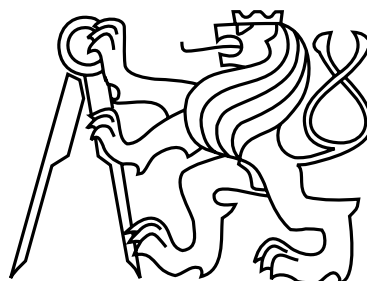


České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra měření



Diplomová práce

Identifikace rizikových situací v ultralehkém létání

Bc. Eva Pomíchalová

Vedoucí práce: Ing. Ondřej Bruna

Studijní program: Kybernetika a robotika, prezenční, Navazující magisterský

Obor: Letecké a kosmické systémy

12. května 2014



ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Eva Pomíchalová**

Studijní program: **Kybernetika a robotika**
Obor: **Letecké a kosmické systémy**

Název tématu česky: **Identifikace rizikových situací v ultralehkém létání**

Název tématu anglicky: **Indentification of Critical Situations in Ultra-Light Flying**

Pokyny pro vypracování:

Klasifikujte letové režimy, které mohou vést ke zničení letounu a mohou ohrozit pilota na životě. Vytvořte trénovací množinu simulovaných dat (například na simulátoru katedry měření), která budou obsahovat letové veličiny navržených situací. Definujte parametry pro jejich rozpoznání a pokuste se určit nejmenší množství parametrů nutných k jejich identifikaci. Pokuste se vytvořit algoritmus s cílem tyto situace identifikovat postprocessingem, popř. v reálném čase. K řešení lze použít neuronových sítí, rozpoznávání či jiného přístupu podle vlastního uvážení.

Seznam odborné literatury:

- [1] Saarlal, M.: *Aircraft performance*. Wiley 2007, Hoboken, ISBN 978-0-4700-4416-2.
- [2] Bc. Milan Hruška: *Parametrizace letu letadla na základě údajů avionického systému*. Bakalářská práce 2011, katedra počítačů FEL.
- [3] Russell, J.B.: *Performance and stability of aircraft*. Arnold, London 1996, ISBN 9780340631706.
- [4] Duda, Richard O., Hart, Peter E., Stork, David G.: *Pattern classification*. Wiley, New York 2001. ISBN 9780471056690.

Vedoucí bakalářské práce: Ing. Ondřej Bruna

Datum zadání bakalářské práce: 29. listopadu 2013

Platnost zadání do¹: 31. srpna 2015

Prof. Ing. Vladimír Haasz, CSc.
vedoucí katedry



Prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 29.11.2013

¹ Platnost zadání je omezena na dobu tří následujících semestrů.

Poděkování

Chtěla bych poděkovat předně svým rodičům za umožnění studia a poskytnutí prostředků potřebných pro jeho realizaci. Dále panu Ing. Ondřejovi Brunovi za odborné konzultace a celé vedení mé práce. Velký dík také patří panu Janu Kottovi za podporu při tvorbě algoritmů.

Prohlášení

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

V Praze dne 12. 5. 2014

.....

Abstrakt

Autorka: Bc. Eva Pomíchalová

Název diplomové práce: Identifikace rizikových situací v ultralehkém létání

Škola: České vysoké učení technické v Praze, Fakulta elektrotechnická

Rok vydání: 2014

Počet stran bez příloh: 70

Počet příloh: 4

Klíčová slova: ultralehké létání, rizikové manévry, rizikové situace, analýza dat, letecký simulátor, metody strojového učení, automatické rozpoznávání, klasifikace

Tato práce se zabývá rozborem rizikových situací v ultralehkém létání. Jedná se o klasifikaci těchto situací a vytvoření algoritmu, který se bude schopen naučit a následně identifikovat předem definované manévry. Algoritmus pracuje s údaji ze simulátoru, nicméně je zde i možnost přizpůsobení pro údaje z reálného letadla. Data jsou zpracovávána postprocesingem.

Abstract

Author: Bc. Eva Pomíchalová

Master's thesis title: Identification of Critical Situations in Ultra-Light Flying

University: CTU in Prague, Faculty of Electrotechnical Engineering

Year: 2014

Number of pages: 70

Number of appendices: 4

Key words: ultra-light flying, critical manoeuvres, critical situations, data analysis, flight simulator, machine learning methods, automatic pattern recognition, classification

This thesis is concerned with risk situations in ultra-light flying. It consists of a classification of such situations and creation of an algorithm that is able to learn and then identify some predefined manoeuvres. The algorithm works with data from a simulator but adjusting to the real data is possible. The data are post-processed.

Obsah

Seznam obrázků	xv
Seznam tabulek	xvii
1 Úvod	1
2 Související práce	3
2.1 Kognitivní zátěž pilotů	3
2.1.1 Stresové faktory v ultralehkém létání	3
2.1.2 Stres pilotů bojových letounů	3
2.2 Rozpoznávání letových situací na letovém simulátoru	5
2.2.1 Pravidla vs. pravděpodobnost	5
2.2.2 PCA analýza a neuronové sítě	8
2.3 Určování fází letu z radarových dat	10
2.4 Souhrn souvisejících prací	12
3 Vzorek letových dat a jeho analýza	13
3.1 Získání dat	13
3.2 Anotace dat	14
3.3 Předpoklady	17
3.4 Klasifikace nebezpečných situací	17
3.5 Výběr vhodných parametrů	18
4 Rule-based systém	21
4.1 Analýza zajímavých parametrů	21
4.2 Vybrané parametry	22
4.3 Algoritmus pro identifikaci vývrtky	22
4.4 Výsledné grafy	23
4.5 Možnosti rozšíření systému	25
4.6 Zhodnocení systému	25

5	Učící se systém	27
5.1	Strojové učení	27
5.1.1	Učení s učitelem (supervised learning)	28
5.2	Výběr validační metody	28
5.3	1. fáze: Zúžení výběru na tři metody	28
5.3.1	Výběr vhodných parametrů	28
5.3.2	Výběr tří nejlepších metod	30
5.4	2. fáze: Optimalizace tří nejlepších metod	35
5.4.1	Testování neuronových sítí	35
5.4.2	Testování diskriminační analýzy	36
5.4.3	Testování nejbližších sousedů	36
5.4.4	Zlepšení přesností po optimalizaci metod	37
5.5	3. fáze: Vyhlazení	38
5.6	Vyhodnocení výsledků	39
6	Aplikace nejlepší metody	45
6.1	Trénovací skript	45
6.2	Klasifikační skript	45
6.3	Možnosti rozšíření	46
7	Závěr	47
	Reference	49
A	Seznam použitých zkratk	53
B	Nácvik manévrů	55
C	Vzorek surových txt dat	61
D	Obsah příloženého CD	63

Seznam obrázků

2.1	Architektura ANN a MLR modelu [9]	4
2.2	Ukázka ověření funkce modelů [9]	5
2.3	Ukázka části stavového přechodového diagramu pro systém používající jednoduchá IF-THEN pravidla[8]	6
2.4	Pravděpodobnostní síť určující pravděpodobnost počátku situace [8]	7
2.5	Clusterování ve 2 PCA projekcích [7]	10
2.6	Architektura použité Elmanovy neuronové sítě [7]	11
2.7	Vizualizace fází letu určených z polohových dat [11]	11
3.1	Letecký simulátor Fakulty elektrotechnické [2]	14
3.2	Ukázka záznamu dat z jednoho letu	15
3.3	Grafické zobrazení anotací - vývrтка	15
3.4	Grafické zobrazení anotací - spirála	16
3.5	Grafické zobrazení anotací - ostrý pád	16
4.1	Indikace nebezpečí vývrtky na referenčních datech	23
4.2	Indikace nebezpečí vývrtky na běžném letu	24
4.3	Indikace nebezpečí vývrtky na referenčních datech zatáček	24
5.1	Zjednodušené schéma rozhodovacího stromu pro klasifikaci vybraných nebezpečných manévřů	32
5.2	Bayesova věta ve vztahu k Bayesovu klasifikátoru [12]	33
5.3	Blokové schéma pro výběr nejlepší metody a kombinace parametrů	33
5.4	Vývojový diagram výběru nejlepší metody a kombinace parametrů	34
5.5	Grafické zobrazení výsledku KNN na letu obsahujícím 2 vývrtky	40
5.6	Grafické zobrazení výsledku KNN na letu obsahujícím 1 spirálu	40
5.7	Grafické zobrazení výsledku KNN na letu obsahujícím 2 ostré pády	41
5.8	Grafické zobrazení výsledku DA na letu obsahujícím 2 vývrtky	41
5.9	Grafické zobrazení výsledku DA na letu obsahujícím 1 spirálu	42
5.10	Grafické zobrazení výsledku DA na letu obsahujícím 2 ostré pády	42

5.11 Grafické zobrazení výsledku NN na letu obsahujícím 2 vývrtky	43
5.12 Grafické zobrazení výsledku NN na letu obsahujícím 1 spirálu	43
5.13 Grafické zobrazení výsledku NN na letu obsahujícím 2 ostré pády	44
7.1 Grafický výstup testování systému na neanotovaném letu	48
D.1 Obsah přiloženého CD	63

Seznam tabulek

2.1	Let Cessnou po standardním okruhu - rozpoznávání situací pomocí IF-THEN systému [8]	8
2.2	Let Cessnou po standardním okruhu - rozpoznávání situací pomocí pravděpodobnostního systému [8]	8
3.1	Přehled parametrů a jejich kategorizace	19
5.1	Kombinace parametrů	29
5.2	Identifikace vývrtky	32
5.3	Identifikace jednotlivých manévrů	33
5.4	Výsledky testování modifikací NN	36
5.5	Výsledky testování modifikací DA	36
5.6	Výsledky testování modifikací KNN	37
5.7	Přesnost identifikace jednotlivých manévrů po optimalizaci metod	38
5.8	Hledání nejlepšího limitu pro vyhlazení výsledků	39
5.9	Přesnost identifikace jednotlivých manévrů po vyhlazení výsledku	39

Kapitola 1

Úvod

Tato práce se zabývá klasifikací letových režimů, které mohou vést ke zničení letounu a tím i ohrožení pilota, případně další posádky. V letecké dopravě, především v ultralehkém létání mohou být takové situace velice nebezpečné a pokud je pilot neidentifikuje včas, může to mít fatální následky. Z toho vyplývá, že první skupinou, která by se mohla zajímat o výsledky takové práce jsou sami piloti.

Piloti často nejsou totožní s majiteli letadel a stává se, že se ať už úmyslně nebo neúmyslně dostávají do zakázaných zón a nebezpečných situací. Další skupinou, kterou by tedy mohly výsledky této práce, nebo spíše jejich aplikace, zajímat jsou tedy majitelé ultralehkých letadel.

Tato letadla nebývají vybavena složitými monitorovacími systémy, které jsou mimo jiné finančně náročné. Proto v současné době vznikají různé aplikace do mobilních telefonů či tabletů s úkolem monitorovat pohyb ultralehkých letadel. Jednou z takových aplikací je například MobileTracer od firmy CS SOFT a.s. Tato a jí podobné aplikace ovšem umožňují pouze sledovat, kde se letadlo pohybuje, nevypovídají téměř nic o manévrech, které s ním pilot prováděl. A to je právě ta část, na kterou je zaměřena tato práce.

Hlavním cílem práce je algoritmus, který bude identifikovat rizikové situace postprocessingem. Prvním dílčím cílem je získat vzorek dat, který bude tvořit trénovací množinu, dále pak vybrat nebezpečné situace, které budou klasifikovány, určit parametry vhodné ke klasifikaci a najít vhodný způsob klasifikace.

Výstupem z této práce bude algoritmus, detailní popis jeho funkcí a anotované vzorky dat, se kterými algoritmus pracuje. Dále také vyhodnocení úspěšnosti různých přístupů k dané problematice.

V této práci bude nejprve provedena rešerše, kde budou popsány související práce vytvořené nejen na Českém vysokém učení technickém. Dále bude popsán úvod do dané problematiky a její specifikace (získání referenčních dat, klasifikace situací, výběr vhodných

parametrů). Následovat bude výčet možných přístupů a jejich popis, detailní popis samotné implementace a tvorby algoritmu a seznámení s výsledky práce. Na závěr budou jednotlivé výsledky porovnány, vybrána nejvhodnější metoda a shrnuta doporučení plynoucí z provedené analýzy výsledků.

Kapitola 2

Související práce

2.1 Kognitivní zátěž pilotů

Nebezpečné situace analyzované v této práci jistě patří k situacím se zvýšenou mírou stresu pro piloty, proto bylo vhodné zrevidovat několik prací týkajících se nejen rozpoznávání, ale také právě hodnocení míry stresu pilotů v různých situacích.

2.1.1 Stresové faktory v ultralehkém létání

Stresovým faktorům v extrémních situacích v ultralehkém létání se věnovala slečna Jíchová v loňském roce přímo na Fakultě elektrotechnické ČVUT. Její práce [10] se zabývá záznamy a analýzou biologických signálů pro určení míry stresu pilotů v různých fázích letu. Experiment probíhal na hydraulickém simulátoru s uzavřenou kabinou, který dnes již není k dispozici. Počet testovaných subjektů v této práci je podobný počtu subjektů, který byl k dispozici pro tuto práci. Fáze letů byly vyhodnocovány a parametrizovány jako úseky biosignálů.

Práce si rozděluje fáze letu na 4 části oddělené začátkem měření, výpadkem motoru, minutou po výpadku motoru, minutou před přistáním a samotným přistáním. Toto dělení letu ani následná parametrizace není pro účely této práce příliš užitečné.

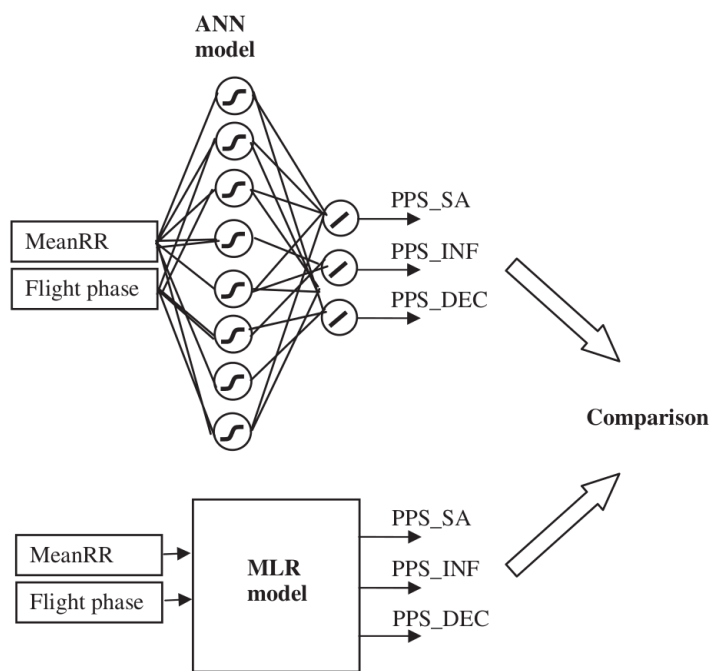
2.1.2 Stres pilotů bojových letounů

Zajímavému tématu se věnoval tým z Finska. Tento tým složený z odborníků nejen z univerzitní půdy provedl zajímavý experiment [9] s piloty bojových letounů F/A-18 Hornet. Experiment absolvovalo 14 subjektů, kterým bylo puštěno stejné cvičení na simulátoru a zaznamenáno jejich EKG¹. Pomocí záznamu o změnách v srdeční aktivitě byla stanovena

¹Elektrokardiogram - záznam časové změny elektrického potenciálu způsobené srdeční aktivitou. [17]

míra kognitivní zátěže v jednotlivých fázích letu. Těchto fází bylo celkem 13, převážně různé bojové fáze neaplikovatelné pro ultralehké létání.

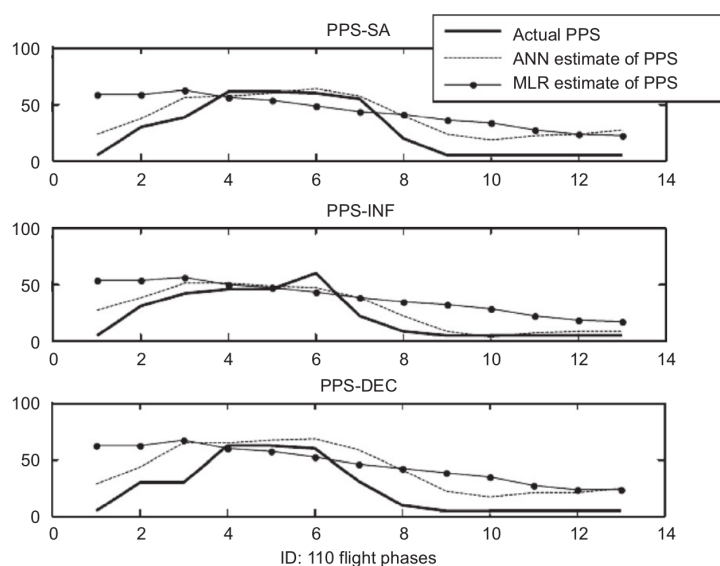
Tým poté provedl detailní analýzu naměřených hodnot ve vztahu k jednotlivým fázím letu a sestavil umělou neuronovou síť (ANN) a multilineární regresní model (MLR) s cílem predikovat kognitivní zátěž v každé fázi. Architektura těchto modelů je vidět na obrázku 2.1. Vstupy tvoří střední hodnota zaznamenané doby mezi dvěma údery srdce (MeanRR) a fáze letu (Flight phase). Vyhodnocení míry psychofyziologického stresu (PPS) je provedeno pomocí tří parametrů a to: povědomí o situaci (PPS_SA), informační zátěž (PPS_INF) a rozhodování (PPS_DEC). Poté jsou výsledky obou modelů porovnávány (Comparison).



Obrázek 2.1: Architektura ANN a MLR modelu [9]

Neuronové sítě v této aplikaci poskytovaly o 13-23% lepší výsledky než multilineární regresní model. Detailní popis měření stresu, jednotlivých fází letu, modelů i výsledků je k dispozici v jejich studii [9]. Ukázka výsledného grafu je uvedena na obrázku 2.2. Jednotlivé fáze letu jsou zobrazeny na ose x. Osa y označuje míru psychofyziologického stresu v rozsahu 0-100. V grafech je zobrazena skutečná (naměřená) míra stresu (Actual PPS), odhadovaná míra stresu pomocí neuronové sítě (ANN estimate of PPS) a odhadovaná míra stresu pomocí multilineárního regresního modelu (MLR estimate of PPS).

Měřených subjektů nebylo velké množství. Neuronové sítě jsou zajisté zajímavým nástrojem pro rozpoznávání a v této práci budou také použity viz. kapitola 5.



Obrázek 2.2: Ukázka ověření funkce modelů [9]

2.2 Rozpoznávání letových situací na letovém simulátoru

Rozpoznáváním letových situací na datech ze simulátorů se zabývali na Delft University of Technology. Z jejich tvorby jsou z hlediska této diplomové práce zajímavé dva články zmíněné níže. Na této univerzitě byl takový rozpoznávač součástí projektu ICE², který měl sloužit k usnadnění práce pilotů zejména ve stresových situacích.

