

# **Vyvažování nelineárního matematického modelu letounu aplikací evolučních algoritmů**

Trimming of nonlinear mathematic aircraft model by application of evolutionary algorithms

Bc. Lukáš Tomšů

---

Diplomová práce  
2007



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---



\*\*\* nescannované zadání str. 1 \*\*\*

\*\*\* nescannované zadání str. 2 \*\*\*

## **ABSTRAKT**

Tato práce se zabývá aplikací evolučních algoritmů při vyvažování nelineárního matematického modelu letounu. První část je zaměřena na teoretický úvod k problému. Stručný popis řídicích prvků letounu obsahuje základní údaje o reálných dopadech simulací. V další části jsou prezentovány evoluční algoritmy, jejich rozdělení a princip práce. Následuje rozbor matematického modelu letounu, s nímž se dále pracuje. V praktické části jsou pak demonstrovány výsledky simulací.

**Klíčová slova:** evoluční algoritmy, matematický model, letoun, aerodynamika

## **ABSTRACT**

This work deals with application of evolutionary algorithms on trimming of aircraft mathematic model. First section is focused on theoretical background of this problem. Brief description of aircraft control mechanisms contains basic facts about real impacts of simulations. There are presented evolutionary algorithms in next section, their system and principles of work. Analysis of aircraft mathematical model, with witch is worked, follows. Outcomes of simulations are demonstrated in practical part.

**Keywords:** evolutionary algorithms, mathematical model, aircraft, aerodynamics

Děkuji doc. Ing. Ivanu Zelinkovi, Ph.D.za vedení mé diplomové práce, Danielu Kurfüstovi a dalším zaměstnanců firmy Hexagon za rychlou a přesnou spolupráci.

Prohlašuji, že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně  
.....

.....

Podpis diplomanta

## OBSAH

ÚVOD.....	9
<b>I TEORETICKÁ ČÁST.....</b>	<b>10</b>
<b>1DEFINICE PROBLÉMU.....</b>	<b>11</b>
1.1POUŽITÉ METODY.....	11
<b>2MECHANIKA LETU.....</b>	<b>12</b>
2.1OSY POHYBU.....	12
2.2HLAVNÍ ŘÍDÍCÍ PLOCHY.....	13
2.2.1Křídélka.....	13
2.2.2Výškové kormidlo.....	14
2.2.3Směrové kormidlo .....	14
2.3DALŠÍ DŮLEŽITÉ POJMY.....	15
2.3.1Úhel náběhu.....	15
2.3.2Úhel vybočení.....	17
<b>3EVOLUČNÍ ALGORITMY.....</b>	<b>18</b>
3.1GENETICKÉ ALGORITMY.....	19
3.2DIFERENCIÁLNÍ EVOLUCE.....	20
3.2.1Struktura algoritmu DE.....	20
3.3EVOLUČNÍ ALGORITMUS SOMA.....	23
3.3.1Struktura algoritmu SOMA.....	23
3.3.2Strategie algoritmu SOMA.....	25
<b>4POHYBOVÉ ROVNICE LETADLA.....</b>	<b>26</b>
4.1SOUŘADNÉ SOUSTAVY.....	26
4.2POHYBOVÉ ROVNICE.....	28
4.2.1Použitá integrační metoda.....	32
4.2.2Stavový vektor.....	32
4.3TRIM.....	34
<b>5ÚVOD K PRAKTICKÉ ČÁSTI .....</b>	<b>36</b>
5.1ZÁKLADNÍ BODY POSTUPU.....	36
5.2VYVAŽOVACÍ VEKTOR.....	36
5.3ÚČELOVÁ FUNKCE A DERIVACE STAVOVÉHO VEKTORU.....	38
5.3.1Složky derivace stavového vektoru.....	39
<b>II PRAKTICKÁ ČÁST .....</b>	<b>39</b>
<b>6GRAFY – ŘEZY ÚČELOVOU FUNKCÍ .....</b>	<b>40</b>
<b>7SIMULAČNÍ POKUSY .....</b>	<b>42</b>
7.1SOMA: ATAA 20 m/s.....	42
7.2SOMA: ATAA 40 m/s.....	47
7.3TRIM 40 m/s.....	51
<b>ZÁVĚR.....</b>	<b>57</b>
<b>ZÁVĚR V ANGLIČTINĚ .....</b>	<b>58</b>
<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>59</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>60</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>61</b>

SEZNAM TABULEK.....	62
SEZNAM PŘÍLOH.....	63



## ÚVOD

Evoluční algoritmy, jejichž používání v posledních letech zažívá nebývalý růst, nalézají uplatnění v nejrozličnějších oborech lidské činnosti. Kdekoliv je potřeba zjistit optimální kombinaci libovolných parametrů, lze evoluční algoritmy aplikovat a získat tak mocný nástroj pro dříve obtížně řešitelné úlohy. Tam kde klasické postupy selhávají a standardní optimalizační algoritmy vrací nedostatečně kvalitní, nebo dokonce zcela nevyhovující výsledky, poslouží evoluční algoritmy svou robustností. Jednou z jejich velkých výhod je vysoká pravděpodobnost nalezení globálního extrému i v případě výrazně multimodálních funkcí (funkcí s mnoha lokálními extrémy). Jak již sám název napovídá, jsou evoluční algoritmy principiálně inspirovány evolučním procesem v přírodě – na základě jednoduchých pravidel vzniká komplikované chování, v tomto případě jde o upřednostňování a přežití schopnějšího jedince, schopného lépe se přizpůsobit okolním podmínkám. Podmínky jsou simulovány účelovou funkcí a vhodnost jedince je dána hodnotou této účelové funkce.

Letectví je oborem, ve kterém mohou evoluční algoritmy pomoci vyřešit nejen obtížný problém. Ať již ve fázi návrhu, nebo při samotném ovládní je možné použitím evolučních algoritmů dosáhnout zajímavých výsledků.

Cílem této diplomové práce bylo aplikovat evoluční algoritmy na vyvažování nelineárního matematického modelu letounu ve spolupráci s firmou Hexagon, s.r.o. Při tomto procesu bylo třeba nastavit řadu vzájemně propojených veličin tak, aby bylo dosaženo požadovaného vyváženého stavu. Jako testovací algoritmus byl zvolen SOMA.

## **I. TEORETICKÁ ČÁST**

## 1 DEFINICE PROBLÉMU

Cílem této diplomové práce byla aplikace evolučních algoritmů při vyvažování nelineárního matematického modelu ve spolupráci s brněnskou firmou Hexagon. Základ práce spočíval v nalezení co možná nejvíce se blížící optimální kombinace nastavení parametrů (a později i formy účelové funkce) při řízení letounu tak, aby bylo dosaženo vyváženého letu (viz 4.3).

### 1.1 Použité metody

Při dosavadním procesu vyvažování byl používán simplexový algoritmus, který však neposkytoval uspokojivé výsledky. Jako možná alternativa byly proto zvoleny evoluční algoritmy, které svou robustností a úspěšností při aplikaci na jiné optimalizační problémy slibovaly možné zlepšení.

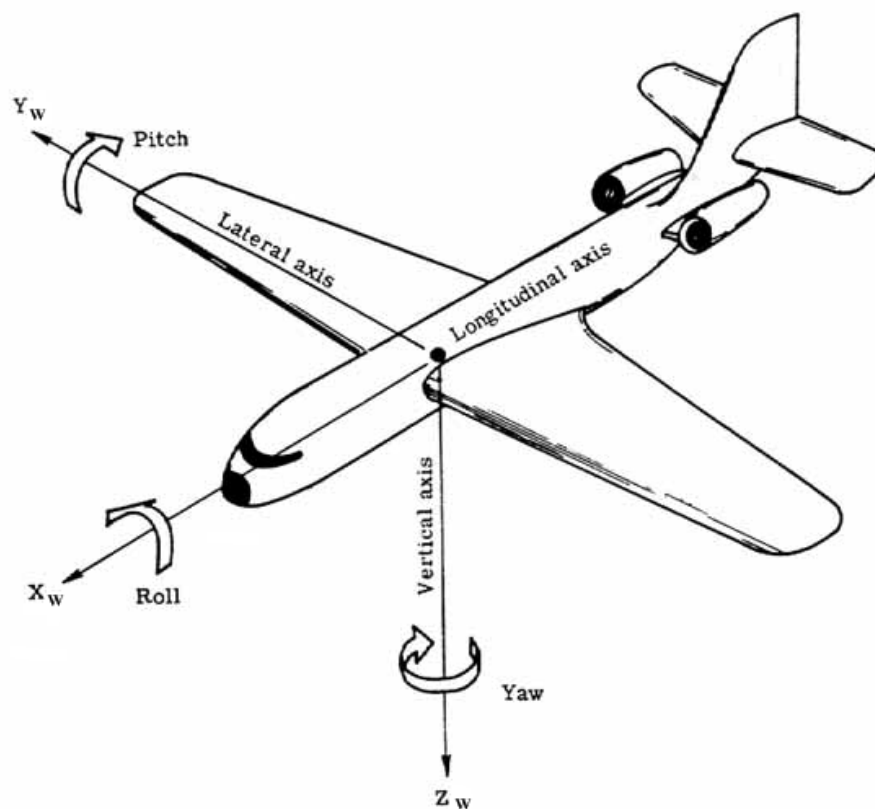
Testovacím algoritmem byl SOMA, zvolený vzhledem k mnohaletým zkušenostem s jeho prací na našem ústavu a dobré znalosti principů jeho chodu, také zejména pro jeho ověřenou velkou robustnost a schopnost úspěšně řešit složité optimalizační úlohy a velkou variabilitu, co se týče možnosti použití různých strategií samotného algoritmu a možnosti nastavení jednotlivých parametrů. Poměrně velkou výhodou při práci na těchto složitých úlohách je též uživatelská jednoduchost.

## 2 MECHANIKA LETU

Tato část popisuje, jak síly a momenty vzniklé pohybem letadla působí na jeho pohyb. Zevrubně se také zbývá prvky letového ovládání a věnuje se konkrétním prostředkům, které pilotovi umožňují let kontrolovat.

### 2.1 Osy pohybu

Letoun může volně rotovat kolem tří os, které jsou navzájem kolmé a protínají se v těžišti. Aby pilot mohl ovlivnit směr a pozici letounu, musí být schopen ovládat rotaci kolem každé z nich.



Obr. 1 – Rotace kolem tří os

Vertikální osa (Vertical Axis) prochází letounem shora dolů. Rotace kolem této osy se nazývá **zatačení** (Yaw). Zatačení mění směr, kam míří nos letounu - doleva nebo doprava. Primární ovládání zatačení je uskutečňováno prostřednictvím směrovky, ale je ovlivňováno i křídélky.

Podélná osa (Longitudal axis) prochází letounem od nosu po ocas. Rotace kolem této osy se nazývá **klonění**. Náklon mění orientaci křídel vzhledem k dolů směřující gravitační síle.

Pilot mění úhel náklonu zvýšením vztlaku na jednom a snížením na druhém křídle. Primární ovládání náklonu je prováděno pomocí křidélek, jistý vliv má i směrovka.

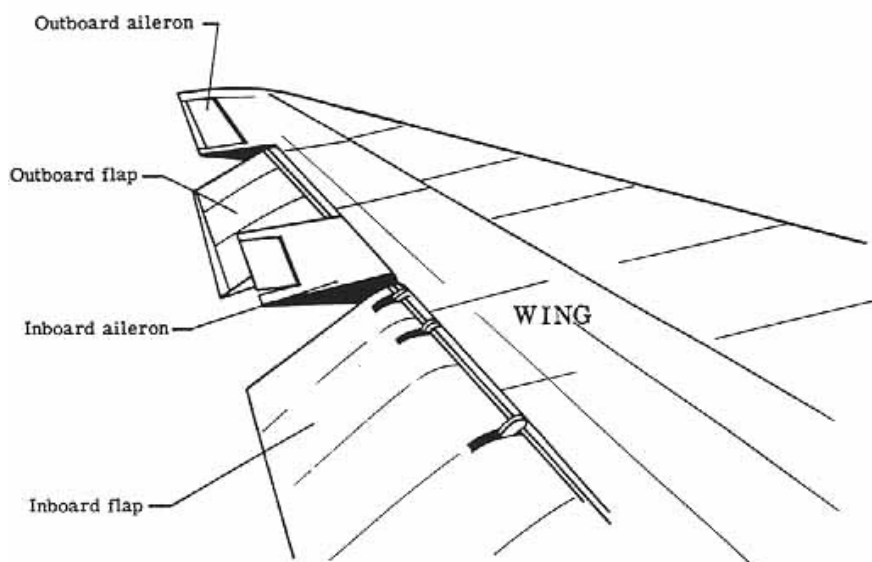
Příčná osa (Lateral axis) prochází letounem od konce jedno křídla do konce druhého. Rotace kolem této osy se nazývá **klopení**. Klopení mění vertikálně směr, kam míří nos letounu. Primární ovládání klopení je řízeno výškovým kormidlem.

## 2.2 Hlavní řídicí plochy

Hlavní řídicí plochy jsou pohyblivé části letounu, pomocí kterých lze vychýlit proud vzduchu procházející kolem nich. Toto přesměrování proudícího vzduchu má za následek vznik nevyvážené síly, jejímž prostřednictvím je zapříčiněna rotace kolem příslušné osy.

### 2.2.1 Křidélka

Křidélka jsou pohyblivé řídicí plochy nacházející se na odtokové hraně křídla.



Obr. 2 – Křidélka

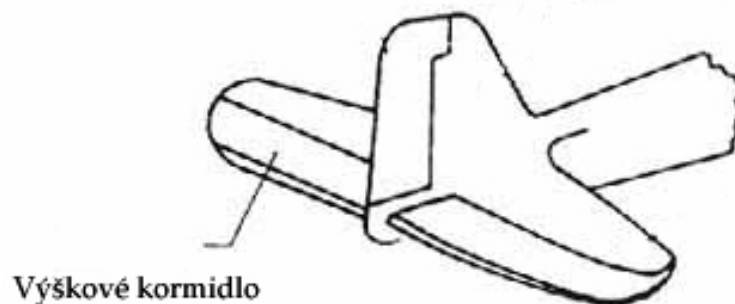
Jsou používány k ovládní klonění. Dvě křidélka jsou propojena tak, že když jde jedno nahoru, druhé se vykloní dolů: křidélko směřující dolů zvýší na svém křídle vztlak, kdežto nahoru směřující vztlak naopak snižuje. Tímto rozdílem vzniká klonící moment kolem podélné osy.

Nežádoucím efektem křidélek je protikladný zatáčivý moment v opačném směru k otáčení produkovaném křidélky, tzn. při použití křidélek pro naklonění doleva vznikne zatáčivý pohyb doprava. To je způsobeno zvýšeným indukovaným tahem kvůli většímu efektivnímu klenutí na křídle se směrem dolů sklopeným křidélkem a opačným efektem na křídle druhém.

Protikladný zatáčivý moment je také způsoben křídlem vně zatáčky, které se pohybuje rychleji než křídlo uvnitř a působí na něj díky tomu větší vztlak a tah.

### 2.2.2 Výškové kormidlo

Výšková kormidla (výškovky) jsou řídicí plochy většinou umístěné v zadní části, sloužící k ovládní letounu změnou klopení (tj. rotací kolem příčné osy) a tím pádem také k nastavení úhlu náběhu křídla.



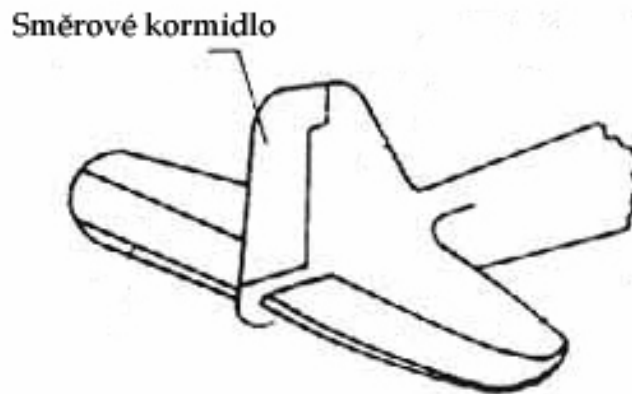
Obr. 3 – Výškové kormidlo

Zadní křídlo, ke kterému je výškovka připevněna, má na křídlo opačný vliv, což znamená, že vytváří dolů směřující tlak, který vyrovnává rotační moment vznikající pohybem těžiště kolem aerodynamického středu<sup>1</sup> letounu. Výškovka snižuje nebo zvyšuje vztlak zadního křídla.

