

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE



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

SADA TESTOVACÍCH ÚLOH PRO MĚŘENÍ ODEZVY LIDSKÉHO OPERÁTORA

A SET OF THE TESTING TASKS TO MEASURE HUMAN OPERATOR RESPONSES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Tomáš Pátek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Miroslav Jirgl, Ph.D.

BRNO 2020



Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Tomáš Pátek

ID: 186156

Ročník: 2

Akademický rok: 2019/20

NÁZEV TÉMATU:

Sada testovacích úloh pro měření odezvy lidského operátora

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je návrh a implementace sady testovacích úloh pro účely měření odezvy lidského operátora při řízení základních dynamických systémů a vyhodnocení jeho vybraných vlastností a omezení.

1. Prostudujte problematiku popisu lidského operátora a používaných modelů chování.
2. Navrhněte koncepci systému zahrnující HMI, resp. HCI.
3. Navrhněte a popište testovací scénáře zahrnující různé vstupní signály a různé dynamické systémy.
4. Implementujte navržený systém v prostředí MATLAB/Simulink a otestujte.
5. Proveďte základní měření odezvy lidského operátora na vybrané vstupní signály dle navržených scénářů.
6. Diskutujte získané výsledky a možnosti jejich využití.

DOPORUČENÁ LITERATURA:

HAVLÍKOVÁ, Marie. Diagnostika systémů s lidským operátorem. Brno, 2008. Dizertační práce. Vysoké učení technické v Brně.

Termín zadání: 3.2.2020

Termín odevzdání: 1.6.2020

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

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové 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

Práce se zabývá návrhem simulátoru pro měření odezvy lidského operátora v programu MATLAB a nadstavbě Simulink. V první části je shrnuta teorie matematických modelů lidského operátora, na kterou navazuje popis realizovaného simulátoru pro měření odezvy s grafickým uživatelským rozhraním. V poslední části je provedeno základní měření odezvy lidského operátora na různé scénáře, porovnání kvality řízení pro různé operátory a identifikace modelu pro jednotlivé operátory.

KLÍČOVÁ SLOVA

Lidský operátor, měření odezvy, matematický model, MATLAB, simulátor.

ABSTRACT

This master's thesis deals with the design of a simulator for measuring the response of a human operator in the MATLAB program and the Simulink extension. The first part summarizes the theory of mathematical models of human operator, which is followed by a description of a simulator for response measurement with a graphical user interface. In the last part, the basic measurement of human operator response to different scenarios, comparison of quality of control for different operators and identification of model for individual operators are performed.

KEYWORDS

Human operator, response measurement, mathematical model, MATLAB, simulator.

PÁTEK, Tomáš. *Sada testovacích úloh pro měření odezvy lidského operátora*. Brno, 2020, 97 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce: Ing. Miroslav Jirgl, Ph.D

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Sada testovacích úloh pro měření odezvy lidského operátora“ jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno 26.5.2020

.....
podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Miroslavu Jirglovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno 26.5.2020

.....

podpis autora

Obsah

Úvod	12
1 Lidský operátor	13
1.1 Způsoby řízení lidského operátora	13
1.1.1 Přímé řízení	13
1.1.2 Kompenzační řízení	14
1.1.3 Výhledové řízení	14
1.1.4 Prediktivní řízení	15
1.1.5 Architektura modelu lidského operátora	15
1.2 Matematické modely lidského operátora při zpětnovazebním řízení	15
1.2.1 Crossover model	16
1.2.2 Gross model	18
1.2.3 Precision model	18
1.2.4 Využití fuzzy systému	18
1.3 Metody vyhodnocení kvality regulace	20
1.3.1 Kritéria podle tvaru přechodové charakteristiky regulované veličiny - časová oblast	20
1.3.2 Kritéria kvality regulace ve frekvenční oblasti	24
1.4 identifikace systému	26
1.4.1 Identifikace z přechodové charakteristiky	26
1.4.2 Identifikace systému s využitím programu MATLAB	27
2 Návrh simulátoru pro měření odezvy lidského operátora	29
2.1 Žádaná hodnota w	30
2.2 Porucha v	30
2.3 Regulátor R	30
2.4 Soustava S	30
2.5 Rozhraní pro styk s měřenou osobou	31
3 Realizace simulátoru	32
3.1 Vlastnosti simulátoru	32
3.2 Obsluha simulátoru	33
3.2.1 Načíst systém	33
3.2.2 Vytvořit nový systém	33
3.2.3 Konfigurace zobrazení	37
3.2.4 Nastavit vstupní zařízení	38
3.2.5 Info	39

3.2.6	Struktura uložení konfiguračních proměnných a výchozí hodnoty	39
3.3	Struktura podadresářů simulátoru	40
3.4	Popis MATLAB skriptů	41
3.4.1	Skripty pro nastavení výchozích hodnot	41
3.4.2	Grafické uživatelské rozhraní (GUI)	43
3.4.3	Struc toolbox	46
3.5	Popis Simulink modelu	49
3.5.1	Univerzální část	49
3.5.2	Subsystemy A a B	49
3.5.3	Virtuální světy	50
4	Základní měření a vyhodnocení	58
4.1	Popis měření	58
4.1.1	Testovací scénáře	58
4.1.2	Vstupní zařízení	60
4.2	Zpracování dat	60
4.2.1	Vyhodnocení kvality regulace	61
4.3	Identifikace parametrů modelu lidského operátora	68
4.4	Závěr z měření	73
	Závěr	76
	Literatura	77
	Seznam příloh	79
	A Detailní struktura uložení konfiguračních proměnných a výchozí hodnoty	80
	B Detailní popis adresářové struktury	87
	C Detailní simulink schémata	90

Seznam obrázků

1.1	Zpětnovazební regulační obvod člověka	13
1.2	Přímé řízení	14
1.3	Kompenzační řízení	14
1.4	Výhledové řízení [8]	14
1.5	Prediktivní řízení [8]	15
1.6	Architektura modelu lidského operátora [8]	16
1.7	Uzavřená smyčka s blokem crossover modelu	18
1.8	Přechodová charakteristika regulačního obvodu	20
1.9	Porovnání regulační odchylky a akčního zásahu regulátoru	23
1.10	Amplitudová a fázová frekvenční charakteristika regulačního obvodu	24
1.11	Přechodová charakteristika systému 1. řádu s dopravním zpožděním	26
1.12	Okno System Identification Tool	27
2.1	Blokové schéma simulátoru	29
3.1	Hlavní menu	33
3.2	Konfigurace systému	34
3.3	Konfigurace generátoru žádané hodnoty	35
3.4	Konfigurace regulátoru	36
3.5	Konfigurace soustavy	37
3.6	Konfigurace zobrazení	38
3.7	Konfigurace vstupního zařízení	39
3.8	Zjednodušené schéma subsystému A	50
3.9	3D World Editor	51
3.10	Virtual World 1	51
3.11	Přepočítání polohy kuličky na úhel jejího natočení	52
3.12	Výpočet matice rotace	52
3.13	Výpočet matice rotace kolem osy Y	53
3.14	Blok VW1	53
3.15	Virtual World 2	54
3.16	Blok VW2	55
3.17	Virtual World 3	56
3.18	Přepočítání úhlu a rychlosti na polohu xy	57
4.1	Příklad žádané hodnoty náhodných pulzů	60
4.2	Žádaná hodnota superpozice sinusoid	61
4.3	Volant Genius SpeedWheel 3 MT [12]	62
4.4	Užitečná část signálu, scénář 1	63
4.5	Žádaná hodnota a řízená veličina ($0 = 7, S = 9$)	69
4.6	Scénář 8, Operátor 1, $F_S = \frac{1}{p^2}$	71

4.7	Akční zásahy operátora 8 při řízení systému $F_S = \frac{1}{p^2}$	72
4.8	Derivace akčních zásahů operátora 8 při řízení systému $F_S = \frac{1}{p^2}$. . .	73
4.9	Scénář 8, Operátor 5, $F_S = \frac{1}{p^2}$	74
4.10	Scénář 8, Operátor 2, $F_S = \frac{1}{p^2}$	75
C.1	To Workspace	90
C.2	To Workspace	91
C.3	Vstup	92
C.4	Subsystem A	93
C.5	Generátor žádané hodnoty	94
C.6	Regulátor	94
C.7	Omezení regulátoru	95
C.8	Soustava	95
C.9	Blok VW3	96
C.10	Blok VW3	97

Seznam tabulek

1.1	Přenosové funkce soustav a příklady jejich uplatnění v praxi [2] . . .	17
1.2	Přenosové funkce systému a přibližné adaptované přenosové funkce lidského operátora, $b \ll \omega_c$ [2]	17
1.3	Matematické modely lidského operátora	19
4.1	Odpovědi testovaných operátorů na položené otázky	58
4.2	Testovací scénáře	59
4.3	Průměrný překmit v % žádané hodnoty rozptyl	61
4.4	Integrál absolutní chyby (IAE)	64
4.5	Integrál čtverce chyby (ISE)	64
4.6	ITAE kritérium	65
4.7	ITSE kritérium	65
4.8	Porovnání scénářů podle integrálních kritérií	66
4.9	Porovnání operátorů podle integrálních kritérií při řízení volantem . .	66
4.10	Pořadí operátorů podle nejnižší hodnoty ISE kritéria (volant)	67
4.11	Porovnání operátorů podle integrálních kritérií při řízení pedály . . .	67
4.12	Pořadí operátorů podle nejnižší hodnoty ISE kritéria (pedály)	67
4.13	Střední dopravní zpoždění pro jednotlivé operátory a scénáře [s] . . .	68
4.14	Identifikované parametry precision modelu operátorů pro scénář 1 . .	70

Seznam výpisů

3.1	Kód skriptu START.m	41
3.2	Kód skriptu Joy_SetDefault.m	42
3.3	Kód funkce KonfiguraceVstupu.m	44
3.4	Kód funkce KonfiguraceVstupu.m	45
3.5	Ukázka užití výjimky	46

Úvod

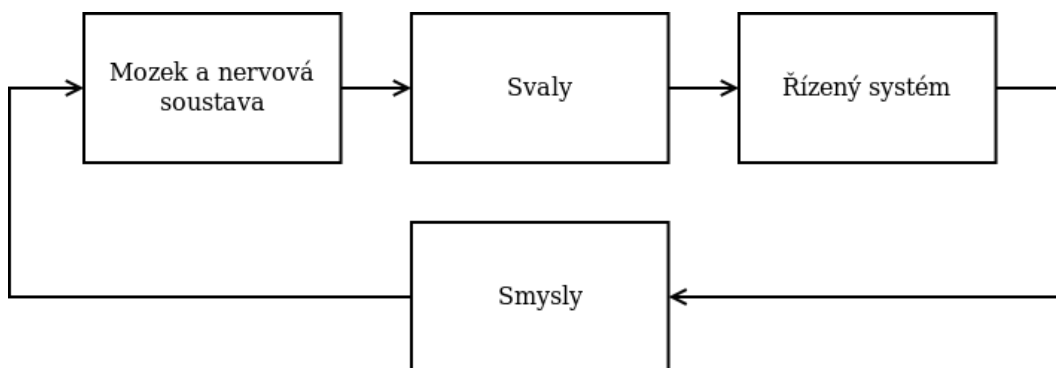
Tato studentská práce se zabývá popisem lidského operátora systému. Konkrétně návrhem a implementací sady testovacích úloh pro měření odezvy lidského operátora.