2.2.1 Pravidla vs. pravděpodobnost

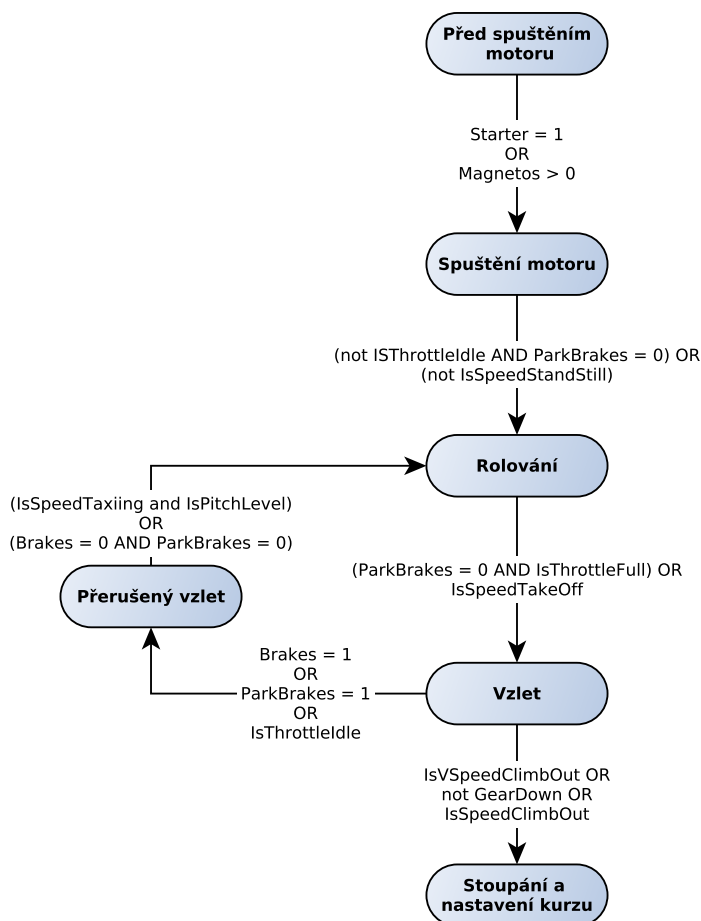
Jejich práce se zaměřuje na rozpoznání několika situací pomocí různých metod. Článek s názvem A Rule-Based and a Probabilistic System for Situation Recognition in a Flight Simulator [8] se zabývá tzv. real time rozpoznáváním letových situací na simulátoru. Tým se zaměřoval na rozpoznávání situací, které nejsou nebezpečné, ale vyjadřují určité fáze letu. Rozpoznávanými situacemi byly:

- Nastartování motoru
- Pojždění na vzletovou dráhu
- Vzlet
- Normální let

²Intelligent Cockpit Environment

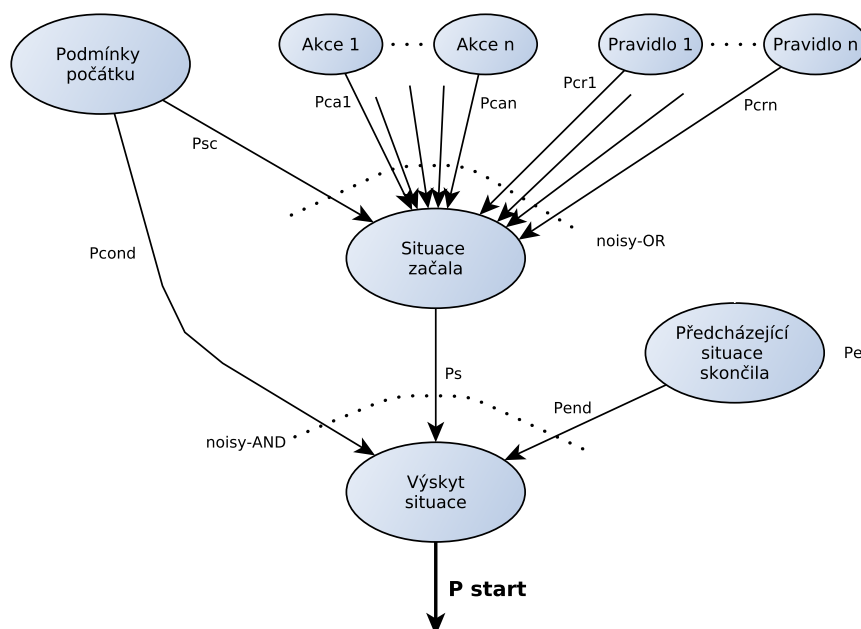
- Přistání
- Pojízdní od přistávací dráhy
- Vypnutí motoru

K rozpoznání těchto situací byly vytvořeny dva různé systémy s odlišnými přístupy. První systém vycházel z heuristických pravidel založených na přechodovém stavovém diagramu. Tento systém rozhoduje o současné situaci na základě hodnot parametrů jako jsou vzdušná rychlost, vertikální rychlost, tah motoru, status brzd, status podvozku atd. Používá jednoduchá IF-THEN pravidla. Jeho chování nejlépe vystihuje obrázek 2.3. Systém byl implementován v jazyce Borland Delphi 5 a testován na charakteristikách letu strojem Cessna 172C Skyhawk na simulátoru Flightgear, verze 0.7.10.



Obrázek 2.3: Ukázka části stavového přechodového diagramu pro systém používající jednoduchá IF-THEN pravidla[8]

Jednou z nevýhod systému, který se rozhoduje na základě předem daných pravidel IF-THEN je, že je vždy deterministický. Podmínka buďto je a nebo není splněná. Druhý systém oproti tomu pracuje s mírou pravděpodobnosti určení začátku a konce situací. Přejchodový stavový diagram již není potřebný, ale jsou přidány určité předpoklady, za kterých může daná situace nastat. Jsou zde stále daná pravidla, ale rozhodovací mechanismus pracuje s pravděpodobnostmi výskytu namísto přímých deterministických rozhodnutí. Systém byl implementován v jazyce Java a testován na Microsoft Flight Simulátoru 2002, stroj Cessna C172 a F-16. Ukázka pravděpodobnostní sítě, která vypočítává pravděpodobnost, že situace započala je na obrázku 2.4.



Obrázek 2.4: Pravděpodobnostní síť určující pravděpodobnost počátku situace [8]

Výsledky těchto systémů jsou velice podobné, takže autoři nemohli jednoznačně určit, který přístup je lepší. Jejich výsledky jsou k nahlédnutí v tabulce 2.1 a 2.2. Zde je nejdůležitějším faktorem rychlost rozpoznání začátku situace, protože se jedná o real-time systém. Zatímco tato práce se zabývá následným zpracováním dat, tzv. postprocessingem, kde rychlost rozpoznání začátku situace není tím nejdůležitějším faktorem.

Tabulka 2.1: Let Cessnou po standardním okruhu - rozpoznávání situací pomocí IF-THEN systému [8]

Situace	Čas počátku (s)	Čas detekce (s)
Nastartování motoru	0	0
Rolování na RWY ³	16	17
Vzlet	27	29
Normální let	73	72
Přistání	187	179
Rolování z RWY	272	275
Vypnutí motoru	302	302

Error: 15 seconds (5.0%)

Tabulka 2.2: Let Cessnou po standardním okruhu - rozpoznávání situací pomocí pravděpodobnostního systému [8]

Situace	Čas počátku (s)	Čas detekce (s)
Nastartování motoru	0	0
Rolování na RWY	7	10
Vzlet	22	27
Normální let	61	59
Přistání	119	121
Rolování z RWY	220	221
Vypnutí motoru	251	251

Error: 13 seconds (5.2%)

První zmíněný přístup je přístupem nejčastěji používaným. Podobným způsobem je prováděna analýza letů také ve společnosti ČSA a velice podobný způsob bude mimo jiné uplatněn i pro tuto práci v kapitole 4.

2.2.2 PCA analýza a neuronové sítě

Tento článek [7] předcházela výše zmiňované práci [8]. Autoři se zde zaměřují na rozpoznávání situací a predikci dalších vstupů pomocí Elmannovy neuronové sítě. Tentokrát

³Runway neboli vzletová a přistávací dráha

pracují se vstupy pouze ze simulátoru FlightGear. Více informací o simulátoru FlightGear je k nalezení přímo na webových stránkách simulátoru [6].

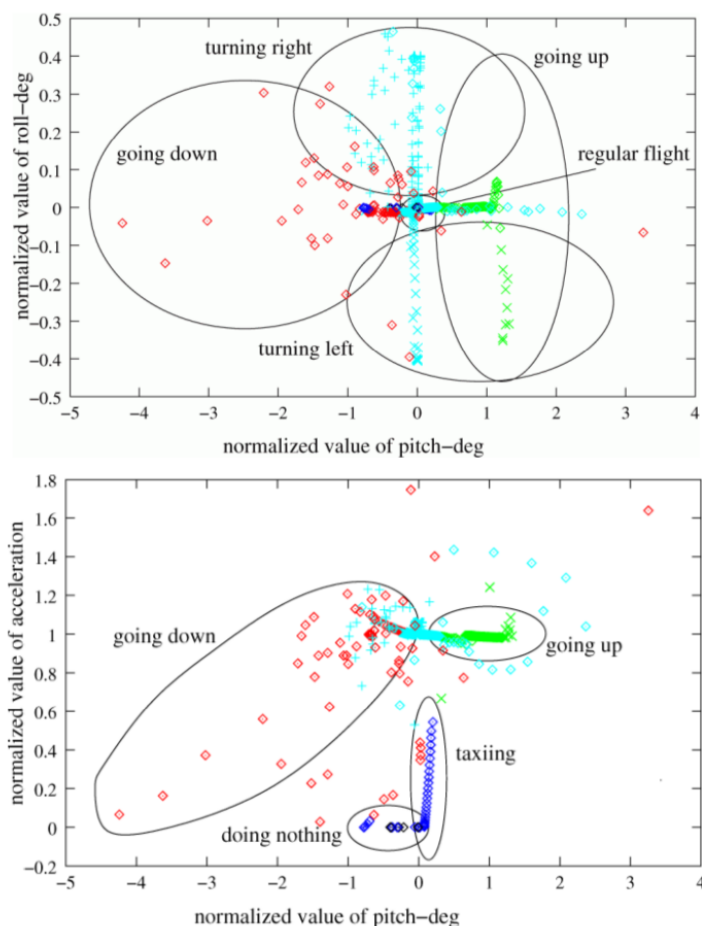
V první části provádí explorační analýzu dat. Během této analýzy pracují se čtyřmi proměnnými: klopení, tah motoru, zrychlení a klonění. Pomocí těchto proměnných v čase identifikovali 6 situací: rolování (*taxiing*), stoupání (*going up*), přímý let (*regular flight*), zatáčení vlevo (*turning left*), zatáčení vpravo (*turning right*), klesání (*going down*).

Poté se pokusili získat automatickou identifikaci lehce modifikovaných situací pomocí PCA⁴ analýzy. Předcházejících 6 situací bylo rozšířeno o další, a to stání na zemi (*doing nothing*). S pomocí PCA a Sammonova mapování bylo možné vytvořit clustery dat dle definovaných situací, jak lze vidět na obrázku 2.5. Osa x prvního grafu i druhého grafu znázorňuje normalizované hodnoty klopení ve stupních. Osa y prvního grafu znázorňuje normalizované hodnoty klonění ve stupních, zatímco osa y druhého grafu znázorňuje normalizované hodnoty zrychlení. Z obrázku vyplývá, že vytvořit clustery v prostoru z logovaných dat by mělo být možné. Následně se týmu podařilo namapovat clustery na celý let a tudíž se jim podařilo získat automatickou identifikaci definovaných situací.

Podobného výsledku bylo dosaženo také pomocí rekurzivních neuronových sítí, konkrétně Elmanovou neuronovou sítí s jednou skrytou vrstvou. Architektura použité sítě je vidět na obrázku 2.6. Testovací vstup byl stejný jako u PCA a výstupem bylo výše zmíněných 7 situací. Neuronovou sít se podařilo natrénovat pro rozpoznávání situací s chybovostí 13,5% na jednom setu testovacích dat. Neuronové sítě byly použity také pro predikci budoucích hodnot logovaných parametrů. Více informací o těchto testech je k dispozici v technické zprávě [3].

Pro modul rozpoznávání situací byl nakonec vybrán knowledge based systém popisovaný v části 2.2.1 a to především pro transparentnost jeho rozhodování. Autoři článku dále popisují tvorbu prototypu rozpoznávače a jeho rozšíření pro bojové letouny konkrétně F16. Zajímavé je, že se autoři pro každý stav snaží využít co nejvíce dostupných proměnných (parametrů), což jim umožňuje indikovat situaci i v případě, že jeden z parametrů vykazuje v dané situaci abnormální hodnoty. V ultralehkém létání jsou oproti tomu požadavky na použití co nejmenšího počtu parametrů a to ještě takových, které se dají jednoduše získat například z tzv. chytrého mobilního telefonu, kterým dnes disponuje většina pilotů i vlastníků letadel či aeroklubů.

⁴Analýza hlavních komponent - často se používá ke snížení dimenze dat s co nejmenší ztrátou informace [24].



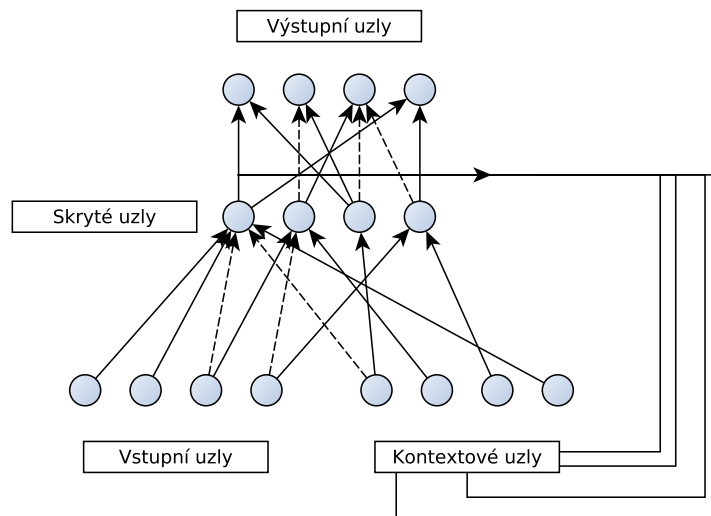
Obrázek 2.5: Clusterování ve 2 PCA projekcích [7]

2.3 Určování fází letu z radarových dat

Na Fakultě dopravní vznikla letos mimo jiné práce, jejímž hlavním cílem bylo vytvořit nástroj, který by na základě polohové informace sestavil scénář letu [11]. Jednou z částí této práce bylo navržení postupu a algoritmů pro detekci základních manévrů (přímý let, zatáčení, let v hladině, stoupání, klesání, start, přistání, pojíždění atd.). Algoritmy byly vytvořeny v programu Matlab.

Autor neuvádí procento úspěšnosti detekce manévrů ani vůči čemu byla úspěšnost ověřena, nicméně z jeho ukázek (viz. obrázek 2.7) lze soudit, že algoritmus detekuje fáze letu poměrně přesně. Autor vychází pouze z polohových údajů, tedy z radarových údajů či údajů z GPS ⁵. Tím se mimo jiné jeho práce liší od této, která pracuje přímo s letovými údaji. Nicméně pro

⁵Global Positioning System



Obrázek 2.6: Architektura použité Elmanovy neuronové sítě [7]

budoucí využití se může počet parametrů přiblížit pouze údajům z GPS přístroje. Autor detekuje pouze základní "normální" manévry nikoli nebezpečné manévry typu vývrтка, spirála apod.



Obrázek 2.7: Vizualizace fází letu určených z polohových dat [11]

2.4 Souhrn souvisejících prací

Všechny zmíněné práce se zabývají zajímavou problematikou. Tzv. Rule based systém bude zajisté dobrým začátkem pro identifikaci rizikových situací ultralehkého létání. Dále bude vhodné vyzkoušet neuronové sítě a další metody automatického rozpoznávání. Nejprve je ale nutné získat a analyzovat vzorek dat.

Kapitola 3

Vzorek letových dat a jeho analýza

Aby bylo možné identifikaci nebezpečných situací provést, bylo nejprve nutné získat referenční data. Tato data byla získána na Leteckém simulátoru Fakulty elektrotechnické ČVUT v Praze viz. obrázek 3.1.

Hlavní náplní této práce je tvorba algoritmu pro identifikaci nebezpečných situací. Jaké situace to ale jsou, na jakých datech budou identifikovány a které k tomu budou použité parametry? Těmito otázkami se zabývá právě tato kapitola, jež obsahuje popis způsobu získání dat a jejich následnou analýzu.

3.1 Získání dat

Jak již bylo zmíněno, data byla získána prostřednictvím Leteckého simulátoru Fakulty elektrotechnické ČVUT v Praze. Tato platforma používá letecký simulátor FlightGear. Platforma má několik stupňů volnosti. Svým pohybem simuluje zrychlení ve 3 osách.

Jako nebezpečné manévry byly po diskuzích s odborníky určeny vývrtka, spirála, ostrý pád, vyplavání, odskok, mírný pád, skluz a výkluz.

Pro získání referenčních dat byly vybráni čtyři piloti, kteří se snažili na simulátoru provést výše zmíněné manévry. Dále je k dispozici testovací sada dat od pátého pilota. Ta obsahuje různé lety s nebezpečnými manévry i bez nich.

Ukládaná data ze simulátoru byla vzorkována frekvencí 50Hz. Data byla z původního binárního formátu, který je výstupním formátem simulátoru, převedena do textového formátu a následně nahrána pomocí předem připraveného skriptu do Matlabu.

Záznam dat v Matlabu potom vypadá jako tabulka jejíž sloupce jsou pojmenovány podle veličiny, kterou obsahují, zatímco řádky označují jednotlivé logy. Jednotlivé logy (řádky) jsou označeny indexem 1-X podle toho, jak dlouhý záznam letu je. Ukázka dat nahraných v Matlabu je k dispozici na obrázku 3.2.

3.2 Anotace dat

Anotace byly provedeny autorem práce pomocí vizualizace letěné trajektorie a vyhledání zlomových bodů. Během anotací se ukázalo, že některá data není možné anotovat, neboť v nich nelze dané situace rozpoznat. Technicky anotace spočívají v tom, že ke každému záznamu letu je přiřazena množina indexů (řádků), které popisují dle anotátora nebezpečný manévr. Příklad anotace vývrtek vypadá následovně:

```
test_11_11_vyvrtka: 4708:5450
test_11_11_vyvrtka2: 1107:2218
test_11_13_vyvrtka: 265:2303
test_11_13_vyvrtka2: 741:2679
test_11_18_vyvrtka: 1322:2220 3858:5007 7321:7955
test_11_8_piper_vyvrtka3: 1671:2564
test_11_8_piper_vyvrtka4: 6148:7114 9019:10290
test_11_8_vyvrtka1: není
test_11_8_vyvrtka2: 1334:2174 4893:5848
```

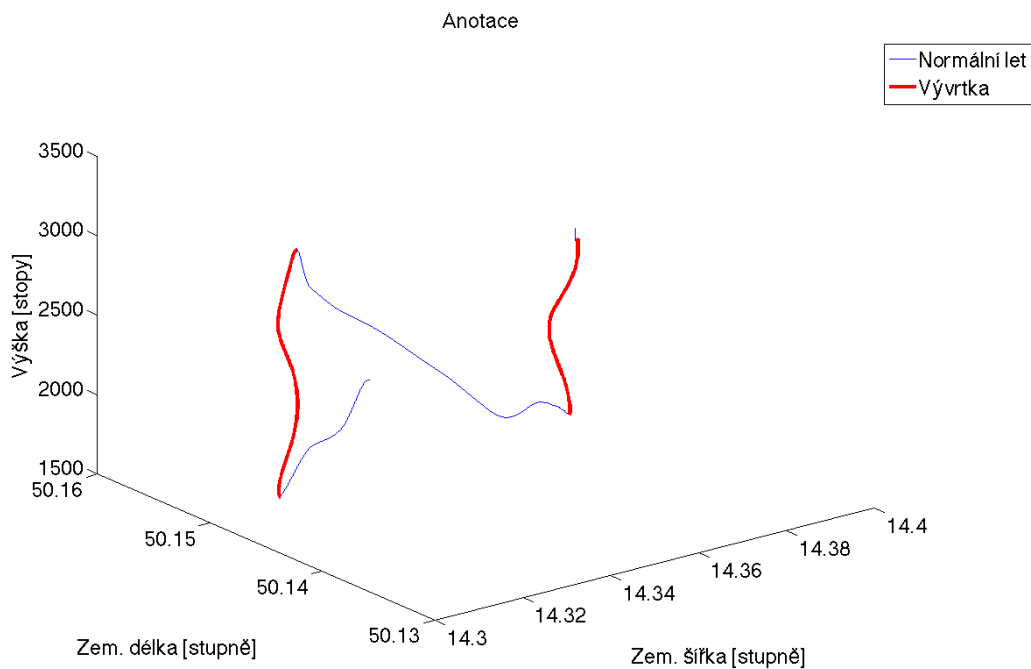
Vzorky grafického zobrazení anotací jsou k dispozici na obrázcích [3.3](#), [3.4](#) a [3.5](#)



