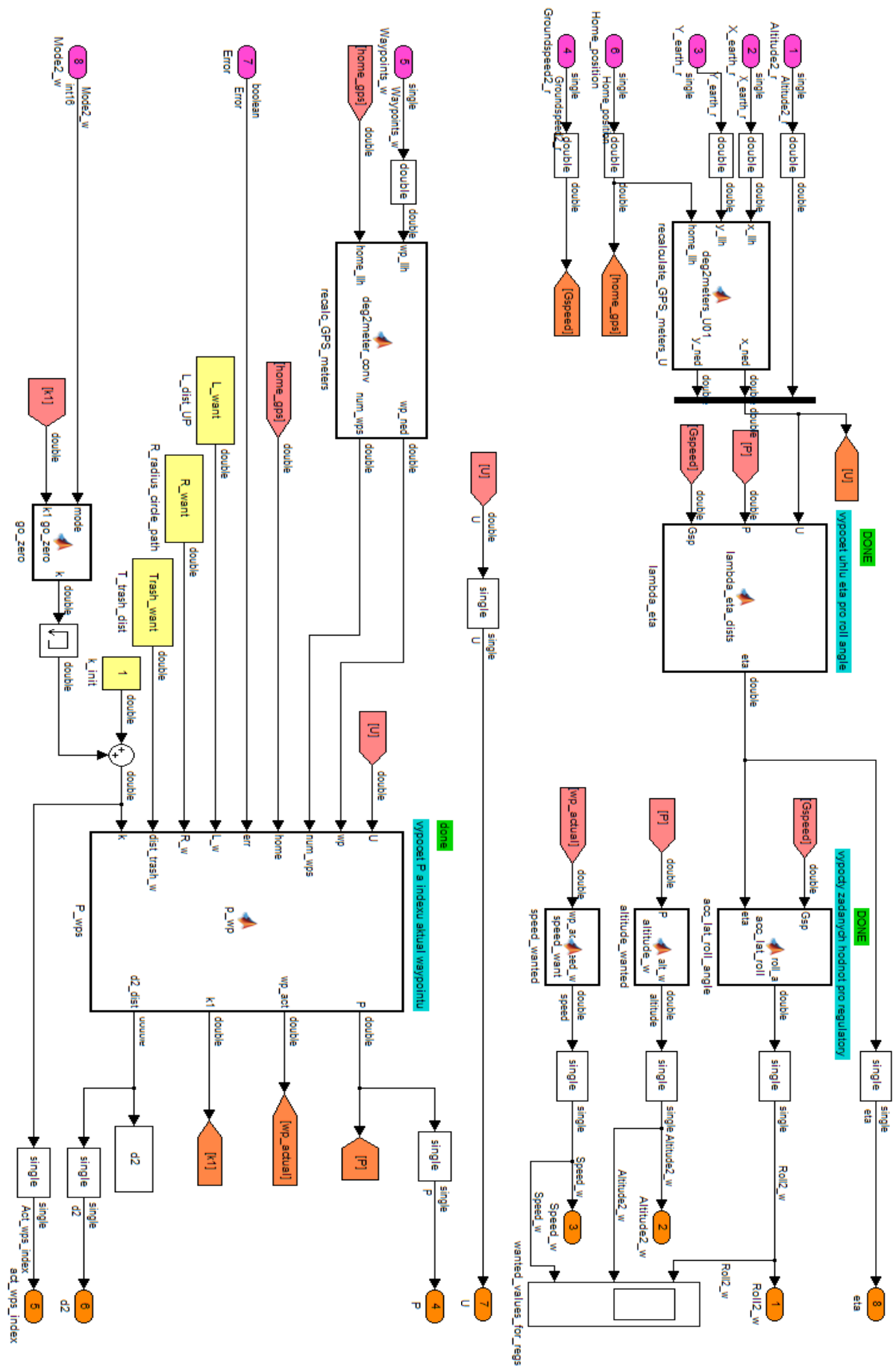
Vstupy jsou z GPS souřadnic přepočteny na vzdálenost od domácí stanice. Přepočet je uveden v kapitole 5.1.1.

Ze vstupů polohy a rychlosti letadla a bodů v seznamu algoritmus počítá polohu referenčního bodu P v hlavní výpočetní smyčce dle bloku „P_wps“. Tento blok dále vyhodnocuje vstup chyby „Error“. Do smyčky vstupují i parametry geometrie a odchylky „L_want“, „R_want“ a „Trash_want“, které definují polohu bodu P a podmínku inkrementace žádaného bodu. Hodnota „k_init“ udává počáteční index prvního bodu v seznamu. Dále inkrementaci řeší hlavní výpočet s ohledem na blok „go_zero“, který řeší resetování při přepnutí.

Popis vstupů

č.	Název	Popis
1	Altitude_r	Aktuální výška letadla (m.n.m.)
2	X_Earth_r	Aktuální poloha letadla GPS východní délka (°)
3	Y_Earth_r	Aktuální poloha letadla GPS severní šířka (°)
4	Groundspeed_r	Aktuální rychlost letadla v NED souřadném systému Země (m/s, m/s, m/s)
5	Waypoints_w	Seznam bodů, kterými má letadlo proletět v GPS souřadnicích (m.n.m., °, °)
6	Home_position	Poloha domácí stanice v GPS (m, °, °)
7	Error	Vstupní hodnota hlášení chyby (Bool)
8	Mode_W	Letový mod letadla (využitý pro reset indexu k, přepínání mezi RC, stabilizací + RC a autopilotem)

Tabulka 3: Vstupní hodnoty do řídicí vrstvy



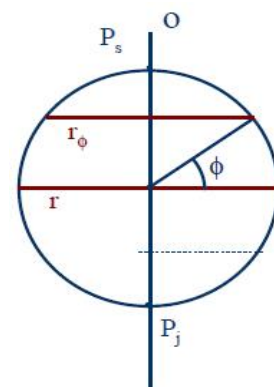
Ilustrace 18: Celkový algoritmus v prostředí Matlab/Simulink

5.1.1 Přepočet GPS na vzdálenosti od domácí stanice

Navržený algoritmus pracuje se základními jednotkami SI – metry, sekundy. Proto je nutné si GPS souřadnice ve stupních zemské šířky (Y_{earth}) a délky (X_{earth}) přepočítat na vzdálenosti od domácí stanice.

Poloha domácí stanice je známá (zadáno obsluhou). Přepočet ze stupňů na metry je realizován z hodnot geometrie Země (poloměr Země na rovníku: 6378,16 km). Zemi pro malé letadlo můžeme uvažovat jako kouli. Poledníky se směrem k pólu Země přibližují a koeficient přepočtu se mění. Letadlo se pohybuje téměř ve stejné zemské šířce, která nám určuje vzájemnou vzdálenost jednotlivých poledníků.

Při výpočtu je nutná korekce rovníkového poloměru do poloměru v užívané světové šířce.



Ilustrace 19: výpočet koeficientu přepočtu (zdroj [7])

$$deg2m = \frac{2 \pi R \cos\left(Y_{earth} \frac{\pi}{180}\right)}{360} \quad (40)$$

Kde: R - poloměr Země v oblasti rovníku (m)

$deg2m$ - koeficient přepočtu ze stupňů na metry (-)

Výsledná hodnota pozice vzhledem k domácí stanici je tedy:

$$\begin{aligned} X_{NED} &= (X_{earth} - home_x) deg2m \\ Y_{NED} &= (Y_{earth} - home_y) deg2m \end{aligned} \quad (41)$$

Kde: $home$ - poloha domácí stanice v GPS souřadnicích v jednotlivých osách (°)

X_{NED} , Y_{NED} - vzdálenost od domácí stanice v NED souřadném systému s počátkem v „home“ (m)

Každou hodnotu v GPS souřadnicích vstupující do algoritmu přepočteme tímto způsobem na vzdálenost v metrech od domácí stanice v jednotlivých zeměpisných osách. Tím získáme vhodné vstupní hodnoty pro hlavní smyčku „P_wps“.

5.2 Součásti řídicí vrstvy autopilota

Autopilot je rozdělen do několika funkcí. Funkce jsou reprezentovány bloky, ve kterých je výpočet proveden pomocí „matlab function“. Tyto skripty se chovají jako funkce. Tyto funkce lze konvertovat do C kódu, který využijeme v samotné desce autopilota. Prostředí Simulink slouží k návrhu a optimalizaci výpočetních procesů.

Jednotlivé funkce

Název	Vstupy	Výstupy
Deg2metersU01	x_llh, y_llh (souřadnice GPS); home_llh (domácí stanice GPS)	x_ned, y_ned (souřadnice v metrech od domácí stanice)
deg2meter_conv	wp_llh (seznam bodů GPS); home_llh (domácí stanice GPS)	wp_ned (seznam bodů v NED); num_wps (počet bodů v seznamu)
P_wp	U (aktuální poloha letadla v NED); wp (seznam bodů); num_wp (počet bodů); home (poloha domácí stanice GPS); err (hlášení chyby); L_w (žádaná vzdálenost P od U); R_w (poloměr při kruhové traj.); dist_trash_w (nastavení threshold); k (index aktuál. bodu před letadlem)	P (poloha referenčního bodu); wp_act (poslední prolétnutý a následující bod v seznamu); k1 (index bodu v seznamu); d2_dist (vzdálenost od bodu, ke kterému letadlo letí)
lambda_eta_dist	U (aktuální poloha letadla v NED); P (poloha referenčního bodu); Gspeed (rychlost letadla vzhledem k Zemi GPS)	Eta (úhel mezi vektorem aktuální rychlosti a vektorem UP)
go_zero	Mode (mod řízení); k1 (index aktuál. bodu před letadlem)	K (index aktuál. bodu před letadlem)
acc_lat_roll	Eta (úhel mezi vektorem aktuální rychlosti a vektorem UP); Gspeed (rychlost letadla vzhledem k Zemi GPS)	roll_w (výstupní hodnota žádaného úhlu náklonu pro regulátor)
altitude_w	P (poloha referenčního bodu)	alt_w (výstupní hodnota žádané výšky letu pro regulátor)
speed_want	wp_act (poslední prolétnutý a následující bod v seznamu – tento vektor obsahuje žádanou rychlost v daných bodech)	speed_w (výstupní hodnota žádané rychlosti pro regulátor)

Tabulka 4: Popis funkčních bloků řídicí vrstvy

5.3 Výstupní hodnoty

Funkční výstupní hodnoty z řízení jsou vstupní žádané hodnoty pro regulátory výšky, směru a rychlosti (č. 1-3 v tabulce „Popis výstupů“). Hodnoty U a P jsou pro posouzení správnosti a přesnosti navrženého řešení. Ostatní hodnoty zaznamenávají rychlost a přesnost regulace.

Popis výstupů

č.	Název	Popis
1	Roll2_w	Žádaný úhel náklonu (rad)
2	Altitude2_w	Žádaná výška letu (m.n.m.)
3	Speed_w	Žádaná rychlost letu (m/s)
4	P	Poloha referenčního bodu pro grafické zobrazení (m)
5	act_wps_index	Index aktuálního bodu v seznamu kam letadlo letí (-)
6	d2	Vzdálenost od bodu, ke kterému letadlo letí (m)
7	U	Aktuální poloha letadla NED od home (m)
8	eta	Úhel mezi vektorem rychlosti letadla a vektorem UP (rad)

Tabulka 5: Výstupní hodnoty řídicí vrstvy

6 Ověření funkčnosti řešení v simulačním prostředí

Funkčnost byla ověřena definovanými letovými zkouškami, které postupně od jednoduchých trajektorií vyzkouší všechny možné stavy, které mohou během letu nastat. Zobrazením a popisem vhodných grafů zjistíme, zda se autopilot chová dle požadavků na něj kladených popsaných v kapitole 3.

6.1 Použitý model letadla k ověření

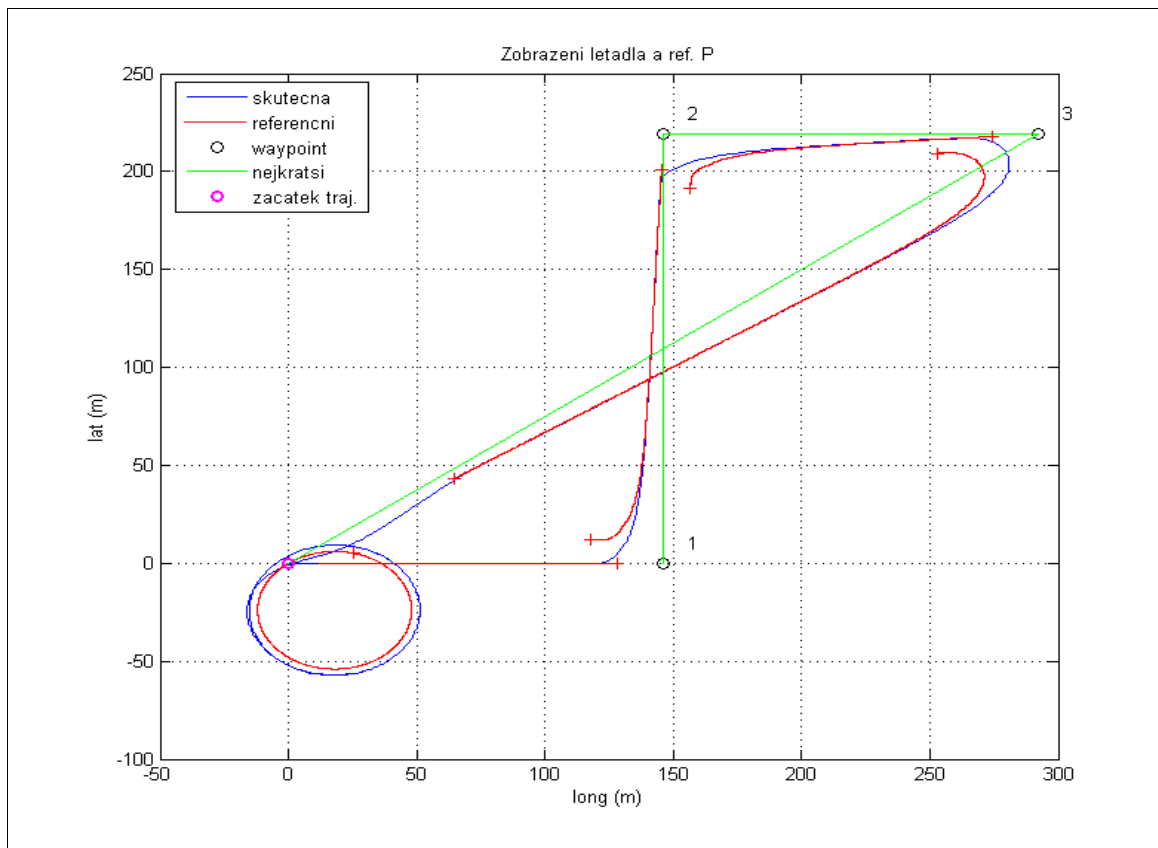
Základní dynamický popis letadla je uveden v kap. 2.2, rovnice (6-9). Model letadla není součástí této práce, ale kolegy Bc. Libora Příleského, který na projektu SkyDog spolupracoval. Letadlo je matematicky popsáno dle aerodynamických koeficientů konstrukce, elektromotoru a sil vyvozených touto konfigurací. Geometrické zobrazení letadla v příloze A.

6.2 Letové manévry

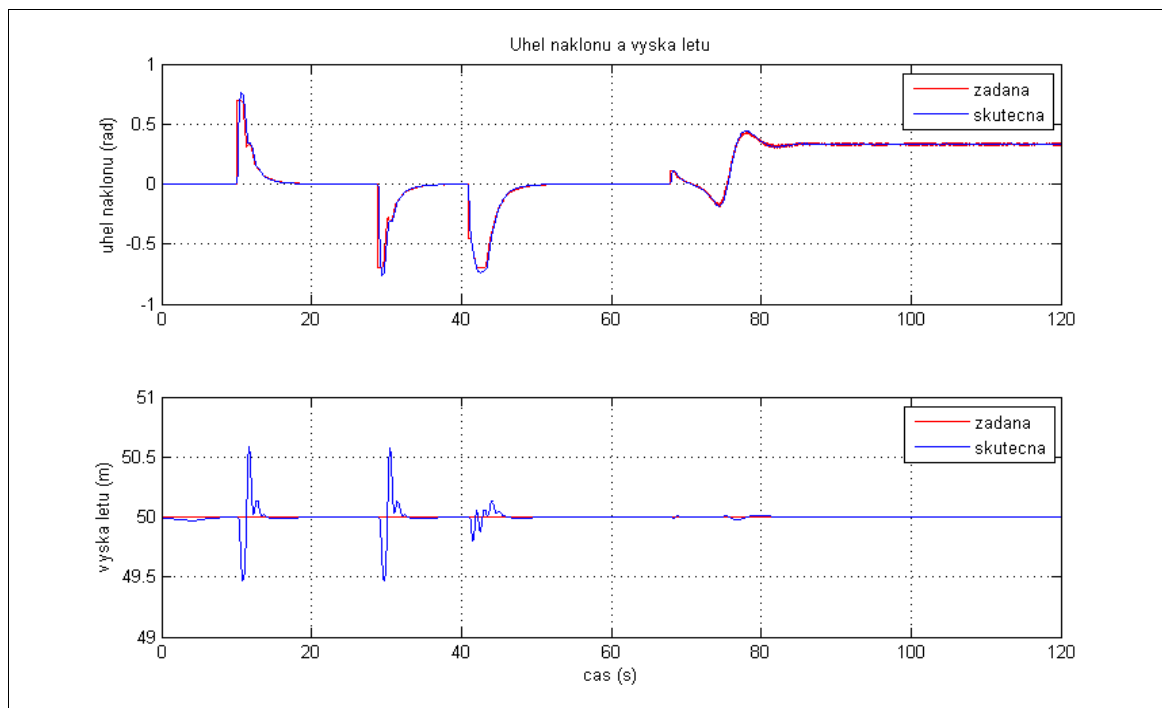
6.2.1 Přímá trajektorie s pravoúhlou zatáčkou

Jde o pohyb po přímce. Letadlo se stabilizuje na kurzu přímo k waypointu.

Grafy



Ilustrace 20: Simulace pravoúhlé zatáčky



Ilustrace 21: Žádané a skutečné hodnoty úhlu náklonu a výšky letu

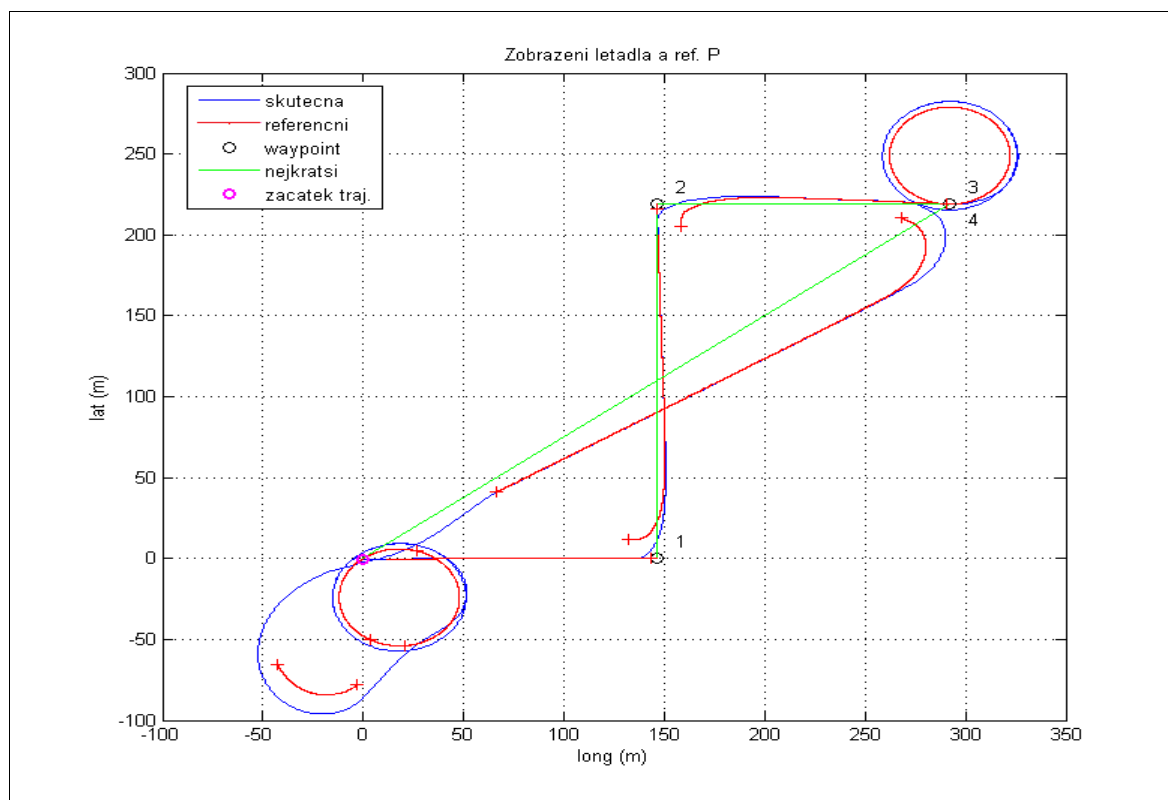
Zhodnocení

Algoritmus je schopný držet letadlo na přímé trajektorii a ve stabilní výšce. V *Ilustrace 21* letová výška není konstantní. Tyto přechody jsou způsobeny náhlým náklonem letadla a krátkodobou ztrátou vztlaku na křídlech. Po přechodu do stabilního letu regulátor opět dosáhne žádané výšky letu. Tento algoritmus je stabilní. Při přepnutí plánování na další bod v seznamu nastane skoková změna polohy referenčního bodu P (viz. *Ilustrace 20*). Toto je způsobeno principem výpočtu bodu P, který se snaží letadlo vést k dalšímu bodu. Lze vidět i ukončení letu a návrat k domácí stanici.