### 2.2.3 Směrové kormidlo

Směrové kormidlo (směrovka) je řídicí plocha, umístěná většinou na svislé ocasní ploše, která pilotovi umožňuje ovládat zatáčení letounu, tj. měnit horizontální směr, do kterého míří nos.

<sup>1</sup> Aerodynamický střed: Bod profilu křídla, ve kterém je klopivý moment vytvářený profilem konstantní nezávisle na úhlu náběhu



Obr. 4 – Směrové kormidlo

Vychýlení směrovky je ovládáno pohybem nožních pedálů pilotem. Vychýlením směrovky vznikne zatáčivý moment, díky němuž se letoun natočí bokem k nabíhajícímu proudu vzduchu. Bočním ofukováním vznikne boční síla, která začne posunovat letadlem. Zatačením kolem kolmé osy se urychluje vnější polovina křídla a zpomaluje vnitřní polovina, což vede ke zvýšení, resp. Ke snížení vzlaku na těchto polovinách a letadlo se začne navíc ještě naklánět.

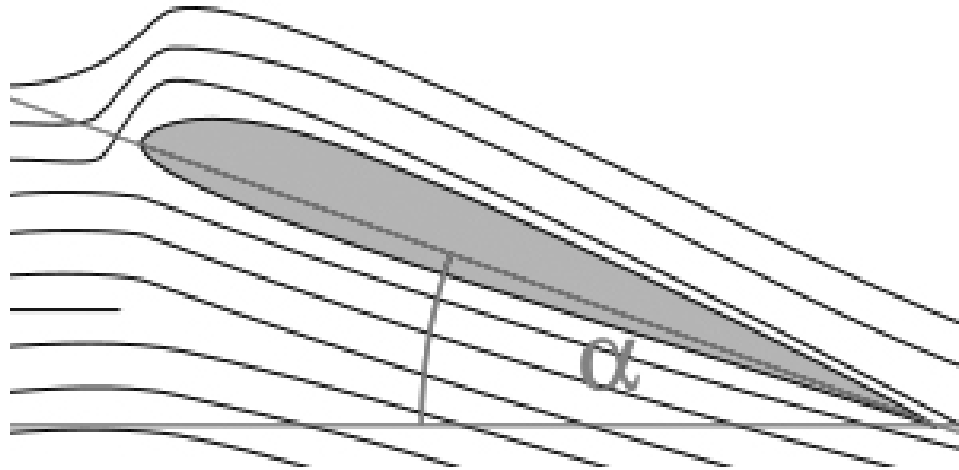
Klonění a zatačení spolu úzce souvisí a nelze provádět ani nejjednodušší obraty bez součinnosti křidélek+výškovky a směrovky.

## 2.3 Další důležité pojmy

Základní aerodynamické prvky.

### 2.3.1 Úhel náběhu

Úhel náběhu je termín používaný v aerodynamice pro úhel mezi čarou profilu a směrem vzdušného proudu, tzn. směrem, ve kterém se letoun právě pohybuje.



Obr. 5 – Úhel náběhu

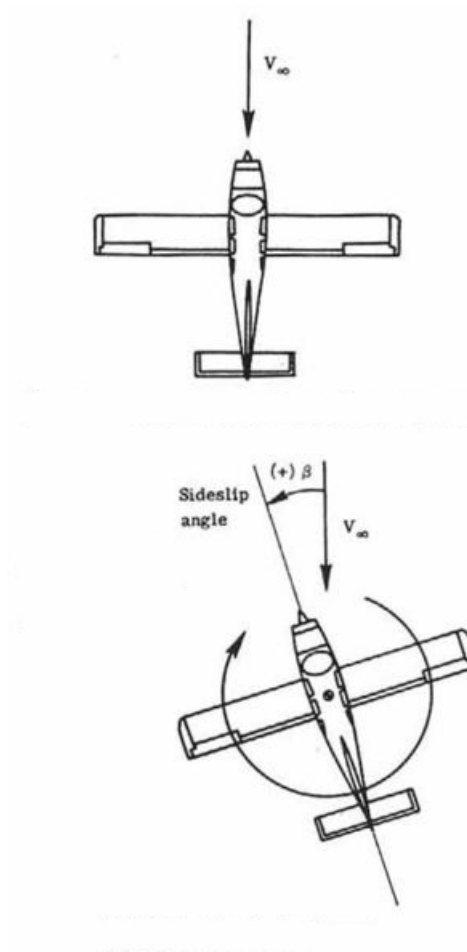
Lze jej také popsat jako úhle mezi tím, kam křídlo míří a tím, kam se ve skutečnosti pohybuje. Velikost vztlaku generovaného křídlem se přímo vztahuje k úhlu náběhu – čím větší úhel, tím větší vztlaková síla. Toto platí až do určitého bodu, od kterého již dochází k odtržení vzdušného proudění od profilu křídla a prudkému poklesu vztlakové síly.



Kritický úhel je obvykle kolem  $15^\circ$  pro většinu profilů křídel, u moderních (bojových) letounů se kritické úhly pohybují v rozmezí  $20^\circ$  až  $45^\circ$ .

### 2.3.2 Úhel vybočení

Úhel vybočení popisuje odchylku podélné osy od směru relativního větru<sup>2</sup>. V letecké dynamice má označení  $\beta$ .



Obr. 6 – Úhel vybočení

<sup>2</sup> Relativní vítr: rychlost proudu nad profilem křídla, tj. vektorový rozdíl mezi skutečným (absolutním) větrem a rychlostí letounu.

### 3 EVOLUČNÍ ALGORITMY

Evoluční algoritmy vznikly jako matematická analogie k přirozeným procesům probíhajících v přírodě během evoluce, která pokud probíhá zcela náhodným výběrem zajistí, že přežijí jen ti jedinci, kteří jsou schopni odolat v souboji s přírodními vlivy. Dochází tak k přirozenému šlechtění populace jedinců, kdy se mění vlastnosti jedince tak, aby lépe vyhovovaly přírodním podmínkám. Toto se stalo základním principem evolučních algoritmů, které z prvotní náhodně vygenerované populace jedinců tvoří nové generace jedinců s lepšími vlastnostmi, popřípadě mění jejich parametry tak, aby hodnota účelové funkce dosáhla neoptimálnější hodnoty. Zpravidla tedy hledáme extrém funkce, nejčastěji minimum, na  $n$ -rozměrné hyperploše. Problematiku optimalizace účelové funkce můžeme zapsat ve tvaru:

$$\min(f_{\text{cost}}(x)) \quad (1)$$

pomocí optimálních hodnot argumentů:

$$X = (x_1, \dots, x_D) \quad (2)$$

kde  $X$  představuje vektor složený z  $D$  parametrů účelové funkce, tyto mají svá omezení:

$$x_j^{(Lo)} \leq x_j \leq x_j^{(Hi)} \quad j = 1, \dots, D \quad (3)$$

Díky svým vlastnostem se evoluční algoritmy stávají zcela univerzálním nástrojem při optimalizaci nejrůznějších procesů. Nejčastější uplatnění nalézají při hledání optimálních parametrů technologických procesů, k učení neuronových sítí, stejně tak jako při hledání globálního minima u složitých funkcí s mnoha lokálními extrémy, nebo k nalezení optimálního modelu řízení. Díky své univerzálnosti mohou při optimalizaci celé řady problematik vylepšit dřívější řešení považované za optimální nebo zcela nahradit expertovy apriorní vědomosti a tedy nalézt optimum automaticky bez přítomnosti experta. Vyznačují se relativní jednoduchostí softwarové implementace a nevyžadují složité programátorské znalosti, některé z nich jsou schopné plnohodnotné práce i v tabulkových procesorech a tak mohou být kdekoli v praxi vždy rychle a snadno dostupné. Jednotlivé evoluční algoritmy se od sebe liší zejména způsobem vytváření nové populace jedinců a v současné době známe celkem tři základní druhy evolučních algoritmů:

1. Genetické algoritmy (GA)
2. Diferenciální evoluce (DE)

### 3. SOMA

Genetické algoritmy pracují na principu napodobení křížení genetické informace uložené v DNA, tak jak se to děje v živých organismech. Jsou již ale poměrně zastaralé a jejich používání přinášelo pro uživatele spoustu obtíží, byť dávaly uspokojivé výsledky. Byly ovšem překonány poměrně novými výkonnějšími a robustnějšími algoritmy jako DE a SOMA. Strukturou svého algoritmu se částečně vzdalují původní myšlenky napodobit mechanismus křížení a přizpůsobování organismů, nicméně i u nich probíhá proces křížení a vytváření nové populace v případě diferenciální evoluce a zdokonalování jedné populace jedinců v případě algoritmu SOMA. Jejich struktura přinesla spoustu výhod a zjednodušení oproti genetickým algoritmům a otevřely se tak nové možnosti na poli optimalizace.

#### 3.1 Genetické algoritmy

Genetické algoritmy vznikly na počátku 60. let jako analogie k biologické genetice a teorii evoluce probíhající v přírodě. Z biologie plyne, že všechny základní informace o každém živém jedinci jsou uloženy v DNA (deoxyribonukleové kyselině). DNA je dlouhý molekulární řetězec tvořený sekvencí složenou ze čtyř druhů složek (bází), jejichž uspořádání definuje genetický kód daného jedince.

Genetické algoritmy definují tyto základní pojmy:

- Chromozom - část DNA obsahující větší množství genů
- Gen - část DNA složená ze sekvence alel určující jednu vlastnost jedince,
- Alela – Označení pro množinu hodnot, jež mohou geny nabývat. V přírodě jsou to čtyři báze, u genetických algoritmů se jedná o hodnoty 0 nebo 1.

Kompletní genetický popis organismu je označován pojmem genotyp. Kompletní fyziologický popis vlastností organismu se označuje jako fenotyp. U genetických algoritmů se pod pojmem genotyp rozumí sekvence nul a jedniček. Fenotyp potom udává velikost konkrétních parametrů definovaných genotypem.

V přírodě existují dva druhy rozmnožování - bezpohlavní a pohlavní. V prvním případě nedochází ke křížení, čímž se omezuje schopnost vytvářet jedince s novými vlastnostmi, jediný způsob vzniku odlišného jedince je mutace genetického kódu rodiče. U pohlavního rozmnožování se uplatňují dva vlivy při vzniku nových jedinců, jsou to mutace a křížení. Mutace je zcela náhodná změna některého genu. Křížení vzniká při vytváření nové DNA jedince z DNA rodičů. Na stejném principu jsou založeny i genetické algoritmy. Vlastní algoritmus je v podstatě cyklus, v němž prvotní náhodně vygenerovaná populace jedinců se mezi sebou kříží a vytváří novou populaci jedinců, kteří jsou použiti jako rodiče v dalším

cyklu. Pro výběr jedinců se používají předem zvolená kritéria. Při křížení a mutaci se potom využívá generátoru náhodných čísel pro vygenerování způsobu dělení genotypů mezi rodiči, a pro určení místa a velikosti náhodné změny. popis. Podrobný popis struktury a principu jednotlivých kroků algoritmu zde není uveden, především z důvodu zastarání genetických algoritmů a velkého množství nevýhod, jež jsou úspěšně řešeny pomocí moderních evolučních algoritmů, zejména pak Diferenciální evoluce a algoritmu SOMA. Tyto nové algoritmy přinesly spoustu výhod při řešení optimalizačních úloh a jsou dále podrobněji popsány v rámci účelnosti pro tuto práci.

## 3.2 Diferenciální Evoluce

Diferenciální evoluce je poměrně novým typem evolučního algoritmu, pomocí kterého lze efektivně řešit takové úlohy optimalizace, kdy se v parametrech účelové funkce kombinují celá, diskrétní a nespojitá čísla. Jeho struktura je velice podobná algoritmům genetickým, ale na rozdíl od nich disponuje celou řadou výhod a vylepšení.

Významnými rysy jsou zejména schopnost kombinovat parametry z množiny reálných, nebo celých čísel případně z libovolně nadefinované diskrétní množiny čísel a především pak používání dekadických čísel, z čehož plyne výhoda snadné implementace a vysoké rychlosti díky celkové jednoduchosti celého algoritmu a také proto, že není nutné provádět převody binárních čísel na dekadické hodnoty a zpět. Díky nezávislosti křížení na kvalitě rodičů je diferenciální evoluce méně náchylná ke konvergenci k lokálnímu extrému, takže téměř vždy nalezne globální extrém. Za velkou výhodu lze též považovat i to, že můžeme získat více řešení a následně vybrat to nejvíce vyhovující. Diferenciální evoluce tak díky svým vlastnostem umí nalézat řešení i u funkcí trpících takovými patologiemi jako je např. šum, vysoký počet dimenzí, „multimodalita“ (více lokálních extrémů) a „epistáze“ (interakce parametrů).

### 3.2.1 Struktura algoritmu DE

Princip diferenciální evoluce je následující. Celý algoritmus se cyklicky opakuje v tzv. generacích (jedna generace - jeden cyklus) a pracuje s tzv. populacemi. Populaci si lze představit jako matici  $N \times M$ , kde každý sloupcový vektor je tzv. jedinec. Každý jedinec se skládá z konečného počtu prvků, které pak spolu s hodnotou účelové funkce udávají dimenzi, ve které probíhá daná evoluce.

Algoritmus se člení do následujících kroků:

1. Stanovení parametrů - jedná se o parametry, které určují chod evolučního procesu, jako:  $F$  - mutační konstanta (0-2),  $CR$  - práh křížení (0-1),  $NP$  - počet jedinců v populaci,  $D$  - rozměr jedince. Nutno definovat též prototyp jedince - z jakého typů čísel se bude skládat (Real, Integer). Parametry diferenciální evoluce jsou voleny pevně, ale je vhodné je během evoluce měnit podle toho, jak kvalitně probíhá. Pro optimální nastavení parametrů je možno použít diferenciální evoluci, pak se mluví o meta-diferenciální evoluci (METADE). Pomocí tohoto způsobu optimalizace parametrů DE dosáhneme přesnějšího řešení, i když za cenu vyšší časové náročnosti.
2. Tvorba populace - populace se tvoří vygenerováním množiny jedinců (matice) podle prototypového vektoru. U každého jedince je nutno počítat s jedním prvkem navíc - hodnota účelové funkce.
3. Započítí cyklu generace - během každé generace zabezpečuje postupné evoluční šlechtění každého jedince z populace. V tomto cyklu se postupně vybírají všichni jedinci z populace a pro každého z nich je proveden následný evoluční cyklus.
4. Evoluční cyklus - v tomto cyklu se pro každého jedince náhodně zvolí tři další různí jedinci z populace a s pomocí předem definovaných parametrů a přesného algoritmu je vytvořen nový jedinec tak, aby bylo dosaženo lepších hodnot účelové funkce a tím zajištění tvorby nové generace jedinců s lepšími vlastnostmi. Celý cyklus se opakuje až do vyčerpání populace.
5. Vyhodnocení a ukončení - celý proces generací se opakuje, dokud není vyčerpán zadaný počet generací nebo dokud libovolný uživatelem definovaný ukončovací parametr dosáhne své zadané hodnoty.

Kvalitu a průběh evolučního procesu lze ovlivnit mnoha faktory, zejména:

- Nastavením parametrů – jejich kombinace může mít podstatný vliv na průběh a rychlost evoluce.
- Velikostí populace - při malé populaci bude horší výběr, při velké bude potřeba více času na její postupný průchod a vytvoření nové.
- Definicí účelové funkce - nevhodné či špatné nadefinování může evoluci zpomalit až zastavit.
- Počtem generací – při malém počtu generací může evoluce skončit ještě dříve než nalezne extrém.