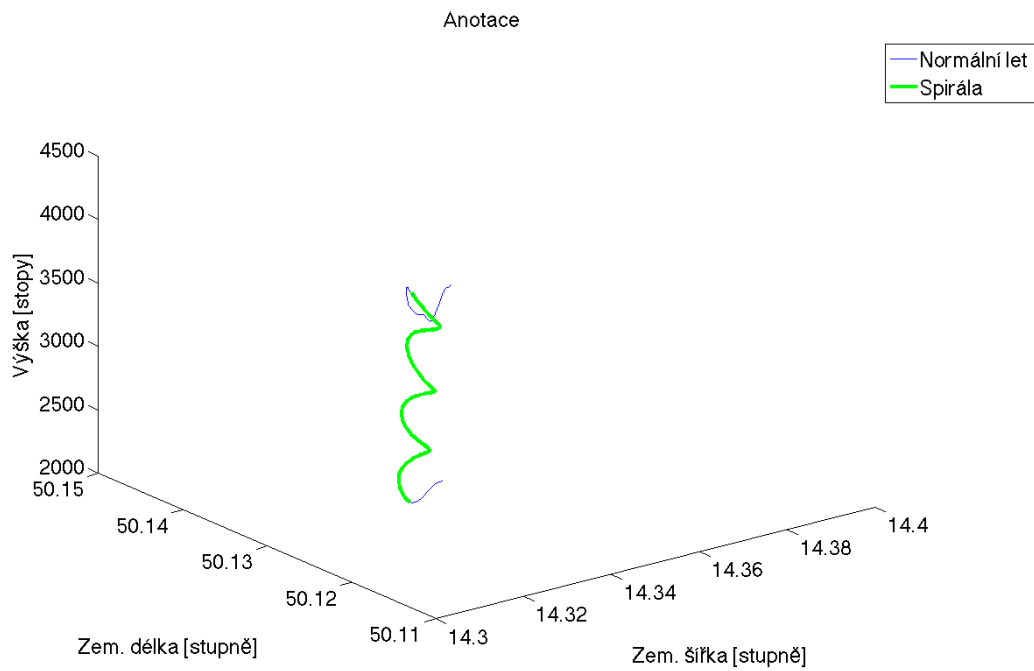
Obrázek 3.1: Letecký simulátor Fakulty elektrotechnické [2]

	1	2	3	4	5	6	7	8	9	10
	Latitude_deg	Longitude_deg	Altitude_ft	Roll_deg	Pitch_deg	Speed_down	Speed_east	Speed_north	vBody_fps	Speed_horizontal
1527	50.1385	14.3563	4.3317e+03	-49.1946	-57.3739	169.9199	-105.8266	-26.3208	21.1885	109.0507
1528	50.1385	14.3562	4.3261e+03	-50.3324	-57.2673	169.6475	-105.8320	-29.7929	20.9917	109.9456
1529	50.1385	14.3562	4.3232e+03	-50.9061	-57.2197	169.5195	-105.7719	-31.5370	20.8841	110.3734
1530	50.1385	14.3562	4.3190e+03	-51.7723	-57.1553	169.3383	-105.6026	-34.1606	20.7118	110.9903
1531	50.1385	14.3562	4.3176e+03	-52.0625	-57.1356	169.2809	-105.5249	-35.0365	20.6515	111.1893
1532	50.1385	14.3562	4.3148e+03	-52.6448	-57.0992	169.1710	-105.3371	-36.7897	20.5260	111.5769
1533	50.1385	14.3562	4.3091e+03	-53.8172	-57.0370	168.9702	-104.8345	-40.2979	20.2591	112.3129
1534	50.1385	14.3562	4.3063e+03	-54.4067	-57.0114	168.8799	-104.5190	-42.0508	20.1179	112.6610
1535	50.1385	14.3562	4.2922e+03	-57.3805	-56.9377	168.5374	-102.2990	-50.7602	19.3423	114.2002
1536	50.1385	14.3562	4.2894e+03	-57.9792	-56.9337	168.4921	-101.7268	-52.4836	19.1748	114.4678
1537	50.1385	14.3561	4.2852e+03	-58.8788	-56.9345	168.4393	-100.7888	-55.0522	18.9168	114.8439
1538	50.1385	14.3561	4.2838e+03	-59.1790	-56.9365	168.4258	-100.4550	-55.9035	18.8292	114.9626
1539	50.1385	14.3561	4.2810e+03	-59.7798	-56.9433	168.4053	-99.7557	-57.5978	18.6516	115.1899
1540	50.1385	14.3561	4.2768e+03	-60.6816	-56.9602	168.3906	-98.6282	-60.1169	18.3800	115.5057
1541	50.1385	14.3561	4.2726e+03	-61.5835	-56.9852	168.3956	-97.4071	-62.6058	18.1030	115.7913
1542	50.1385	14.3561	4.2683e+03	-62.4850	-57.0183	168.4207	-96.0934	-65.0611	17.8213	116.0469
1543	50.1385	14.3561	4.2669e+03	-62.7853	-57.0311	168.4336	-95.6351	-65.8714	17.7266	116.1255
1544	50.1385	14.3561	4.2641e+03	-63.3854	-57.0595	168.4664	-94.6882	-67.4792	17.5360	116.2725
1545	50.1385	14.3561	4.2599e+03	-64.2842	-57.1089	168.5330	-93.1926	-69.8569	17.2480	116.4682
1546	50.1385	14.3561	4.2557e+03	-65.1809	-57.1665	168.6208	-91.6079	-72.1905	16.9583	116.6340
1547	50.1385	14.3561	4.2515e+03	-66.0749	-57.2325	168.7301	-89.9358	-74.4768	16.6678	116.7700
1548	50.1384	14.3561	4.2486e+03	-66.6691	-57.2811	168.8151	-88.7731	-75.9730	16.4741	116.8442
1549	50.1384	14.3560	4.2472e+03	-66.9656	-57.3068	168.8613	-88.1776	-76.7124	16.3774	116.8764
1550	50.1384	14.3560	4.2430e+03	-67.8525	-57.3896	169.0143	-86.3352	-78.8941	16.0883	116.9532
1551	50.1384	14.3560	4.2388e+03	-68.7351	-57.4809	169.1896	-84.4102	-81.0187	15.8012	117.0005
1552	50.1384	14.3560	4.2345e+03	-69.6130	-57.5808	169.3872	-82.4047	-83.0829	15.5172	117.0184
1553	50.1384	14.3560	4.2317e+03	-70.1954	-57.6522	169.5314	-81.0240	-84.4240	15.3300	117.0141
1554	50.1384	14.3560	4.2303e+03	-70.4857	-57.6893	169.6073	-80.3207	-85.0838	15.2372	117.0071
1555	50.1384	14.3560	4.2260e+03	-71.3527	-57.8067	169.8499	-78.1603	-87.0182	14.9621	116.9666
1556	50.1384	14.3560	4.2218e+03	-72.2136	-57.9328	170.1152	-75.9257	-88.8832	14.6929	116.8971
1557	50.1384	14.3560	4.2175e+03	-73.0681	-58.0680	170.4030	-73.6193	-90.6760	14.4304	116.7987
1558	50.1384	14.3560	4.2118e+03	-74.1965	-58.2622	170.8221	-70.4367	-92.9490	14.0924	116.6226
1559	50.1384	14.3560	4.2090e+03	-74.7558	-58.3654	171.0467	-68.8011	-94.0333	13.9291	116.5155
1560	50.1384	14.3560	4.2047e+03	-75.5884	-58.5278	171.4023	-66.2944	-95.5926	13.6919	116.3309

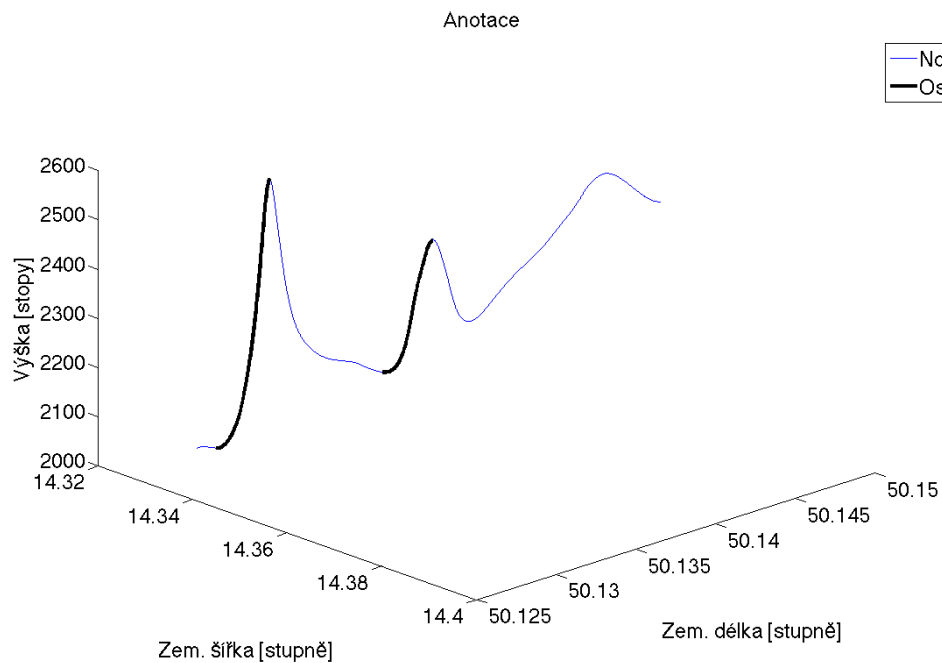
Obrázek 3.2: Ukázka záznamu dat z jednoho letu



Obrázek 3.3: Grafické zobrazení anotací - vývrтка



Obrázek 3.4: Grafické zobrazení anotací - spirála



Obrázek 3.5: Grafické zobrazení anotací - ostrý pád

3.3 Předpoklady

Před začátkem samotné práce s daty, popisu algoritmu a vyhodnocení výsledků je vhodné zmínit předpoklady, za kterých dané výsledky platí.

1. Jednotlivé manévry jsou v získaných datech správně anotované.
2. Piloti porozuměli manévřům, které měli zaletět na simulátoru jako referenční.

3.4 Klasifikace nebezpečných situací

V první řadě je důležité provést výběr a definici nebezpečných situací, na které se tato práce bude zaměřovat. V datech, které se podařilo během této práce získat byly k dispozici následující manévry po anotaci v uvedených četnostech.

- Vývrtka: 9x (8 anotovaných)
- Spirála: 4x (3 anotované)
- Ostrý pád: 3x
- Vyplavání: 4x
- Odskok: 4x
- Mírný pád: 3x
- Skluz: 3x
- Výkluz: 2x

Identifikace manévřů pomocí Rule-based systému popsaného v kapitole 4 byla provedena pouze pro vývrtku neboť je k dispozici nejvíce vzorků a z daných dat je společně se spirálou nejlépe identifikovatelná.

Pro práci s učícími se rozpoznávacími systémy byly vybrány vzhledem k omezenému počtu dat 3 manévry a to:

- Vývrtka
- Spirála
- Ostrý pád

Uvedené manévry byly vybrány, protože jsou většinou dobře rozeznatelné od normálního letu a tudíž

1. Jednodušší pro anotaci učitelem.
2. Snáze naučitelné pro algoritmus.

Vývrтка je v akrobatickém létání definována jako sestupný let po strmé šroubovici [14]. V ultralehkém létání patří mezi nebezpečné situace, je způsobena ztrátou vztlaku na jednom křídle. Vývrтка je považována v ultralehkém létání za ustálený režim letu, který by měl pilot být schopný vybrat.

Pro definici spirály lze vycházet z popisu takové situace pro kluzáky popsané na webových stránkách Aeroweb.cz. "Spirála na rozdíl od vývrčky není ustáleným letovým režimem. K přechodu kluzáku do spirály dojde buď nesprávným uvedením do vývrčky, kdy nedojde k odtržení proudnic z jednoho křídla, nos kluzáku klesá pod horizont, nožní řízení bylo použito při příliš vysoké rychlosti. Kluzák za stálého zrychlování a v utahující se zatáčce klesá pod značným úhlem k zemi. Rozdílem patrným na první pohled je zvyšující se rychlost a méně strmý úhel klesání, než u vývrčky, rostoucí kladné přetížení. Pokud by nedošlo k včasné reakci, mohlo by velmi rychle dojít k překročení maximální rychlosti kluzáku a nepřiměřenému namáhání konstrukce" [5].

Strmý pád je možné definovat jako velmi rychlé klesání. Přesný úhel strmosti klesání není definován. Nicméně nácvik takového pádu pro kluzáky včetně vizualizace je popsán v příloze B.

3.5 Výběr vhodných parametrů

Seznam všech dostupných parametrů rozřazených do 6 kategorií definovaných níže je v tabulce 3.1.

1. Parametry, které se obvykle za letu ztelně nemění, nebo nejsou relevantní.
2. Parametry, které nejsou pro určení nebezpečného manévru směrodatné.
3. Parametry obtížně získatelné z reálného ultralehkého letounu.
4. **Parametry použité pro *Rule-based* systém.**
5. **Parametry použité pro učící systém.**

Tabulka 3.1: Přehled parametrů a jejich kategorizace

Parametr	Popis	Kategorie
APU_Generator	Pomocná pohonná jednotka	1
Aileron	Pozice křidélek	3
Aileron_trim	Vyvážení křidélek	1
Airspeed_kt	Vzdušná rychlost v uzlech	3
Altitude_ft	Výška ve stopách	2
Altitude_select	Nastavení výšky pro autopilota	2
Bank_angle_select	Nastavení úhlu klonění pro autopilota	2
Brake_left	Levá brzda	1
Brake_parking	Parkovací brzda	1
Brake_right	Pravá brzda	1
Cutoff	Motorový indikátor	1
Electric_pump	Elektrické čerpadlo	1
Elevator	Výškové kormidlo	3, 4
Elevator_Trim	Vyvážení výškového kormidla	1
Engage_autopilot	Spuštění autopilota	2
Engine_pump	Motorové čerpadlo	1
Engine_running	Indikátor běhu motoru	1
External_power	Externí napájení	1
Flaps	Klapky	2
Fuel_pump	Palivové čerpadlo	1
Gear_down	Indikátor vysunutí podvozku	1
Gear_position	Pozice podvozku	1
Glideslope	Sestupová rovina	2
Heading_deg	Kurz ve stupních	2
Heading_select	Nastavení kurzu pro autopilota	2
Ignition	Zapalování	1
Latitude_deg	Zeměpisná šířka ve stupních	2
Longitude_deg	Zeměpisná délka ve stupních	2
Mach	Machovo číslo	2
Magnetos	Dynama	1
Mixture1	Palivová směs	1
NED ¹ _down_accel_fps_sec	Zrychlení ve směru dolů	1 ²

Pokračování na další stránce

¹North-East-Down - souřadná soustava os směřujících na sever, východ a dolů.²Naměřená data neobsahují žádné relevantní hodnoty.

Tabulka 3.1 – pokračování z předešlé stránky

Parametr	Popis	Kategorie
NED_east_accel_fps_sec	Zrychlení ve směru na východ	1 ³
NED_north_accel_fps_sec	Zrychlení ve směru na sever	1 ⁴
Pitch_deg	Úhel příčného sklonu ve stupních	5
Propeller_pitch	Nastavení vrtule	1
Roll_deg	Úhel podélného náklonu ve stupních	5
rpm	Otáčky motoru	3
Ruder	Směrové kormidlo	3, 4
Rudder_trim	Vyvážení směrového kormidla	1
Side_slip_rad	Skluz v radiánech	2
Slats	Sloty	2
Speed_down	Rychlostní složka směrem dolů v NED soustavě	5
Speed_east	Rychlostní složka směrem na východ v NED soustavě	5
Speed_north	Rychlostní složka směrem na sever v NED soustavě	5
Speed_select	Nastavení horizontální rychlosti pro autopilota	2
Speedbrake	Rychlostní brzda	3
Starter	Startér	1
Steering	Zatáčení při pojíždění	1
Throttle	Tah motoru	3
uBody_fps	Rychlost otáčení v ose x ()	2
vBody_fps	Rychlost otáčení v ose y ()	5
Vertical_speed_fps	Vertikální rychlost	5 ⁵
Vertical_speed_select	Nastavení vertikální rychlosti pro autopilota	2
wBody_fps	Rychlost otáčení v ose z ()	2

³Naměřená data neobsahují žádné relevantní hodnoty.

⁴Naměřená data neobsahují žádné relevantní hodnoty.

⁵Obsahuje stejné hodnoty jako Speed.down, ale s opačným znaménkem.

Kapitola 4

Rule-based systém

Rule-based systém je poměrně jednoduchý systém používaný například pro analýzu dat v ČSA. Jedná se o algoritmus, který rozhoduje o zařazení logu do třídy na základě předem definovaných pravidel. Takový algoritmus pouze testuje, zda byla překročena hraniční hodnota některého z parametrů. Detailní popis vytvoření takového systému včetně hlubší analýzy zajímavých parametrů a následného testování je uveden níže.

4.1 Analýza zajímavých parametrů

Základní výběr parametrů je popsán v části 3.5. Tento výběr byl udělán pomocí hrubé analýzy a to tak, že byly vykresleny grafy všech logů z jednoho letu pro každý parametr. Na těchto grafech byly sledovány úseky, kde dle anotátora měl být nebezpečný manévr. Pokud se v těchto úsecích vyskytly významné změny, byl takový parametr považován za zajímavý a tudíž použitelný pro identifikaci rizikových situací.

Mezi zajímavé parametry patří:

Altitude: Prudký pokles výšky může indikovat pád. Dále také extrémní manévry jsou mnohem nebezpečnější v nízkých výškách.

Roll: Prudké klonění letadla, zejména v nízkých rychlostech může znamenat ztrátu vztlaku a následný pád.

Pitch: Při vysokých kladných hodnotách klopení letadla je možné očekávat snižování rychlosti letadla a následnou ztrátu vztlaku. Vysoké záporné hodnoty vyjadřují strmost pádu letadla.

Heading: Pomocí tohoto údaje lze sledovat, zda se letadlo pohybuje v kruzích. Toto může pomoci například při identifikaci vývrtky nebo spirály.

Airspeed: Rychlé změny, případně extrémní hodnoty rychlosti vypovídají o stavu letadla. Nízká rychlost představuje riziko ztráty vztlaku, zatímco velké zrychlení (zejména nad maximální rychlost letadla) může indikovat pád.

Elevator Position: Z polohy výškového kormidla lze vyčíst budoucí změny vertikální rychlosti letadla.

Left/Right Aileron Position: Vychýlení křidélek v opačném směru může indikovat nebezpečné klonění letadla.

Rudder Position: Velké vychýlení směrového kormidla po delší dobu může uvést letadlo do otáčivého pohybu.

4.2 Vybrané parametry

Vzhledem k potřebě odhalit především místo vzniku nebezpečné situace (počátek vývrtky), byly použity zejména údaje o pozici řízení. Ukázalo se, že většině sledovaných nebezpečných situací předchází podobný zásah v řízení. Pro vyhledání nebezpečí vývrtky byly použity hodnoty Elevator Position a Rudder Position. Změna výškového kormidla uvede letadlo do klesání, zatímco velké vychýlení směrového kormidla uvede letoun do otáčivého pohybu.

4.3 Algoritmus pro identifikaci vývrtky

Rule-based systém se omezuje na identifikaci vývrtky a to ze dvou hlavních důvodů. Získaný vzorek letových dat obsahoval mnohem více vývrtek než ostatních manévru. Identifikace jednoho manévru je pro tuto fázi dostačující, protože algoritmus má za úkol ověřit, zda je taková identifikace vůbec možná.

Pomocí programu MatLab byl vytvořen skript, zvýrazňující v trojrozměrném grafu letu místa, kde hrozí nebezpečí vývrtky. Následovalo ladění hraničních hodnot pro nalezení nejvyšší možné přesnosti určení nebezpečného manévru. Takto byly označeny části letu, které mají výškové kormidlo vychýlené pod hodnotu -0.2 a zároveň absolutní hodnotu směrového kormidla nad 0.2.

