

UNIVERZITA OBRANY

FAKULTA VOJENSKÝCH TECHNOLOGIÍ

Katedra leteckých elektrotechnických systémů

Studijní program: Vojenské technologie
Studijní obor: Letecké elektrotechnické systémy

Ev. číslo: 1381/15



DIPLOMOVÁ PRÁCE

**Analýza identifikačních metod pro identifikaci parametrů
dynamického modelu**

Autor: rtm. Bc. Pavel Pospíšil
Vedoucí práce: Prof. Ing. Rudolf Jalovecký, CSc.

Brno 2015

Schvaluji:

Studijní program: Vojenské technologie

Vedoucí katedry:

Studijní obor: Letecké elektrotechnické systémy

plk. doc. Ing. Miloš Andrlé, CSc.

V Brně dne: 28. 11. 2014

Katedra leteckých elektrotechnických systémů

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: rtm. Bc. Pavel POSPÍŠIL

Téma: **Analýza identifikačních metod pro identifikaci parametrů dynamického modelu**

Vedoucí BP:

Prof. Ing. Rudolf Jalovecký, CSc.

Konzultant:

Začátek práce dne:

1. 2. 2015

Ukončení práce dne:

9. 6. 2015

Zadání přijal dne:

28. 11. 2014

.....
Podpis studenta

Pokyny pro zpracování diplomové práce

I. Obsah práce (výčet úkolů, které je nutno zpracovat a obhájit) a základní údaje pro DP

Zpracujete přehled identifikačních metod vhodných pro řešení identifikace parametrů přenosových funkcí chování člověka při řízení letu letounu. V maximální míře využijte algoritmy a knihovny funkcí dostupných v simulačním prostředí MATLAB®.

Práce bude obsahovat:

- Přehled matematických metod identifikace parametrů přenosových funkcí chování člověka při řízení letu letounu.
- Přehled funkcí ve vývojovém prostředí MATLAB® z hlediska identifikačních metod.
- Výběr identifikačních metod z hlediska vývojového prostředí MATLAB®.
- Návrh vhodného normování naměřené datové struktury pro obecné použití v identifikačních metodách.
- Porovnání vybraných identifikačních metod na konkrétních naměřených údajích chování člověka při řízení letu letounu.

Dodržujte pravidla pro citace, která jsou doporučena normou ISO 690, viz Kratochvíl, Jiří, Sejk, Petr, Anthová, Věra, Stehlík, Marek. Metodika tvorby bibliografických citací, http://is.muni.cz/do/rect/el/estud/lf/js10/metodika/web/ebook_citace.html Popřípadě využijte elektronický generátor citací na <http://generator.citace.com/>. Dodržujte pravidla pro psaní matematických vztahů, která jsou doporučena normami ČSN, viz JULÁNKOVÁ, Eva. Rovnice, jednotky a veličiny – Jak s nimi? Dostupné z: [http://web.natur.cuni.cz/cho/images/stories/ProAutory/chem_listy_99\(2005\)250.pdf](http://web.natur.cuni.cz/cho/images/stories/ProAutory/chem_listy_99(2005)250.pdf)

II. Doporučená literatura

- [1] VROŽINA, Milan, Zora JANČÍKOVÁ a Jiří DAVID. Identifikace systémů. Vyd. 1. Ostrava: Vysoká škola báňská - Technická univerzita, 2012, 1 CD-ROM. ISBN 978-80-248-2594-6.
- [2] Help vývojového prostředí MATLAB®
- [3] MATLAB® Documentation – MathWorks – <http://www.mathworks.com/products/help/matlab/>
- [4] Internet

V Brně dne 21. 11. 2014

.....
Vedoucí DP (podpis)

DP převzata vedoucím dne

DP odevzdána oponentovi dne

Poděkování

Na tomto místě bych chtěl poděkovat vedoucímu diplomové práce profesoru Ing. Rudolfu Jaloveckému, CSc., za odborné vedení a mnoho užitečných rad týkajících se tématu mé diplomové práce. V neposlední řadě bych rád poděkoval rodině a přátelům za jejich podporu nejen při tvorbě této práce, ale i během mého studia.

Prohlášení

Ev. číslo: 1381/15

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a vyznačil v ní všechny prameny, které jsem použil. Jsem si vědom následků nepravdivosti těchto údajů.

rtm. Bc. Pavel Pospíšil

.....

Datum, hodnost, jméno, příjmení

ABSTRAKT

Tato diplomová práce je zaměřena do oblasti analýzy metod pro identifikaci parametrů dynamických modelů. Metody pro analýzu byly vybrány z možných variant použití pro identifikaci parametrů v prostředí MATLAB® a také z jeho System Identification Toolboxu. Právě pro tyto jednotlivé identifikační metody byly vyhledány matematická řešení a provedena jejich následná analýza na datech, které byly získány z měření na softwaru X-Plane 10 a simulátoru na katedře leteckých elektrotechnických systémů. Další data, která byla použita k analýze, pocházela z programu Měření odezvy pilota (MOP). Z takto analyzovaných dat byly vyhodnoceny závěry pro vhodnost použití jednotlivých metod pro další zpracování a jejich použití například pro rozšíření programu MOP o výběr jednotlivých metod.

Klíčová slova: identifikační metody, System Identification Toolbox, parametry dynamického modelu, MATLAB®, odezva na jednotkový skok, analýza

ABSTRACT

This thesis is focused in analysis methods for parameter identification of dynamic models. Methods for the analysis were chosen of the alternatives used in the identification parameters in MATLAB® and also from the System Identification Toolbox. It is for these individual identification methods were searched mathematical solutions and their subsequent analysis performed on measured data obtained from X-Plane 10 software simulator at the Department of Aerospace Electrical Systems. Other data that was used to analyse was from the program Měření odezvy pilota (MOP). From these data were analysed to evaluate the conclusions of the appropriateness of the methods used for further processing and use their example to expand the MOP program on the selection of individual methods.

Keywords: identification methods, System Identification Toolbox, the parameters of the dynamic model, MATLAB®, step response analysis

Obsah

Seznam obrázků.....	10
Seznam použitých zkratek a symbolů.....	11
Úvod.....	12
1. Základní pojmy.....	14
1.1. Systém.....	14
1.2. Modelování	14
1.3. Simulace.....	15
1.4. Identifikace.....	16
1.5. Dynamický model a jeho parametry	16
2. Model chování pilota	20
2.1. Blokové schéma	20
2.2. Matematický popis.....	21
3. Využití autorských programů pro sběr dat	23
3.1. Simulátor X-PLANE 10.....	23
3.1.1. Charakteristika simulátoru.....	23
3.1.2. Využití simulátoru X-Plane 10.....	25
3.2. Microsoft Flight Simulator X.....	26
3.3. FlightGear Flight Simulator	26
3.4. MOP – Software Měření Odezvy Pilota	26
3.4.1. Popis měřicího pracoviště.....	27
3.4.2. Využití naměřených dat.....	27
3.5. HUFA_M – Program pro analýzu naměřených dat chování člověka při řízení letounu.....	29
4. Identifikační metody.....	32
4.1. Fourierova transformace	32
4.2. Metoda nejmenších čtverců	33
4.3. Nelder-Mead simplex algoritmus.....	34
4.4. Gauss-Newtonova metoda (gn).....	36
4.5. Levenberg-Marquardtova metoda (lm).....	37
4.6. Trust-Region Reflective Newton metoda (lsqnonlin).....	39
4.7. Gradient search (grad).....	40

5. System Identification Toolbox	42
5.1. Data pro identifikaci.....	43
5.2. Použití System Identification Toolboxu	44
5.3. Analýzy jednotlivých identifikačních metod	46
5.4. Algoritmizace analýzy datových struktur	55
Závěr	60
Literatura.....	62
Zdroje obrázků.....	65
Přílohy.....	66

Seznam obrázků

Obr. 1: Přejchodová charakteristika kmitavé soustavy 2. řádu, [I.].....	18
Obr. 2: Přejchodová charakteristika integrační soustavy 1. řádu, [II.]	18
Obr. 3: Přejchodová charakteristika reálné derivační soustavy, [III.]	19
Obr. 4: Blokové schéma modelu chování pilota, [IV.].....	21
Obr. 5: Pilotní kabina letounu King Air, [V.].....	24
Obr. 6: Stránka nastavení okna "Data Input & Output", [VI.].....	25
Obr. 7: Princip měření odezvy člověka na vnější podnět, [VII.]	27
Obr. 8: Software MOP – hlavní okno, [VIII.]	28
Obr. 9: Hlavní okno programu HUF_A_M, [IX.]	29
Obr. 10: Okno průběhu identifikace, [IX.]	30
Obr. 11: Výběr metod v nastavení System Identification Toolbox, [X.].....	32
Obr. 12: Schématický diagram funkce v \mathbb{R}^2 a několika iterací, [XI.].....	35
Obr. 13: Příklad algoritmu gradient descent, [XII.].....	40
Obr. 14: Grafické uživatelské prostředí System Identification Toolbox, [X.]	42
Obr. 15: Skoková změna výšky a následná reakce pilota, [X.]	43
Obr. 16: Okna Process Models a volby a nastavení identifikačních metod, [X.].....	45
Obr. 17: Okno Transfer Function Models, [X.].....	46
Obr. 18: Modelový soubor dat test_001 pro analýzu jednotlivých metod, [X.]	47
Obr. 19: Naměřená a vypočítaná data pomocí metody: Gauss-Newton, [X.]	48
Obr. 20: Naměřená a vypočítaná data pomocí metody: Adaptive Gauss-Newton, [X.]	49
Obr. 21: Naměřená a vypočítaná data pomocí metody: Levenberg-Marquardt, [X.]....	50
Obr. 22: Naměřená a vypočítaná data pomocí metody: Gradient Search, [X.]	51
Obr. 23: Srovnání metod pomocí funkce Process Models. [X.].....	52
Obr. 24: Srovnání metod pomocí funkce Transfer Function Models – Discrete time...	53
Obr. 25: Srovnání metod pomocí funkce Transfer Function Models – Continuous time	54
Obr. 26: Výpis výsledků identifikace	54
Obr. 27: Okno Compare při ovládání Systém Identification Toolboxu přes Command window [X.].....	59

Seznam použitých zkratk a symbolů

DLS	Damped least-squares	
FAA	Federal Aviation Administration	Federální letecký úřad USA
FPE	Final Prediction Error	Akaikeho konečná chyba predikce
GNA	Gauss–Newton algorithm	Gauss–Newtonův algoritmus
GUI	Graphical User Interface	Grafické uživatelské rozhraní
HUFA	Human Factor	Zkratka autorského programu HUFA
IFR	Instrument Flight Rules	Let podle přístrojů
ILS	Instrument landing system	Systém pro přistání podle přístrojů
LAN	Local Area Network	Lokální síť
LMA	Levenberg-Marquardth algorithm	Levenberg-Marquardtův algoritmus
MOP		Měření odezvy pilota
MSE	Mean Squared Error	Střední kvadratická chyba
SOP	Standard operation procedure	Létání podle standardních provozních postupů
TRR	Trust-Region reflective method	
UAV	Unmanned Aerial Vehicle	Bezpilotní létající prostředek

E	kvadratická odchylka
$F_{(p)}$	přenosová funkce
H	výška letu
J	Jacobián
K	zesílení
p	Laplaceův operátor
\mathbf{R}^n	n -rozměrný vektorový prostor reálných čísel
T_1	reakční časová konstanta
T_2	zpožďující časová konstanta
T_3	integrační časová konstanta
$X_{(p)}$	jmenovatel přenosové funkce
$Y_{(p)}$	čitatel přenosové funkce
δ_v	výchylka výškového kormidla
τ	konstanta dopravního zpoždění

Úvod

Téma mé diplomové práce jsem si zvolil analýzu identifikačních metod pro identifikaci parametrů dynamického modelu. Bylo to z důvodu nahlédnutí do zajímavé problematiky, která je součástí jak výuky na katedře, tak úzce souvisí s volbou mého budoucího povolání a to jako pracovník inženýrsko-letecké služby na jednom z vojenských letišť.

Jedním z hlavních předpokladů pro efektivní řízení zvoleného objektu je znalost vlastností, kterými daný objekt disponuje. Pro neoptimálnější řízení celého systému, je nutné dokonale znát vlastnosti dílčích řízených objektů. Z tohoto důvodu je kladen velký význam na tvorbu matematických modelů systémů, jejich objektů, výběru vhodných identifikačních algoritmů a volbou parametrů systému.

Matematické modelování těchto systémů za pomoci počítačové simulace se neomezuje pouze na řízení, ale zasahuje do mnohých dalších technických i vědních disciplín jako je biologie, ekonomika, a stala se tak nedílnou součástí návrhů mnoha složitých zařízení a to bez rizika tvorby havarijních stavů a s využitím mnohem menších nákladů z důvodu nepotřebnosti reálného objektu.

I v dnešním světě, kde je ve většině letadlových systémech a zařízení určitý a nezanedbatelný podíl automatizace, který s postupem času narůstá, je pro zabezpečení vyšší spolehlivosti, tak i zvýšení bezpečnosti celého systému nutnost znát do určité míry různé fyzikální, technické a systémové vlastnosti systémů pro jejich správné řízení, kontrolu a udržování. Ve většině případů se tyto znalosti omezují do míry, která značí, do jaké hloubky známe zkoumaný prvek tzv. white, grey nebo black boxy. To je dáno zvyšující se konstrukční složitosti jednotlivých systémů, jejich prvků a nároků na výpočetní úkony pro zpracování daleko většího objemu dat. I když výkon a náročnost stoupají, je stále pilot, který je buď přítomen na palubě letounu, nebo jako operátor UAV z pozemního stanoviště, hlavní a nepostradatelný výkonný orgán v systému řízení a rozhodování.

Samotný pilot se dá pomocí různých matematických modelů méně či více přesně nasimulovat a provádět analýzu matematických metod identifikace parametrů přenosových funkcí chování člověka při řízení letu letounu. Výběrem vhodné metody a jejím proložením křivky reprezentovanou výstupními hodnotami určitého pokusu se dá velice přesně zjistit, který pilot dokáže rychle a hlavně přesně zareagovat na určitou potuchu a vyrovnat letoun do stavu v jakém byl před vznikem poruchy. Z takto analyzovaných dat se dají zjistit skutečnosti, které mohou být užitečné pro samotného pilota, který si může prověřit své zlepšení řízení letounu, kontroly a tvorby návyků, anebo pro porovnání jednotlivých uživatelů.

1. Základní pojmy

V dnešní době se identifikačními metodami a vlastní identifikací parametrů dynamických modelů zabývá mnoho autorů ve svých literaturách, které jsou ale mnohdy cizojazyčné [2], [8], [9], [12], [19], [21], [24], [25], [28]. Metody identifikace jsou také implementovány jako součásti (toolboxy) nebo jako knihovny v mnoha specializovaných programech.

Na úvod je dobré připomenout základní pojmy a jejich význam, zmíněné při tvorbě této práce.

1.1. Systém

Pojem systém můžeme chápat jako množinu prvků a vazeb mezi nimi, které nám dávají představu o chování a vlastnostech celku. Vnitřní uspořádání systému se nazývá struktura a je definována vzájemnými vazbami.

1.2. Modelování

Pojem modelování je chápán jako proces tvorby modelu podle zkoumaného originálu (reálného objektu), který popíšeme jak kvalitativně, tak kvantitativně. Námí zvolený reálný objekt si zanalyzujeme, postupně zjednodušíme, tak že vylučujeme nepodstatné vlastnosti, které by nám analýzu objektu zesložitovaly. Systém rozdělíme na dílčí podsystemy, které půjdou jednodušeji matematicky vyjádřit a popisujeme jednotlivé závislosti, charakteristické vlastnosti a chování. Takto popsané podsystemy jsou zpětně složeny do modelu. Složitost popisu je dána mírou zjednodušení modelu od reálné předlohy. Takto sestavený model se může do značné míry lišit od skutečného. Nastane-li takováto situace, je nutné vrátit se k rozložení systému na podsystemy a zpřesňovat tento model tak, aby v co největší míře odpovídal předloze.

Soubor problémů a úkonů spojených s modelováním můžeme tedy rozdělit do několika etap:

1. Přesná formulace úlohy, pro kterou se model vytváří.
2. Rozložení složitého systému na relativně samostatné podsystémy, jejichž identifikaci jsme schopni realizovat.
3. Tvorba modelů podsystémů, získaných při dekompozici, což vyjadřuje:
 - a) zvolením správných vstupních a výstupních veličin jednotlivých dílčích systémů a volbou vhodného řešení provedení výzkumu
 - b) měření zvolených vzájemně si odpovídajících vstupů a výstupů,
 - c) vyhodnocením naměřených údajů vytvořit takový model, který dostatečně přesně dokáže ke zvoleným vstupním signálům přiřazovat správné odezvy na výstupech. Zde řešíme zpravidla následující tři okruhy problémů:
 - volba vhodné struktury modelu,
 - volba kritéria na porovnání shody modelu s vyšetřovaným objektem,

Metody ztotožnění modelu s vyšetřovaným objektem jsou předmětem (cílem) vědní disciplíny, která se nazývá identifikace [25], [28].

1.3. Simulace

Počítačová simulace nebo počítačový model je počítačový program, který se pokusí simulovat abstraktní model konkrétního systému.

Počítačové simulace se staly užitečnou součástí matematického modelování mnoho přirozených systémů ve fyzice, chemii a biologii, lidských systémů, v ekonomii, psychologii a sociálních věd a v procesu strojírenské technologie, kde se snaží získat stejné vlastnosti a fungování těchto systémů.

Tradičně se formální modelování systémů děje s pomocí matematického modelu, který se snaží nalézt analytické řešení problémů, které mu umožňuje, předvídat chování systému ze sady parametrů a počátečních podmínek [3].

1.4. Identifikace

Pro identifikace systémů se používají statistické metody k vytvoření matematických modelů dynamických systémů z naměřených dat. K identifikaci systému je také zahrnut optimální návrh experimentů pro efektivní vytváření informačních dat pro určení takového modelu, který odpovídá zredukovanému objektu.