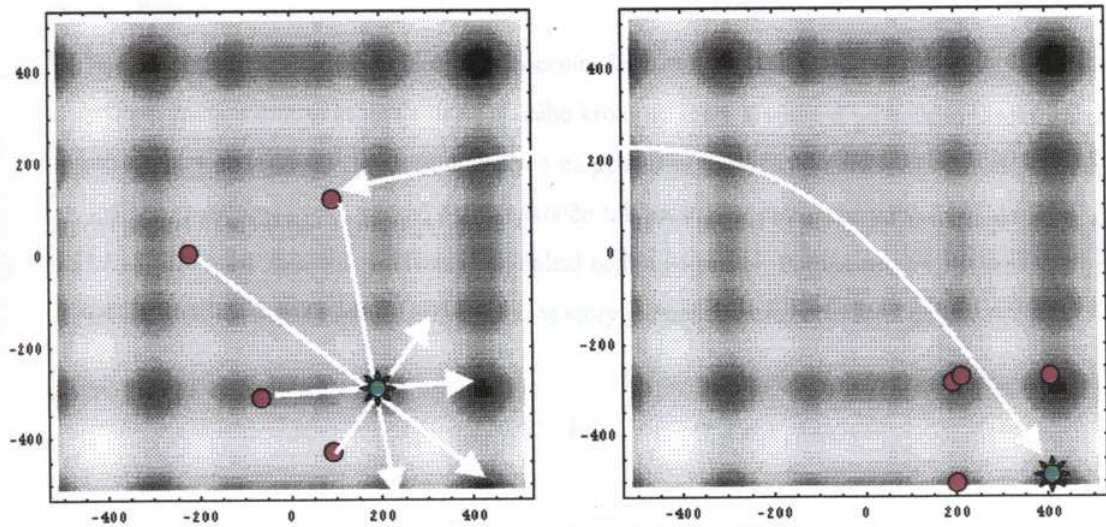
- Definicí intervalu - Je lepší definovat interval evoluce, pokud jsou nejasnosti o funkci, na které hledáme extrém, aby se evoluční proces udržoval v prostoru předpokládaných řešení.

### 3.3 Evoluční algoritmus SOMA

Samo-Organizující se Migrační Algoritmus (SOMA) lze považovat za nový typ evolučního algoritmu, i když se od klasického evolučního algoritmu poněkud liší a spadá spíše do kategorie matematických algoritmů. Spolu s diferenciální evolucí má SOMA oproti genetickým algoritmům celou řadu výhod a bylo již o nich zmíněno. Hlavním rozdílem je, že nevytváří žádné nové populace. Po spuštění algoritmu SOMA je vytvořena pouze jedna základní populace, jejíž vývoj není dán ve smyslu vytváření nových lepších populací tj. vlastní proces není založený na evolučním principu, ale na pravidle, které velmi jednoduše popisuje migrační chování např. skupiny živočichů starající se o potravu. Analogií k tomuto chování pak vznikne proces migrace jednotlivců po  $n$ -rozměrné ploše, jejíž globální extrém představuje optimální řešení daného problému. Algoritmus SOMA můžeme označit jako evoluční, protože jeden migrační cyklus je ekvivalentní jednomu cyklu klasického evolučního algoritmu, neboť jednotlivci získali nové postavení na  $n$ -rozměrné ploše. Chování jednotlivců je tvořeno podle jednoduchých pravidel a populace všech jednotlivců pak vykazuje vlastnosti samoorganizace.

#### 3.3.1 Struktura algoritmu SOMA

SOMA pracuje v tzv. migračních kolech, které mají stejný význam jako generace u evolučních algoritmů. Před spuštěním algoritmu se vygeneruje populace náhodně rozložená po  $n$ -rozměrné ploše účelové funkce. Základní strategie algoritmu SOMA je založená na principu, že jedinec, který dosahuje v populaci nejoptimálnější hodnoty účelové funkce, je vůdce a všichni ostatní jedinci migrují směrem k němu po předem stanovených krocích. V jaké pozici vůči vůdci ukončí jedinci migraci, určuje parametr *PathLength*. Během každého kroku jedince při migraci po  $n$ -rozměrné ploše se přepočítává hodnota účelové funkce. Po ukončení migračního kola se všichni jedinci vrátí na nejlepší pozici na své trajektorii, kterou našli během migrace, tedy kde dosahuje účelová funkce nejlepší hodnoty. Vyhodnotí se nejlepší jedinec a ten se stává novým vůdcem. Celý proces se opakuje dokud se všichni jedinci neseťkají v jednom extrému.



Obr. 7 - Princip algoritmu SOMA

Realizaci algoritmu SOMA lze rozčlenit do několika kroků :

1. Definice parametrů - před spuštěním algoritmu je nutné zvolit parametry jako jsou: *Step* (velikost kroku migrace), *PathLength* (max. vzdálenost migrace), *MinDif* (max. rozdíl hodnot účelové funkce jedinců postačující k ukončení algoritmu), *PopSize* (velikost populace), *Migrace* (počet migračních kol), *PRT* (konstantu perturbace). Přičemž *PRT* parametr je v určitém smyslu ekvivalent *CR* parametru u genetického algoritmu nebo diferenciální evoluce. Má vliv na skutečnost, zda jedinec bude migrovat přímo směrem k vůdci, nebo jeho trajektorie bude odkloněna a dojde tak k dokonalejšímu prohledávání *n*-rozměrné plochy a tedy k vyšší robustnosti při hledání globálního extrému. Bez použití *PRT* parametru, SOMA často našla pouze lokální extrém.
2. Generování populace - V tomto kroku je náhodně vygenerována počáteční populace za pomoci vzorového jedince – *Specimenu*, v němž je přesně definován typ a rozsah hodnot  $\langle Lo;Hi \rangle$  každého z parametrů jedince. Populace pak je číselná matice, jejíž sloupce jsou jednotliví jedinci, stejně tak jako u DE.
3. Migrační proces - V tomto kroku dochází k samotné migraci jednotlivých jedinců po *n*-rozměrné hyperploše dle pravidel jednotlivých strategií algoritmu SOMA.
4. Vyhodnocení – Po ukončení migračního procesu dojde k vyhodnocení rozdílů hodnot účelové funkce jednotlivých jedinců, pokud je tento rozdíl menší než parametr *MinDif* algoritmus se ukončí - krok 5, jinak opětovné spuštění migračního procesu - krok 3.



5. Konec algoritmu SOMA a jako řešení je vybrán jedinec s nejlepší hodnotou účelové funkce.

### 3.3.2 Strategie algoritmu SOMA

V současné době existuje několik základních strategií algoritmu SOMA, všechny tyto verze jsou navzájem srovnatelné při hledání globálního extrému.

„AllToOne“ – Princip této strategie již byl výše popsán. Všichni jedinci putují směrem k vůdci, ten zůstává na svém místě.

„AllToAll“ - Při této strategii jedinci nemigrují k vůdci, ale ke všem ostatním jedincům. Zde neexistuje vůdce. Po ukončení migračního procesu se jedinec vrací na místo, kde našel nejlepší hodnotu účelové funkce při migracích za ostatními jedinci. Tato strategie je více robustní při hledání globálního extrému oproti AllToOne, protože každý jedinec hledá během jednoho migračního kola v širokém prostoru na n-rozměrné hyperploše, ale zároveň je více výpočtově náročná.

„AllToAllAdaptive“ - Princip je totožný jako u strategie AllToAll, s tím rozdílem, že jedinec se přesouvá do lepší pozice vždy po ukončení migrace k jinému jedinci a odtud začíná další migraci k ostatním jedincům.

„AllToRand“- je strategie, kde všichni jedinci migrují k vůdci, ten ale není vybírán dle nejoptimálnější hodnoty účelové funkce, nýbrž je zvolen náhodně.

Kvalitu a průběh algoritmu SOMA při hledání řešení lze ovlivnit mnoha faktory stejně tak jako u diferenciální evoluce, kde je jejich možný vliv stručně popsán.

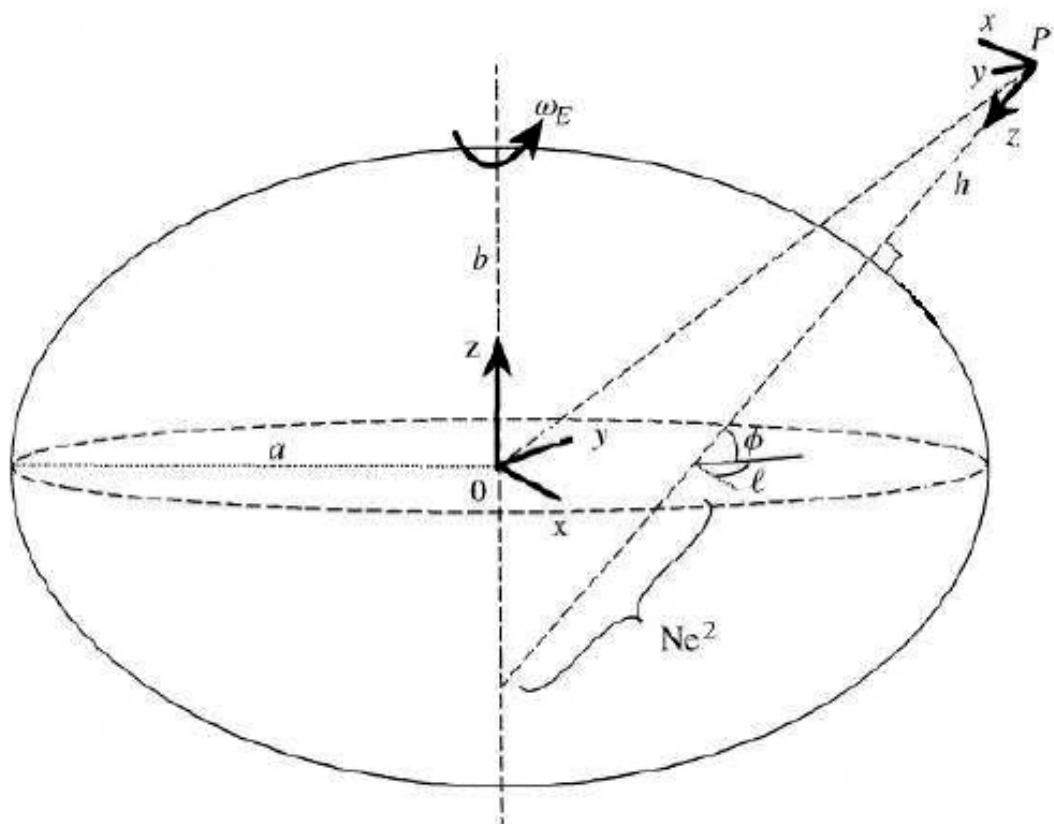
## 4 POHYBOVÉ ROVNICE LETADLA

V této kapitole je popsán matematický základ modelu letounu s nímž se dále pracuje.

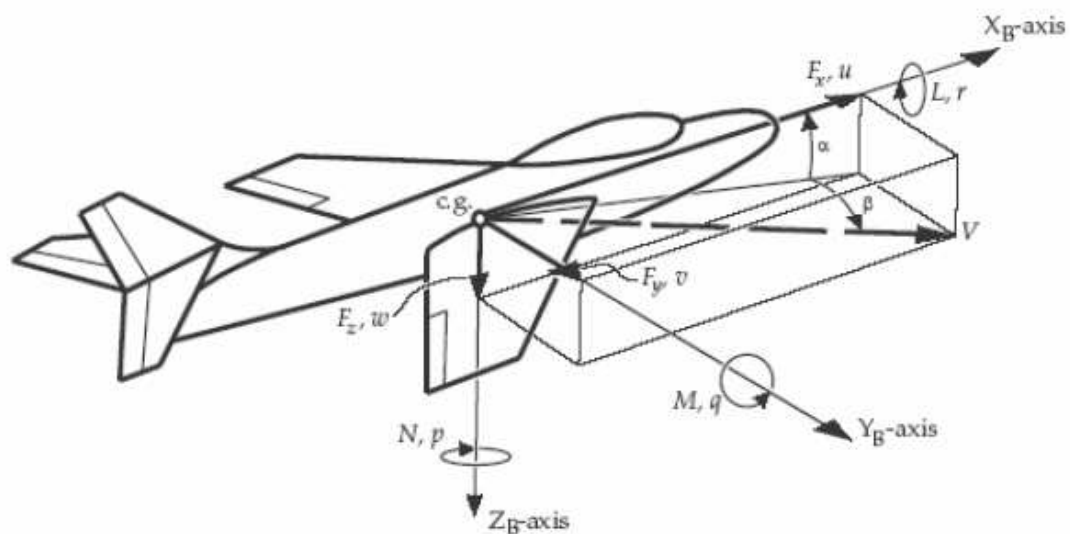
Jsou zavedeny tyto předpoklady:

- Letoun je dokonale tuhé těleso a je souměrné v rovině  $xz$
- Vliv gyroskopických momentů od pohonné jednotky je zanedbán

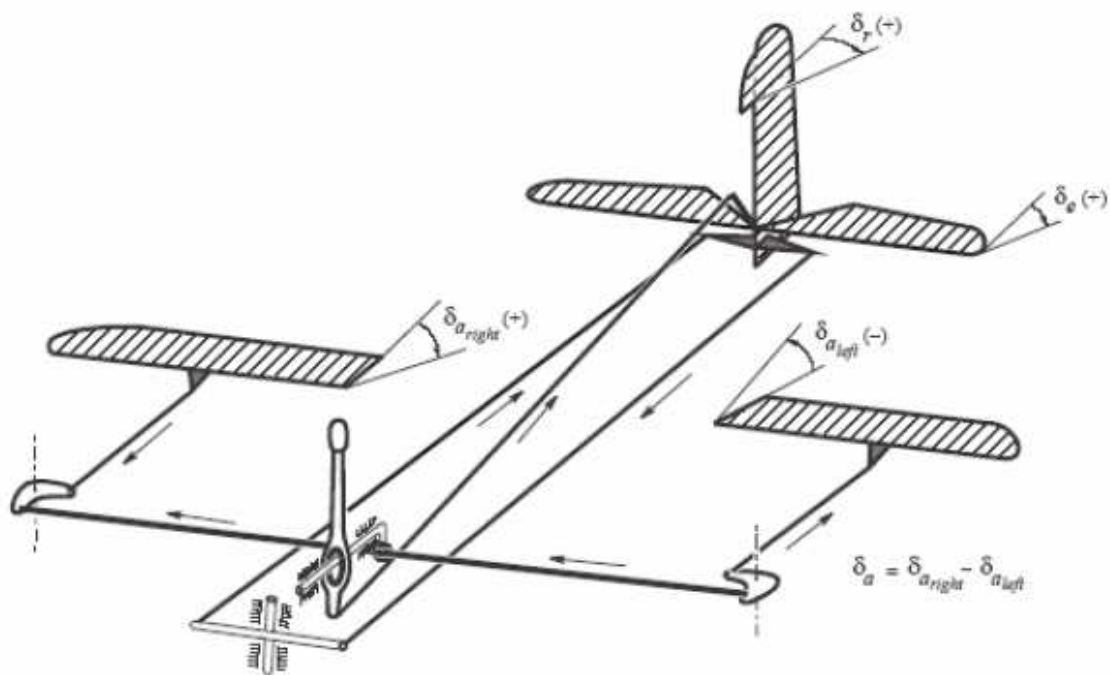
### 4.1 Souřadné soustavy



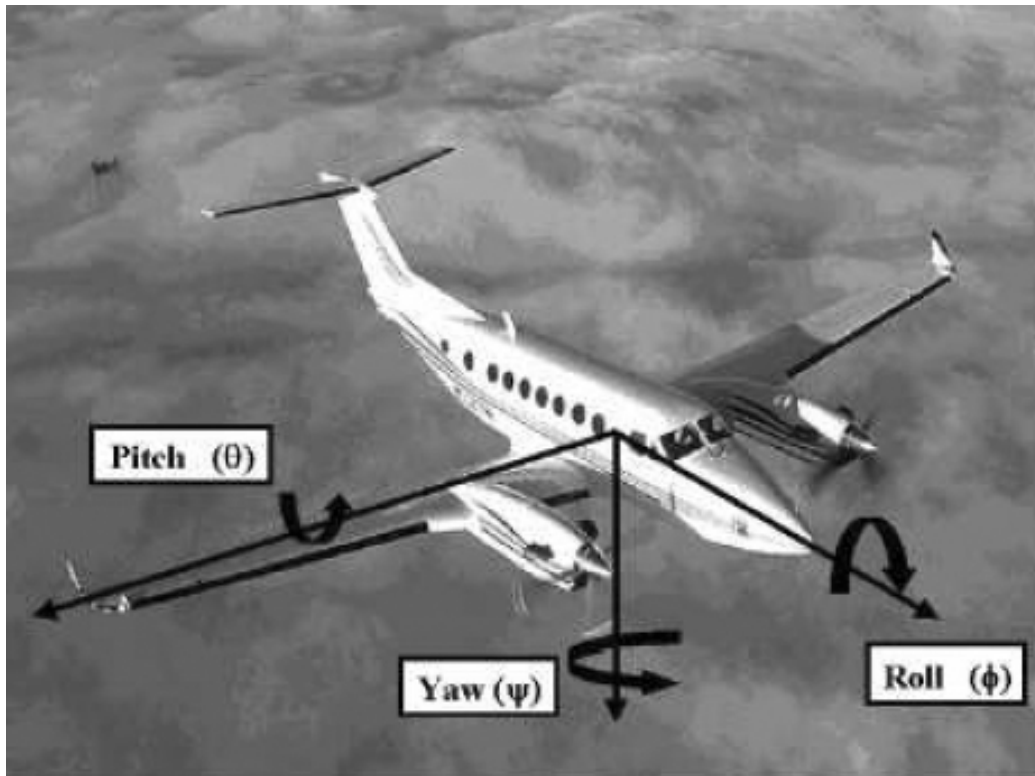
Obr. 8 – Souřadná soustava ECI (Earth-Centered Inertial)