Lidský operátor, tedy člověk, jako regulátor vyniká zejména ve schopnosti adaptovat se na regulovanou soustavu a využívat schopnosti predikce. Bohužel matematický popis lidského operátora je velmi široká a složitá problematika. V této práci se budu zabývat návrhem simulátoru v programu MATLAB s nadstavbou Simulink, který umožní zaznamenat chování lidského operátora při řízení jednoduchých dynamických systémů. V navrženém simulátoru pak provedu testovací měření.

Získané chování lidského operátora při řízení na elementární úrovni může přispět k výzkumu chování lidského operátora při řízení složitějších reálných systémů, například při výzkumu chování řidiče automobilu nebo pilota letounu. Nahrazení člověka v praktických aplikacích technickými prostředky odstraní některé neduhy charakteristické pro lidského operátora (únava, emoce, fyzikální omezení, ...).

1 Lidský operátor

Lidský operátor je člověk, který působí na řízený technický systém (akčním zásahem) tak, aby ho uvedl do požadovaného stavu. Typicky k tomu využívá smysly (oči, sluch, hmat,...) pro zjištění aktuálního stavu, mozek a nervovou soustavu pro zpracování signálu a svaly pro působení na řízený systém (soustavu), neboli neuromuskulární systém. Zjednodušený zpětnovazební regulační obvod je zobrazený na obrázku 1.1. [1]



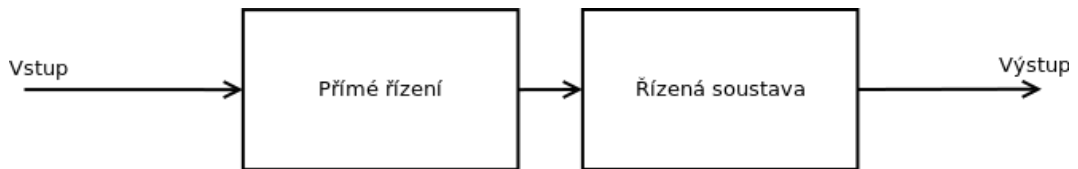
Obr. 1.1: Zpětnovazební regulační obvod člověka

1.1 Způsoby řízení lidského operátora

Pro popis chování lidského operátora nejsou dostatečné prosté modely kompenzačního řízení, protože do řízení vstupuje také schopnost predikce a rozhodování na základě analogií lidského mozku.

1.1.1 Přímé řízení

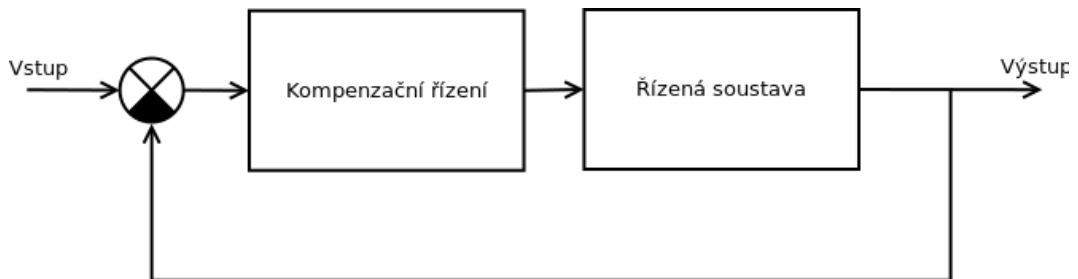
Přímé řízení nevyužívá zpětné vazby a lidský operátor tedy nemá informaci o stavu systému. Lidský operátor využívá tento styl řízení při nacvičených reakcích jako prekognitivní řízení. Tento styl řízení lze používat pouze při dobré znalosti soustavy.



Obr. 1.2: Přímé řízení

1.1.2 Kompenzační řízení

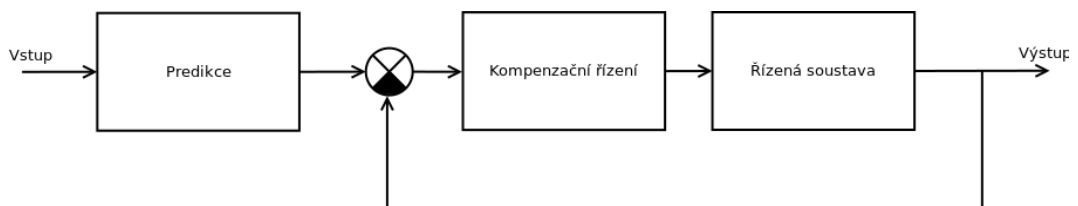
Při kompenzačním řízení nebo také zpětnovazebním řízení lidský operátor využívá zpětnou vazbu o stavu systému k minimalizaci regulační odchylky. Lze tedy mluvit o regulaci. Při regulaci není nutné znát přesné vlastnosti soustavy.



Obr. 1.3: Kompenzační řízení

1.1.3 Výhledové řízení

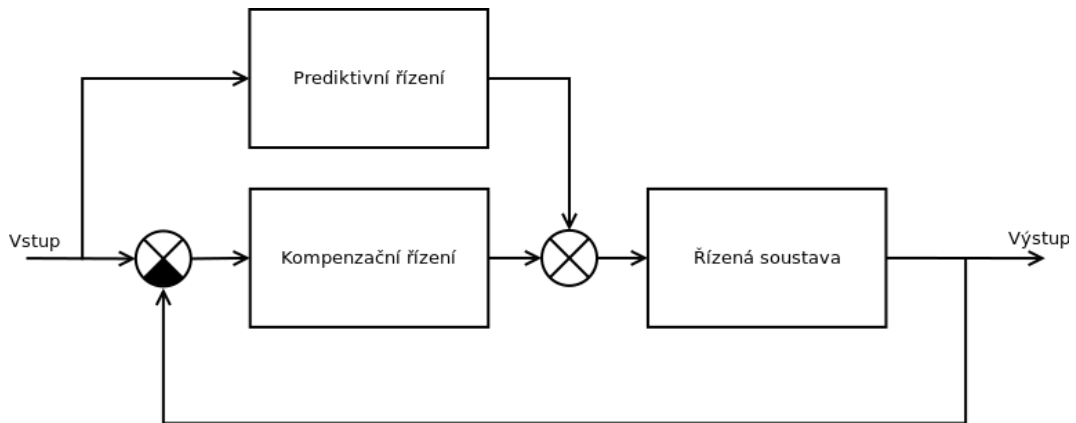
Výhledové řízení je podobné kompenzačnímu řízení, využívá ale schopnost mozku předvídat vývoj žádané hodnoty. To umožňuje začít regulovat změny v předstihu. Regulační odchylka je minimalizována na základě aktuálního stavu soustavy. [8]



Obr. 1.4: Výhledové řízení [8]

1.1.4 Prediktivní řízení

Prediktivní řízení opět vychází z kompenzačního řízení, ale navíc je aplikována schopnost lidského mozku predikovat chování soustavy. To umožňuje působit na soustavu tak, aby nevznikala regulační odchylka. [8]



Obr. 1.5: Prediktivní řízení [8]

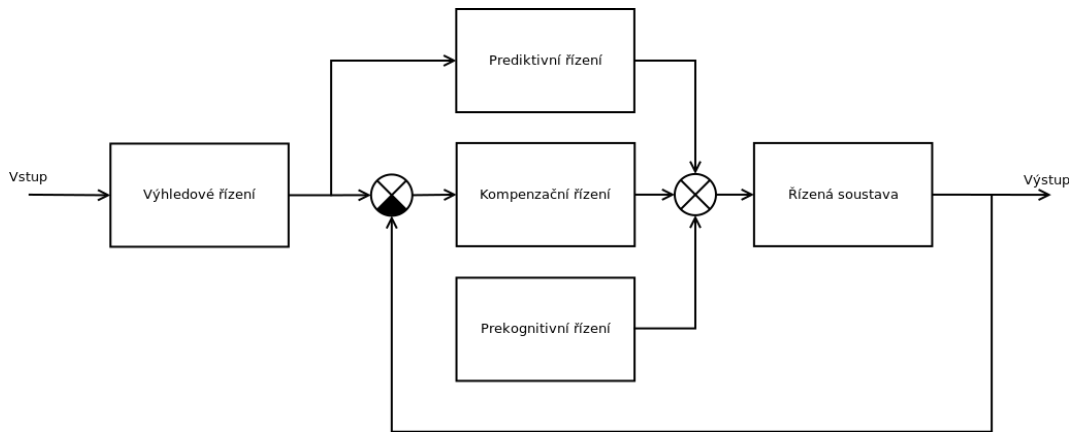
1.1.5 Architektura modelu lidského operátora

Skutečnému chování lidského operátora nejvíce odpovídá kombinace kompenzačního, prekognitivního a prediktivního řízení. Navíc se může uplatnit i výhledové řízení. [8]

Chování celého modelu lze přiblížit na příkladu řízení rychlosti automobilu. Řidič reguluje rychlost na požadovanou hodnotu, uplatňuje se kompenzační řízení, zároveň řidič predikuje budoucí chování vozidla (např. zpomalení při jízdě do kopce), na základě kterého upravuje akční zásah. V případě nutnosti náhlé změny rychlosti (např. objeví-li se v trajektorii automobilu překážka) řidič provede patřičné úkony na základě prekognitivního řízení (např. sešlápne pedál brzdy a spojky, za účelem co nejrychlejšího zastavení vozidla). Výhledové řízení se uplatní ve chvíli, kdy dojde ke změně žádané hodnoty (např. omezení maximální rychlosti).

1.2 Matematické modely lidského operátora při zpětnovazebním řízení

Z technického pohledu se lidský operátor chová jako adaptivní regulátor s více vstupy a více výstupy (MIMO systém). Protože globální analýza chování člověka



Obr. 1.6: Architektura modelu lidského operátora [8]

jako MIMO systému, vzhledem k velkému množství vstupů, vzájemných vazeb a výstupů, je velmi náročná, budu se věnovat případu, kdy se lidský operátor soustředí maximálně na dvě úlohy, které mohou být navzájem provázané (řízení MIMO systému se dvěma vstupy a dvěma výstupy).