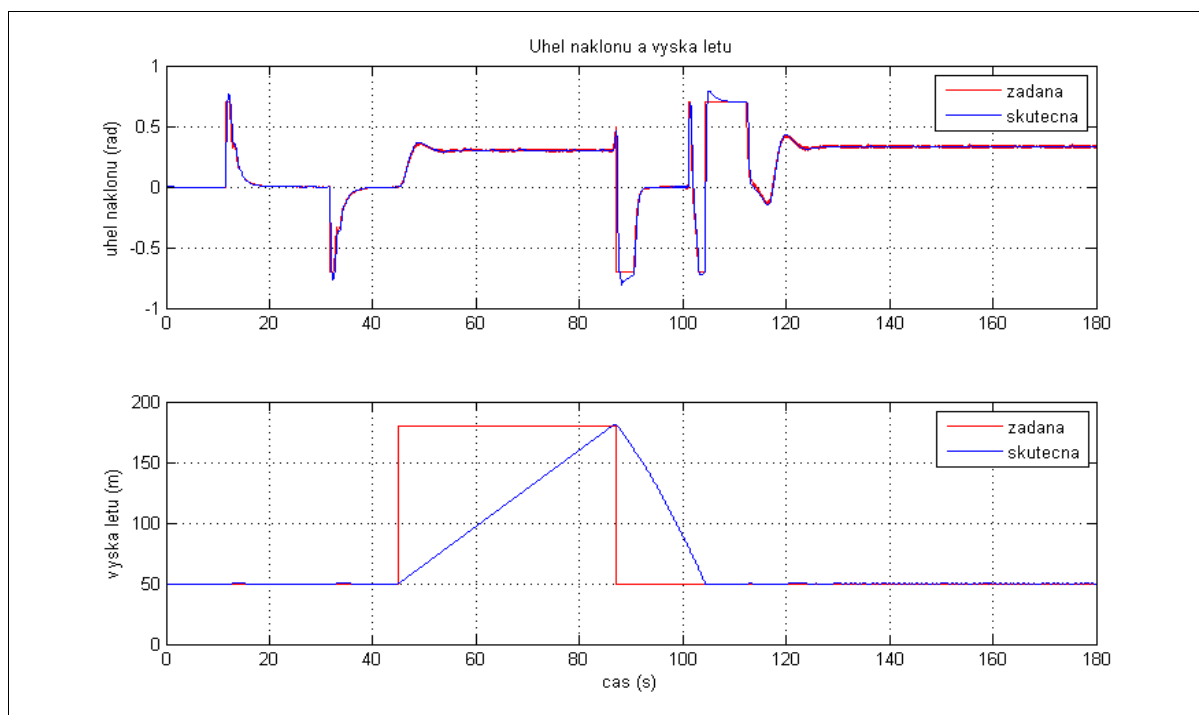
6.2.2 Speciální polohy bodů

V trajektorii se nachází sekvence bodů, které nemají standartní polohy vhodné pro letadlo. Algoritmus musí vyřešit krouživé stoupání či klesání. V místě [290, 225] jsou dva body nad sebou, které musí letadlo proletět stoupavou trajektorií (viz. *Ilustrace 23* – průběh žádané letové výšky).

Grafy



Ilustrace 22: Krouživé stoupání a klesání



Ilustrace 23: Žádané hodnoty z řízení a skutečné hodnoty letadla

Zhodnocení

Při pohledu na *Ilustraci 22* vidíme, že v určité vzdálenosti letadla od waypointu na pozici [290, 225] nastává změna plánování a to na kruhovou stoupavou trajektorii.

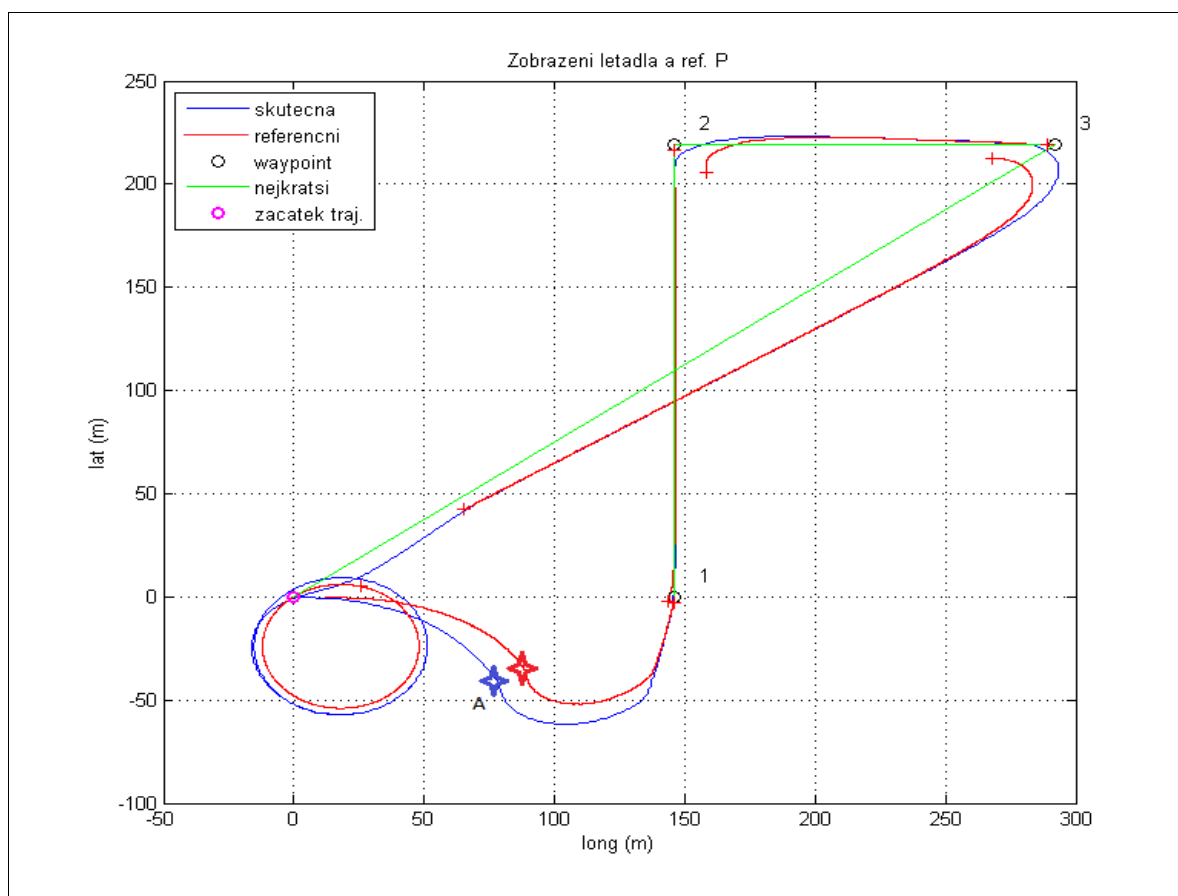
Body zájmu jsou z pozemní stanice zadávány na interaktivní mapě. Pokud nebudeme polohy manuálně upravovat tak pravděpodobně nedosáhneme zadání dvou bodů přesně nad sebou. Může nastat situace, kdy dva body budou blízko sebe, ale s rozdílnou výškou. Tuto kombinaci řeší algoritmus stejným způsobem, opět kruhovým stoupáním či klesáním.

V blízkosti domácí stanice [0, 0] vidíme skokovou změnu plánování letu. To je způsobeno přesáhnutím vzdálenosti letadla od bodu, kdy je plánováno krouživé stoupání či klesání. Algoritmus přepne do plánování bodu P mezi letadlem a waypointem. Tímto se dostane letadlo opět do blízkosti waypointu a pokračuje krouživá trajektorie.

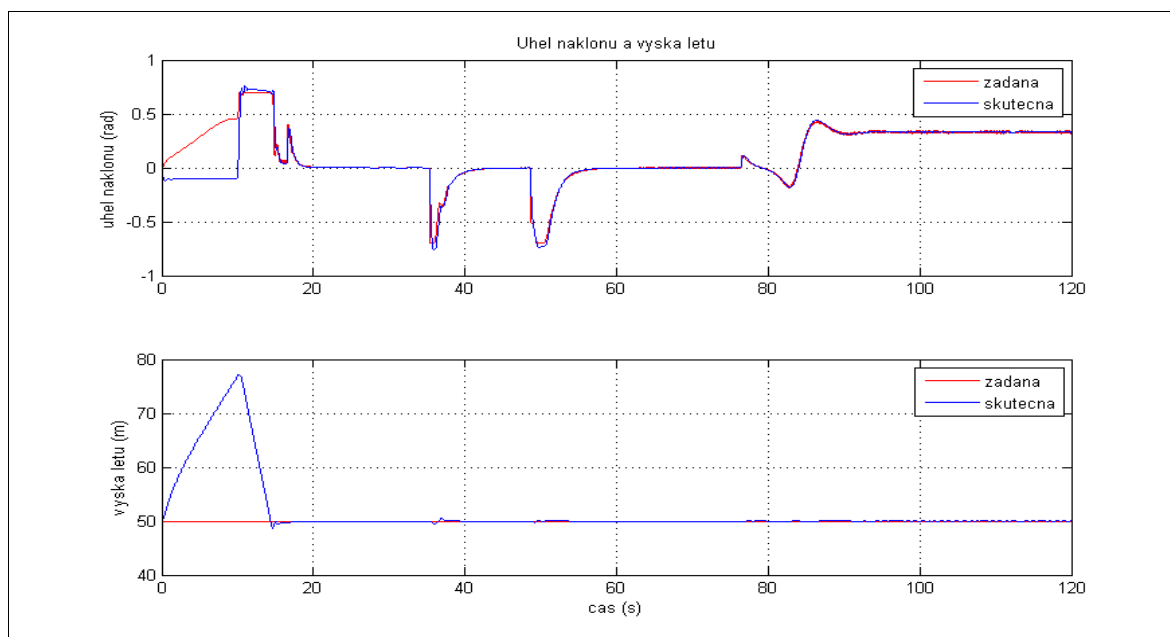
6.2.3 Reakce na přepnutí z manuálního modu

Letadlo letí dle pokynů z RC vysílačky a v čase 10s je přepnuto do automatického režimu. Algoritmus navádí letadlo k prvnímu bodu v seznamu. V manuálním modu je nastaven stabilní úhel náklonu -0.1 rad. Letadlo se začne odklánět od směru k prvnímu waypointu.

Grafy



Ilustrace 24: Zobrazení trajektorie při přepnutí z manuálního do automatického modu



Ilustrace 25: Žádané hodnoty a reálný stav letadla (manualní mod v čase 0 až 10s)

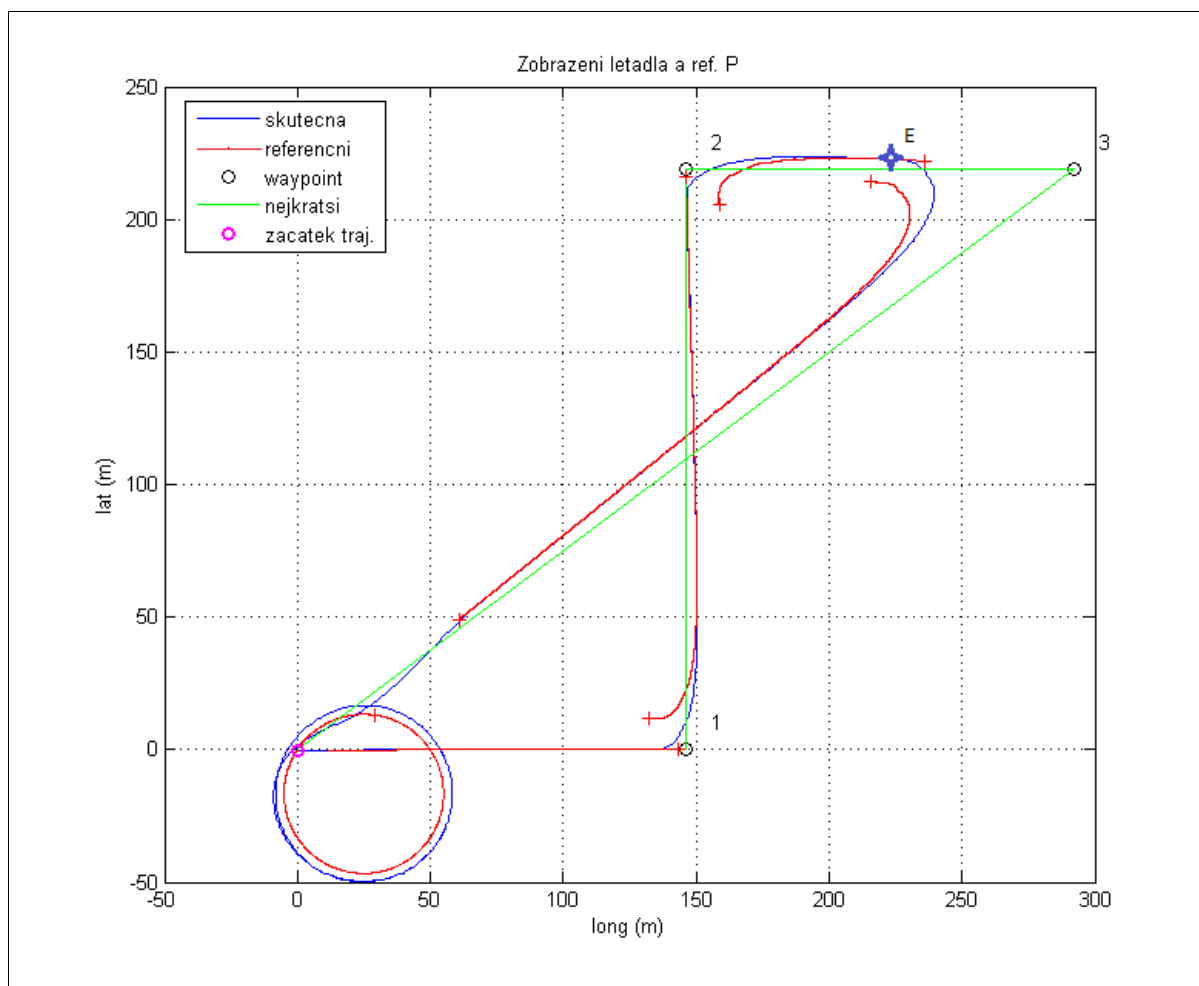
Zhodnocení

Letadlo na základě simulačně nastavených klapěk směřem od ideální trajektorie v prvních 10s letu (v *Ilustrace 24* od domácí stanice k bodu A) v manuálním modu se stabilizací odletí od domácí stanice. Letadlo se nachází v prostoru mimo ideální trajektorii. Po přepnutí do autonomního modu se začne řídit dle referenčního bodu a zamíří k žádanému waypointu. V autonomním modu letadlo využívá všech algoritmů.

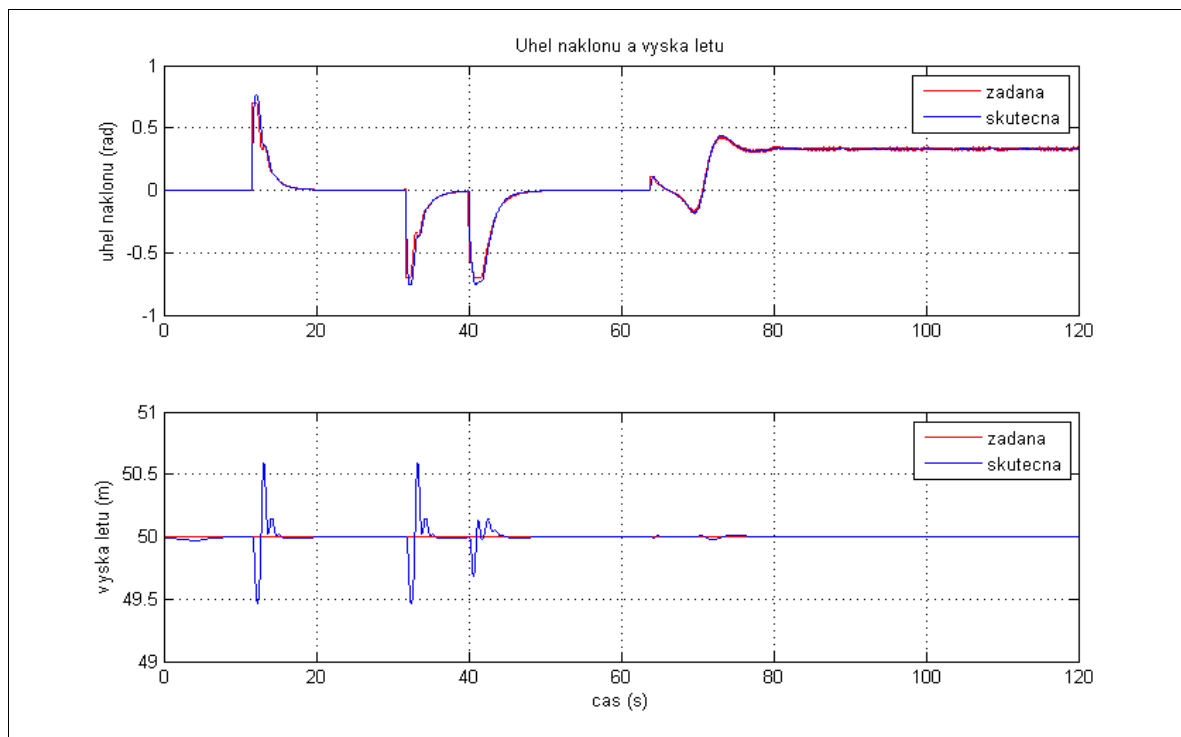
V *Ilustraci 25* lze vidět děj při přepnutí. Úhel náklonu při semi-automatickém modu je jiný, než co autopilot požaduje. Letadlo se neřídí příkazy autopilota. V 10. sekundě dojde k přepnutí do autonomního modu. Regulátory se začnou řídit požadavky z autopilota. Toto se projeví v průběhu žádaného a skutečného úhlu náklonu.

6.2.4 Vybavení chyby

V simulačním prostředí vybavíme chybu. V čase 40s je indikátor chyby přepnut do hodnoty 1 – v chybě. Algoritmus ukončí let a letadlo zamíří bezpodmínečně k domácí stanici.



Ilustrace 26: Zobrazení reakce na vybavení chyby (let k domácí stanici)



Ilustrace 27: Žádané hodnoty a skutečné stavy letadla

Zhodnocení

V *Ilustraci 26* je patrná změna plánování letu (označeno jako bod E). Letadlo na tuto situaci reaguje okamžitou změnou plánu letu. Veškeré další body v seznamu nejsou při tomto stavu dosaženy a letadlo zamíří k domácí stanici. Po dosažení okolí domácí stanice začne letadlo kroužit u domácí stanice.

6.2.5 Změna parametrů

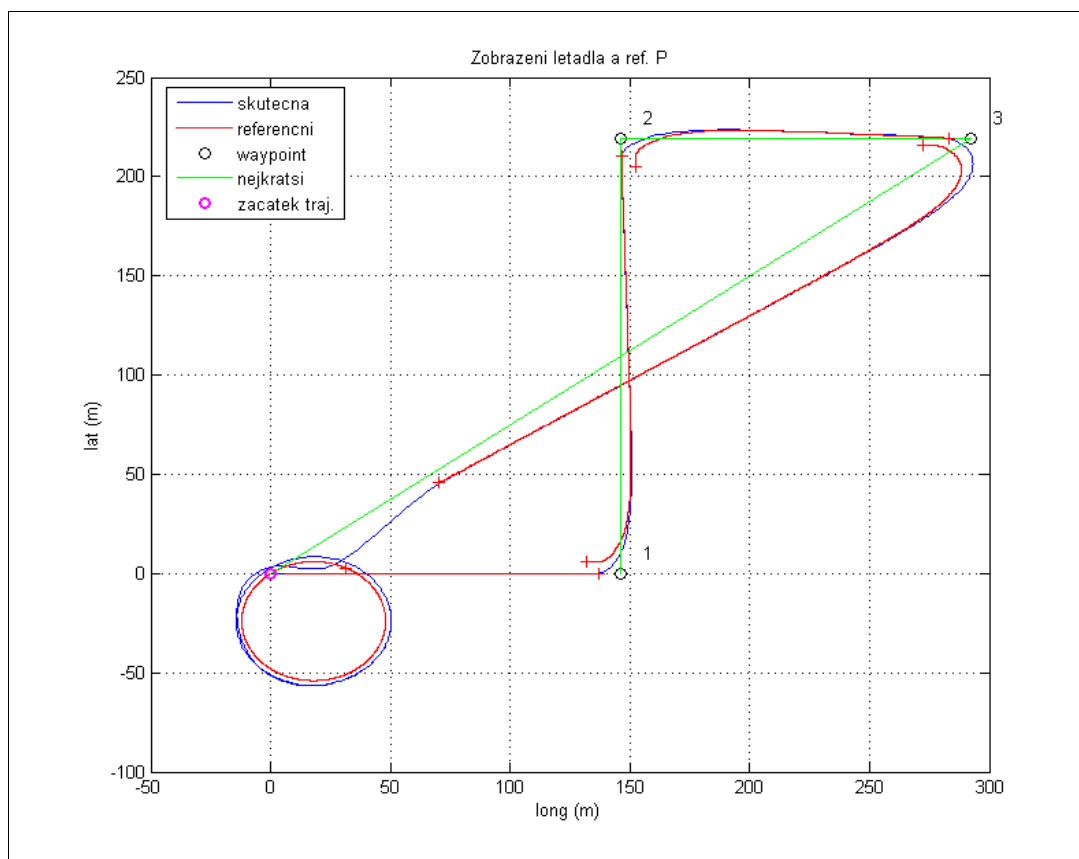
Sledujeme chování při změně volitelných parametrů (popsáno v kapitole 4.4). Parametry jsou voleny jako polovina a dvojnásobek původní hodnoty. Původní hodnoty jsou:

Parametr	Původní hodnota (m)
L	12
R	30
threshold	15

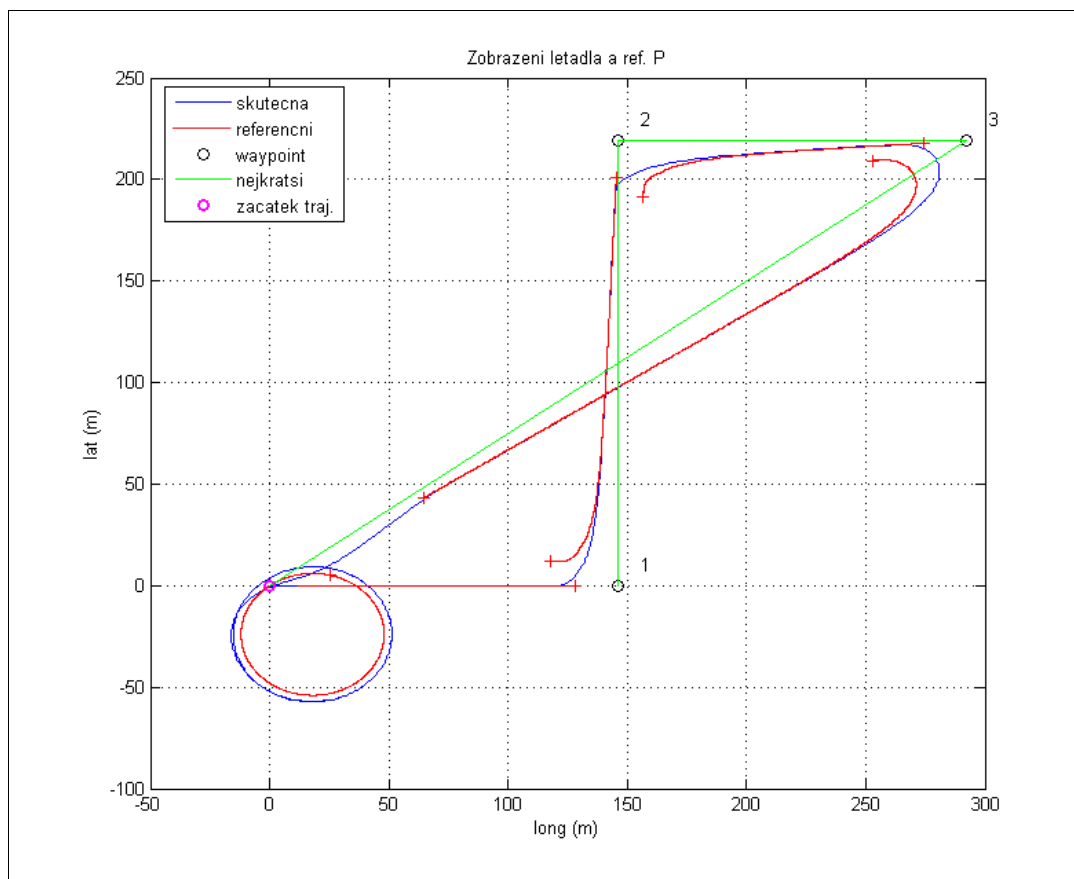
Tabulka 6: Počáteční hodnoty parametrů

6.2.5.1 Změna L

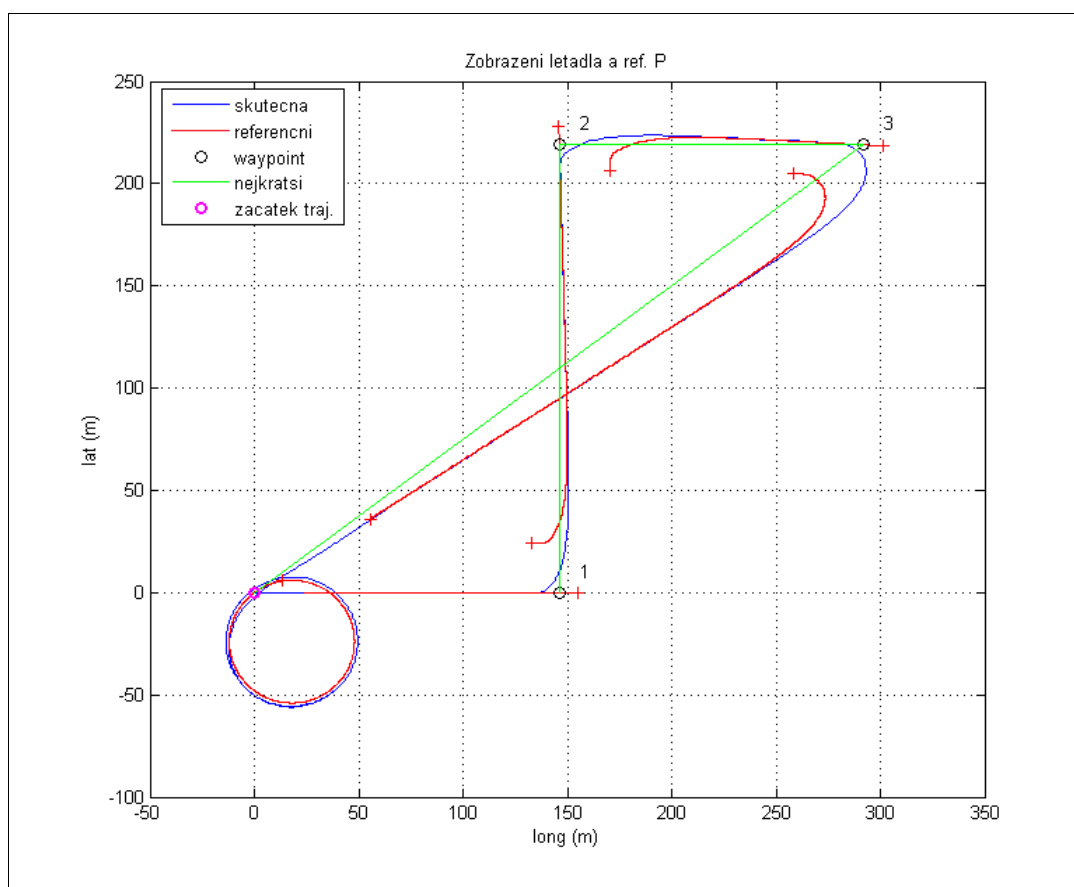
Grafy



Ilustrace 28: Změna L na polovinu původní hodnoty



Ilustrace 29: Původní hodnota L



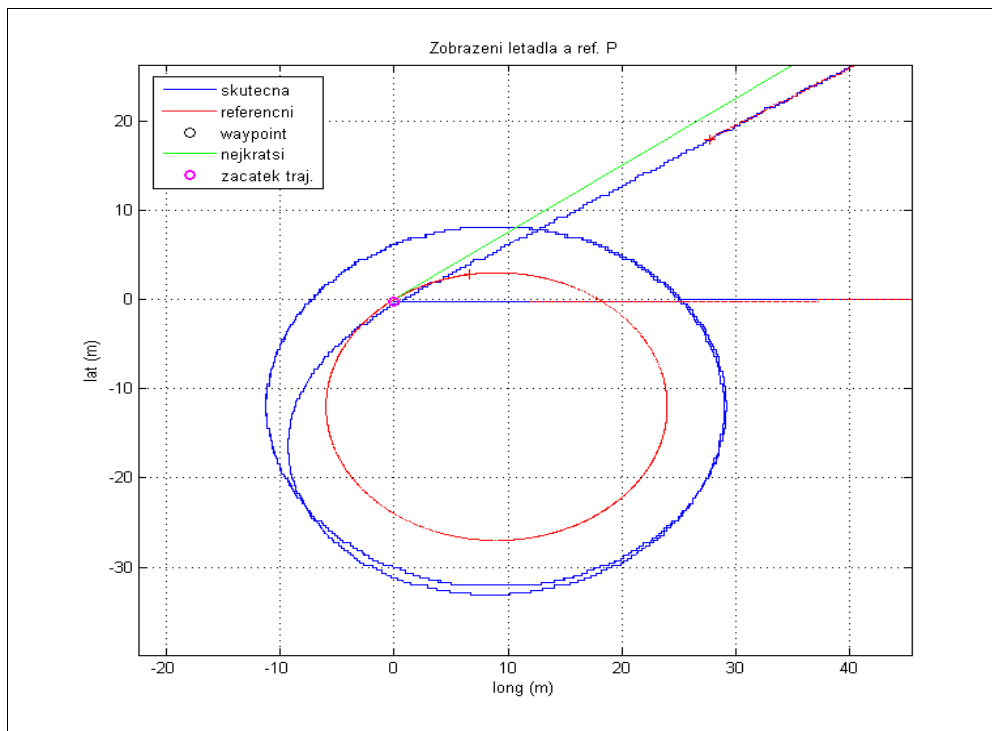
Ilustrace 30: Změna L na dvojnásobek původní hodnoty

Zhodnocení

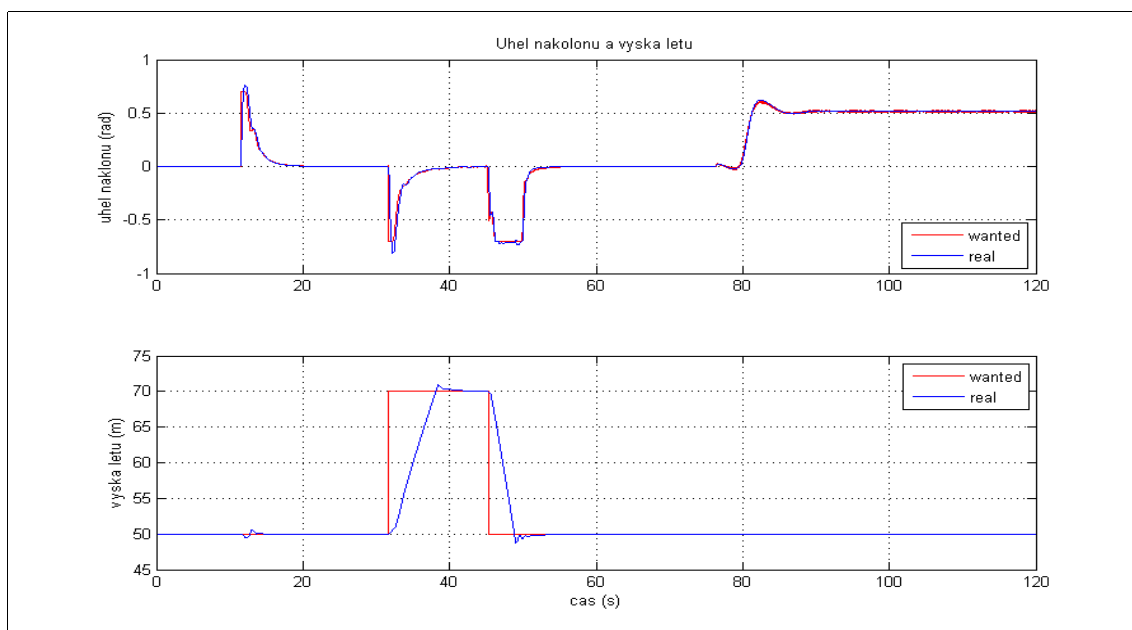
Z Ilustrací 28 – 30 je patrné, že tato hodnota při tomto způsobu plánování nemá výrazný vliv na přesnost průletu bodem. Při kruhovém pohybu vidíme se zvyšující se hodnotou L přiblížení skutečné trajektorie k ideální.

6.2.5.2 Změna R

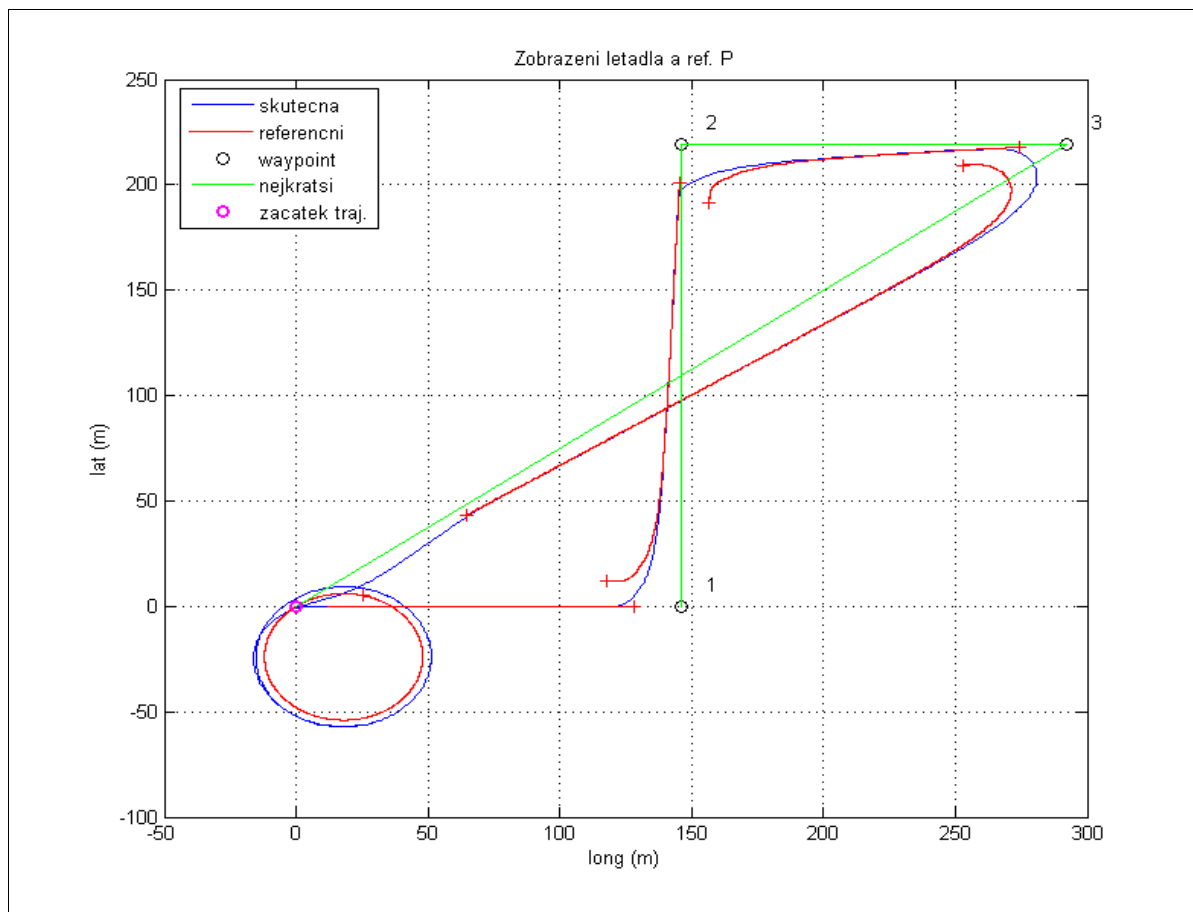
Grafy



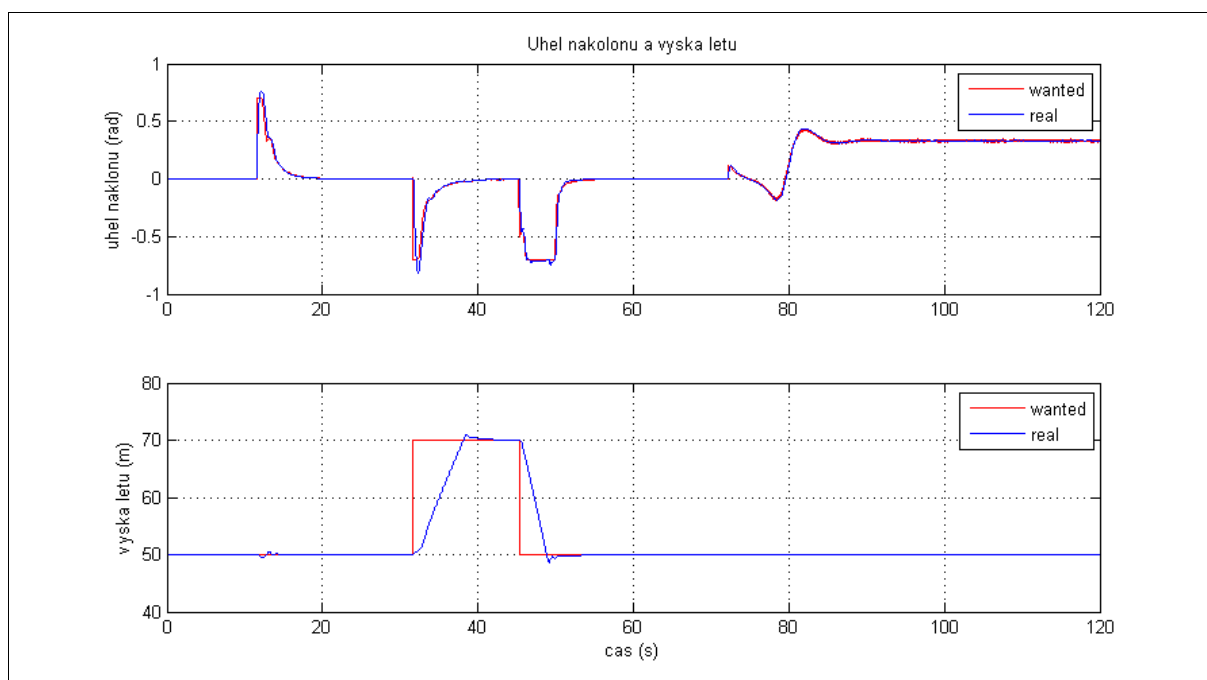
Ilustrace 31: Změna R na polovinu původní hodnoty



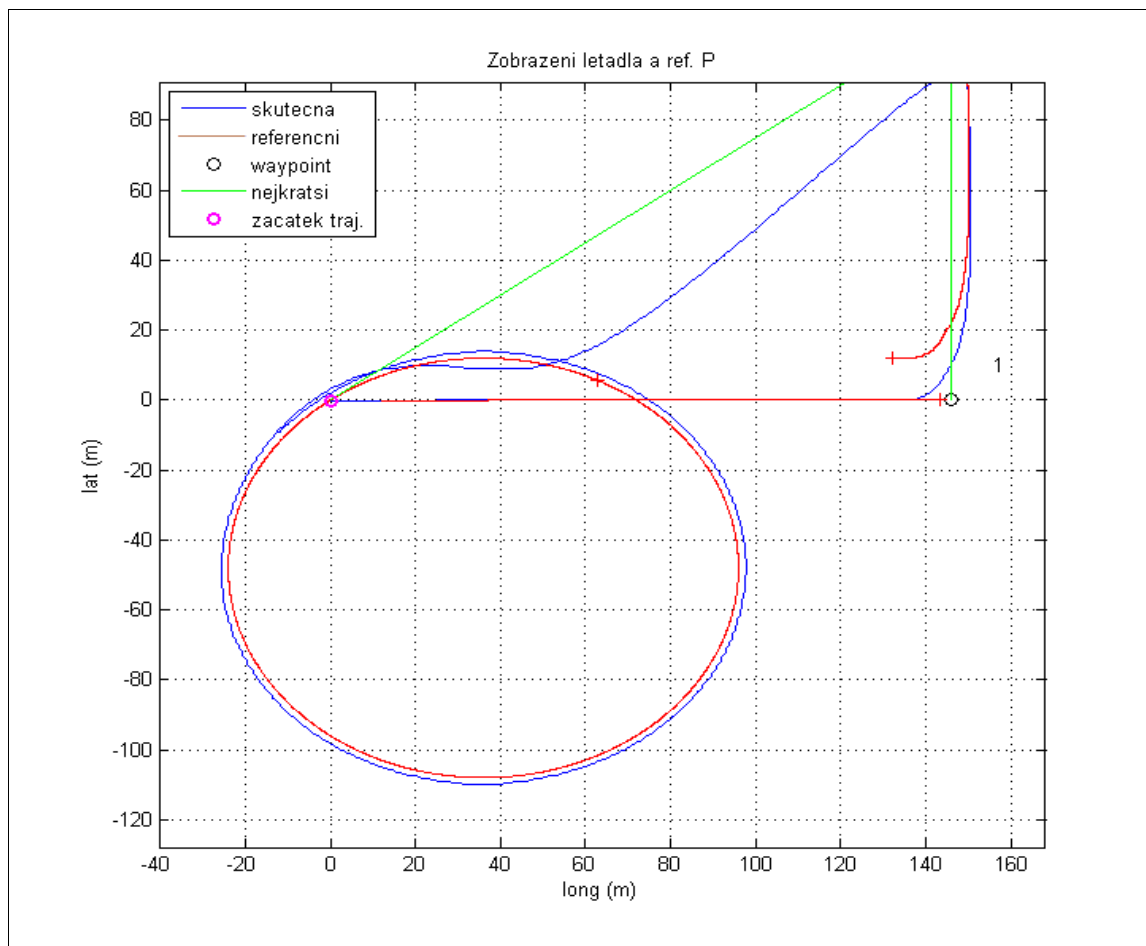
Ilustrace 32: Žádané hodnoty a reálné stavy letadla ($R = 15m$)



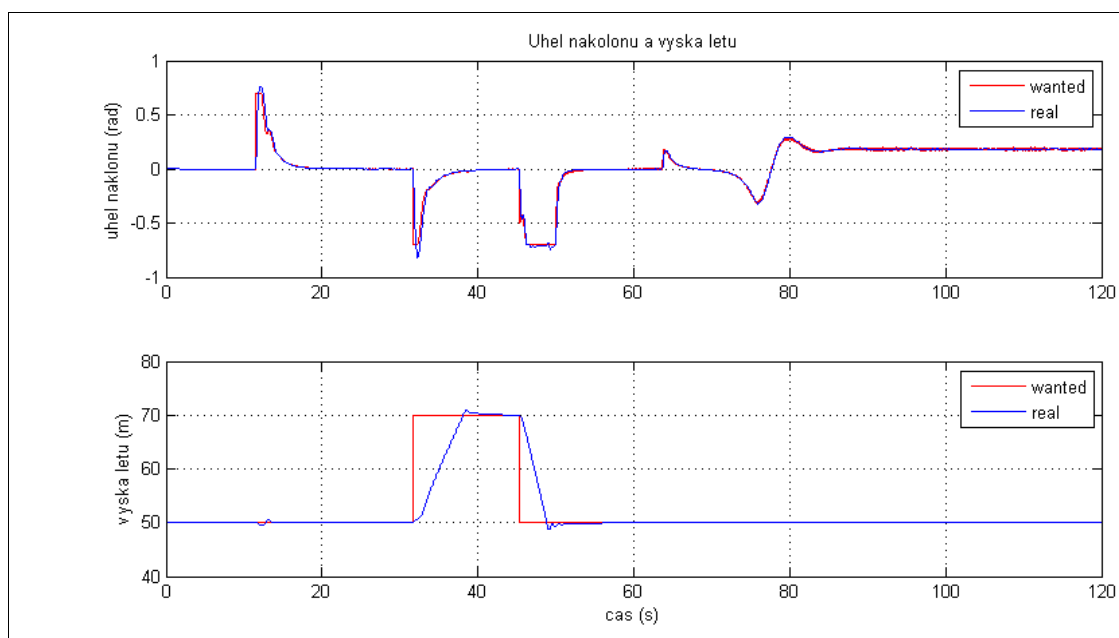
Ilustrace 33: Původní hodnota R



Ilustrace 34: Žádané hodnoty a skutečné stavy letadla ($R = 30m$)