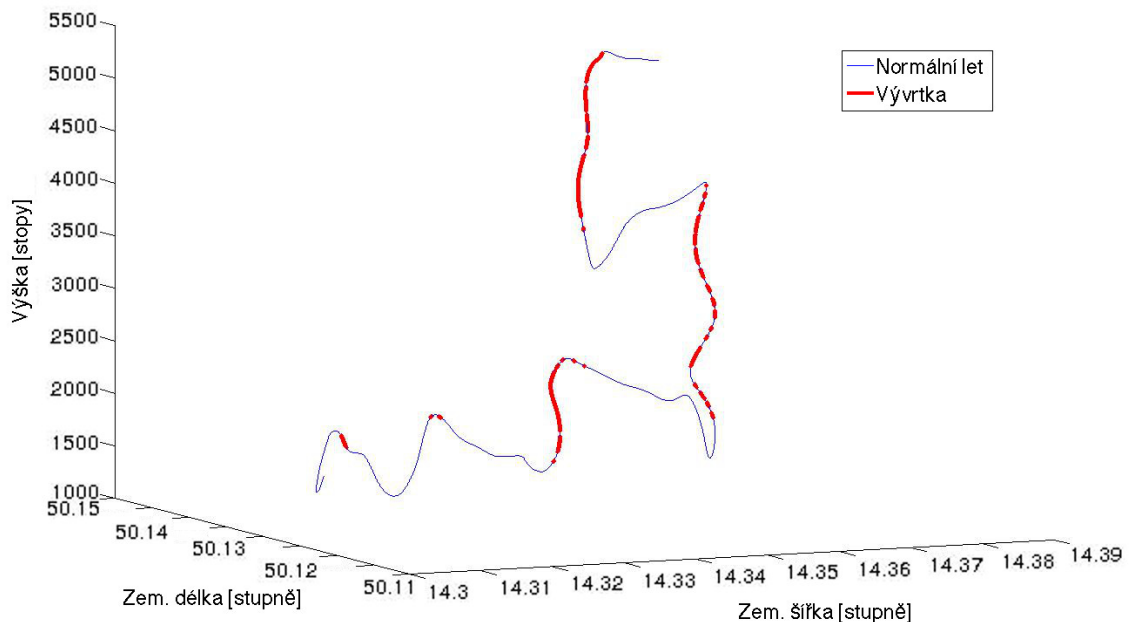

```

% Zvýraznění požadovaných bodů
% zakresluj do stávajícího grafu
hold on
% pro každý řádek v tabulce flight data
for j=1:length(flight_data)
% pokud elevator position < -0.2 a zároveň |rudder position| > 0.2
if (flight_data.elevator_pos_norm(j) < -0.2) & (abs(flight_data.rudder_pos_norm(j)) > 0.2)
% zakresli bod červeně
plot3(flight_data.Longitude_deg(j),flight_data.Latitude_deg(j),flight_data.Altitude_ft(j), 'r.')
end
end
end
hold off

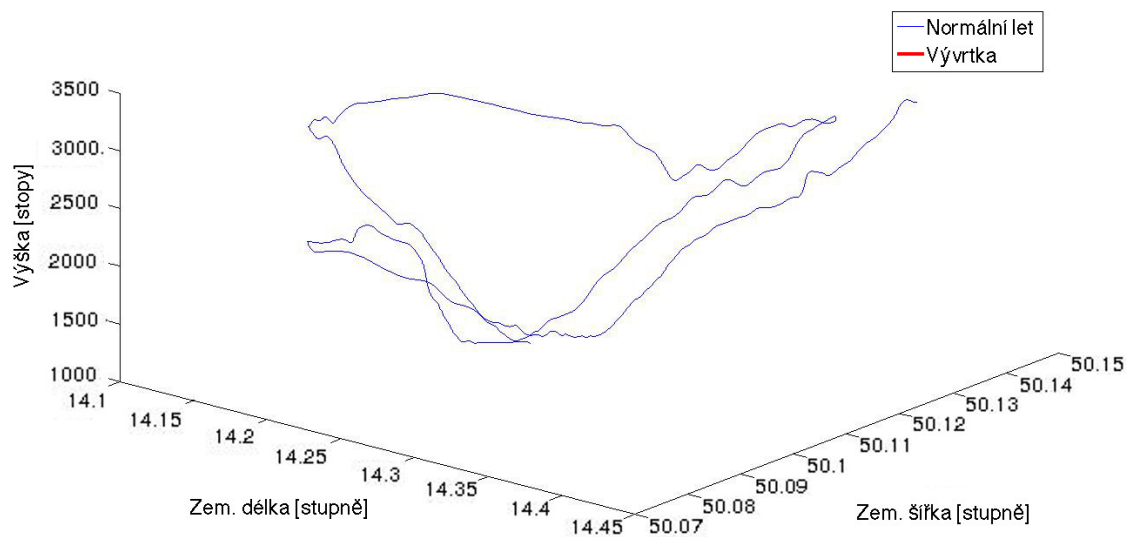
```

4.4 Výsledné grafy

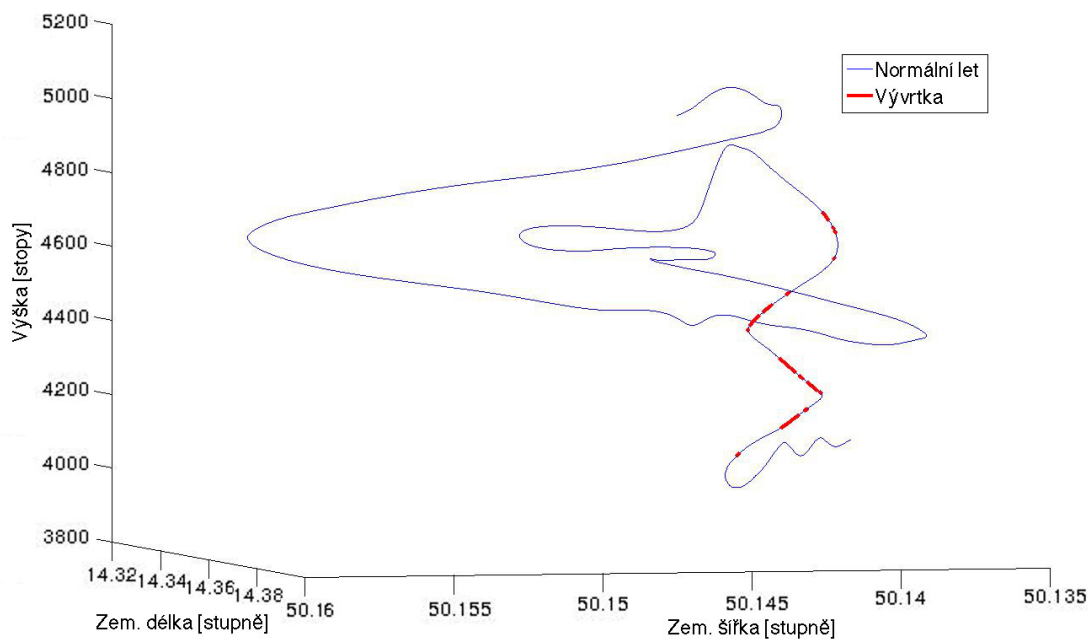
Indikace nebezpečí vývrtky na referenčních datech vývrtky je zobrazena na obrázku 4.1. Výsledky stejného algoritmu použitého nad testovacími daty běžného letu jsou zobrazeny na obrázku 4.2. Běžný let dle algoritmu neobsahuje žádnou vývrtku, což odpovídá názoru anotátora. Pro zajímavost jsou uvedeny také výsledky stejného algoritmu aplikovaného na referenční data označená anotátorem jako nácvik zatáček 4.3. Zde je vidět, že algoritmus indikoval v několika místech nebezpečí vývrtky, což může být dle následného posouzení anotátorem správné.



Obrázek 4.1: Indikace nebezpečí vývrtky na referenčních datech



Obrázek 4.2: Indikace nebezpečí vývrvky na běžném letu



Obrázek 4.3: Indikace nebezpečí vývrvky na referenčních datech zatáček

4.5 Možnosti rozšíření systému

S použitím více údajů by bylo možné rozšířit informace o úroveň nebezpečnosti manévru. Například záporné hodnoty výškového kormidla jsou nebezpečnější pokud je záporná hodnota klopení (a naopak). Prudké zvýšení hodnoty výškového kormidla při nízké rychlosti. Dále by se za pomoci údajů výšky a kurzu dal zpřesnit rozsah nebezpečného manévru, nicméně tato část práce se zaměřovala zejména na určení vzniku nebezpečné situace (vývrtky).

4.6 Zhodnocení systému

Algoritmus používá parametry, které nejsou běžně v reálných ultralehkých letadlech dostupné. Je příliš specializovaný a odladěný na konkrétních datech ze simulátoru. Navíc se podobné algoritmy k analýze letových dat již používají v praxi, proto nebude více rozváděn ani validován. K jeho validaci postačí vizuální kontrola výsledků viz. část 4.4.

Pro vybrané nebezpečné manévry lze jen velmi obtížně nastavit deterministicky pravidla pro jejich detekci tak, aby pracovala pouze s parametry, které jsou běžně dostupné v ultralehkých letadlech. Takový algoritmus by pravděpodobně nebyl ani příliš úspěšný.

Nadále se práce bude věnovat detekci nebezpečných manévru pomocí strojového učení a automatického rozpoznávání.

Kapitola 5

Učící se systém

Tato kapitola je věnována práci se systémy, které jsou schopny se učit a následně rozpoznávat, tedy rozřadit data do předem definovaných tříd. Existuje velké množství metod z nichž bylo vybráno několik, kterým bude věnována zvýšená pozornost.

Veškeré výpočty, testovací i finální algoritmy jsou provedeny v programu Matlab R2013b. Toolboxy použité pro zpracování dat byly Statistics a Neural Network Toolbox. Tyto Toolboxy jsou součástí univerzitní licence programu Matlab. Program Matlab byl zvolen, protože obsahuje všechny nástroje, které jsou potřeba pro pozdější práci, tedy statistiku, neuronové sítě, matematické výpočty, práci s daty, grafické výstupy apod. Navíc umožňuje export kódu do jazyka C nebo C++.

Nejdříve bude stručně představeno strojové učení a výběr validační metody. Následovat bude výběr nejlepší metody a parametrů rozdělený na několik fází.

5.1 Strojové učení

Strojové učení patří do podoblasti umělé inteligence. Zabývá se algoritmy, které se dokáží učit z referenčních dat a následně klasifikovat nová data. Základní rozdělení algoritmů strojového učení je podle vstupních dat.

Pokud jsou k dispozici anotovaná data, jedná se o učení s učitelem (supervised learning). Pokud vstupní data anotována nejsou, jedná se o učení bez učitele (unsupervised learning). Pokud jsou anotována pouze některá vstupní data, jedná se o kombinaci předcházejících dvou - tedy kombinace učení s učitelem a bez učitele (semi-supervised learning). [28, 26, 27]

Data použitá v této práci jsou anotována učitelem, tudíž bude použito učení s učitelem.

5.1.1 Učení s učitelem (supervised learning)

Učení s učitelem probíhá tak, že známá data a jejich anotace jsou předložena algoritmu učení. Tento algoritmus vytvoří takzvaný klasifikátor. Nová data jsou následně anotována pomocí tohoto klasifikátoru. Učení s učitelem lze dále rozdělit na klasifikaci a regresi. Klasifikace dělí výstup z algoritmu učení do tříd (například normální let, vývrtka spirála, ...), zatímco výstup regrese je spojitá hodnota. Pro účely této práce bude tudíž použita klasifikace. [28, 16]

5.2 Výběr validační metody

Existují v podstatě dvě hlavní metody validace - resubstituční metoda a křížová validace (cross-validation). Resubstituční metoda používá stejná data pro trénování i pro validaci. Tato metoda je nejjednodušším způsobem validace, ale její výsledky jsou oproti skutečným výsledkům na nových datech často příliš optimistické, a proto se moc nepoužívá.

V této práci byla tedy vybrána metoda křížové validace. Tato metoda rozděluje známá data na více částí. Následně používá jednu část k validaci a ostatní k trénování klasifikátoru. Tento postup opakuje pro všechny části a výsledná přesnost je průměrem jednotlivých přesností. [28, 4]

5.3 1. fáze: Zúžení výběru na tři metody

Tato fáze se dělí na dvě části. Nejprve bude popsán výběr parametrů a následně výběr třech nejlepších metod společně s nejlepší kombinací parametrů pro každou z nich.

5.3.1 Výběr vhodných parametrů

Výběr parametrů je popsán v kapitole 3. Algoritmus navíc pracuje s parametry `Speed_east` a `Speed_north` jako s jedním údajem a to tak, že z nich vypočítává výslednou rychlost v horizontálním směru pomocí vektorového sčítání.

$$Horizontal_Speed = \sqrt{|Speed_east|^2 + |Speed_north|^2} \quad (5.1)$$

Další drobnou výjimkou je parametr `Vertical_speed`, který obsahuje stejné hodnoty jako `Speed_down` jen s opačným znaménkem. Algoritmus pracuje s druhým zmiňovaným ačkoli v tabulce 3.1 je označen i první zmiňovaný.

Ostatní atributy jsou používány dle tabulky 3.1.

Použití všech vhodných parametrů pro rozhodování o třídě se neukázalo vždy jako to nejlepší. Proto bylo vhodné testovat vždy všechny možné kombinace parametrů, které jsou zaznamenány v tabulce 5.1. Označení parametrů bylo vysvětleno v části 3.5 v tabulce 3.1.

Tabulka 5.1: Kombinace parametrů

Číslo	Parametr 1	Parametr 2	Parametr 3	Parametr 4	Parametr 5
1	Speed_down	Roll_deg	Pitch_deg	Speed_horizontal	vBody_fps
2	Speed_down	Roll_deg	Pitch_deg	Speed_horizontal	nepoužito
3	Speed_down	Roll_deg	Pitch_deg	vBody_fps	nepoužito
4	Speed_down	Roll_deg	Speed_horizontal	vBody_fps	nepoužito
5	Speed_down	Pitch_deg	Speed_horizontal	vBody_fps	nepoužito
6	Roll_deg	Pitch_deg	Speed_horizontal	vBody_fps	nepoužito
7	Pitch_deg	Speed_horizontal	vBody_fps	nepoužito	nepoužito
8	Roll_deg	Speed_horizontal	vBody_fps	nepoužito	nepoužito
9	Roll_deg	Pitch_deg	vBody_fps	nepoužito	nepoužito
10	Roll_deg	Pitch_deg	Speed_horizontal	nepoužito	nepoužito
11	Speed_down	Speed_horizontal	vBody_fps	nepoužito	nepoužito
12	Speed_down	Pitch_deg	vBody_fps	nepoužito	nepoužito
13	Speed_down	Pitch_deg	Speed_horizontal	nepoužito	nepoužito
14	Speed_down	Roll_deg	vBody_fps	nepoužito	nepoužito
15	Speed_down	Roll_deg	Speed_horizontal	nepoužito	nepoužito
16	Speed_down	Roll_deg	Pitch_deg	nepoužito	nepoužito
17	Speed_horizontal	vBody_fps	nepoužito	nepoužito	nepoužito
18	Pitch_deg	vBody_fps	nepoužito	nepoužito	nepoužito
19	Pitch_deg	Speed_horizontal	nepoužito	nepoužito	nepoužito
20	Roll_deg	vBody_fps	nepoužito	nepoužito	nepoužito
21	Roll_deg	Speed_horizontal	nepoužito	nepoužito	nepoužito
22	Roll_deg	Pitch_deg	nepoužito	nepoužito	nepoužito
23	Speed_down	vBody_fps	nepoužito	nepoužito	nepoužito
24	Speed_down	Speed_horizontal	nepoužito	nepoužito	nepoužito
25	Speed_down	Pitch_deg	nepoužito	nepoužito	nepoužito
26	Speed_down	Roll_deg	nepoužito	nepoužito	nepoužito

5.3.2 Výběr tří nejlepších metod

Výběr metod pro učení a následné určování nebezpečných manévřů byl v první fázi zúžen na klasifikační metody, které pracují nad anotovanými daty (tzv. učení s učitelem neboli supervised learning):

1. K-Nearest Neighbors

- Klasifikace podle nejbližších sousedů. Tato klasifikační metoda je jeden z nejjednodušších algoritmů strojového učení. Algoritmus umístí každý prvek trénovací množiny do N -rozměrného prostoru. Při klasifikaci umístí nový prvek do stejného prostoru a vyhledá pro něj nejbližší sousedy.
- Objekt je následně klasifikován do té třídy, do které náleží většina jeho nejbližších sousedů. Speciálním případem metody Klasifikace podle nejbližších sousedů je tzv. Metoda nejbližšího souseda, kde $k = 1$. Pro zpřesnění ostatních případů ($k > 1$) této metody lze přidat váhy pro jednotlivé sousedy. Nejčastěji se používají váhy $\frac{1}{v}$ a $\frac{1}{v^2}$ kde v je vzdálenost souseda od klasifikovaného prvku. [28, 20, 21]
- V textu a tabulkách bude tato metoda nadále označována zkratkou KNN.

2. Classification Tree

- Rozhodovací strom je stromová struktura, ve které každý uzel reprezentuje test na vstupní parametry, každá větev z něj vystupující reprezentuje možný výsledek testu a jednotlivé listy stromu jsou výsledné třídy. [28, 18, 19]
- Při vytváření rozhodovacího stromu se hledají takové testy, které od sebe nejlépe oddělí jednotlivé třídy. Příkladem pro data použitá v této práci může být vysoká rychlost klesání oddělující ostrý pád od normálního letu, nebo vysoká rychlost otáčení dělicí spirálu a vývrtku od normálního letu s ostrým pádem. [28, 18, 19]
- V textu a tabulkách bude tato metoda nadále označována CTree. Zjednodušené základní schéma rozhodovacího stromu pro tuto práci je k dispozici na obrázku 5.1.

3. Discriminant Analysis

- Pro každou definovanou třídu se vypočítá diskriminační funkce, na základě které se počítá pravděpodobnost náležení prvku do dané třídy. [16, 15, 28]
- Diskriminační funkce jsou funkce, které definují rozdělující nadplochy. Rozdělující nadplochou je označena množina vektorů příznaků (v této práci označované slovem parametr), které jsou hraničními body dvou množin jež patří do různých tříd. Diskriminační funkcí r -té třídy je každá funkce $g_r(x)$ splňující nerovnost $g_r(x) > g_s(x)$ pro každý vektor příznaků patřící do množiny vektorů příznaků náležejících r -té třídě a pro $s = 1, 2, \dots, R, s \neq r$. [13]

- V textu a tabulkách bude tato metoda nadále označována zkratkou DA.

4. Support Vector Machines

- "Metoda podpůrných vektorů hledá nadrovinu, která v prostoru příznaků optimálně rozděluje trénovací data. Optimální nadrovina je taková, že body leží v opačných poloprostorech a hodnota minima vzdáleností bodů od roviny je co největší. Jinými slovy, okolo nadroviny je na obě strany co nejširší pruh bez bodů." [25]
- "Na popis nadroviny stačí pouze nejbližší body, kterých je obvykle málo - tyto body se nazývají podpůrné vektory (angl. support vectors) a odtud název metody. Tato metoda je ze své přirozenosti binární, tedy rozděluje data do dvou tříd. Rozdělující nadrovina je lineární funkcí v prostoru příznaků." [25]
- Tato metoda umožňuje rozdělení pouze do dvou tříd, tudíž není vhodná pro klasifikaci dat v této práci a v dalších krocích již nebude obsažena.

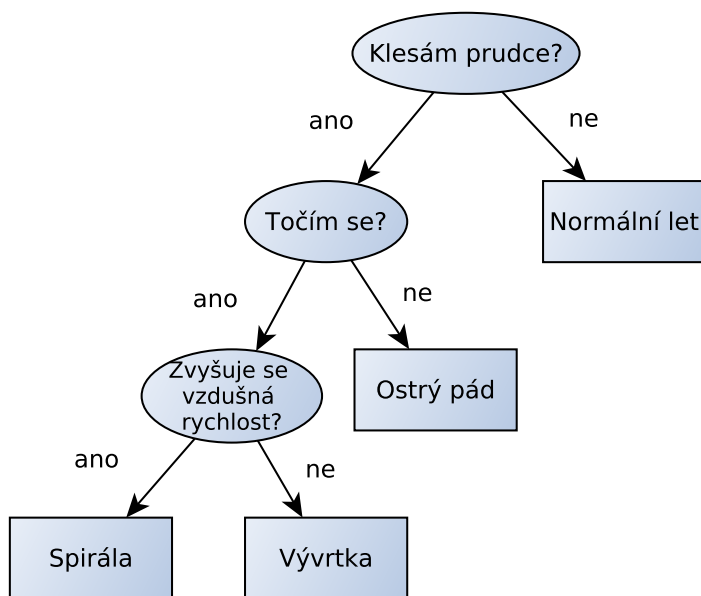
5. Naive Bayes Classification

- Vychází z výpočtu pravděpodobnosti podle Bayesovy věty viz. obrázek 5.2. Pro každý nový vzorek metoda počítá posteriorní pravděpodobnost tohoto vzorku pro každou třídu. Následně klasifikuje vzorek podle největší posteriorní pravděpodobnosti. [28]
- Naivní Bayesův klasifikátor předpokládá, že parametry jsou na sobě nezávislé. Ve většině případů ovšem pracuje dobře i pokud tomu tak není.
- V textu a tabulkách bude tato metoda nadále označována zkratkou NB.

6. Neural Networks

- Neuronové sítě jsou složeny z prvků "neuronů". Tyto neurony mají váhy, které určují jejich výstup. Neuronová síť obsahuje jednu nebo více vrstev těchto neuronů.
- Algoritmus učení probíhá tak, že se nejdříve nastaví počáteční váhy (většinou náhodně) a následně se na vstup použijí trénovací data. Podle výsledku klasifikace na výstupu jsou vždy upraveny váhy uvnitř neuronové sítě. [28, 22, 23, 16, 1]
- V textu a tabulkách bude tato metoda nadále označována zkratkou NN, případně NNX přičemž X označuje počet skrytých vrstev neuronů.

