



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

APLIKACE PRO ANALÝZU PARAMETRŮ MCRUEROVÝCH MODELŮ

APPLICATION FOR ANALYSIS OF MCRUER'S MODELS PARAMETERS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Ondřej Pohorský

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Miroslav Jirgl, Ph.D.

BRNO 2022

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Ondřej Pohorský

ID: 221011

Ročník: 3

Akademický rok: 2021/22

NÁZEV TÉMATU:

Aplikace pro analýzu parametrů McRuerových modelů

POKyny PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout a následně implementovat aplikaci pro identifikaci a analýzu parametrů McRuerových modelů používaných pro modelování chování lidského operátora v systémech člověk-stroj. Aplikace by měla být realizována v prostředí MATLAB s využitím dostupných nástrojů.

1. Seznamte se s McRuerovou teorií Crossover Law a souvisejícími modely popisujícími chování lidského operátora při řízení základních dynamických systémů.
2. Seznamte se s nástrojem MATLAB System Identification Toolbox a prostředím App Designer.
3. Seznamte se s formátem a strukturou dodaných naměřených dat reprezentujících odezvy lidského operátora při řízení jednoduchých soustav.
4. Navrhněte a popište koncept aplikace pro automatizované zpracování těchto dat, identifikaci parametrů odpovídajících McRuerových modelů a prezentaci výsledků.
5. Implementujte tuto aplikaci v prostředí MATLAB App Designer.
6. Otestujte aplikaci na dostupné sadě dat.
7. Sepište dokumentaci aplikace včetně uživatelského manuálu.

DOPORUČENÁ LITERATURA:

MULDER, Max, Daan M. POOL, David A. ABBINK, et al. Manual Control Cybernetics: State-of-the-Art and Current Trends. IEEE Transactions on Human-Machine Systems [online]. 2018, 48(5), 468-485 [cit. 2021-9-7]. ISSN 2168-2291. Dostupné z: doi:10.1109/THMS.2017.2761342

Termín zadání: 7.2.2022

Termín odevzdání: 23.5.2022

Vedoucí práce: Ing. Miroslav Jirgl, Ph.D.

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato bakalářská práce se věnuje zejména návrhu a způsobu implementace aplikace pro automatickou analýzu parametrů McRuerových modelů. Obsahem práce je také vysvětlení jednotlivých McRuerových modelů vycházejících z teorie Crossover law. Další součástí je popis rozšiřujícího modulu Matlabu System Identification Toolbox a způsoby, kterými lze z naměřených dat získat parametry modelu. V práci se lze také dočíst jakým způsobem jsou získávána data pro sestavování modelů člověka jako regulátoru. Aplikace je navržena ve vývojovém prostředí MATLAB App designer, které je v práci blíže popsáno.

Klíčová slova

Aplikace, McRuerovy modely, Crossover law, System Identification toolbox, MATLAB App designer.

Abstract

This bachelor's thesis deals with design and implementation of Application for automatic analysis parameters of McRuer's models. This work contains an explanation of McRuer's models based on Crossover law theory. Next part is description of System Identification toolbox and description of how to get model from measured data. In this work you can find information about data acquisition for human model estimation. Application is designed in MATLAB App designer, which is described in more detail in this work.

Keywords

Application, McRuer models, Crossover law, System Identification toolbox, MATLAB App designer.

Bibliografická citace

POHORSKÝ, Ondřej. Aplikace pro analýzu parametrů McRuerových modelů. Brno, 2022. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/142562>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Miroslav Jirgl.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	<i>Ondřej Pohorský</i>
VUT ID studenta:	<i>221011</i>
Typ práce:	<i>Bakalářská práce</i>
Akademický rok:	<i>2021/22</i>
Téma závěrečné práce:	<i>Aplikace pro analýzu parametrů McRuerových modelů</i>

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 20. května 2022

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Miroslavu Jirglovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé semestrální práce.

V Brně dne: 20. května 2022

podpis autora

Obsah

SEZNAM OBRÁZKŮ	8
SEZNAM TABULEK.....	9
1. ÚVOD.....	10
2. MODELÝ ČLOVĚKA NAHRAZUJÍCÍ REGULÁTOR.....	11
2.1 MATEMATICKÉ MODELÝ LIDSKÉHO OPERÁTORA.....	11
2.2 TEORIE CROSSOVER LAW	12
2.3 GROSS MODEL.....	13
2.4 PRECISION MODEL	14
3. SYSTEM IDENTIFICATION TOOLBOX	17
3.1 PRACOVNÍ POSTUP IDENTIFIKACE SYSTÉMU	17
3.2 PŘÍKAZY PRO SESTAVENÍ MODELU	18
3.3 APLIKACE SYSTEM IDENTIFICATION	18
3.3.1 <i>Postup identifikace v aplikaci</i>	19
3.4 POSTUP IDENTIFIKACE POMOCÍ PŘÍKAZŮ V PROSTŘEDÍ MATLAB	21
3.4.1 <i>Identifikace modelů ve tvaru přenosové funkce</i>	21
3.4.2 <i>Identifikace procesních modelů</i>	22
3.4.3 <i>Odhad dopravního zpoždění</i>	24
3.4.4 <i>Vyhodnocení kvality estimace modelu</i>	24
4. MATLAB APP DESIGNER.....	25
4.1 POPIS VÝVOJOVÉHO PROSTŘEDÍ	25
4.2 PRACOVNÍ POSTUP VÝVOJE APLIKACÍ.....	26
4.3 DISTRIBUCE APLIKACÍ V MATLAB APP DESIGNER	27
5. NAMĚŘENÁ DATA REPREZENTUJÍCÍ CHOVÁNÍ LIDSKÉHO OPERÁTORA	28
5.1 PROCES ZÍSKÁVÁNÍ DAT	28
5.2 FORMÁT A STRUKTURA DAT.....	29
6. KONCEPT APLIKACE	32
6.1 POPIS OKNA APLIKACE	32
6.2 PRINCIP FUNKCE APLIKACE	34
7. UŽIVATELSKÝ MANUÁL APLIKACE	39
7.1 SPUŠTENÍ APLIKCE.....	39
7.2 PRACOVNÍ POSTUP IDENTIFIKACE MODELU	39
8. UKÁZKA FUNKCE APLIKACE	42
9. ZÁVĚR.....	47
LITERATURA.....	48
SEZNAM SYMBOLŮ A ZKRATEK	49
SEZNAM PŘÍLOH.....	50

SEZNAM OBRÁZKŮ

Obrázek 1 - Blokové schéma regulačního obvodu člověka [2]	11
Obrázek 2 - Okno aplikace System Identification	19
Obrázek 3 - Průběh signálu ze systému a zadaného signálu	20
Obrázek 4 - Ukázka okna model info	21
Obrázek 5 - Ukázka identifikace modelu příkazem tfest	23
Obrázek 6 - Ukázka identifikace modelu příkazem procest	23
Obrázek 7 - Vývojové prostředí MATLAB App designer	25
Obrázek 8 - Ukázka vytvoření callbacku	26
Obrázek 9 - Ukázka vytvořené části kódu callbacku	26
Obrázek 10 - Schéma simulátoru v MATLAB Simulink	28
Obrázek 11 - Základní schéma regulačního obvodu	30
Obrázek 12 – Struktura výstupních dat ze simulátoru	30
Obrázek 13 - Obsah struktury Fs	30
Obrázek 14 – Obsah struktury out	30
Obrázek 15 - Ukázka naměřených hodnot regulační odchylky	31
Obrázek 16 – Okno aplikace – záložka Identifikace dílčích úseků	33
Obrázek 17 – Okno aplikace – záložka Kompletní zobrazení naměřených dat	33
Obrázek 18 – Okno aplikace – záložka Zobrazení výsledků estimace všech úseků	34
Obrázek 19 - Vývojový diagram aplikace	38
Obrázek 20 - Ukázka výběrového seznamu EQ	40
Obrázek 21 - Ukázka výběrového seznamu NMS	40
Obrázek 22 – Ukázka funkce 1	42
Obrázek 23 - Ukázka funkce 2	43
Obrázek 24 - Ukázka funkce 3	44
Obrázek 25 - Ukázka funkce 4	45
Obrázek 26 - Ukázka funkce 5	45
Obrázek 27 - Ukázka funkce 6	46

SEZNAM TABULEK

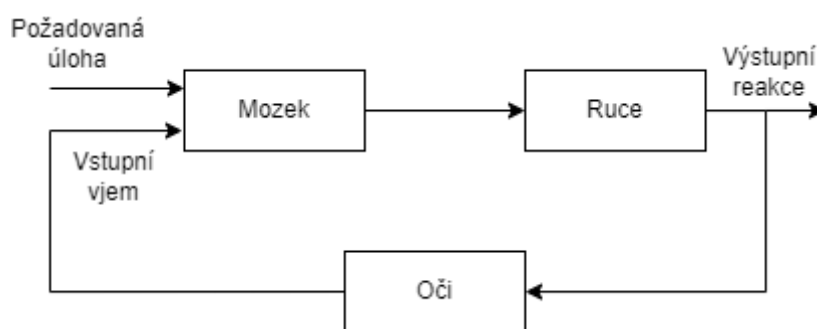
Tabulka 1 - Ukázka aproximačních přenosových charakteristik pro různé přenosy soustav [5]	13
Tabulka 2 - Příklady přenos. funkcí popisující chování člověka v závislosti na reg. soustavě [5]	14
Tabulka 3 - Příkazy pro výpočet různých typů modelů [3].....	18

1. ÚVOD

Hlavním cílem této práce je návrh a následná implementace aplikace pro automatické vyhodnocení parametrů aproximačních modelů člověka jako regulátoru. Aby bylo možné provést návrh této aplikace je nutné pochopit problematiku sestavování aproximačních přenosových funkcí modelů člověka. Modelů člověka jako regulátoru existuje několik. Modely vysvětlené v této práci vychází z teorie Crossover law kterou představil prof. McRuer. Každý z těchto modelů je vhodný pro modelování různých situací. V aplikaci budou implementovány všechny tyto modely a umožní uživateli si zvolit který model chce použít pro výpočet. Dalším krokem při tvorbě této aplikace je pochopení principu měření dat na simulátoru, a hlavně způsobem ukládání dat a jejich strukturou. Implementaci algoritmů, které vyhodnocují parametry daného modelu z předzpracovaných naměřených dat zajišťuje rozšiřující modul Matlabu System Identification toolbox. Vývojové prostředí zvolené pro tvorbu aplikace je grafický editor MATLAB App designer. Toto prostředí je nejvhodnější ze dvou důvodů. Zaprvé je to jednoduché a intuitivní prostředí pro tvorbu aplikací. Druhým důvodem je, že vývojové prostředí MATLAB App designer je implementováno přímo v prostředí MATLAB, což zajišťuje kompatibilitu s modulem System Identification toolbox.

2. MODEL ČLOVĚKA NAHRAZUJÍCÍ REGULÁTOR

Úkolem regulátoru je v ideálním případě odstranit regulační odchylku, tedy přenést žádanou hodnotu na výstup. Aby byl regulátor schopen docílit odstranění regulační odchylky, potřebuje zpětnou vazbu.[6] Pokud jako regulátor uvažujeme člověka, tak je zpětná vazba získána pomocí očí. [2] Grafické znázornění regulačního obvodu člověka je zobrazeno na obrázku 1.



Obrázek 1 - Blokové schéma regulačního obvodu člověka [2]

S modelováním člověka jako regulátoru přišel prof. J. Rasmussen. Jeho základní model je rozdělen do tří úrovní. Nejnižší úroveň je organizační – tato úroveň představuje podvědomě naučené automatické činnosti pro řízení. Nad touto úrovní je úroveň koordinační – ta představuje řízení na základě pravidel, vyžaduje tedy rozhodování. Nejvyšší je úroveň organizační – ta spočívá v řízení na základě zkušeností. Na této úrovni tedy člověk musí přemýšlet a rozhodnout na základě předchozích znalostí, aby co neoptimálněji dosáhl požadovaného cíle. [5]

2.1 Matematické modely lidského operátora

Obecně se interakce člověka se strojem označují MMS (Man Machine System). Tyto interakce mohou mít různé úrovně složitosti. Od nejjednodušších jako například pouhé sledování výstupních dat, nebo kontrolování chodu nějakého automatického stroje, až po velmi složité úlohy typu řízení raketoplánu. Při popisu komplexnějších úloh se nevyhneme rozdělení na několik veličin. Jedná se tedy takzvané MIMO systémy (Multi Input Multi Output). Modelování takovýchto systémů je velmi složitá úloha. Ovšem tato

práce je zaměřena pouze na modelování úloh s jednou vstupní hodnotou a jednou výstupní veličinou, tedy systémy nazývané SISO (Single Input Single Output). [5]

2.2 Teorie Crossover law

Autorem této teorie je prof. Duan T. McRuer, který se se svým týmem zabýval dynamický popisem člověka a díky této teorii patří mezi nejvýznamnější osoby v oblasti modelování člověka jako regulátoru.

Teorie crossover law vychází z principu teorie lineárních dynamických systémů. Tato teorie říká, že se člověk snaží adaptovat na danou regulovanou soustavu tak, aby platila rovnice 1. [1]

$$F_0(s) = F_R(s) F_S(s) = \frac{\omega_c}{s} e^{-s\tau}, \quad (1)$$

,kde	s	–	$j\omega$, laplaceův operátor
	$F_0(s)$	–	přenos otevřené regulační smyčky
	$F_R(s)$	–	přenos regulátoru (člověka)
	$F_S(s)$	–	přenos regulované soustavy
	ω_c	–	frekvence řezu [rad/s]
	τ	–	reakční zpoždění [s]

Z rovnice 1 vyplývá, že se člověk snaží udržet přenos otevřené smyčky tak, aby ve frekvenčních charakteristikách měl v okolí frekvence řezu integrační charakter. To odpovídá sklonu -20 dB/dek.

Přenosové funkce aproximující člověka jako regulátor pro různé přenosové funkce jsou zobrazeny v tabulce 1. Z tabulky lze vyčíst, že člověk přizpůsobuje své chování tak, aby výsledný přenos otevřené smyčky měl tvar $F_0(s) = \frac{K_r \cdot K_s}{s} e^{-s\tau}$.