Ilustrace 35: Změna R na dvojnásobek původní hodnoty



Ilustrace 36: Žádané hodnoty a skutečné stavy letadla ($R = 60m$)

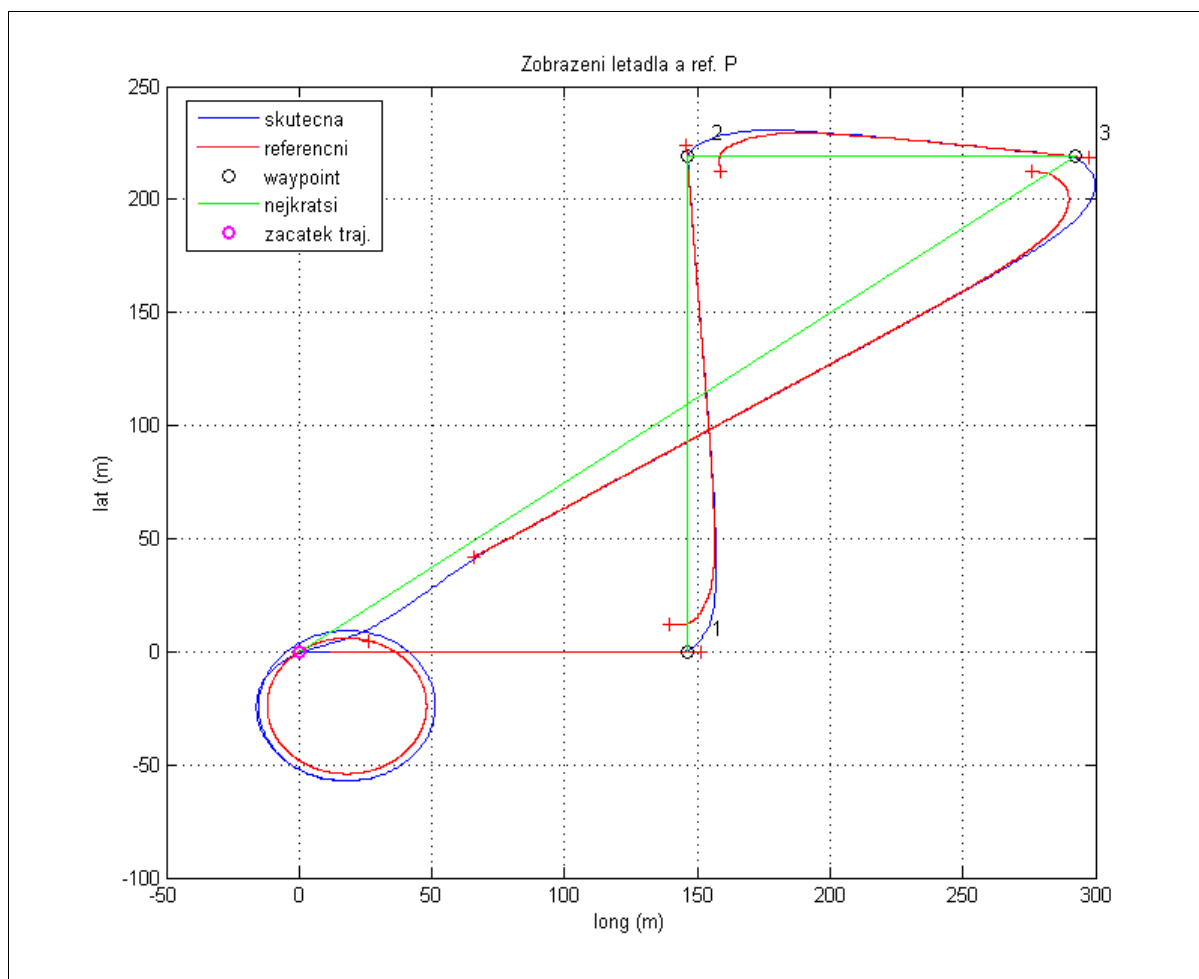
Zhodnocení

Změna hodnoty R se projeví při kruhové trajektorii. Letadlo je schopno kroužit s určitým minimálním poloměrem dle omezení úhlu náklonu. V *Ilustraci 32* je vidět, že při hodnotě $R = 15$ m je žádaný úhel náklonu 0.5 rad. Maximální dovolená hodnota je 0.7 rad. Letadlo by bylo schopno letět po trajektorii s menším poloměrem v ideálním prostředí.

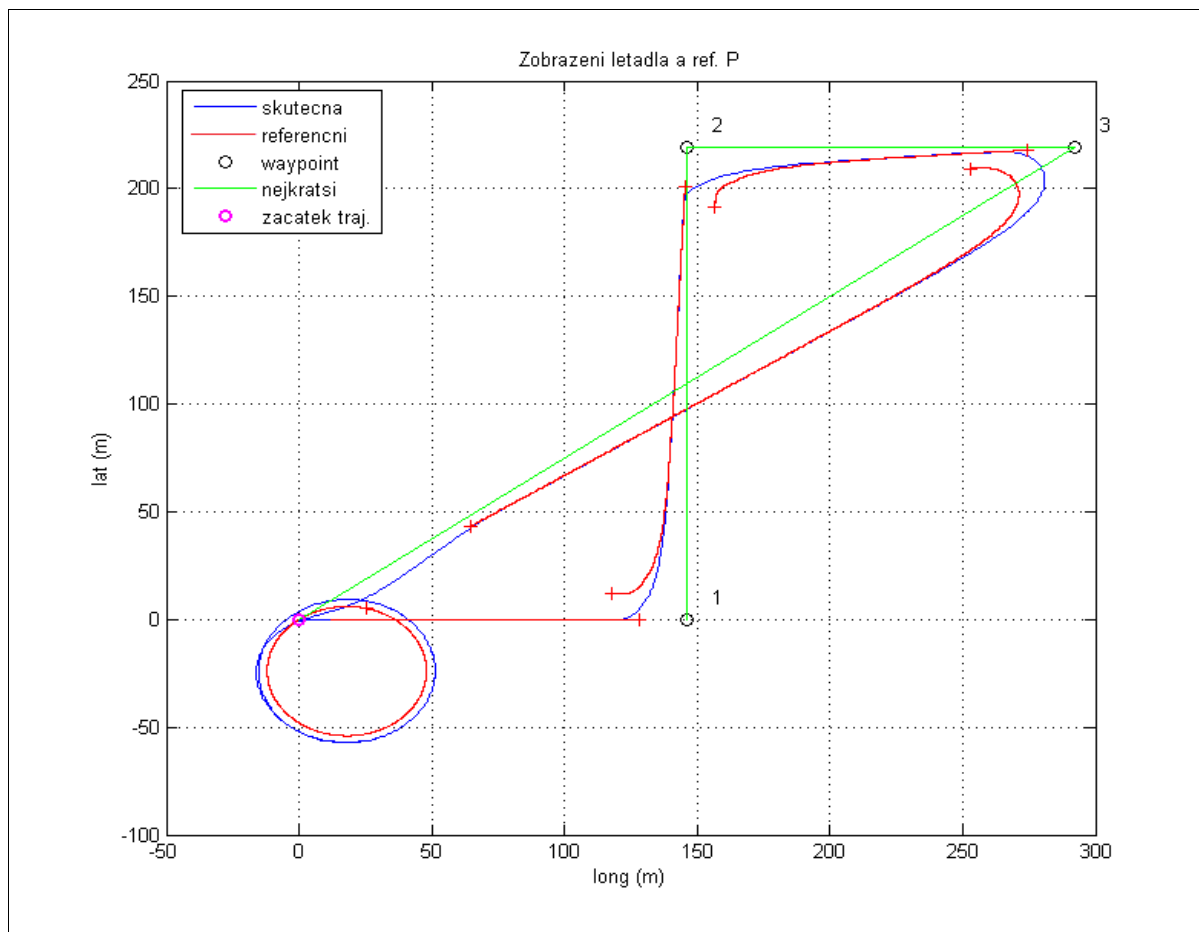
Při uvažování letu v reálných podmínkách prostředí musíme ponechat prostor k manévrování a tedy nepožadovat po letadle let s maximálním úhlem náklonu. Již při kroužení po trajektorii s poloměrem $R = 30$ m je z *Ilustrace 34* zřejmé, že se pohybujeme v okolí poloviny dovoleného úhlu náklonu (letadlo krouží přibližně od 90. sekundy). Tato hodnota se zdá optimální. Při dalším zvětšování poloměru už jsme v oblasti malých úhlů náklonu. Toto je nadměrné omezení a letadlo by se pohybovalo zbytečně daleko.

6.2.5.3 Změna „threshold“

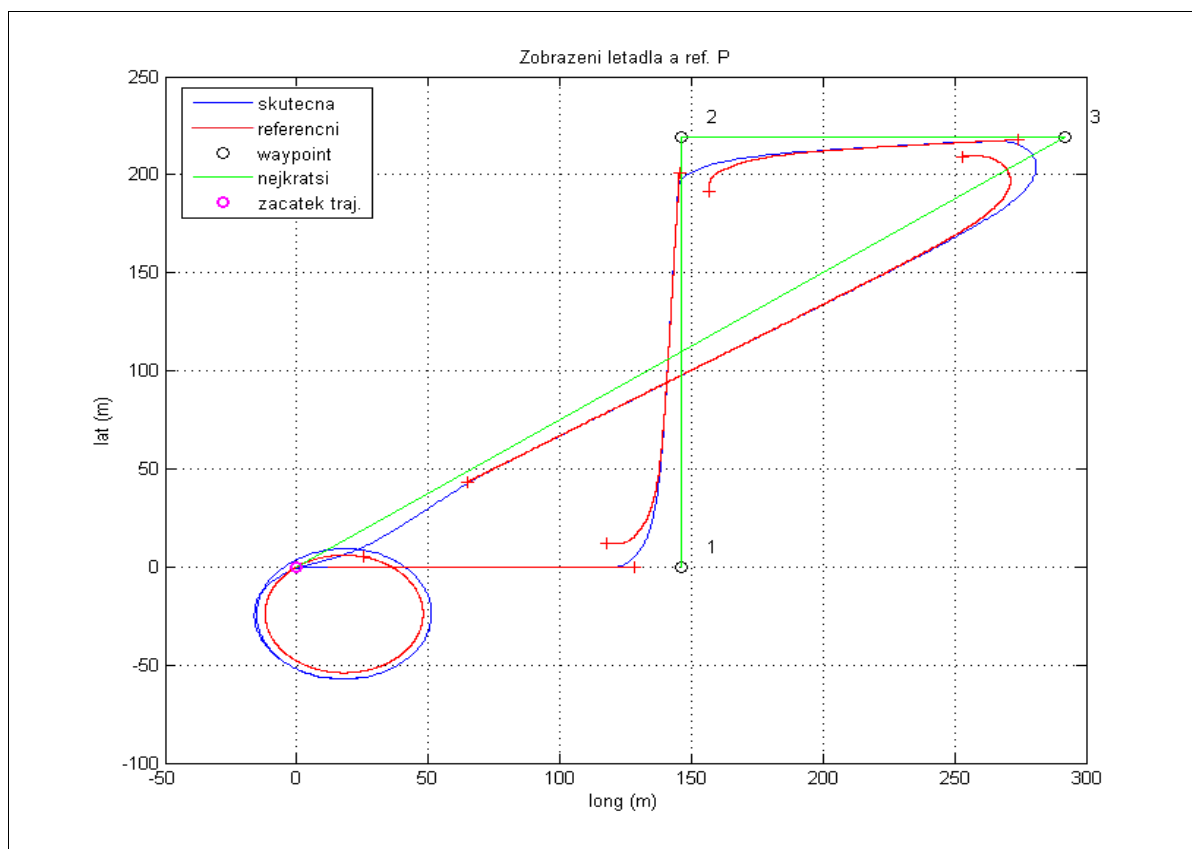
Grafy



Ilustrace 37: Změna threshold na polovinu původní hodnoty



Ilustrace 38: Původní hodnota threshold



Ilustrace 39: Změna threshold na dvojnásobek původní hodnoty

Zhodnocení

Porovnáním *Ilustrací 37 – 39* vidíme, že při zvyšování hodnoty threshold se přesnost průletu bodem snižuje. Je to způsobeno zásahem do podmínek přepínání na další bod. V těchto podmínkách se porovnává vzdálenost letadla od bodu s hodnotou threshold. Podmínka se vybaví dříve, tedy dále od bodu, a letadlo již míří na další waypoint.

V ideálním prostředí se dá tato hodnota velmi snížit. Opět je třeba si uvědomit, že letadlo se bude pohybovat v reálném prostředí a hodnotu threshold podle toho nastavit.

7 Zhodnocení výsledných dat

Pro určení přesnosti průletu naplánovanou trajektorií využijeme průměrnou hodnotu odchylky. V okolí aktuální polohy letadla U hledáme nejbližší bod naplánované trajektorie P v rovině NE. Tuto hodnotu zaznamenejme a takto projdeme celou trajektorií. Z hodnot spočteme průměrnou hodnotu odchylky (rovnice 42).

Tento rozbor provedeme pro různé letové scénáře.

Průměrná hodnota

$$E(X) = \frac{1}{n} \sum_{i=1}^n |UP| \quad (42)$$

Kde: n – počet vzorků statistického souboru (-)

UP – zjištěná minimální vzdálenost P od aktuálně zkoumané polohy U (m)

7.1 Výsledky simulace letu v Matlab/Simulink

V simulačním prostředí se zaměříme na klasické plánování, speciální polohu bodů a chování při změně parametrů (dle kapitoly 6.2.5)

7.1.1 Přímá trajektorie s pravoúhlou zatáčkou

Letadlo necháme proletět trajektorií dle klasického plánování ve stejné letové výšce. Následně spočteme průměrnou odchylku skutečné trajektorie od referenční. Odchylka je minimální vzdálenost aktuální polohy letadla od referenční trajektorie bez ohledu na časový posun. Simulace probíhá v diskrétních časových krocích. Porovnáním intervalu reálné trajektorie s bodem na referenční trajektorii a vyhledáním minima určíme minimální vzdálenost mezi reálnou a referenční trajektorií.

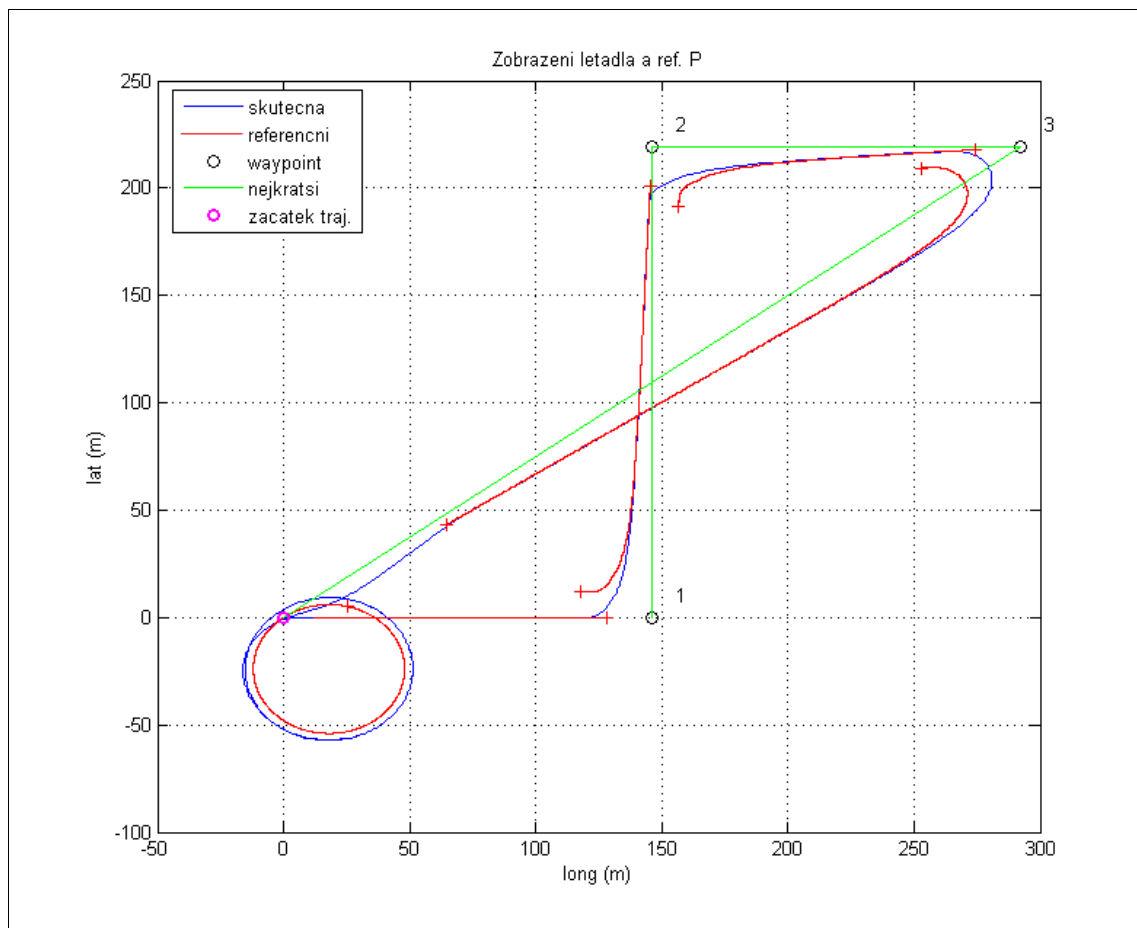
V *Ilustraci 40* vidíme, že reálná trajektorie letadla není příliš daleko od ideální. Vyhodnotíme průběh odchylky (viz. *Ilustrace 41*) po začátek kruhové trajektorie kolem domácí stanice.

Průměrná odchylka je 6,61 m pro zadanou konfiguraci bodů.

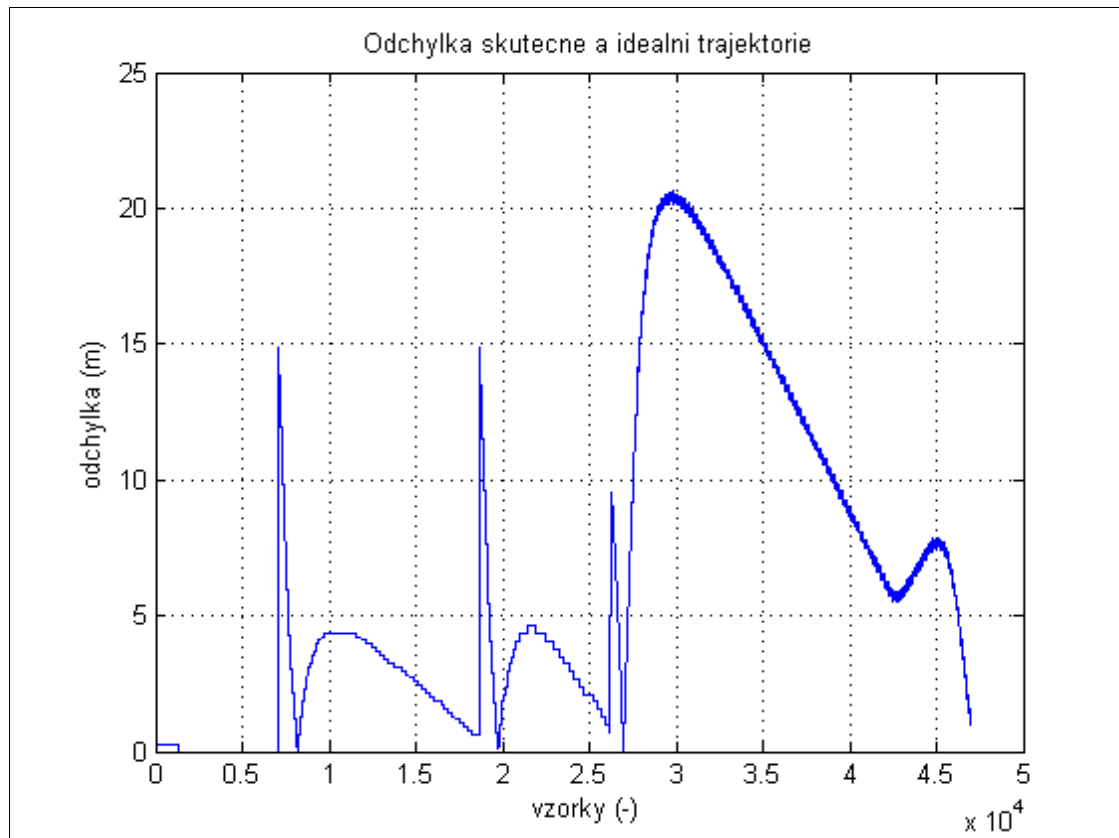
Poznámka:

V následujících grafech průběhů odchylek u krouživých tr si můžeme všimnout, že průběh se jeví jako šumící hodnota. Pohybuje se ovšem v simulaci s diskrétním krokem výpočtu. Principem výpočtu vznikají skokové změny odchylky v malém rozsahu. Toto se projeví hlavně u krouživé trajektorie.

U trajektorie přímé je vidět velká skoková změna při přepnutí aktuálního waypointu. Tato změna je způsobena skokovou změnou referenční trajektorie.