Obr. 9 – Letadlová souřadná soustava (aircraft body-fixed axes)



Obr. 10 – Výchylky kormidel (kladný smysl)



Obr. 11 – Eulerovy úhly

## 4.2 Pohybové rovnice

$$\begin{bmatrix} \dot{p} \\ \dot{v}_B \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \Omega_E & B^T & 0 & 0 \\ -B\Omega_E^2 & -(\Omega_B + B\Omega_E B^T) & 0 & 0 \\ 0 & 0 & -J^{-1}\Omega_B J & 0 \\ 0 & 0 & 0 & -\frac{1}{2}\Omega_q \end{bmatrix} * \begin{bmatrix} p \\ v_B \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ Bg(p) + \frac{F_B}{m} \\ J^{-1}M_B \\ 0 \end{bmatrix}$$

(4)

Kde:

$\Omega_E$ ,  $\Omega_B$ ,  $\Omega_q$  - matice koeficientů úhlových rychlostí

$$\Omega_E = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -e \\ 0 & e & 0 \end{bmatrix}, \Omega_B = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}, \Omega_q = \begin{bmatrix} 0 & p & q & r \\ -p & 0 & -r & q \\ -q & r & 0 & -p \\ -r & -q & p & 0 \end{bmatrix} \quad (5)$$

- $e$  - úhlová rychlost otáčení Země
- $p$  – úhlová rychlost klonění
- $q$  – úhlová rychlost klopení
- $r$  – úhlová rychlost zatačení

$B$  - transformační matice pro přepočítání ze soustavy ECI do letadlové souřadné soustavy

$B_B$  - transformační matice pro přepočítání mezi letadlovou souřadnou soustavou a souřadnou soustavou tečnou k povrchu země

$B_G$  - transformační matice pro přepočítání mezi souřadnou soustavou tečnou k povrchu země a souřadnou soustavou ECI

$$B = B_B B_G \quad (6)$$

$$B = \begin{bmatrix} q_{0^2} + q_{1^2} - q_{2^2} - q_{3^2} & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_3 - q_0 q_3) & q_{0^2} - q_{1^2} + q_{2^2} - q_{3^2} & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_{0^2} - q_{1^2} - q_{2^2} + q_{3^2} \end{bmatrix} \quad (7)$$

$$B_G = \begin{bmatrix} \cos \mu & -\sin \mu \sin l & \sin \mu \cos l \\ 0 & \cos l & \sin l \\ -\sin \mu & -\cos \mu \sin l & \cos \mu \cos l \end{bmatrix} \quad (8)$$

$B^T$  - transponovaná matice  $B$ , tato matice slouží pro přepočítání z letadlové souřadné soustavy do soustavy ECI

$J$  - matice momentů setrvačnosti

$$J = \begin{bmatrix} I_X & 0 & -I_{XZ} \\ 0 & I_Y & 0 \\ -I_{XZ} & 0 & I_Z \end{bmatrix} \quad (9)$$

$\mathbf{p}$  - polohový vektor v ECI, tento vektor může mít tvar  $p = [p_x, p_y, p_z]^T$  nebo

$$p = [r, h]^T$$

$v_B$  - rychlost v letadlové souřadné soustavě  $v_B = [u, v, w]^T$

$\omega_B$  - úhlová rychlost v letadlové souřadné soustavě

$\mathbf{q}$  - vektor quaternionů  $q = [q_0, q_1, q_2, q_3]$

$F_B$  - vektor sil působících na letadlo v letadlové souřadné soustavě (součet sil aerodynamických, od motoru, od podvozku)

$M_B$  - vektor momentů působících na letadlo v letadlové souřadné soustavě (součet momentů aerodynamických, od motoru, od podvozku)

$m$  - hmotnost letadla

$\mathbf{g}$  - vektor gravitačního zrychlení v souřadné soustavě ECI

$\dot{p}$  - derivace vektoru polohy v ECI

$\dot{v}_B$  - derivace vektoru rychlosti v letadlové souřadné soustavě

$\dot{\omega}_B$  - vektor úhlového zrychlení v letadlové souřadné soustavě

$\dot{q}$  - derivace quaternionů

Numerickou integrací  $\dot{p}, \dot{v}_B, \dot{\omega}_B, \dot{q}$  dostaneme nové hodnoty  $\mathbf{p}, v_B, \omega_B, \mathbf{q}$

Pro rychlost letadla platí:

$$V = \sqrt{u^2 + v^2 + w^2} \quad (10)$$

Úhel náběhu:

$$= \arctan \frac{w}{u} \quad (11)$$

Úhel vybočení:

$$= \arcsin \frac{v}{v_T} \quad (12)$$

Časová změna úhlu náběhu respektive úhlu vybočení je stanovena podle:

$$\dot{\cdot} = \frac{u \dot{w} - w \dot{u}}{u^2 + w^2} \quad (13)$$

$$\dot{\cdot} = \frac{\dot{v} v_T - v \dot{v}_T}{v_T^2 \cos} \quad (14)$$

Derivace rychlosti se vypočítá z:

$$\dot{V} = \frac{u \dot{u} + v \dot{v} + w \dot{w}}{v_T} \quad (15)$$

Pro Eulerovy úhly platí:

$$= \arctan \left( \frac{2 * (q_2 q_3 + q_0 q_1)}{q_0 - q_1 - q_2 + q_3} \right) \quad (16)$$

$$= -\arcsin \left( 2 (q_1 q_3 - q_0 q_2) \right) \quad (17)$$

$$= \arctan \left( \frac{2 * (q_1 q_2 + q_0 q_3)}{q_0 + q_1 - q_2 - q_3} \right) \quad (18)$$

#### 4.2.1 Použitá integrační metoda

Pro numerickou integraci byla zvolena metoda ABM (Adams-Bashforth-Moulton) druhého řádu, která se používá při integraci s proměnným integračním krokem:

$$x(n+1) = x(n) + \frac{T}{2} [3 \dot{x}(n) - \dot{x}(n-1)] \quad (19)$$

#### 4.2.2 Stavový vektor

Stavový vektor výše uvedených pohybových rovnic má tvar:

$$X = \begin{bmatrix} p_x \\ p_y \\ p_z \\ u \\ v \\ w \\ p \\ q \\ r \\ q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (20)$$



Z praktických důvodů (nastavování počátečních podmínek) se při vyvažování a linearizaci používá následující stavový vektor (resp. derivace stavového vektoru):

$$X = \begin{bmatrix} V \\ p \\ q \\ r \\ lat^* \\ lon^* \\ H \end{bmatrix} \quad \dot{X} = \begin{bmatrix} \dot{V} \\ \cdot \\ \cdot \\ \dot{p} \\ \dot{q} \\ \dot{r} \\ \cdot \\ \cdot \\ lat_{dot}^* \\ lon_{dot}^* \\ \dot{H} \end{bmatrix} \quad (21)$$

\* Zeměpisná šířka a délka (resp. jejich derivace) nemají z hlediska vyvažování žádný praktický význam a nepoužívají se.

Vektor řízení:

$$u = \begin{bmatrix} PLA \\ \partial_a \\ \partial_e \\ \partial_r \end{bmatrix} \quad (22)$$

*PLA* – poloha páky nastavení motoru

$\partial_a$  - výchylka křidélek

$\partial_e$  - výchylka výškovky

$\partial_r$  - výchylka směrového kormidla

### 4.3 TRIM

#### Obecně

Vyvážený stav je takový stav, kdy jsou stavové veličiny konstantní nebo rovny nule. Z toho vyplývá, že ve vyváženém stavu jsou síly a momenty působící na letadlo nulové nebo konstantní.

Platí:

$$\dot{p} = \dot{q} = \dot{r} = \dot{V} = \dot{\alpha} = \dot{\beta} = 0 \quad (23)$$

$\mathbf{X} = 0$   $\mathbf{u} = \text{konst.}$

#### Definice vyvážených stavů

Jednotlivé vyvážené stavy jsou definovány:

Ustálený horizontální let:  $\dot{\alpha} = \dot{\beta} = \dot{\gamma} = 0$

Ustálená zatáčka:  $\dot{\alpha} = \dot{\beta} = 0$   $\dot{\gamma} = \text{konst.}$

Ustálené klonění:  $\dot{\alpha} = \dot{\beta} = 0$   $\dot{\gamma} = \text{konst.}$

Ustálené klopení:  $\dot{\alpha} = \dot{\beta} = \dot{\gamma} = 0$   $\dot{V} = \text{konst.}$

#### Vyvažování

Vlastní vyvažování probíhá numericky. Počáteční podmínky (stavový vektor) jsou specifikovány uživatelem. Vyvažovací algoritmus – pomocí minimalizační funkce – potom hledá takovou kombinaci vyvažovacího vektoru (tj. výchylek výškovky, křídélek, směrovky,

nastavení páky ovládání motoru, úhlu náběhu, úhlu vybočení, úhlu trajektorie), která co nejlépe (resp. s danou přesností) splňuje podmínku  $\dot{\mathbf{X}} = 0$ .

V programu jsou použity dvě minimalizační (váhové) funkce<sup>3</sup>.

Jednoduchá:

$$J = A * v^2 + B * (\alpha^2 + \beta^2) + C * (p^2 + q^2 + r^2) \quad (24)$$

Podrobná:

$$J = A * v^2 + B * \alpha^2 + C * \beta^2 + D * p^2 + E * q^2 + F * r^2 \quad (25)$$

<sup>3</sup> V programu je použita podrobná váhová funkce s koeficienty: A = 1, B = 30, C = 50, D = 5, E = 5, F = 30.

$J$  - hodnota váhové funkce

**Vyvažovací vektor:**

$$V = \begin{bmatrix} PLA \\ \partial_a \\ \partial_e \\ \partial_r \end{bmatrix} \quad (26)$$

- úhel trajektorie

## 5 ÚVOD K PRAKTICKÉ ČÁSTI

Při experimentech byl využíván software Mathematica verze 5.2.

### 5.1 Základní body postupu

Cílem je nalézt co možná nejlepší kombinaci parametrů **vyvažovacího vektoru** tak, aby byly co nejlépe splněny podmínky pro vyvážený let, za daných **letových podmínek**, tj. aby hodnota účelové funkce byla co nejmenší.

Samotné nastavování parametrů a vyhodnocování účelové funkce probíhá pomocí externích *DLL* knihoven připojených k Mathematice. Tyto knihovny obsahují funkce pro přístup k jednotlivým parametrům a pro výpočet potřebných proměnných.

#### Postup při optimalizaci:

V první fázi jsou nastaveny letové podmínky, za kterých optimalizace probíhá – konkrétně jsou to: výška letu, hmotnost letounu, rychlost letu a rychlost bočního větru. To je uskutečňováno pomocí funkcí volaných prostřednictvím Mathematicy:

```
1 setWeightVelocityAltitude[hmotnost, rychlost, vyska];  
2 setWind[rychlostvetru];
```

Dále již probíhají jednotlivé iterační kroky optimalizace, popsáno bude, co se děje během jednoho kroku.

- V prvním kroku jsou nastaveny hodnoty vyvažovacího vektoru (26).
- Na základě těchto hodnot je spočítán stavový vektor, respektive jeho derivace.
- Podle hodnot derivace stavového vektoru je spočítána hodnota účelové funkce.
- Nakonec se na základě optimalizačního algoritmu určí nové hodnoty vyvažovacího vektoru a vracíme se na první krok.

### 5.2 Vyvažovací vektor

Vyvažovací vektor má následující parametry:

---

$PLA$	Nastavení polohy páky ovládání motoru. Rozsah <18;130> Funkce <code>setPowerLeverAngle[]</code>
$\partial_a$	Nastavení výchylky křidélek [rad] Rozsah <-0.26;0.26> Funkce <code>setAileronDeflection[]</code>
$\partial_e$	Nastavení výchylky výškového kormidla [rad] Rozsah <-0.51;0.34> Funkce <code>setElevatorDeflection[]</code>
$\partial_r$	Nastavení výchylky směrového kormidla [rad] Rozsah <-0.61;0.51> Funkce <code>setRudderDeflection[]</code>
	Nastavení úhlu náběhu [rad] Funkce <code>setAngleOfAttack[]</code>

Nastavení úhlu vybočení [rad]

Funkce `setAngleOfSideslip[]`

Nastavení úhlu náklonu [rad]

Funkce `setRollAngle[]`

### 5.3 Účelová funkce a derivace stavového vektoru

V počátcích byla při ohodnocování využívána výše popsaná účelová funkce (25). Po jejím bližším prozkoumání však byla nahrazena funkcí novou. Druhé mocniny v původní účelové funkci při poklesu hodnot odpovídajících prvků derivace stavového vektoru pod velikost 1 způsobovaly nepřirozený pokles hodnoty účelové funkce a celá optimalizace směřovala k „falešnému“ neexistujícímu minimu.

Nová, při simulacích používaná účelová funkce má tvar:

$$J = A * |\dot{v}| + B * |\dot{\gamma}| + C * |\dot{\alpha}| + D * |\dot{p}| + E * |\dot{q}| + F * |\dot{r}| \quad (27)$$

Přičemž hodnoty váhových koeficientů A, B, C, D, E, F byly zvětšeny 100000krát.

Cílem účelové funkce je minimalizovat relevantní složky derivace stavového vektoru tak, aby byla co nejlépe splněna podmínka vyváženého letu (23).

### 5.3.1 Složky derivace stavového vektoru

Prvky, jejichž minimalizace je cílem celé optimalizace. Aby letoun setrval ve vyváženém letu, je nutné tyto složky minimalizovat na hodnotu co možná nejbližší 0, tj. nalézt takovou kombinaci vyvažovacího vektoru, na základě které je spočítán takový stavový vektor, že odpovídající hodnoty derivací jeho složek se blíží 0.

$\dot{v}$	Derivace rychlosti TAS <sup>4</sup> [ $m/s^2$ ]
$\cdot$	Derivace úhlu náběhu [ $rad/s$ ]
$\cdot$	Derivace úhlu vybočení [ $rad/s$ ]
$\dot{p}$	Derivace úhlové rychlosti klonění [ $rad/s^2$ ]
$\dot{q}$	Derivace úhlové rychlosti klopení [ $rad/s^2$ ]
$\dot{r}$	Derivace úhlové rychlosti zatáčení [ $rad/s^2$ ]

## II. PRAKTICKÁ ČÁST

---

<sup>4</sup> TAS (True AirSpeed, pravá vzdušná rychlost) je rychlost, kterou se letadlo pohybuje vůči vzduchu.