Přenosová funkce popisující regulovanou soustavu v okolí frekvence řezu	Přenosová funkce představující chování člověka jako regulátoru	Výsledný přenos otevřené smyčky
$F_S(s) = K_S$	$F_r(s) \approx \frac{K_r}{s} \cdot e^{-s\tau}$	
$F_S(s) = \frac{K_S}{s}$	$F_r(s) \approx K_r \cdot e^{-s\tau}$	
$F_S(s) = \frac{K_S}{s^2}$	$F_r(s) \approx K_r \cdot s \cdot e^{-s\tau}$	$F_0(s) \approx \frac{K_r \cdot K_S}{s} e^{-s\tau}$
$F_S(s) = \frac{K_S}{s \cdot (s + a)}$	$F_r(s) \approx K_r \cdot (s + a) \cdot e^{-s\tau}$	

Tabulka 1 - Ukázka aproximačních přenosových charakteristik pro různé přenosy soustav [5]

2.3 Gross model

Modely vycházejí z teorie Crossover law jsou použitelné pouze v okolí frekvence řezu. Z tohoto důvodu vznikl zobecněný model, který je rozšířený na větší frekvenční rozsah. Tento model je označován jako Gross („hrubý“) model. Jeho operátorový přenos popisuje rovnice 2. [5]

$$F_R(s) = K_R \frac{(T_L s + 1)}{(T_I s + 1)} e^{-s\tau} \quad (2)$$

, kde

s	–	$j\omega$, Laplaceův operátor
K_R	–	zesílení [-]
T_L	–	derivační (prediktivní) časová konstanta [s]
T_I	–	integrační (setrvačná) časová konstanta [s]
τ	–	reakční zpoždění [s]

Poměr konstant T_I a T_L určuje charakter přenosové funkce modelu člověka jako regulátoru. Charakter tedy může být buď integrační, nebo derivační.

V následující tabulce jsou uvedeny příklady výsledných aproximačních přenosových funkcí popisující chování člověka v závislosti na regulované soustavě a charakteru přenosové funkce [5]

Přenosová funkce popisující regulovanou soustavu	Charakter přenosové funkce	Výsledná přenosová funkce popisující člověka
$F_S(s) = K_S$	Integrační	$F_R(s) = \frac{K_R}{(T_i s + 1)} e^{-s\tau}$
$F_S(s) = \frac{K_S}{s}$	Zesílení	$F_R(s) = K_r \cdot e^{-s\tau}$
$F_S(s) = \frac{K_S}{s^2}$	Derivační	$F_R(s) = K_r (T_L s + 1) e^{-s\tau}$
$F_S(s) = \frac{K_S}{s \cdot (s + a)}$	Derivační, $T_L \approx 1/a$	$F_R(s) = K_r (T_L s + 1) e^{-s\tau}$
	Pro $T \gg \tau$: Derivační	$F_R(s) = K_r (T_L s + 1) e^{-s\tau}$
$F_S(s) = \frac{K_S}{T^2 s^2 + 2\xi Ts + 1}$	Pro $T \ll \tau$: Integrační	$F_R(s) = \frac{K_R}{(T_i s + 1)} e^{-s\tau}$

Tabulka 2 - Příklady přenos. funkcí popisující chování člověka v závislosti na reg. soustavě [5]

2.4 Precision model

Dalším rozšířením teorie Crossover law je zahrnutí modelu popisující neuromuskulární systém člověka do Gross modelu. Takto vzniklý model byl nazván Precision model. Popis neuromuskulárního systému vyháží z fyziologického a neurologického popisu člověka. Obecný tvar přenosové funkce Precision modelu popisuje rovnice 3. [5]

$$F_R(s) = K_R \frac{(T_L s + 1)}{(T_i s + 1)} e^{-s\tau} \cdot \frac{1}{\left(\frac{s}{\omega_N}\right)^2 + \frac{2\xi_N}{\omega_N} s + 1} \quad (3)$$

, kde	s –	$j\omega$, laplaceův operátor
	K_r –	zesílení [-]
	T_L –	derivační (prediktivní) časová konstanta [s]
	T_I –	integrační (setrvačná) časová konstanta [s]
	τ –	reakční zpoždění [s]
	ω_N –	vlastní frekvence kmitů neuromuskulárního systému [rad/s]
	ξ_N	tlumení neuromuskulárního systému [-]

Přenosová funkce tohoto modelu má poměrně složitý tvar. Z toho důvodu vzniklo několik aproximací, které přímo vycházejí z tvaru přenosové funkce Precision modelu. Tyto aproximace nemají zásadní vliv na jeho výslednou odezvu, protože jsou zaměřeny na oblast v okolí frekvence řezu, která se nejčastěji pohybuje v řádu jednotek rad/s. [5]

Prvním příkladem aproximace je hrubá aproximace dynamiky neuromuskulárního systému pomocí čistého dopravního zpoždění τ_N . Toto dopravní zpoždění lze přičíst k celkovému dopravnímu zpoždění Gross modelu. Výsledný tvar přenosové funkce popisuje rovnice 4. [5]

$$F_R(s) = K_R \frac{(T_L s + 1)}{(T_I s + 1)} e^{-s(\tau + \tau_N)} \quad (4)$$

, kde	s –	$j\omega$, laplaceův operátor
	K_r –	zesílení [-]
	T_L –	derivační (prediktivní) časová konstanta [s]
	T_I –	integrační (setrvačná) časová konstanta [s]
	τ –	reakční zpoždění [s]
	τ_N –	dopravní zpoždění aproximující neuromuskulární systém [s]

Tento aproximační tvar přenosové funkce je vhodný zejména pro experimenty, které se zaměřují na vyhodnocení maximální celkové doby reakce.

Dalším příkladem aproximace Precision modelu je aproximace neuromuskulárního systému setrvačným členem prvního řádu. Tento člen obsahuje jednu setrvačnou konstantu T_N , která je označována jako neuromuskulární časová konstanta. Přenosovou funkci této aproximace popisuje rovnice 5. [5]

$$F_R(s) = K_R \frac{(T_L s + 1)}{(T_N s + 1)(T_I s + 1)} e^{-s\tau} \quad (5)$$

, kde	s	–	$j\omega$, laplaceův operátor
	K_r	–	zesílení [-]
	T_L	–	derivační (prediktivní) časová konstanta [s]
	T_I	–	integrační (setrvačná) časová konstanta [s]
	T_N	–	neuromuskulární časová konstanta [s]
	τ	–	reakční zpoždění [s]

Aproximace Precision modelu ve tvaru popisující rovnice 5 se stal výchozím pro mnoho experimentů. Někdy model v tomto tvaru bývá označován jako Tustin-McRuerův. Oblíbenost tohoto modelu je ovlivněna zejména možností fyziologické a neurologické interpretace jednotlivých parametrů. Experimentálně zjištěné hodnoty parametrů tohoto modelu jsou uvedeny v následující části textu. [5]

Zesílení regulátoru K_R – souvisí s poměrem vstupního a výstupního signálu, jeho hodnota podstatně závisí na aplikaci.

Setrvačná časová konstanta T_N – udává reakční zpoždění činnosti člověka dané neuromuskulárním systémem, není závislá na tréninku. Její typická hodnota se pohybuje od 0,05 do 0,2 sekund.

Setrvačná časová konstanta T_I – souvisí s prováděním naučených stereotypů a s postupy danými dlouhodobými zkušenostmi. Její běžná hodnota nabývá hodnot od 0,1 až jednotek sekund.

Prediktivní časová konstanta T_L – odráží schopnost předvídání situací. Tato schopnost se zlepšuje tréninkem a zkušenostmi. Její běžná hodnota nabývá hodnot od 0,2 až jednotek sekund.

Časová konstanta τ – udává zpoždění reakce mozku na pohybový a oční vjem. Tato konstanta může být ovlivněním mozku prodloužena. A to zejména vlivem únavy člověka. V případě, kdy je tato konstanta prodloužena ve větší míře, mohou regulační schopnosti člověka naprosto selhat. Typické hodnoty této konstanty se pohybuji v intervalu 0,2 až 1 sekunda.

3. SYSTEM IDENTIFICATION TOOLBOX

System Identification Toolbox je rozšiřující balíček pro MATLAB, který poskytuje funkce, bloky Simulinku a aplikaci pro konstrukci matematických modelů dynamických systémů z naměřených sad dat. Umožňuje tedy vytváření a používání modelů dynamických systémů, které nelze jednoduše sestavit základními principy. K identifikaci spojitých a diskrétních přenosových funkcí lze použít vstupní a výstupní data v oblasti jak časové, tak frekvenční. [3]

3.1 Pracovní postup identifikace systému

Identifikace systému je iterativní proces, při kterém se identifikují modely s různými strukturami a porovnává se jejich výkonost. Následně se vybere model, který nejlépe popisuje daný systém. [4]

Možný postup identifikace systému je popsán v následujících krocích:

1. Zpracování dat pro identifikaci systému:

- Načtení dat do MATLAB workspace
- Reprezentace dat v System Identification aplikaci nebo jako iddata či idfrd objekt v MATLAB workspace
- Vykreslení časového průběhu dat z důvodu vyšetření chování v časové i frekvenční oblasti.
- Předzpracování dat úpravami jako: odstranění offsetů, interpolace chybějících hodnot, filtrace pro zvýraznění specifického frekvenčního rozsahu, nebo převzorkování s jiným časovým intervalem.

2. Identifikace lineárních nebo nelineárních modelů:

- Přenosová funkce
- Polynomický model
- Diferenční nebo diferenciální rovnice
- Stavový popis

3. Ověření modelů:

Pokud nebylo dosaženo uspokojivého modelu, lze zkusit zvolit jinou strukturu modelu nebo zkusit použít jiný identifikační algoritmus. V některých případech je možné dosáhnout zlepšení výsledků zahrnutím modelu šumu. Případně je nutné předzpracovat data před dalším odhadem. Například pokud je v datech příliš mnoho vysokofrekvenčního šumu je nutné ho filtrovat, nebo převzorkovat data.

4. Zpracování modelů:

- Převod mezi spojitou a diskrétní časovou oblastí.
- Změna reprezentace modelu
- Extrakce numerických dat modelu
- Subreference, spojení a sloučení modelů
- Linearizace nelineárních model

3.2 Příkazy pro sestavení modelu

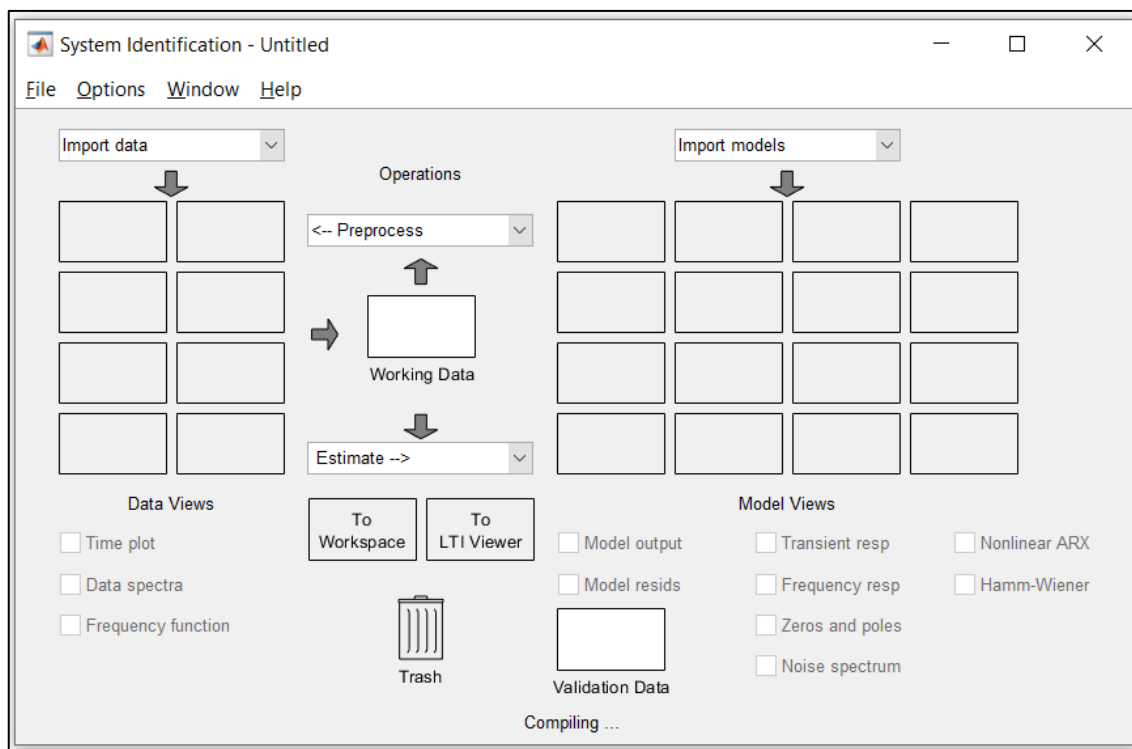
V následující tabulce je zobrazen výběr některých příkazů System Identification Toolbox.

Typ modelu	Příkazy
Přenosová funkce	tfest
Procesní modely	procest
Lineární vstupně-výstupní polynomické modely	armax arx bj iv4 oe polyest
Frequency-response modely	etfe spa spafdr
Korelační modely	cra impulsect
Lineární modely v časové oblasti	ar arx ivar
Lineární a nelineární modely	pem

Tabulka 3 - Příkazy pro výpočet různých typů modelů [3]

3.3 Aplikace System Identification

Jednoduchým a intuitivním způsobem, jak získat model systému je použití System Identification aplikace. Pro spuštění této aplikace stačí do command window v Matlabu zadat příkaz `systemIdentification`. Tím se otevře okno aplikace, které je zobrazeno na obrázku 2.