K identifikaci používáme testovací signál, který aplikujeme na daný systém a zaznamenáváme jeho odezvu. Model systému určíme z vyhodnocených naměřených dat.

Dále také rozlišujeme postup procesu identifikace systému na volbu algoritmu, který při zadané struktuře modelu minimalizuje prostřednictvím parametrů modelu hodnotu kritériální funkce [25], [28].

1.5. Dynamický model a jeho parametry

Pod pojmem dynamický model si můžeme představit takový model, který je úplný a popisuje nejen statické, ale i dynamické vlastnosti systému. Výstupem tohoto modelu je průběh závislosti výstupní veličiny v čase při daném vstupu a stavu systému. Tuto vazbu mezi vstupy a výstupy nám vyjadřují diferenciální rovnice. Použití dynamických modelů se uplatňuje v oblasti řízení, kde jsou zkoumány jevy, jako jsou přechody z jednoho stavu systému do druhého.

Podle závislosti parametrů rozlišujeme dynamické modely na:

- **stacionární** (časově nezávislé), kde jsou parametry dynamických modelů konstantní
- **nestacionární** (časově závislé), kde jsou naopak parametry dynamických modelů časově závislé.

Dále rozlišujeme dynamické modely podle způsobu identifikace na:

- **analytický model**, který získáme analytickými metodami identifikace, vycházejících z bilančních rovnic jako jsou chemické, fyzikální, atd.,
- **experimentální model** získaný z měření reálných objektů.

Podle typu matematického popisu rozlišujeme modely na:

- **nelineární**, kde alespoň jedna matematická operace popisu systému je nelineární
- **lineární**, kde všechny matematické operace jsou lineární.

Další rozdělení dynamických modelů se zabývá, zda je model **spojitý** či **diskrétní** z hlediska způsobu zpracování informace, podle obsažených parametrů ve funkčních závislostech na **parametrický**, **neparametrický**, dále pak na **vnitřní**, **vnější**, **konvenční** a **nekonvenční** [25].

Rozdělení lineárních dynamických soustav

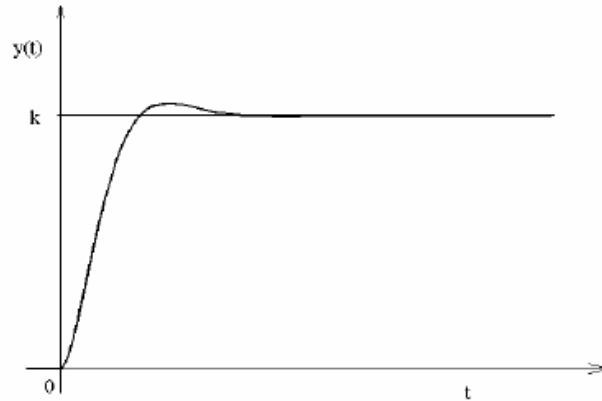
Rozdělení lineárních dynamických soustav lze provést podle několika kritérií. Jedním z těch důležitých je to, jaké má vlastnosti ustálený stav, zda existuje nebo neexistuje. Tyto vlastnosti se posuzují z průběhu přechodové charakteristiky. Na základě tohoto hlediska rozdělujeme soustavy na proporcionální, integrační a derivační.

- **Proporcionální soustava** – při vychýlení z ustáleného stavu se sama ustálí na hodnotě, která je nenulová. Takovéto soustavy se vyznačují tím, že v čitateli a jmenovateli přenosu $G(p)$ nelze vytknout komplexní proměnnou p .

Na příkladu, kdy máme soustavu proporcionálního členu se setrvačností 2. řádu je patrné, že o průběhu přechodové charakteristiky rozhoduje rozložení pólů a nul v komplexní rovině, které jsou závislé na hodnotě diskriminantu D při řešení charakteristické rovnice. Jsou-li kořeny charakteristické rovnice komplexně sdružené, pak tvar přenosu je vyjádřen ve tvaru:

$$G(p) = \frac{k}{T_0^2 p^2 + 2\xi T_0 p + 1} \quad (1)$$

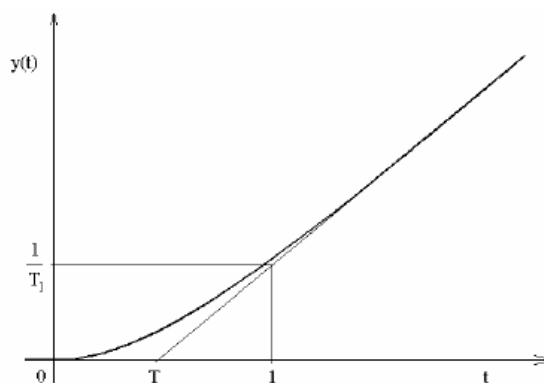
Kde: T_0 je perioda kmitání a ξ je koeficient tlumení.



Obr. 1: Přechodová charakteristika kmitavé soustavy 2. řádu, [I.]

- **Integrační soustava** – při vychýlení z ustáleného stavu se tato soustava již neustálí v novém stabilním stavu. Integrační soustavy se vyznačují tím, že ve jmenovatelích jejich přenosů $G(p)$ lze vytknout komplexní proměnnou p .

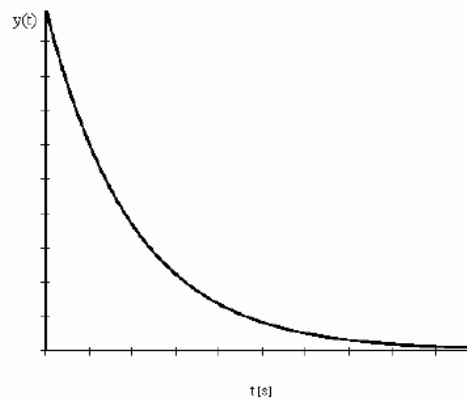
$$G(p) = \frac{k}{T_1 p (T_1 p + 1)} \quad (2)$$



Obr. 2: Přechodová charakteristika integrační soustavy 1. řádu, [II.]

- **Derivační soustava** – u tohoto typu soustavy je ustálený stav nulový. Přenosy derivačních soustav se vyznačují tím, že v jejich čitatelích lze vytknout komplexní proměnnou p .

$$G(p) = T_D * p \quad (3)$$



Obr. 3: Přejchodová charakteristika reálné derivační soustavy, [III.]

2. Model chování pilota

I v dnešní době, kdy je většina zařízení řízena digitálními počítači nebo systémy s umělou inteligencí, které tedy nejsou schopny myslet samostatně, ale posloupnost úkonů definují podle vyhodnocení situace a znalostí možných variant postupů uložených v databázi, je člověk stále nejdůležitějším prvkem v systému řízení letadla. V budoucnu, kdy počítače a jejich naprogramování bude tak dokonale popisovat, jak chování pilota, tak všechny reakce na situace, které mohou nastat, nebude už pilota za potřebí. Dnes je stále ještě pilot nedílnou součástí celého souboru už od nasednutí do kokpitu, kontroly před letem, pilotáží a velice fyzicky i psychicky náročným přistáním [1], [11].

2.1. Blokové schéma

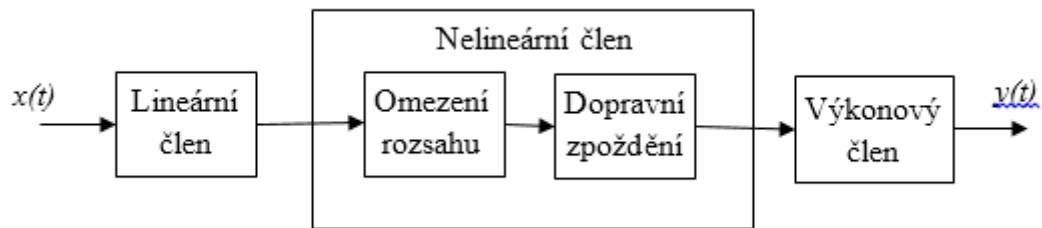
Už dříve se objevovaly modely chování člověka, ale tyto modely byly z dnešního pohledu pokročilých simulačních systémů velice zjednodušené a orientační.

Vlastnosti pilota se mohou charakterizovat různě složitými blokovými schémata, které více či méně přesně vystihují většinu vlivů ovlivňujících chování člověka. To je dáno změnou parametrů a časových konstant, které se mění s časem a ještě jsou ovlivňovány dalšími faktory, jako je např.: vycvičenost pilota, únava, hluk a stres. Některé z těchto nežádoucích jevů, které ovlivňují tvorbu modelu, lze zmírnit studiem a tréninkem pro osvojení návyků. Proto nelze vytvořit univerzální náhradní model chování pilota z pohledu automatické regulace, který by úplně charakterizoval dynamické vlastnosti v rozmanitých úlohách řízení [1], [11].

Zjednodušeně, ale z dostupných materiálů velmi výstižně, lze blokově (Obr.: 4) vyjádřit model chování pilota jako sériové spojení bloku lineárního, nelineárního a výkonového členu.

- **Lineární člen** – smyslové orgány, přes které vstupují informace do nervové soustavy pilota.

- **Nelineární člen** – který je rozdělený na blok omezení rozsahu a dopravní zpoždění. První blok je charakterizován délkou paže a nohy a druhý je vlastní reakce člověka na rušivý podmět (rychlost šíření informace v nervové soustavě člověka v rozmezí $5 - 125 \text{ ms}^{-1}$).
- **Výkonový člen** – výstup bloku charakterizuje sílu, kterou působí pilot na řízení.



Obr. 4: Blokové schéma modelu chování pilota, [IV.]

Pilot využívá tři různé regulátory pro získání požadovaného průběhu letu:

- **Prediktivní regulátor** – který udržuje požadovaný průběh letu z informací o pohybu letounu získaných ze zrakového ústrojí a vlastních pocitů.
- **Zpětnovazební regulátor** – který obsahuje vizuální informace a pocitové vjemy z letu vůči požadovanému průběhu.
- **Prekognitivní regulátor** – který vyvolává z paměti pilota naučený manévr a ten má za následek požadovaný pohyb letounu.

2.2. Matematický popis

Jak již bylo řečeno v předchozí kapitole, v systému řízení může být člověk do určité míry velice přesně reprezentován jako sestava lineárního, nelineárního a výkonového členu. Popsat takto pilota matematicky je o dost složitější, protože nejsou známy všechny biologické a fyziologické procesy lidského mozku. Z této neznalosti nemůžeme vytvořit funkční popis procesu myšlení využitelný při řízení letounu.

V literatuře se používá lineární model s dopravním zpožděním (4) charakterizovaný přenosovou funkcí:

$$F_{(p)} = \frac{Y_{(p)}}{X_{(p)}} = K \frac{(T_3 p + 1)}{(T_1 p + 1)(T_2 p + 1)} e^{-\tau p} \quad (4)$$

kde:

K – zesílení reprezentující následný přírůstek síly na kormidla při zásahu pilota v reakci na změnu dynamiky soustavy (1 - 100).

T₁ – reakční časová konstanta, udává zpoždění, které je dáno reakcí na vzniklou změnu vstupního signálu a tím i provedení rutinních a naučených činností.

T₂ – zpožďující setrvačná konstanta, udává zpoždění činností způsobené funkcí neuromuskulárního systému.

T₃ – integrační časová konstanta, schopnost pilota předvídat příchozí situaci a přizpůsobit na ně další činnost. Tato schopnost se získává ze zkušeností pilota.

τ – tato časová konstanta udává dopravní zpoždění pilota na pohybový a oční vjem. Velikost dopravního zpoždění se zvětšuje s rostoucí únavou.

Lineární model chování pilota, je ve skutečnosti pouze zjednodušení reálných vlastností, které jsou dynamické a ještě zatíženy různými vlivy jako je necitlivost, hystereze a mnoho dalšího [1], [11].

3. Využití autorských programů pro sběr dat

V této kapitole jsou uvedeny některé druhy autorských programů, za pomoci nichž lze z předem zvolené simulace a vkládáním umělých ruchů, získávat data o reakci uživatele (v našem případě pilota) na vzniklý podnět a takto nasnímaná data ukládat pro pozdější vyhodnocování.

Jedná se o komerční software v podobě na trhu dostupných placených i neplacených tzv. open source leteckých simulátorů (X-Plane, Microsoft Flight Simulator, FlightGear), ale i samostatných programů, které byly speciálně vytvořeny za účelem sběru a vyhodnocení dat (MOP, HUFA).

3.1. Simulátor X-PLANE 10

Tato kapitola je rozdělena na část, která se zabývá stručnou charakteristikou a vlastnostmi leteckého simulátoru X-Plane a částí zaměřenou na konkrétní využití programu pro účel sběru a vyhodnocování dat.

3.1.1. Charakteristika simulátoru

Software X-Plane 10 patří mezi světově nejvíce rozšířené letecké simulátory vhodné pro osobní počítače, které jsou nabízené na trhu a jsou určeny pro letecké nadšence i profesionály. Tento software není jen počítačová hra, ale vysoce precizní inženýrský nástroj, který díky realistickým modelům vzdušných prostředků s pevnými i rotačními vztlakovými plochami, umožňuje provádět předpovědi chování použitých prostředků a konstrukčních řešení při různých podmínkách použití s vysokou přesností. Díky těmto vlastnostem je X-Plane 10 vhodný nástroj pro piloty, kteří si potřebují osvojit a vylepšovat určité návyky pilotáže SOP, reakce na vzniklé situace v průběhu letu, let podle přístrojů IFR, přistání pomocí ILS, ale i pro konstruktéry, kteří si mohou navrhnout model vložit do prostředí, simulovat různé stavy a testovat si jeho vlastnosti.

Základní balíček softwaru X-Plane 10 obsahuje 30 modelů letounů od malých podzvukových sportovních kluzáků až po nejmodernější nadzvukové stíhací letouny. Oblíbenost softwaru mezi obyčejnými uživateli podtrhuje i to, že si mohou navrhnout nejen vlastní model letounu s utilitou Plane Maker a rovnou si ho i otestovat, ale i navrhnout a tvořit prostředí jako je okolní krajina, vlastní letiště nebo s pomocí Airfoil makeru testovat vlastní profily křídel.



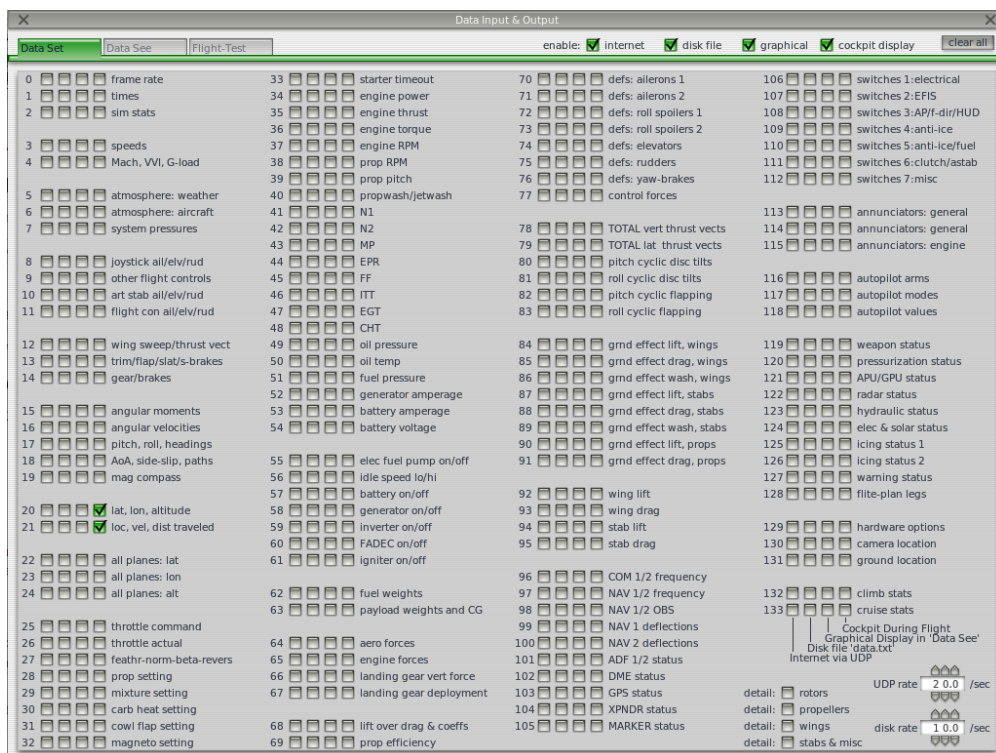
Obr. 5: Pilotní kabina letounu King Air, [V.]

Celé prostředí softwaru X-Plane 10 je vyvedeno ve vysokém rozlišení a nabízí uživateli přes 33000 letišť z celého světa a nabízí manuální nebo zcela automatickou změnu počasí podle aktuální předpovědi a tvorbu poruch na letounu jak náhodných, tak i definovaných podle toho, pro koho je zrovna simulovaná situace určena. Dále si uživatel může navolit nespočet různých scénářů, jako je let s rádiem řízeným modelem, hašení lesních požárů, lety do vesmíru, vzdušné souboje nebo si může navrhnout zcela vlastní situace. Tyto všechny scénáře se mohou odehrávat na jednom počítači nebo spojit více uživatelských stanic pomocí sítě LAN na síť s jedním instruktorem a skupinou žáků. Software X-Plane

10, je plně certifikovaný FAA a je také využíván jako nástroj pro vyšetřování leteckých nehod [17].

3.1.2. Využití simulátoru X-Plane 10

Na katedře Leteckých elektrotechnických systémů se nachází pracoviště, které je tvořeno instruktorskou stanicí a samotným simulátorem složeným z několika monitorů, kniplu, pedálů a dalších ovládacích prvků. Toto pracoviště slouží především pro sběr dat, informací o letu a výzkumné účely.



Obr. 6: Stránka nastavení okna "Data Input & Output", [VI.]

Důležitou a velmi užitečnou funkcí programu X-Plane 10, je možnost využití okna "Data Input & Output" s nastavením několika desítek údajů pro zápis vstupních a výstupních dat ze simulátoru, které mohou být využívány pro další analýzy. Pro naše účely dalšího zpracování byly nejdůležitější data o čase letu (real,time), výchylce kormidla pomocí kniplu (elev,yoke1) a změně výšky letu (alt,ftmsl). Využití těchto údajů bude popsáno v jedné z následujících kapitol [2].