## 6 GRAFY – ŘEZY ÚČELOVOU FUNKCÍ

Vybrané trojrozměrné řezy hyperplochou účelové funkce (27).

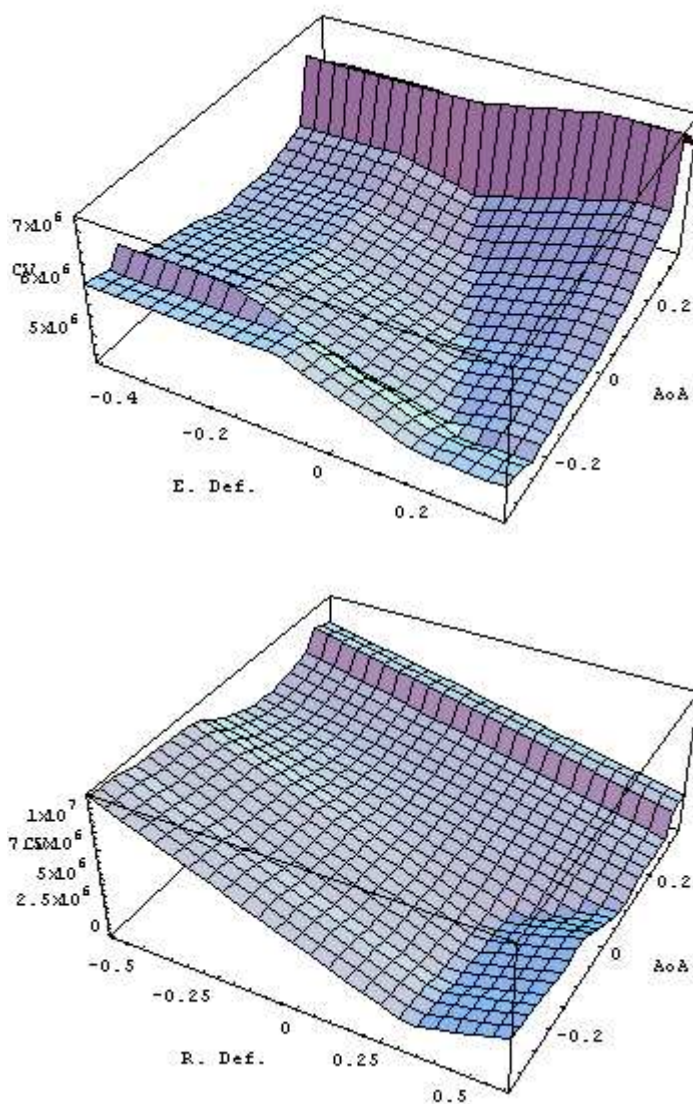
### Podmínky:

hmotnost: 6000

rychlost: 100

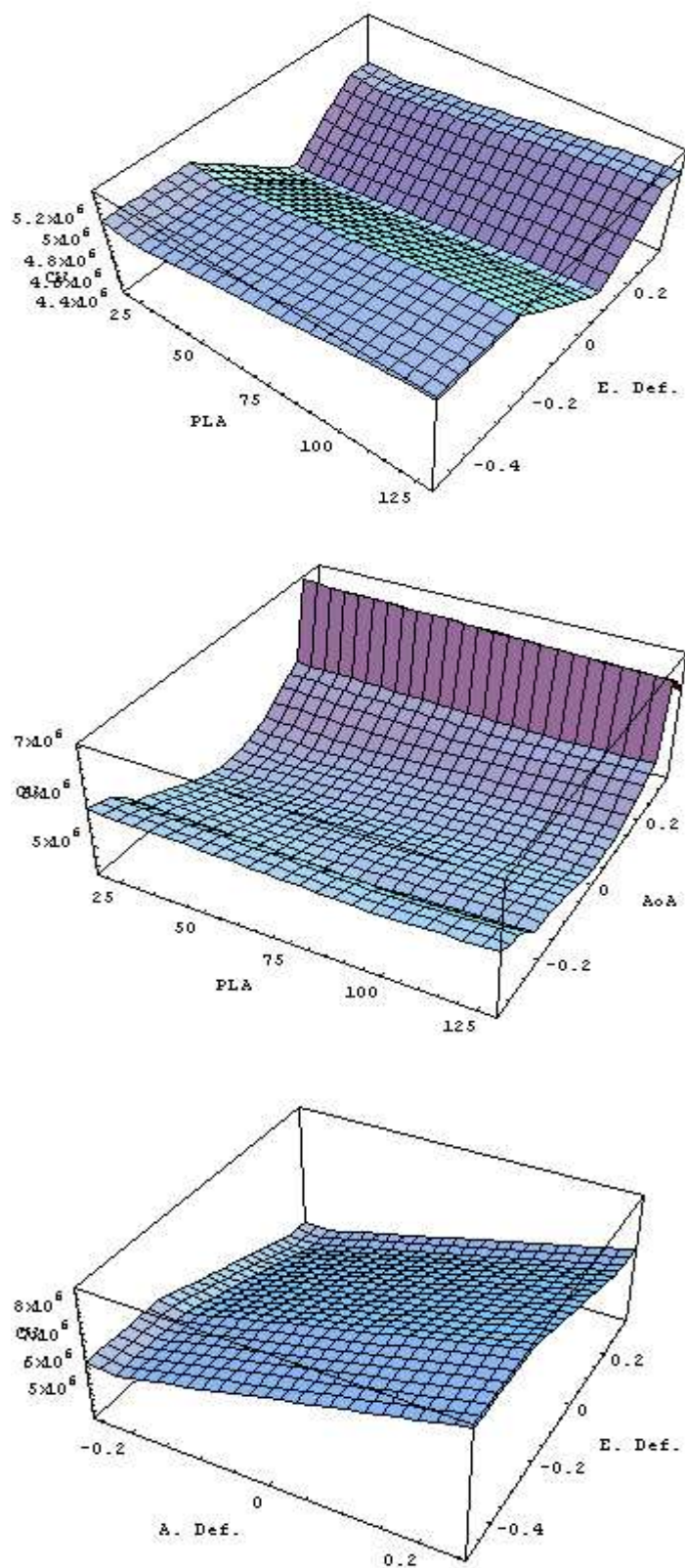
výška: 3000

rychlost bočního větru: 30



Obr. 12 - Řezy hyperplochou účelové funkce





Obr. 13 - Řezy hyperplochou účelové funkce

## 7 SIMULAČNÍ POKUSY

### 7.1 SOMA: ATAA 20 m/s

Při rychlosti větru 20 m/s při různých letových podmínkách.

Použitá verze algoritmu SOMA: ATAA.

Parametry: PopSize: 20, Migrations: 35, Step: .11, PathLength: 3., PRT: .1, MinDiv: .1

Účelová funkce (27).

Tabulka 1 – Hodnoty minimalizovaných parametrů při simulačních pokusech

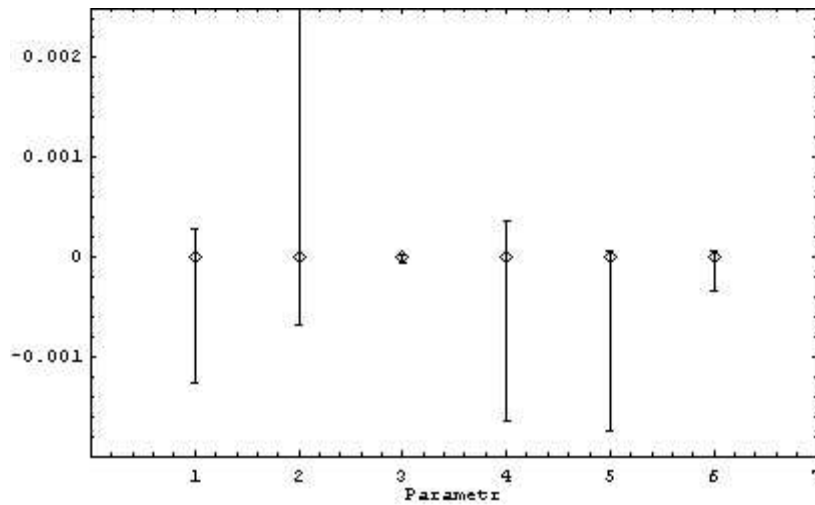
Derivace rychlosti	Derivace úhlu náběhu	Derivace úhlu vybočení
$3.37473 \times 10^{-12}$	$-6.61382 \times 10^{-13}$	$2.83969 \times 10^{-13}$
$1.15044 \times 10^{-8}$	$-3.2271 \times 10^{-10}$	$4.58048 \times 10^{-12}$
$1.85003 \times 10^{-13}$	$1.28695 \times 10^{-12}$	$3.5728 \times 10^{-14}$
-0.000193446	$2.5872 \times 10^{-6}$	$9.33823 \times 10^{-7}$
0.0000393188	-0.0000666755	$-3.33013 \times 10^{-6}$
$-2.88013 \times 10^{-12}$	$-1.88513 \times 10^{-14}$	$-2.76159 \times 10^{-14}$
$-6.85348 \times 10^{-12}$	$1.27704 \times 10^{-13}$	$-7.76841 \times 10^{-14}$
0.0000512809	$-7.90109 \times 10^{-7}$	$-3.87805 \times 10^{-7}$
-0.0000104156	-0.0000839476	$2.04821 \times 10^{-6}$
$1.30598 \times 10^{-7}$	$-2.17172 \times 10^{-7}$	$-7.01057 \times 10^{-9}$
0.0000639919	0.0000336924	$-1.01859 \times 10^{-6}$
-0.0000692114	0.0000210122	$-1.54264 \times 10^{-6}$
0.0000802489	-0.0000131121	$-2.06323 \times 10^{-7}$
-0.0000626965	0.000406233	$-5.71047 \times 10^{-7}$
0.000283437	-0.0000694478	$-8.08735 \times 10^{-7}$
-0.0000221912	-0.0000291099	$5.86652 \times 10^{-8}$
$-9.99939 \times 10^{-6}$	-0.00067466	$1.58688 \times 10^{-7}$
-0.0000194059	0.000609907	$-1.88943 \times 10^{-7}$
-0.000186085	0.00631532	$-0.0000547448$
-0.000537345	0.00031565	$-7.9517 \times 10^{-7}$
$-5.42798 \times 10^{-6}$	0.00309087	-0.000023049
-0.00126243	0.0111298	0.0000138734
0.000108834	0.0168484	$5.9111 \times 10^{-6}$
-0.0000310021	0.0201161	$-4.00277 \times 10^{-7}$

Tabulka 2 - Hodnoty minimalizovaných parametrů při simulačních pokusech

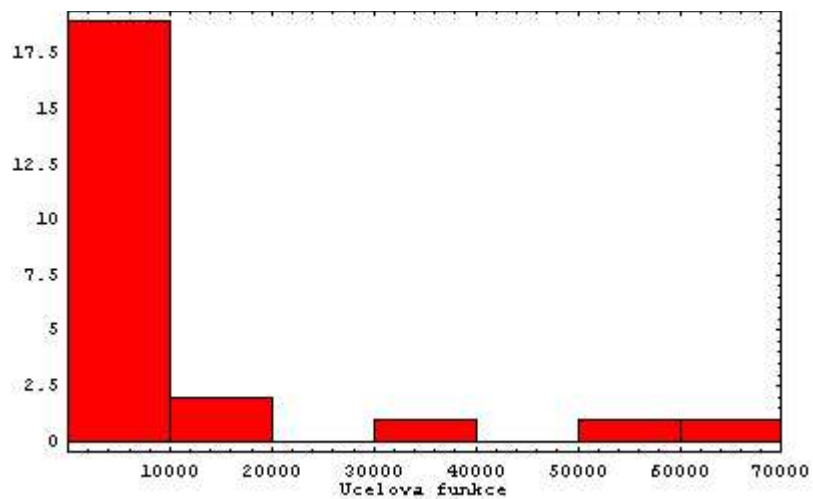
Derivace úhlové rychlosti klonění	Derivace úhlové rychlosti klopení	Derivace úhlové rychlosti zatažení
$-1.68262 \times 10^{-12}$	$-1.75958 \times 10^{-13}$	$-1.75958 \times 10^{-13}$
$6.64338 \times 10^{-10}$	$3.80412 \times 10^{-9}$	$3.80412 \times 10^{-9}$
$4.57151 \times 10^{-16}$	$-9.6697 \times 10^{-13}$	$-9.6697 \times 10^{-13}$
0.0000408605	-0.0000826027	-0.0000826027
0.0000273073	$-3.4464 \times 10^{-6}$	$-3.4464 \times 10^{-6}$
$5.39213 \times 10^{-12}$	$7.3311 \times 10^{-13}$	$7.3311 \times 10^{-13}$
$-1.17322 \times 10^{-12}$	$-3.54805 \times 10^{-13}$	$-3.54805 \times 10^{-13}$
-0.0000175059	$-6.5327 \times 10^{-6}$	$-6.5327 \times 10^{-6}$
-0.0000714801	$-8.371 \times 10^{-6}$	$-8.371 \times 10^{-6}$
$-1.70742 \times 10^{-7}$	$1.37244 \times 10^{-8}$	$1.37244 \times 10^{-8}$
$-2.40626 \times 10^{-6}$	$1.67917 \times 10^{-6}$	$1.67917 \times 10^{-6}$
-0.0000166174	$2.83333 \times 10^{-6}$	$2.83333 \times 10^{-6}$
0.0000376769	-0.0000515778	-0.0000515778
-0.000131152	0.0000121071	0.0000121071
0.0000541494	-0.0000649156	-0.0000649156
0.0000675148	0.0000184472	0.0000184472
$-2.44162 \times 10^{-6}$	$5.38557 \times 10^{-7}$	$5.38557 \times 10^{-7}$
$-5.94848 \times 10^{-6}$	$3.78531 \times 10^{-6}$	$3.78531 \times 10^{-6}$
0.0000349128	-0.0000345625	-0.0000345625
0.0000561914	-0.000235901	-0.000235901
-0.000396573	-0.0017458	-0.0017458
-0.0016413	0.0000555528	0.0000555528
0.000353683	-0.000179637	-0.000179637
$2.26785 \times 10^{-6}$	$2.64934 \times 10^{-7}$	$2.64934 \times 10^{-7}$