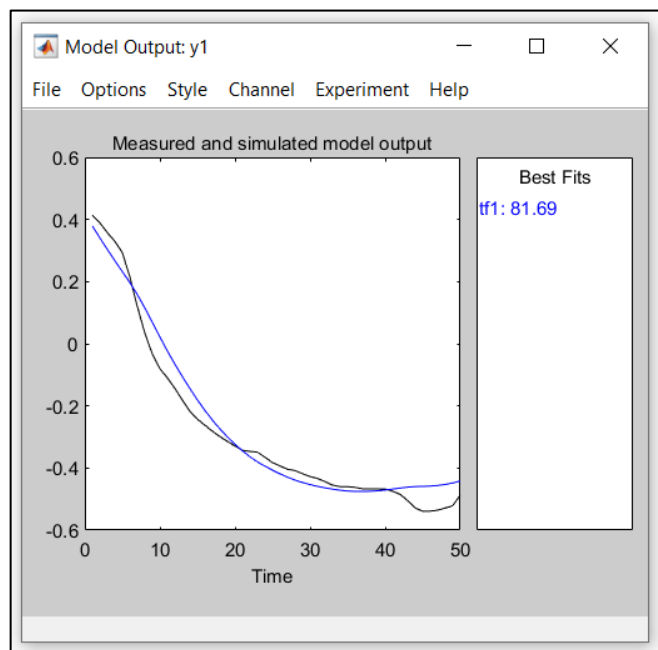
Obrázek 2 - Okno aplikace System Identification

3.3.1 Postup identifikace v aplikaci

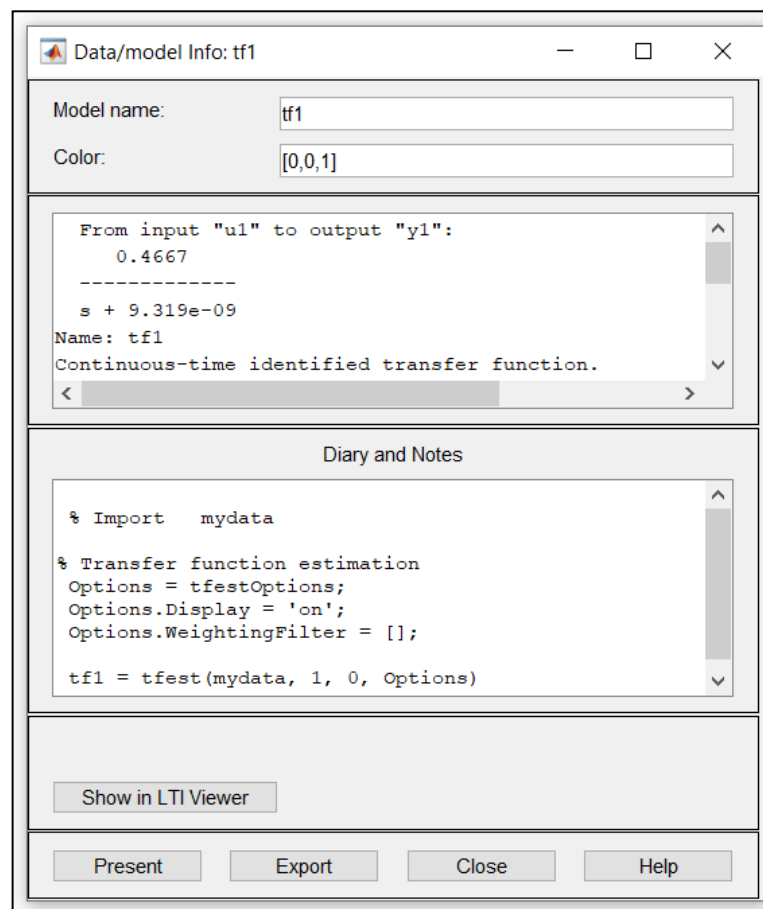
V této části práce je popsán způsob identifikace systému na základě naměřených dat získaných na simulátoru (viz kapitola 5). Nejprve je nutné do MATLAB Workspace načíst naměřená data. To lze provést příkazem `load`. Data použitá v této ukázce byla dále předzpracována (způsob předzpracování je podrobně popsán v kapitole 6) v MATLAB skriptu a následně uložena do vektoru `u` (vstupní data) a vektoru `y` (výstupní data). Poté je možné data načíst do aplikace kliknutím na rozbalovací okno `import data`. Zde je nutné vybrat typ vstupních dat. V tomto případě jsou data naměřená v časové oblasti. Po výběru typu dat se otevře okno `Import data`. Zde se zadají vektory vstupních a výstupních dat. Dále je nutné zadat periodu vzorkování dat a počáteční čas. Případně je možné data pojmenovat. Stiskem tlačítka `import` se data načtou a zobrazí se v levé části okna aplikace. V tomto okamžiku lze zavřít okno `import data`.

Dalším krokem je v záložce Estimate vybrat typ modelu, který chceme identifikovat. V našem případě chceme získat přenosovou funkci. Vybereme tedy Transfer function model. Tím se otevře okno Transfer functions. V tomto okně zadáme počet pólů a počet nul požadované přenosové funkce a zvolíme, zda se jedná spojitý, či diskrétní systém. V tomto případě hledáme přenosovou funkci s jedním pólem a jednou nulou. Po zadání parametrů stiskneme tlačítko Estimate a zavřeme okno.

Nyní se v pravé části okna zobrazí identifikovaný model. Dvojklikem na vygenerovaný model se zobrazí okno s informacemi o daném modelu. Ukázku okna s informacemi o modelu zobrazuje Obrázek 4. V tomto okně najdeme tvar přenosové funkce modelu. Zaškrtnutím checkboxu v okně aplikace System Identification se zobrazí průběh výstupního signálu systému ve srovnání se zadaným výstupním signálem. Ukázka okna s tímto průběhem je na obrázku 3.



Obrázek 3 - Průběh signálu ze systému a zadaného signálu



Obrázek 4 - Ukázka okna model info

3.4 Postup identifikace pomocí příkazů v prostředí MATLAB

Pro sestavení modelu v command window nebo v MATLAB skriptu je nutné nejdříve vytvořit input-output data objekt. To provedeme voláním funkce `iddata`, které dáme jako parametry výstupní data, vstupní data a periodu vzorkování. Ukázka volání funkce: `data = iddata(y, u, Ts)`.

3.4.1 Identifikace modelů ve tvaru přenosové funkce

Po vytvoření objektu `iddata` můžeme zavolat v command window příkaz `tfest` pro vypočtení modelu ve tvaru přechodové funkce dle tabulky 3. Jako parametry příkazu zadáváme objekt `iddata`, počet pólů a počet nul. Ukázka volání příkazu: `tfest(data, 1, 1)`. Tím se do command window vypíše tvar přenosové funkce a další informace o vytváření modelu. Ukázka identifikace pomocí příkazu `tfest` v MATLAB command window je na obrázku 5. Případně je možné uložit model systému do objektu `idtf` voláním ve tvaru:

```
system = tfest(data, 1, 1);
```

Poznámka: Objekt typu idtf reprezentuje model jako přenosovou funkci s identifikovatelnými parametry [7].

3.4.2 Identifikace procesních modelů

Dalším příkazem z tabulky 3, který je vhodný použít pro vypočtení parametrů modelu je příkaz `procest`. Tento příkaz provádí estimaci tzv. procesních modelů. Procesní modely jsou předurčené spojité přenosové funkce popisující lineární dynamické systémy složené z jedné nebo více následujících částí: [4][7]

Statické zesílení - K_p

Časové konstanty pólů - T_{pk}

Časová konstanta nuly - T_z

Dopravní zpoždění - T_d

Vynucený integrátor

Pro komplexní póly jsou časové konstanty pojmenovány T_w a model obsahuje koeficient kmitání ξ . Ukázka jednoho z možných tvarů procesního modelu je na následující rovnici 6.

$$F(s) = \frac{K_p}{s(1 + sT_{p1})} e^{-sT_d} \quad (6)$$

,kde	s –	$j\omega$, Laplaceův operátor
	K_p –	statické zesílení [-]
	T_{p1} –	časová konstanta pólu [s]
	T_d –	dopravní zpoždění [s]

Pro sestavení procesního modelu voláme tedy příkaz `procest`, kterému dáváme jako první parametr objekt `iddata` uchovávající naměřená data. Druhý parametr příkazu `procest` je `type`, který představuje strukturu procesního modelu. Tato struktura je reprezentována pomocí textového řetězce, který reprezentuje akronym struktury modelu. Tento akronym se skládá z následujících součástí: [7]

P – Póly. Všechny akronymy budou obsahovat toto písmeno, protože procesní model musí obsahovat alespoň jeden pól.

0, 1, 2 nebo 3 – Číslo udávající počet pólů (časových konstant) modelu.

I – Vynucený integrátor.

D – Dopravní zpoždění.

Z – Model obsahuje nulu.

U – Kmitavý model (komplexně sdružené póly) povolen. Počet pólu musí být minimálně 2. [7]

Akronym pro model popisující rovnice 6 by tedy vypadal následovně: P1ID. Tyto akronymy musí vždy začínat písmenem P, které následuje číslo udávající počet pólů. Na pořadí ostatních písmen nezáleží.

```
>> tfest(data,2,0)

ans =

    From input "u1" to output "y1":
           0.1079
    -----
    s^2 + 0.7518 s + 0.3786
```

Obrázek 5 - Ukázka identifikace modelu příkazem tfest

```
>> procest(data, 'P1ID')

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

           Kp = 0.012951
           Tp1 = 0.71223
           Td = 0.2923
```

Obrázek 6 - Ukázka identifikace modelu příkazem procest

Na obrázku 6 lze vidět ukázkou identifikace pomocí příkazu `procest`. Příkazem `procest` lze model uložit do objektu typu `idproc` voláním ve tvaru:

```
system = procest(data, 'P1DI')
```

Poznámka: Objekt typu `idproc` reprezentuje model jako časově spojitý procesní model identifikovatelnými parametry [7].

3.4.3 Odhad dopravního zpoždění

Při sestavování modelů člověka jako regulátoru lze předpokládat, že tyto modely budou obsahovat dopravní zpoždění. Příkazy `tfest` a `procest` popsané výše estimaci dopravního zpoždění umožňují automaticky. Tento automatický odhad zpoždění ovšem může způsobovat chyby, nebo dokonce k estimaci dopravního zpoždění nemusí vůbec dojít. Z tohoto důvodu je vhodné dopravní zpoždění z dat analyzovat předem. K tomuto účelu slouží příkaz `delayest`. Parametrem tohoto příkazu jsou naměřená data v objektu `iddata`. Takto získané dopravní zpoždění je možné použít jako další parametr příkazů `tfest` a `procest`, přičemž je při zadávání parametru takto vyhodnoceného dopravního zpoždění nutné zpoždění násobit periodou vzorkování. [4] [7]

3.4.4 Vyhodnocení kvality estimace modelu

Nejjednodušším způsobem ověření kvality estimace je nechat na model působit vstupní signál ze kterého byl model sestaven a porovnat jeho časovou odezvu s výstupním signálem ze kterého byl daný model sestaven. K tomuto účelu slouží příkaz `compare`. Pokud zavoláme tento příkaz a jako parametry mu dáme data ve tvaru objektu `iddata` a identifikovaný model vykreslí se graf s průběhem časové odezvy modelu na vstupní signál. Do grafu se také vykreslí průběh výstupního signálu a v legendě grafu se zobrazí procentuální vyjádření shodnosti těchto průběhů. Příkaz `compare` je také možné volat ve tvaru:

```
[y, fit, x0] = compare(data, model)
```

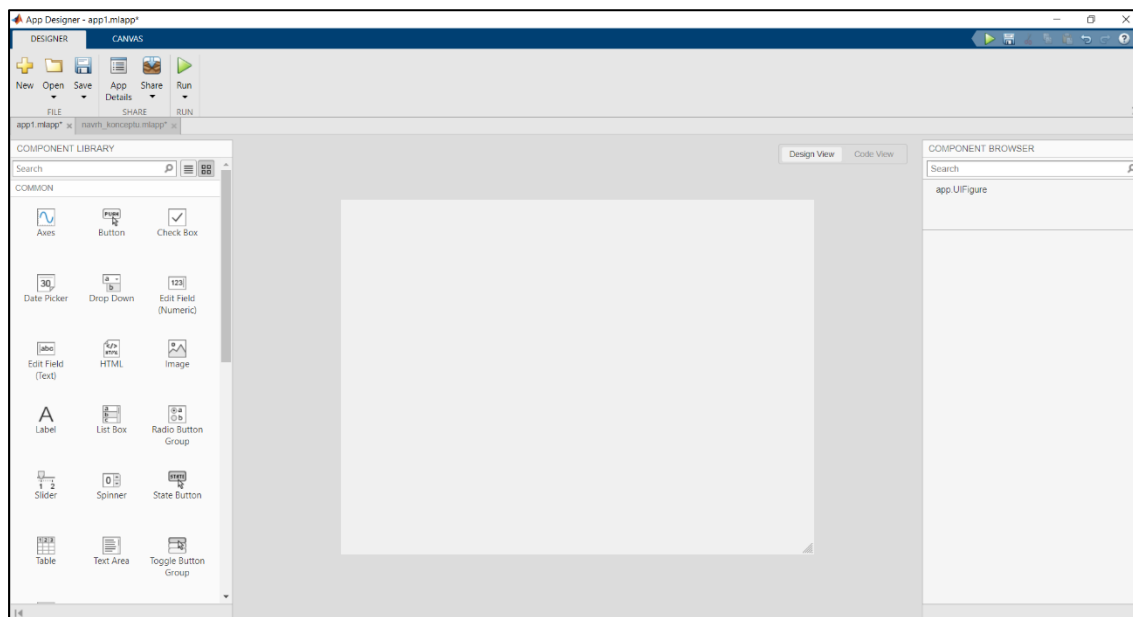
Při tomto způsobu volání nedochází k vykreslení grafů, ale do proměnných v hranatých závorkách jsou uloženy výsledky porovnání. Do proměnné `y` se uloží odezva modelu, do proměnné `fit` se uloží procentuální shodnost a do proměnné `x0` počáteční podmínky. [7]

4. MATLAB APP DESIGNER

MATLAB App Designer je interaktivní vývojové prostředí pro návrh grafického uživatelského rozhraní a programování jeho chování. Poskytuje velkou sadu interaktivních komponent uživatelského rozhraní. Nabízí také možnost automatického přeformátování, aby aplikace detekovala a reagovala na změny velikosti obrazovky. Distribuce aplikace je buď zabalením do instalačního souboru přímo z panelu nástrojů v App designer nebo vytvořením samostatné desktopové nebo webové aplikace. Ukázka vývojového prostředí MATLAB App designer je na obrázku 7. [8]

4.1 Popis vývojového prostředí

V levé části okna vývojového prostředí se nachází knihovna se standardními komponenty, jako například: tlačítka, zaškrtačací pole, výběrové pole, textové pole, číselné pole atd. Knihovna komponent také obsahuje grafy, obrázky a řídicí prvky jako například: posuvníky, knoflíky, budíky, přepínače apod. Dále je možné použít kontejnerové komponenty jako panel, tab, mřížka, určené pro organizaci a rozložení jednotlivých komponentů.