3.2. Microsoft Flight Simulator X

Dalším simulátorem dostupným na univerzitě je Microsoft Flight Simulator X. Tento produkt patří do stejné kategorie simulátorů, jako již dříve zmíněný X-Plane 10, a proto ho není nutné zdlouhavě popisovat. Simulátor nabízí jak detailní provedení modelů různých druhů a typů letadel, tak i letišť a prostředí se satelitními mapami. I zde je možná tvorba scénářů letu, procvičování a zdokonalování letových návyků.

Toto programové vybavení slouží především pro studenty pilotního zaměření, kteří se seznamují s obsluhou jednotlivých ovládacích prvků v pracovišti pilota. Program také nabízí tvorbu výstupních dat. Možnost analýzy výstupních dat z tohoto simulátoru zde ještě nebyla realizována.

3.3. FlightGear Flight Simulator

Velmi oblíbeným a rozšířeným nástrojem je také simulátor FlightGear, který je dostupný jako open source – to znamená, že je volně dostupný a uživatelsky velmi přizpůsobivý z hlediska tvorby vlastních prostředků a mnoho dalšího. Tento simulátor je využíván nejen k zábavě, ale i k akademickým a výzkumným účelům.

Tento produkt má klíčové vlastnosti obou předchozích jmenovaných simulátorů jako jsou realistický model letounu a jeho vlastností, mapový terén, změna počasí a denní doby. Zajímavostí je varianta použití simulátoru i jako možnost uživateli stát se řídicím letového provozu. Tento produkt nebyl dostupný na univerzitě a nemohla být prozkoumána možnost sběru a analýzy dat pro další zpracování.

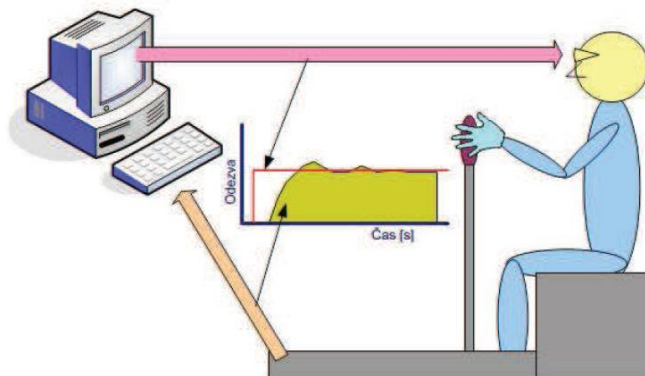
3.4. MOP – Software Měření Odezvy Pilota

MOP – Software měření odezvy pilota patří do zmiňované skupiny programů speciálně vytvořených za účelem měření a vyhodnocování dat, konkrétně měření a vyhodnocování odezvy pilota na oční vjem [8].

3.4.1. Popis měřicího pracoviště

Rozložení pracoviště je velice jednoduché. Skládá se z počítače s programovým vybavením MOP, připojeným joystickem a tiskárnou pro tisk protokolů. Taková to konfigurace měřicího pracoviště je praktická, velice mobilní a hodí se pro sběr dat z různých míst, jako jsou samotné letecké základny nebo pro dynamické ukázky na různých akcích.

Princip získávání dat pro další analýzu je takový, že testovaný subjekt sleduje na počítači vizuální podnět, v našem případě to byla skoková změna a snaží se výchylkou joysticku sesouhlasit aktuální pozici bodu s nastavenou. Vstupním signálem je křivka, která se mění skokově, přičemž tato změna vzniká náhodně. Uživatel se snaží co nejrychleji zareagovat a sesouhlasit odchylku co možná nejpřesněji [2], [9].

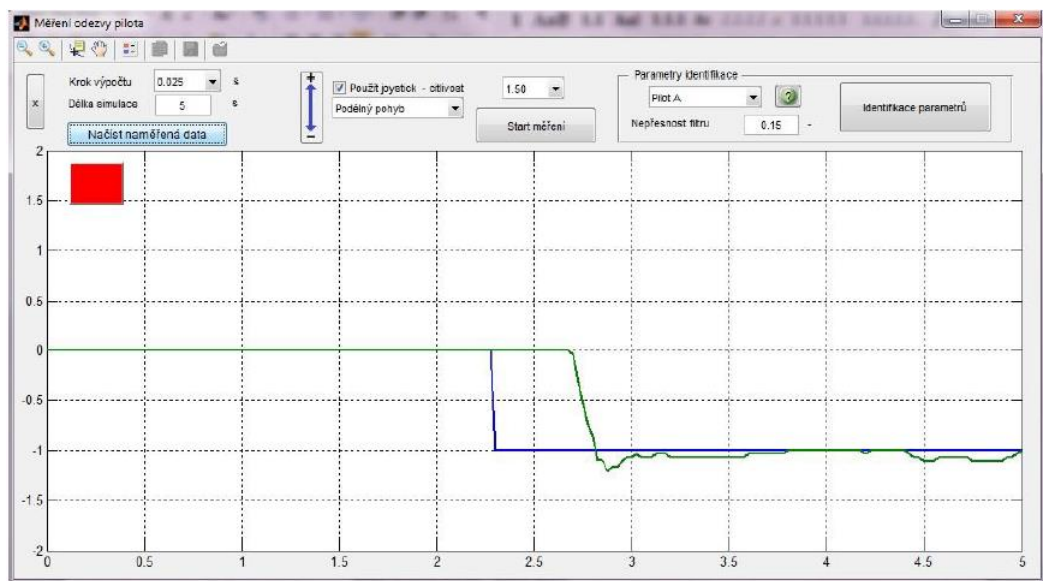


Obr. 7: Princip měření odezvy člověka na vnější podnět, [VII.]

3.4.2. Využití naměřených dat

Časový průběh výchylky kniplu je zaznamenáván spolu s rychlostí a velikostí vnějšího podnětu člověka na jednotkový skok a tyto údaje jsou dále použity pro matematické zpracování. Jednou z připravovaných možností programu MOP je nabídka s výběrem identifikačních metod, které jsou jinak implementovány v System Identification Toolboxu.

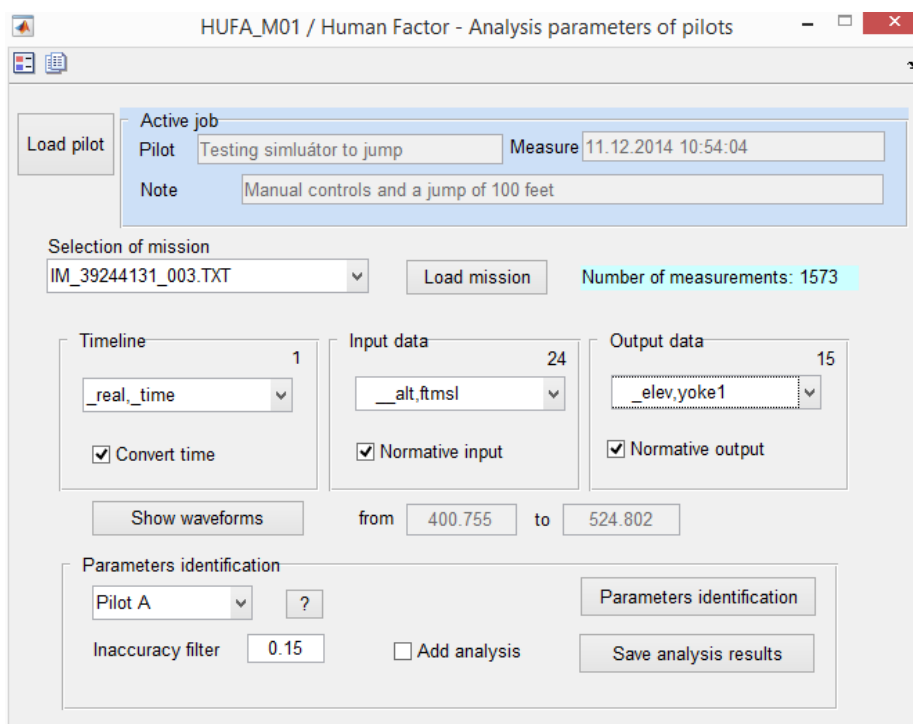
Výstupem z programu je okno s grafickým zobrazením výsledku identifikace parametrů daného měření a zvoleného modelu chování pilota. V grafu je pak patrné proložení křivky zvolenou metodou identifikace a výpočet odezvy pilota na jednotkový skok. Dále se zobrazují výsledky identifikace pro zvolený typ modelu chování pilota. Vypisují hodnoty vypočítaných časových konstant, řády čitatele a jmenovatele přenosové funkce, směrodatná odchylka, hodnota dopravního zpoždění, zesílení a počet iterací výpočtu. Takto spočítaná a vygenerovaná data se dají uložit do přehledného protokolu, který slouží jako informace pro testovaného uživatele a pro archivaci.



Obr. 8: Software MOP – hlavní okno, [VIII.]

3.5. HUFA_M – Program pro analýzu naměřených dat chování člověka při řízení letounu

Analýza dynamických systémů je matematicky náročná operace, zvláště pak při zpracovávání velkého množství naměřených údajů. Využití simulačních nástrojů, jako je MATLAB® tuto úlohu značně zjednodušuje. Pokud vhodně připravíme pro zvolenou metodu analýzy naměřené údaje do takové formy, s níž si vývojové prostředí simulačního programu MATLAB® dobře poradí, můžeme očekávat i výsledky přiměřené kvality. Vhodnou volbou prezentační formy, budou výsledky srozumitelné i mimo odbornou veřejnost. Analýza chování člověka při řízení letu letounu je jednou z mnoha aplikací, v níž lze vhodně využít prostředků, které právě simulační nástroj MATLAB® poskytuje [13].

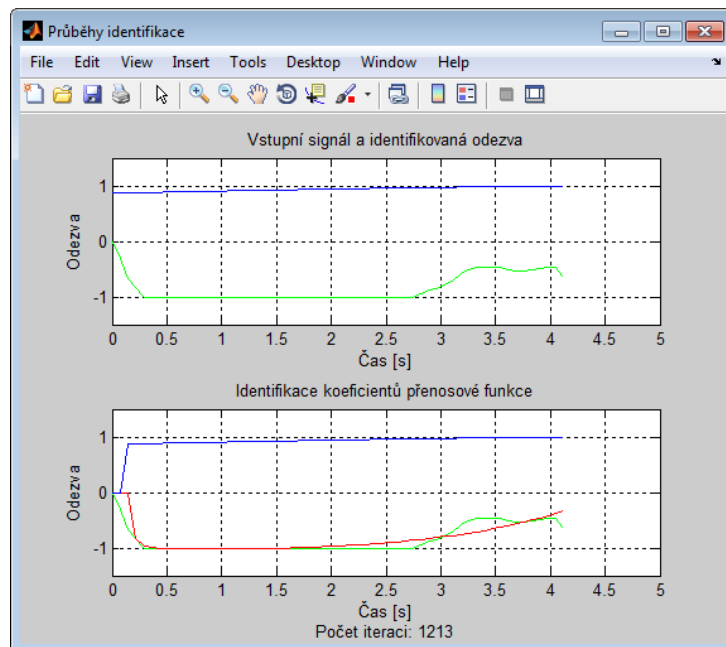


Obr. 9: Hlavní okno programu HUFA_M, [IX.]

Po přípravě dat z měření chování člověka následuje výběr metody analýzy ze dvou rozpracovaných analytických postupů a zvolit model chování pilota. Výsledkem celého procesu analýzy jsou výstupy, které představují vypočtené časové konstanty, dopravní

zpoždění, zisk lineárního členu modelu a statistické údaje, především grafický výstup s přehlednými informacemi o analýze, jako je: Průběh identifikace, rozložení pólů a nul v komplexní rovině a vyjádření přenosové funkce modelu chování pilota.

Hlavní okno programu HUFA_M (Obr.: 9) slouží k ovládání všech funkcí programu. Při spuštění programu je načten aktivní pilot, včetně aktivních misí. Obsluha programu poté volí misi (data), které načte do analytické části programu. Pro potřebu analýzy je nutné vybrat nejen údaje o časovém měřítku, ale především vstupní a výstupní data z celé řady zaznamenaných dat ze simulátoru. Takto vybraná data je možné zobrazit v okně „Graf výběr“ a zde vybrat konkrétní úsek z měření, který následně podrobíme analýze.



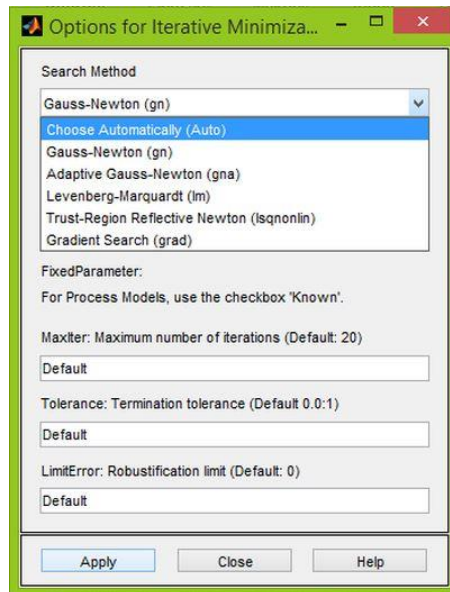
Obr. 10: Okno průběhu identifikace, [IX.]

Po spuštění analýzy je zobrazeno okno (Obr.: 10) s grafy vstupní informace a „posunutím“ skoku vstupní informace v důsledku výpočtu dopravního zpoždění. V tomto grafu se zobrazuje průběh jako křivka každého kroku iterace. Počet iterací je pak zobrazován v dolní části okna.

Všechny takto spočtené údaje se následně přenesou přes vybraný formát sešitu Excel do evidenčního programu, který umožňuje dále s výsledky pracovat a např. i vygenerovat celkový protokol z měřené mise [2], [9].

4. Identifikační metody

Tato kapitola je zaměřena na přehled a použití identifikačních metod. V tomto přehledu nejsou popsány všechny identifikační metody, ale jen ty, které jsou obsaženy v knihovnách prostředí MATLAB® a toolboxu System Identification, kde jsou zahrnuty jako parametr analýzy ve volbě identifikační metody.



Obr. 11: Výběr metod v nastavení System Identification Toolbox, [X.]

V následující části kapitoly je uveden přehled funkcí a algoritmů, které jsou použity pro jejich další analýzu. Jedná se o Fourierovu transformaci, funkci `fminsearch`, `fminunc`, ale i metody, nejčastěji využívající pro zpracování a analýzu signálu metodu nejmenších čtverců [22].

4.1. Fourierova transformace

Jean Baptiste Joseph Fourier publikoval v roce 1822 *Théorie analytique de la chaleur*, kde zdůvodnil Newtonův ochlazovací zákon a dokazuje, že každá periodická funkce může být rozložena do řady sinů a kosinů celočíselných násobků nezávisle proměnné. Fyzikální procesy a signály v časové oblasti jsou reprezentovány komplexní funkcí $f(t)$,

signálem, který vyjadřuje závislost určité veličiny na čase, a signály ve frekvenční doméně jsou reprezentovány obecně komplexní funkcí, která je složena ze sinů a kosinů.

Matematické vyjádření Fourierovy transformace (5) a inverzní Fourierovy transformace (6):

$$FT\{f(\vec{x})\} = A^N \int_{-\infty}^{\infty} f(\vec{x}) e^{-ik\vec{X}*\vec{x}} d^N \vec{x} \quad (5)$$

$$FT^{-1}\{F(\vec{X})\} = B^N \int_{-\infty}^{\infty} F(\vec{X}) e^{ik\vec{X}*\vec{x}} d^N \vec{X} \quad (6)$$

Dále musíme předpokládat, že $f(\vec{x})$ a $F(\vec{X})$ jsou absolutně integrovatelné po částech hladké komplexní funkce reálných proměnných $\vec{x}, \vec{X} \in E_N$. Konstanty A, B mohou být komplexní a konstanta k musí být reálná. Konstanty A, B a k můžeme volit libovolně s dodržením podmínky (7), která vychází z fundamentální věty Fourierovy transformace.

$$AB = \frac{|k|}{2\pi} \quad (7)$$

V nejčastějších případech matematických aplikací se využívá symetrický tvar kde: $A = B = 1, k = 2\pi$. Je nutné dávat pozor na volbu těchto konstant, které se v různých tabulkách a vzorcích pro transformace různí [15], [25], [28].

4.2. Metoda nejmenších čtverců

Metoda nejmenších čtverců je matematicko-statistická metoda pro aproximaci řešení soustav rovnic nebo zpracování dat získaných experimentálním měřením. S aplikací metody nejmenších čtverců se můžeme setkat při prokládání naměřených dat přímkou, parabolou nebo obecným polynomem daného stupně. Tato metoda bývá používána při regresní analýze a pro řešení rozsáhlých úloh, kde jsou jediným vhodným a prakticky použitelným řešením iterační metody [23].

$$E(a, b) = \sum_{n=1}^N (y_n - (ax_n + b))^2 \quad (8)$$

Metodou nejmenších čtverců rozumíme takovou metodu, která hledá vhodné kritérium pro určení co nejlepšího proložení bodů funkcí tak, aby součet druhých mocnin (čtverců) odchylek v jednotlivých bodech byl minimální. Funkce, která toto splňuje, se nazývá nejlepší aproximací experimentálních dat v dané třídě funkcí ve smyslu metody nejmenších čtverců. Ve výrazu (8) je symbolem E reprezentována kvadratická odchylka.

Dále je nutné stanovit podmínku (9), aby E (a, b) nabývala minimum:

$$\frac{\partial(E)}{\partial a} = 0, \quad \frac{\partial(E)}{\partial b} = 0 \quad (9)$$

Metoda nejmenších čtverců je obsažena v mnoha nalezených identifikačních metodách a některé z nich budou uvedeny a popsány v následujících kapitolách.