Ilustrace 40: Zobrazení reálné a ideální trajektorie - klasické plánování letu



Ilustrace 41: Odchylka reálné a ideální trajektorie - klasické plánování letu

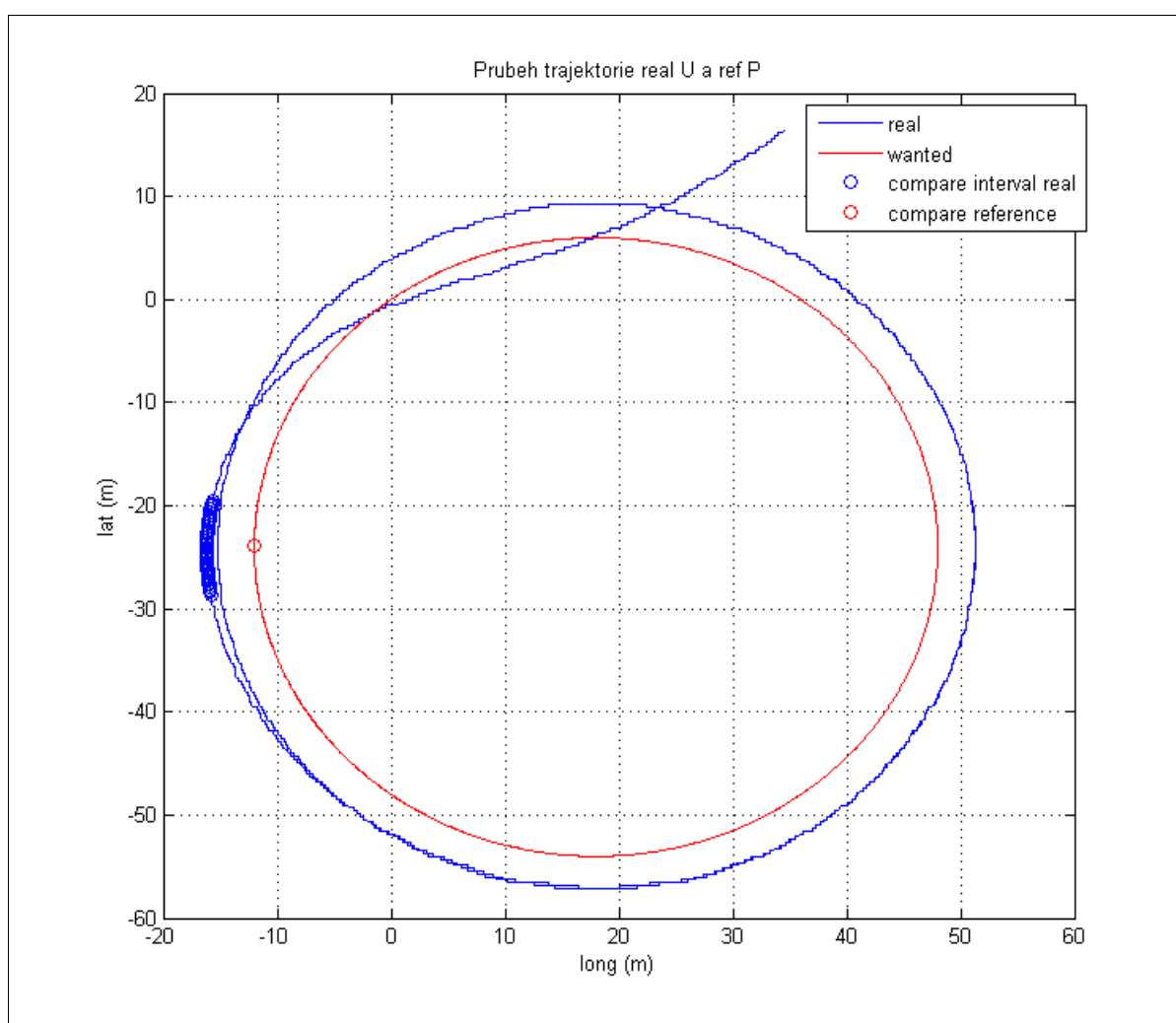
7.1.2 Kruhová trajektorie kolem bodu

Letadlo necháme kroužit kolem domácí stanice. Za ideální trajektorii považujeme vypočtenou žádanou trajektorii dle *Ilustrace 42*. odchylka je minimální vzdálenost aktuální polohy letadla od žádané trajektorie.

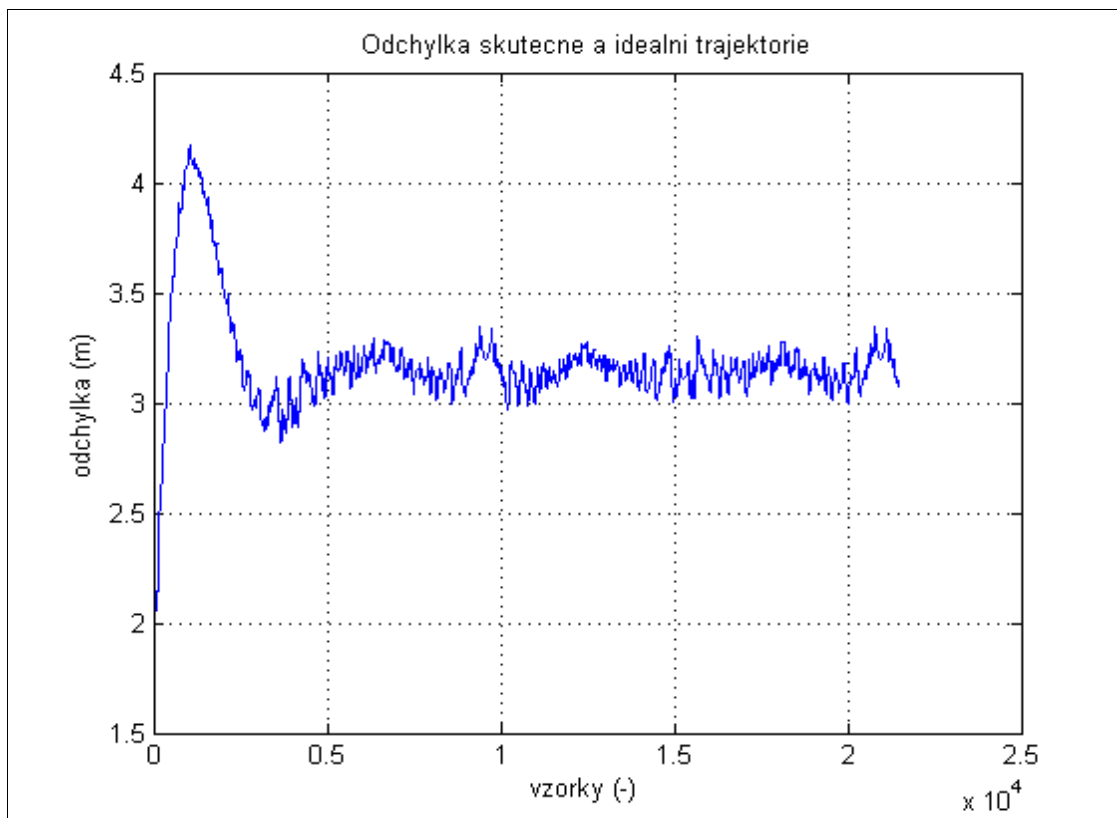
Výpočtem byla zjištěna průměrná odchylka 3,17 m při zvolených parametrech.

Ilustrace 42 zobrazuje žádanou a skutečnou trajektorii při krouživém letu a naznačuje výpočet minimální odchylky od žádané trajektorie. K výpočtu je využit úsek reálné trajektorie. Porovnáním vzdáleností se zjistí minimum a toto se použije pro výpočet průměrné odchylky. Lze vidět, že při začátku krouživé trajektorie letadlo nejdříve protne žádanou trajektorii a pak se stabilizuje na kružnici.

Ilustrace 43 zobrazuje průběh odchylky po celé trajektorii krouživého letu kolem domácí stanice. Trajektorie obsahuje dvě kružnice kolem domácí stanice.



Ilustrace 42: zobrazení žádané a reálné trajektorie - plánování kroužení



Ilustrace 43: Průběh odchylky reálné trajektorie od žádané - krouživý let

7.1.3 Změna parametrů algoritmu

Při změně parametrů L , R a threshold sledujeme chování algoritmu z hlediska přesnosti. Počáteční hodnoty jsou stejné jako v kapitole 6.2.5. (dle Tabulka 6: Počáteční hodnoty parametrů).

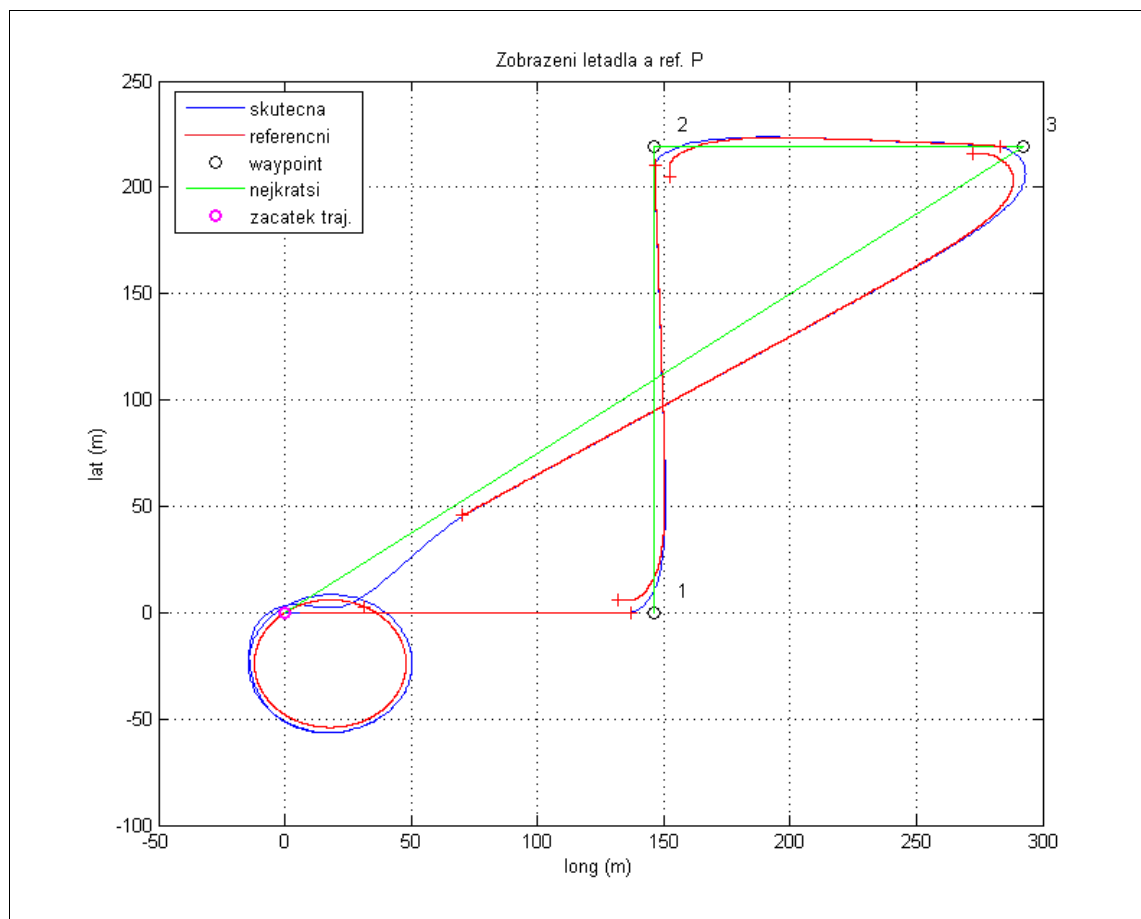
7.1.3.1 Chování při změně vzdálenosti referenčního bodu - L

Vzdálenost L volíme poloviční a dvojnásobnou vzhledem k původní hodnotě L_0 . Vzdálenost L se projeví v odchylce od referenční trajektorie při obou typech plánování (klasické i krouživé).

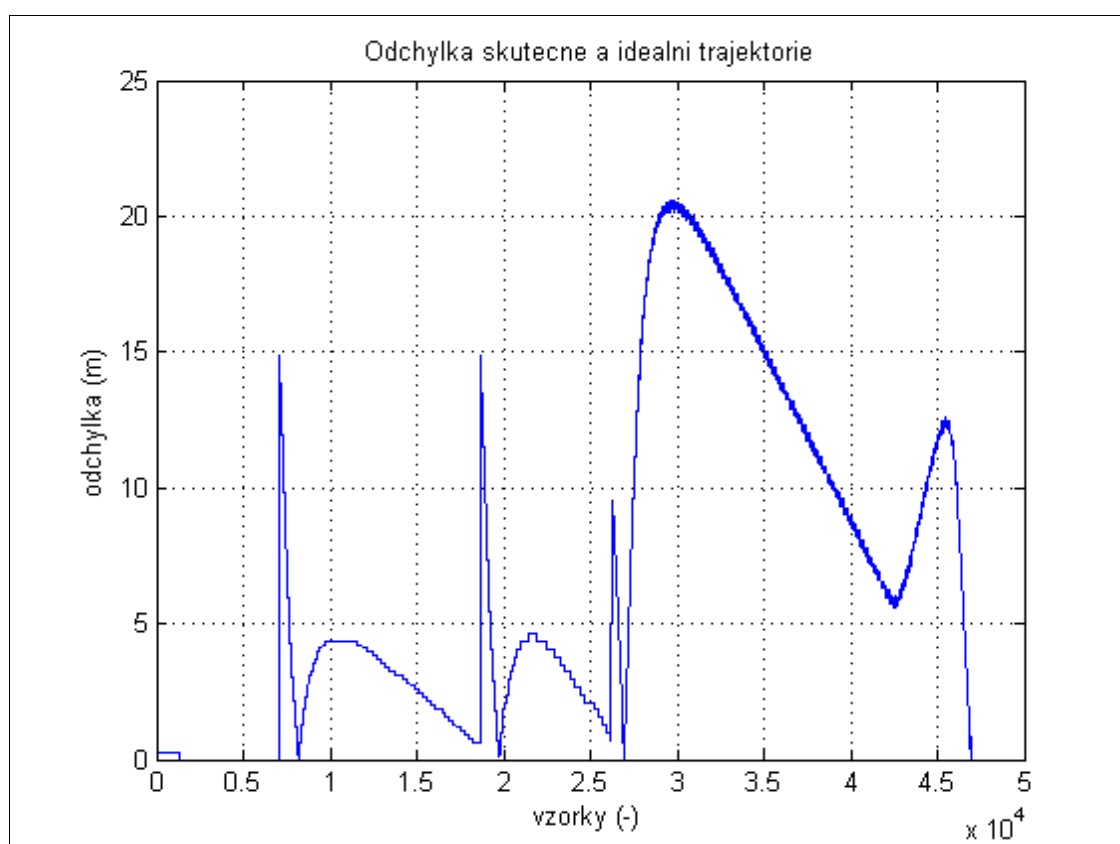
Z výsledných hodnot průměrné odchylky uvedené v Tabulce 7 je zřejmé, že při hodnotě $L = 2 \cdot L_0$ je průměrná hodnota odchylky menší než odchylka při hodnotě L_0 .

Vzdálenost L (m)	Průměrná hodnota odchylky $E(X)$ (m)
Klasické plánování	
$L = 6$	$E(X) = 6,84$
$L = 24$	$E(X) = 6,35$
Kruhové plánování	
$L = 6$	$E(X) = 2,33$
$L = 24$	$E(X) = 1,47$

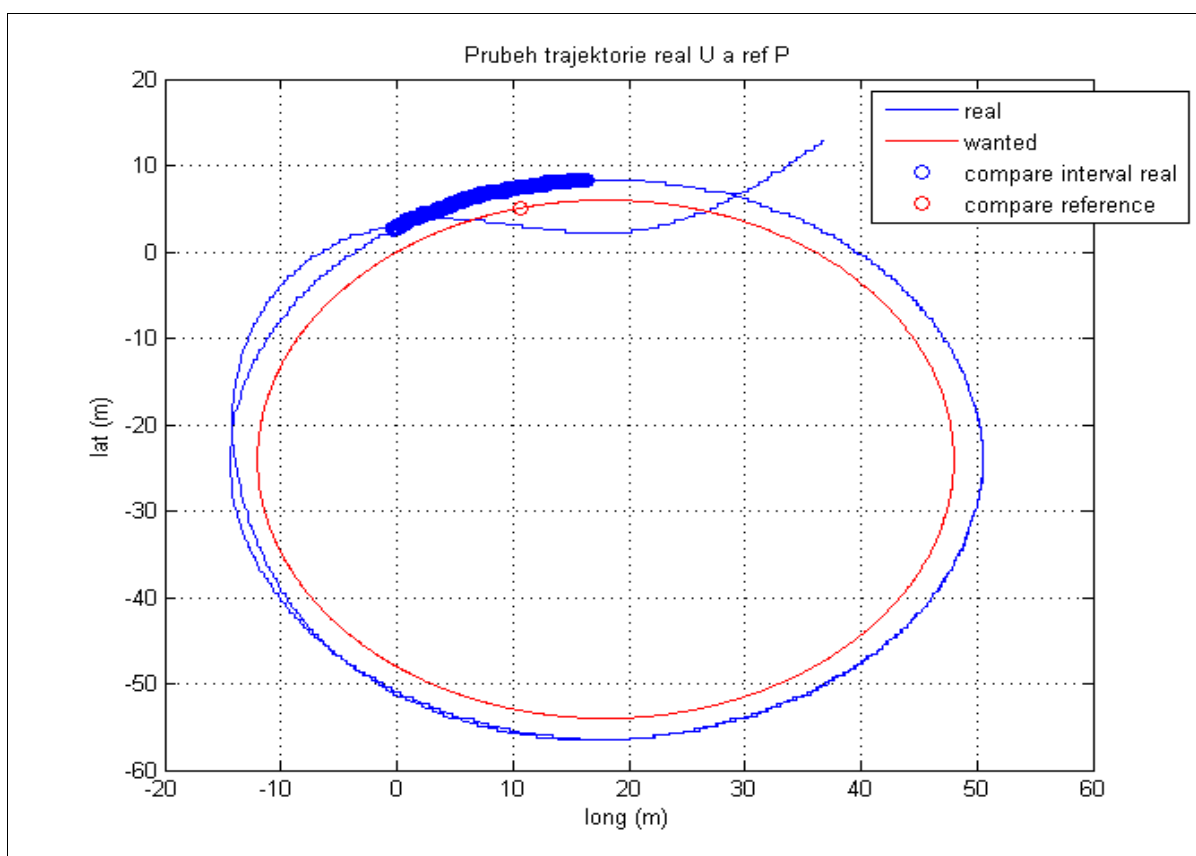
Tabulka 7: Přehled průměrných hodnot odchylek při změně L



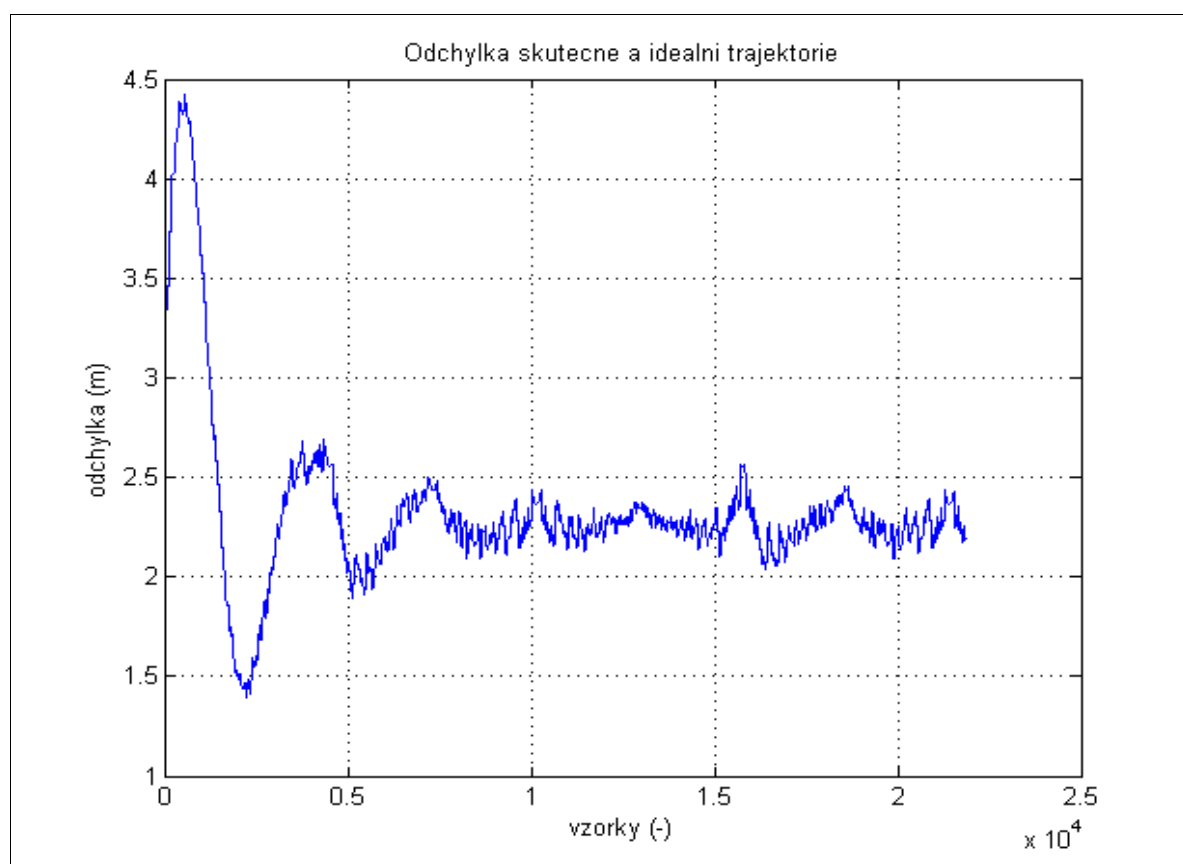
Ilustrace 44: Zobrazení reálné a ideální trajektorie ($L = 6m$)