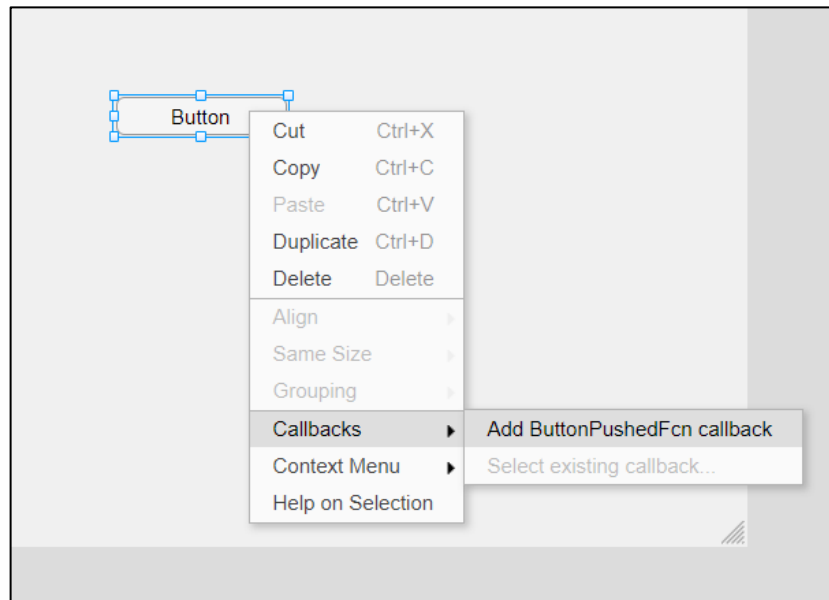


Obrázek 7 - Vývojové prostředí MATLAB App designer

Uprostřed se nachází okno náhledu GUI tvořené aplikace. Do tohoto okna se vkládají jednotlivé komponenty. Po přesunutí komponentu do okna se automaticky vygeneruje objektově orientovaný kód popisující vlastnosti dané komponenty.

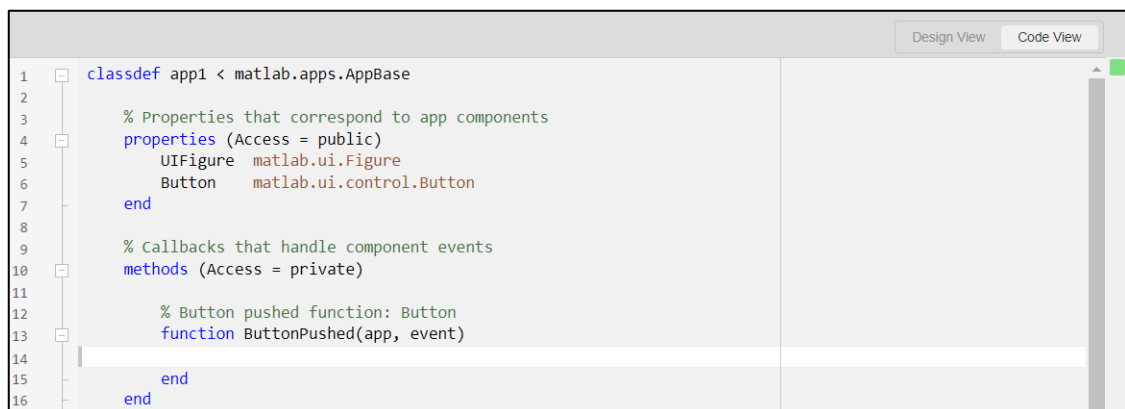
4.2 Pracovní postup vývoje aplikací

Pro vytvoření interakce komponentu, například reakce na kliknutí myši na tlačítko, je nutné vygenerovat obsluhující kód – tzv. callback. Toho docílíme kliknutím pravým tlačítkem na daný komponent a následným výběrem požadovaného callbacku. Vytvoření callbacku ukazuje obrázek 8.



Obrázek 8 - Ukázka vytvoření callbacku

Nahoře nad náhledem GUI okna tvořené aplikace se nachází přepínač, kterým se přepne zobrazení na kód tvořené aplikace. Jak lze vidět na obrázku 9 po vytvoření callbacku na tlačítko se v kódu vytvoří funkce do které stačí dopsat kód, který se má vykonat po stisku tlačítka.



Obrázek 9 - Ukázka vytvořené části kódu callbacku

4.3 Distribuce aplikací v MATLAB App designer

Aplikace vytvořené v MATLAB App designer je možné sdílet několika následujícími způsoby. [7]

Přímé sdílení Matlabovských souborů – tento způsob je nejjednodušší, ale uživatelé musí mít nainstalovaný MATLAB a další jeho rozšíření podstatné pro chod aplikace. Musí také znát způsob spouštění příkazů v command window a vědět, jak spravovat cestu k Matlabu. Přímé sdílení aplikací vytvořených v MATLAB App designer spočívá ve sdílení souboru .mlapp a dalších závislých souborů podstatné pro chod aplikace. [7]

Zabalení aplikace – MATLAB dodává nástroj umožňující zabalení aplikace. Zabalením aplikace vznikne instalační soubor .mlappinstall. Když si uživatelé nainstalují zabalenou aplikaci, zobrazí se na kartě Aplikace v pásu nástrojů MATLAB. Tento způsob také vyžaduje nainstalovaný MATLAB včetně jeho rozšíření podstatných pro chod aplikace, ale pro uživatele je z hlediska spouštění aplikace jednodušší variantou. [7]

Vytvoření webové aplikace – tento způsob umožňuje vytvářet aplikace, které mohou uživatelé v rámci organizace spouštět ve svých webových prohlížečích. Pro vytvoření této webové aplikace je ovšem nutné mít nainstalovaný MATLAB Compiler. Uživatelé musí mít nainstalovaný pouze webový prohlížeč s přístupem k intranetu, ale nemusí mít nainstalovaný MATLAB. [7]

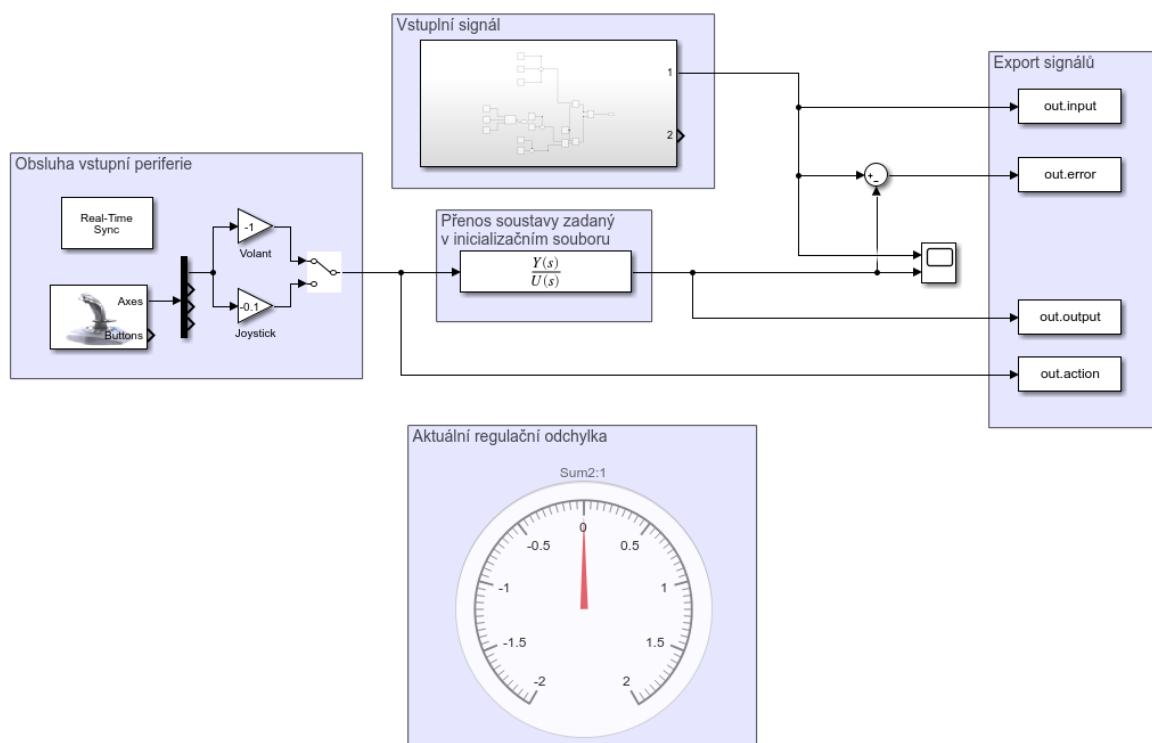
Vytvoření samostatné desktopové aplikace – pokud je třeba aplikaci sdílet s uživateli, kteří nevlastní MATLAB je nutné zvolit tuto metodu distribuce aplikace. K vytvoření samostatné desktopové aplikace je nutné mít nainstalovaný MATLAB Compiler. Pro spuštění aplikace musí mít uživatelé ve svých systémech nainstalovaný MATLAB Runtime. [7]

5. NAMĚŘENÁ DATA REPREZENTUJÍCÍ CHOVÁNÍ LIDSKÉHO OPERÁTORA

V této kapitole je vysvětlen způsob, kterým jsou měřena data pro zpracování. Data jsou měřena na simulátoru, který je blíže popsán v následující podkapitole. Druhá podkapitola vysvětluje formát a strukturu naměřených dat.

5.1 Proces získávání dat

Simulátor pro měření dat potřebuje k počítači připojit vstupní periférii, kterou může být například joystick, nebo volant. Samotný program simulátoru je vytvořen v prostředí MATLAB – Simulink. Celkové schéma simulátoru je na obrázku 10. Toto schéma je rozčleněno na jednotlivé bloky. Tyto bloky jsou podrobně popsány a vysvětleny v následující části textu.



Obrázek 10 - Schéma simulátoru v MATLAB Simulink

První částí simulátoru je blok Obsluha vstupní periferie. Tento blok obstarává příjem signálů ze vstupní periferie (volantu nebo joysticku). Z tohoto bloku je vyveden signál Axes, který reprezentuje natočení volantu (polohu joysticku). Tento signál je následně násoben různými konstantami dle připojené vstupní periferie volantu nebo joysticku, tak aby byl výsledný signál v rozsahu $<-1,1>$. Z tohoto důvodu je nutné přepínačem vybrat kterou vstupní periferii pro měření používáme. Dále se v této části programu nachází blok Real-Time Sync, který zajišťuje synchronizaci vstupní periferie s programem.

Druhá část programu obsahuje bloky pro různé vstupní signály. V této části programu pomocí přepínačů lze vybrat průběh vstupního signálu na kterém bude probíhat měření. Jedná se o signály se sinusovým průběhem, obdélníkovým průběhem a signál s pseudonáhodně měnící se šířkou pulzu.

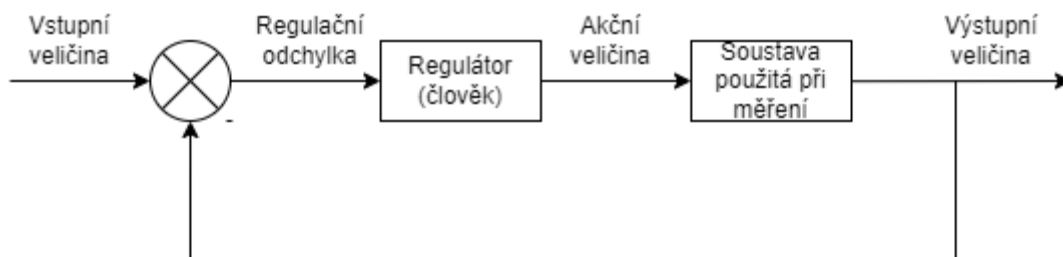
Další část programu obsahuje blok, který reprezentuje soustavu ve formě MATLAB transfer function uloženou předchozím inicializačním skriptem v MATLAB Workspace. Tento blok je připojen za první část popsanou výše.

V části programu Export signálů se nachází bloky, které zajišťují uložení naměřených dat z jednotlivých signálů. Struktura těchto dat je podrobněji rozebrána v následující kapitole.

Poslední částí programu simulátoru je indikátor, který v reálném čase graficky zobrazuje hodnotu regulační odchylky při měření. Díky tomuto indikátoru je člověk, na kterém probíhá měření schopen kompenzovat regulační odchylku.

5.2 Formát a struktura dat

Data jsou uložena v MATLAB struktuře se dvěma prvky (Obrázek 12). Prvním prvkem je struktura transfer function (Obrázek 13), ve které je uložen číselný a jmenovatel přenosové funkce soustavy, na které probíhalo měření. Druhým prvkem je struktura out typu SimulationOutput, ve které se nachází jednotlivé objekty naměřených veličin datového typu timeseries. Tyto objekty obsahují časový vektor a vektor naměřených hodnot. Ukázku obsahu struktury out zobrazuje Obrázek 14. Ve struktuře se nachází 4 objekty reprezentující naměřené signály: vstupní signál, výstupní signál, regulační odchylku a akční veličinu dle schématu základního regulačního obvodu (Obrázek 11). Z těchto čtyř uložených signálů jsou pro sestavení modelu člověka jako regulátoru podstatná zejména naměřená data hodnot regulační odchylky a akční veličiny. Obrázek 15 zobrazuje ukázkou části naměřených hodnot. Struktura také obsahuje časový vektor měření tout. Dále jsou zde objekty SimulationMetaData a ErrorMessage. Tyto objekty jsou pro následující zpracování nepotřebné.