Tabulka 3 - Legenda

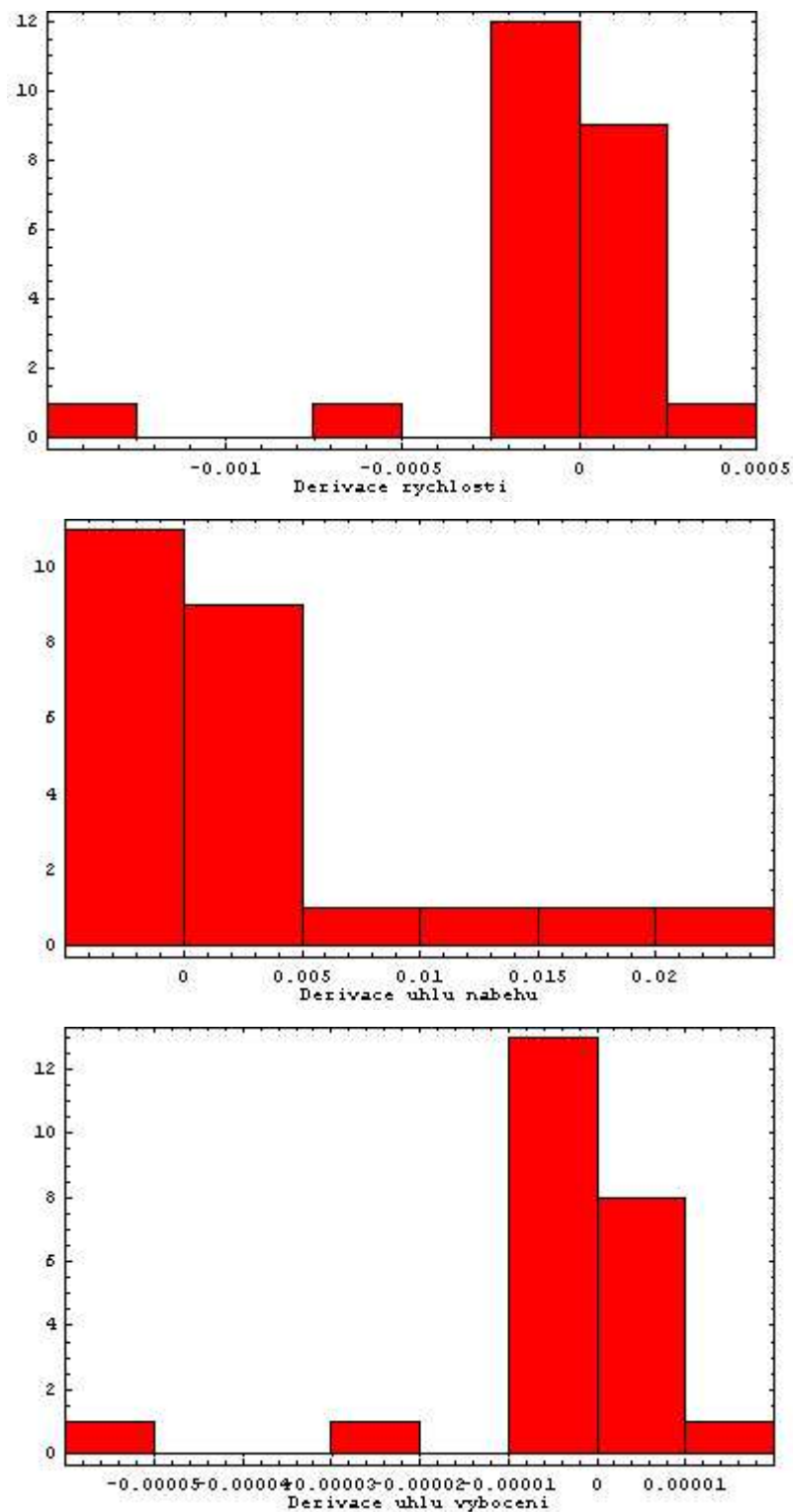
Parametr	
1	Derivace rychlosti TAS
2	Derivace úhlu náběhu
3	Derivace úhlu vybočení
4	Derivace úhlové rychlosti klonění
5	Derivace úhlové rychlosti klopení
6	Derivace úhlové rychlosti zatačení



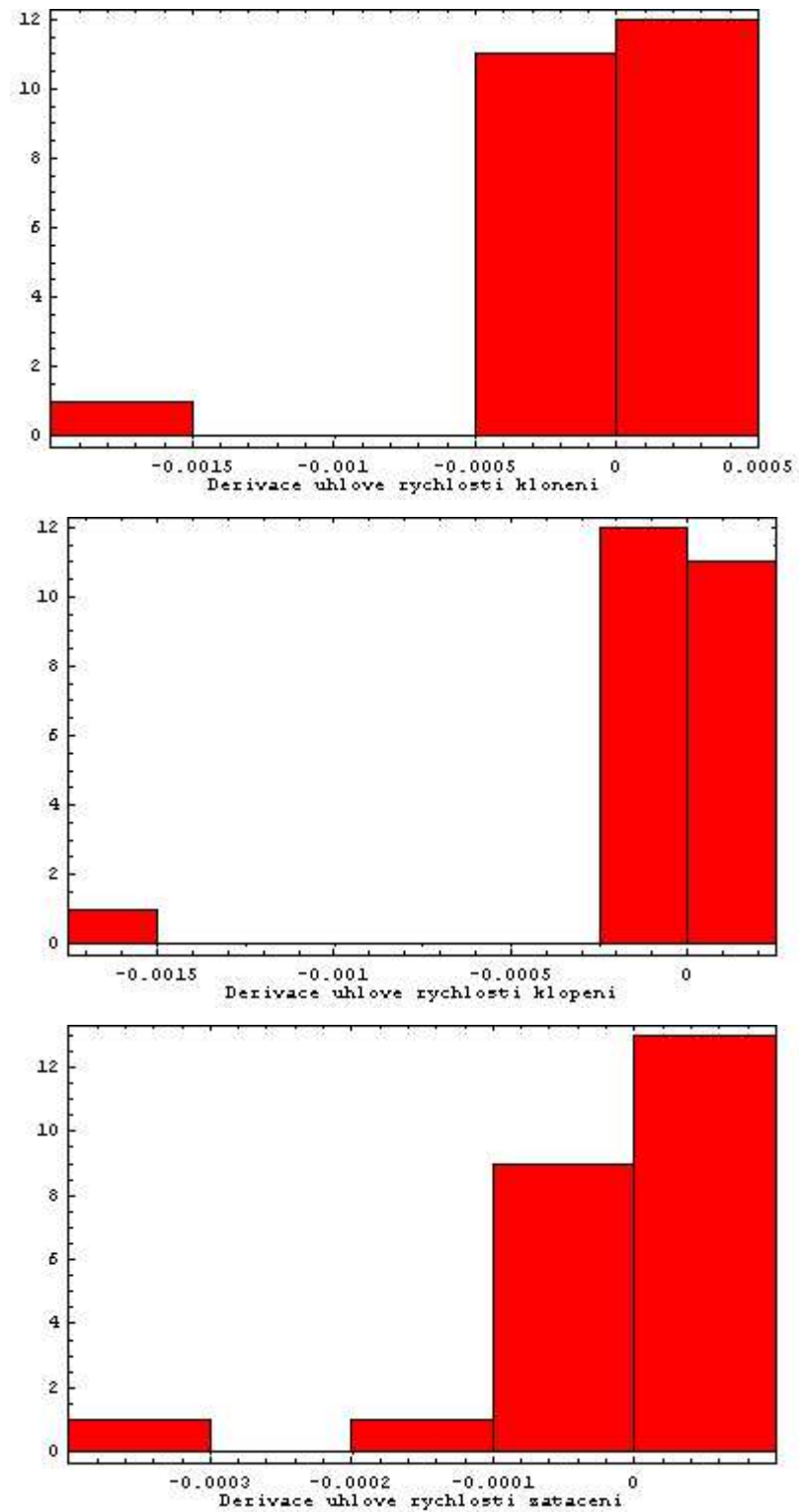
Obr. 14 – Výchyly hodnot jednotlivých parametrů



Obr. 15 – Hodnoty účelové funkce



Obr. 16 – Grafy četností hodnot parametrů



Obr. 17 – Grafy četností hodnot parametrů

## 7.2 SOMA: ATAA 40 m/s

Při rychlosti větru 40 m/s při různých letových podmínkách.

Použitá verze algoritmu SOMA: ATAA.

Parametry: PopSize: 20, Migrations: 35, Step: .11, PathLength: 3., PRT: .1, MinDiv: .1

Účelová funkce (27).

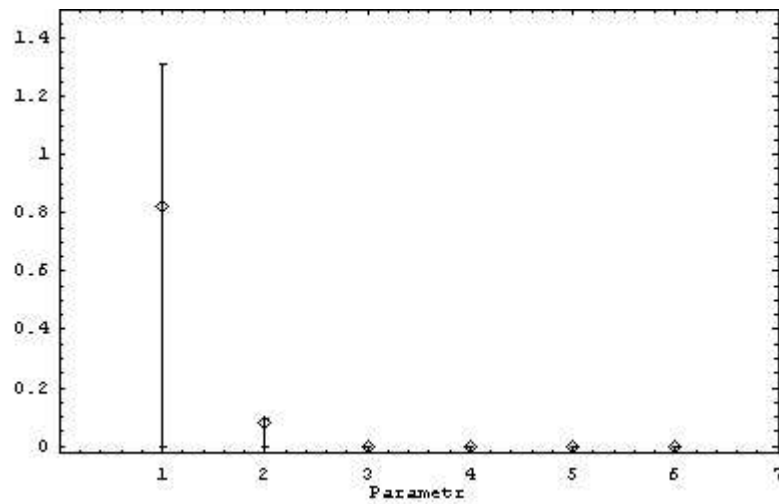
Tabulka 4 – Hodnoty minimalizovaných parametrů při simulačních pokusech

Derivace rychlosti	Derivace úhlu náběhu	Derivace úhlu vybočení
1.07146	0.0802247	$-5.28561 \times 10^{-8}$
1.01108	0.0820544	$-2.95242 \times 10^{-7}$
0.956884	0.0836369	$2.22912 \times 10^{-7}$
0.905411	0.0851542	$-0.0000357938$
0.972728	0.0840654	$-1.3729 \times 10^{-8}$
0.915917	0.0858929	$8.55174 \times 10^{-6}$
0.868001	0.0869708	$-7.79586 \times 10^{-6}$
0.823265	0.088148	$-0.0000117706$
1.01118	0.0820418	$-3.86121 \times 10^{-11}$
0.972722	0.084066	$-6.07878 \times 10^{-8}$
0.917873	0.0856055	$8.38039 \times 10^{-7}$
0.86579	0.0873152	$3.4796 \times 10^{-6}$
0.823984	0.0881253	$1.76653 \times 10^{-6}$
1.27809	0.0869138	$-2.54156 \times 10^{-9}$
1.2055	0.0882491	$-2.41833 \times 10^{-7}$
1.13943	0.0895466	$-1.93868 \times 10^{-7}$
1.07854	0.0905717	$-0.0000558624$
1.31018	0.0897204	$-2.34534 \times 10^{-6}$
1.23559	0.0908553	$-9.10072 \times 10^{-8}$
1.16875	0.0918498	$-5.24359 \times 10^{-7}$
1.107	0.092802	$-0.0000240275$

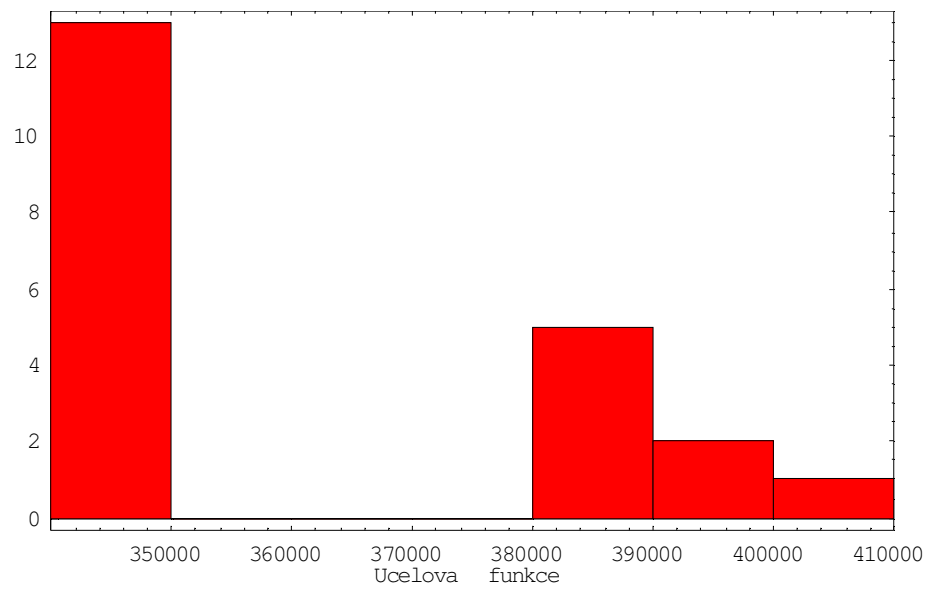
Tabulka 5 – Hodnoty minimalizovaných parametrů při simulačních pokusech

Derivace úhlové rychlosti klonění	Derivace úhlové rychlosti klopení	Derivace úhlové rychlosti zatažení
$-1.87811 \times 10^{-7}$	$-5.35157 \times 10^{-7}$	$-1.85097 \times 10^{-9}$
$1.69731 \times 10^{-6}$	$-1.95901 \times 10^{-6}$	$-2.46263 \times 10^{-7}$
$-1.4914 \times 10^{-6}$	$-9.65945 \times 10^{-6}$	$4.91402 \times 10^{-8}$
-0.0000953308	-0.000198965	-0.0000155432
$-2.21055 \times 10^{-8}$	$-6.51909 \times 10^{-8}$	$-7.51019 \times 10^{-8}$
-0.000302483	-0.000378715	-0.0000656389
0.0000695884	$1.76961 \times 10^{-6}$	$-7.17124 \times 10^{-7}$
-0.0000111416	-0.0000896775	$9.94675 \times 10^{-6}$
$-2.2898 \times 10^{-10}$	$8.21257 \times 10^{-11}$	$-5.24993 \times 10^{-11}$
$5.73049 \times 10^{-7}$	$-1.41676 \times 10^{-7}$	$-3.33572 \times 10^{-8}$
$7.37289 \times 10^{-6}$	0.0000119903	$3.43961 \times 10^{-7}$
0.0000250952	-0.00013678	-0.0000308302
-0.0000751694	-0.000127535	$-3.06972 \times 10^{-7}$
$-2.15982 \times 10^{-8}$	$-1.48293 \times 10^{-8}$	$-1.30096 \times 10^{-8}$
$-1.64147 \times 10^{-6}$	$4.39805 \times 10^{-6}$	$4.01672 \times 10^{-7}$
$-7.2284 \times 10^{-7}$	$-6.92121 \times 10^{-6}$	$-1.93648 \times 10^{-6}$
0.0000526125	0.0000168819	0.000102096
$-1.23032 \times 10^{-6}$	$-1.66945 \times 10^{-6}$	$6.89053 \times 10^{-8}$
$-4.1343 \times 10^{-8}$	$-2.02957 \times 10^{-6}$	$-6.71443 \times 10^{-7}$
$-2.48784 \times 10^{-6}$	$3.35814 \times 10^{-6}$	$5.59321 \times 10^{-7}$
-0.0000717417	-0.0000151493	$-3.28473 \times 10^{-6}$