4.3. Nelder-Mead simplex algoritmus

Tuto metodu poprvé formulovali pánové John Nelder a Roger Mead v roce 1965. Nelder-Mead algoritmus je využíván k hledání minima nebo maxima funkce ve vícerozměrném prostoru. Tento algoritmus je optimalizační metodou, která se snaží vytvořit globální lineární aproximaci n-rozměrné topografie ze simplexů nebo soustavy bodů na určitém intervalu. Zjednodušeně to funguje tak, že v n-rozměrech udržuje uspořádanou soustavu simplexů resp. bodů, na kterou extrapoluje chování cílové funkce naměřené v každém zkušebním bodě, s cílem nalézt nový testovací bod a techniku výpočtu znovu opakovat. Bod je obvykle vybírán jako vrchol, kde funkce, které mají být minimalizovány, dosahuje její nejvyšší hodnoty [16].

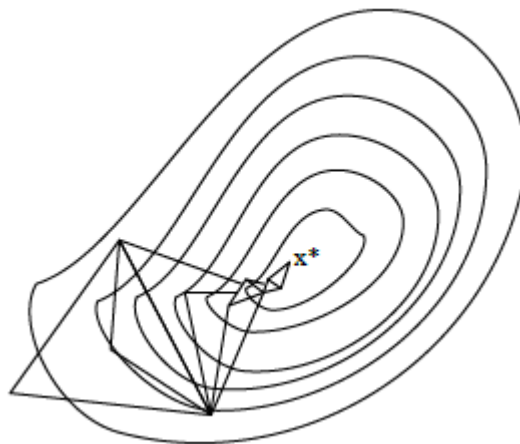
Při každé iteraci, identifikujeme x_j^k pro které platí $f(x_j^k) = \max\{f(x_i^k)\}$ a vypočítá se těžiště zbývajících vrcholů \bar{x} . Dále se spočítá pozice následujícího (odraženého) vrcholu $x_r(10)$, kde $1 > a > 0$:

$$x_r = (1 + a)\bar{x} - ax_j^k \quad (10)$$

Když splníme podmínku $\min\{f(x_i^k)\} < f(x_r) < f(x_j^k)$, pak x_j^k nahradíme x_r a začneme s počítáním znovu s novým simplexem. Kdybychom použili $a = 0,5$ tak by se simplex zredukoval na polovinu s transformací se středem v $\min\{f(x_i^k)\}$:

$$x_i^{k+1} = \frac{x_i^k + x_l^k}{2} \quad (11)$$

Kde $f(x_l^k) = \min\{f(x_i^k)\}$. Celý postup se opakuje, dokud se nedosáhne konečného kritéria.



Obr. 12: Schématický diagram funkce v \mathbb{R}^2 a několika iterací, [XI.]

Na tomto vyhledávacím algoritmu je založena funkce v prostředí MATLAB®, která se nazývá *fminsearch*. Tato funkce je přímá vyhledávací metoda, která nevyužívá numerický nebo analytický gradient jako funkce *fminuns*. Hledá minimum skalární funkce více proměnných (12). Minimum se hledá pomocí počátečního odhadu a je to obecně označováno jako nekonečná nelineární optimalizace.

Kde $f(x)$ je funkce a výsledkem je skalár a x je vektor nebo matice. Syntaxí pro zápis příkazu *fminsearch* jak hledat minimum z dané funkce je mnoho, od prostého zápisu které funkce a počátečního bodu má počítat až po zadávání jednotlivých zpřesňujících parametrů jak výstupních (levá strana zápisu syntaxe), tak i výstupních (pravá strana).

$$\min_x f(x) \quad (12)$$

Omezující podmínky pro použití funkce *fminsearch* jsou takové, že funkce dokáže vyřešit pouze nediferencovatelné problémy a může často pracovat nespojitě a to zejména v případě, že nemůže nalézt řešení. Pokud je x reálné číslo, *fminsearch* vrátí také reálné číslo, pokud ale máme x jako komplexně sdružené, musí být rozděleno na reálnou a imaginární část.

4.4. Gauss-Newtonova metoda (gn)

Gauss-Newtonův algoritmus je metoda používaná pro řešení nelineárních nejmenších čtverců. Jedná se o modifikaci Newtonovy metody pro nalezení minima funkce. Gauss-Newtonův algoritmus může být použit pouze pro minimalizaci součtu kvadratických hodnot funkce.

Popis funkce, kde je m funkcí $r = (r_1 \dots r_m)$ z n proměnných $\beta = (\beta_1 \dots \beta_n)$, $m \geq n$, vypadá následovně:

$$S(\beta) = \sum_{i=1}^m r_i(\beta)^2 \quad (13)$$

Řešení metody začíná zadáním počátečního odhadu $\beta^{(0)}$ (14) a pokračuje iteracemi, kde r a β jsou sloupcové vektory, J je matice Jacobiánu (15) a index T značí, že je matice transponovaná a rovnice r_i (17) jsou rezidua.

$$\beta^{(s+1)} = \beta^{(s)} - (J_r^T J_r)^{-1} J_r^T r(\beta^{(s)}) \quad (14)$$

$$(J_r)_{ij} = \frac{\partial r_i(\beta^{(s)})}{\partial \beta_j} \quad (15)$$

Když $m = n$, pak se iterace zjednoduší na:

$$\beta^{(s+1)} = \beta^{(s)} - (J_r)^{-1} r(\beta^{(s)}) \quad (16)$$

$$r_i(\beta) = y_i - f(x_i, \beta) \quad (17)$$

Tento výraz se považuje za přímé zjednodušení Newtonovy metody v jednom rozměru. Hlavním cílem metody je najít parametry β , které určují, jak se nejlépe proloží model daný funkcí $y = f(x, \beta)$, s daty reprezentujícími například body (x_i, y_i) .

Předpoklad $m \geq n$ v příkazu algoritmu je nutný, aby byly matice J_r^T a J_r regulární, jinak by normální rovnice nešly vyřešit.

Konvergence algoritmu může nastat v případě, že přírůstek $\beta - \beta_s$ klesá. Potom se mez stane stacionárním bodem S . Konvergence u této metody není zaručena, jak je tomu třeba u Newtonovy metody, kde může nastat lokální konvergence. Míra konvergence Gauss-Newtonova algoritmu se může blížit kvadratické. Algoritmus může konvergovat pomalu nebo vůbec, to je dáno počátečním odhadem, který může být daleko od minima nebo když je matice $J_r^T J_r$ špatně podmíněná [18], [22].

4.5. Levenberg-Marquardtova metoda (lm)

V matematice a při počítačových výpočtech je Levenberg-Marquardtův algoritmus (LMA) známý také jako metoda damped least-squares (DLS), která je jedna z variant řešení metody nejmenších čtverců pro nelineární problémy, které vznikají zejména při řešení minimalizace křivek.

Levenberg-Marquardtův algoritmus se nachází na pomezí řešení mezi metodami Gauss-Newtonova algoritmu (GNA) a metodou gradient descent. LMA je více robustní než

GNA, což znamená, že v mnoha případech najde řešení, i když začne počítat mnohem dříve od konečného minima. Nachází ale jen lokální minimum, což nemusí být nutně globální minimum. Pro obyčejné funkce a parametry se tato metoda zdá být pomalejší než Gauss-Newtonova. V mnoha případech je tato metoda zobrazena jako Gauss-Newtonova s použitím Trust-region. LMA je velmi populární v mnoha softwarových aplikacích pro prokládání dat křivkou.

LMA se používá tam, kde je pro metodu nejmenších čtverců daný soubor m empirických vztažných párů nezávislých a závislých proměnných x_i, y_i a optimalizující parametr β modelu křivky $f(x, \beta)$ tak, že součet čtverců odchylek bude minimální (18).

$$S(\beta) = \sum_{i=1}^m [y_i - f(x_i, \beta)]^2 \quad (18)$$

Jako ostatní numerické minimalizační algoritmy je Levenberg-Marquardtův algoritmus založen na iteračním postupu. Pro spuštění minimalizace musí uživatel poskytnout pro řešení počáteční odhad vektoru parametrů β . V případech s jedním minimem je možné použít $\beta = 1$. V případech s mnohočetnými minimy, algoritmus konverguje k celkovému minimu pouze tehdy, pokud je počáteční odhad již v blízkosti konečného řešení. V každém kroku iterace je parametr β nahrazen novým odhadem $\beta + \delta$.

Levenbergův algoritmus má tu nevýhodu, že v případě kdy je hodnota damping faktoru λ velká, nemůže se použít inverze $J^T J + \lambda I$. Marquardt vycházel z předpokladu, že můžeme škálovat každý prvek gradientu podle jeho tvaru a tím nám dovoluje větší pohyb ve směru, kde je gradient menší. Tím se zabrání pomalé konvergenci ve směru malého gradientu. Z toho důvodu Marquardt nahradil jednotkovou matici I maticí diagonální s prvky $J^T J$. Takto vzniklý algoritmus se nazývá Levenberg-Marquardtův (19) [5].

$$(J^T J + \text{diag}(J^T J))\delta = J^T [y - f(\beta)] \quad (19)$$

4.6. Trust-Region Reflective Newton metoda (lsqnonlin)

Mnoho metod použitých v optimalizačním toolboxu jsou založeny na metodě Trust-Region, která je velmi účinná při řešení optimalizačních problémů [27].

Termín trust region je používán v matematické optimalizaci a vyznačuje oblast funkce $f(x)$, (x může být i vektor), která se snaží najít bod x_{i+1} s menší funkční hodnotou než je aktuální bod x_i a funkce f může být aproximována modelem funkce (často kvadratickým \rightarrow Taylorův rozvoj kolem x_i). Pokud se odpovídající model cílové funkce nalézá uvnitř oblasti (trust regionu - N), pak je oblast rozšířena. V opačném případě, kdy aproximace neodpovídá modelu, je oblast zmenšena. Krok $s_i = x_{i+1} - x_i$ se přibližně zjistí z rovnice

$$\min_{s \in N} \psi_i(s) \quad (20)$$

kde \mathbf{g} a \mathbf{H} jsou gradient a Hessova matice:

$$\psi_i(s) = \mathbf{g}^T s + \frac{1}{2} s^T \mathbf{H} s \quad (21)$$

Za těchto podmínek je tvar pro řešení pomocí nelineární metody nejmenších čtverců následující:

$$\min_x \sum_{i=1}^n z_i^2(x) = \min_x \sum \|Z(x)\|^2 \quad (22)$$

kde \mathbf{Z} je vektorové pole.

Metoda trust region reflective (TRR) dává přesnější a méně náročné výsledky než ostatní metody používané v prostředí MATLAB®.

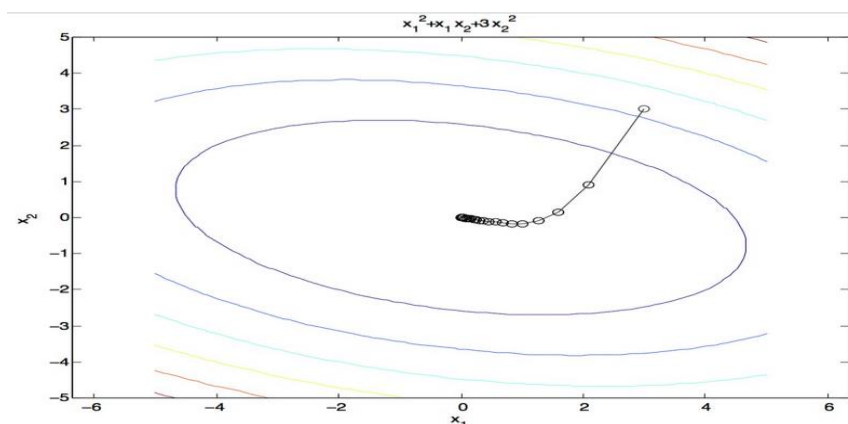
4.7. Gradient search (grad)

Gradientní metoda je optimalizační algoritmus prvního řádu pro hledání lokálního minima dané funkce pomocí gradientního sestupu. Způsob minimalizace funkce je takový, že se musí stanovit počáteční odhad řešení x_0 a potom bereme gradient funkce $f(x)$ v tomto bodě, který musí být v záporném směru od počátečního odhadu, jinak musíme zvolit jiný odhad a proces opakovat. Algoritmus bude konvergovat, když bude gradient nula. Tento stav se nazývá lokální minimum. V opačném případě, pokud bude směr kladný, algoritmus nalézá lokální maximum. Formální zápis této definice je následující:

$$x_{k+1} = x_k - \lambda \nabla f(x_k) \quad (23)$$

kde: ∇f představuje gradient funkce, $k \geq 0$ a $\lambda > 0$ je krok funkce, který je závislý na každé iteraci. Jeho hodnota závisí na dané funkci, a čím menší krok zvolíme, tím bude algoritmus stabilnější. S velkým krokem bude divergovat a s příliš malým bude trvat dlouho, než začne konvergovat. Maximální volbu kroku volíme tzv. Lipschitzovu konstantu. Ta se vypočte ze vztahu (24) za předpokladu, že každá diferencovatelná funkce má maximální derivační hodnotu, tj. maximum derivací ve všech bodech. Jestliže je toto maximum konečné, funkce se nazývá spojitá Lipschitzova.

$$\frac{\|f(x) - f(y)\|}{\|x - y\|} \leq L(f) \text{ pro libovolná } x, y \quad (24)$$

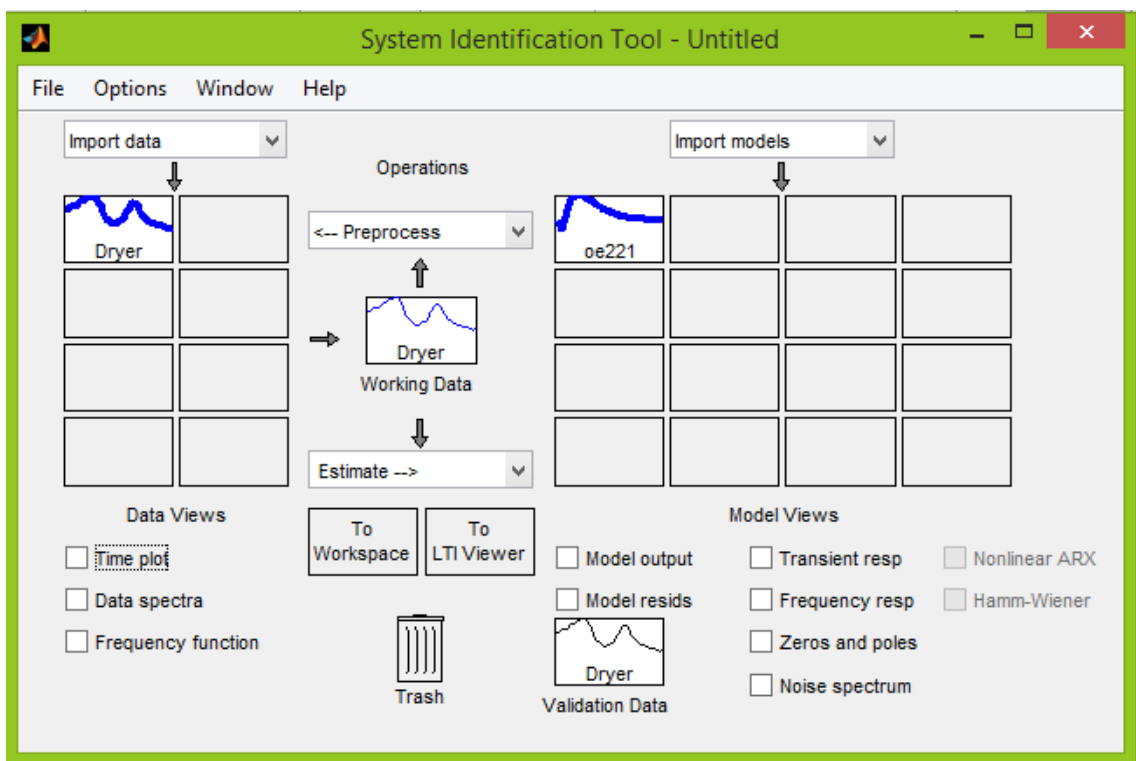


Obr. 13: Příklad algoritmu gradient descent, [XII.]

V prostředí MATLAB® je definována funkce *fminuns*, která vychází z obou předchozích metod a to: Trust-Region a Gradient search. Tato funkce hledá minimum funkce více proměnných. Výsledkem funkce je skalární hodnota a vstupní veličina je vektor nebo matice. Syntaxe zápisu je velice podobná jako předchozí popsaná funkce *fminsearchtr*. V jednom z parametrů si volíme, jestli se minimum počítá pomocí algoritmu Trust-Region nebo pomocí gradientního sestupu [14].

5. System Identification Toolbox

System Identification Toolbox je nástroj pro tvorbu dynamických modelů komplexních systémů z naměřených vstupních a výstupních dat. To uživateli umožňuje vytvářet a používat modely systémů, které nejsou na první pohled jednoduše zřejmé. Vstupní data mohou být jak časově, tak i frekvenčně závislé a provádět identifikaci spojitých a nespojitých funkcí a provádět online identifikaci parametrů. Identifikační techniky obsažené v tomto nástroji jsou vhodné pro různé aplikace od návrhu řídicího systému, zpracování signálů analýzou časových řad, vibrační analýzu a mnoho dalšího. Dále je System Identification Toolbox vhodný i pro modely vytvořené v prostředí Simulink, pro které je poskytnuta knihovna obsahující bloky pro provádění identifikačního procesu v blokovém prostředí Simulink [6], [20], [21], [22].