Podstata koncepce člověka jako regulátoru je takováto:

1. Stanovení regulačních smyček mezi vjemy lidského operátora a řídicími prvky
2. Optimalizace řídicích smyček adaptací lidského operátora
3. Penalizace řízení

Z měření chování lidského operátora americkým vědcem McRuerem při řízení dynamických systémů byly určeny přenosy otevřené smyčky zapsané v tabulce 1.2. Z této tabulky je patrné, že přenos otevřené smyčky pro různé typy soustavy je po přijetí zjednodušení pokaždé ve stejném tvaru. Z dané skutečnosti vychází "crossover model." [2]

1.2.1 Crossover model

Crossover model lze popsat vztahem [2]:

$$F_0(j\omega) = F_R(j\omega)F_S(j\omega) \doteq \frac{\omega_c e^{-j\omega\tau}}{j\omega} \quad (1.1)$$

Z rovnice 1.1 vyplývá, že lidský operátor se adaptuje na soustavu takovým způsobem, aby výsledný přenos otevřené smyčky měl v blízkém okolí frekvence řezu integrační charakter. Tomu odpovídá sklon frekvenční charakteristiky v logaritmických souřadnicích -20dB/dek.

Přenos crossover modelu v uzavřené smyčce je pak:

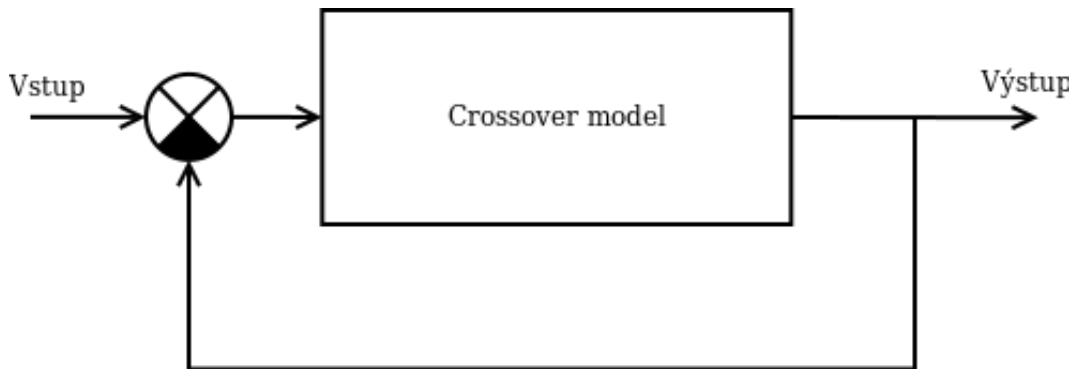
$$\frac{F_0(j\omega)}{1 + F_0(j\omega)} = \frac{\omega_c e^{-j\omega\tau}}{j\omega - \omega_c e^{-j\omega\tau}} \quad (1.2)$$

Přenosová funkce soustavy	Odpovídající chování v letectví	Odpovídající chování v řízení vozidla
K_c	Řízení náklonu letadla nastavením náklonu	Řízení rychlosti
$\frac{K_c}{p}$	Řízení náklonu letadla pomocí autopilota	Řízení směru při malých rychlostech
$\frac{K_c}{p(Tp+1)}$	Řízení výkrutu konvenčních letadel s poklesem časové konstanty výkrutu T	•
$\frac{K_c}{p^2}$	Řízení náklonu kosmických lodí s vypnutým tlumičem	Řízení odstupu od předcházejícího vozidla
$\frac{K_c}{p-a}$	Řízení kurzu směrovkou u kovenčních letounů	•
$\frac{K_c}{(p+a)(p-b)}$	Řízení klopení v nestabilních krátkých okamžicích	Řízení směru přetáčivého vozidla při nadkritické rychlosti

Tab. 1.1: Přenosové funkce soustav a příklady jejich uplatnění v praxi [2]

Přenosová funkce řízené soustavy	Přibližná přenosová funkce lidského operátora	Přenos otevřené smyčky	Efektivní zpoždění pro konkrétního operátora
F_S	F_R	$F_R F_S$	τ
K_S	$\frac{K_R e^{-\tau p}}{p}$	$\frac{\omega_c e^{-\tau p}}{p}$	NA
$\frac{K_S}{p}$	$K_R e^{-\tau p}$	$\frac{\omega_c e^{-\tau p}}{p}$	0,14
$\frac{K_S}{p(p+a)}$	$K_R(p+a)e^{-\tau p}$	$\frac{\omega_c e^{-\tau p}}{p}$	0,16
$\frac{K_S}{p^2}$	$K_R p e^{-\tau p}$	$\frac{\omega_c e^{-\tau p}}{p}$	0,43
$\frac{K_S}{p-b}$	$K_R p e^{-\tau p}$	$\frac{\omega_c e^{-\tau p}}{p}$	0,07
$\frac{K_S}{(p+a)(p-b)}$	$K_R(p+a)e^{-\tau p}$	$\frac{\omega_c e^{-\tau p}}{p}$	0,2

Tab. 1.2: Přenosové funkce systému a přibližné adaptované přenosové funkce lidského operátora, $b \ll \omega_c$ [2]



Obr. 1.7: Uzavřená smyčka s blokem crossover modelu [2]

1.2.2 Gross model

Crossover model 1.1 je platný pouze v blízkém okolí frekvence řezu ω_c . Pro větší rozsah frekvencí je nutné přijmout zobecnění. Takový model se nazývá "Gross" a má tvar: [3]

$$F_R(j\omega) = K_R \frac{T_D j\omega + 1}{T_I j\omega + 1} e^{-j\omega\tau} \quad (1.3)$$

Pomocí zesílení K_R , derivační konstanty T_D a integrační konstanty T_I je realizována adaptace regulátoru (lidského operátora) na regulovanou soustavu. [4]

1.2.3 Precision model

Precision model vznikne zahrnutím dynamiky neuromuskulárního systému člověka do "Gross" modelu. Výsledný vztah tak má tvar: [2]

$$F_R(j\omega) = K_R \frac{T_D j\omega + 1}{T_I j\omega + 1} e^{-j\omega\tau} e^{-j\frac{\alpha}{\omega}} \frac{1}{T_N j\omega + 1} \quad (1.4)$$

Kde:

$$\alpha \doteq \frac{1}{T_K} - \frac{1}{T'_K}$$

$$T_N \doteq T_{N1} + \frac{2\xi_N}{\omega_N}$$

Detailnější podobu precision modelu lze nalézt v [2]

1.2.4 Využití fuzzy systému

Takagi a Sugeno využívají jiný přístup k matematické aproximaci lidského operátora, a to pomocí fuzzy logiky. Popisují tři metody k získání pravidel pro fuzzy systémy. [5]

Crossover model [2]	$F_0(j\omega) = F_R(j\omega)F_S(j\omega) \doteq \frac{\omega_c e^{-j\omega\tau}}{j\omega}$
Gross model [3]	$F_R(j\omega) = K_R \frac{T_D j\omega + 1}{T_I j\omega + 1} e^{-j\omega\tau}$
Precision model [2]	$F_R(j\omega) = K_R \frac{T_D j\omega + 1}{T_I j\omega + 1} e^{-j\omega\tau} e^{-j\omega \frac{\alpha}{\omega}} \frac{1}{T_N j\omega + 1}$

Tab. 1.3: Matematické modely lidského operátora - shrnutí

1. Metoda založená na zkušenostech lidského operátora – nejčastěji používaná metoda, metoda není zobecněná ani zformalizovaná až do návrhu regulátoru.
2. Metoda založená na fuzzy modelu procesu – metoda vycházející z postupů klasické nebo moderní teorie řízení, kdy je regulátor navržen na základě fuzzy identifikace procesu.
3. Metoda založená na akčních zásazích lidského operátora – metoda vyžaduje záznamy z chování lidského operátora, z kterých se určí pravidla a parametry fuzzy systému.
4. Metoda založená na učení – metoda založená na strojovém učení fuzzy regulátoru.

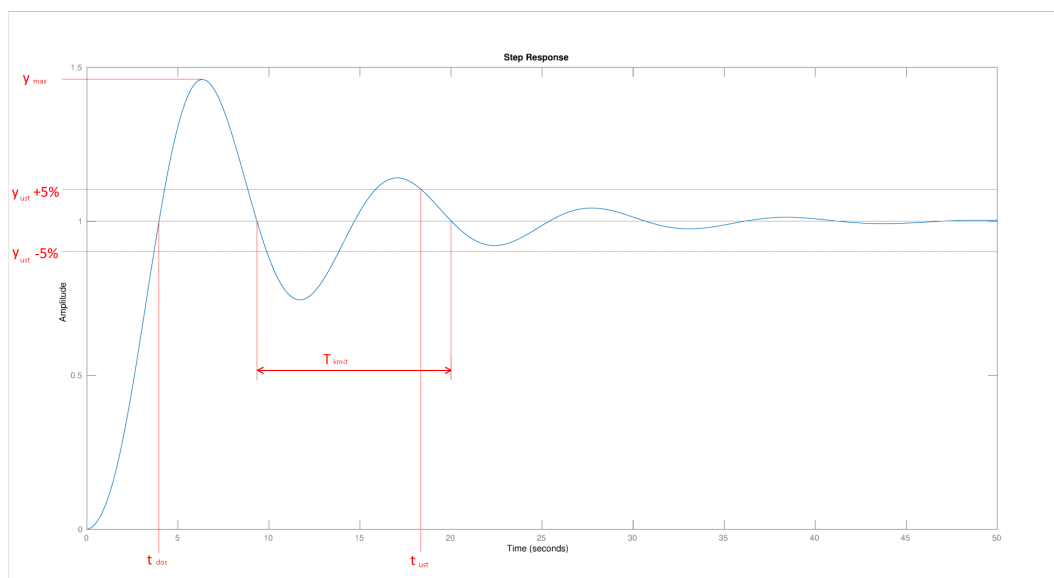
Výhodou užití fuzzy systému je, že model operátora lze sestavit na základě vágního popisu hodnot, které jsou pro člověka přirozené.

1.3 Metody vyhodnocení kvality regulace

Pro vyhodnocení kvality regulace lze využít velké množství různých kritérií, založených na různých principech. Žádné kritérium není univerzální a vždy je nutné zvolit pro konkrétní případ nejvhodnější. Posuzování kvality regulace lze provádět v časové oblasti, z průběhu frekvenčních charakteristik nebo z kořenového hodografu. Pro vyhodnocení kvality regulace lidským operátorem jsou nejvhodnější kritéria časové oblasti, protože nevyžadují identifikaci regulátoru a soustavy. Navíc je lze vyhodnocovat v reálném čase - lze zobrazovat průběžné hodnoty během simulace.

1.3.1 Kritéria podle tvaru přechodové charakteristiky regulované veličiny - časová oblast

Tento typ kritérií vychází z tvaru přechodové charakteristiky a často stanovuje základní požadavek na regulátor při jeho návrhu.



Obr. 1.8: Přechodová charakteristika regulačního obvodu

Překmit

Překmit je maximální hodnota, které dosáhne řízená veličina během regulačního děje. V některých aplikacích je nežádoucí a je nutné užít robustnější regulátor. Na obrázku 1.8 je označen jako y_{MAX} .