Je vhodné nejprve vyzkoušet, zda výběr použitých parametrů a metod nebyl zcela nevhodný. Pro tuto zkoušku postačí identifikace 2 tříd (vývrtka, normální let). Tabulka 5.2 potvrzuje, že učení nad těmito parametry pomocí vybraných metod je možné. Hodnoty



Obrázek 5.1: Zjednodušené schéma rozhodovacího stromu pro klasifikaci vybraných nebezpečných manévrů

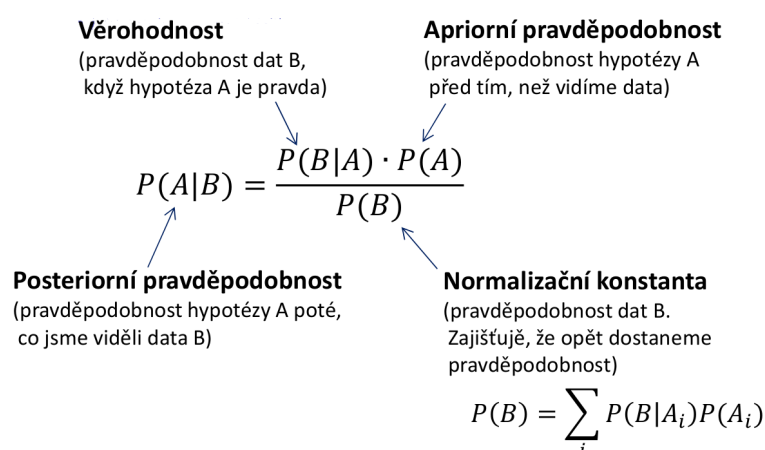
Tabulka 5.2: Identifikace vývrtky

Metoda roz- poznávání	NN	DA	KNN	C'Tree	NB
Přesnost určení	84,8895	76,1435	81,9364	84,1743	77,2786

uvedené v tabulce znázorňují přesnost klasifikace oproti klasifikaci provedené anotátorem. Výpočet přesností bude detailně popsán dále.

V této fázi je provedeno testování jednotlivých metod a kombinací parametrů s cílem vybrat 3 nejlepší metody společně s jejich nejlepšími kombinacemi parametrů. Blokované schéma algoritmu pro výběr nejlepší metody s nejlepší kombinací parametrů je uvedeno na obrázku 5.3. Vývojový diagram tohoto algoritmu je na obrázku 5.4.

Testování je provedeno nad základním nastavením jednotlivých metod. To znamená, že pro většinu metod bylo použito základní nastavení Matlabu s výjimkou NN a DA. Pro NN je v základu nastaveno 10 skrytých vrstev, to je pro první testování zbytečné a příliš zdlouhavé, proto bylo zvoleno 6 vrstev. Testování pro různé počty vrstev je provedeno ve 2. fázi 5.4. Pro DA je v základu nastavena lineární DA, ta není pro daná data vhodná, proto byla rovnou zvolena vhodnější kvadratická DA. Testování různých typů DA je taktéž provedeno ve 2. fázi 5.4.

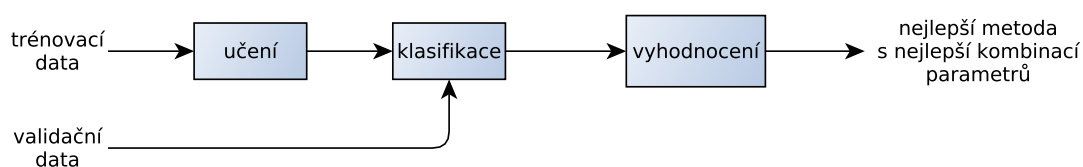


Obrázek 5.2: Bayesova věta ve vztahu k Bayesovu klasifikátoru [12]

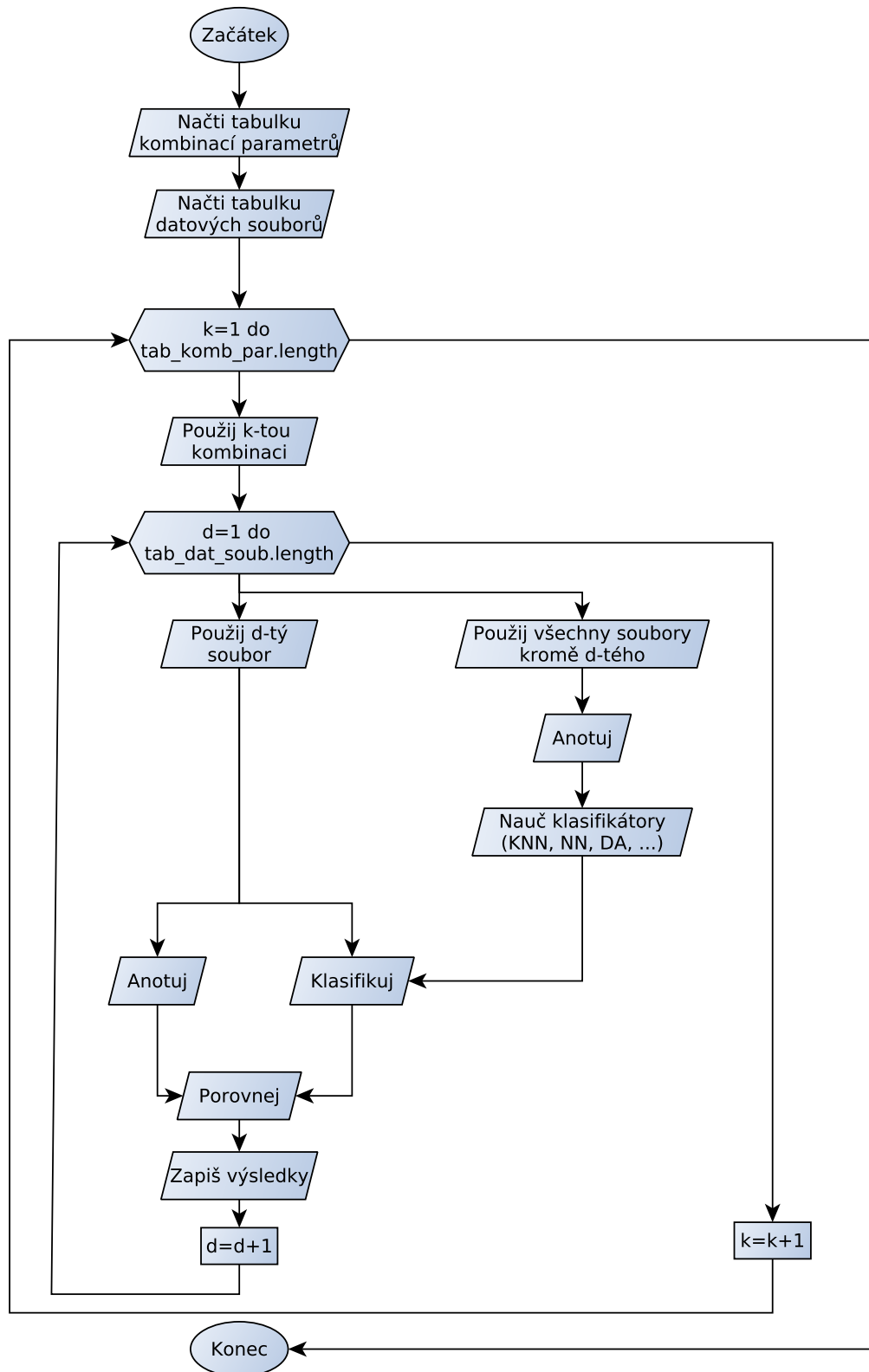
Výsledky identifikace všech 4 tříd najednou jsou v tabulce 5.3. Tabulka ukazuje jednak výsledky pro každý manévr (vývrtka, spirála, ostrý pád) zvlášť a pak také výsledky analýzy pro všechny manévry dohromady. Hodnoty uvedené v tabulce znázorňují přesnost klasifikace oproti klasifikaci provedené anotátorem v %.

Tabulka 5.3: Identifikace jednotlivých manévrů

Manévr / Metoda	NN	DA	KNN	CTree	NB
Vývrtka	80,9943	82,0324	80,9928	77,1041	80,4875
Spirála	68,0974	71,8475	76,1713	68,8386	72,9188
Ostrý pád	86,7466	83,1644	84,7589	81,1796	83,1644
Všechny manévry	80,2101	81,0388	80,8769	79,3778	79,9506
Nejlepší kombinace parametrů pro 4 třídy	14	20	3	14	20



Obrázek 5.3: Blokové schéma pro výběr nejlepší metody a kombinace parametrů



Obrázek 5.4: Vývojový diagram výběru nejlepší metody a kombinace parametrů

Přesnost identifikace manévru (klasifikace) je určena jako součet všech záznamů, kde se anotace a identifikace shodovaly, dělený celkovým počtem záznamů (viz. vzorec 5.2). Takto je určena přesnost klasifikace pro všechny metody (5), pro všechny validační množiny dat (14) a pro všechny kombinace parametrů (26).

Pro každou kombinaci parametrů každé metody byla spočítána celková přesnost jako vážený průměr jednotlivých přesností. Váhy byly určeny podle počtu anotovaných dat pro každou třídu, tedy 8 pro validační data vývrtek, 3 pro validační data spirál a 3 pro validační data ostrých pádů. Nejvyšší hodnota váženého průměru přesnosti pro každou metodu byla zapsána do tabulky, včetně kombinace parametrů pro kterou byla vypočítána.

$$A_c = \frac{z_{a=i}}{z} * 100 \quad (5.2)$$

A_c úspěšnost metody v procentech

$z_{a=i}$ počet logů, které byly shodně klasifikovány metodou i anotátorem

z celkový počet logů

Pseudokód pro výpočet úspěšnosti každé metody pro každou kombinaci.

```
for kombinace_parametrů=1:26
  for metoda=1:6
    součet_vážených_prvků(kombinace_parametrů,metoda)=sum(vážené_prvky(1:14,metoda,kombinace_parametrů))
    vážený_průměr(kombinace_parametrů,metoda)=součet_vážených_prvků(kombinace_parametrů,metoda)/součet_vah
  end
end
end
```

Třemi nejlepšími metodami jsou KNN, DA a NN. Ty budou nadále testovány výhradně s nejlepšími kombinacemi parametrů, které mají zapsané v tabulce 5.3.

5.4 2. fáze: Optimalizace tří nejlepších metod

Následuje popis jednotlivých testování modifikací atributů každé ze 3 metod vybraných v předcházející fázi. Jinými slovy bude zkoušeno různé nastavení metod s cílem vybrat nejlepší nastavení pro každou metodu.

5.4.1 Testování neuronových sítí

Neuronové sítě byly testovány pro různý počet skrytých vrstev. Výsledky tohoto dlíého testování jsou k dispozici v tabulce 5.4

Tabulka 5.4: Výsledky testování modifikací NN

Počet vrstev	Přesnost pro 4 třídy
1	74,4661
2	75,9562
3	75,8998
4	77,9381
5	79,1294
6	80,2101
7	79,0763
8	77,2469
9	79,7235
10	80,5789
11	81,0586
12	80,8883

5.4.2 Testování diskriminační analýzy

Pro diskriminační analýzu byly testovány různé typy diskriminačních funkcí, konkrétně lineární diskriminační funkce (ta není příliš vhodná, tudíž ani nemá dobré výsledky), kvadratická, diagonální lineární a diagonální kvadratická diskriminační funkce. Diagonální znamená, že pro odhad zařazení do třídy používá diagonální kovarianční matici (naivní Bayesovské klasifikátory).

Výsledky tohoto testování jsou uvedeny v tabulce 5.5.

Tabulka 5.5: Výsledky testování modifikací DA

Diskr. fce	Přesnost pro 4 třídy
Linear	69,2852
Quadratic	81,0388
diagLinear	69,5603
diagQuadratic	79,8622

5.4.3 Testování nejbližších sousedů

KNN metoda byla testována pro $k = \langle 2..6 \rangle$, $k = 1$ není nutné testovat, protože jeho výsledky jsou totožné s výsledky pro $k = 2$ váhované. V tabulce 5.6 je vidět jeden zajímavý jev a to, že pro neváhované KNN jsou výsledky lepší než pro váhované KNN.

Řádky v tabulce 5.6 označují metriku použitou pro výpočet vzdáleností (*Euclidean*, *City block*, *Chebyshev*, ...), vyhledávací metodu (*kdtree*, *exhaustive*) a zároveň způsob váhování (*equal*, *inverse*, *squared inverse*). Zapsané hodnoty potom označují jaká byla přesnost identifikace v % a pro jaké k (v tabulce je uveden pouze nejlepší výsledek pro všechna testovaná k). V tabulce nejsou uvedeny hodnoty pro kombinace některých metrik s vyhledávací metodou *kdtree*, protože tyto kombinace nelze použít.

Tabulka 5.6: Výsledky testování modifikací KNN

	kdtree equal	kdtree inverse	in- verse	kdtree squared inverse	exhaustive equal	exhaustive inverse	exhaustive squared inverse
Euclidean	2NN - 80,8769	2NN - 80,8455	-	2NN - 80,8455	2NN - 80,8769	2NN - 80,8455	2NN - 80,8455
Standard. Euclidean					2NN - 80,7958	3NN - 80,7648	3NN - 80,7698
City block	6NN - 80,8734	2NN - 80,6347	-	2NN - 80,6347	6NN - 80,8734	2NN - 80,6347	2NN - 80,6347
Chebyshev	2NN - 80,8297	2NN - 80,6867	-	2NN - 80,6867	2NN - 80,8297	2NN - 80,6867	2NN - 80,6867
Minkowski	2NN - 80,8769	2NN - 80,8455	-	2NN - 80,8455	2NN - 80,8769	2NN - 80,8455	2NN - 80,8455
Mahalanobis					6NN - 80,1822	6NN - 80,0482	6NN - 80,0444
Cosine					4NN - 76,0134	6NN - 75,8170	6NN - 75,8025
Correlation					4NN - 69,9777	6NN - 68,9641	6NN - 68,8757
Spearman					4NN - 73,5213	4NN - 73,5213	4NN - 73,5213
Hamming					3NN - 69,3860	3NN - 69,3860	3NN - 69,3860
Jaccard					3NN - 69,3860	3NN - 69,3860	3NN - 69,3860

Z výsledků vyplývá, že původní nastavení KNN bylo již optimalizované, protože změnou atributů není dosaženo lepších výsledků.

5.4.4 Zlepšení přesností po optimalizaci metod

Po optimalizaci jednotlivých metod došlo ke zvýšení přesnosti rozpoznání u NN. KNN a DA mají hodnoty stejné. Výsledky jsou uvedeny v tabulce 5.7. Je možné je porovnat s výsledky dosaženými bez optimalizace viz. 5.3.

Tabulka 5.7: Přesnost identifikace jednotlivých manévrů po optimalizaci metod

Manévr	NN 11 vrstev	QDA	KNN: k=2, Euclidean, equal
Vývrtka	81,9085	82,0324	80,9928
Spirála	69,1120	71,8475	76,1713
Ostrý pád	86,9614	83,1644	84,7589
Všechny manévry	81,0586	81,0388	80,8769
Kombinace parametrů	14	20	3

5.5 3. fáze: Vyhlazení

Metody se učí a klasifikují takzvaně naslepo. To znamená, že vyhodnocují každý log zvlášť. Tím pádem výsledky nejsou na trajektorii celistvé. Prakticky se stává to, že algoritmus metody nalezne například dva logy, které jsou klasifikovány jako vývrtka někde uprostřed normálního letu. To je při vzorkování logů 50Hz evidentně chybné vyhodnocení. Proto je dalším krokem eliminace těchto chyb, pracovně nazývaná *vyhlazení*.

Funkce vyhlazení prochází klasifikace jednotlivých logů. Pokud nalezne změnu klasifikace, zkontroluje délku této klasifikace (kolik logů po sobě má stejnou klasifikaci). Pokud je tato délka kratší než předem definovaný limit, překlasifikuje logy do předcházející nalezené třídy. Ukázka zdrojového kódu pro tuto funkci je uvedena níže.

```
%% Correct evaluated data
threshold = 100;
last_index=1;
last_value=1;
corrected_data=nn_results;
corrected_data(length(corrected_data)+1)=99;
for i=2:length(corrected_data)
    if corrected_data(last_index)~=corrected_data(i)
        if (i - last_index) < threshold
            corrected_data(last_index:i-1) = last_value;
        end
        last_index=i;
        last_value=corrected_data(i-1);
    end
end
% Remove temporary value 99 from the end of the array.
nn_results=corrected_data(1:length(corrected_data)-1);
```

Změny v úspěšnostech celkové klasifikace pro různé limity jsou vidět v tabulce 5.8.

Výsledky s vyhlazením jsou zaznamenány v tabulce 5.9. Limit byl nastaven pro každou metodu zvlášť dle nejlepších výsledků v tabulce 5.8

Tabulka 5.8: Hledání nejlepšího limitu pro vyhlazení výsledků

Limit	NN 11 vrstev	QDA	KNN: k=2, Euclidean, equal
0	81,0586	81,0388	80,8769
50	81,9460	81,1299	82,3003
100	82,5907	81,4257	82,5500
150	81,9319	81,9014	81,5568
200	80,8361	81,4762	81,2271

Tabulka 5.9: Přesnost identifikace jednotlivých manévrů po vyhlazení výsledku

Manévr	NN 11 vrstev	QDA	KNN: k=2, Euclidean, equal
Vývrtka	83,7064	82,6409	82,8274
Spirála	69,8319	75,3726	77,6521
Ostrý pád	87,4153	83,1714	85,4751
Všechny manévry (4 třídy)	82,5907	81,9014	82,5500

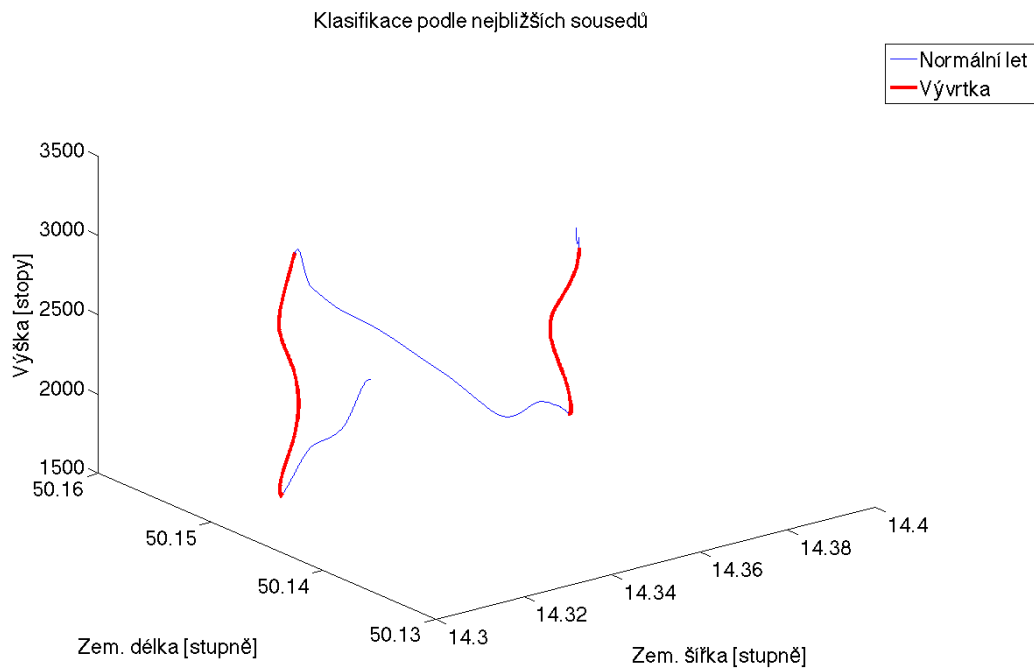
5.6 Vyhodnocení výsledků

Eliminace chybné klasifikace osamělých logů je velice žádoucí, jako výchozí výsledky pro rozhodnutí budou tedy uvažované výsledky dosažené po vyhlazení viz. tabulka 5.9. Přesnosti určení všech manévrů (4 tříd) najednou jsou ty nejzajímavější, ale zároveň jsou velmi podobné. Proto je nyní nejlepší udělat kontrolu grafických výstupů (obrázky 5.5 - 5.13) samotným anotátorem. Pro porovnání jsou k dispozici stejné vzorky označené anotátorem v části 3.2

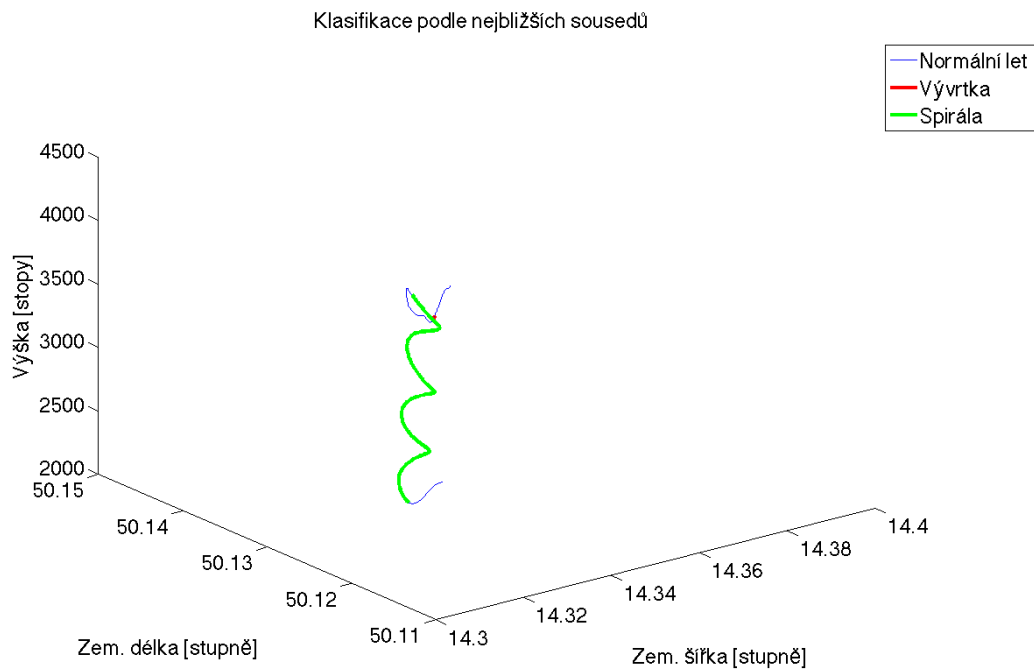
Po vizuální kontrole grafických výstupů je zřejmé, že NN nedávají ve skutečnosti nejlepší výstup ačkoli mají nejvyšší přesnost. To je způsobené váženým výpočtem přesností (vývrtky mají větší váhu, protože je k dispozici více referenčních dat). NN tedy nejlépe rozpoznává největší počet vývrtek, u ostatních manévrů ovšem zaostává. Podstatně lepší grafické výstupy má metoda KNN, která se navíc v hodnotě přesnosti liší jen o 0,04%.