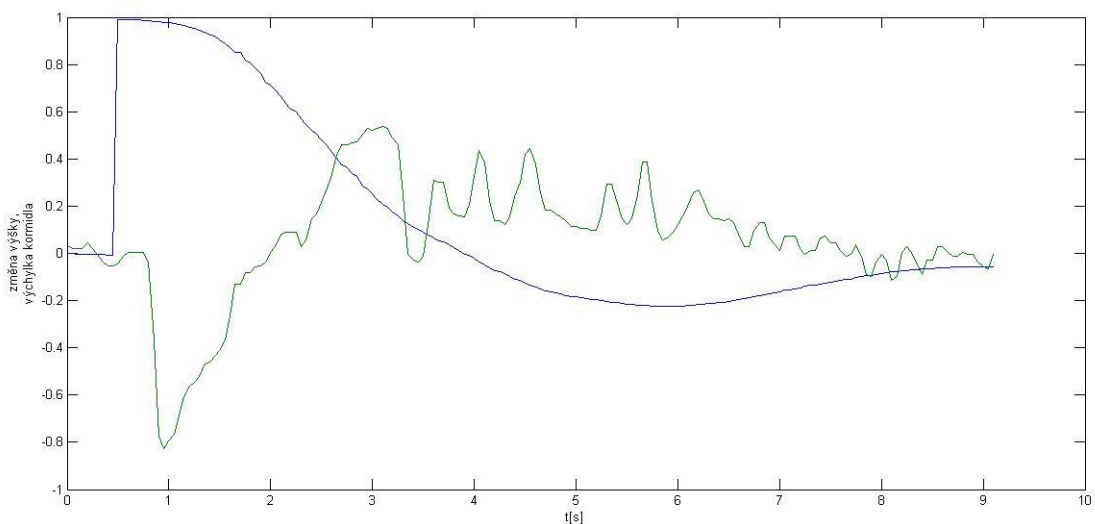
Obr. 14: Grafické uživatelské prostředí System Identification Toolbox, [X.]

Program má flexibilní grafické uživatelské prostředí GUI, které pomáhá při organizaci dat a modelů a výběr jednotlivých možností práce s vloženými daty. Grafické prostředí je rozděleno na část pro vkládání dat (např. z časové nebo frekvenční oblasti). Tyto

vstupní a výstupní data si uživatel může zobrazit jako časové nebo frekvenční průběhy. Dále na oblast dělí na práci s daty a zpracování a oblast pro vkládání modelů, které mohou být importovány z pracovní plochy (Workspace) prostředí MATLAB®, otevření dřívějších relací s možností porovnání výsledků z jednotlivých modelů. Pro zobrazení modelů jsou tu funkce přechodové charakteristiky, frekvenční charakteristiky, mapa pólů a nul a zobrazení spektra šumu daného modelu. Modely a soubory dat můžeme odeslat z grafického prostředí toolboxu na pracovní plochu (Workspace) prostředí MATLAB®, kde můžeme dále s daty pomocí příkazů pracovat a takto upravená data odeslat zpět do grafického prostředí.

5.1. Data pro identifikaci

V tomto případě byly, jako vstupní data použity naměřené hodnoty ze simulátoru X-Plane, které obsahovaly změnu výšky, závislou na výchylce řídicí páky. Výstupní data reprezentovala naměřené hodnoty výchylky řídicí páky v podélném směru. Takto nasnímané data byly upraveny v MATLABu a následně importovány do prostředí System Identification Toolboxu.



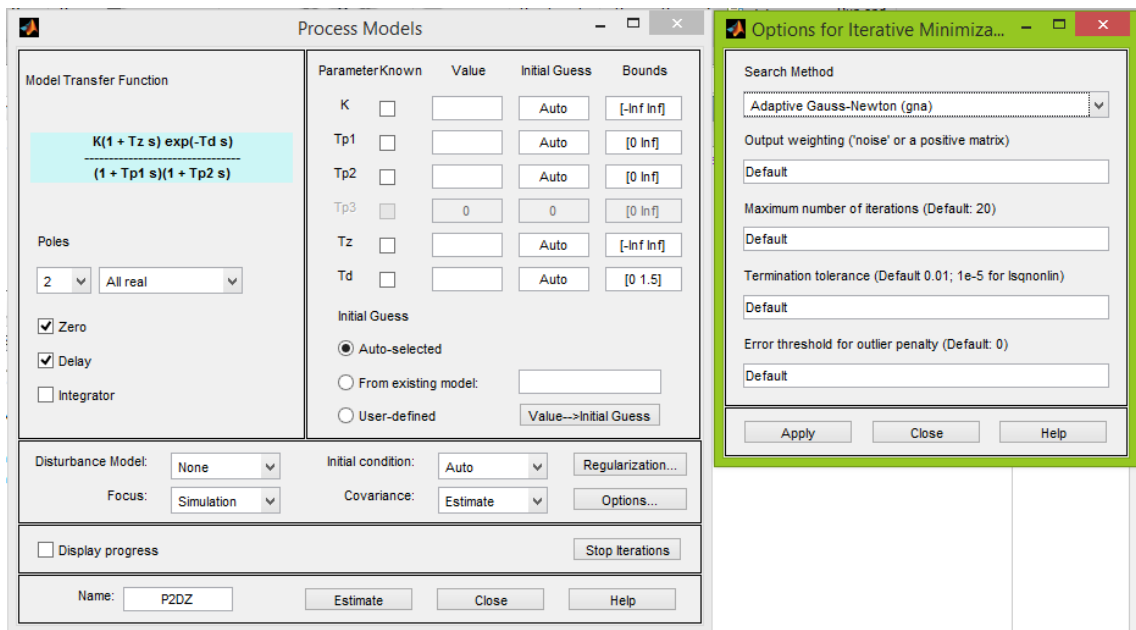
Obr. 15: Skoková změna výšky a následná reakce pilota, [X.]

5.2. Použití System Identification Toolboxu

System Identification Toolbox se v prostředí MATLAB vyvolá příkazem *ident* napsaným do Command window. Po načtení toolboxu (Obr.: 14) je nutné importovat data, která mohou být časově nebo frekvenčně závislá anebo si můžeme importovat datový objekt. Vstupní a výstupní data se zadávají z proměnných, které jsou načteny ve Workspace. V tomto okně se dále nastavují hodnoty jednotek proměnných, vzorkovacího intervalu a názvu relace. V našem případě byl vzorkovací interval nastaven na 0,05s s ohledem na údaje z měření, které se vzorkovaly v tomto intervalu. Importovaná data se zobrazí v levé části toolboxu, která slouží pro organizaci vstupních a výstupních dat a pro jejich zobrazení. Pro analýzu dat slouží v hlavním okně System Identification Toolboxu nabídka *Estimate* a po jejím rozbalení funkce *Process Models* a *Transfer Function Models*. Náznový grafický návod použití System Identification Toolboxu a možných nastavení je uveden v příloze #1.

Process Models

Okno *Process Models* je rozděleno do několika částí. V levé části se zobrazuje model přenosové funkce, který je možné do určité míry upravovat. V pravé části se nachází okna s jednotlivými parametry přenosové funkce, jejich vypočtenými hodnotami, počátečními odhady a mezemi. Spodní část okna je vymezena nastavováním dalších parametrů, jako jsou: především pro tvorbu této práce důležitá nabídka nastavení jednotlivých identifikačních metod se svými zpřesňujícími nastaveními, dále pak model poruchovosti, počáteční podmínky, kovariance a omezení. Toto okno uzavírá pole se zkráceným názvem modelu přenosové funkce podle zvolených parametrů (např.: P2DZ – P2 - dva póly, D - časové zpoždění a Z - část čitatele zde v toolboxu nazvaná Zero)



Obr. 16: Okna Process Models a volby a nastavení identifikačních metod, [X.]

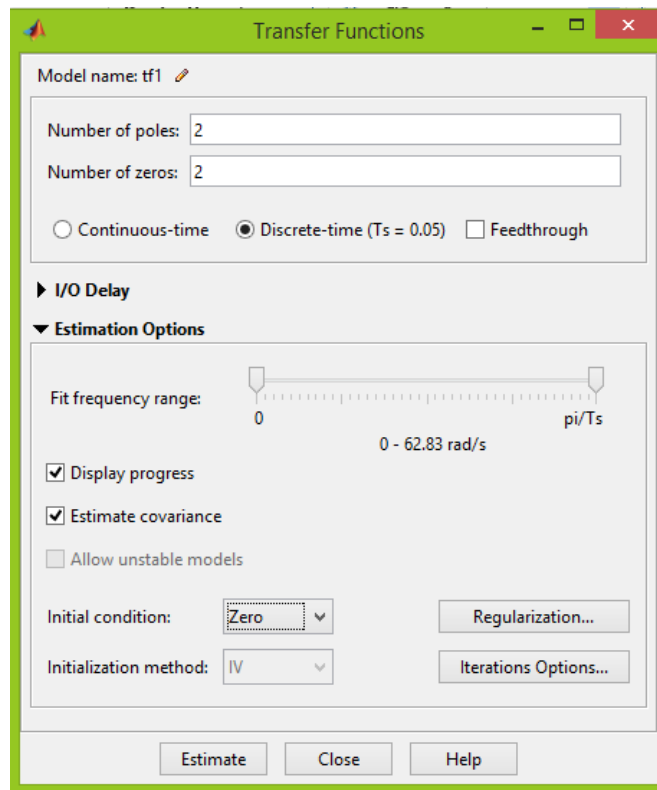
Nabídka *Process Models* dovoluje uživateli nastavit velké množství parametrů, které mohou ovlivnit výsledek analýzy dat. Výhody jsou spatřovány v možnosti nastavení počátečních odhadů jednotlivých koeficientů a jejich mezí nebo načtení již existujícího modelu. Mezi nevýhody této nabídky patří omezení při definování tvaru přenosové funkce, která se omezuje jen na jeden zero koeficient a maximálně tři póly.

Transfer Function Models

Okno *Transfer Function Models* je rozděleno na základní nastavení přenosové funkce, kde se volí počet pólů, nul a čas, který je možné volit spojitý nebo diskrétní se zvolenou periodou vzorkování. Dále se v rozbalovacích nabídkách volí časové zpoždění výstupu proti vstupu (I/O Delay) a v další nabídce pak nastavení odhadu (Estimation Options). Zde se nastavují počáteční podmínky, regularizace a pro nás nejdůležitější nastavení volby vyhledávacího algoritmu a jeho nastavení.

Výhodou této nabídky je detailnější nastavení tvaru přenosové funkce v čitateli i jmenovateli a nastavení spojitě nebo diskrétně časové základny. Nevýhodou by mohlo

být možné zjednodušení voleb v uživatelském prostředí, které ale nemělo velký vliv na analýzu dat v této práci.



Obr. 17: Okno Transfer Function Models, [X.]

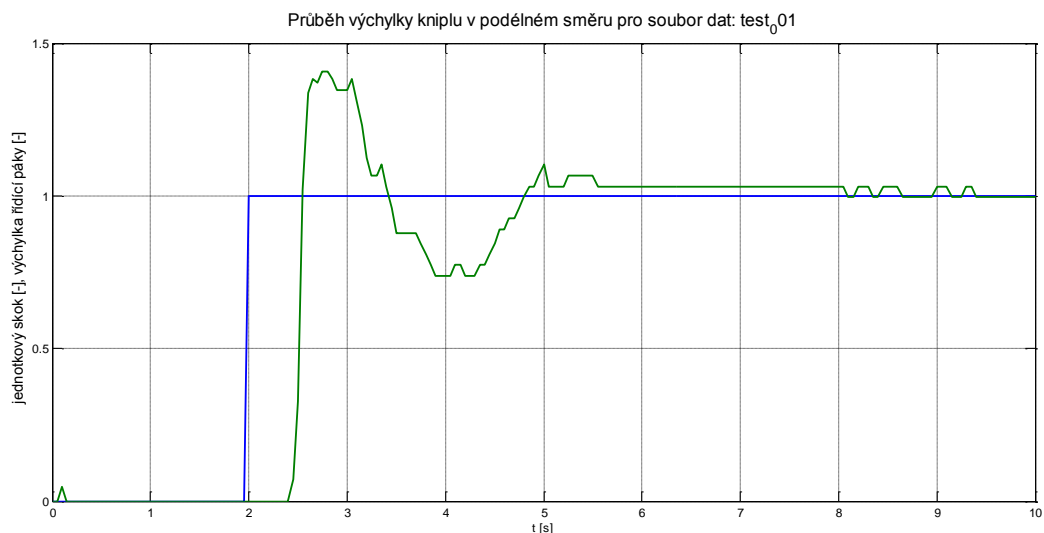
5.3. Analýzy jednotlivých identifikačních metod

Pro analýzu jednotlivých identifikačních metod jsou importována data reprezentující výchylku kniplu a změnu výšky, na které budou použity jednotlivé metody a provedeno jejich porovnání. Při analýze metod se používá stejný datový soubor a model přenosové funkce je nastaven podle přenosové funkce (4) uvedené v teoretické části této práce. Bylo provedeno dvojí porovnání výsledků jednotlivých metod a to pomocí funkcí *Process Models* a *Transfer Function Models*.

Jednotlivé metody identifikačních algoritmů byly porovnávány na jednom souboru dat test_001, se stejnými počátečními podmínkami a se stejným nastavením, jako byly nulové počáteční podmínky a vynechání poruchového modelu. Měnila se jen metoda identifikace

parametrů dynamického modelu a to tak, že se vybrala z výběru metody. Její další nastavení zůstalo výchozí pro všechny metody.

Na modelovém souboru dat je patrné, že při skokové změně výšky, zde reprezentovaném jednotkovým skokem, se pilot snaží pomocí výchylky kniplu sesouhlasit svojí současnou výšku s nově vzniklou změnou. Z Obr. 18 je patrné, že pilot na změnu nereaguje okamžitě, to má za následek dopravní zpoždění. S co nejkratším zpožděním vychýlí páku řízení do maximální výchylky a to má za následek překročení požadované hladiny. Po tomto překročení nastává proces pozvolného vyrovnávání, kdy pilot dokáže jemnou manipulací a s předchozími znalostmi vyrovnat letoun na požadovanou hladinu a držet jí, bez dalších větších překmitů.

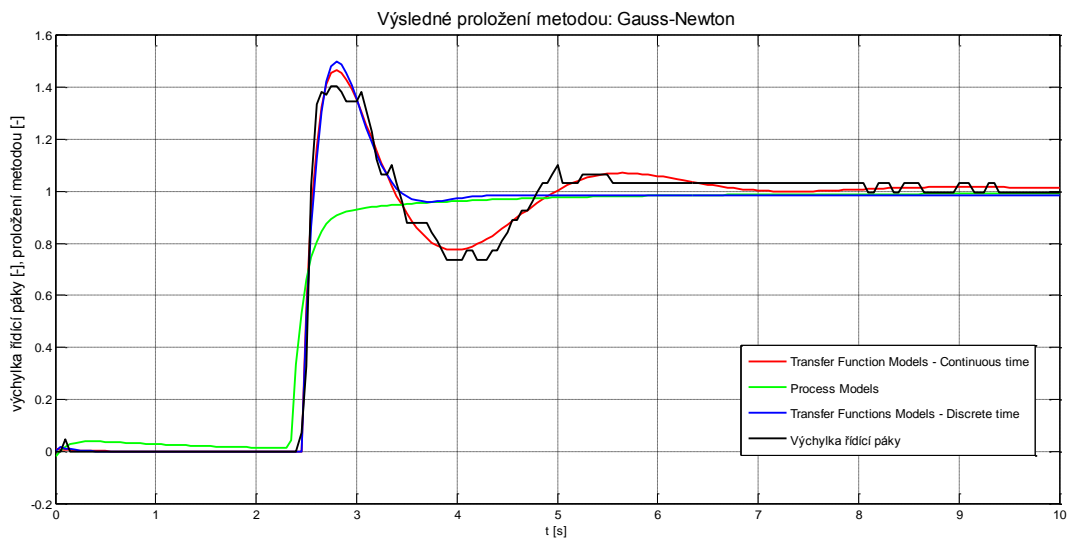


Obr. 18: Modelový soubor dat test_001 pro analýzu jednotlivých metod, [X.]

Následné porovnání výsledků bylo vztaženo na použití jedné metody pro identifikaci a možnosti využití jednotlivých funkcí *Process Models* a *Transfer Function Models*.

Gauss-Newton (gn)

V případě použití Gauss-Newtonovy metody bylo patrné, že nejhorší výsledky proložení křivky byly docíleny pomocí funkce Process Models. Jak již bylo uvedeno v popisu této funkce, omezené možnosti nastavení tvaru přenosové funkce mají za následek špatné proložení křivky výchylky kormidla a z toho vyplývající špatně určené jednotlivé koeficienty přenosové funkce. Zlepšení nastalo při použití funkce Transfer Function Models, kde se zvolil stejný počet kořenů a nul jako v předchozím případě, ale manuálně se nastavil diskretní čas s periodou 0,05s a časové zpoždění 9 vzorků, aby vyšlo odečtené časové zpoždění z grafu 0,45s. Po tomto zásahu se výsledky jeví mnohem relevantnější.



Obr. 19: Naměřená a vypočítaná data pomocí metody: Gauss-Newton, [X.]

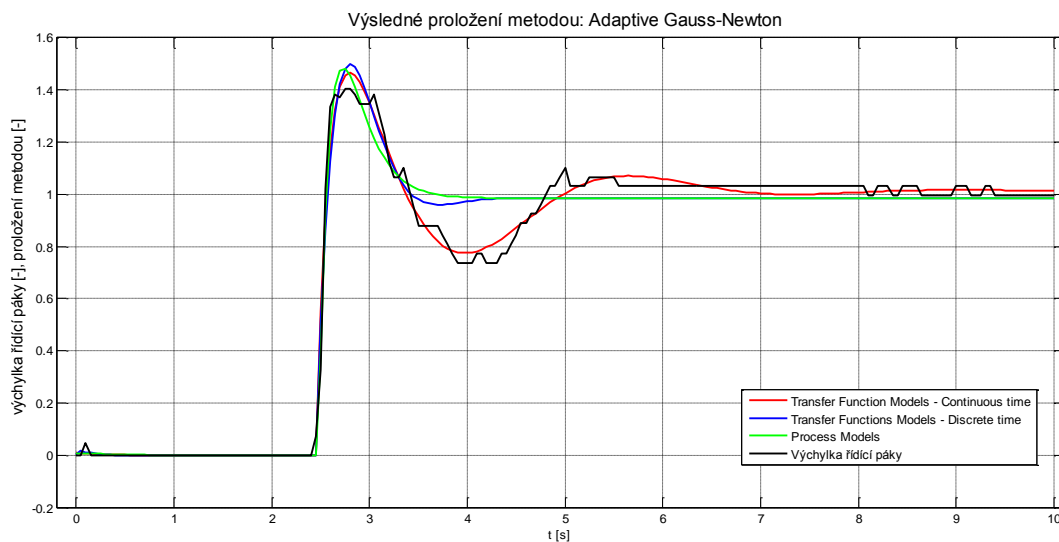
Metoda: Gauss-Newton			
	Výsledky		
	Fit [%]	FPE [-]	MSE [-]
Process Models	68,76	0,021	0,020
Transfer Function Models - Discrete time	83,36	0,006	0,006
Transfer Function Models - Continuous time	92,13	0,001	0,001

Pro docílení nejlepšího proložení se musela přenosová funkce (4) upravit na tvar, kde se v čitateli nachází dvě nuly a ve jmenovateli tři póly a zvolit možnost spojitého času

s nastaveným časovým zpožděním 0,45s. Po této úpravě a konzultaci se proložení jeví jako dostačující pro identifikaci parametrů přenosové funkce.