V případě lidského operátora odpovídá jeho přístupu (agresivní - velký překmit, zdrženlivý menší nebo žádný překmit). Může být taky způsoben v době adaptace operátora, když se teprve seznamuje s vlastnostmi soustavy.

Doba do ustálení

Časový okamžik, za kterou hodnota řízené veličiny dosáhne tolerančního pásma žádané hodnoty a setrvá v něm. Po proběhnutí této doby lze považovat hodnotu řízené veličiny za ustálenou, ačkoli může stále mírně kmitat. Na obrázku 1.8 je doba do ustálení označena jako t_{ust} .

Dobu do ustálení lze ve většině případů (někdy v kombinaci s překmitem) považovat za nejdůležitější parametr pro lidského operátora.

Perioda kmitů

Perioda kmitů je na obrázku 1.8 označena jako T_{kmit} . Lze jí definovat jako časový okamžik mezi dvěma průchody hodnoty řízené veličiny požadovanou hodnotou. Kmity se v přechodové charakteristice nachází pouze u kmitavých systémů. Pro lidského operátora, je periodu kmitů, pokud se vyskytuje, obtížné kvantifikovat.

Doba regulace

Časový okamžik, za který řízená veličina poprvé dosáhne požadované hodnoty. Na obrázku 1.8 je označena jako t_{dos} .

Pro většinu systémů je ale důležitější čas do ustálení, protože cílem je dosáhnout požadované hodnoty a tuto hodnotu udržet.

Integrální kritéria

Jiný typ kritérií vychází z principu integrace regulační chyby e . Mezi nejvyužívanější patří tyto 4

- Lineární kritérium
- Usměrněné lineární kritérium
- Kvadratické kritérium
- ITAE kritérium
- ITSE kritérium

Lze ovšem nadefinovat i libovolné, jiné, integrální kritérium, které bude nejlépe vystihovat konkrétní požadavky na regulátor.

Lineární kritérium

Nejjednodušší z výše zmíněných kritérií. Jedná se o prostou integraci regulační

odchylky. Nevýhodou je, že chyba může mít různou polaritu a záporná chyba snižuje integrál kladné chyby. Kmitavý jev tak může, z pohledu tohoto kritéria, působit velmi kvalitně. Lineární kritérium je definováno vztahem: [6]

$$J_L = \int_0^{\infty} [e(t) - e(\infty)] dt \quad (1.5)$$

Usměrněné lineární kritérium

Usměrněné lineární kritérium, v anglické literatuře Integral of Absolute Error (IAE), vznikne modifikací lineárního kritéria. Integrací absolutní hodnoty regulační odchylky e dojde k odstranění nedostatku v podobě zkreslení hodnoty kritéria při kmitavých jevech. Usměrněné lineární kritérium je definováno vztahem: [6]

$$J_{IAE} = \int_0^{\infty} |e(t) - e(\infty)| dt \quad (1.6)$$

Kvadratické integrální kritérium

Jinou modifikací lineárního kritéria je kvadratické lineární kritérium, v anglické literatuře Integral of Square Error (ISE). Protože velká odchylka často bývá méně žádoucí než drobnější déletrvající odchylka, kvadratické lineární kritérium "pokutuje" větší odchylky více než menší. Navíc, protože kvadrát reálného čísla je vždy kladné číslo, také odstraňuje nedostatek v podobě zkreslení hodnoty kritéria při kmitavých jevech. Kvadratické lineární kritérium je definováno vztahem: [6]

$$J_{ISE} = \int_0^{\infty} [e(t) - e(\infty)]^2 dt \quad (1.7)$$

Pro snížení hodnoty kvadratického integrálního kritéria by se měl člověk při regulaci chovat spíše zdrženlivě.

ITAE kritérium

ITAE (Integral of Time multiplied by Absolute value of Error) kritérium váhuje regulační odchylku e časem t , nebo-li regulační odchylka v pozdějším čase je "pokutována" více než stejná odchylka v dřívějším čase. ITAE kritérium je definováno takto: [6]

$$J_{ITAE} = \int_0^{\infty} t|e(t)| dt \quad (1.8)$$

Pro snížení hodnoty ITAE kritéria by měl člověk regulovat spíše agresivně, aby doba do ustálení byla co nejkratší.

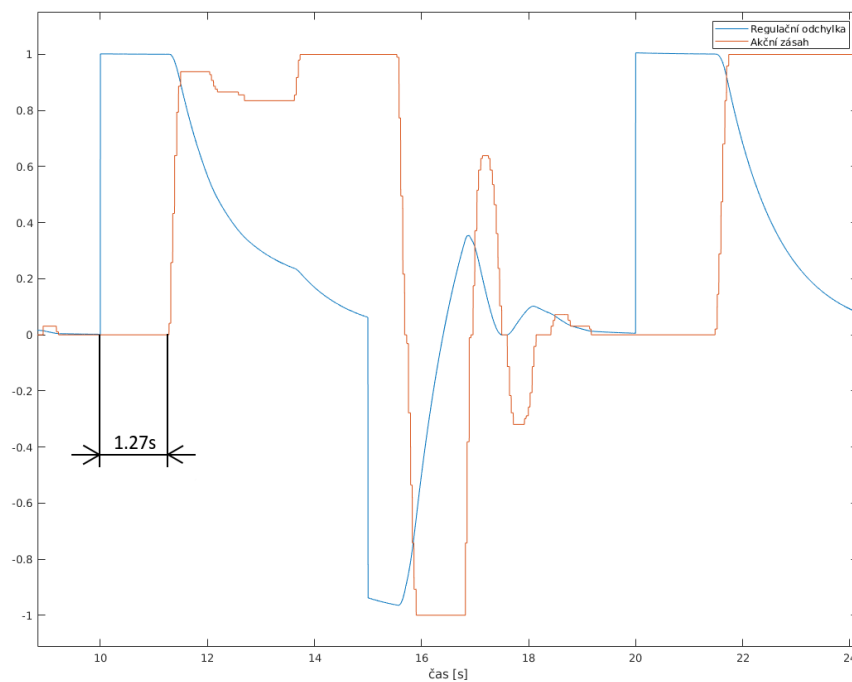
ITSE kritérium

ITSE (Integral of Time multiplied by Square of Error) kritérium kombinuje kvadratické integrální kritérium a ITAE kritérium. Větší regulační odchylku e pokutuje více než drobnější, ale zároveň je váhováno časem t . Odchylka v pozdějším čase t je tedy také více pokutována. ITSE kritérium je definováno takto:

$$J_{ITSE} = \int_0^{\infty} te(t)^2 dt \quad (1.9)$$

Zpoždění reakcí regulátoru

Dalším kvantifikovatelným kritériem pro hodnocení kvality regulátoru může být zpoždění regulátoru, neboli dopravní zpoždění. Lze jej definovat jako časový okamžik mezi změnou regulační odchylky a adekvátním zásahem regulátoru. Hodnotu zpoždění reakcí regulátoru lze zjistit přímo odečtením z grafu průběhu, jak je znázorněno na obrázku 1.9. Jinou možností je aplikace korelační analýzy.



Obr. 1.9: Porovnání regulační odchylky a akčního zásahu regulátoru

Korelační analýza

Za předpokladu, že výstup regulátoru je korelovaný s velikostí regulační odchylky, lze určit zpoždění regulátoru vhodnou aplikací korelační analýzy. Obecně pro výpočet zpoždění signálu D lze odhadnout z $\tau = \widehat{D}$, pro které vzájemná korelace $\phi(\tau)$ dosahuje maxima tj. [7]

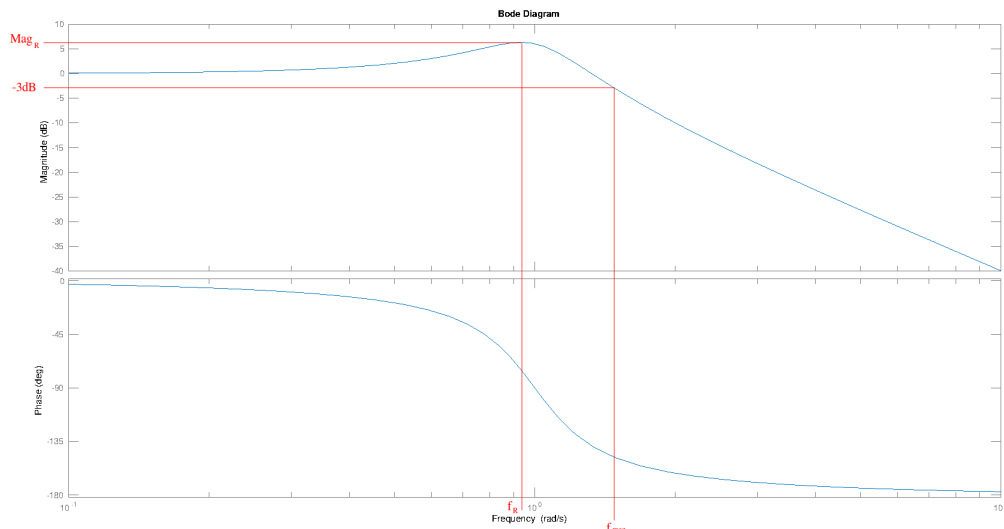
$$\phi(\widehat{D}) = \max_{\tau} \phi(\tau) = \max_{\tau} \int_{-\frac{T}{2}}^{\frac{T}{2}} dt x(t) y(t + \tau) \quad (1.10)$$

Vztah 1.10 však nelze obecně aplikovat pro výpočet zpoždění přenosu regulátoru, ale je nutné uvážit charakter soustavy.

1.3.2 Kritéria kvality regulace ve frekvenční oblasti

Člověk jako regulátor má specifické frekvenční charakteristiky dány omezením neuromuskulární soustavy. Výkonnost neuromuskulární soustavy je navíc ovlivněna psychickými a fyzickými stavy, jako je třeba únava nebo stres.

Pro vyhodnocení kvality regulace ve frekvenční oblasti lze použít frekvenční charakteristiky v logaritmických souřadnicích nebo v komplexní rovině.



Obr. 1.10: Amplitudová a fázová frekvenční charakteristika regulačního obvodu

Mezní frekvence

Je frekvence, při níž amplituda poklesne o 3dB. Pro větší pokles se již předpokládá, že systém přestává přenášet signál. Mezní frekvence vymezuje **frekvenční**

rozsah systému. Na obrázku 1.10 je mezní frekvence označena jako f_{mez} . Frekvenční rozsah zobrazeného systému je tedy 0 až 1,5 rad/s.

Při frekvencích vyšších než mezní frekvence, lidský operátor postupně nezvládá reagovat dostatečně rychle na vstupní signál.

Rezonanční frekvence a rezonanční převýšení

Rezonanční frekvence je taková, při které amplitudová frekvenční charakteristika dosahuje svého maxima. Velikost tohoto maxima, rezonanční převýšení, odpovídá velikosti překmitu. Rezonanční kmitočet je na obrázku 1.10 označen jako f_r , rezonanční převýšení Mag_R .