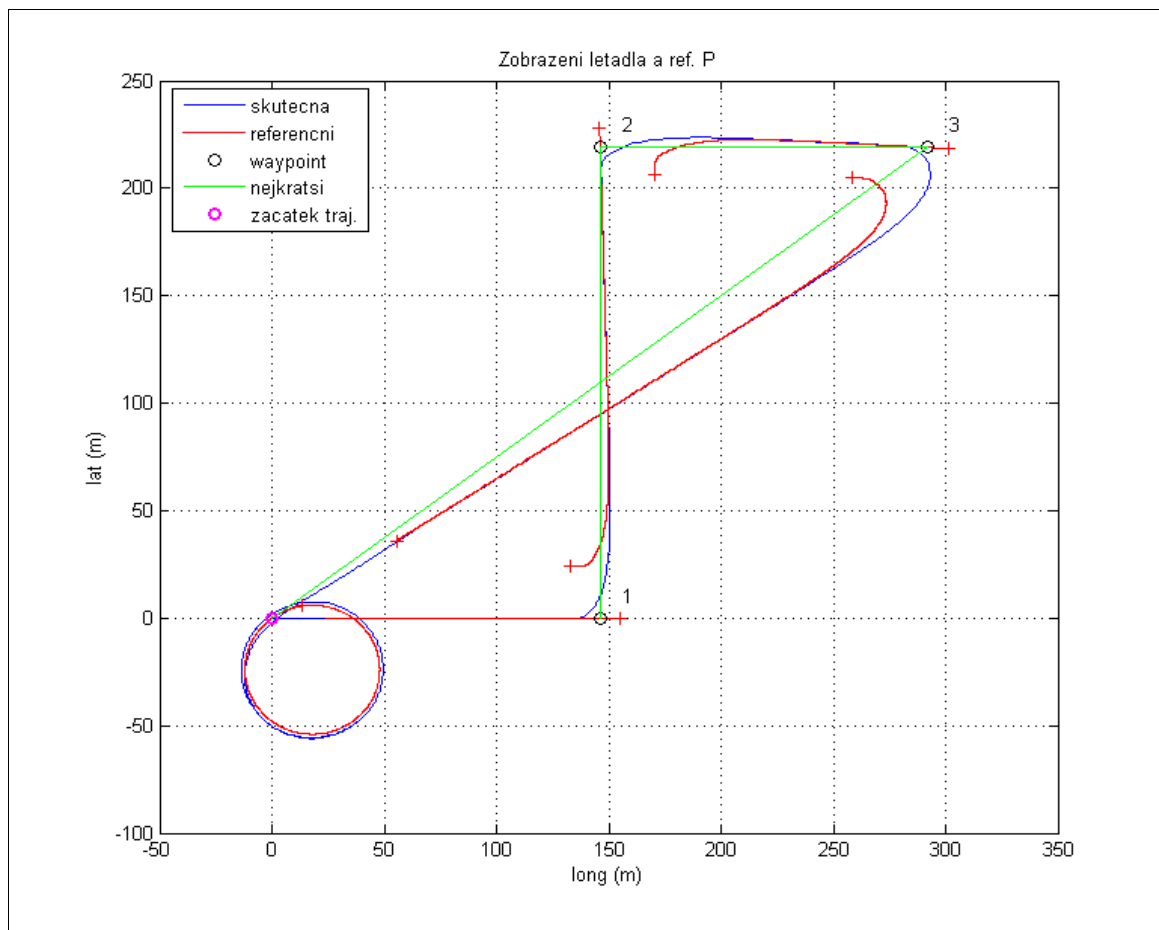
Ilustrace 45: Průběh odchylky ($L = 6m$)



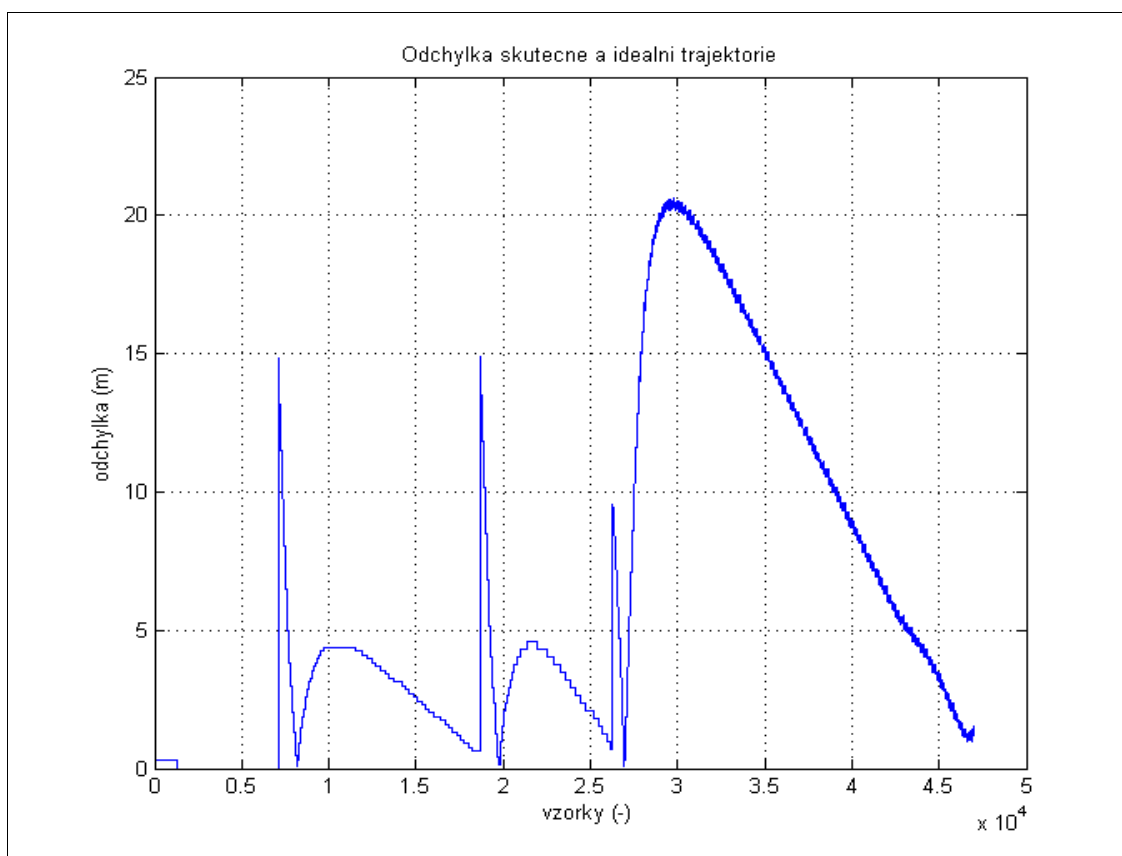
Ilustrace 46: Zobrazení reálné a žádané trajektorie při krouživém letu ($L = 6m$)



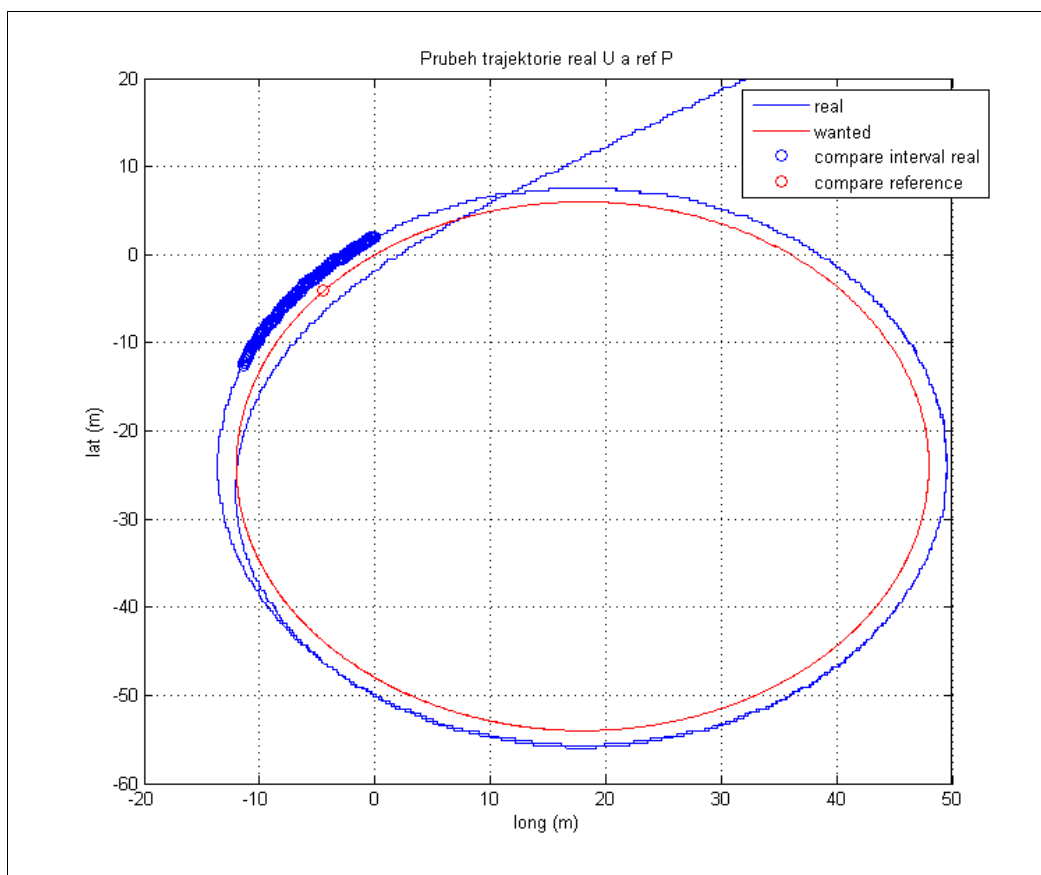
Ilustrace 47: Průběh odchylky pro celou kruhovou trajektorii ($L = 6m$)



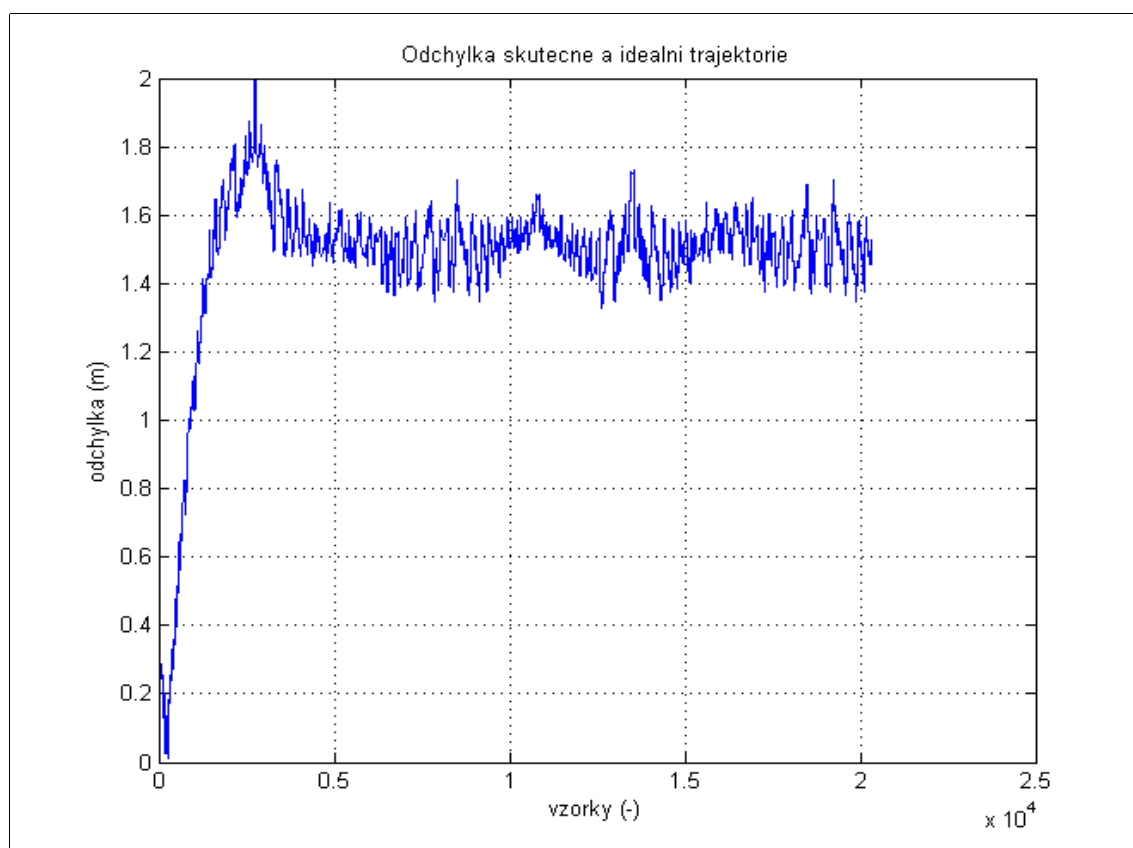
Ilustrace 48: Zobrazení reálné a ideální trajektorie ($L = 24m$)



Ilustrace 49: Průběh odchylky ($L = 24m$)



Ilustrace 50: Průběh reálné a žádané trajektorie při kroužení ($L = 24\text{m}$)



Ilustrace 51: Průběh odchylky pro celou kruhovou trajektorii ($L = 24\text{m}$)

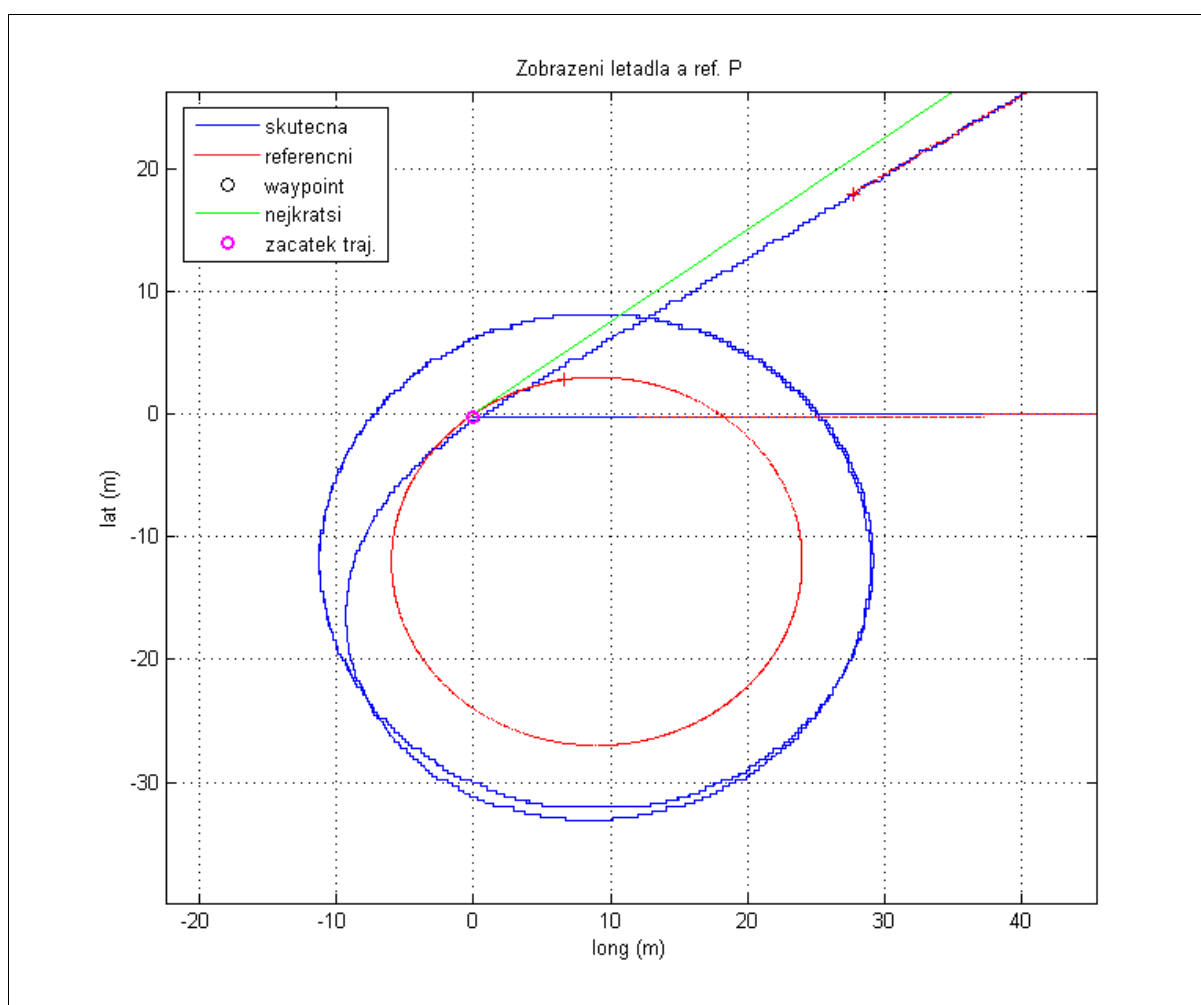
7.1.3.2 Chování při změně poloměru krouživé trajektorie - R

Parametr R se projeví pouze u kroužení kolem bodu. Bude vyhodnocena pouze odchylka při kroužení.

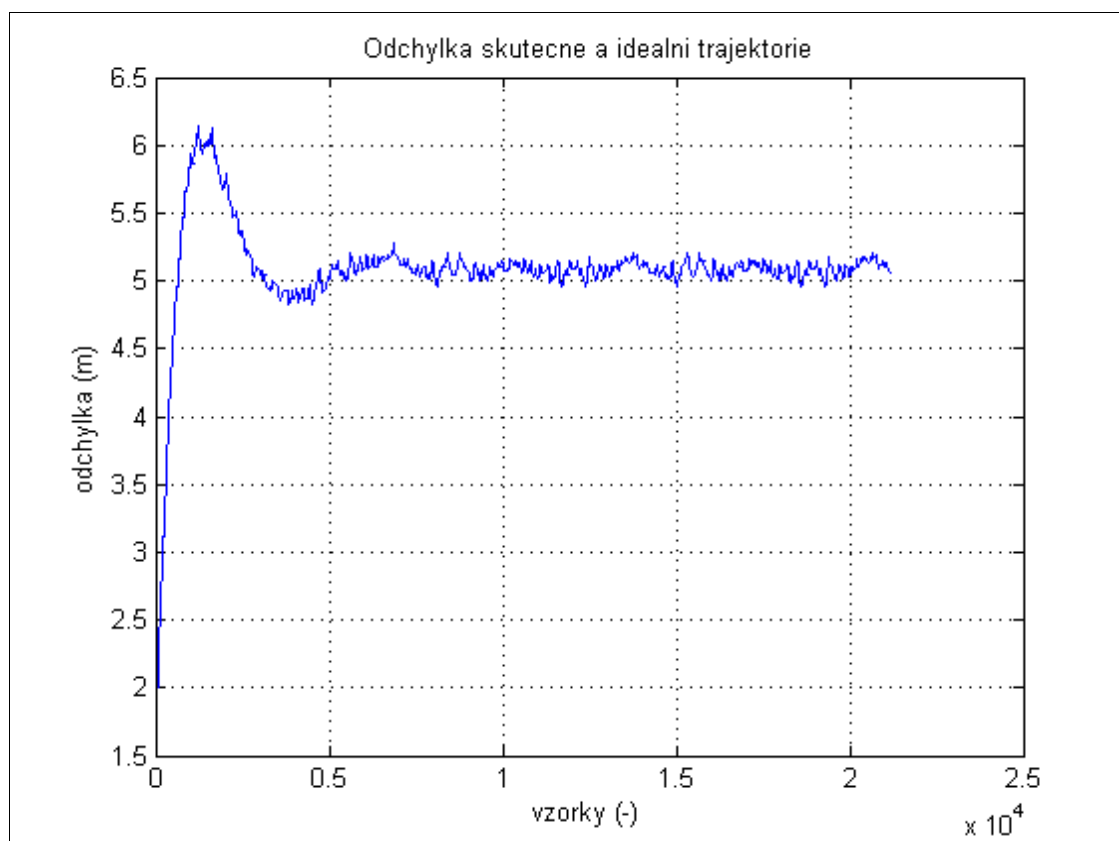
Z grafů na *Ilustracích 52 – 55* je vidět změnu průběhu trajektorie při změně parametru R. Z výsledných hodnot průměrné odchylky reálné trajektorie od žádané, uvedených v *Tabulce 8*, lze vyčíst, že při zvýšení poloměru a zachování ostatních parametrů se reálná trajektorie přiblíží k žádané.