Adaptive Gauss-Newton (gna)

Výsledky identifikace parametrů při použití metody Adaptive Gauss-Newton ukázaly zlepšení proložení průběhu výchylky pomocí funkce Process Models, která dokázala bez problému prolínout první překmit, ale nedokázala v tomto prolínání pokračovat. Tyto výsledky nebyly opět dostatečně relevantní ani v případě použití funkce Transfer Function Models, která ovšem jeví známky zlepšení. K dostatečně ideálnímu proložení došlo při opětovné modifikaci přenosové funkce a nastavení spojitého času a časového zpoždění 0,45s.

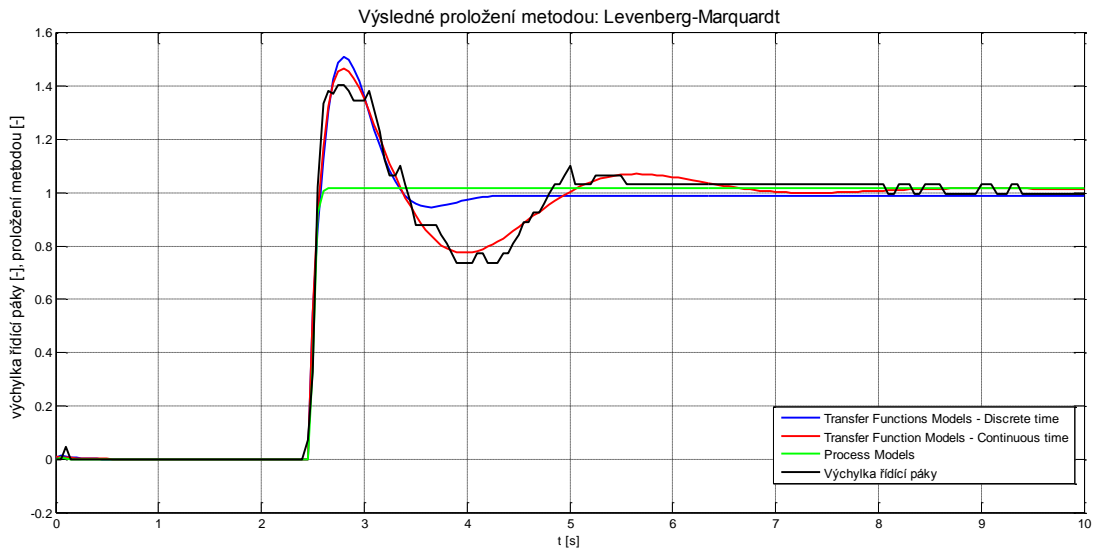


Obr. 20: Naměřená a vypočítaná data pomocí metody: Adaptive Gauss-Newton, [X.]

Metoda: Adaptive Gauss-Newton			
	Výsledky		
	Fit [%]	FPE [-]	MSE [-]
Process Models	82,92	0,006	0,006
Transfer Function Models - Discrete time	83,36	0,006	0,006
Transfer Function Models - Continuous time	92,13	0,001	0,001

Levenberg-Marquardt (lm)

Funkce Process Models a nastavení přenosové funkce do tvaru (4) si opět nedokázala poradit s testovaným průběhem. Opět lepší výsledky proložení a identifikace zaznamenalo použití funkce Transfer Function Models a pozdější modifikace přenosové funkce.

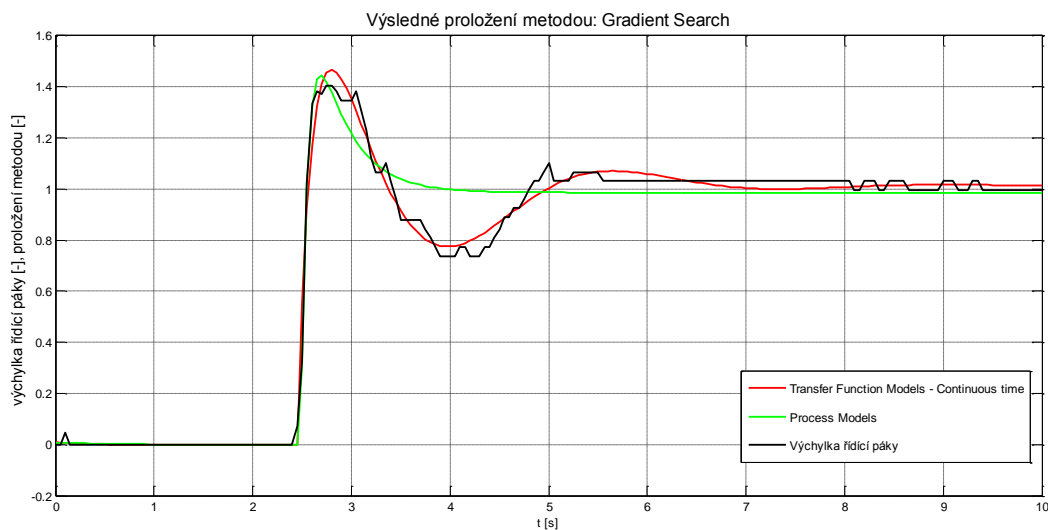


Obr. 21: Naměřená a vypočítaná data pomocí metody: Levenberg-Marquardt, [X.]

Metoda: Levenberg-Marquardt			
	Výsledky		
	Fit [%]	FPE [-]	MSE [-]
Process Models	74,81	0,013	0,013
Transfer Function Models - Discrete time	83,24	0,006	0,006
Transfer Function Models - Continuous time	92,13	0,001	0,001

Gradient Search (grad)

Předpokládané výsledky se dostavily i při použití další identifikační metody s tím rozdílem, že při použití nastavení pro Transfer Function Models s diskretním časem nastal jev, který byl znázorněn rozkmitáním průběhu proložení do takové míry, že v žádném případě nebylo možno uvažovat o použití tohoto nastavení a z toho důvodu nebyl zařazen ani do výsledného výstupu (Obr. 22). Pro názornost je průběh uveden jako příloha #2.



Obr. 22: Naměřená a vypočítaná data pomocí metody: Gradient Search, [X.]

Metoda: Gradient Search			
	Výsledky		
	Fit [%]	FPE [-]	MSE [-]
Process Models	82,27	0,007	0,006
Transfer Function Models - Discrete time	41,21	0,077	0,072
Transfer Function Models - Continuous time	92,13	0,001	0,001

Trust-Region Reflective Newton (lsqnonlin)

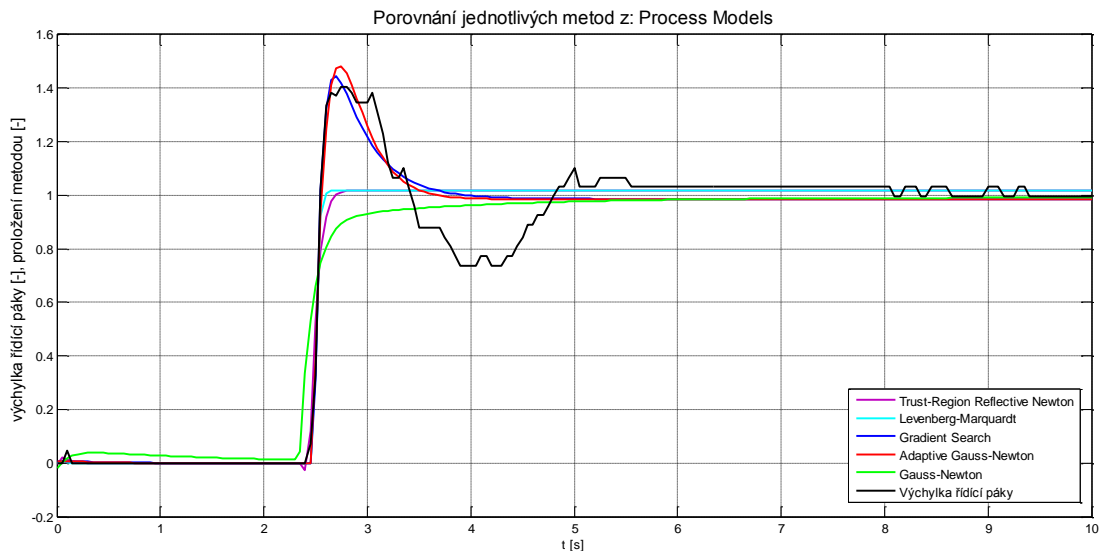
Metodu Trust Region Reflective Newton jde použít pouze v nastavení funkce Process Models, kde vykazovala podobné výsledky jako metoda Levenberg-Marquardt. Při použití funkce Transfer Function Models program MATLAB® nahlásil vnitřní chybu použití této metody při provádění procesu.

Metoda: Trust Region Reflective Newton			
	Výsledky		
	Fit [%]	FPE [-]	MSE [-]
Process Models	73,89	0,015	0,014

Obrázky 23 a 24 ukazují názorné porovnání průběhů identifikace jednotlivých metod pomocí funkcí Process Models a Transfer Function Models. Nastavení identifikace bylo opět stejné pro jednotlivé porovnání metod.

Process Models

Z Obr. 23 je patrné, že při počítání parametrů si nejlépe vedly metody Adaptive Gauss-Newton a Gradient Search. Tyto dvě metody dokázaly dobře zareagovat na dopravní zpoždění a vcelku obstojně počítat s prvním překmitem. Posléze, ale nedokázaly počítat s pilotovým podrovnáním a průběh identifikace se zastavil na konečné hladině skoku. Zbývající metody byly v procesu zcela nevyhovující.



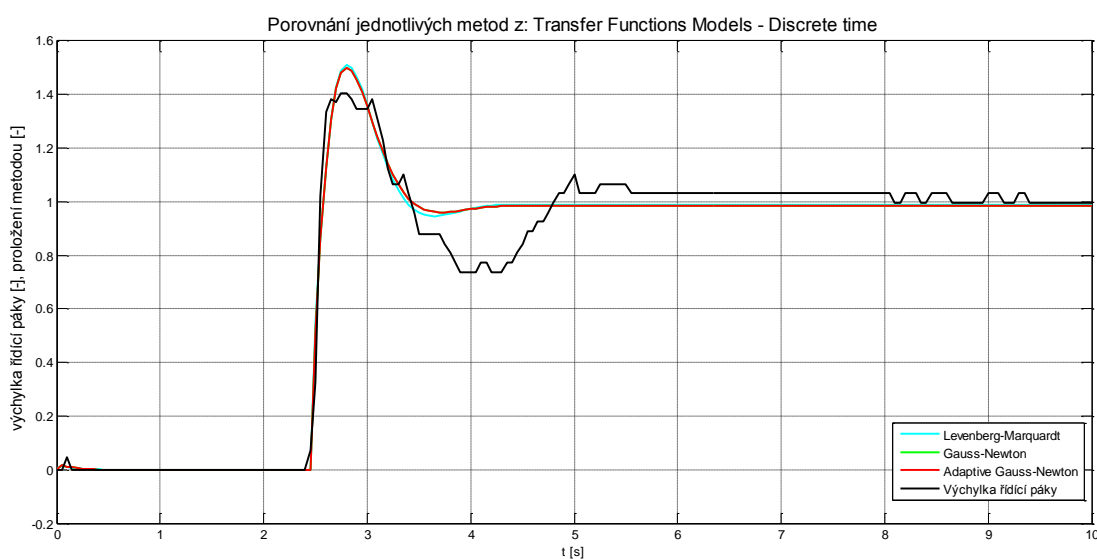
Obr. 23: Srovnání metod pomocí funkce Process Models. [X.]

Transfer Function Models – Discrete time

Při použití této funkce a stejného tvaru přenosové funkce jsem docílil velice podobných výsledků jako v předchozím případě. Změna nastala až s využitím možnosti změny tvaru přenosové funkce.

O poznání lepších výsledků bylo dosaženo použitím funkce Transfer Function Models se zavedením možnosti Discrete time ($T=0,05s$) a ručním nastavením časového zpoždění na hodnotu 9 vzorků (pozn.: $9 \cdot 0,05 = 0,45s$ – hodnota zpoždění výstupního signálu od vstupního). Zde je patrné vhodnější použití této funkce pro identifikaci parametrů z důvodu zavedení více koeficientů do tvaru přenosové funkce. Všechny využitě metody vykazovaly podobné výsledky při identifikaci.

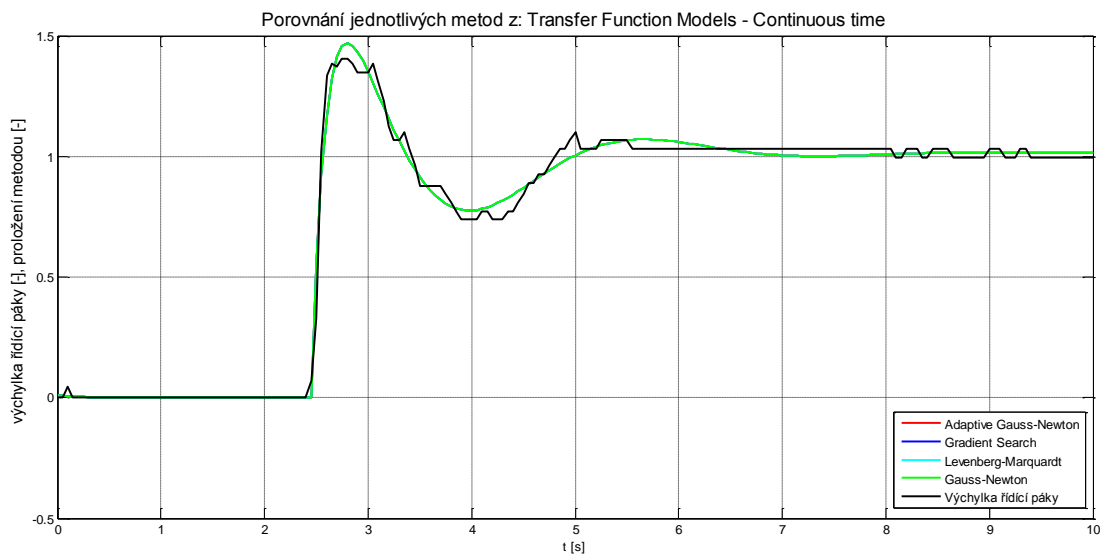
V tomto případě, ale nemohly být použity všechny metody v této funkci a to Gradient Search a Trust Region Reflective Newton z důvodů, které již byly uvedeny dříve.



Obr. 24: Srovnání metod pomocí funkce Transfer Function Models – Discrete time

Transfer Function Models – Continuous time

Nejlepších výsledků bylo dosaženo nastavením funkce Transfer Function Models na Continuous time a nastavením časového zpoždění na hodnotu 0,45s. Opět byl použit rozšířený tvar přenosové funkce ve tvaru 3 póly a 2 nuly. Z překrývajících průběhů je patrné, že všechny metody si vedly v určování parametrů shodně.



Obr. 25: Srovnání metod pomocí funkce Transfer Function Models – Continuous time

Pro získání jednotlivých hodnot koeficientů, tvaru přenosové funkce a výsledků identifikace stačí kliknout pravým tlačítkem na myši na námi zvolený vypočítaný výstup v hlavním okně Systém Identification Toolboxu, poté se zobrazí okno *Data/model Info* s informacemi. V okně je vypsáno i nastavení pro provádění identifikace. Zobrazené výsledky se mohou pomocí tlačítka *Present* přenést do okna Command window.

```

Command Window

agn =
Process model with transfer function:
          1+Tz*s
G(s) = Kp * ----- * exp(-Td*s)
          (1+Tp1*s) (1+Tp2*s)

      Kp = 0.98538 +/- 0.0069981
      Tp1 = 0.17559 +/- 699.15
      Tp2 = 0.1756 +/- 699.19
      Td = 0.47585 +/- 0.0071965
      Tz = 0.55856 +/- 0.064921

Name: agn
Parameterization:
  'P2DZ'
  Number of free coefficients: 5
  Use "getpvec", "getcov" for parameters and their

Status:
Termination condition: Maximum number of iterations
Number of iterations: 20, Number of function evaluat

Estimated using PROCEST on time domain data "mydata"
Fit to estimation data: 82.92%
FPE: 0.006309, MSE: 0.00601

```

Obr. 26: Výpis výsledků identifikace

5.4. Algoritmizace analýzy datových struktur

Dalším bodem této práce bylo mimo provedení analýzy a vyhodnocení použití jednotlivých identifikačních metod, také sestrojení algoritmu pro ovládání samotného System Identification Toolboxu pomocí m-file souboru [22]. Tyto soubory mohou být implementovány do autorských souborů uvedených v kapitole 3 pro jejich další rozšíření. Takto vytvořené algoritmy nabízejí uživateli přehledné a pro někoho jistě i příjemnější ovládání toolboxu. Jak již bylo řešeno, analýza dat se dá provést dvěma způsoby a to pomocí *Process Models* a přes *Transfer Function Models*. Obě tyto řešení se dají ovládat přes grafické rozhraní anebo přes příkazy v Command window. Vzorový m-file s ovládáním přes Command window je uveden v příloze #3.