Signály se silně zastoupenými frekvencemi v okolí operátorovy rezonanční frekvence mohou vést k rezonanci s destruktivními následky.

Zásoba stability v amplitudě

Udává velikost zesílení, které přivede otevřený regulační obvod na mez stability. [6]

Na frekvenční charakteristice odpovídá poklesu amplitudové frekvenční charakteristiky při frekvenci, kdy fázová frekvenční charakteristika odpovídá 180° .

Zásoba stability ve fázi

Udává záporně velikost změny fáze, která přivede otevřený regulační obvod na mez stability. [6]

Na frekvenční charakteristice odpovídá vzdálenosti fázové frekvenční charakteristiky od hodnoty 180° při frekvenci, kdy amplitudová frekvenční charakteristika odpovídá 0 dB.

1.4 identifikace systému

Naměřené průběhy odezvy jednotlivých lidských operátorů při řízení je nutné identifikovat, tedy popsat je vhodným matematickým modelem. Matematický model může vycházet z apriorních znalostí o systému, tedy vycházet z dosavadních znalostí funkce neuromuskulárního systému. Jinou možností může být čistě matematicky nejvhodnější aproximace chování zkoumaného systému.

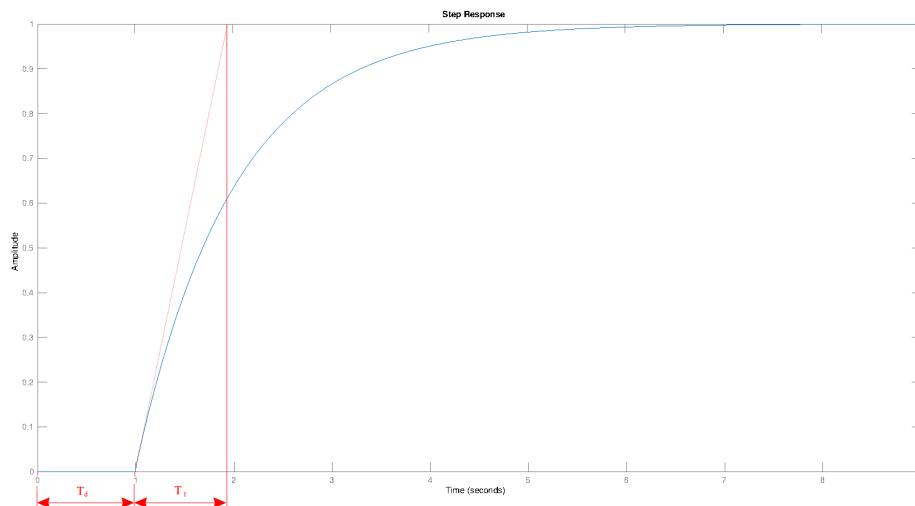
1.4.1 Identifikace z přechodové charakteristiky

Při této metodě se odečítají parametry modelu z naměřené přechodové charakteristiky – aproximuje se přechodová charakteristika. Tato metoda patří mezi deterministické metody. Její použití je vhodné při zanedbatelném šumu. [10]

Aproximace proporciální soustavou se setrvačností 1. řádu a dopravním zpožděním

Tvar přenosu soustavy se setrvačností 1. řádu a dopravním zpožděním T_d : [10]

$$F(p) = \frac{K}{T_1 p + 1} \exp^{-T_d p} \quad (1.11)$$



Obr. 1.11: Přechodová charakteristika systému 1. řádu s dopravním zpožděním

Zesílení K lze vypočítat z poměru ustálené hodnoty výstupu a vstupu $\frac{y(\infty)}{u(\infty)}$. Dopravní zpoždění T_d odpovídá době mezi změnou vstupního signálu a začátkem reakce ve výstupním signálu.

Časovou konstantu T_1 lze určit stanovením tečny k přechodové charakteristice v době T_d .

Postup identifikace složitějších průběhů je popsán například zde: [10].

1.4.2 Identifikace systému s využitím programu MATLAB

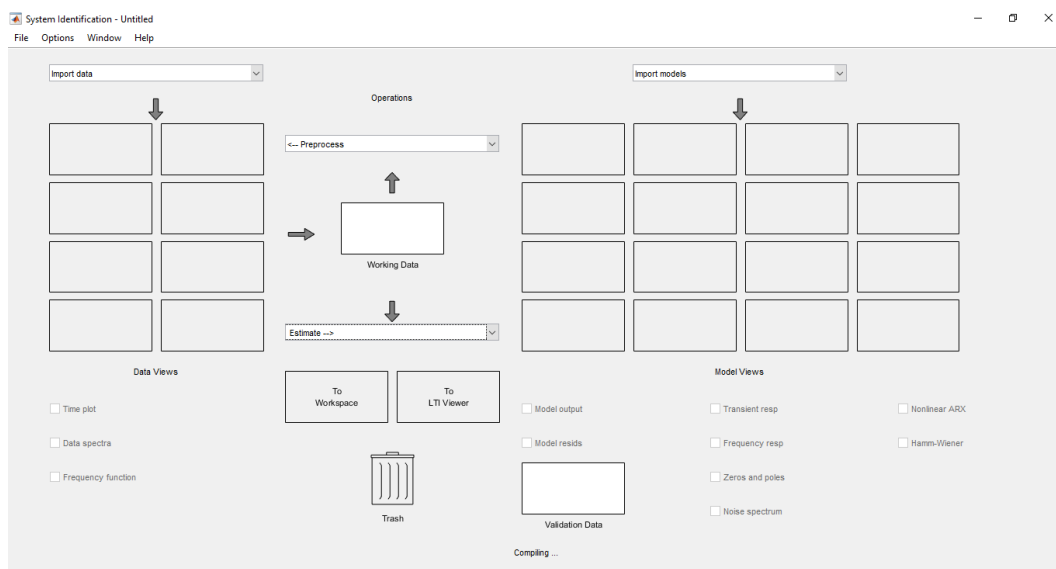
Pro pohodlnou identifikaci systému bez nutnosti širší znalosti matematického pozadí procesu identifikace systému lze využít program MATLAB. K provedení identifikace systému lze použít více technik, ale nejvhodnější je použití grafické aplikace System Identification Tool, kterou lze vyvolat příkazem `ident`. Tento nástroj využívá k nalezení nejvhodnějšího modelu metodu nejmenších čtverců. Ta je založena na minimalizaci chyby modelu a lze ji popsat vztahem: [11]

$$J(\vartheta) = \frac{1}{2} \sum_{k=1}^N \varepsilon^2 \quad (1.12)$$

kde

$\varepsilon(k) = y(k) - \varphi^T(k)\theta$ je chyba modelu

$\varphi^T(k)\theta$ je odhad výstupu modelu



Obr. 1.12: Okno System Identification Tool

Po levé straně okna "System Identification" je oblast pro práci s daty a po pravé straně je oblast pro práci s modely. Nejprve je nutné nainportovat data. To se pro-

vede rozbalením roletky "Import data" a následně volbou formátu dat. Ta můžou být v časové nebo frekvenční oblasti, případně lze zvolit možnost "Data object...", či "Examples..." pro již existující dataset. Otevře se okno importu dat, kde se zadají vstupní a výstupní data a následně se může stiskem tlačítka "Import" vytvořit dataset. V případě, že naměřená data jsou uložena jako struktura s časem, je nutné zadat přímo hodnoty signálu, tedy `STRUKTURA.signals.values`. Velikost vstupních i výstupních dat musí být stejná a s konstantní vzorkovací periodou. Takovou úpravu dat lze provést pomocí funkcí ze struc toolboxu `StrucZOHE` a `StrucInterp`. Dále se zadá název datasetu, čas počátku a vzorkovací perioda. Pro případ struktury s časem se počáteční čas může zadat jako `STRUKTURA.time(1)` a pokud je konstantní vzorkovací perioda, lze ji zadat jako `STRUKTURA.time(2)-STRUKTURA.time(1)`.

Data je možné porovnat v časové oblasti zaškrtnutím volby "Time plot" nebo frekvenční oblasti - "Data spectra".

Data lze také nejprve předzpracovat (například odstranit střední hodnotu nebo trend), to lze provést přetažením vybraného datasetu na pozici "Working Data", rozbalením roletky "Preprocess" a volbou požadované operace.

Návrh matematického modelu na základě dat zvolených jako "Working Data" lze provést rozbalením roletky "Estimate" a volbou požadovaného tvaru modelu. V případě volby přenosové funkce ("Transfer function") lze volit počet pólů, počet nul a zda se má jednat o diskrétní nebo spojitý model. Lze zvolit velikost dopravního zpoždění a to buď uživatelem zadané a nebo automaticky určené. Je možné také zvolit další parametry estimace modelu, jako frekvenční rozsah, počáteční podmínky a další. Jinou možností je "Process model", kdy lze zvolit počet pólů a zda je v modelu přítomna nula, dopravní zpoždění nebo integrátor. Všechny parametry lze nastavit na předem nastavenou konstantu, nebo nechat určit automaticky z daného rozsahu. Estimovaný model je možné porovnat s validačními daty. Například lze porovnat výstup na základě vstupních dat datasetu ("Model output"), zároveň se zobrazí míra shody systémů. Ta se vypočítá podle vztahu 1.13. Validační data by neměla být shodná s daty, ze kterých bylo vycházeno při určování modelu, aby bylo zabráněno závislosti modelu na vstupních datech.