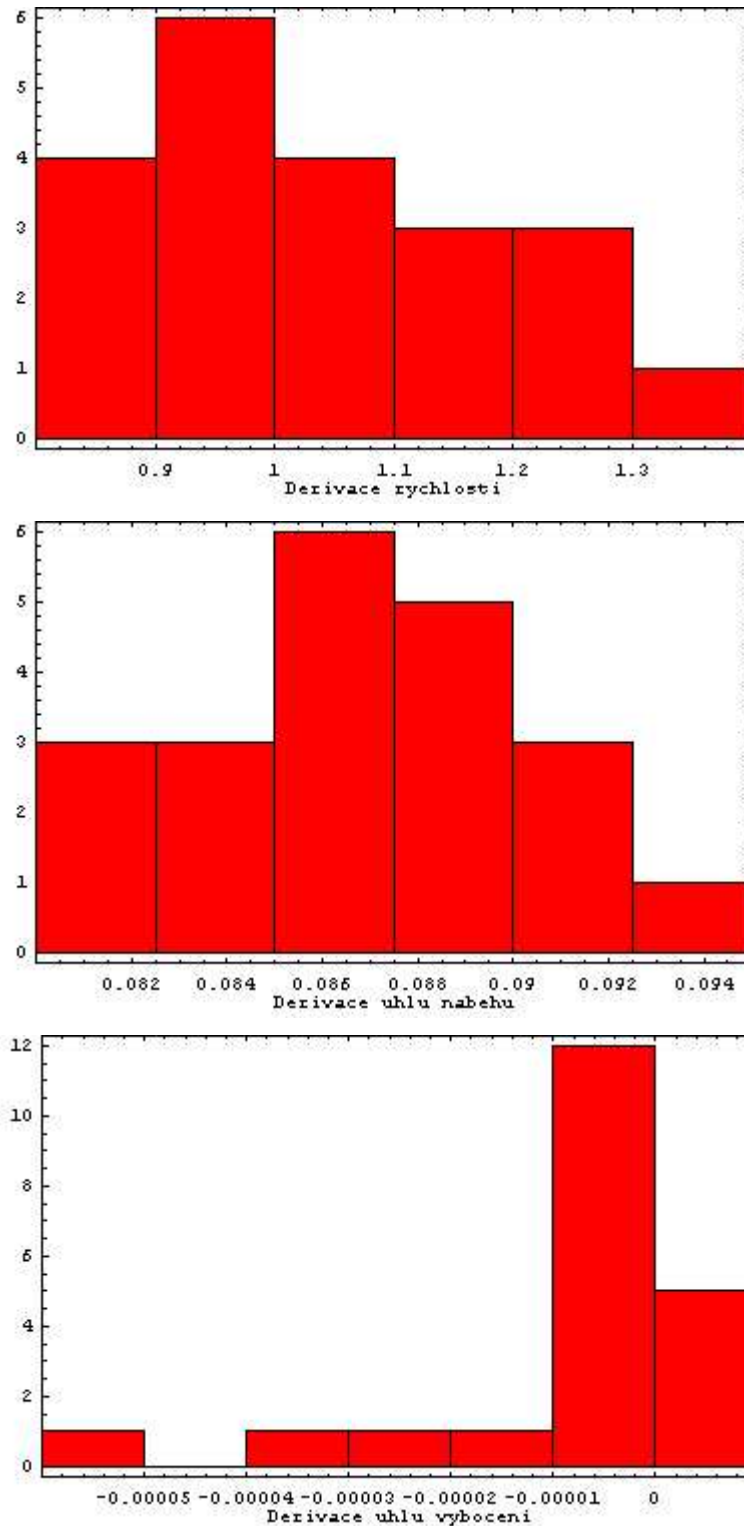




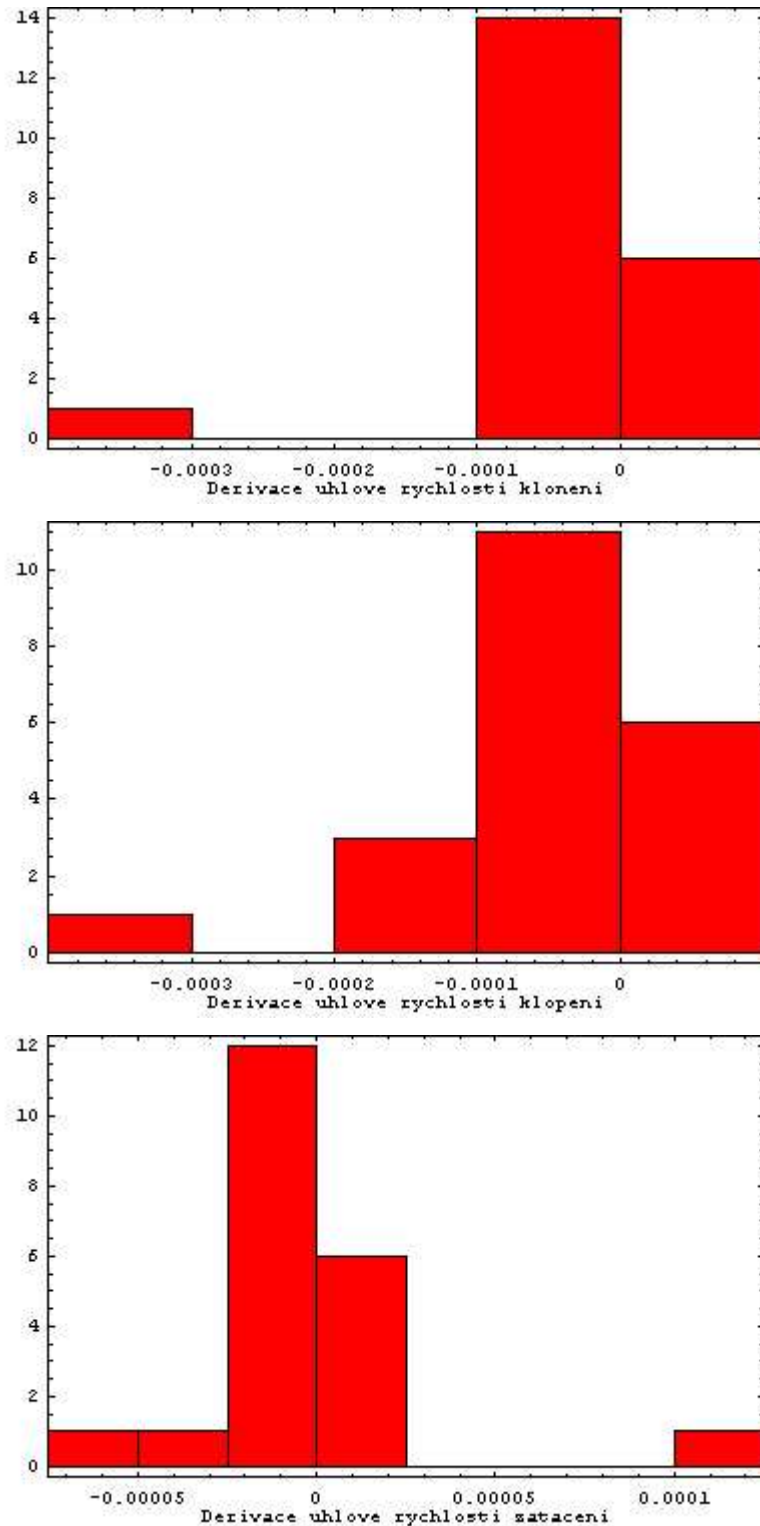
Obr. 18 - Výchyly hodnot jednotlivých parametrů



Obr. 19 – Hodnoty účelové funkce



Obr. 20 - Grafy četností hodnot parametrů



Obr. 21 - Grafy četností hodnot parametrů

text

### 7.3 Trim 40 m/s

Při rychlosti větru 40 m/s při různých letových podmínkách.

Použitá verze algoritmu: Trim

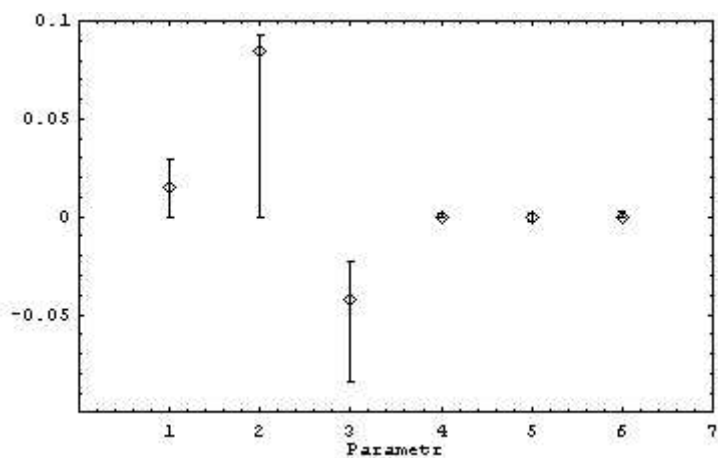
Účelová funkce (27).

Tabulka 6 - Hodnoty minimalizovaných parametrů při simulačních pokusech

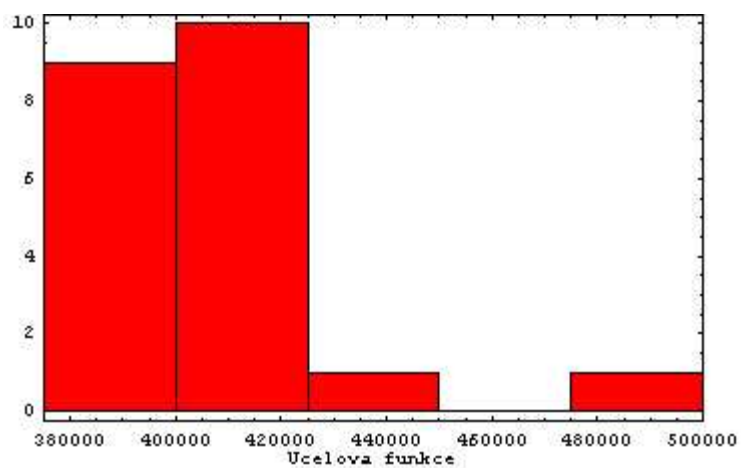
Derivace rychlosti	Derivace úhlu náběhu	Derivace úhlu vybočení
0.01852	0.0849429	-0.0266559
0.020007	0.0860254	-0.0251449
0.0165291	0.08731	-0.0237954
0.0155287	0.0882894	-0.0225807
0.0187455	0.0875587	-0.027957
0.0173284	0.0886016	-0.0278443
0.0167148	0.0896425	-0.0249497
0.0158557	0.0904988	-0.0236727
0.020007	0.0860254	-0.0251449
0.0187455	0.0875587	-0.027957
0.0173284	0.0886016	-0.0278443
0.0167148	0.0896425	-0.0249497
0.0158557	0.0904988	-0.0236727
0.0188485	0.0897379	-0.0290077
0.0177655	0.0907242	-0.0273599
0.0167964	0.0916038	-0.0258816
0.0159302	0.0923667	-0.0245561
0.0187889	0.0915762	-0.0297185
0.0177055	0.0924732	-0.0280271
0.0167373	0.0932718	-0.0265111
0.0290269	0.0913241	-0.042093

Tabulka 7 - Hodnoty minimalizovaných parametrů při simulačních pokusech

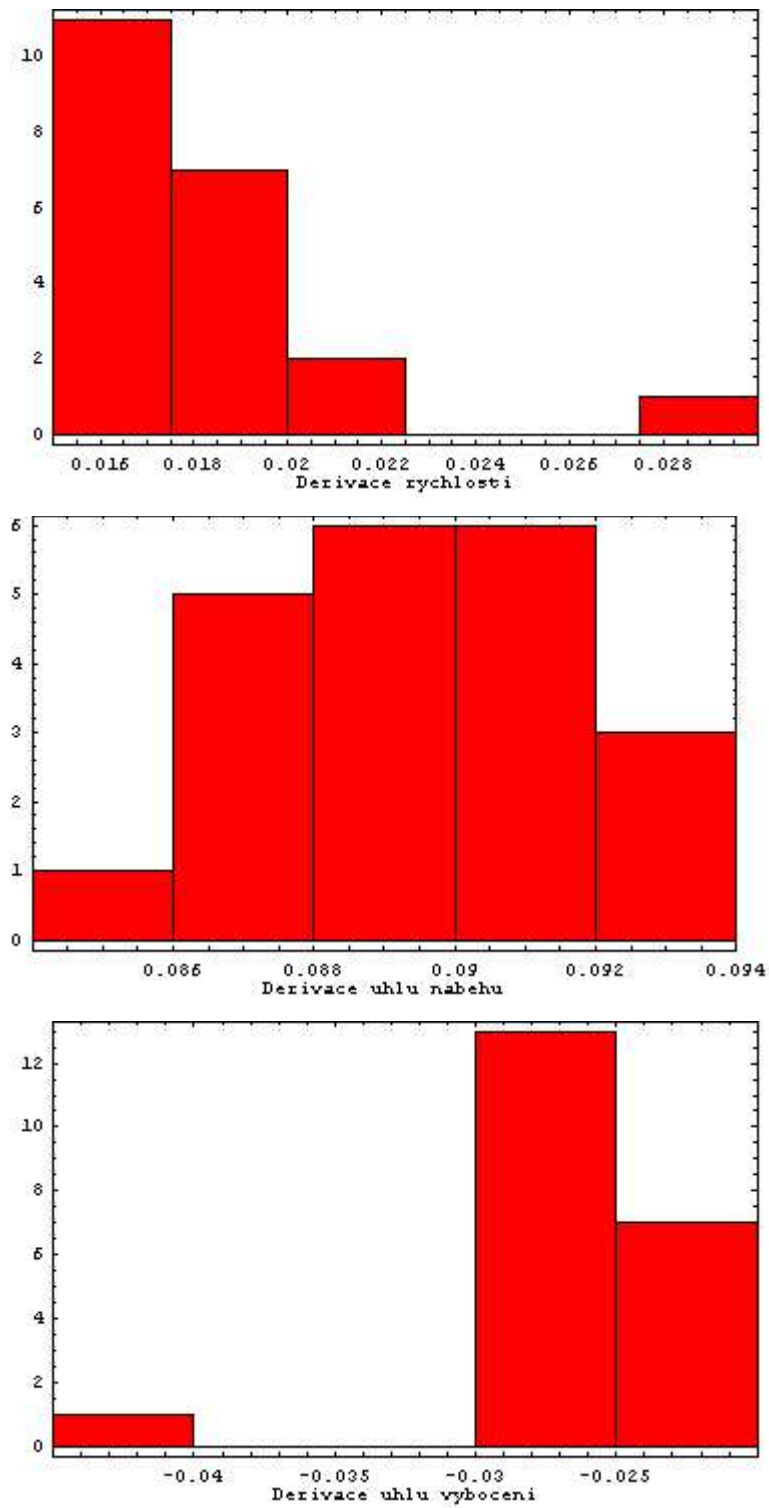
Derivace úhlové rychlosti klonění	Derivace úhlové rychlosti klopení	Derivace úhlové rychlosti zatažení
0.000599416	-0.000527181	0.00117646
0.00156002	-0.000543538	0.00244979
0.000606078	-0.000427606	0.00122356
0.000688914	-0.00046836	0.00130575
0.000609025	-0.000527234	0.00119756
0.00038148	-0.00063955	0.00121673
0.000614044	-0.000426642	0.00124166
0.000601161	-0.000386076	0.00136144
0.00156002	-0.000543538	0.00244979
0.000609025	-0.000527234	0.00119756
0.00038148	-0.00063955	0.00121673
0.000614044	-0.000426642	0.00124166
0.000601161	-0.000386076	0.00136144
0.000621428	-0.000520635	0.00122322
0.000623617	-0.000466881	0.00124756
0.000624389	-0.000420468	0.00126385
0.000610276	-0.000380118	0.00138412
0.000640132	-0.000506247	0.00125975
0.000640509	-0.000453582	0.00128128
0.000639767	-0.000408147	0.00129505
-0.00046315	-0.0023893	-0.000282206



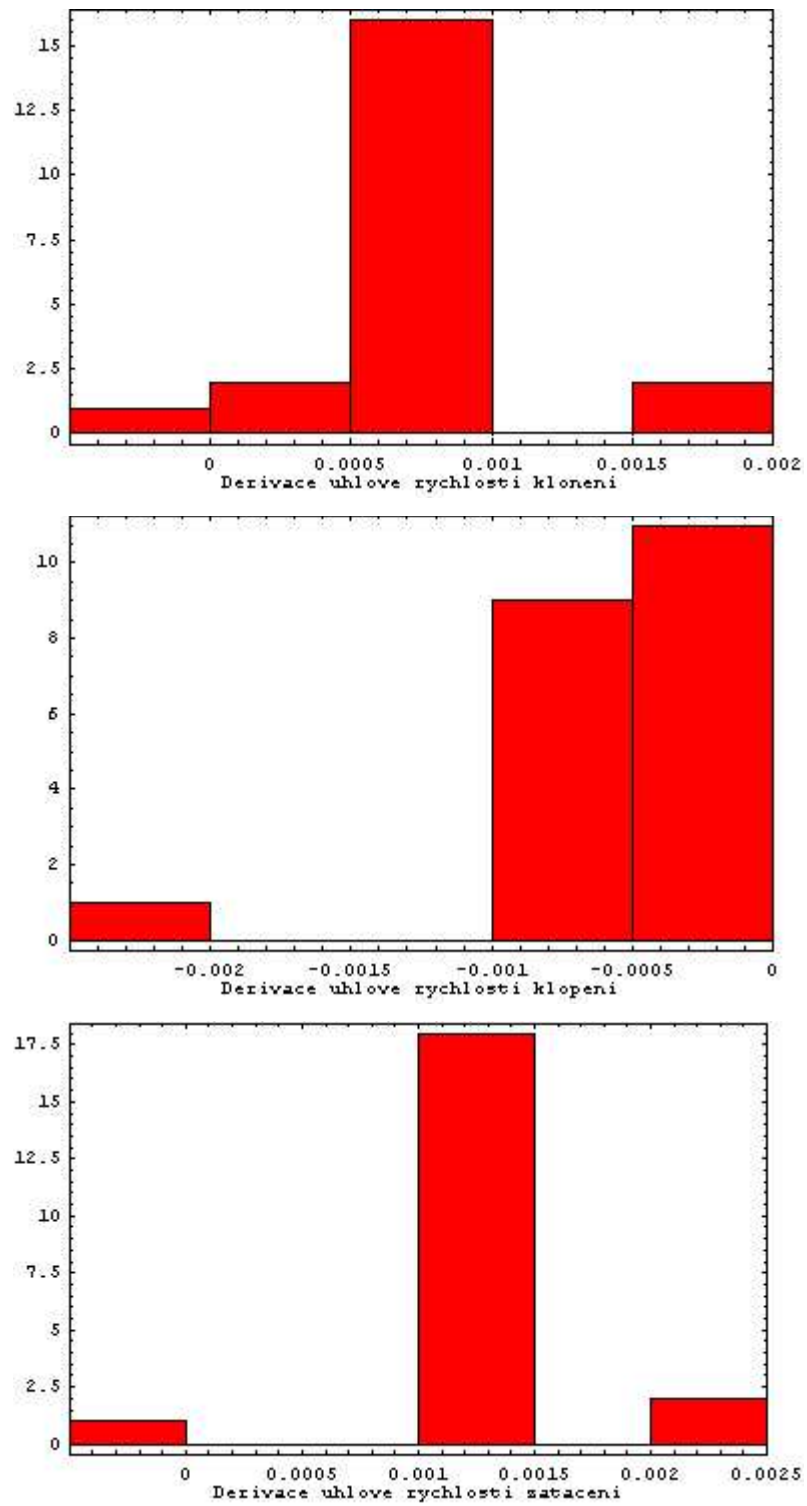
Obr. 22 - Výchylky hodnot jednotlivých parametrů



Obr. 23 - Hodnoty účelové funkce



Obr. 24 - Grafy četností hodnot parametrů



Obr. 25 - Grafy četností hodnot parametrů



## ZÁVĚR

Cílem této práce bylo aplikovat evoluční algoritmy na vyvažování nelineárního matematického modelu letounu. V průběhu práce byla provedena řada simulací, jejichž část je prezentována v praktické sekci. Hlavním problémem bylo vyvažování s vlivem bočního větru, kdy výsledky dosahované původní metodou nebyly uspokojivé. Při použití algoritmu SOMA bylo zejména při vyšších hodnotách bočního větru dosaženo jistého zlepšení, přesto však výsledky stále nejsou dostačující.