V první řadě je potřeba načíst do Workspace prostředí MATLAB® danou úlohu, která představuje definovanou situaci a výslednou reakci pilota na ní. Pro další zpracování jsou nejdůležitější načtené proměnné Up (průběh výšky letu), Ym (výchylka řídicí páky v podélném směru) a Ts , což je vzorkovací perioda (0.05). Načtení úlohy se provede příkazem *load*.

Ovládání *Process models* přes Command window

Pro odhad modelu s využitím časově závislých dat je použit příkaz *procest*. Syntaxe zápisu tohoto příkazu je opět rozdělena na vstupní a výstupní argumenty.

Jedna z možných variant zápisu syntaxe: $sys = procest(data,type,opt)$ a následný výpis použitých příkazů a jejich argumentů:

Vstupní argumenty:

<i>data</i>	<p>Odhadovaná data.</p> <p>Tento argument skrývá možnost volby odhadu z časově nebo frekvenčně závislých dat. V našem případě jsou to časově závislá data <i>iddata</i>, která obsahují vstupní a výstupní signály. Pro správnou funkci odhadu dat je potřeba načtená data mít ve formátu <i>iddata(Ym,Up,Ts)</i>.</p>
<i>type</i>	<p>Process model – struktura přenosové funkce</p> <p><i>type</i> je akronym, který definuje strukturu přenosové funkce a skládá se z:</p> <ul style="list-style-type: none"> • P – začátek zápisu značí póly (poles) • 0, 1, 2, 3 – počet časových konstant (pólů) • I – integrátor • D – časové zpoždění • Z – konstanta zero • U – vkládá komplexně sdružené póly
<i>opt</i>	<p>Nastavení odhadu</p> <p>Tento argument je set nastavení tvořený pomocí <i>procestOptions</i> a zahrnuje: počáteční podmínky, volbu identifikační metody a mnoho dalšího.</p>
<i>init_sys</i>	<p>Dynamický systém, který nastavuje počáteční parametry pro <i>sys</i>. Při použití vlastností <i>init_sys</i> můžeme nastavovat počáteční odhady a konstanty $K_p, T_{p1}, T_{p2}, T_{p3}, T_w, T_d$ a T_z.</p>

Párové argumenty: Název – Hodnota

<i>'InputDelay'</i>	Numerický vektor, který specifikuje časové zpoždění pro každý vstupní kanál.
---------------------	--

Výstupní argumenty:

<i>sys</i>	Identifikovaný proces model. <i>sys</i> je <i>idproc</i> model se strukturou definovanou pomocí <i>type</i>
------------	--

Tabulka nastavení *procestOptions*:

' <i>InitialCondition</i> ' –	Specifikuje počáteční podmínky odhadu ' <i>zero</i> ' – počáteční podmínka nastavena na nulu ' <i>estimate</i> ' – vlastní odhad počátečních parametrů ' <i>auto</i> ' – software volí metodu výběru počátečních podmínek na základě vstupních dat
' <i>Display</i> ' –	Zobrazení informací o průběhu procesu odhadu ' <i>off</i> ' / ' <i>on</i> '
' <i>SearchMethod</i> ' –	Volba identifikační metody pro odhad parametrů. (Metody zvlášť jsou popsány v kapitole 4) ' <i>gn</i> ' – Gauss-Newtonova metoda ' <i>gna</i> ' – Adaptive Gauss-Newtonova metoda ' <i>lm</i> ' – Levenberg-Marquardtova metoda ' <i>lsqnonlin</i> ' – nelineární metoda nejmenších čtverců ' <i>grad</i> ' – metoda Gradient search ' <i>auto</i> ' – algoritmus volí jednu z metod
...	

Samotné porovnání modelu a změřených výstupních dat je řešeno přes příkaz *compare*. Příkaz má opět vstupní a výstupní argumenty. V mém případě byly použity vstupní argumenty *data* a *sys*, popisované v předchozích tabulkách.

Ovládání *Transfer Function models* přes *Command window*

Zápis algoritmu pro ovládání *Transfer Function Models* v *Identification Toolboxu* je velice podobný jako pro *Process models*. Stejná je příprava dat (*data*) i nastavení počátečních podmínek (*procestOptions*). Změna je akorát v syntaxi zápisu v odhadu přenosové funkce, kde se používá příkaz *tfest*. V mém případě vypadá zápis následovně: $sys = tfest(data,np,nz,opt)$.

Vstupní argumenty:

<i>data</i>	Odhadovaná data. Tento argument skrývá možnost volby odhadu z časově nebo frekvenčně závislých dat ve formátu <i>iddata</i> .
<i>np</i>	Počet pólů – struktura přenosové funkce <ul style="list-style-type: none"> • <i>np</i> je nenulové číslo
<i>nz</i>	Počet nul – struktura přenosové funkce <ul style="list-style-type: none"> • <i>np</i> je nenulové číslo
<i>iodelay</i>	Transportní zpoždění
<i>opt</i>	Nastavení odhadu Tento argument je set nastavení tvořený pomocí <i>procestOptions</i> a zahrnuje: počáteční podmínky, volbu identifikační metody a mnoho dalšího.
<i>init_sys</i>	Dynamický systém, který nastavuje počáteční parametry pro <i>sys</i> . Při použití vlastností <i>init_sys</i> můžeme nastavovat počáteční odhady a konstanty $K_p, T_{p1}, T_{p2}, T_{p3}, T_w, T_d$ a T_z .

Párové argumenty: Název – Hodnota

' <i>InputDelay</i> '	Numerický vektor, který specifikuje časové zpoždění pro každý vstupní kanál.
' <i>Ts</i> '	Vzorkovací perioda <ul style="list-style-type: none"> • 0 – Continuous time • <i>data.Ts</i> – Discrete-time model
...	

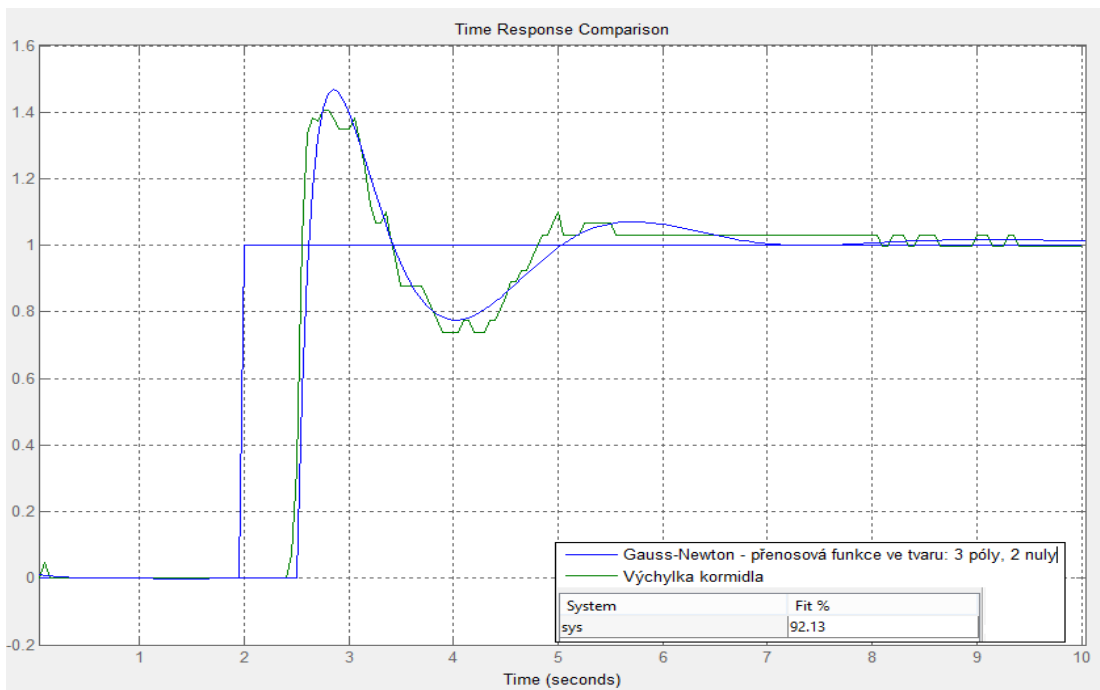
Výtupní argumenty:

<i>sys</i>	Určená přenosová funkce. <i>sys</i> je <i>idtf</i> model
------------	---

Tabulka nastavení *procestOptions*:

'InitialCondition' –	Specifikuje počáteční podmínky odhadu 'zero' – počáteční podmínka nastavena na nulu 'estimate' – vlastní odhad počátečních parametrů 'auto' – software volí metodu výběru počátečních podmínek na základě vstupních dat
'Display' –	Zobrazení informací o průběhu procesu odhadu 'off' / 'on'
'SearchMethod' –	Volba identifikační metody pro odhad parametrů. (Metody zvlášť jsou popsány v kapitole 4) 'gn' – Gauss-Newtonova metoda 'gna' – Adaptive Gauss-Newtonova metoda 'lm' – Levenberg-Marquardtova metoda 'lsqnonlin' – nelineární metoda nejmenších čtverců 'grad' – metoda Gradient search 'auto' – algoritmus volí jednu z metod
...	

Porovnání modelu a změřených výstupních dat je řešeno stejně jako v předchozím případě, jen je tento výstup řešen v odlišné grafice a uspořádání. Toto porovnání se vyvolá přes příkaz *compare* se vstupními argumenty *data* a *sys*.



Obr. 27: Okno Compare při ovládání Systém Identification Toolboxu přes Command window [X.]

Závěr

Cílem této diplomové práce Analýza identifikačních metod pro identifikaci parametrů dynamického modelu bylo vytvořit přehled dostupných matematických metod a následně funkcí pro identifikaci parametrů, které jsou obsaženy v knihovnách vývojového prostředí MATLAB® a dále provést jejich porovnání na naměřeném vzorku dat, který byl reprezentován daty z autorského programu (simulátoru) představujícími informace o reakcích pilota na změnu letu vyvolanou skokovou změnou výšky.

Po prostudování literatury týkající se problematiky identifikace parametrů v prostředí MATLAB® bylo nejvhodnější využití System Identification Toolboxu. Tento toolbox je velice dobře zpracovaný a uživatelsky příjemný. Po získání znalostí jeho obsluhy jsem přistoupil na vlastní zadání práce a to identifikace parametrů jednotlivými identifikačními metodami s pomocí funkcí Process Models a Transfer Function Models.

Nabídka funkce Process Models skýtá větší možnosti nastavení konkrétních počátečních odhadů a mezí pro jednotlivé parametry přenosové funkce. Nevýhodou však je návrh tvaru přenosové funkce, který se omezuje pouze na tvar, kde se v čitateli nachází pouze jedna nula a maximálně tři póly ve jmenovateli. Z toho důvodu byly výsledky identifikace testovaného souboru dat pomocí této metody nejméně účinné. Výhodou bylo vypsání tvaru přenosové funkce a vypočítaných koeficientů.

Na druhou stranu nabídka funkce Transfer Function Models se na první pohled jeví méně uživatelsky přizpůsobitelná. Není zde možnost volby mezí a konkrétních počátečních podmínek, ale je zde možnost tvorby libovolného tvaru přenosové funkce a volby diskrétního nebo spojitého času. Ostatní nastavení jsou pak shodná s nastavením v Process Models. Identifikace parametrů dynamického modelu byla při použití této funkce velice přesná. Bylo to způsobeno zmíněnou možností návrhu tvaru přenosové funkce jak v čitateli, tak i ve jmenovateli. Po konzultaci byla přenosová funkce zaváděna ve tvaru dvou nul a tří pólů. Zvedat jejich počet by mohlo vézt k ještě lepším výsledkům, ale z hlediska matematického popisu člověka bychom se dostali ke koeficientům, které by nebylo možné přirovnat k vlastnostem lidského těla.

Další částí mé práce bylo vytvoření algoritmů pro ovládání System Identification Toolboxu pomocí příkazů v Command window. Byly vytvořeny m-file soubory pro jednotlivé metody identifikace. Takto vytvořené m-file soubory mohou sloužit jako samostatně spustitelné části pro identifikaci parametrů anebo mohou sloužit pro rozšíření autorského programu o další metody identifikace. Pro uživatele, který není vůbec seznámen s používáním a ovládáním System Identification Toolboxu se může jevit zápis těchto souborů více pochopitelný a to z důvodu omezení zápisu pouze pro účely výběru metody a počátečních podmínek, základního nastavení, vlastní identifikace a výpisu parametrů.

Jak již bylo řečeno dříve, práce se zabývala popisem a využitím jednotlivých metod identifikace parametrů dynamických systémů, jejich porovnání a tvorbě algoritmů pro ovládání System Identification Toolboxu. Přínos této práce vidím v osvětlení funkce identifikace pomocí metod v prostředí MATLAB® a využití spočítaných koeficientů pro další porovnání například v zlepšení některých návyků ovládání letounu.

Literatura

- [1] BOŘIL, Jan. *Analýza a porovnání vlastností statického a astatického autopilota*. Brno, 2010. Diplomová práce. Univerzita obrany.
- [2] BOŘIL, Jan. *Analýza mechatronické soustavy pilot - letadlo - autopilot z hlediska systémů automatického řízení letu*. Brno, 2013. Disertační práce. Univerzita obrany.
- [3] Computer simulation. *ScienceDaily* [online]. [cit. 2015-06-04]. Dostupné z: http://www.sciencedaily.com/terms/computer_simulation.htm
- [4] FAJMON, Břetislav a Irena RŮŽIČKOVÁ. VUT FEKT. *Matematika 3*. Dostupné také z: <http://www.umat.feec.vutbr.cz/~hlavicka/skripta/matematika3.pdf>
- [5] GAVIN, Henri P. The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering Duke University*. 2013. Dostupné také z: <http://people.duke.edu/~hpgavin/ce281/lm.pdf>
- [6] GELETU, Abebe. TU-ILMENAU. *Solving Optimization Problems using the Matlab Optimization Toolbox - a Tutorial*. December 13, 2007.
- [7] Hassan Shekarforoush, Marc Berthod, Josiane Zerubia. Direct Search Generalized Simplex Algorithm for Optimizing Non-linear Functions. RR-2535, 1995.<[inria-00074143](#)>
- [8] JALOVECKÝ, R. *Identifikace parametrů matematických modelů chování člověka*. In: Měření, diagnostika, spolehlivost palubních soustav letadel. Brno: Univerzita obrany, 2010, s. 65-69. ISBN 978-80-7231-741-7.
- [9] JALOVECKÝ, R. *NÁSTROJE PRO ANALÝZU NAMĚŘENÝCH DAT CHOVÁNÍ ČLOVĚKA PŘI ŘÍZENÍ LETU LETOUNU*. In: Měření, diagnostika, spolehlivost palubních soustav letadel. Brno: Univerzita obrany, 2014, s. 122-128. ISBN 978-80-7231-970-1.
- [10] JALOVECKÝ, R. *Palubní systémy řízení letu I*. 1. vyd. Brno: Univerzita obrany, 2008. 82 s. ISBN 978-80-7231-500-0.
- [11] JALOVECKÝ, R. *Palubní systémy řízení letu II*. 1. vyd. Brno: Univerzita obrany, 2008. 93 s. ISBN 978-80-7231-593-2.
- [12] JALOVECKÝ, R., JANŮ, P., *Analysis of the Characteristics of a Pilot in the Aircraft Flight*, In Science & Military – Journal of Science. Armed Forces Academy of General Milan Rastislav Stefanik, Liptovský Mikulas 2009, No. 2., s. 37- 41. ISSN 1336-8885

- [13] JALOVECKÝ, Rudolf;. TOOLS FOR ANALYSING MEASURED DATA OF HUMAN BEHAVIOUR WHILE FLIGHING AN AIRCRAFT. In: Modern Safety Technologies in Transportation. Košice, Slovenská Republika: Perpetis, s.r.o., 2015, v tisku.
- [14] JAMESON, Antony. Gradient Based Optimization Methods. *Department of Aeronautics and Astronautics Stanford University*. 2006. Dostupné také z: <http://aero-comlab.stanford.edu/Papers/jameson.gbom.pdf>
- [15] KOMRSKA, Jiří. ÚSTAV FYZIKÁLNÍHO INŽENÝRSTVÍ, FAKULTA STROJNÍHO INŽENÝRSTVÍ VUT. *Fourierovské metody v teorii difrakce a ve strukturní analýze*. 2000. Dostupné také z: <http://physics.fme.vutbr.cz/~komrska/Fourier/KapF02.pdf>
- [16] Lagarias, J. C., J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," *SIAM Journal of Optimization*, Vol. 9, Number 1, pp. 112–147, 1998.
- [17] LAMINAR RESEARCH. *X-Plane 10 - manual*. 2014.
- [18] LAWLESS, A. S. Approximate Gauss–Newton Methods for Nonlinear Least Squares Problems. *SIAM Journal on Optimization*. 2007, **18**(1): 106-132. DOI: 10.1137/050624935. ISSN 1052-6234. Dostupné také z: <http://epubs.siam.org/doi/abs/10.1137/050624935>
- [19] Ljung, L. "Experiments with Identification of Continuous-Time models." *Proceedings of the 15th IFAC Symposium on System Identification*. 2009.
- [20] LJUNG, Lennart. *System Identification Toolbox: For Use with MATLAB® User's Guide*. 2003.
- [21] LJUNG, Lennart. *System identification: theory for the user*. London: Prentice-Hall, 1987, xxi, 519 s. Prentice Hall information and system science series. ISBN 0138816409.
- [22] MATLAB® Documentation [online]. [cit. 2015-06-04]. Dostupné z: <http://www.mathworks.com/help/matlab/index.html>
- [23] MILLER, Steven J. The Method of Least Squares. *Mathematics Department Brown University*. Dostupné také z: http://web.williams.edu/Mathematics/sjmiller/public_html/BrownClasses/54/handouts/MethodLeastSquares.pdf
- [24] *NEOS: Algorithms* [online]. [cit. 2015-06-04]. Dostupné z: <http://www.neos-guide.org/algorithms>

- [25] NOSKIEVIČ, Petr. Modelování a identifikace systémů. Ostrava: Montanex, 1999, 276 s. ISBN 80-722-5030-2.
- [26] Optimization algorithms and methods. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-06-04]. Dostupné z: http://en.wikipedia.org/wiki/Category:Optimization_algorithms_and_methods
- [27] SORENSEN, D. C. Newton's Method with a Model Trust Region Modification. *SIAM Journal on Numerical Analysis*. 1982, **19**(2): 409-426. DOI: 10.1137/0719026. ISSN 0036-1429. Dostupné také z: <http://epubs.siam.org/doi/abs/10.1137/0719026>
- [28] VROŽINA, Milan, Zora JANČÍKOVÁ a Jiří DAVID. Identifikace systémů. Vyd. 1. Ostrava: Vysoká škola báňská - Technická univerzita, 2012, 1 CD-ROM. ISBN 978-80-248-2594-6.