Vybraný model lze uložit do Workspace programu matlab přetažením modelu na ikonu "To Workspace" a dále s ním pracovat v základním prostředí MATLABu. [9]

$$FIT = 100\left(1 - \frac{\|y - \hat{y}\|}{\|y - \text{mean}(y)\|}\right) \quad (1.13)$$

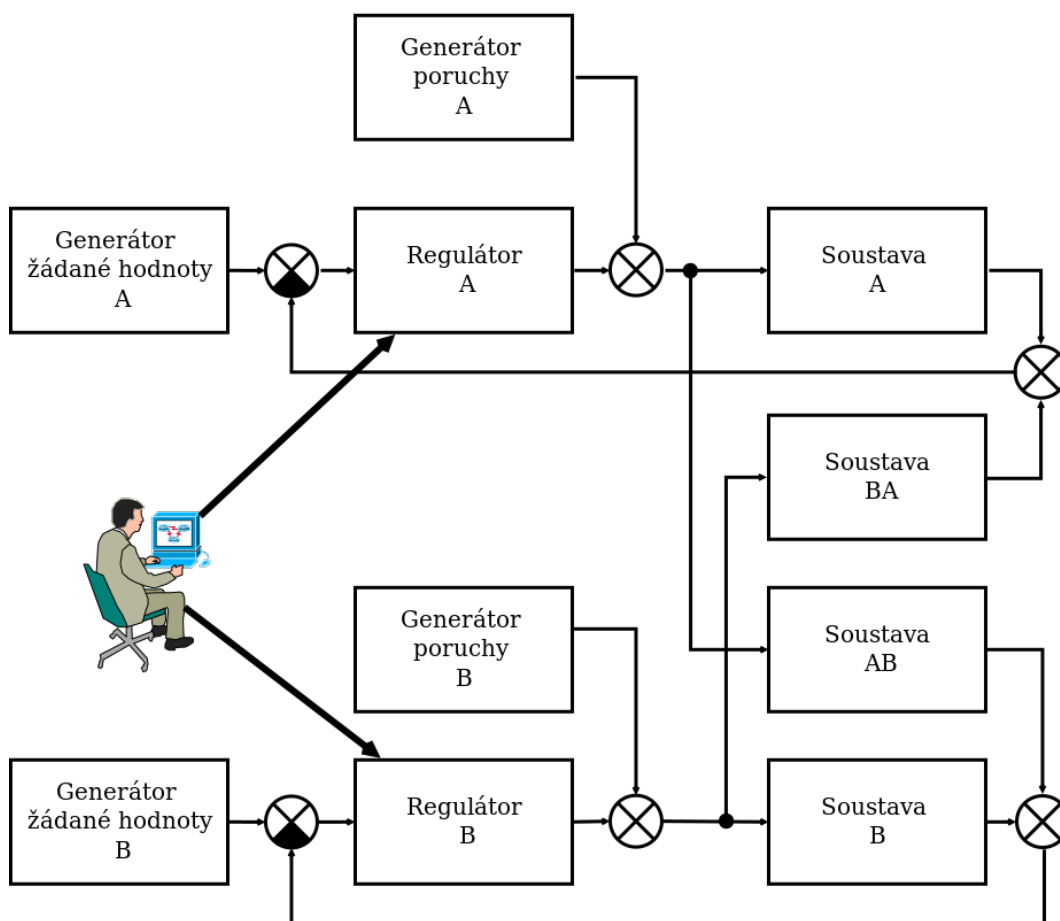
Kde

y validační data
 \hat{y} výstup modelu

2 Návrh simulátoru pro měření odezvy lidského operátora

Jak vyplývá z předešlých kapitol základem simulátoru by měla být regulační smyčka. Celá smyčka bude v simulátoru zastoupena dvakrát, označena jako subsystém A a subsystém B. To umožní otestovat člověka i v případě, že se má soustředit na dvě úlohy zároveň. Celý simulátor bude mít formu aplikace s grafickým uživatelským rozhraním (GUI). Naměřené hodnoty se budou ukládat do workspace programu MATLAB. Vzhledem k dostupnému softwarovému vybavení by měl být simulátor kompatibilní se staršími verzemi programu MATLAB (r2015b) a s operačními systémy Microsoft Windows a GNU/Linux.

Testovací scénáře budou tvořeny různou konfigurací parametrů jednotlivých bloků, zejména žádané hodnoty (pulzy, superpozice sinusoid,...) a soustav (přenosy, dopravní zpoždění). To umožní tvorbu široké škály testovacích scénářů.



Obr. 2.1: Blokové schéma simulátoru

2.1 Žádaná hodnota w

Žádaná hodnota w bude generována na základě uživatelem nastavených parametrů. Tvar signálu žádané hodnoty bude volitelný:

1. Obdelníkový signál – pro měření odezvy na jednotkový skok
2. Náhodný signál – při dlouhé periodě vzorkování se chová jako jednotkový skok, ovšem nepredikovatelně velký
3. Lineárně rostoucí signál – zejména pro soustavy integračního charakteru
4. Superpozice sinusoid – může mít tvar jedné sinusoidy, případně superpozice více sinusoid
5. Konstanta 0 – žádaná hodnota neaktivní

Signál žádané hodnoty bude možné před vstupem do regulační smyčky zpozdít dopravním zpožděním, ale zobrazován bude před zpožděním. To umožní testovat prediktivní řízení lidského operátora.

2.2 Porucha v

Generátor poruchové veličiny bude formálně shodný s výše zmíněným generátorem žádané hodnoty.

2.3 Regulátor R

Regulátor může být přepnut do jednoho ze dvou stavů:

1. Akční zásah z externího vstupu – externím vstupem může být libovolné herní zařízení (volant, pedály, joystick) s alespoň jedním analogovým výstupem.
2. Akční zásah na základě výstupu matematického regulátoru se zadanými parametry

Dva stavy umožní porovnat a demonstrovat kvality matematického regulátoru proti lidskému operátorovi. Velikost akčního zásahu pro oba stavy bude omezena v intervalu $[-1; 1]$. Dále může být akční zásah zesílen, případně omezen základními typy nelinearit (omezení, pásmo necitlivosti, hystereze, omezení strmosti) a dopravním zpožděním.

2.4 Soustava S

Soustava bude propojovat regulační smyčku A a regulační smyčku B. Bude tvořena 4 SISO soustavami (soustava v regulační smyčce A, soustava mezi regulační

smyčkou A a B, soustava mezi regulační smyčkou B a A a soustava v regulační smyčce B). Jednotlivé soustavy budou zadány přenosy a dopravními zpožděními.

Jak již bylo zmiňováno, dvě regulační smyčky a soustavy, které je vzájemně propojují, umožní testovat počínání lidského operátora při řešení dvou paralelních úloh.

2.5 Rozhraní pro styk s měřenou osobou

Rozhraní pro styk s měřenou osobou na straně vstupu do regulační smyčky bude moci tvořit libovolné vstupní herní zařízení s alespoň jedním analogovým vstupem. Vzhledem k rozmanitosti herních zařízení bude simulátor umožňovat přiřazení vybrané osy k danému subsystému s možností reverzace osy. Dále vzhledem k omezenému rozlišení AD převodníku v herních zařízení bude možné signál z tohoto zařízení filtrovat filtrem typu dolní propust.

Rozhraní pro styk s měřenou osobou na straně výstupu bude realizováno monitorem, na kterém bude možné zobrazit všechny veličiny regulačního obvodu. Formát zobrazení bude volitelný v GUI ze tří možností:

1. Pursuit display – zobrazení bodu na ploše
2. Scope – zobrazení průběhu v čase (pro každý subsystém zvlášť)
3. VW – Virtual World - zobrazení ve virtuální realitě

Různé způsoby zobrazení umožní posoudit jejich vliv na kvalitu regulace lidským operátorem.

3 Realizace simulátoru

Pro realizaci simulátoru zpětnovazebního systému s lidským operátorem byl zvolen program MATLAB s nadstavbou Simulink. Model dynamického systému sestává ze dvou nezávislých smyček, které lze propojit MIMO soustavou. Umožňuje tedy testovat řízení jednoduchého systému (v praxi může odpovídat regulaci výkonu), dvou jednoduchých systémů zároveň (řízení polohy letadla vůči horizontální rovině) nebo řízení systému se dvěma vstupy a simulaci nelinearit akčního zásahu a porovnání s PID regulátorem nebo regulátorem popsaným přenosovou funkcí.

Při tvorbě simulátoru jsem dbal na co největší univerzálnost. Jako vstupní zařízení lze použít libovolný analogový herní kontrolér. Simulátor je implementován tak, aby byl podporován operační systémy Microsoft Windows a GNU/Linux. Program simulátoru lze rozdělit do tří částí:

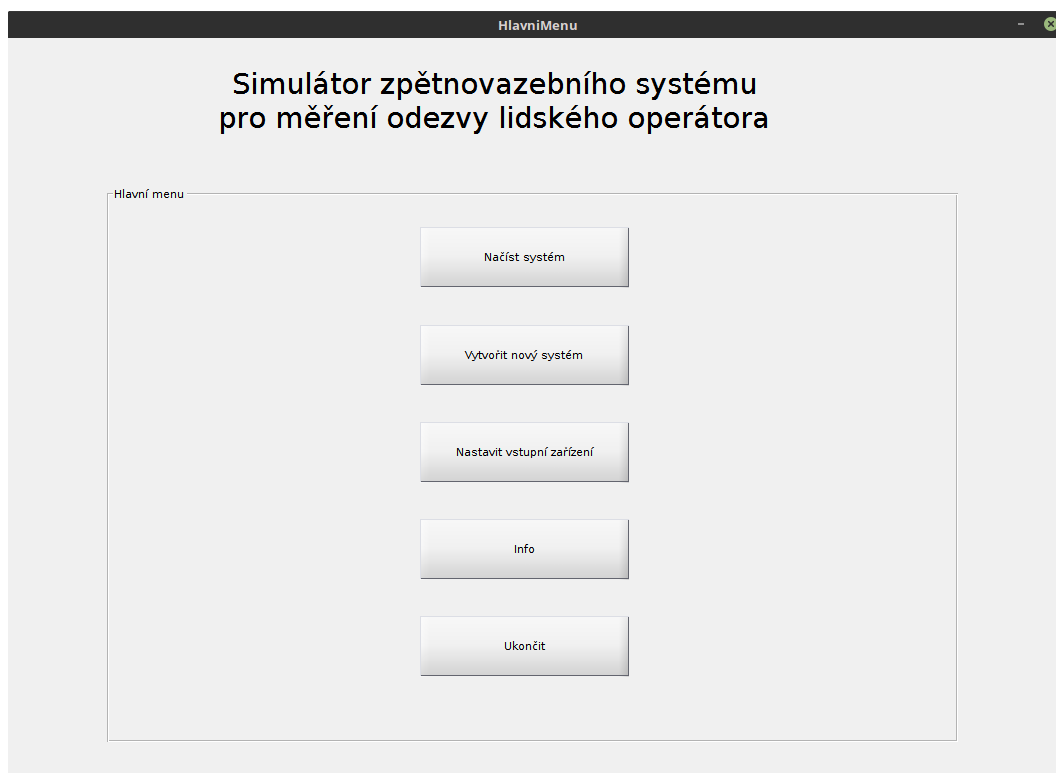
1. GUI – nastavení parametrů simulace, řízení simulace
2. Simulink – výpočet simulací, zobrazení
3. Struc toolbox – zpracování naměřených hodnot

3.1 Vlastnosti simulátoru

- Dva nezávislé subsystemy A a B, které mohou být propojeny do MIMO systému
- Běh v reálném čase
- Volba vstupu a předzpracování vstupního signálu
- Volba žádané hodnoty (konstanta, obdélníkový signál, náhodný signál, lineárně rostoucí signál, superponované sinusoidy)
- Volba poruchové veličiny (konstanta, obdélníkový signál, náhodný signál, lineárně rostoucí signál, superponované sinusoidy)
- Volba regulátoru, možnost použití PID regulátoru nebo regulátoru popsaného přenosovou funkcí pro porovnání
- Volba nelinearit a dopravního zpoždění akčního zásahu
- Volba přenosu a dopravního zpoždění soustavy, možnost MIMO systému
- Volba zobrazení - možnost kompenzace regulační odchylky nebo následování žádané hodnoty
- Logování naměřených hodnot
- Uložení nakonfigurovaného systému
- Struc toolbox pro práci s naměřenými daty

3.2 Obsluha simulátoru

Zavoláním skriptu START.m dojde ke spuštění GUI. Zobrazí se hlavní menu viz obr. 3.1. Jednotlivé volby budou popsány podrobněji.