Toto lze přičíst dvěma faktorům a to stále i po úpravě nedokonalé účelové funkce a pak také samotnému modelu. Pokud jsme dvěma různými metodami dosáhli přibližně stejných výsledků (v případě SOMY skutečně robustních), které neodpovídají skutečné vyvažitelnosti letounu, je možné, že chyba je v samotném modelu. Tato hypotéza je rovněž námětem pro další práci.

Při simulacích byla používána nová účelová funkce, která byla vytvořena jako upravená verze předchozí, u níž byl zjištěn možný nedostatek v konstrukci.

V průběhu práce byla řešena řada problémů související s přenesením softwarového základu pro simulace do prostředí programu Mathematica, což zapříčinilo zpoždění. Nepodařilo se také provést všechny naplánované simulace z důvodu časové náročnosti jednotlivých simulací.

Tato práce je základem pro další pokračování, zejména pro hledání nové účelové funkce, která by splňovala požadavek na funkčnost při letu bez i s bočním větrem.

## ZÁVĚR V ANGLIČTINĚ

Main target of this work was application of evolutionary algorithms on trimming of nonlinear mathematic aircraft model. Batch of simulations was accomplished, its part is presented in practical section. The main issue was trimming with influence of side wind, where the results reached by the original method weren't satisfactory. There was some improvement by use of SOMA algorithm, however the results weren't still adequate enough.

This may be caused by two factors – non-perfect cost function and also the model itself. If we had reached approximately same results while using two different methods (in SOMA case really robust), which don't correspond with real aircraft trimming possibilities, there is an option, that there is an error in model itself. This hypothesis is also a subject for further work.

New cost function, which was created as modified original cost function, was used during simulations.

There were several technical issues related to transfer of software basis for simulations in Mathematica environment, which caused delay in work. We were are also not able to perform all simulations we had planned because of time heftiness of simulations.

This work is base for continuation, mainly for search of new cost function, which would realize demand on functionality during flight without and with side wind influence.

**SEZNAM POUŽITÉ LITERATURY**

- [0] ZELINKA, Ivan. *Umělá inteligence I. Volume 1*. Zlín: Vutium, Brno, 1998. 126 s. ISBN 80-214-1163-5.
- [0] ZELINKA, Ivan, *Umělá inteligence / kap.6 "Diferenciální evoluce"*, Academia, 33 s. ISBN 80-200-1044-0.
- [0] KVASNIČKA V., POSPÍCHAL J., TIŇO P. 2000, *Evoluční algoritmy*, STU Bratislava, 2000. ISBN 85-246-2000.
- [0] ZELINKA, Ivan, *New Optimization Techniques in Engineering / kap.7 "SOMA - Self Organizing Migrating Algorithm"*, Springer-Verlag, 2004. ISBN 3-540-20167X.
- [0] KOZA J.R., *Genetic Programming*, MIT Press, 1998. ISBN 0-262-11178-6.
- [0] KOZA J.R., BENNET F.H., ANDRE D., KEANE M., *Genetic Programming III*, Morgan Kaufmann pub., 1999. ISBN 1-55860-543-6.
- [0] HOUGHTON I.L., CARPENTER P.W, *Aerodynamics for Engineering Students*, Butterworth Heinemann, 2003. ISBN 0-7506-5111-3.
- [0] *Flight dynamics* [online]. December 2006. Dostupný z WWW:  
<[http://en.wikipedia.org/wiki/Flight\\_dynamics](http://en.wikipedia.org/wiki/Flight_dynamics)>
- [0] *Učebnice Pilotů*, Svět křidel, Cheb, 2003. 612 s. ISBN 80-85280-89-2
- [0] *Trimmed Aircraft* [online]. Dostupný z WWW:  
<<http://www.grc.nasa.gov/WWW/K-12/airplane/trim.html>>
- [0] *Pointing an Instrument on an Airborne Platform* [online]. Dostupný z WWW:  
<<http://mtp.jpl.nasa.gov/notes/pointing/pointing.html>>
- [0] *An Introduction to the Airplane* [online]. Dostupný z WWW:  
<[http://www.centennialofflight.gov/essay/Theories\\_of\\_Flight/airplane/TH2.htm](http://www.centennialofflight.gov/essay/Theories_of_Flight/airplane/TH2.htm)>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

ABM	Adams-Bashforth-Moulton
ATAA	All To All Adaptive
DE	Diferenciální evoluce
ECI	Earth-Centered Inertial
GA	Genetické algoritmy
SOMA	Samo-Organizující se Migrační Algoritmus
TAS	True Air Speed

**SEZNAM OBRÁZKŮ**

<b>III</b>	<b>OBR. 1 – ROTACE KOLEM TŘÍ OS.....</b>	<b>12</b>
<b>IV</b>	<b>OBR. 2 – KŘIDÉLKA.....</b>	<b>13</b>
<b>V</b>	<b>OBR. 3 – VÝŠKOVÉ KORMIDLO.....</b>	<b>14</b>
<b>VI</b>	<b>OBR. 4 – SMĚROVÉ KORMIDLO.....</b>	<b>15</b>
<b>VII</b>	<b>OBR. 5 – ÚHEL NÁBĚHU.....</b>	<b>16</b>
<b>VIII</b>	<b>OBR. 6 – ÚHEL VYBOČENÍ.....</b>	<b>17</b>
<b>IX</b>	<b>OBR. 7 - PRINCIP ALGORITMU SOMA.....</b>	<b>24</b>
<b>X</b>	<b>OBR. 8 – SOUŘADNÁ SOUSTAVA ECI (EARTH-CENTERED INERTIAL).....</b>	<b>26</b>
<b>XI</b>	<b>OBR. 9 – LETADLOVÁ SOUŘADNÁ SOUSTAVA (AIRCRAFT BODY-FIXED AXES).....</b>	<b>27</b>
<b>XII</b>	<b>OBR. 10 – VÝCHYLKY KORMIDEL (KLADNÝ SMYSL).....</b>	<b>27</b>
<b>XIII</b>	<b>OBR. 11 – EULEROVY ÚHLY.....</b>	<b>28</b>
<b>XIV</b>	<b>OBR. 12 - ŘEZY HYPERPLOCHOU ÚČELOVÉ FUNKCE.....</b>	<b>40</b>
<b>XV</b>	<b>OBR. 13 - ŘEZY HYPERPLOCHOU ÚČELOVÉ FUNKCE.....</b>	<b>41</b>
<b>XVI</b>	<b>OBR. 14 – VÝCHYLKY HODNOT JEDNOTLIVÝCH PARAMETRŮ... 44</b>	
<b>XVII</b>	<b>OBR. 15 – HODNOTY ÚČELOVÉ FUNKCE.....</b>	<b>44</b>
<b>XVIII</b>	<b>OBR. 16 – GRAFY ČETNOSTÍ HODNOT PARAMETRŮ .....</b>	<b>45</b>
<b>XIX</b>	<b>OBR. 17 – GRAFY ČETNOSTÍ HODNOT PARAMETRŮ.....</b>	<b>46</b>
<b>XX</b>	<b>OBR. 18 - VÝCHYLKY HODNOT JEDNOTLIVÝCH PARAMETRŮ.....</b>	<b>49</b>
<b>XXI</b>	<b>OBR. 19 – HODNOTY ÚČELOVÉ FUNKCE.....</b>	<b>49</b>
<b>XXII</b>	<b>OBR. 20 - GRAFY ČETNOSTÍ HODNOT PARAMETRŮ.....</b>	<b>50</b>
<b>XXIII</b>	<b>OBR. 21 - GRAFY ČETNOSTÍ HODNOT PARAMETRŮ.....</b>	<b>51</b>
<b>XXIV</b>	<b>OBR. 22 - VÝCHYLKY HODNOT JEDNOTLIVÝCH PARAMETRŮ.....</b>	<b>54</b>
<b>XXV</b>	<b>OBR. 23 - HODNOTY ÚČELOVÉ FUNKCE.....</b>	<b>54</b>
<b>XXVI</b>	<b>OBR. 24 - GRAFY ČETNOSTÍ HODNOT PARAMETRŮ.....</b>	<b>55</b>
<b>XXVII</b>	<b>OBR. 25 - GRAFY ČETNOSTÍ HODNOT PARAMETRŮ.....</b>	<b>56</b>

**SEZNAM TABULEK**

<b>XXVIII</b>	<b>TABULKA 1 – HODNOTY MINIMALIZOVANÝCH PARAMETRŮ PŘI SIMULAČNÍCH POKUSECH.....</b>	<b>42</b>
<b>XXIX</b>	<b>TABULKA 2 - HODNOTY MINIMALIZOVANÝCH PARAMETRŮ PŘI SIMULAČNÍCH POKUSECH.....</b>	<b>43</b>
<b>XXX</b>	<b>TABULKA 3 - LEGENDA.....</b>	<b>44</b>
<b>XXXI</b>	<b>TABULKA 4 – HODNOTY MINIMALIZOVANÝCH PARAMETRŮ PŘI SIMULAČNÍCH POKUSECH .....</b>	<b>47</b>
<b>XXXII</b>	<b>TABULKA 5 – HODNOTY MINIMALIZOVANÝCH PARAMETRŮ PŘI SIMULAČNÍCH POKUSECH.....</b>	<b>48</b>
<b>XXXIII</b>	<b>TABULKA 6 - HODNOTY MINIMALIZOVANÝCH PARAMETRŮ PŘI SIMULAČNÍCH POKUSECH.....</b>	<b>52</b>
<b>XXXIV</b>	<b>TABULKA 7 - HODNOTY MINIMALIZOVANÝCH PARAMETRŮ PŘI SIMULAČNÍCH POKUSECH.....</b>	<b>53</b>

## SEZNAM PŘÍLOH

Příloha P I: Výstup programu trim

Příloha P II: Výstup z Mathematicy – soma

## PŘÍLOHA P I: VÝSTUP PROGRAMU TRIM

```
c:\Projects\_bin\WIN32\DynModelTrim\DynModelTrimTest.exe
Ini store initialization...DONE
Geoengine initialization...DONE
Atmosphere initialization...DONE
Engine initialization...DONE
DynModel initialization...DONE

Zadej hmotnost letadla [kg]: 4700
Zadej rychlost letadla TAS [m/s]: 150
Zadej vysku letu [m]: 5000

Zadej rychlost bocniho vetru [m/s]: 30

PRIMARNI RIZENI:
Poloha paky ovladani motoru: 23.6496
Uchylka kridelek [deg]: -3.55807
Uchylka vyskovky [deg]: 2.25134
Uchylka smerovky [deg]: 18.5703

STAVOVE VELICINY:
Rychlost TAS [m/s]: 150
Uhel nabehu [deg]: 1.90126
Uhel vybočení [deg]: 11.537
Uhlova rychlost klonení [deg/s]: 0
Uhlova rychlost klopení [deg/s]: 0
Uhlova rychlost zatačení [deg/s]: 0
Naklon [deg]: 24.2398
Podelny sklon [deg]: 1.90126
Vyska [m]: 5000

DERIVACE STAVOVYCH VELICIN:
Derivace rychlosti [m/s^2]: -4.19507e-09
Derivace uhlu nabehu [deg/s]: -2.0745e-07
Derivace uhlu vybočení [deg/s]: 1.25166e-07
Derivace uhlove rychlosti klonení [deg/s^2]: 2.35611e-07
Derivace uhlove rychlosti klopení [deg/s^2]: -6.68594e-08
Derivace uhlova rychlosti zatačení [deg/s^2]: -4.45695e-08
Rychlost stoupaní [m/s]: -11.8802

HMOTNOST, POLOHA TEZISTE A MACHOVO CISLO:
Machovo cislo [1]: 0.467976
Hmotnost letadla [kg]: 4700
Poloha teziste [%SAT]: 26.448

PARAMETRY UYVAZENI:
Pocet iteraci [1]: 1502
Hodnota vahove funkce [1]: 8.84853e-16
Chcete opakovat vypocet [a/n]:
```



## PŘÍLOHA P II: VÝSTUP Z MATHEMATICY – SOMA

Population has been initialized							
5.10288×10 <sup>7</sup>	2.71675×10 <sup>7</sup>	7.63372×10 <sup>6</sup>	5.54191×10 <sup>7</sup>	4.57071×10 <sup>7</sup>	2.53493×10 <sup>7</sup>	2.9008×10 <sup>7</sup>	5.14879×10 <sup>7</sup>
121.801	53.1099	65.5775	126.402	91.0372	38.9361	57.2555	84.1004
-0.173972	0.0105895	-0.147051	0.119728	-0.163537	-0.0262065	0.0826477	0.197557
0.499713	0.39689	0.0555144	-0.0844909	-0.264612	0.374351	0.442556	0.316411
0.592432	0.538482	0.0079424	-0.567904	-0.465938	0.456766	-0.131001	-0.308348
-0.0799766	0.293185	0.016254	0.205659	-0.250408	0.31941	0.0817694	-0.0648204
0.116934	0.182151	-0.189217	0.240331	-0.203601	0.395177	-0.258228	0.139342
-0.000490576	-0.141225	0.922596	-0.136869	-0.401605	-0.278998	0.798747	0.384245
Final population is							
0.00087299	0.0119445	0.00143201	0.00332995	0.0636061	0.00341118	0.0152055	0.00343386
71.9917	71.9917	71.9917	71.9917	71.9917	71.9917	71.9917	71.9917
-0.0258358	-0.0258358	-0.0258358	-0.0258358	-0.0258358	-0.0258358	-0.0258358	-0.0258358
0.05594	0.05594	0.05594	0.05594	0.05594	0.05594	0.05594	0.05594
0.130031	0.130031	0.130031	0.130031	0.130031	0.130031	0.130031	0.130031
-0.00848002	-0.00848002	-0.00848002	-0.00848002	-0.00848002	-0.00848002	-0.00848002	-0.00848002
-0.1536	0.0962813	-0.00507759	-0.0733958	0.00639406	-0.0379255	0.291993	-0.119563
0.554328	0.554328	0.554328	0.554328	0.554328	0.554328	0.554328	0.554328
Best individual is on position 12 with CV +3.91550×E-4 and parameters {71.9917, -0.0258358, 0.05594, 0.130031, -0.00848002, -0.152472, 0.554328}							