Metodou, která bude implementována do výsledných skriptů 6 je tedy KNN. Použitá kombinace parametrů je číslo 3, tedy Speed_down, Roll_deg, Pitch_deg a vBody_fps. Metoda vyhledává dva nejbližší sousedy ($k = 2$), používá *Euclidovskou* metriku, nepřirazuje sousedům *žádné váhy* a používá vyhledávací metodu *kdtree*. Možností nastavení atributů KNN bylo více, tato je však nejjednodušší a tedy i výpočetně nejrychlejší.

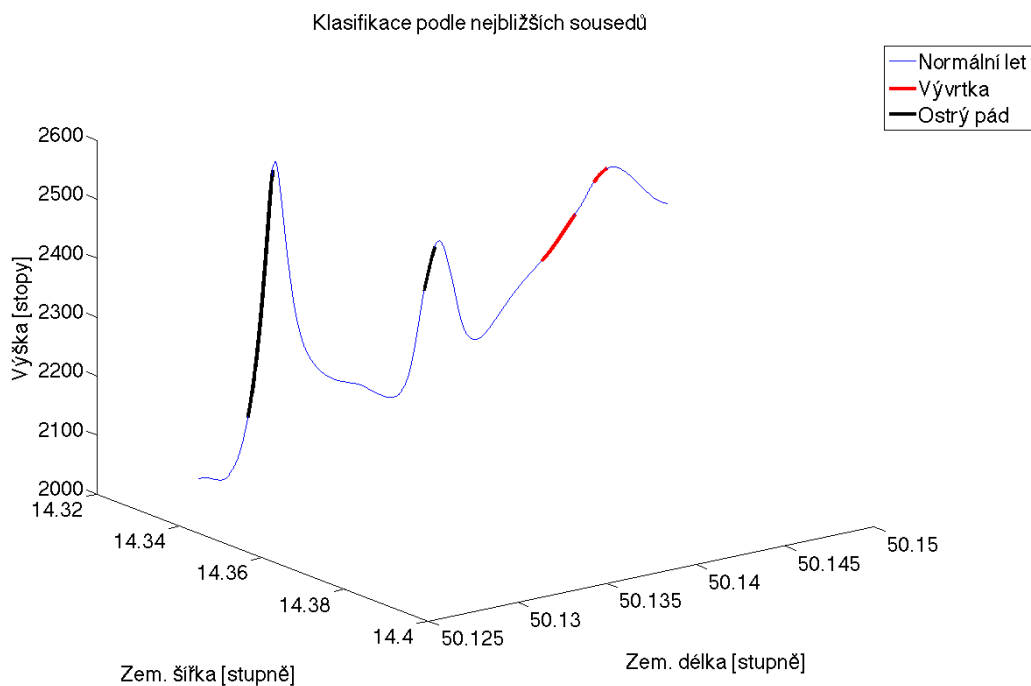
KNN má i další nespornou výhodu oproti NN a tou je náročnost výpočtu, která je u KNN podstatně nižší.



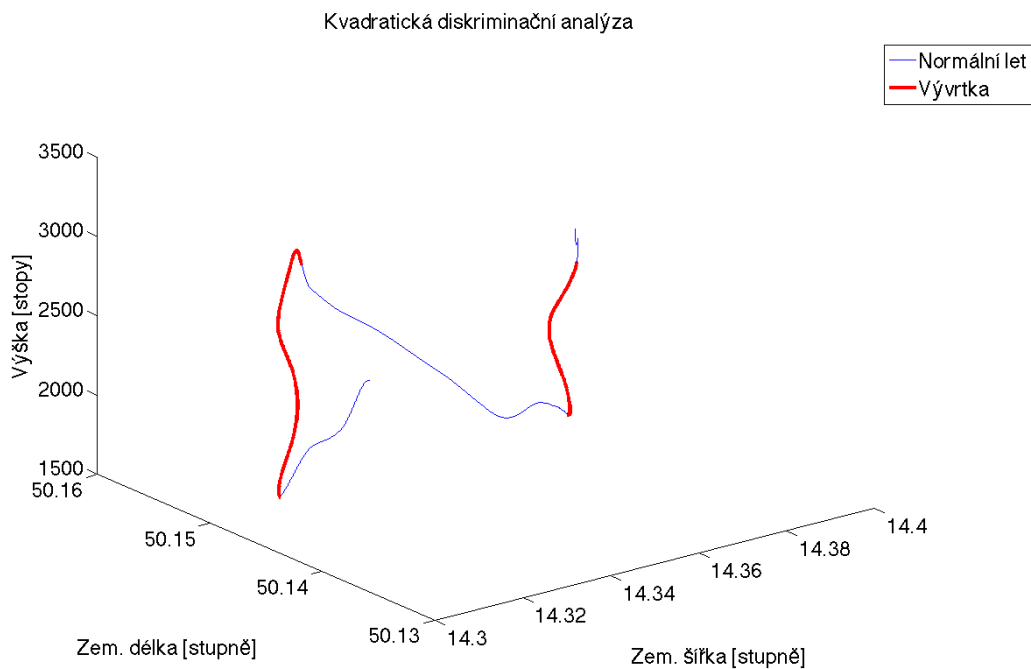
Obrázek 5.5: Grafické zobrazení výsledku KNN na letu obsahujícím 2 vývrtky



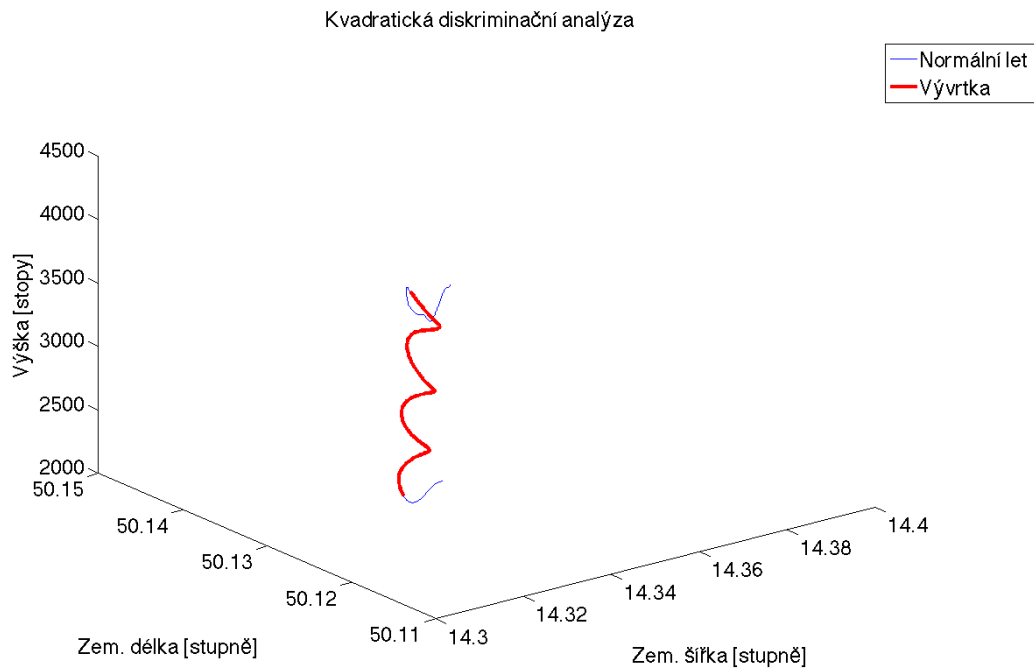
Obrázek 5.6: Grafické zobrazení výsledku KNN na letu obsahujícím 1 spirálu



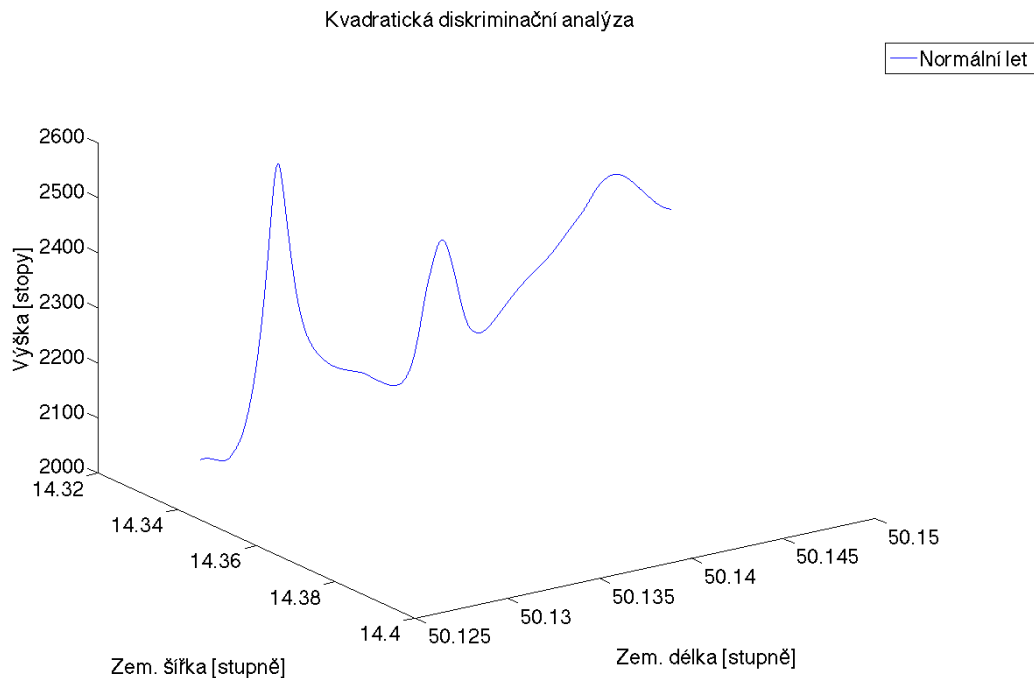
Obrázek 5.7: Grafické zobrazení výsledku KNN na letu obsahujícím 2 ostré pády



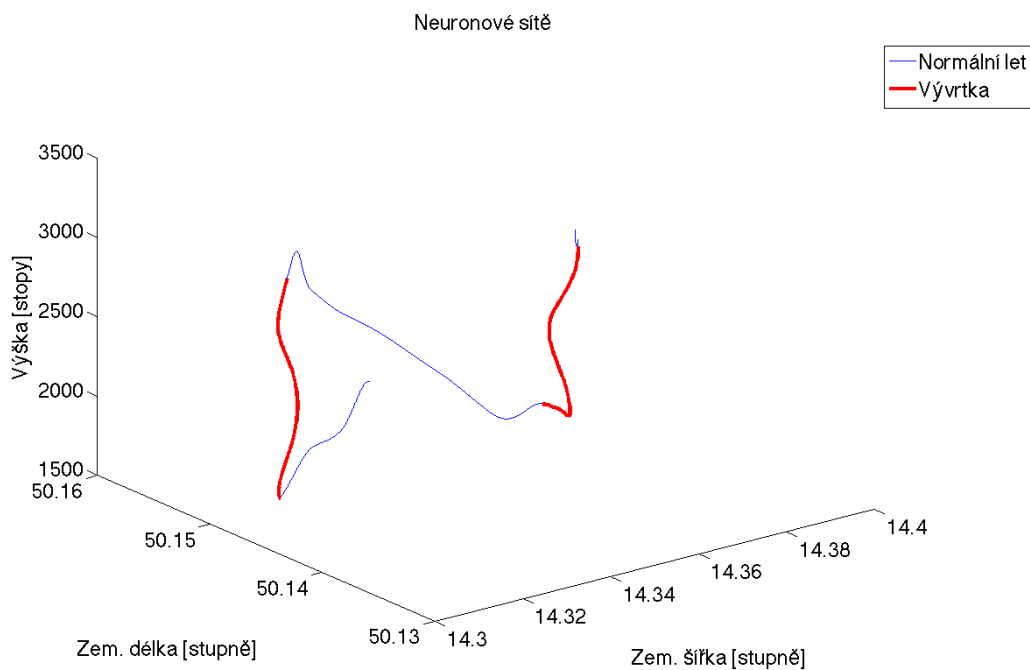
Obrázek 5.8: Grafické zobrazení výsledku DA na letu obsahujícím 2 vývrtky



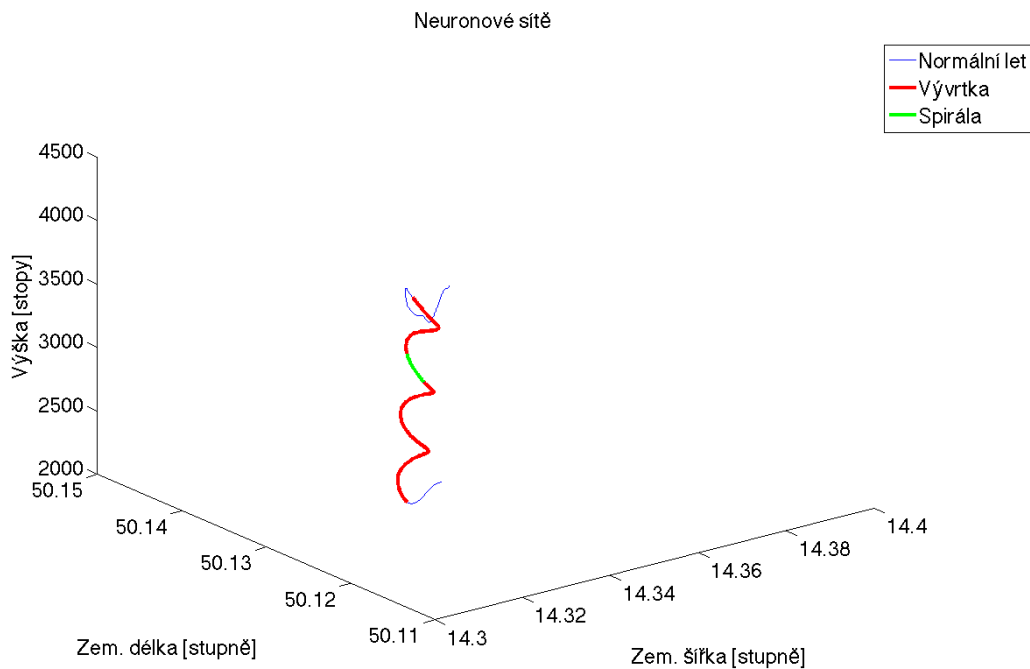
Obrázek 5.9: Grafické zobrazení výsledku DA na letu obsahujícím 1 spirálu



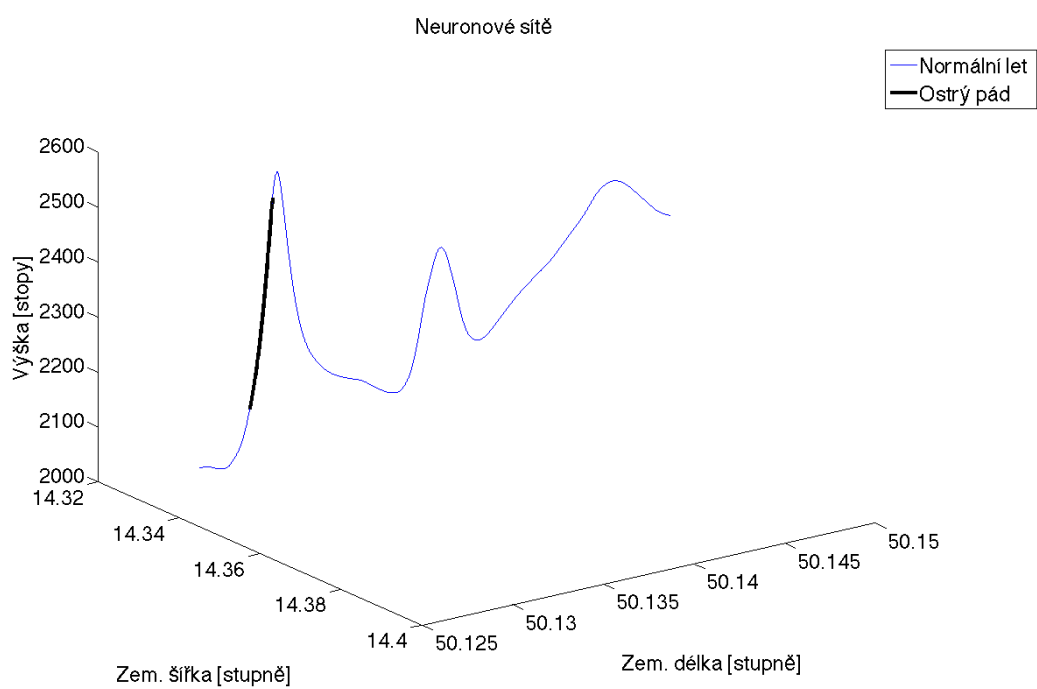
Obrázek 5.10: Grafické zobrazení výsledku DA na letu obsahujícím 2 ostré pády



Obrázek 5.11: Grafické zobrazení výsledku NN na letu obsahujícím 2 vývrtky



Obrázek 5.12: Grafické zobrazení výsledku NN na letu obsahujícím 1 spirálu



Obrázek 5.13: Grafické zobrazení výsledku NN na letu obsahujícím 2 ostré pády

Kapitola 6

Aplikace nejlepší metody

Pro uživatele jsou připraveny skripty pro naučení a použití nejlepší metody (tedy KNN). První skript je určen k přípravě (naučení) klasifikátoru. Počítá se s budoucím rozšířením trénovací množiny dat. Druhý skrip používá naučené klasifikátory ke klasifikaci nových dat.

Oba skripty jsou uloženy v elektronické podobě na CD viz. příloha [D](#).

6.1 Trénovací skript

Pro spuštění učícího skriptu (`teach_knn`) je třeba mít připraveny stávající trénovací data v jednom adresáři. Skript po spuštění vyzve k zadání adresáře obsahující tato data. Po načtení trénovacích dat skript připraví daný klasifikátor. **POZOR!** Skript při startu smaže všechny proměnné v prostředí. Pro přidání nových trénovacích dat je třeba doplnit ve zdrojovém souboru skriptu název souboru do pole “`filenames`” a přidat anotaci dat do přepínače (`switch`) v bloku “`Load test data`”. Dále je potřeba mít nová data ve stejném adresáři s ostatními trénovacími daty.