Obrázek 11 - Základní schéma regulačního obvodu

1x1 struct with 2 fields	
Field ^	Value
Fs	1x1 struct
out	1x1 SimulationOutput

Obrázek 12 – Struktura výstupních dat ze simulátoru

nactenaData.Fs	
Field ^	Value
num	1
den	[1,0]

Obrázek 13 - Obsah struktury Fs

nactenaData.out	
Property ^	Value
action	1x1 double timeseries
error	1x1 double timeseries
input	1x1 double timeseries
output	1x1 double timeseries
tout	276x1 double
SimulationMetadata	1x1 SimulationMetadata
ErrorMessage	"

Obrázek 14 – Obsah struktury out

nactenaData.out.error	
Time series name: <input type="text"/>	
Time	Data:1
0	0.5000
0.1000	0.5000
0.2000	0.5000
0.3000	0.5000
0.4000	0.5000
0.5000	0.5000
0.6000	0.5000
0.7000	0.5000
0.8000	0.5000
0.9000	0.4997
1	0.4992

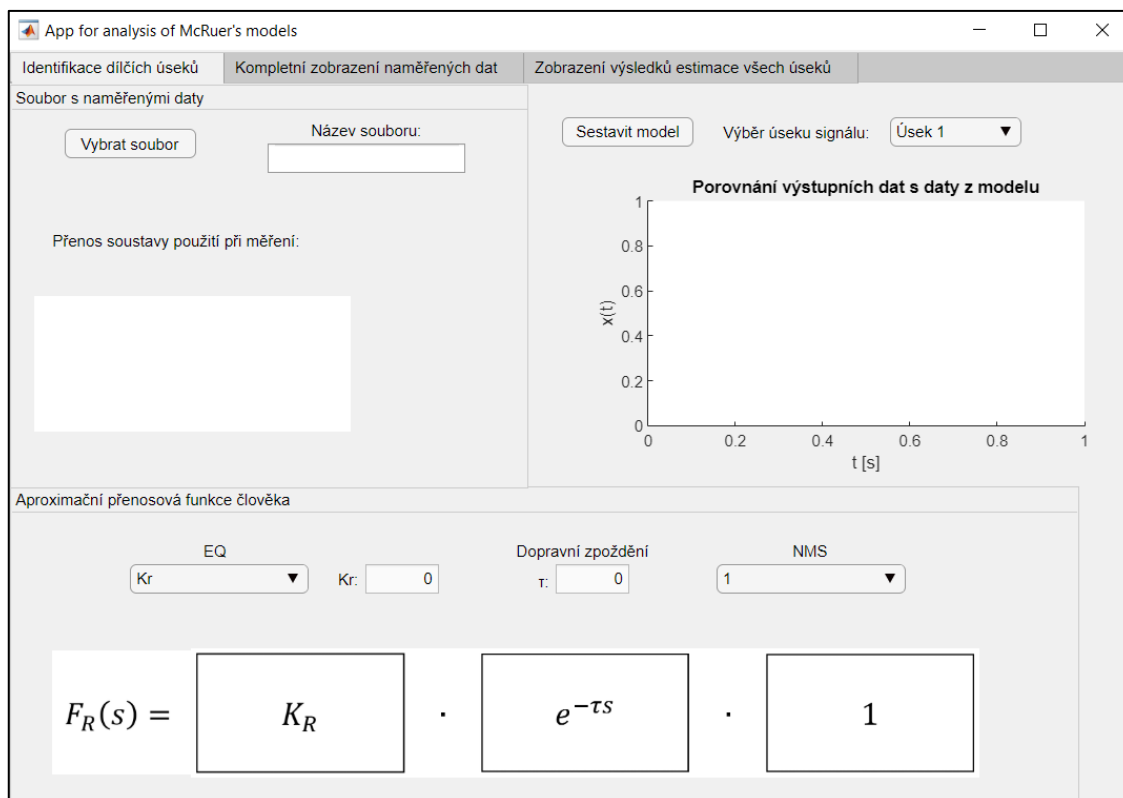
Obrázek 15 - Ukázka naměřených hodnot regulační odchylky

6. KONCEPT APLIKACE

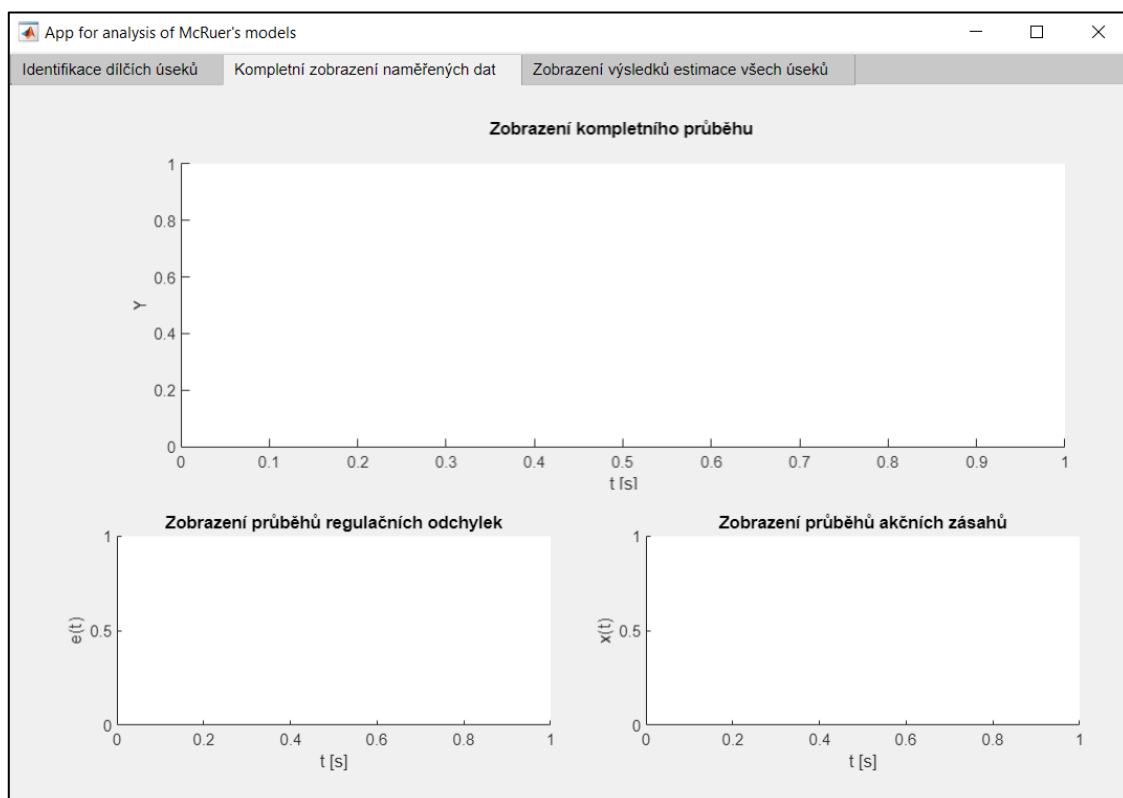
Hlavní požadovanou funkcí aplikace je sestavení modelu člověka nahrazující regulátor z dodaných naměřených dat na simulátoru popsaném v předchozí kapitole. Aplikace je vytvořena v prostředí MATLAB App designer. Aplikace je exportována jako instalační balíček MATLAB aplikace. Okno aplikace se přizpůsobuje velikosti obrazovky. Princip aplikace znázorňuje vývojový diagram Obrázek 19. Návrh vzhledu okna aplikace je zobrazen na následujících obrázcích: Obrázek 16 , Obrázek 17, Obrázek 18.

6.1 Popis okna aplikace

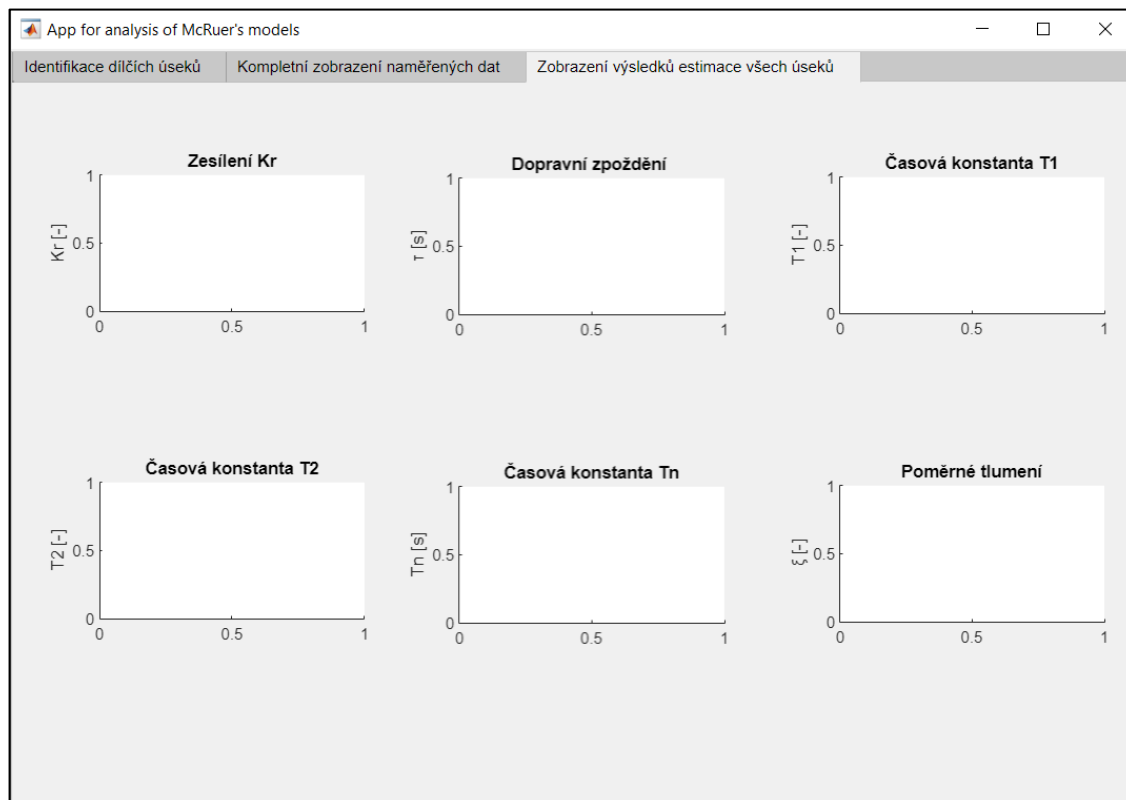
Okno aplikace je rozděleno celkem do tří záložek. Na obrázku 16 je zobrazena první záložka aplikace s názvem Identifikace dílčích úseků. Tato záložka je dále rozdělena na tři části. První část nacházející se v levé horní části okna aplikace je určena k výběru souboru s naměřenými daty. Soubor lze načíst pomocí tlačítka Vybrat soubor, které se nachází v levém horním rohu této části. Po výběru souboru a ověření jeho kompatibility se název souboru zobrazí v textovém poli vedle tohoto tlačítka. Pod tlačítkem Vybrat soubor se nachází kolonka, ve které se zobrazí tvar přenosové funkce soustavy použité při měření. Druhá část nacházející se v dolní části okna je určena k volbě tvaru aproximační přenosové funkce a k následnému zobrazení hodnot konstant této přenosové funkce. Nachází se zde obrázek znázorňující tvar zvolené aproximační přenosové funkce. Nad tímto obrázkem se nachází dvě rozbalovací nabídky. Jedna určená pro výběr modelu zahrnující ekvivalentní přenos člověka a druhá pro výběr přenosové funkce popisující neuromuskulární systému člověka. Vedle těchto rozbalovacích nabídek se podle zvolených tvarů těchto přenosových funkcí zobrazují numerická pole obsahující číselnou hodnotu jednotlivých konstant daných přenosových funkcí. Poslední část záložky Identifikace dílčích úseků nacházející se v pravé horní části je určena k vyhodnocení modelu. Nachází se zde tlačítko Sestavit model po jehož stisku dojde k sestavení modelů a následnému vykreslení průběhů naměřených a vypočtených dat do grafu nacházejícím se pod tímto tlačítkem. Vedle tlačítka se nachází rozbalovací nabídka určená k volbě úseku signálu, který má být vykreslen. Na obrázku 17 je zobrazena druhá záložka okna aplikace. Na této záložce se nachází celkem tři grafy. První graf nacházející se v horní části této záložky slouží k vykreslení kompletního průběhu naměřených dat. Pod tímto grafem jsou umístěny dva menší grafy. Do prvního grafu se vykreslují průběhy regulačních odchylek včetně jejich průměrného průběhu. Do druhého grafu se vykreslují průběhy akčních veličin také včetně jejich průměrného průměru. Poslední záložka okna aplikace nazvaná Zobrazení výsledků estimace všech úseků je zobrazena na obrázku 18. Obsahem této záložky je celkem 6 grafů do kterých se vykreslují krabicové diagramy zobrazující statistické vyhodnocení konstant modelů pro všechny úseky.



Obrázek 16 – Okno aplikace – záložka Identifikace dílčích úseků



Obrázek 17 – Okno aplikace – záložka Kompletní zobrazení naměřených dat



Obrázek 18 – Okno aplikace – záložka Zobrazení výsledků estimace všech úseků

6.2 Princip funkce aplikace

Principiální popis funkčnosti aplikace popisuje vývojový digram na obrázku 19. Podrobnější popis činnosti jednotlivých bloků vývojové digramu zahrnuje následující část textu.