Poloměr R (m)	Průměrná hodnota odchylky E(X) (m)
Kruhové plánování	
R = 15	E(X) = 5,09
R = 60	E(X) = 1,67

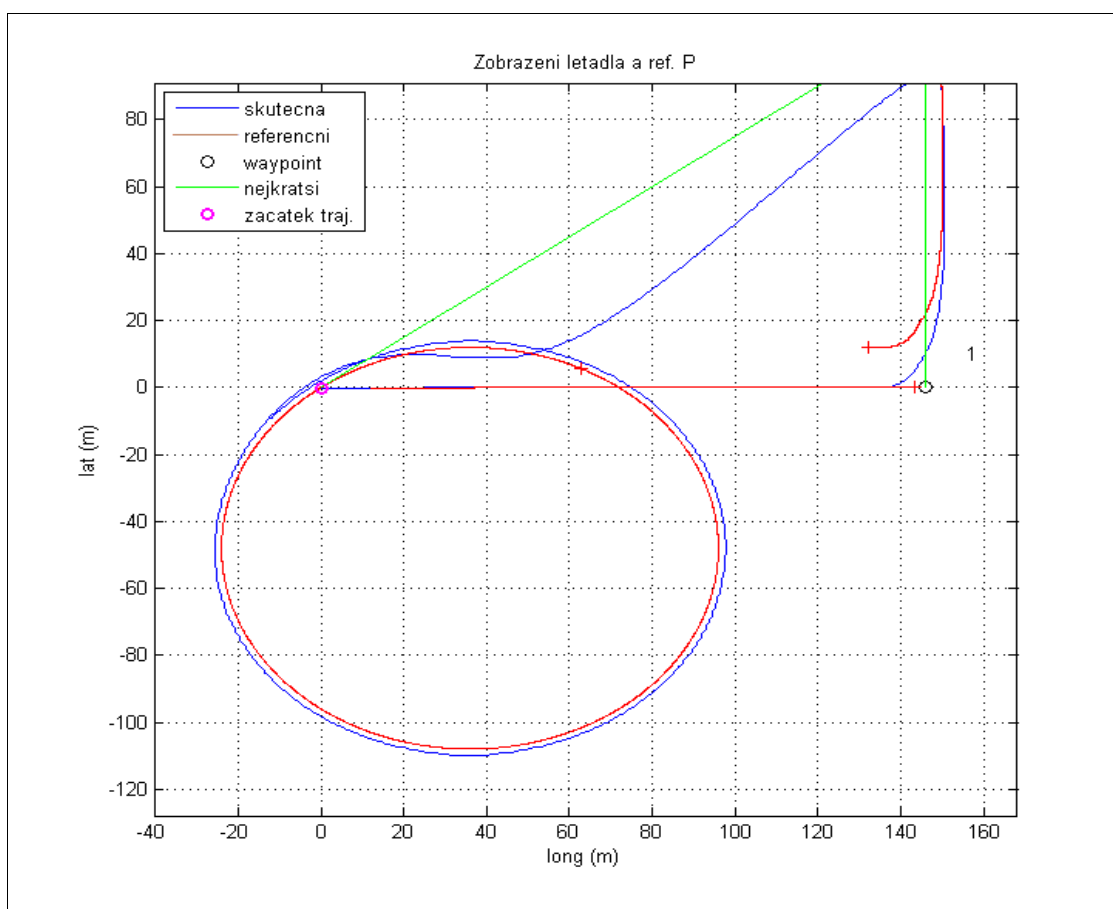
Tabulka 8: Přehled průměrných hodnot odchylek při změně parametru R



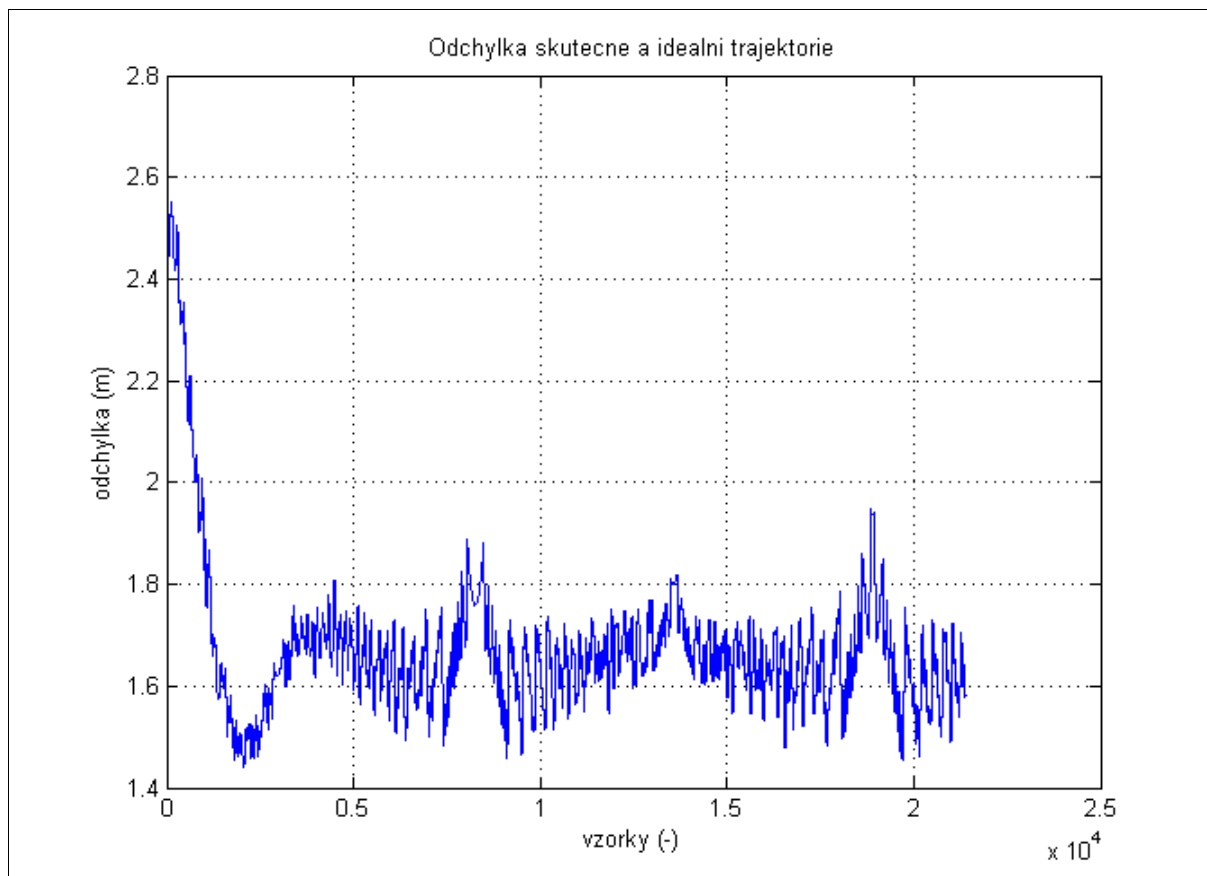
Ilustrace 52: Zobrazení reálné a žádané trajektorie (R = 15m)



Ilustrace 53: Průběh odchylky při kroužení ($R = 15m$)



Ilustrace 54: Zobrazení reálné a žádané trajektorie ($R = 60m$)



Ilustrace 55: Průběh odchylky ($R = 60m$)

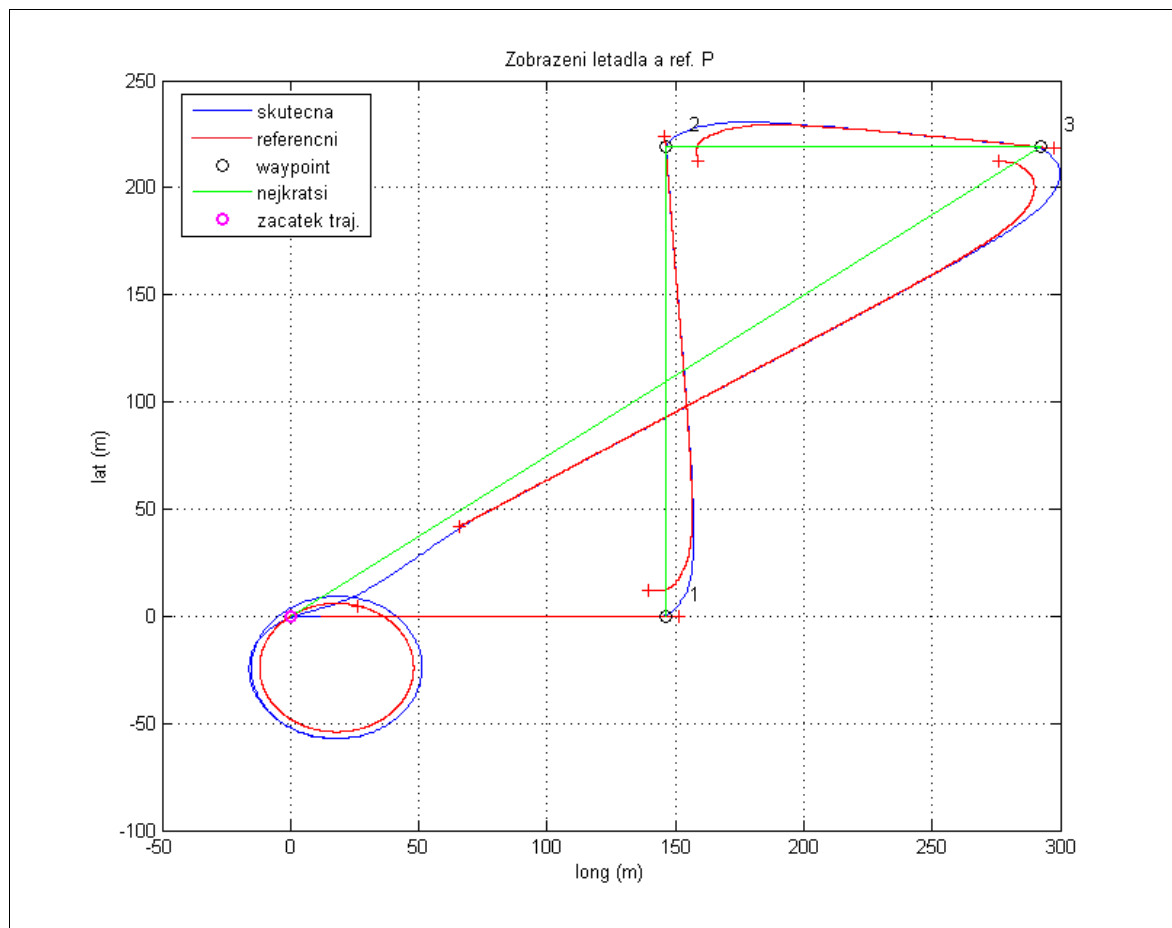
7.1.3.3 Chování při změně hodnoty threshold

Parametr threshold se projeví pouze u klasického plánování. Při krouživém nemá na přesnost vliv. V *Tabulce 9* vidíme, že příliš malá nebo příliš velká hodnota threshold má vliv na přesnost průletu trajektorií.

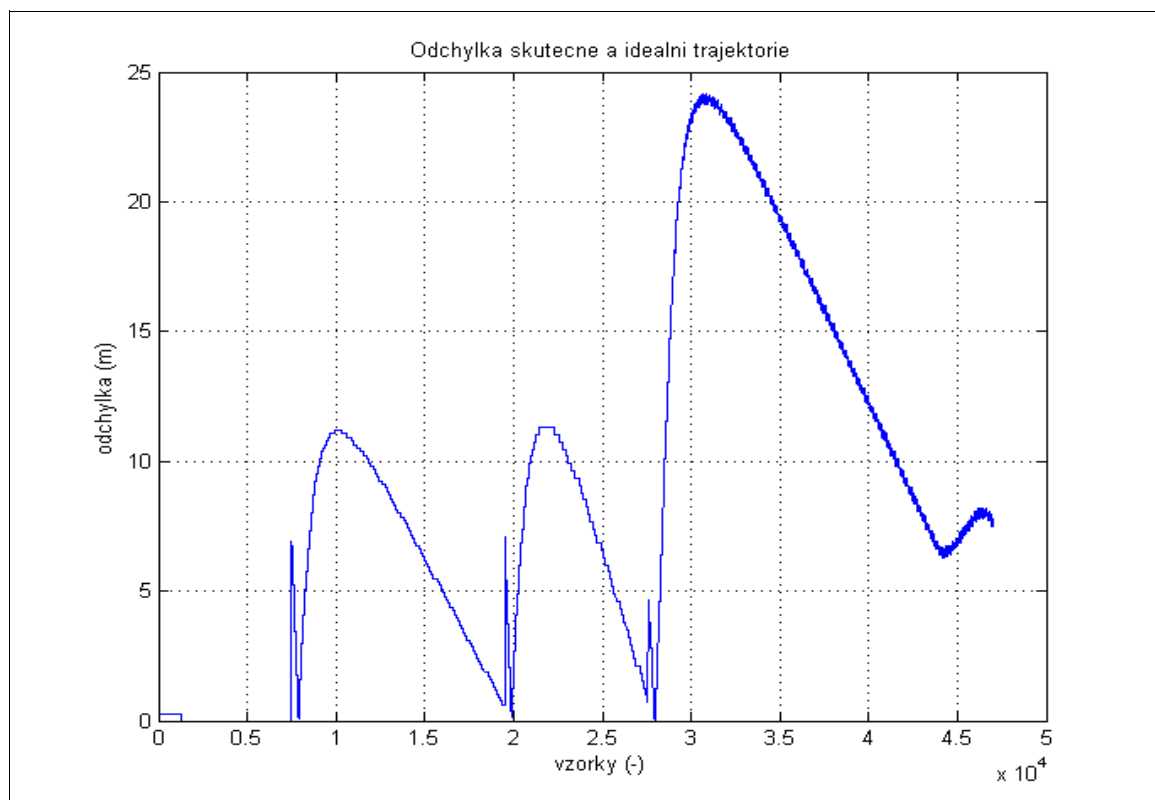
V *Ilustraci 56* lze vidět, že odchylka pro malé hodnoty T je způsobena setrvačností letadla v daném směru. Tím se zvýší kolmá vzdálenost od ideální trajektorie a tedy přesnost průletu ideální trajektorií proti původnímu nastavení parametru. Naopak při pohledu na *Ilustraci 58* letadlo viditelně nedoletí k danému bodu a algoritmus již vybaví podmínku přepnutí k dalšímu. Odchylka tedy vzniká předčasným stočením letadla k dalšímu bodu zájmu v seznamu.

Parametr threshold - T (m)	Průměrná hodnota odchylky $E(X)$ (m)
Klasické plánování	
$T = 7$	$E(X) = 8,90$
$T = 30$	$E(X) = 7,55$

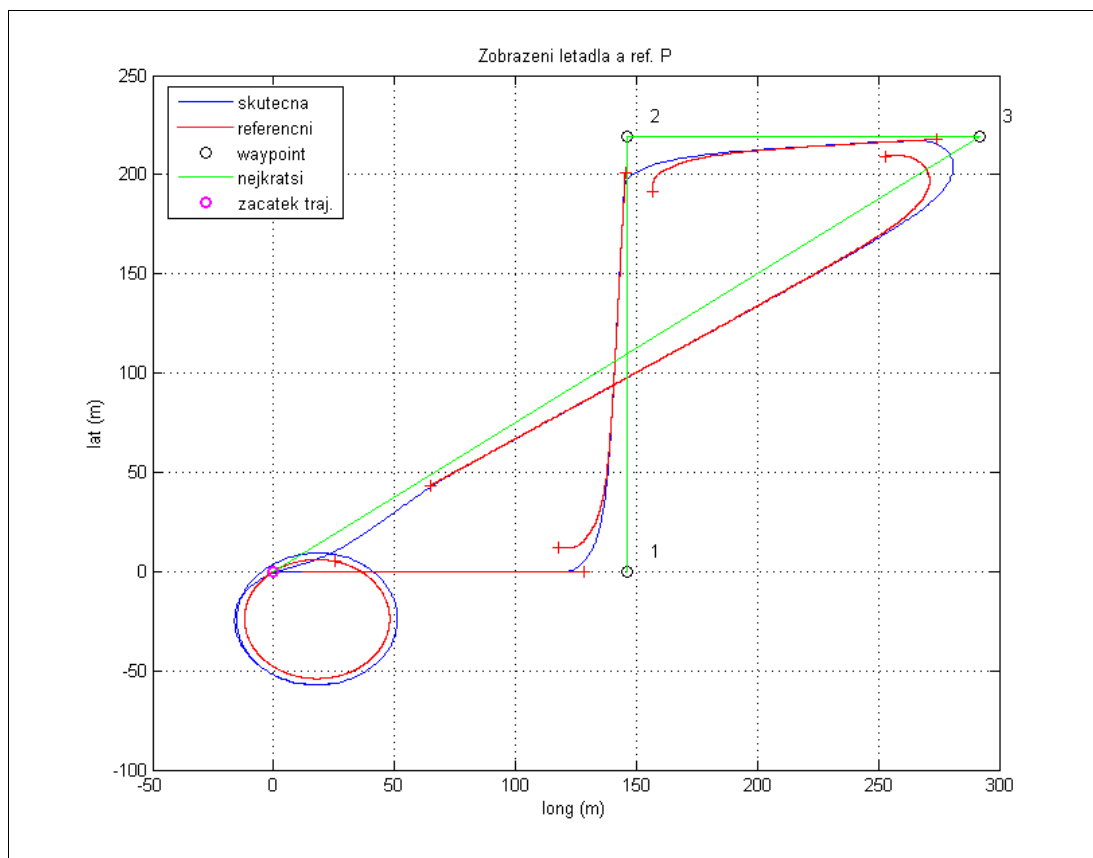
Tabulka 9: Přehled průměrných hodnot odchylky při změně threshold



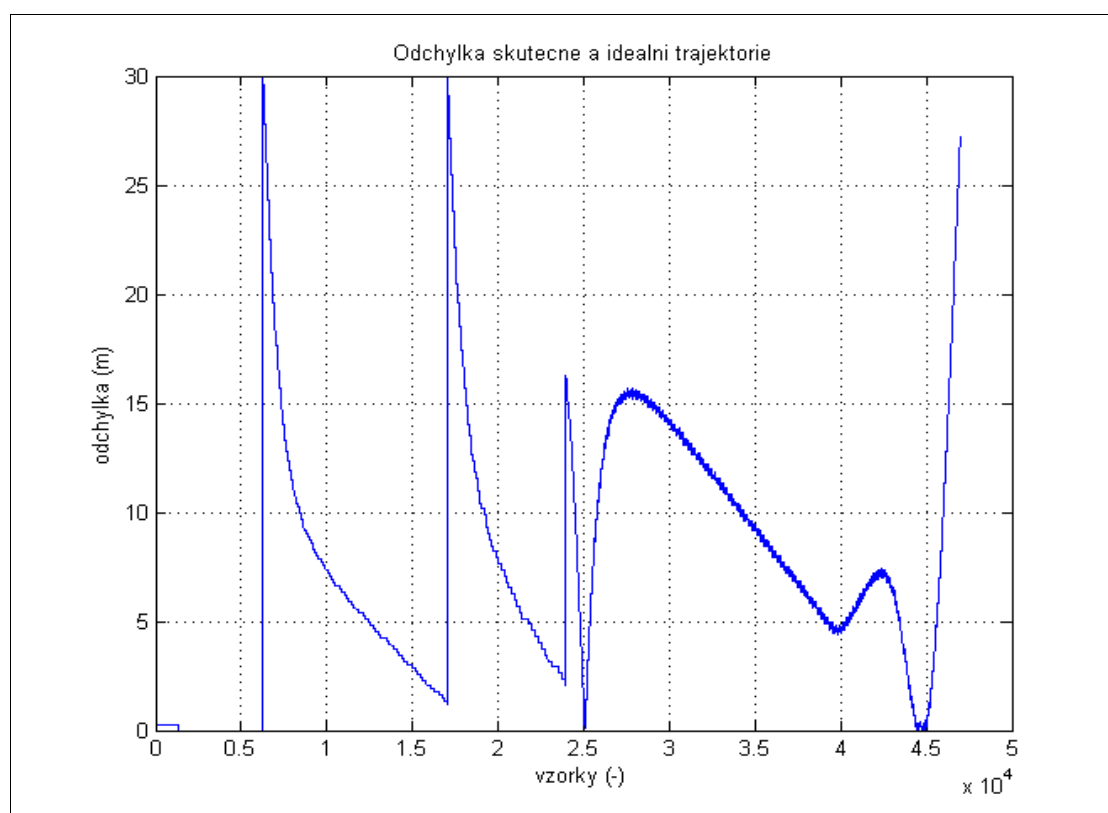
Ilustrace 56: Zobrazení reálné a ideální trajektorie ($T = 7m$)



Ilustrace 57: Průběh odchylky ($T = 7m$)



Ilustrace 58: Zobrazení reálné a ideální trajektorie ($T = 30m$)



Ilustrace 59: Průběh odchylky ($T = 30m$)

7.2 Výsledky hardware in the loop simulace

Využití simulace HIL („hardware in the loop“) usnadní řešení možných problémů s navigačním algoritmem a regulátory. HIL simulace odhalí špatný návrh bez případné fyzické destrukce letadla.

HIL simulace opět pracuje s matematickým modelem letadla, ale navigaci a regulaci řídí elektronika, která bude ve výsledku ovládat letadlo. Testujeme tedy, zda jednotka stíhá počítat tyto smyčky a reaguje správně na případné příkazy obsluhy.



Ilustrace 60: Zobrazení pozemní stanice s mapou a aktuálním stavem letadla (zdroj Bc. Vojtěch Kuchař)



Ilustrace 61: Zobrazení letadla v programu X-plain (zdroj Bc. Vojtěch Kuchař)

Zhodnocení

Simulace HIL ukázala, že letadlo je schopné podle navrženého algoritmu proletět zadanou trajektorii. Průběh trajektorie je značně kmitavý z důvodu nedostatečného odlazení regulátorů laterálního pohybu.

Reálný let

Reálný let letadla v autonomním modu se z důvodu špatného počasí nepodařilo otestovat do termínu odevzdání práce.

8 Závěr

Při návrhu algoritmu jsem postupoval od základních smyček výpočtu, které jsem postupně testoval s kombinací navržených regulátorů a výpočtového modelu letadla. Po zpracování těchto základů jsem se začal věnovat případům, kdy tyto smyčky nejsou schopny řešit nevhodně zadané body v seznamu. Upravený algoritmus je schopen tyto situace odhalit a zpracovat tento způsob plánování trajektorie.

Navržený algoritmus principiálně vychází z algoritmu popsáném v kapitole 2.4. Tento základ řeší pouze autonomní pohyb bez ohledu na možnost přepnutí do manuálního řízení. Proto musel být navržen jiný přístup pro výpočet referenčního bodu P. Původně se tento bod pohyboval po ideální trajektorii mezi body, od které letadlo nikdy nebylo příliš daleko. Při uvažování manuálního modu, kdy letadlo může zaletět daleko od ideální trajektorie se toto jevilo jako nevhodné řešení. Proto je v novém přístupu poloha bodu P počítána jako bod mezi letadlem a aktuálním bodem ze seznamu waypoints. Tímto je zajištěn pohyb letadla žádaným směrem.

Důležitou součástí autopilota je řešení chybových stavů nebo ukončení trajektorie. Tyto stavy jsou řešeny ukončením plánování na aktuální bod a letem na domácí stanici.

Při simulaci letových scénářů lze vidět, že algoritmus je schopen zpracovat plánování trajektorie ve všech řešených situacích s dostatečnou přesností. Je schopen na základě GPS dat rozpoznat průlet bodem a přepnout plánování na další bod v seznamu.

Při určení přesnosti průletu body se ukázalo, že touto metodou jsme schopni dosáhnout odchylky od ideální trajektorie v řádech jednotek metrů.

8.1 Možné další vylepšení autopilota

Optimalizací parametrů lze snížit odchylku od žádané trajektorie. Je možné algoritmus rozvinout v oblasti oblétání překážek a omezení letového prostoru. Toto vše lze realizovat s navrženým principem. Pokud bychom neomezovali regulátory saturací řídicích ploch, maximálním úhlem náklonu a sklonu, byli bychom schopni z tohoto letadla vytvořit akrobatické autonomní letadlo.