6.2 Klasifikační skript

Spuštění klasifikačního skriptu (`use_knn`). Pro funkci skriptu je nutné mít klasifikátor již natrénovaný. Po spuštění je uživatel požádán o zadání cesty k souboru, který chce klasifikovat. Výstupem klasifikačního skriptu je 3D graf s barevně odlišenými třídami letu. Vývrтка je označena červenou, spirála zelenou a ostrý pád černou barvou. Dále jsou do pracovního prostředí vypsány řádky obsahující nebezpečné manévry. Příklad takového výpisu je:

Od 1437 do 2188 byla detekována vývrтка.

Od 5017 do 5849 byla detekována vývrтка.

6.3 Možnosti rozšíření

Předpokládá se, že trénovací množina se bude dále rozrůstat. Algoritmus bude nutné více trénovat a opětovně testovat. Až se dostane na přijatelnou přesnost klasifikace, bude možné ho použít v konkrétních aplikacích, například pro tvorbu aplikace do mobilního telefonu či tabletu.

Autorka předpokládá, že vybrané parametry by měly být dostupné i například z mobilního telefonu. Tuto skutečnost je však potřeba ověřit. Pokud by bylo zjištěno, že tyto parametry nejsou dostupné z tzv. chytrých mobilních telefonů, bylo by nutné provést analýzu s novými vstupními parametry následovanou spuštěním trénovacího skriptu na všech dostupných datech. Takto natrénovaný algoritmus by pak bylo vhodné implementovat například v programovacím jazyce C/C++ (aplikace pro operační systém iOS) nebo v jazyce Java (aplikace pro operační systém Android).

Kapitola 7

Závěr

Identifikace rizikových situací v ultralehkém létání rozhodně nepatří mezi snadné úlohy. Skrývá v sobě mnoho kroků, které samotné identifikaci předcházejí.

V první řadě je potřeba si ujasnit, které rizikové situace mají být identifikovány. V této práci to byly tři nebezpečné manévry, vývrtka, spirála a ostrý pád. Tyto situace samozřejmě nepokrývají veškerá rizika, ale výsledný algoritmus je snadné rozšířit o identifikaci dalších situací (za předpokladu, že budou k dispozici referenční data v dostatečném množství).

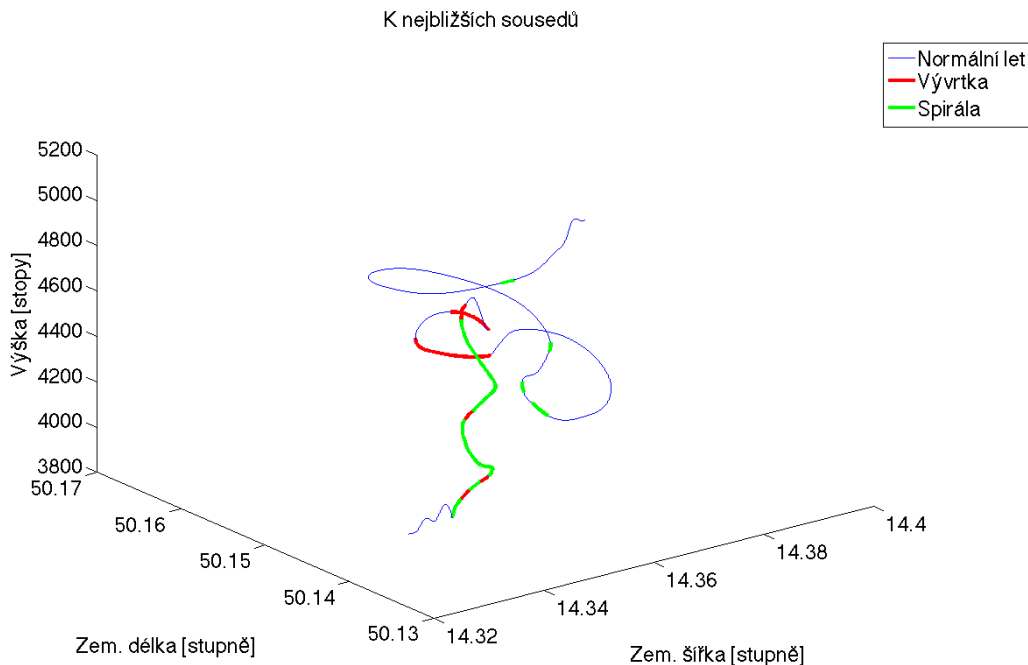
Další důležitou otázkou je: Jak tyto situace identifikovat? Možných přístupů je mnoho, v každém případě je však potřeba mít nějaká vstupní data. Vstupní data v této práci byla získána pomocí simulátoru na Fakultě elektrotechnické ČVUT.

Na těchto datech byl nejprve odzkoušen jednoduchý "rule-based" systém, který ukázal, že nějaký způsob identifikace daných manévru existuje a mohl by být funkční. Tento systém tak, jak byl nastaven by ovšem v reálných situacích nemohl fungovat, protože pracuje čistě s informacemi o nastavení řídicích ploch, které ze skutečného letadla běžně získat nelze.

Další možností bylo použití algoritmů strojového učení a automatického rozpoznávání. Zde vyvstalo spoustu dalších otázek. Učení s učitelem nebo bez? Které vstupní parametry použít? Které algoritmy? Jak mají být algoritmy nastaveny? Je potřeba výstupy dále upravovat? ... Kapitola 5 odpovídá právě na tyto otázky.

Učení s učitelem, které bylo v této práci použito, s sebou přináší další důležitý úkol. Tím je anotace referenčních dat. Tato data byla v rámci práce anotována s nejlepším možným odhadem autorky. To ale ovšem neznamená, že všechny anotace jsou bezpodmínečně správné. Navíc bylo během anotací zjištěno, že jen malé procento získaných dat je opravdu vzorovým manévrem. Některé soubory musely být dokonce z množiny referenčních dat vyloučeny. Doplnění většího množství dat (nejen ze simulátoru, ale i z reálných letů) se vzorovými manévry řádně anotovanými pravděpodobně výrazně zlepší funkci celého systému.

Algoritmus byl i přes všechny obtíže vytvořen a natrénován. Jak vypadá jeho grafický výstup na neanotovaném testovacím souboru lze vidět na obrázku 7.1.



Obrázek 7.1: Grafický výstup testování systému na neanotovaném letu

Práce jednoduše dokazuje, že identifikovat rizikové manévry z letových dat je možné. Referenčních dat nebylo k dispozici mnoho. Předpokládá se, že s větším množstvím dat bude vhodnější jiné nastavení učení klasifikátorů, či vhodnější jiná volba vstupních parametrů. Všechny skripty vytvořené během této práce jsou psány tak, aby případné další úpravy byly co nejjednodušší.

Strojové učení je obvykle otázkou vývoje několika let. Anotují se tisíce situací a systémy se trénují a stále vyvíjejí. Od systému, který vznikl v rámci této práce nelze očekávat dokonalé výsledky, přesné vyhodnocování manévrů nad jakýmkoli daty. Je to ovšem velký první krok k vytvoření spolehlivého systému.

Reference

- [1] BISHOP, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN 0387310738.
- [2] BRUNA, O. Letecký simulátor Fakulty elektrotechnické ČVUT v Praze. <https://www.facebook.com/simulatorfel>, stav ze 7. 5. 2014.
- [3] CAPKOVA, I. – JUZA, M. – ZIMMERMAN, K. Explorative data analysis of flight behaviour. Technical report, Delft University of Technology, 2002.
- [4] DOHNALOVÁ, L. Systém pro selekci příznaků z EEG signálu. České vysoké učení technické, Fakulta elektrotechnická, 2011. Bakalářská práce.
- [5] DOUDA, K. Vývrtka ještě jednou. <http://www.aeroweb.cz/clanek.asp?ID=962&kategorie=14>, článek z 13. 1. 2008.
- [6] DREYER, T. – OLSON, C. L. – TALSMA, D. FlightGear Flight Simulator. <http://http://www.flightgear.org/>, 2014.
- [7] EHLERT, P. A. – MOUTHAN, Q. M. – ROTHKRANTZ, L. J. Recognising Situations in a Flight Simulator Environment. In *Proceedings of 3rd Int. Conference of Intelligent Games and Simulation (GAME-ON 2002)*, s. 201–205. SCS Publishing House, 2002.
- [8] EHLERT, P. A. – MOUTHAN, Q. M. – ROTHKRANTZ, L. J. A Rule-based and a Probabilistic System for Situation Recognition in a Flight Simulator. In *Proceedings of 4th Int. Conference of Intelligent Games and Simulation (GAME-ON 2003)*, s. 165–169. Eurosis, 2003.
- [9] HANNULA, M. et al. Comparison between artificial neural network and multilinear regression models in an evaluation of cognitive workload in a flight simulator. *Computers in Biology and Medicine*. 2008, 38, s. 1163–1170.
- [10] JÍCHOVÁ, B. A. Analýza biosignálů pro potřeby vyhodnocení stresových faktorů při řešení extrémních situací v ultra lehkých letadlech. Master's thesis, České vysoké učení technické, Fakulta elektrotechnická, 2013.

- [11] KODAD, J. Scénář letu na základě polohové informace. Master's thesis, České vysoké učení technické, Fakulta dopravní, 2014.
- [12] KORDÍK, P. – MOTL, J. Bayesovská klasifikace. České vysoké učení technické, Fakulta informačních technologií, 2011. Podklady k přednáškám předmětu Vytěžování znalostí z dat.
- [13] KOTEK, Z. et al. *Metody rozpoznávání a jejich aplikace*. Academia Praha, 1993. ISBN 8020002979.
- [14] NEZNÁMÝ. Letecké názvosloví. *Naše řeč*, 21, s. 169.
- [15] PEŠKOVÁ, L. Diskriminační analýza. Univerzita Palackého v Olomouci, Přírodovědecká fakulta, 2009. Bakalářská práce.
- [16] PROF. ING. ZDENĚK KOTEK, C. et al. *Adaptivní a učící se systémy*. SNTL - Nakladatelství technické literatury, 1980.
- [17] PŘÍSPĚVOVATELÉ WIKIPEDIE. Elektrokardiogram.
<http://cs.wikipedia.org/wiki/Elektrokardiogram>, stav ze 17. 2. 2014.
- [18] PŘÍSPĚVOVATELÉ WIKIPEDIE. Rozhodovací stromy, .
http://cs.wikipedia.org/wiki/Rozhodovac%C3%AD_stromy, poslední editace 9. 3. 2013.
- [19] PŘÍSPĚVOVATELÉ WIKIPEDIE. Decision tree, .
en.wikipedia.org/wiki/Decision_tree, poslední editace 7. 5. 2014.
- [20] PŘÍSPĚVOVATELÉ WIKIPEDIE. Algoritmus k-nejbližších sousedů, .
http://cs.wikipedia.org/wiki/Algoritmus_k-nejbli%C5%BE%C5%A1%C3%ADch_soused%C5%AF, poslední editace 17. 3. 2013.
- [21] PŘÍSPĚVOVATELÉ WIKIPEDIE. k-nearest neighbors algorithm, .
en.wikipedia.org/wiki/K-nearest_neighbors_algorithm, poslední editace 16. 4. 2014.
- [22] PŘÍSPĚVOVATELÉ WIKIPEDIE. Neuronová síť, .
[cs.wikipedia.org/wiki/Neuronová_síť](http://cs.wikipedia.org/wiki/Neuronov%C3%A1_s%C3%ADt), poslední editace 12. 3. 2013.
- [23] PŘÍSPĚVOVATELÉ WIKIPEDIE. Artificial neural network, .
en.wikipedia.org/wiki/Artificial_neural_network, poslední editace 5. 5. 2014.
- [24] PŘÍSPĚVOVATELÉ WIKIPEDIE. Analýza hlavních komponent.
[http://cs.wikipedia.org/wiki/Analýza_hlavních_komponent](http://cs.wikipedia.org/wiki/Anal%C3%ADza_hlavn%C3%ADch_komponent), stav z 30. 7. 2013.

- [25] PŘISPĚVOVATELÉ WIKIPEDIE. Support vector machines, .
http://cs.wikipedia.org/wiki/Support_vector_machines, poslední editace
17. 2. 2014.
- [26] PŘISPĚVOVATELÉ WIKIPEDIE. Strojové učení, .
cs.wikipedia.org/wiki/Strojové_učení, poslední editace 24. 4. 2014.
- [27] SCHLESINGER, M. I. – HLAVÁČ, V. *Ten Lectures on Statistical and Structural Pattern Recognition*. 24. Kluwer Academic Publishers, 2002. ISBN 140200642.
- [28] THE MATHWORKS, INC. Documentation Center, 2014.
<http://www.mathworks.com/help/>.

Příloha A

Seznam použitých zkratek

- ANN** Umělé neuronové síť
- CD** Kompaktní disk
- CTree** Rozhodovací strom
- ČVUT** České vysoké učení technické
- DA** Diskriminační analýza
- EKG** Elektrokardiogram
- GPS** Global Positioning System
- KNN** Vyhledávání k-nejbližších sousedů
- LDA** Lineární diskriminační analýza
- MLR** Multilineární regrese
- NN** Neuronové síť
- PCA** Principal Component Analysis, analýza hlavních komponent
- PPS** Psychofyzilogický stres
- QDA** Kvadratická diskriminační analýza
- RWY** Runway, vzletová a přistávací dráha
- SVM** Support Vector Machine (podpůrné vektory)

Příloha B

Nácvik manévrů

Tato příloha obsahuje vzorek podkladů, které mají piloti k dispozici pro nácvik různých situací a manévrů.

VÝVRTKA

DŮ před vývrtkami:

1. VYVÁŽENÍ - neutrální
2. BRZDÍCÍ A VZTLAKOVÉ KLAPKY -
- zavřeny, zajištěny
3. KABINA - zavřena, zajištěna
4. VĚTRÁNÍ - zavřeno
5. NOŽNÍ A RUČNÍ ŘÍZENÍ - volnost
6. UPÍNAČI PASY - zapnuty a utaženy
7. VOJNÉ PŘEDMĚTY - zajištěny
8. VÝŠKA NAD TERÉNEM - kontrola
9. VÝHLEDOVÁ ZATAČKA - kontrola prostoru

Chyby při vybírání kluzáku z vývrtky:

1. Pozdě potlačená řídicí páka - kluzák přejde do opačné vývrtky
2. Pozdě nebo málo použito směrové kormidlo - kluzák přejde do spirály
3. Málo potlačená řídicí páka - velké zpoždění ve vybírání
4. Použití křídílek - přechod do spirály nebo vybrání výkluzem
5. Velké potlačení řídicí páky - velká ztráta výšky
6. Nesrovnané nožní řízení - vybrání výkluzem
7. Prudké přitažení řídicí páky - pád na velké rychlosti,
- ztráta rychlosti, pád

VYBÍRÁNÍ VÝVRTKY:

- nožní řízení - PLNĚ VYŠLÁPNOUT proti směru otáčení
- ruční řízení - ŘÍDICÍ PÁKU POTLAČIT
- po zastavení otáčení:
- nožní řízení - NEUTRÁLNÍ POLOHA
- ruční řízení - PLYNULÝM PŘITAŽENÍM vybrat kluzák ze stěmhlavého letu

DŮ



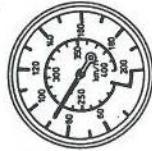
65 km/h



UVEDENÍ DO VÝVRTKY:

- přitažením řídicí páky uvést PŘÍD kluzáku mírně NAD HORIZONT (max. 10°)
- rychlost 65 km/h
- nožní řízení PLNĚ VÝCHYLKA
- ruční řízení ŘÍDICÍ PÁKU POZVOLNA PŘITÁHNOUT

80 - 100 km/h



Chyby ve vývrtce:

1. Povolení řídicí páky - kluzák vybíhá z vývrtky
2. Nesprávné rozdělení pozornosti -
- ztráta orientace v prostoru

Při vybrání nepřekročit 160 km/h



Bnš

VÝVRTKA

Z - 42; Z - 142

Důležité upozornění:

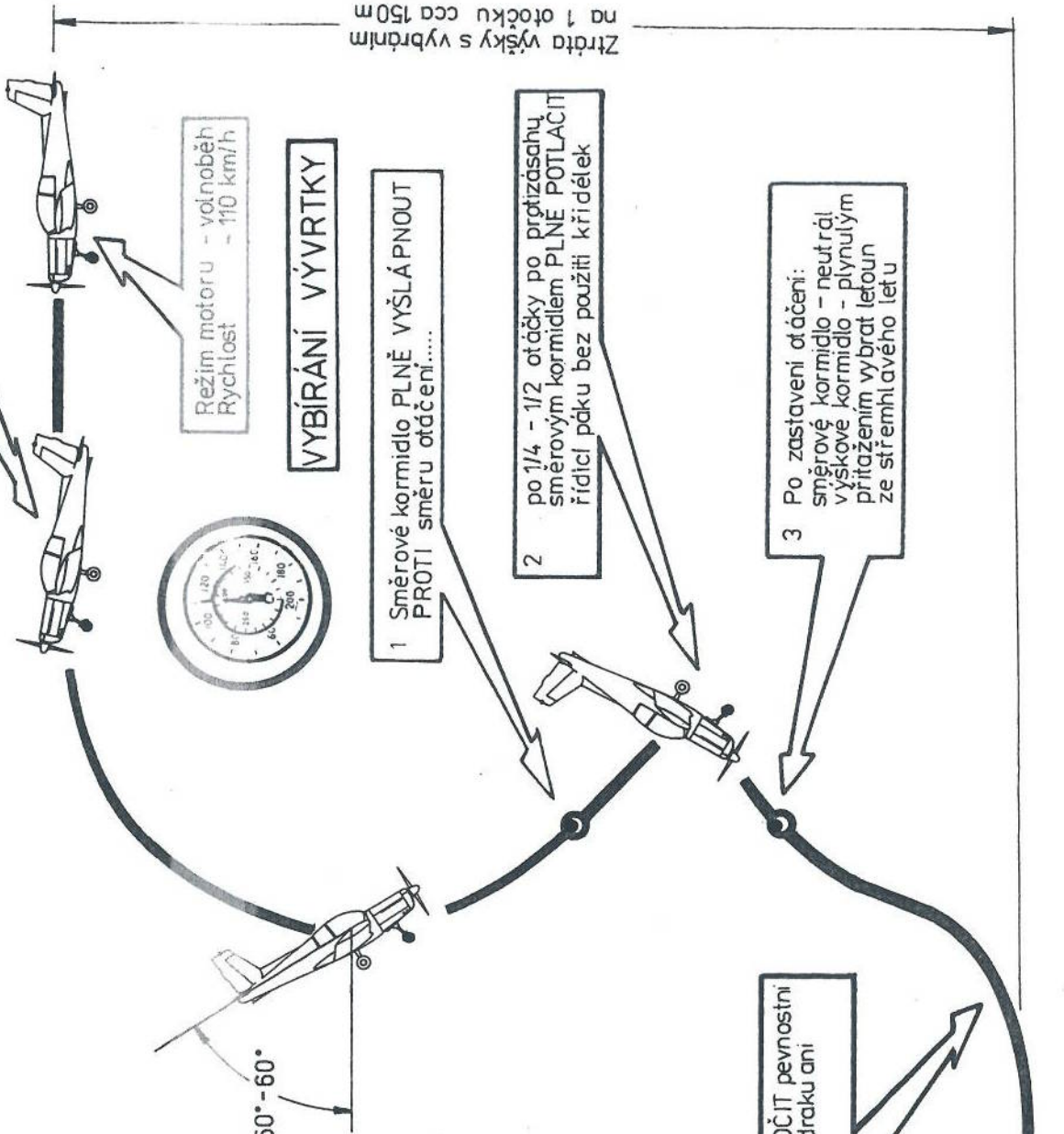
- 1 Při vybrání vývrtky je nutno **ENERGICKY POTLACIT** řídicí páky do **KRAJNÍ** POLOHY bez **POUŽITÍ** KRÍDELEK.
- 2 V případě pádu do **NEÚMYSLNÉ** VÝVRTKY s **OTEVŘENÝMI** VZTLAKOVÝMI KLAPKAMI je nutno při vybrání vývrtky **VZTLAKOVÉ KLAPKY ZAVŘÍT**.
- 3 Při vybrání vývrtky po třech a více otočkách, je **ng. patlačení** řídicí páky zapotřebí **vyvinout VĚTŠÍ SILU** (15 kg a více)

CHYBY PŘI VYBÍRÁNÍ VÝVRTKY

Při nedodržení stanoveného postupu při vybrání vývrtky je nebezpečí, že vývrtka bude vybrána se značným zpožděním. V tomto případě je nutno zachovat následující postup:

- 1 VRÁTIT NOŽNÍ A RUČNÍ ŘÍZENÍ DO POLOHY ODPOVÍDAJÍCÍ VÝVRTCE.
- 2 VYBÍRAT STANOVENÝM ZPŮSOBEM

Uvedení do vývrtky:
-směrové kormidlo **PLNĚ** VÝCHYLKA
ve směr u otáčení ve vývrtce
- výškové kormidlo **PLNĚ** PRITAHNOUT



VYBÍRÁNÍ VÝVRTKY

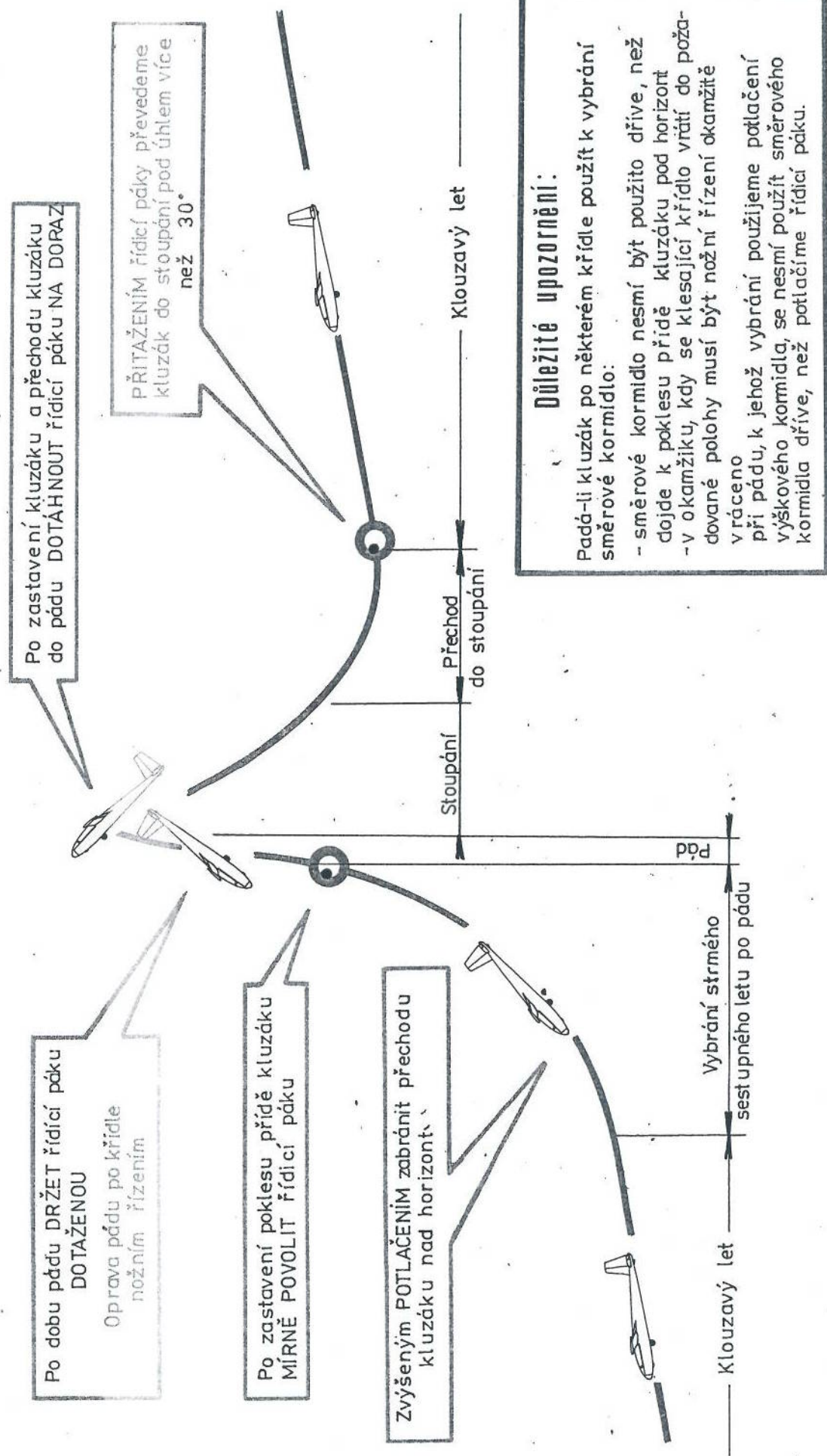
1 Směrové kormidlo **PLNĚ** VYŠLÁPNOUT **PROTI** směr otáčení.....

2 po 1/4 - 1/2 otáčky po průtřásahy směrovým kormidlem **PLNĚ** POTLACIT řídicí páky bez použití křídledek

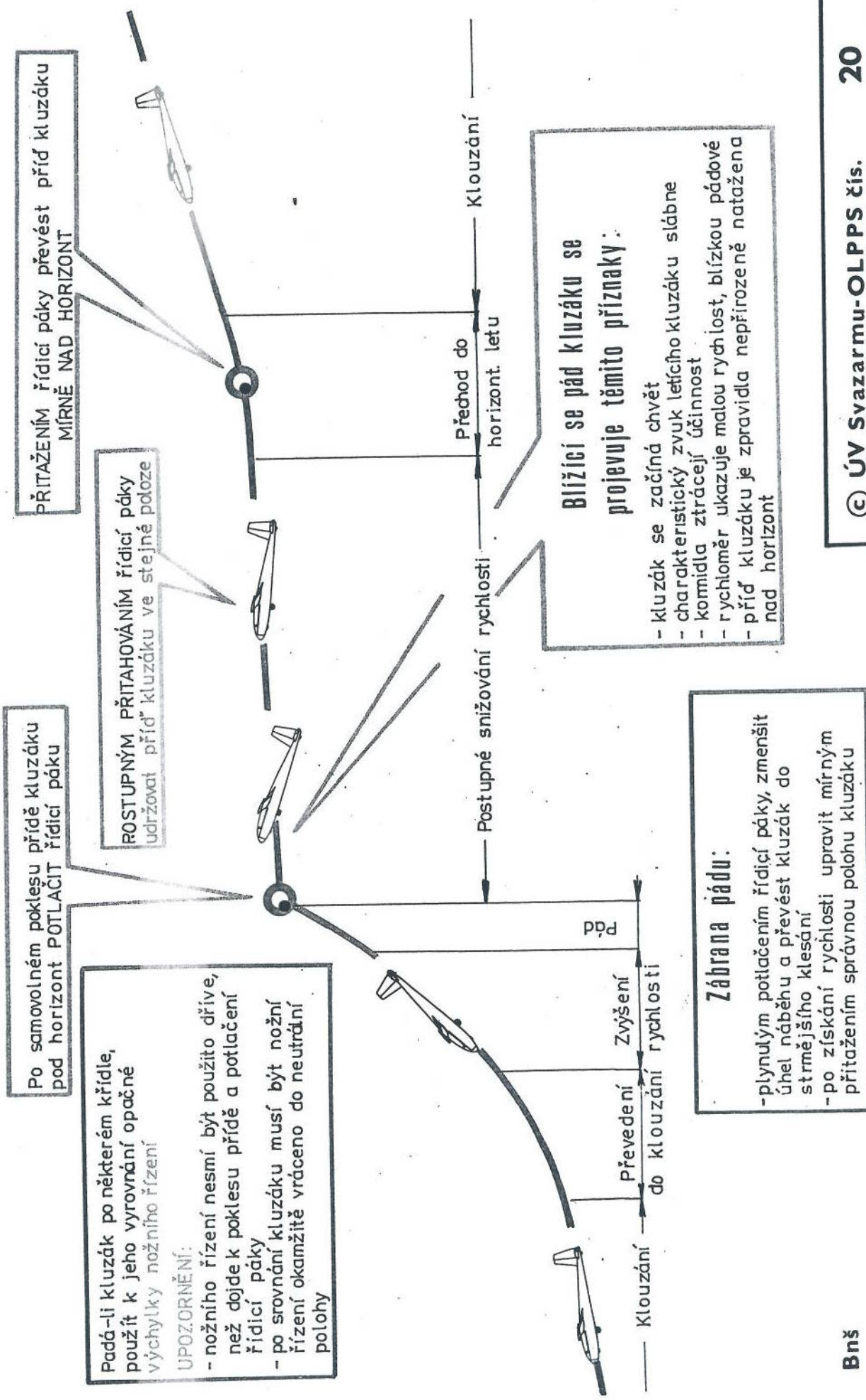
3 Po zastavení otáčení: směrové kormidlo - neutrální výškové kormidlo - plynulým pritažením vybrat letoun ze střemhlavého letu

Při vybrání **NEPŘEKROČIT** pevnostní ani rychlostní **OMEZENÍ** draku ani omezení motoru

OSTRÝ PÁD



MÍRNÝ PÁD



Po samovolném poklesu přide kluzáku pod horizont **POTLAČIT** řídicí páku

Padá-li kluzák po některém křídle, použít k jeho vyrovnání opačné výchylky nožního řízení

UPOZORNĚNÍ:

- nožního řízení nesmí být použito dříve, než dojde k poklesu přídě a potlačení řídicí páky
- po srovnání kluzáku musí být nožní řízení okamžitě vráceno do neutrální polohy

ROSTUPNÝM PŘITAHOVÁNÍM řídicí páky udržovat příd kluzáku ve stejné poloze

PŘITÁŽENÍM řídicí páky převést mírně **NAD HORIZONT**

Postupné snižování rychlosti

Přechod do horizont. letu

Bližící se pád kluzáku se projevuje těmito příznaky:

- kluzák se začíná chvět
- charakteristický zvuk letičiho kluzáku slabne
- komidla ztrácí účinnost
- rychloměr ukazuje malou rychlost, blízkou pádové
- příď kluzáku je zpravidla nepřírozně natažena nad horizont

Zábrana pádu:

- plynulým potlačením řídicí páky, zmenšit úhel náběhu a převést kluzák do strmějšího klesání
- po získání rychlosti upravit mírným přitažením správnou polohu kluzáku

Příloha C

Vzorek surových txt dat

Tento vzorek je pouze malou ukázkovou částí jednoho textového souboru.

0.147617,-7.002069,-132.394119,-79.212456,153.055359,-20.274254,3.854839,7.002069
0.146363,-8.182196,-131.665161,-77.823135,152.699051,-11.254791,3.928092,8.182196
0.146177,-8.304525,-131.522217,-77.688835,152.596802,-10.076100,3.912138,8.304525
0.146095,-8.355494,-131.455750,-77.639023,152.545120,-9.641798,3.907316,8.355494
0.146079,-8.365750,-131.441986,-77.629486,152.534195,-9.559297,3.906443,8.365750
0.146029,-8.396682,-131.399826,-77.601715,152.500244,-9.320765,3.903960,8.396682
0.145978,-8.427956,-131.356293,-77.575233,152.464615,-9.096079,3.901538,8.427956
0.145893,-8.480949,-131.280838,-77.533890,152.401718,-8.753227,3.897353,8.480949
0.145858,-8.502502,-131.249664,-77.518333,152.375366,-8.627438,3.895502,8.502502
0.145823,-8.524271,-131.217926,-77.503334,152.348373,-8.508243,3.893496,8.524271
0.145771,-8.557344,-131.169312,-77.481865,152.306717,-8.341960,3.890110,8.557344
0.145682,-8.613566,-131.085709,-77.448769,152.234375,-8.098516,3.883213,8.613566
0.145664,-8.624987,-131.068604,-77.442551,152.219482,-8.054940,3.881597,8.624987
0.145610,-8.659538,-131.016617,-77.424667,152.174042,-7.934410,3.876266,8.659538
0.145556,-8.694573,-130.963562,-77.407959,152.127472,-7.829301,3.870073,8.694573
0.145465,-8.753875,-130.873001,-77.382576,152.047501,-7.688248,3.857725,8.753875
0.145446,-8.765871,-130.854568,-77.377876,152.031174,-7.665156,3.854919,8.765871
0.145391,-8.802041,-130.798737,-77.364449,151.981583,-7.605975,3.845862,8.802041
0.145336,-8.838516,-130.742065,-77.352081,151.931091,-7.561929,3.835748,8.838516
0.145243,-8.899744,-130.646011,-77.333656,151.845108,-7.521553,3.816595,8.899744
0.145225,-8.912049,-130.626556,-77.330299,151.827652,-7.518380,3.812406,8.912049
0.145169,-8.948975,-130.567825,-77.320808,151.774796,-7.518414,3.799199,8.948975
0.145113,-8.985950,-130.508499,-77.312225,151.721222,-7.532655,3.784967,8.985950
0.145020,-9.047409,-130.408569,-77.299744,151.630493,-7.586972,3.759175,9.047409
0.145001,-9.059669,-130.388428,-77.297523,151.612137,-7.602323,3.753706,9.059669
0.144945,-9.096299,-130.326874,-77.293251,151.556747,-7.659700,3.736766,9.096299
0.144908,-9.120590,-130.285782,-77.290527,151.519455,-7.705532,3.724993,9.120590
0.144870,-9.144750,-130.244583,-77.288086,151.481903,-7.757469,3.712876,9.144750
0.144796,-9.192601,-130.161972,-77.283997,151.406006,-7.879079,3.687718,9.192601
0.144777,-9.204463,-130.141266,-77.283134,151.386871,-7.913076,3.681246,9.204463
0.144722,-9.239701,-130.079178,-77.280846,151.329147,-8.023303,3.661524,9.239701
0.144666,-9.274477,-130.017044,-77.279030,151.270905,-8.145554,3.641312,9.274477
0.144574,-9.331171,-129.913681,-77.276833,151.172760,-8.374342,3.606914,9.331171
0.144556,-9.342324,-129.893036,-77.276512,151.152985,-8.423674,3.599941,9.342324
0.144501,-9.375303,-129.831314,-77.275688,151.093353,-8.578324,3.578984,9.375303
0.144465,-9.396930,-129.790283,-77.275276,151.053360,-8.686789,3.564960,9.396930

0.144428,-9.418247,-129.749374,-77.274948,151.013184,-8.799311,3.550939,9.418247
0.144356,-9.459911,-129.668030,-77.274452,150.932343,-9.035742,3.523023,9.459911
0.144320,-9.480242,-129.627625,-77.274246,150.891678,-9.159253,3.509185,9.480242
0.144285,-9.500226,-129.587433,-77.274055,150.850876,-9.286027,3.495462,9.500226
0.144249,-9.519858,-129.547440,-77.273849,150.809921,-9.415858,3.481881,9.519858
0.144214,-9.539132,-129.507690,-77.273621,150.768845,-9.548543,3.468469,9.539132
0.144144,-9.576586,-129.428940,-77.273010,150.686310,-9.821648,3.442246,9.576586
0.144109,-9.594756,-129.389969,-77.272598,150.644882,-9.961652,3.429482,9.594756
0.144074,-9.612552,-129.351288,-77.272087,150.603363,-10.103680,3.416979,9.612552
0.144040,-9.629969,-129.312927,-77.271461,150.561768,-10.247525,3.404758,9.629969
0.144005,-9.647006,-129.274872,-77.270714,150.520081,-10.392979,3.392838,9.647006
0.143938,-9.679934,-129.199768,-77.268753,150.436539,-10.687882,3.369974,9.679934
0.143904,-9.695823,-129.162735,-77.267517,150.394714,-10.836920,3.359063,9.695823
0.143870,-9.711328,-129.126068,-77.266090,150.352844,-10.986742,3.348520,9.711328
0.143837,-9.726450,-129.089767,-77.264450,150.310974,-11.137145,3.338358,9.726450
0.143804,-9.741188,-129.053864,-77.262596,150.269104,-11.287927,3.328589,9.741188
0.143739,-9.769520,-128.983231,-77.258156,150.185410,-11.589833,3.310275,9.769520
0.143706,-9.783115,-128.948517,-77.255547,150.143616,-11.740563,3.301747,9.783115
0.143674,-9.796333,-128.914215,-77.252670,150.101883,-11.890888,3.293650,9.796333
0.143642,-9.809175,-128.880341,-77.249496,150.060226,-12.040617,3.285989,9.809175
0.143610,-9.821641,-128.846893,-77.246025,150.018661,-12.189563,3.278770,9.821641
0.143546,-9.845407,-128.780548,-77.237755,149.935013,-12.484369,3.265681,9.845407
0.143514,-9.856691,-128.747391,-77.232803,149.892670,-12.629873,3.259908,9.856691
0.143482,-9.867589,-128.714264,-77.227303,149.850006,-12.773880,3.254730,9.867589
0.143449,-9.878111,-128.681183,-77.221245,149.807053,-12.916219,3.250197,9.878111
0.143416,-9.888267,-128.648132,-77.214622,149.763824,-13.056726,3.246349,9.888267
0.143350,-9.907543,-128.582214,-77.199661,149.676651,-13.331598,3.240834,9.907543
0.143317,-9.916694,-128.549347,-77.191315,149.632751,-13.465655,3.239208,9.916694
0.143283,-9.925542,-128.516556,-77.182373,149.588669,-13.597259,3.238353,9.925542
0.143249,-9.934104,-128.483841,-77.172844,149.544418,-13.726264,3.238272,9.934104
0.143215,-9.942397,-128.451202,-77.162720,149.500046,-13.852534,3.238964,9.942397
0.143147,-9.958241,-128.386185,-77.140686,149.410980,-14.096331,3.242631,9.958241
0.143112,-9.965826,-128.353821,-77.128769,149.366348,-14.213604,3.245577,9.965826
0.143077,-9.973207,-128.321548,-77.116249,149.321671,-14.327633,3.249239,9.973207
0.143042,-9.980395,-128.289383,-77.103134,149.276978,-14.438303,3.253593,9.980395
0.143007,-9.987408,-128.257339,-77.089417,149.232300,-14.545506,3.258612,9.987408
0.142937,-10.000949,-128.193558,-77.060188,149.143051,-14.749104,3.270524,10.000949
0.142901,-10.007498,-128.161865,-77.044685,149.098526,-14.845307,3.277351,10.007498
0.142866,-10.013913,-128.130280,-77.028595,149.054108,-14.937664,3.284714,10.013913
0.142830,-10.020200,-128.098831,-77.011917,149.009811,-15.026092,3.292576,10.020200
0.142794,-10.026365,-128.067505,-76.994659,148.965668,-15.110517,3.300900,10.026365
0.142722,-10.038348,-128.005280,-76.958420,148.877869,-15.267082,3.318787,10.038348
0.142686,-10.044173,-127.974388,-76.939453,148.834274,-15.339101,3.328276,10.044173
0.142649,-10.049888,-127.943642,-76.919930,148.790909,-15.406872,3.338078,10.049888
0.142613,-10.055492,-127.913040,-76.899857,148.747787,-15.470348,3.348158,10.055492
0.142576,-10.060986,-127.882599,-76.879242,148.704926,-15.529490,3.358480,10.060986

Příloha D

Obsah přiloženého CD

```
data - Hlavní adresář získaných vzorků dat rozdělený na referenční a testovací data
├── referenční_data
│   ├── test_11_11_pad_ostry.txt
│   ├── test_11_11_spirala.txt
│   ├── test_11_11_vyvrtka.txt
│   ├── test_11_11_vyvrtka2.txt
│   ├── test_11_13_ostry_pad.txt
│   ├── test_11_13_spirala.txt
│   ├── test_11_13_vyvrtka.txt
│   ├── test_11_13_vyvrtka2.txt
│   ├── test_11_18_spirala.txt
│   ├── test_11_18_vyvrtka.txt
│   ├── test_11_8_ostry_pad.txt
│   ├── test_11_8_piper_vyvrtka3.txt
│   ├── test_11_8_piper_vyvrtka4.txt
│   └── test_11_8_vyvrtka2.txt
├── testovací_data
│   ├── log_LKPR_02.txt
│   ├── log_LKPR_03.txt
│   ├── log_LKPR_04.txt
│   ├── log_LKRK_01.txt
│   ├── log_LKRK_02.txt
│   ├── log_LKRK_03.txt
│   ├── log_LKRK_04.txt
│   ├── test_11_11_pad_mirny.txt
│   ├── test_11_13_mirny_pad.txt
│   ├── test_11_18_mirny_pad.txt
│   ├── test_11_8_spirala.txt
│   └── test_11_8_vyvrtka1.txt
├── EvaPomichalova-DP-2014.pdf - Kompletní diplomová práce
├── uživatelské_skripty - Uživatelské skripty pro použití v programu Matlab R2013b a vyšší
│   ├── teach_knn.m
│   └── use_knn.m
└── 4 directories, 31 files
```

Obrázek D.1: Obsah přiloženého CD