Po spuštění aplikace se nastaví základní počáteční stav všech prvků grafického uživatelského rozhraní a vytvoří se globální proměnné podstatné pro další chod aplikace. V tomto okamžiku samotná aplikace nic neprovádí a čeká na vstup od uživatele. Uživatel má celkem čtyři možnosti vstupů, které může provést.

První možností je pomocí rozbalovacích nabídek zvolit požadovaný tvar aproximační přenosové funkce. Způsob volby aproximační přenosové funkce je blíže popsán v kapitole 7, jejímž obsahem je uživatelský návod aplikace. Změnou alespoň jedné části aproximační přenosové funkce dojde k přenastavení vnitřní proměnné, která uchovává informaci o této volbě. Dle zvolené aproximační přenosové funkce se také

nastaví vnitřní proměnná, která určuje příkaz System identification toolboxu pro sestavení modelu. Tyto příkazy jsou popsány v kapitole 3. Pokud je pro sestavení modelu použita funkce tfest nastaví se také proměnné určující počet pólů a počet nul pro estimaci. Pokud bude model sestavován pomocí příkazu procest nastaví si proměnná, která uchovává akronym určující parametry modelu. Tento uživatelský zásah z důvodu jeho jednoduchosti a zachování čitelnosti vývojové digramu není znázorněn ve vývojovém diagramu. Pro přehlednost aplikace je tato aproximační přenosová funkce znázorněna obrázkem, který se mění dle zvolené možnosti.

Další možností uživatele zásahu do aplikace je stisknutí tlačítka Vybrat soubor. Tím se začnou provádět kroky znázorněné na vývojovém diagramu. Nejprve se otevře dialogové okno pro výběr souboru. Zde má uživatel možnost vybrat soubor s naměřenými daty a to pouze ve formátu .mat. Pokud uživatel nevybere žádný soubor a zavře okno pro výběr souboru zobrazí se okno s upozorněním: „Nebyl vybrán žádný soubor“. V opačném případě kdy uživatel zvolí soubor s naměřenými daty ve formátu .mat aplikace načte soubor a dojde k vypsání názvu souboru do textového pole vedle tohoto tlačítka. Poté dojde ke kontrole obsahu údajů o soustavě v souboru. Pokud soubor neobsahuje údaje o soustavě použité při měření zobrazí okno s chybovou hláškou: „Soubor neobsahuje údaje o soustavě“. Absence tohoto údaje v souboru ovšem není omezující pro estimaci modelu, a proto není přerušen chod programu. Dalším důležitým krokem je otestování obsahu naměřených dat v souboru. Pokud soubor neobsahuje strukturu s naměřenými daty zobrazí se Chybová hláška: „Soubor neobsahuje naměřená data“. Absence naměřených dat už je pro další chod programu omezující, a proto dojde k ukončení chodu programu obsluhující tlačítko Vybrat soubor. V případě, že soubor strukturu s naměřenými daty obsahuje dojde k uložení na měřených do vlastních proměnných aplikace a následně k vykreslení naměřených dat do grafu na druhé kartě okna aplikace. V tomto okamžiku je aplikace opět ve stavu, kdy čeká na vstup od uživatele.

Třetí možností zásahu uživatele do aplikace je stisk tlačítka Sestavit model. Po stisku tohoto tlačítka dochází ke kontrole, zda byl načten soubor. Pokud nedošlo k načtení souboru zobrazí se chybová hláška: „Nebyl načten soubor“ a ukončí se chod programu obsluhující tlačítko Sestavit model. V případě, kdy soubor načten byl dojde k otestování obsahu dat ve vnitřních proměnných všech potřebných signálů. Pokud soubor neobsahuje naměřená data, nebo jsou ve špatné struktuře dojde k výpisu chybové hlášky: „Naměřená data neobsahují všechny podstatné signály“. V případě, kdy soubor obsahuje naměřená ve správné struktuře program pokračuje.

Před sestavením modelu dochází k předzpracování naměřených dat následujícím způsobem. Nejdříve program projde hodnoty signálu žádané hodnoty a najde v jakých časech dochází ke skokovým změnám. Časy skokových změn požadované hodnoty jsou uloženy do vektoru, přičemž první hodnota je z tohoto vektoru odstraněna. Na základě těchto časů se následně naměřené hodnoty regulačních odchylek a akčních zásahů

rozdělí na jednotlivé úseky, které jsou uloženy do matic. Tyto úseky jsou následně zkráceny podle délky nejkratšího úseku, tím je zajištěna estimace modelu pouze na dynamické části průběhu všech úseků signálů. Díky odstranění prvního času skokové změny se do těchto matic neuloží první úsek signálu. Tento úsek je nevhodný zahrnovat do vyhodnocení modelů, protože by mohl nejvíce zkreslovat celkové výsledky. Naměřené hodnoty signálu reprezentující regulační odchylku je nutné posunout tak, aby tento signál končil v nulové hodnotě. Offset pro hodnoty regulační odchylky je tedy určen poslední hodnotou daného úseku. Naměřené hodnoty signálu reprezentující akční zásah je pro správnou estimaci modelu nutné posunout tak, aby každý úsek signálu začínal v nule. Offset pro tento signál je tedy určen první hodnotou jednotlivých úseků. Pro signály žádané hodnoty, které dosahují záporných amplitud jsou odpovídající úseky signálů regulačních odchylek a akčních zásahů převráceny. Následně jsou všechny úseky regulačních odchylek a akčních zásahů vykresleny do grafů na druhé kartě okna aplikace. Do těchto grafů je následně vykreslena průměrná hodnota všech průběhů odpovídajících signálů.

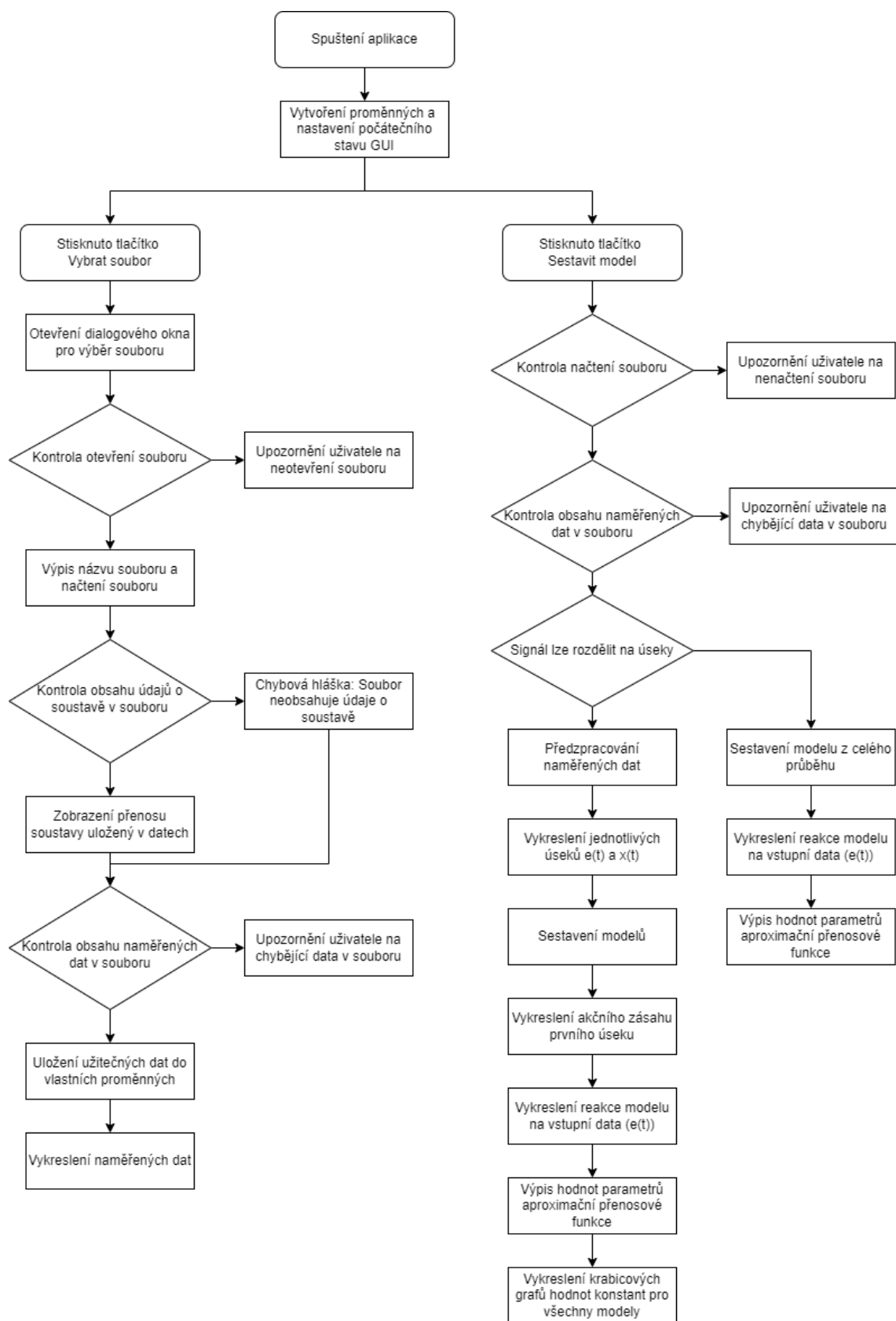
Po nutném předzpracování dat dojde k samotnému sestavení modelů aproximačních přenosových funkcí člověka. Sestavení modelů probíhá dle předchozího nastavení aproximační přenosové funkce. Podle volby této přenosové funkce je vybrán příkaz, kterým bude estimace prováděna a nastaveny parametry daného příkazu. Sestavení modelů probíhá v cyklu vždy pro každý úsek signálu následujícím způsobem. Nejprve se vytvoří objekt `iddata` z odpovídajících úseků signálů, a to regulační odchylky jako vstupní signál a akčního zásahu jako výstupní signál. Poté už dochází k samotné estimaci modelu, modely všech úseků jsou uloženy do pole typu `cell`, které obsahuje objekty typu `idtf`, případně `idproc`. Po estimaci modelu dojde pomocí příkazu `lsim` k vytvoření signálu reprezentující časovou odezvu na odpovídající úsek průběhu regulační odchylky. Tento signál je následně uložen do matice. Posledním krokem při sestavování modelů jednotlivých úseků je pomocí funkce `compare` získat procentuální porovnání shodnosti signálu reprezentující časovou odezvu daného modelu s odpovídajícím naměřeným akčním zásahem.

Po získání modelů se vytvoří matice pro uložení hodnot konstant aproximační přenosové funkce. Následně se do této matice zapíše hodnoty a na první kartě aplikace se zobrazí hodnoty konstanty předem zvolené aproximační přenosové funkce pro první úsek signálu. První úsek signálu ve skutečnosti odpovídá druhému úseku naměřených dat žádané hodnoty, protože první úsek byl odstraněn z důvodu uvedeného výše v odstavci o předzpracování dat. Na první kartě aplikace se do grafu zobrazí signál reprezentující časovou odezvu modelu odpovídající prvnímu úseku signálu regulační odchylky. Dále se do tohoto grafu vykreslí naměřený průběh akčního zásahu prvního úseku a do legendy grafu se vypíše hodnota odpovídající procentuální shodnosti těchto průběhů. Následně se nad tímto grafem naplní hodnoty rozbalovací nabídky s názvem Výběr úseku signálu. Hodnoty této rozbalovací nabídky odpovídají počtu jednotlivých

úseků naměřených signálů. Posledním krokem programu obsluhující tlačítko Sestavit model je vykreslení krabicových grafů na třetí kartu okna aplikace. Tento typ grafů byl zvolen z důvodu nejvhodnější reprezentace výsledků estimace pro porovnání všech úseků signálů.

Pokud průběh signálu žádané hodnoty neobsahuje skokové změny a nelze ho tedy rozdělit na jednotlivé úseky dojde pouze k odstranění prvních 5 sekund průběhů a sestavení modelu z celých průběhů signálů. Poté se pouze vykreslí průběh akční veličiny a časová odezva modelu na regulační odchylku do grafu na kartě Identifikace dílčích úseků. Následně se na této záložce vypíší hodnoty konstant aproximační přenosové funkce a dojde k ukončení chodu programu obsluhující tlačítko Sestavit model.

Poslední možností vstupu uživatele je volba úseku signálu pomocí rozbalovací nabídky umístěné nad grafem na první kartě okna aplikace. Změnou úseku signálu pomocí této rozbalovací nabídky se ve grafu pod touto rozbalovací nabídkou zobrazí průběh akčního zásahu a signál reprezentující časovou odezvu modelu odpovídající zvolenému úseku signálu. Také se změní zobrazení hodnot konstant aproximační přenosové funkce na hodnoty, které odpovídají vybranému úseku.



Obrázek 19 - Vývojový diagram aplikace

7. UŽIVATELSKÝ MANUÁL APLIKACE

Tato kapitola popisuje pracovní postup použití aplikace, která je výsledkem této bakalářské práce. Aplikace slouží k velmi specifickému účelu, a to k vyhodnocení a sestavování McRuerových modelů z dat naměřených na simulátoru, bez kterého není možné tato výzkumná měření provádět. Aplikace je tedy určena pouze pro odborné uživatele s přístupem k naměřeným datům ze zmiňovaného simulátoru.

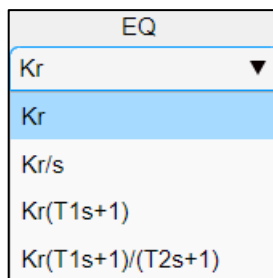
7.1 SPUŠTENÍ APLIKACE

Aplikace je distribuována pouze jako MATLAB App. To znamená, že pro její užívání je nutné mít nainstalovaný MATLAB. Pro instalaci aplikace stačí pouze spustit instalační soubor. Po správně provedené instalaci aplikace se v MATLABU na kartě APPS objeví ikona s názvem aplikace pro analýzu McRuerových modelů. Aplikaci lze spustit kliknutím na tuto ikonu.