Zdroje obrázků

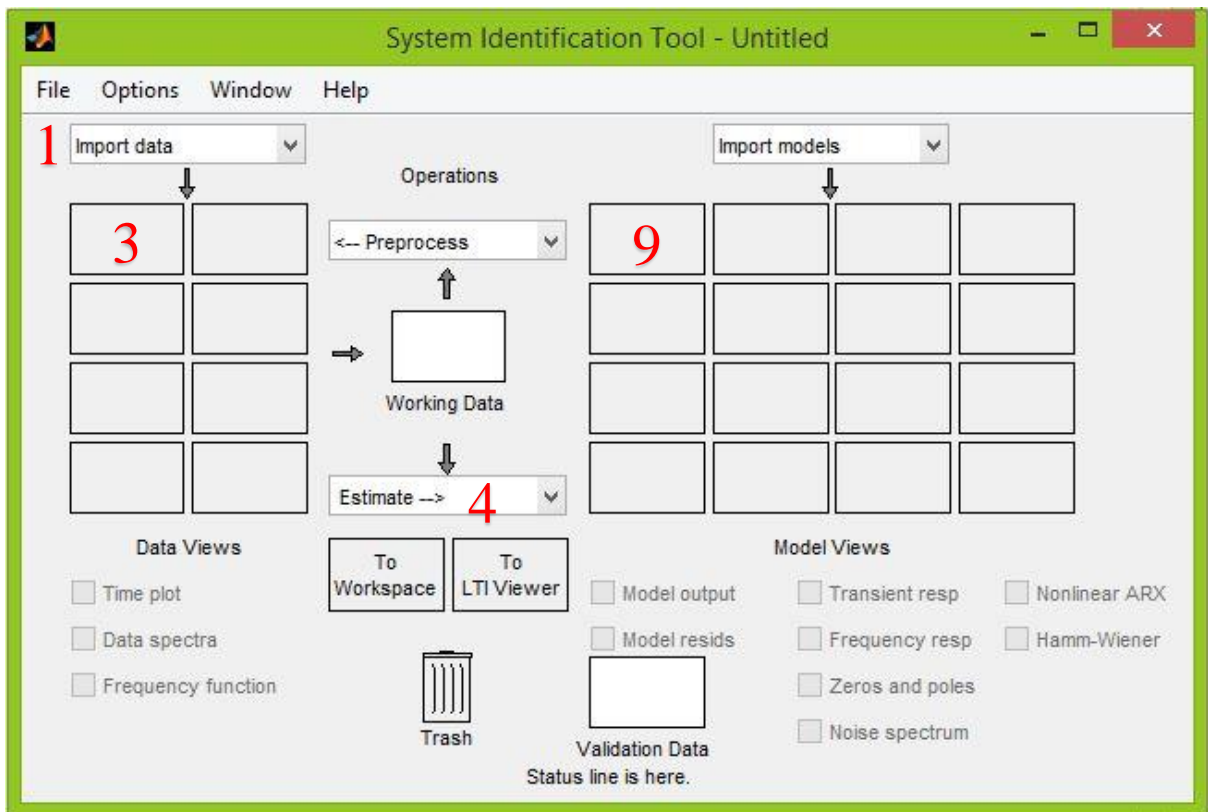
- [I.] Přechodová charakteristika kmitavé soustavy 2. řádu, převzato z: [27]
VROŽINA, Milan, Zora JANČÍKOVÁ a Jiří DAVID. Identifikace systémů
- [II.] Přechodová charakteristika integrační soustavy 1. řádu, převzato z: [27]
VROŽINA, Milan, Zora JANČÍKOVÁ a Jiří DAVID. Identifikace systémů
- [III.] Přechodová charakteristika reálné derivační soustavy, převzato z: [27]
VROŽINA, Milan, Zora JANČÍKOVÁ a Jiří DAVID. Identifikace systémů
- [IV.] Blokové schéma modelu chování pilota, převzato z: [11] JALOVECKÝ, R.
Palubní systémy řízení letu II.
- [V.] Pilotní kabina letounu King Air, převzato z: <http://goo.gl/0ZQEfP>
- [VI.] Stránka nastavení okna "Data Input & Output", převzato z: <http://goo.gl/bzL2tm>
- [VII.] Princip měření odezvy člověka na vnější podnět, převzato z: [12]
JALOVECKÝ, R., JANŮ, P., *Analysis of the Characteristics of a Pilot in the Aircraft Flight*
- [VIII.] Software MOP – hlavní okno, převzato z: [2] BOŘIL, Jan. *Analýza mechatronické soustavy pilot - letadlo - autopilot z hlediska systémů automatického řízení letu*
- [IX.] Hlavní okno programu, Okno průběhu identifikace, převzato z: [9]
JALOVECKÝ, R. *NÁSTROJE PRO ANALÝZU NAMĚŘENÝCH DAT CHOVÁNÍ ČLOVĚKA PŘI ŘÍZENÍ LETU LETOUNU*
- [X.] Vytvořil autor práce
- [XI.] Schématický diagram funkce v \mathbb{R}^2 a několika iterací, převzato z:
<http://goo.gl/TiJx5r>
- [XII.] Příklad algoritmu gradient descent, převzato z: <http://goo.gl/XfGQhQ>

Přílohy

1# Grafický návod pro System Identification Toolbox

Toolbox

Hlavní okno Systém Identification Toolboxu vyvolané zadáním příkazu *ident* do okna Command Window v Prostředí MATLAB®



1 Import dat, se kterými chceme pracovat



2 Data v časové oblasti

The screenshot shows the 'Import Data' dialog box with the following settings:

- Data Format for Signals:** Time-Domain Signals
- Input Properties:** InterSample: zoh, Period: inf
- Workspace Variable:** Input: [], Output: []
- Channel Names:** Input: [], Output: []
- Data Information:** Data name: mydata, Starting time: 1, Sampling interval: 1, Less button
- Physical Units of Variables:** Input: [], Output: []
- Notes:** []

Workspace Variable – proměnné ve Workspace (např.: Ym, Up)

Channel Names – názvy proměnných (např.: výchylka, výška)

Data Information – název dat, počáteční čas (0), vzorkovací perioda (0.05)

Physical Units of Variables – názvy fyzikálních jednotek vstupu a výstupu

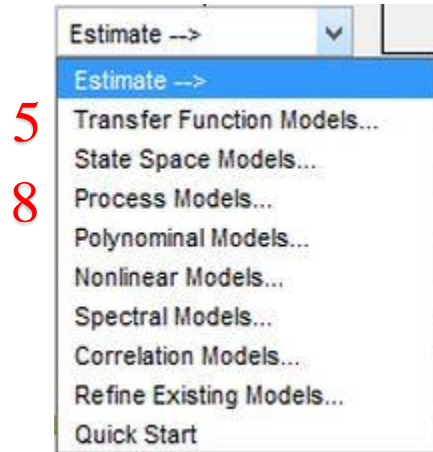
3 Importovaná data

The screenshot shows the 'Import data' dialog box with the following settings:

- Import data:** mydata

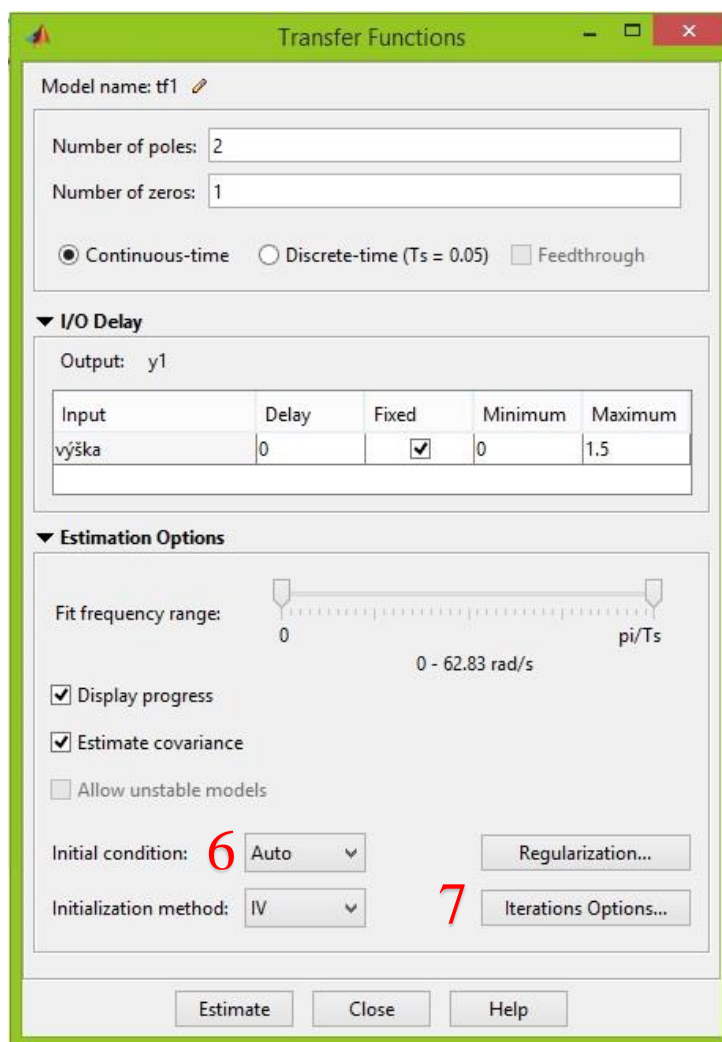
Tyto data se dají zobrazit zaškrtnutím políčka *Time plot* v nabídce Data Views. Jednotlivá data se dají přesouvat přetažením myši po nabídce.

4 Nabídka *Estimate*



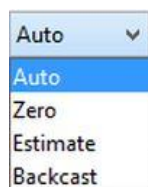
Pro analýzu dat slouží nabídky *Transfer Function Models* a *Process Models*.

5 Transfer Function Models

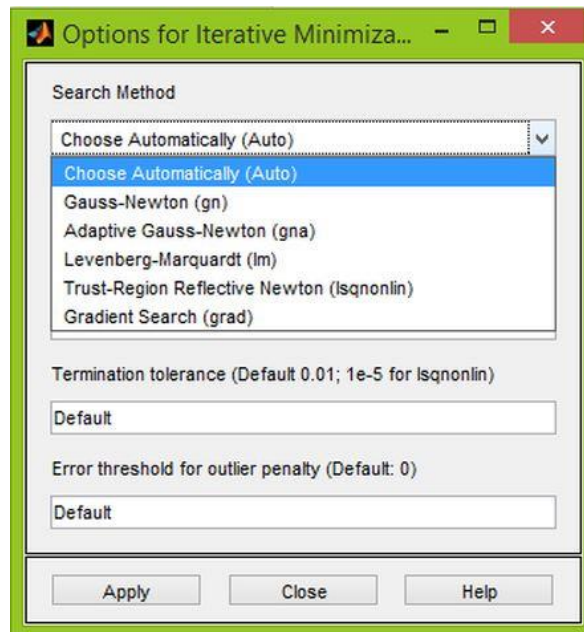


Zde se nastavuje počet pólů a nul, časové zpoždění a nastavení odhadu.

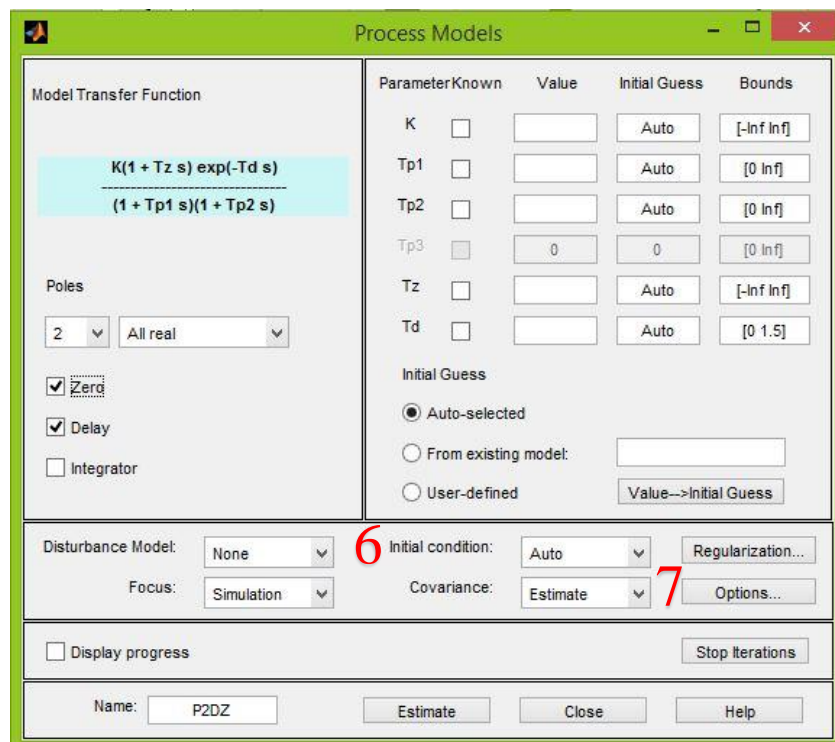
6 Počáteční podmínky



7 Výběr metod pro identifikaci

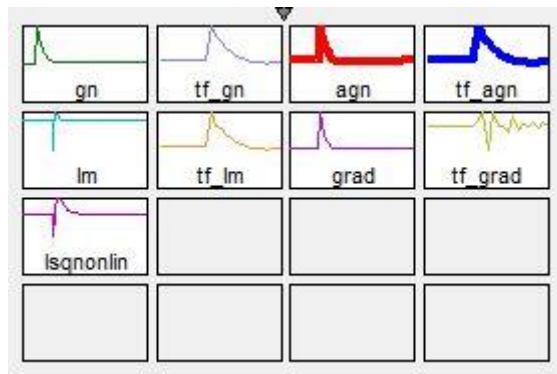


8 Process Models



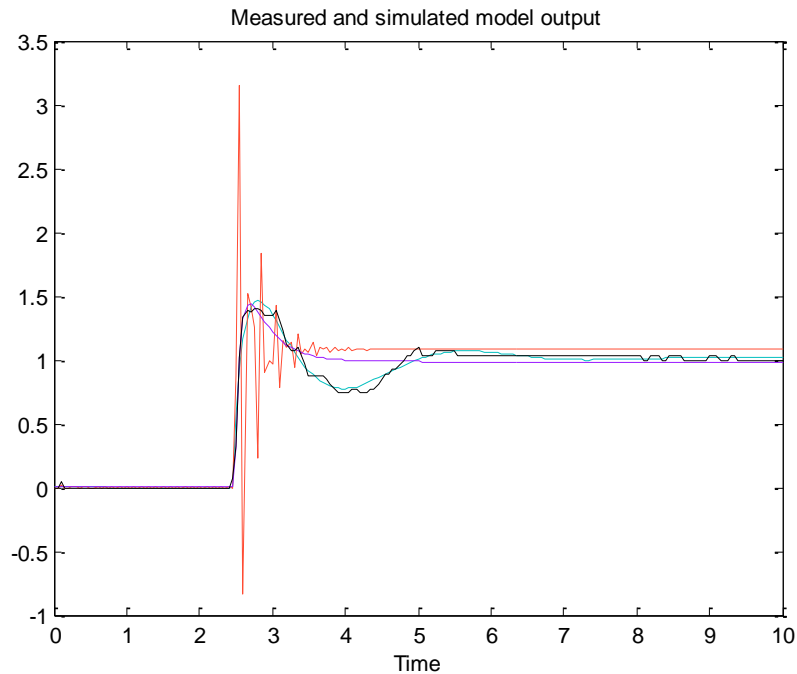
V nabídce Process Models se na levé straně nastavuje omezený tvar přenosové funkce, na pravé straně jsou pak vidět vypočtené parametry jednotlivých koeficientů a také okna pro zadání počátečních odhadů a mezí. Ve spodní části můžeme zavádět další vlastnosti ovlivňující výsledný výpočet. Základní název značí tvar přenosové funkce. (počet pólů, nul, zpoždění,...)

9 Nabídka vypočítaných modelů



Jednotlivé modely se dají po jejich zvýraznění zobrazit a porovnat v okně po zakliknutí *Model Outputs* v nabídce Model Views.

#2: Ukázka chyby identifikační metody



3#: Vzorový m-file algoritmizace použití Systém Identification Toolboxu

```
%%%% m-file pro Gauss-Newton metodu %%%%
%%%% použití Transfer model function %%%%

clc
clear
% načteme: T, Up (výška), Yid, Ym (výchylka)
[FileName,PathName] = uigetfile('*','Vyberte datový soubor');
if FileName ~= 0
    load([PathName FileName] , '-mat');
    h=plot(T,Up,T,Ym);
end

% příprava dat
data=iddata(Ym,Up,0.05);

% nastavení počátečních podmínek
opt=procestOptions;
opt.InitialCondition='zero';
opt.Display='on';
opt.SearchMethod='gn';
iodelay=0.45;
```



```
% odhad přenosové funkce
np=3;
nz=2;
sys = tfest(data,np,nz,iodelay,opt);

% parametry přenosové funkce
par=getpvec(sys, 'free')
[z,p,k] = zpndata(sys);

% porovnání dat
compare(data,sys);
get(sys)
```