Přidáním senzoru schopného snímat povrch pod letadlem bychom mohli vytvořit řídicí systém tzv. Target tracking, kdy se letadlo pohybuje nad zvoleným cílem či objektem. Tyto procesy nejsou součástí základního autopilota, proto jsem se jimi ve své práci nezabýval.

Při plánování trajektorie dochází ke skokovým změnám při přepínání v seznamu waypoints. Tyto skoky by bylo možné odstranit přidáním omezení na změnu polohy bodu P. Principiálně tyto skoky nejsou problémovým prvkem pro správně navržené regulátory.

9 Použitá literatura

- [1] Derek R. Nelson, et al. Vector Field Path Following for Miniature Air Vehicles, *IEEE Trans. on Robotics*, vol. 23, no. 3, pp. 519-529, June 2007, DOI: 10.1109/TRO.2007.898976
dostupné z (duben 2014):
<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4252175&queryText=%3Dvector+field+path+following+for+miniature+air+vehicles>
- [2] Anna Bourmistrova and Sergey Khantsis (2010). Genetic Programming in Application to Flight Control System Design Optimisation, *New Achievements in Evolutionary Computation*, Peter Korosec (Ed.), ISBN: 978-953-307-053-7, InTech, DOI: 10.5772/8055.
dostupné z (duben 2014):
<http://www.intechopen.com/books/new-achievements-in-evolutionary-computation/genetic-programming-in-application-to-flight-control-system-design-optimisation>
- [3] Peter Ortner, Luigi del Re. Autopilot Design Comparison and Flight Experiments for a Small UAV, *Asian Control Conference, 2009, ASCC 2009. 7th*, pp. 314 – 319, E-ISBN: 978-89-956056-9-1
dostupné z (duben 2014):
<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5276192&queryText=%3Dautopilot+design+comparison+and+flight+experiments>
- [4] Reed S. Christiansen. Design of an Autopilot for Small Unmanned Aerial Vehicles, *Master of science thesis on Brigham Young University, Department of Electrical and Computer Engineering, 2004*,
dostupné z (duben 2014):
<http://www.uadrones.net/academia/research/acrobat/0408.pdf>
- [5] Guillaume J.J. Duncard, PhD. Fault-tolerant Flight Control and Guidance Systems, *Springer London*, ISBN: 978-1-84882-560-4, DOI: 10.1007/978-1-84882-561-1
- [6] Sanghyuk Park, John Deyst. A New Nonlinear Guidance Logic for Trajectory Tracking, *Massachusetts Institute of Technology, Cambridge, 2004*, AIAA 2004-4900
dostupné z (duben 2014):
http://my.publications.li/st/cutterm/Park04_GNC
- [7] Webové podklady České vysoké učení technické, Fakulta dopravní, obor Kartografie:
dostupné z (duben, 2014):
<http://www.fd.cvut.cz/departament/k611/PEDAGOG/files/webskriptum/kartografie/kartografie.html>
- [8] PŘÍLESKÝ, L. Simulační modelování bezpilotního letounu. *Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014*. Vedoucí diplomové práce doc. Ing. Jiří Krejsa, Ph.D.

- [9] KUCHAR, V. Návrh a implementace SW architektury bezpilotního letounu. *Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014.* Vedoucí diplomové práce Doc. Ing. Jiří Krejsa, Ph.D
- [10] KRAUS, D. Návrh mechanického a elektrického subsystému bezpilotního letounu. *Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014.* Vedoucí diplomové práce doc. Ing. Jiří Krejsa, Ph.D.

10 Seznam ilustrací a tabulek

Seznam ilustrací

Ilustrace 1: celkové jednoduché schema letadla s autopilotem (zdroj [2]).....	7
Ilustrace 2: zobrazení navrženého vektorového pole pro přímý let a pro kruhovou trajektorii (zdroj [1]).....	8
Ilustrace 3: grafické zobrazení jednotlivých vektorů rychlostí a úhlů pro výpočet (zdroj [1]).....	9
Ilustrace 4: grafické zobrazení pro výpočet pole kolem rovné trajektorie (zdroj [1]).....	9
Ilustrace 5: zobrazení polohy letadla vzhledem k obloukové trajektorii (zdroj [1]).....	10
Ilustrace 6: schema řízení metodou inverzní dynamiky (zdroj [3]).....	11
Ilustrace 7: pozice letadla a žádaného bodu (zdroj [3]).....	13
Ilustrace 8: zobrazení vstupů a výstupů řízení (zdroj [5]).....	15
Ilustrace 9: zobrazení principu výpočtu laterálního zrychlení (zdroj [5]).....	16
Ilustrace 10: letadlo při změně kurzu (zdroj [5]).....	17
Ilustrace 11: průběh konvergence k žádané trajektorii (zdroj [6]).....	18
Ilustrace 12: zobrazení situace pro výpočet žádané výšky letu (zdroj [5]).....	19
Ilustrace 13: zobrazení trajektorie mezi body (zdroj [5]).....	20
Ilustrace 14: řešení výběru typu plánování.....	28
Ilustrace 15: dle typu plánování výpočet P a podmínky pro přepnutí na další bod v seznamu.....	29
Ilustrace 16: výpočet úhlu náklonu (roll angle).....	29
Ilustrace 17: přepínání z manual do autonomního modu - řešení indexu "k".....	29
Ilustrace 18: Celkový algoritmus v prostředí Matlab/Simulink.....	31
Ilustrace 19: výpočet koeficientu přepočtu (zdroj [7]).....	32
Ilustrace 20: Simulace pravoúhlé zatačky.....	35
Ilustrace 21: Žádané a skutečné hodnoty úhlu náklonu a výšky letu.....	36
Ilustrace 22: Krouživé stoupání a klesání.....	37
Ilustrace 23: Žádané hodnoty z řízení a skutečné hodnoty letadla.....	37
Ilustrace 24: Zobrazení trajektorie při přepnutí z manuálního do automatického modu.....	38
Ilustrace 25: Žádané hodnoty a reálný stav letadla (manuální mod v čase 0 až 10s).....	39
Ilustrace 26: Zobrazení reakce na vybavení chyby (let k domácí stanici).....	40
Ilustrace 27: Žádané hodnoty a skutečné stavy letadla.....	40
Ilustrace 28: Změna L na polovinu původní hodnoty.....	41
Ilustrace 29: Původní hodnota L.....	42
Ilustrace 30: Změna L na dvojnásobek původní hodnoty.....	42
Ilustrace 31: Změna R na polovinu původní hodnoty.....	43
Ilustrace 32: Žádané hodnoty a reálné stavy letadla ($R = 15\text{m}$).....	43
Ilustrace 33: Původní hodnota R.....	44
Ilustrace 34: Žádané hodnoty a skutečné stavy letadla ($R = 30\text{m}$).....	44
Ilustrace 35: Změna R na dvojnásobek původní hodnoty.....	45
Ilustrace 36: Žádané hodnoty a skutečné stavy letadla ($R = 60\text{m}$).....	45
Ilustrace 37: Změna threshold na polovinu původní hodnoty.....	46
Ilustrace 38: Původní hodnota threshold.....	47
Ilustrace 39: Změna threshold na dvojnásobek původní hodnoty.....	47
Ilustrace 40: Zobrazení reálné a ideální trajektorie - klasické plánování letu.....	50
Ilustrace 41: Odchylka reálné a ideální trajektorie - klasické plánování letu.....	50
Ilustrace 42: zobrazení žádané a reálné trajektorie - plánování kroužení.....	51
Ilustrace 43: Průběh odchylky reálné trajektorie od žádané - krouživý let.....	52
Ilustrace 44: Zobrazení reálné a ideální trajektorie ($L = 6\text{m}$).....	53
Ilustrace 45: Průběh odchylky ($L = 6\text{m}$).....	53

Ilustrace 46: Zobrazení reálné a žádané trajektorie při krouživém letu ($L = 6\text{m}$).....	54
Ilustrace 47: Průběh odchylky pro celou kruhovou trajektorii ($L = 6\text{m}$).....	54
Ilustrace 48: Zobrazení reálné a ideální trajektorie ($L = 24\text{m}$).....	55
Ilustrace 49: Průběh odchylky ($L = 24\text{m}$).....	55
Ilustrace 50: Průběh reálné a žádané trajektorie při kroužení ($L = 24\text{m}$).....	56
Ilustrace 51: Průběh odchylky pro celou kruhovou trajektorii ($L = 24\text{m}$).....	56
Ilustrace 52: Zobrazení reálné a žádané trajektorie ($R = 15\text{m}$).....	57
Ilustrace 53: Průběh odchylky při kroužení ($R = 15\text{m}$).....	58
Ilustrace 54: Zobrazení reálné a žádané trajektorie ($R = 60\text{m}$).....	58
Ilustrace 55: Průběh odchylky ($R = 60\text{m}$).....	59
Ilustrace 56: Zobrazení reálné a ideální trajektorie ($T = 7\text{m}$).....	60
Ilustrace 57: Průběh odchylky ($T = 7\text{m}$).....	60
Ilustrace 58: Zobrazení reálné a ideální trajektorie ($T = 30\text{m}$).....	61
Ilustrace 59: Průběh odchylky ($T = 30\text{m}$).....	61
Ilustrace 60: Zobrazení pozemní stanice s mapou a aktuálním stavem letadla (zdroj Bc. Vojtěch Kuchař [9]).....	62
Ilustrace 61: Zobrazení letadla v programu X-plain (zdroj Bc. Vojtěch Kuchař [9]).....	62
Ilustrace 62: Zobrazení nejdůležitějších rozměrů v půdoryse (zdroj [8] Bc. Libor Příleský)...	72
Ilustrace 63: Zobrazení nejdůležitějších rozměrů v bokoryse (zdroj [8] Bc. Libor Příleský)...	73

Seznam tabulek

Tabulka 1: Způsob zadávání bodů.....	15
Tabulka 2: Přehled speciálních poloh bodů.....	22
Tabulka 3: Vstupní hodnoty do řídicí vrstvy.....	30
Tabulka 4: Popis funkčních bloků řídicí vrstvy.....	33
Tabulka 5: Výstupní hodnoty řídicí vrstvy.....	34
Tabulka 6: Počáteční hodnoty parametrů.....	41
Tabulka 7: Přehled průměrných hodnot odchylek při změně L	52
Tabulka 8: Přehled průměrných hodnot odchylek při změně parametru R	57
Tabulka 9: Přehled průměrných hodnot odchylky při změně threshold.....	59

11 Seznam značek a symbolů

Seznam je rozdělen dle kapitol, kde se značky vyskytují.

Všeobecné značky a symboly

Uvedené značky a symboly jsou shodné pro celý text.

RC		Označení pro „remote control“ (vysílač a přijímač signálu)
GPS		Označení pro „global position system“ (poloha na Zemi)
NED		Označení pro „North, East, Down“ (souřadnicový systém spojený se Zemí s nulovou hodnotou v domácí stanici)
home		Označení pro „home position“ (domácí stanice letadla)
m.n.m	m	Zkratka pro „metr nad mořem“ (výšková hodnota nad hladinou moře)
U		Označení aktuální polohy letadla
P		Označení referenčního bodu
WP		Označení waypointů
C		Označení polohy středu kružnice kruhové trajektorie
m	kg	Hmotnost letadla
g	m/s ²	Gravitační zrychlení Země
V_a	m/s	Vektor rychlosti letadla vzhledem k okolnímu prostředí (airspeed)
V_g	m/s	Vektor rychlosti letadla vzhledem v s.s. NED
i	-	Index osy
$[\phi, \theta, \psi]$	rad	Eulerovy úhly kolem os souřadného systému letadla (roll, pitch, yaw)
ρ	kg/m ³	Hustota okolního prostředí

Kapitola 2.1

$\dot{x}$	m/s	Rychlost letadla ve směru osy x
$\dot{y}$	m/s	Rychlost letadla ve směru osy y
ψ	rad	nastavený kurz letu bez započítání vlivu větru (heading)
W	m/s	Vektor rychlosti proudění vzduchu v s.s. NED
χ	rad	Aktuální azimuth letadla
χ^d	rad	Žádaná hodnota azimuthu
χ^∞	rad	Azimuth v nekonečné vzdálenosti od žádané trajektorie
y	m	Laterální vzdálenost od zvolené trajektorie
k	-	Koeficient přechodu od χ^∞ ke χ^d

d	m	Aktuální vzdálenost letadla od středu kruhové trajektorie
r	m	Poloměr kruhové trajektorie
γ	rad	úhlová pozice vzhledem ke spojnici bodu C a zemského severu

Kapitola 2.2

$[\chi_{\text{east}}, \chi_{\text{noth}}]$	[m,m]	Vektor aktuální pozice letadla v s.s. NED
$[\chi_{\text{eastD}}, \chi_{\text{northD}}]$	[m, m]	Vektor polohy žádaného bodu v s.s. NED
$[U, V, W]'$	m/s	Rychlosti v souřadném systému letadla
$[X_A, Y_A, Z_A]'$	N	Aerodynamické síly v souřadném systému letadla
$[X_T, Y_T, Z_T]'$	N	Tahové síly letadla (tah, vztlak)
$[P, Q, R]'$	rad/s	Úhlové rychlosti kolem os souřadného systému letadla
$[I_{XX}, I_{YY}, I_{ZZ}]'$	kg m ²	Momenty setrvačnosti letadla
$[l, m, n]'$	Nm	Aerodynamické momenty kolem os souřadného systému letadla
$[p_N, p_E, h]'$	m	Vektor určující polohu letadla vzhledem k Zemi
$[C_l, C_m, C_n]'$	-	Roll, pitch, yaw momentové koeficienty
$[C_D, C_Y, C_L]'$	-	Drag, side, lift silové koeficienty
$[\delta_a, \delta_e, \delta_r]$	rad	Výchyšky ovládacích ploch (a – ailerons, e – elevator, r – rudder)
S	m ²	Plocha křídel
b, c	m	Rozpětí křídel, Délka trupu
α, β	rad	Úhel náběhu (angle of attack), Úhel skluzu (backslide angle)

Kapitola 2.4.1

R	m	Žádaný poloměr letu
a_l	m/s ²	Vektor laterálního zrychlení
η	rad	Úhel mezi V_g a L_1
γ	rad	Úhel mezi směrem laterálního zrychlení a žádaným směrem letu
L_1	m	Vektor žádaného směru letu (UP)
Z_b	N	Vztlaková síla
F_l	N	Síla vyvozující laterální pohyb letadla
θ_a	rad	Úhel mezi vektorem laterálního zrychlení a kolmicí na L_1 v bodě C

Kapitola 2.4.2

d_1	m	Vzdálenost U od posledního WP v rovině NE
λ	rad	odchylka polohy letadla od ideální trajektorie mezi body v rovině NE
γ	rad	Úhel stoupání mezi body
h_c	m	Žádaná výška letu

Kapitola 2.4.3

$[X_E, X_N]$	[m,m]	Vektor aktuální pozice letadla v s.s. NED
ψ_{SEG}	rad	Směr trajektorie mezi body
H	m	Kolmý průmět U na ideální trajektorii
ΔP	m	Vzdálenost HP na ideální trajektorii
$[P_N, P_E]$	[m,m]	Poloha ref. bodu P v s.s. NED

Kapitola 2.4.4

d_2	m	Vzdálenost U od následujícího WP v rovině NE
-------	---	--

Kapitola 4.1

d_{2one}	m	Jednotkový vektor z vektoru d_2
L	m	Navržená vzdálenost ref. bodu P od U

Kapitola 4.2

γ_{UP}	rad	Úhel mezi body UP vzhledem k bodu C
ϕ_{CU}	rad	Směr vektoru UC vzhledem k zeměpisné šířce (sever)
CU	m	Vzdálenost U od C

Kapitola 5.1

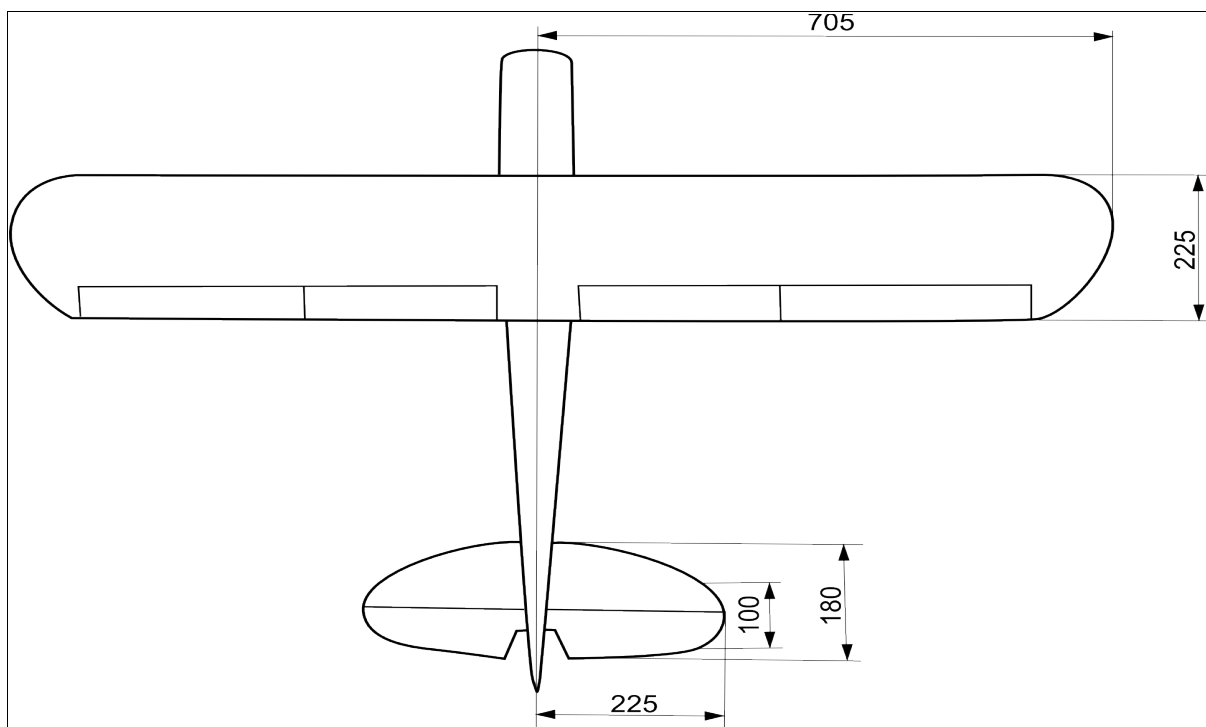
deg2m	-	Koeficient přepočtu GPS stupňů na vzdálenost od home
X_{ned}	m	vzdálenost od domácí stanice v NED (směr zeměpisné délky, východ)
Y_{ned}	m	vzdálenost od domácí stanice v s.s. NED (směr zeměpisné šířky, sever)

12 Přílohy

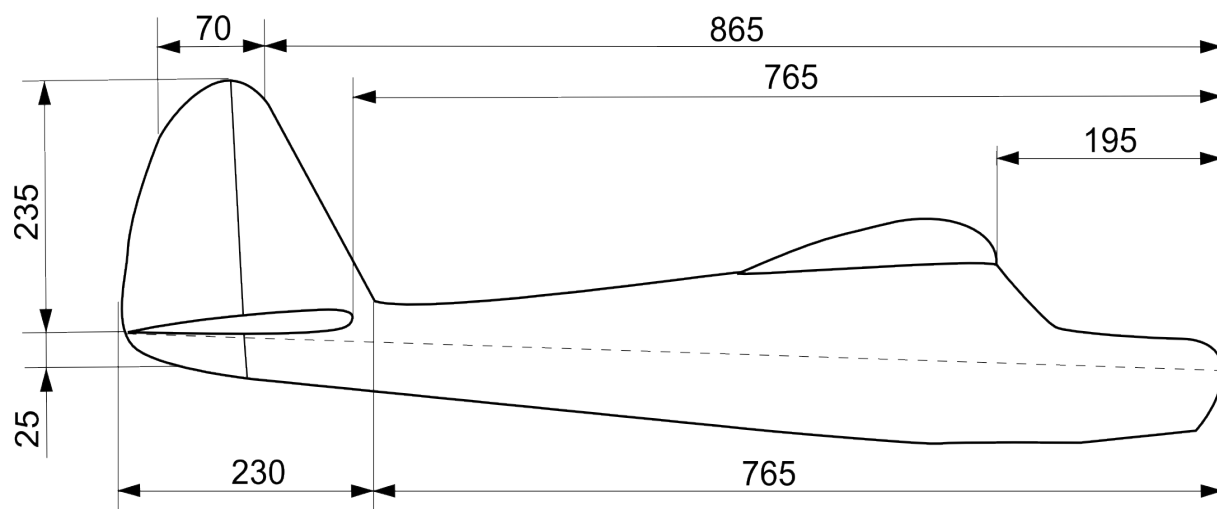
K diplomové práci jsou přiložené soubory:

Název	Popis
SkyDog_Nemecek_DP2014.zip	Archiv souborů obsahující veškeré dokumenty pro spuštění simulace v Matlab/Simulink
<u>Archiv obsahuje soubory:</u>	
SkyDog_Autopilot_model.mdl	Celkový model autopilota včetně bloku regulátorů a modelu letadla (součástí této práce je pouze blok skydog_path_planning)
KoeficientyVrtuleUpdated.mat	Parametry vrtule použité na letadle SkyDog
Skydog2_params.m	Parametry letadla
reg_konst.m	Parametry regulátorů
waypoints.m	Seznam bodů zájmu pro simulační prostředí
start.m	Skript pro načtení všech parametrů
UP_plot.m	Skript pro vykreslení grafu žádané a skutečné trajektorie a grafu žádaných hodnot vystupujících z autopilota a reálných stavů letadla
Spusteni.txt	Postup spuštění simulace
Nemecek_Jakub_DP_2014.pdf	PDF dokument s textem diplomové práce

12.1 Příloha A – geometrie letadla



Ilustrace 62: Zobrazení nejdůležitějších rozměrů v půdoryse (zdroj [8] Bc. Libor Přileský)



Ilustrace 63: Zobrazení nejdůležitějších rozměrů v bokoryse (zdroj [8] Bc. Libor Přileský)

12.2 Příloha B – celkové schema autonomního letadla