7.2 PRACOVNÍ POSTUP IDENTIFIKACE MODELU

Po spuštění aplikace se zobrazí okno s otevřenou první záložkou nazvanou Identifikace dílčích úseků. Prvním krokem pro sestavení modelu je otevřít soubor s naměřenými daty. K tomu slouží tlačítko otevřít soubor. Po kliknutí na toto tlačítko se otevře dialogové okno pro výběr souboru. Soubor musí mít příponu .mat, jiné typy souborů nejsou pro výběr povoleny. Pokud jste vybrali soubor správného formátu, se správnou strukturou naměřených dat zobrazí se název vybraného souboru. Pod tlačítkem pro výběr souboru se nyní také zobrazí tvar přenosové funkce soustavy použité při měření. Tvar přenosové funkce soustavy se zobrazí pouze pokud je tato informace uložena v souboru.

Dalším krokem je zvolit si požadovaný tvar aproximační přenosové funkce člověka. K tomu slouží spodní část záložky Identifikace dílčích úseků. Ukázkou této záložky zobrazuje Obrázek 16. Tato aproximační přenosová funkce nazvaná $F_R(s)$ je rozdělena na tři části. První část je Ekvivalentní přenos člověka označený EQ. Tvar jeho přenosové funkce je možné zvolit pomocí rozbalovací nabídky. Možné tvary této přenosové funkce zobrazuje Obrázek 20. Pro dosažení nejlepších výsledků estimace modelu je vhodné volit tvar této přenosové funkce s ohledem na použitou soustavu při měření dat dle McRuerovy teorie. Příklady tvarů ekvivalentních přenosů člověka s ohledem na platnost teorie Crossover law jsou uvádí Tabulka 1.

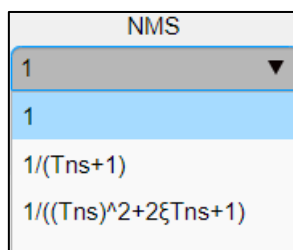


Obrázek 20 - Ukázka výběrového seznamu EQ

Na obrázku 20 si lze povšimnout, že zde chybí aproximační přenosová funkce ve tvaru Ks , která představuje „čistou derivaci“. Tento tvar by byl dle tabulky 1 vhodný použít pro soustavu dvojitého integrátoru. System identification toolbox bohužel estimaci s čistou derivací neumožňuje, a proto je nutné aproximační přenosovou funkci volit ve tvaru $K_R(T_1s + 1)$.

Druhá část aproximační přenosové funkce $F_R(s)$ je dopravní zpoždění $e^{-\tau s}$. Zde není nutné nic nastavovat. Dopravní zpoždění bude totiž v modelu obsaženo vždy.

Poslední část je rozšíření modelu o neuromuskulární systém člověka. Toto rozšíření vychází z Precision modelu. Přenosovou funkci popisující neuromuskulární systém člověka lze do aproximační přenosové funkce zahrnout zvolením jednoho z možných tvarů pomocí rozbalovací nabídky NMS. Ukázku možností zahrnující tato nabídka je znázorňuje Obrázek 21.



Obrázek 21 - Ukázka výběrového seznamu NMS

Po načtení dat a zvolení vhodné aproximační přenosové funkce je možné kliknout na tlačítko sestavit model v pravé horní části okna Identifikace dílčích úseků. Po stisku tohoto tlačítka se objeví okno znázorňující stav průběhu estimace modelů. Nyní dochází na pozadí aplikace k předzpracování načtených naměřených dat a následně k sestavování modelů za pomoci funkcí z rozšiřujícího modulu System Identification Toolbox. Tento proces může nějakou dobu trvat, přibližný čas dokončení estimace je díky ukazateli stavu průběhu estimace alespoň částečně možné odhadnout.

Po dokončení estimace modelů se v pravé části okna záložky Identifikace dílčích úseků do grafu vykreslí průběh naměřeného výstupního signálu (akčního zásahu) a průběh signálu odpovídajícího časové odezvě prvního sestaveného modelu na vstupní signál (regulační odchylku) odpovídajícího úseku. V legendě tohoto grafu se zobrazí procentuální vyjádření shodnosti těchto dvou průběhů. Do části aproximační přenosová funkce se nyní zobrazí hodnoty konstant předdefinované přenosové funkce. Nad grafem s vykresleným průběhem časové odezvy modelu prvního úseku se nachází rozbalovací nabídka Výběr úseku signálu. Pomocí této rozbalovací nabídky je možné změnit úsek signálu. Změnou úseku signálu dojde k vykreslení odpovídajícím průběhům do grafu pod touto výběrovou nabídkou a ke změně hodnot konstant aproximační přenosové funkce dle odpovídajícího úseku.

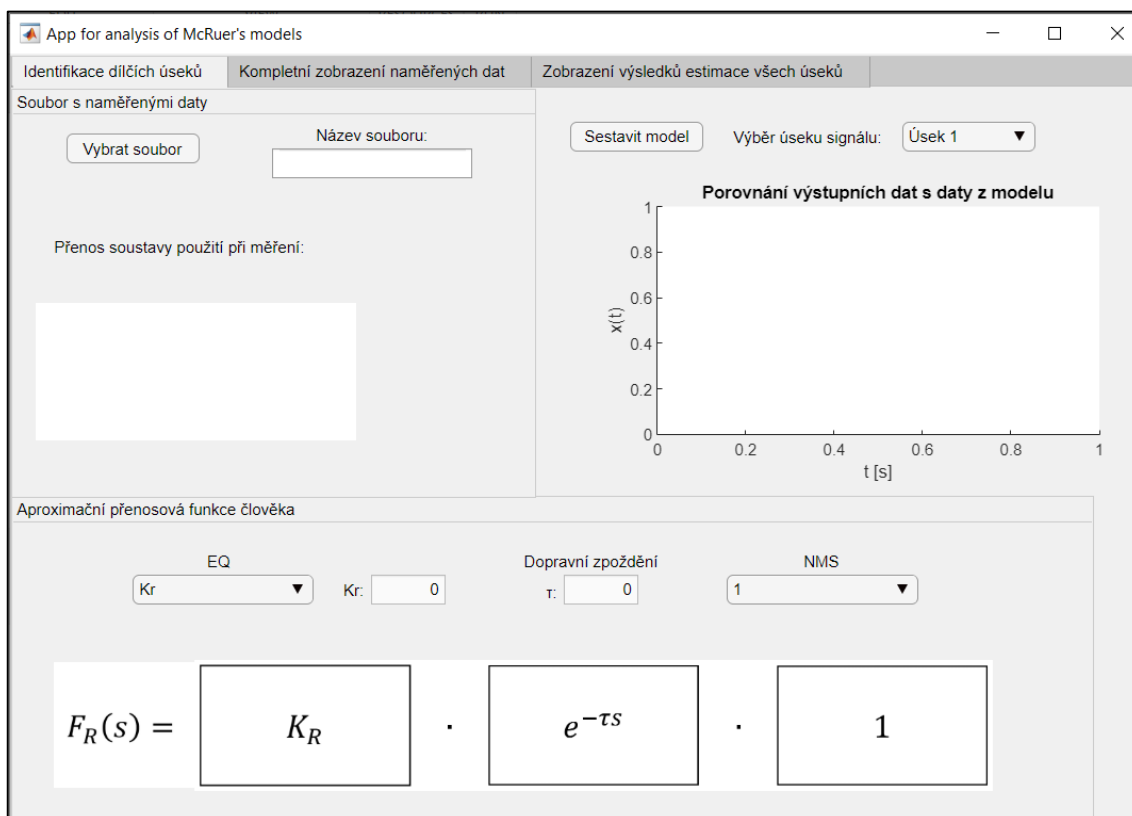
Přepnutím na kartu Kompletní zobrazení naměřených dat se lze podívat na celkový průběh naměřených dat žádané hodnoty, regulační odchylky a akčního zásahu. Dále jsou na této kartě další dva grafy, které zobrazují průběhy regulačních odchylek a akčních zásahů pro jednotlivé úseky. Pro signály, které mají úseky se zápornými hodnotami průběhů jsou tyto úseky invertovány tak, aby byli pouze v kladných hodnotách. V těchto dvou grafech je dále zobrazena červenou tučnou čarou jejich průměrná hodnota.

Na kartě Zobrazení výsledků estimace všech úseků se nachází šest oken grafů odpovídající jednotlivým konstantám aproximačních přenosových funkcí. V těchto oknech jsou pro každou konstantu vykresleny krabicové grafy. Z těchto grafů lze vyčíst medián hodnot, první a třetí kvartil (25. a 75. percentil). Maxima a minima hodnot jsou zobrazena křížkem.

8. UKÁZKA FUNKCE APLIKACE

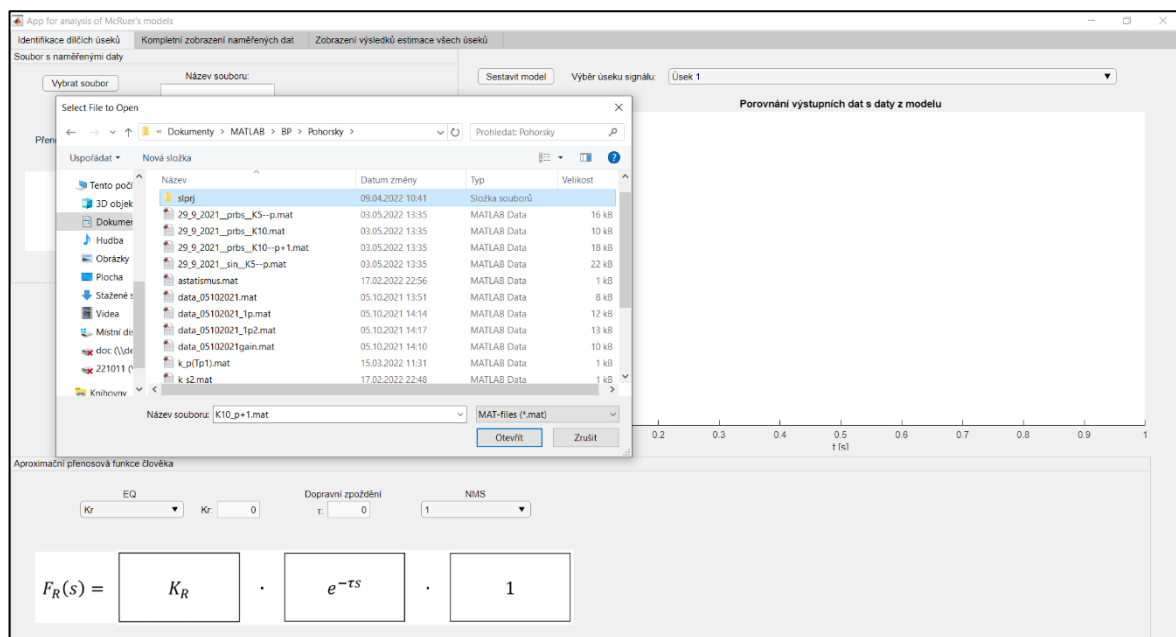
Tato kapitola slouží jako ukázka dosaženého výsledku této práce. V této kapitole je pomocí printscreenů předvedena funkčnost aplikace na dodaných naměřených datech.

Po spuštění aplikace se zobrazí okno, které znázorňuje následující Obrázek 22. Toto okno je možné roztáhnout na celou obrazovku. Zvětšené okno ukazuje Obrázek 23.



Obrázek 22 – Ukázka funkce 1

Nyní je možné stiskem tlačítka Vybrat soubor načíst soubor s naměřenými daty. Po stisku tlačítka se otevře výběrové okno, kde zvolíme požadovaný soubor s naměřenými daty. Tento krok ukazuje Obrázek 23.



Obrázek 23 - Ukázka funkce 2

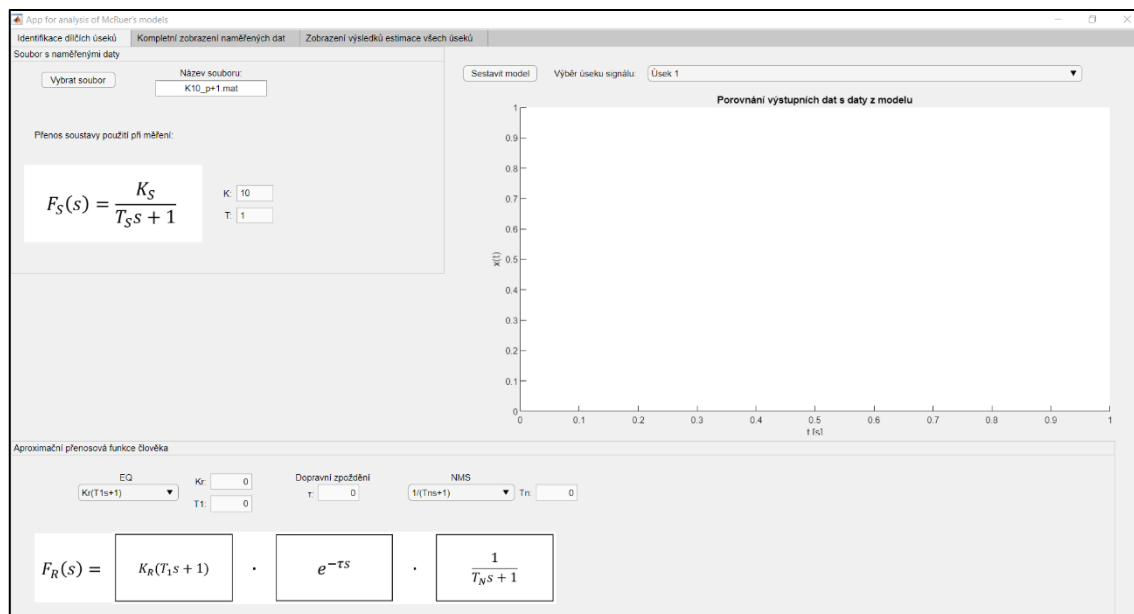
Po výběru souboru se zobrazí přenosová funkce soustavy včetně číselných hodnot konstant této přenosové funkce. V tomto okamžiku je potřeba zvolit aproximační přenosovou funkci. Pro soustavu v této ukázce je vhodná aproximační přenosová funkce ve tvaru:

$$F_R(s) = \frac{K_R(T_1s + 1)}{T_Ns + 1} e^{-\tau s}$$

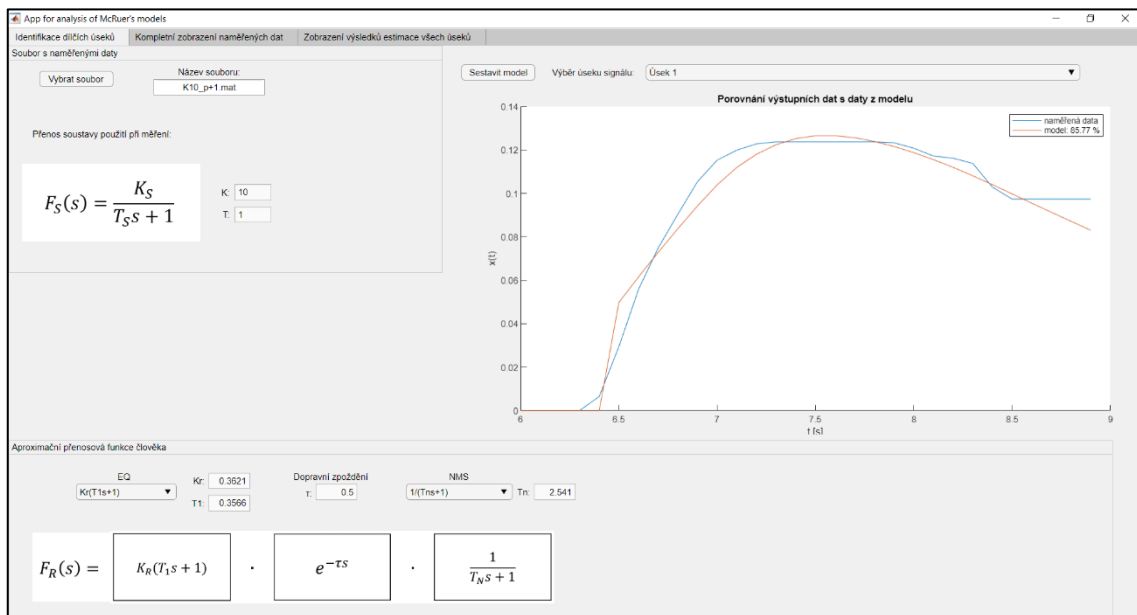
Nastavení této přenosové funkce provedeme pomocí rozbalovací nabídky EQ. Jelikož není možné estimovat model s menším počtem pólů, než je počet nul dojde k automatickému přenastavení přenosu NMS. Provedení tohoto nastavení ukazuje Obrázek 24. Po nastavení aproximační přenosové funkce je možné sestavit modely kliknutím na tlačítko Sestavit model. V tomto okamžiku dojde k předzpracování dat způsobem popsáním v kapitole 6. Následně se sestaví modely pro jednotlivé úseky signálu. Nyní se vykreslí průběh signálu v reakci prvního sestaveného modelu na vstupní signál (regulační odchylku) odpovídajícího úseku a zobrazí se hodnoty konstant aproximační přenosové funkce. Ukázku zobrazuje Obrázek 25.

Po přepnutí na kartu Kompletní zobrazení naměřených dat lze vidět vykreslená naměřená data a průběhy regulační odchylky a akčních zásahů, včetně jejich průměrných průběhů. Ukázku této karty zobrazuje Obrázek 26.

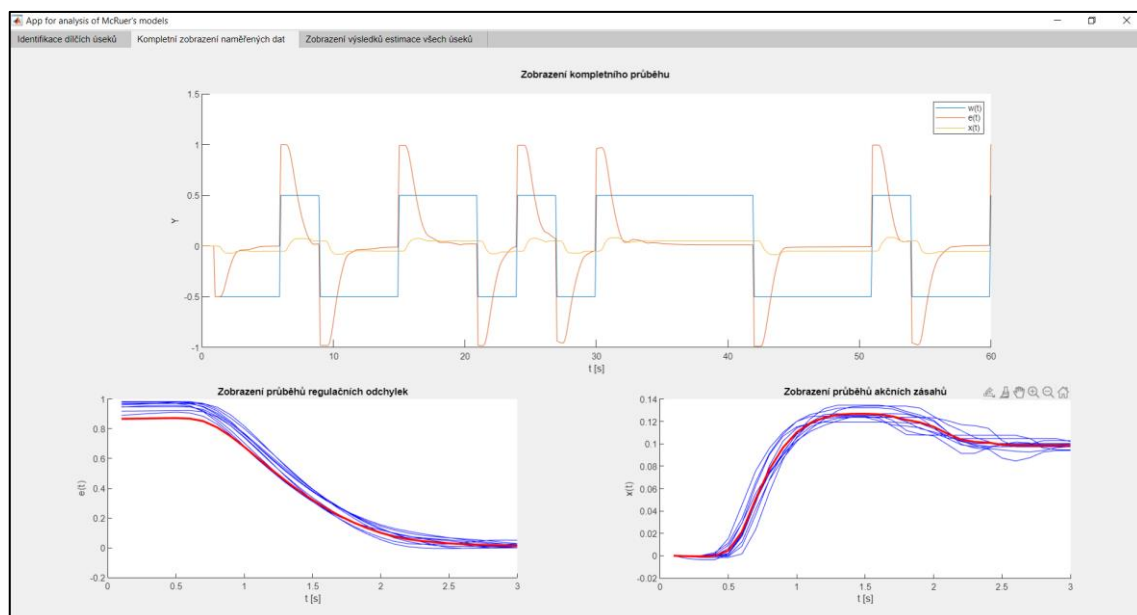
Na kartě Zobrazení výsledků estimace všech úseků jsou nyní zobrazeny krabicové grafy odpovídající jednotlivým konstantám aproximační přenosové funkce. Ukázkou této karty zobrazuje Obrázek 27.



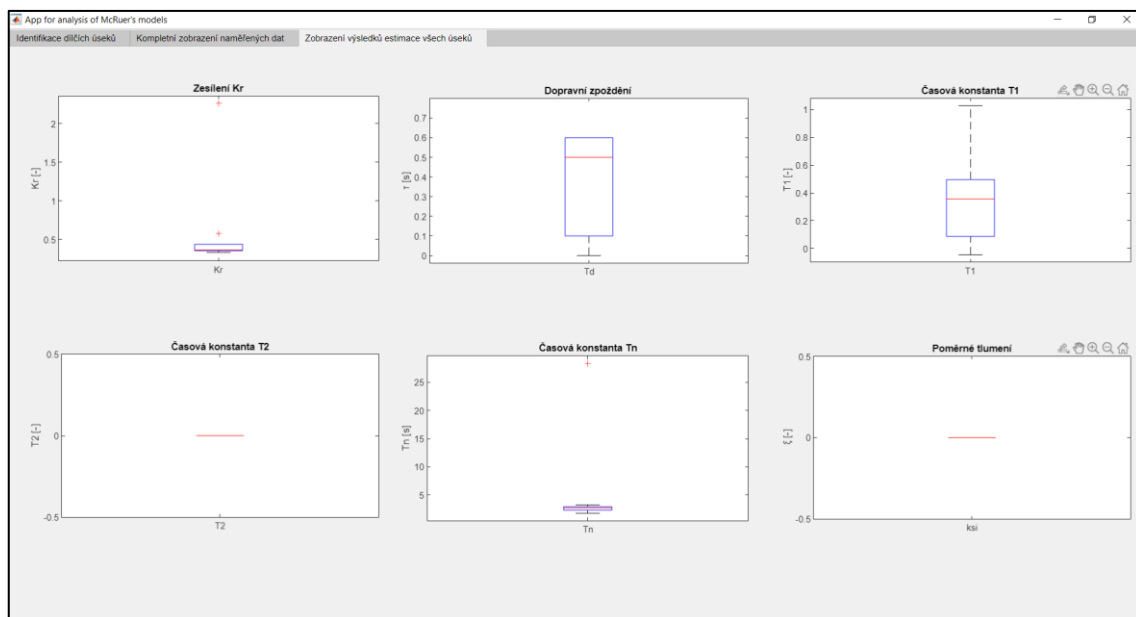
Obrázek 24 - Ukázka funkce 3



Obrázek 25 - Ukázka funkce 4



Obrázek 26 - Ukázka funkce 5



Obrázek 27 - Ukázka funkce 6

9. ZÁVĚR

Výsledkem této bakalářské práce je plně funkční aplikace pro automatické sestavení a následnou analýzu McRuerových modelů. Tato aplikace je distribuována jako instalační soubor aplikace MATLAB a je určena pro úzký okruh vědeckých pracovníků zabývajících se problematikou chování člověka jako regulátoru. Pro vytvoření této aplikace bylo nutné nejdříve pochopit problematiku sestavování modelů člověka jako regulátoru, a to zejména McRuerovu teorii Crossover law a modely člověka vycházející z této teorie. Vysvětlení a popis této problematiky je obsahem druhé kapitoly této práce. Po nastudování této teorie bylo zapotřebí seznámit se s algoritmy za pomoci kterých lze tyto modely sestavovat a naučit se je používat. K tomuto účelu je určen rozšiřující balíček Matlabu nazvaný System Identification Toolbox. Popis tohoto toolboxu a jeho příkazů podstatných pro sestavování modelů člověka obsahuje kapitola 3. Dalším krokem bylo seznámení se vývojovým prostředím MATLAB App Designer, které bylo zvoleno jako nejvhodnější zejména kvůli kompatibilitě se System Identification Toolbox. Vývojové prostředí MATLAB APP designer je popsáno ve čtvrté kapitole této práce. Tato kapitola dále vysvětluje pracovní postup vývoje aplikací v tomto vývojovém prostředí a také možné způsoby distribuce aplikací. Posledním krokem před samotným návrhem aplikace bylo seznámení se s dodanými naměřenými daty. Pro lepší pochopení struktury těchto dat proběhlo nejdříve seznámení se se simulátorem, na kterém jsou tato data získávána. Tento simulátor a struktura na něm získaných dat jsou popsány v kapitole 5. Po nastudování požadované teorie, seznámení se s nástroji potřebnými k vývoji a neměřenými daty pro sestavování modelů došlo k samotnému návrhu aplikace. Nejdříve byl navržen vzhled aplikace a umístění jednotlivých komponentů podstatných pro analýzu modelů člověka jako regulátoru. Po návrhu grafického uživatelského rozhraní byl vytvořen vývojový digram aplikace, který znázorňuje jednotlivé kroky vedoucí k sestavení modelu. Na základě tohoto vývojového digramu byl vytvořen program obstarávající chod aplikace. Průběh vývoje aplikace zaznamenává kapitola 6. Po vytvoření a odladění byla aplikace otestována na sadě dodaných naměřených dat. Ukázka funkčnosti aplikace na jedné z těchto dat je zobrazena v kapitole 8. Po úspěšném otestování aplikace byl sepsán uživatelský manuál aplikace, který je možné nalézt v kapitole 7.

LITERATURA

- [1] MCRUER, D.T. and Ezra KRENDEL. Human Pilot Dynamics in Compensatory systems [online]. USA: Air Force Flight Dynamics Laboratory, Research and Technology Division, Air Force Systems Command, United States Air Force, 1965. [cit. 2021-11-08] Dostupné z: <http://contrails.iit.edu/DigitalCollection/1965/AFFDLTR65-015.pdf>
- [2] HAVLÍKOVÁ, Marie. Diagnostika systému s lidským operátorem. Brno, 2008. Dizertační práce. Vysoké učení technické v Brně.
- [3] System Identification Toolbox. *MathWorks* [online]. Natic: MathWorks, c1994-2021 [cit. 2021-12-03]. Dostupné z: <https://www.mathworks.com/products/sysid.html>
- [4] System Identification Toolbox Getting Started Guide. MathWorks [online]. Natic: MathWorks, c1994-2021 [cit. 2021-11-25]. Dostupné z: https://www.mathworks.com/help/pdf_doc/ident/ident_gs.pdf
- [5] JIRGL, Miroslav. Analýza modelů chování pilota při řízení letounu [online]. Brno, 2016 [cit. 2021-12-01]. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=131942. Disertační práce. Vysoké Učení Technické v Brně.
- [6] BLAHA, Petr. Řízení a regulace I. Brno: Vysoké učení technické, 2009.
- [7] Matlab Documentation. *MathWorks* [online]. Natic: MathWorks, c1994-2021 [cit. 2021-12-28]. Dostupné z: <https://www.mathworks.com/help/>
- [8] Develop Apps Using App Designer. *MathWorks* [online]. Natic: MathWorks, c1994-2021 [cit. 2021-12-20]. Dostupné z: <https://www.mathworks.com/help/matlab/app-designer.html>

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

FEKT	Fakulta elektrotechniky a komunikačních technologií
VUT	Vysoké učení technické v Brně
MMS	Man Machine System
MIMO	Multi Input Multi Output
SISO	Single Input Single Output
GUI	Graphic User Interface
App	Application

Symboly:

s	Laplaceův operátor	(-)
$F_0(s)$	přenos otevřené regulační smyčky	(-)
$F_R(s)$	přenos regulátoru	(-)
$F_S(s)$	přenos regulované soustavy	(-)
ω_c	frekvence řezu	(rad/s)
τ	dopravní (reakční) zpoždění	(s)
K_r –	zesílení	(-)
T_L –	derivační (prediktivní) časová konstanta	(s)
T_I –	integrační (setrvačná) časová konstanta	(s)
T_N –	neuromuskulární časová konstanta	(s)
ω_N –	frekvence kmitů neuromuskulárního systému	(rad/s)
ξ_N –	tlumení neuromuskulárního systému	(-)
K_p –	statické zesílení	(-)
T_{pk} –	časové konstanty pólů	(s)
T_z –	časová konstanta nuly	(s)
T_d –	dopravní zpoždění	(s)

SEZNAM PŘÍLOH

PŘÍLOHA A – Zdrojový kód aplikace je uložen na přiloženém CD

PŘÍLOHA B – Instalační soubor aplikace je uložen na přiloženém CD