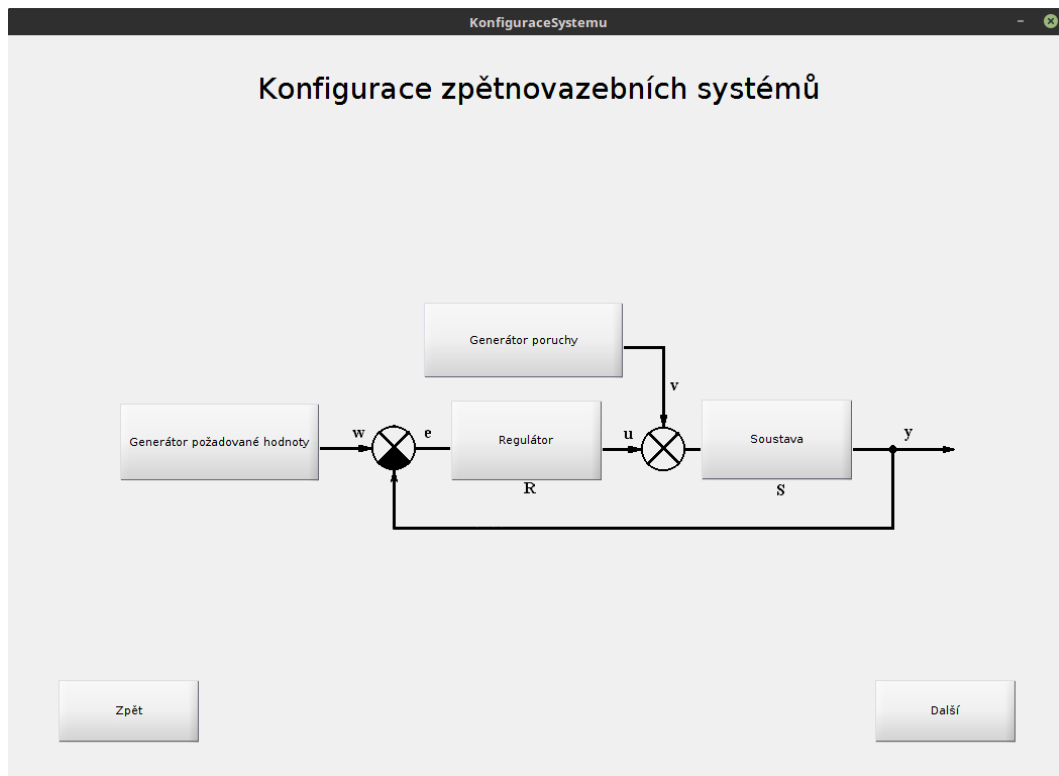
Obr. 3.1: Hlavní menu

3.2.1 Načíst systém

Tato volba otevře okno pro výběr uloženého systému s příponou ".system". Po výběru žádaného systému se otevře okno konfigurace systému, ve kterém lze konfigurovat zvolený systém podobně, jako při tvorbě nového systému.

3.2.2 Vytvořit nový systém

Dojde k otevření okna konfigurace systému, ve kterém se zobrazí celková topologie systému a poklikáním na jednotlivé bloky se otevře konfigurace příslušného bloku. Tlačítkem "Zpět" je možné se vrátit do hlavního menu, tlačítkem "Další" lze přejít ke konfiguraci zobrazení.



Obr. 3.2: Konfigurace systému

Generátor žádané hodnoty

Umožňuje volbu jedné z možností žádané hodnoty pro každý subsystém. Volba "Zpoždění žádané hodnoty" nastavuje zpoždění mezi zobrazením a vstupem do systému. Dále na základě ověřovacího měření bylo přidáno zaškrtačací pole "Náhodná fáze". Pokud je toto pole aktivováno, dojde při spuštění simulace k vygenerování náhodných fázových posunů, případně k vygenerování náhodných výchozí hodnoty pro generátor náhodných pulzů.

Obdelníkový signál

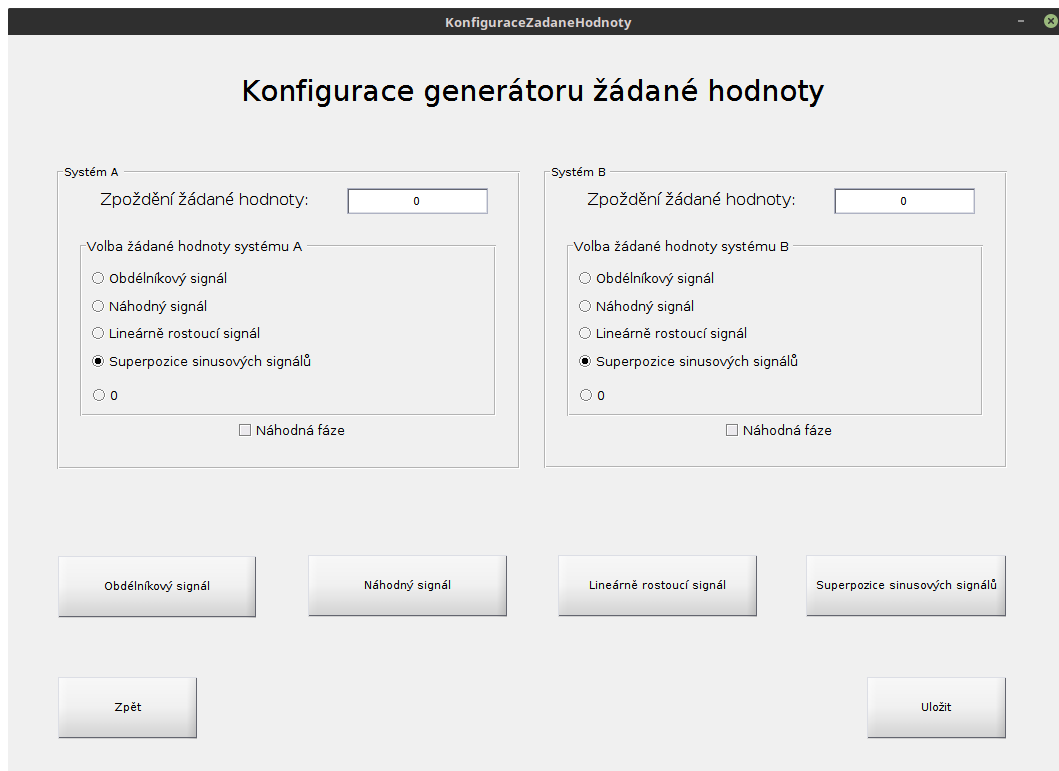
Lze nastavit:

- Amplitudu
- Periodu
- Střídu

Náhodný signál

Lze nastavit:

- Amplitudu
- Periodu vzorkování



Obr. 3.3: Konfigurace generátoru žádané hodnoty

Lineárně rostoucí signál

Lze nastavit:

- Strmost
- Počáteční čas

Superpozice sinusových signálů

Lze superponovat až 10 sinusoid. Pro každou sinusoidu lze nastavit:

- Amplitudu
- Frekvenci
- Bias (stejnoseměrnou složku)
- Fázi

Generátor poruchy

Generátor poruchy je obdobný jako generátor žádané hodnoty.

Regulátor

V bloku regulátoru lze zvolit jednu z možností regulátoru:

- Analogový vstup – regulátorem je lidský operátor
- PID regulátor – regulátor lze zadat ve formátu popsaném rovnicí 3.1
- Regulátor zadaný přenosem – regulátor lze zadat ve formátu $tf([b_n \dots b_1 b_0],[a_n \dots a_1 a_0])$
- Bez regulátoru – regulátor je nahrazen prostým propojením, uplatňují se pouze nelinearity a dopravní zpoždění

$$P + I\frac{1}{p} + D\frac{p}{1 + 0.01p} \quad (3.1)$$

Výstup regulátoru je omezen v rozsahu -1 až 1, aby rozsah matematických regulátorů odpovídal rozsahu analogového vstupu. Dále lze nastavit tyto parametry nelinearit:

- Zesílení regulátoru - po omezení rozsahu
- Maximální strmost akčního zásahu - při růstu i při klesání
- Hystereze - šířka
- Necitlivost - lze nastavit počátek a konec pásma necitlivosti
- Omezení - Omezení maximálního nebo minimálního rozsahu regulátoru
- Dopravní zpoždění regulátoru

Obr. 3.4: Konfigurace regulátoru

Soustava

Umožňuje nastavit přenosy soustavy ve formátu $\text{tf}([b_n \dots b_1 b_0], [a_n \dots a_1 a_0])$ a dopravní zpoždění jednotlivých přenosů. MIMO přenos soustavy je rozdělen na 4 SISO přenosy.

- Soustava A – vstupem je akční zásah a porucha subsystému A a výstup je připojen do subsystému A
- Soustava B – vstupem je akční zásah a porucha subsystému B a výstup je připojen do subsystému B
- Soustava A \Rightarrow B – vstupem je akční zásah a porucha subsystému A a výstup je připojen do subsystému B
- Soustava B \Rightarrow A – vstupem je akční zásah a porucha subsystému B a výstup je připojen do subsystému A

Konfigurace soustavy

Soustava A

Přenos:

Dopravní zpoždění:

Soustava B

Přenos:

Dopravní zpoždění:

Soustava A=B

Přenos:

Dopravní zpoždění:

Soustava B=A

Přenos:

Dopravní zpoždění:

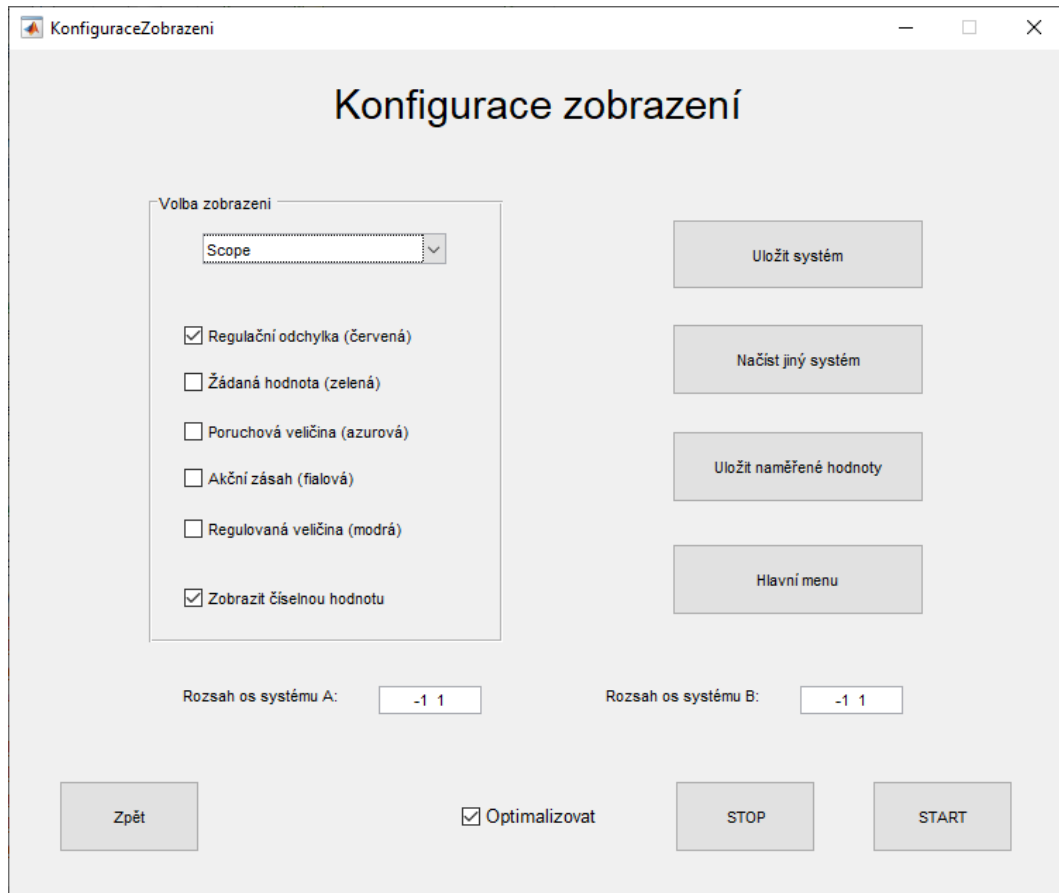
Zpět Uložit

Obr. 3.5: Konfigurace soustavy

3.2.3 Konfigurace zobrazení

V okně konfigurace zobrazení je možné nastavit formát zobrazení, které veličiny mají být zobrazeny, zda má být zobrazena číselná hodnota veličin a rozsah zobrazení. Dále je možné nakonfigurovaný systém uložit, případně načíst jiný nebo uložit

naměřené hodnoty do souboru s příponou .mat. Spustit simulaci lze stiskem tlačítka START, simulaci je poté možné ukončit stiskem tlačítka STOP. Je-li zaškrtnuta volba optimalizovat, dojde po stisku tlačítka START k zakomentování nevyužitých částí Simulink modelu.



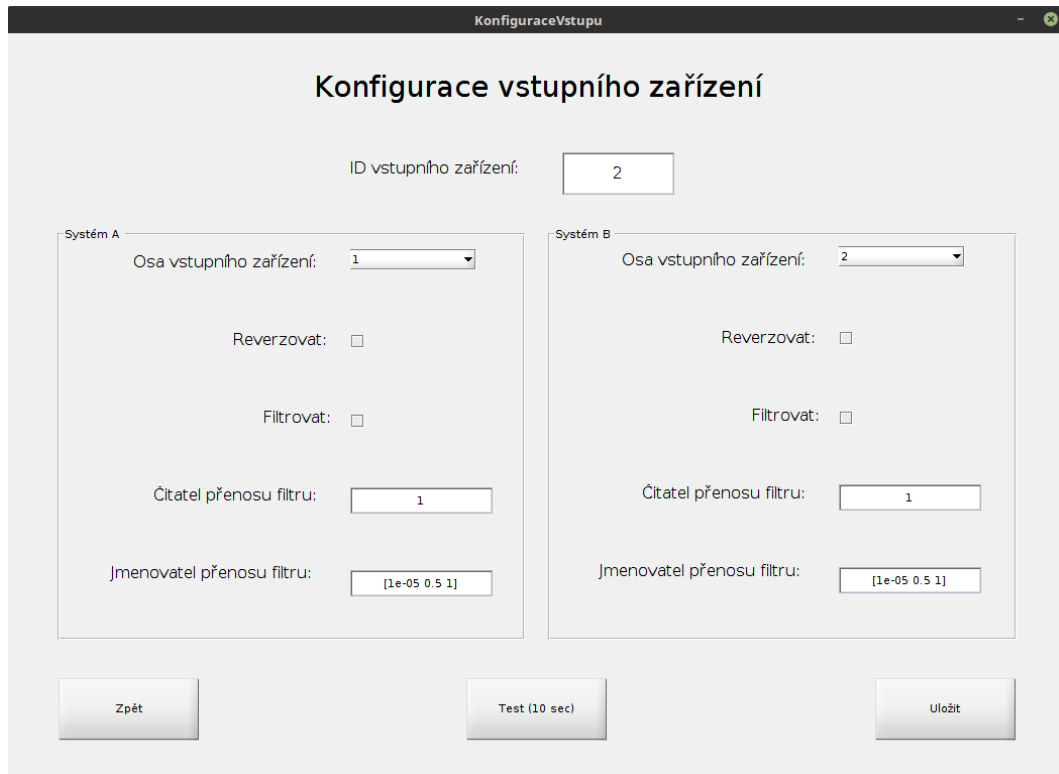
Obr. 3.6: Konfigurace zobrazení

3.2.4 Nastavit vstupní zařízení

Při konfiguraci vstupního zařízení lze volbou ID vybrat, které vstupní zařízení má být použito. Pro operační systém MS Windows je výchozí ID nastaveno na 1, pro operační systém GNU/Linux je výchozí ID nastaveno na 2. Dále lze pro oba subsystemy zvolit, která osa má být použita a zda má být reverzována, případně zaškrtnutím volby Filtrovat se uplatní filtr. Filtr je popsán číselnými a jmenovatelnými koeficienty.

Dále je možné stiskem tlačítka "Test (10 sec)" otestovat po dobu 10 s chování vstupního zařízení. Dojde k zobrazení vstupu obdobně jako při simulaci. Stiskem

tlačítka "Zpět" dojde k obnovení předchozích nastavených hodnot. Naopak stiskem tlačítka "Uložit" zůstanou nastavené hodnoty uchovány.



Obr. 3.7: Konfigurace vstupního zařízení

3.2.5 Info

Zobrazí informace o simulátoru, případně lze stiskem tlačítka otevřít simulink model.

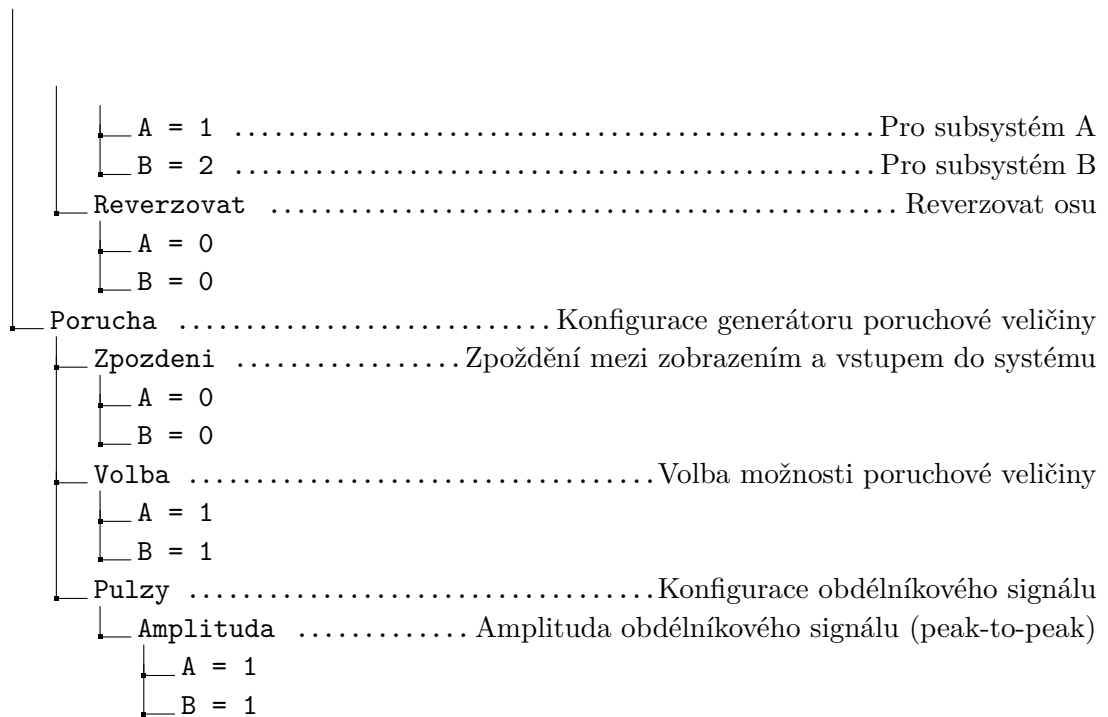
3.2.6 Struktura uložení konfiguračních proměnných a výchozí hodnoty

Konfigurační proměnné jsou uloženy jako vnořené struktury. Hierarchie uložení proměnných je znázorněna níže. Detailní popis struktury uložení konfiguračních proměnných včetně výchozích hodnot je přílohou A.

```

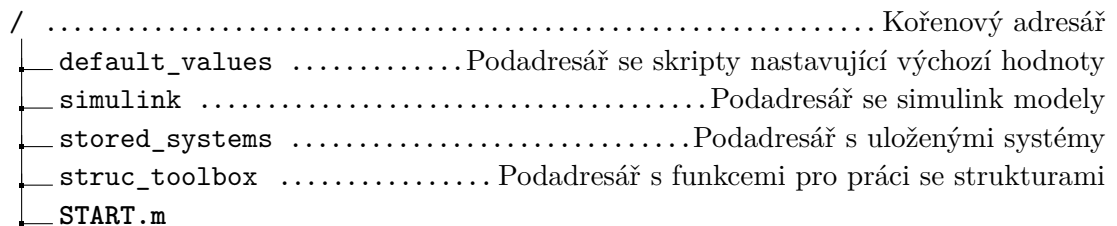
MATLAB base workspace
├── Joy .....Konfigurace vstupního zařízení
│   ├── ID = 1 (MS Windows) nebo 2 (GNU/Linux) ..... ID vstupního zařízení
│   └── Osa ..... Volba použité osy vstupního zařízení

```

3.3 Struktura podadresářů simulátoru

Pro větší přehlednost jsou soubory simulátoru roztříděny do jednotlivých níže uvedených podadresářů. Detailní popis adresářové struktury je jako příloha B.



3.4 Popis MATLAB skriptů

Kód MATLABu zajišťuje především nastavení parametrů pro následnou simulaci a řízení simulace prostřednictvím grafického rozhraní. Spuštění simulátoru se provede zavoláním skriptu `START.m` z kořenového adresáře.

Výpis 3.1: Kód skriptu `START.m`

```
1 %% Uzavření všech otevřených oken
2 close all
3 %% Vymazání všech proměnných
4 clear all
5
6 %% Přidání podadresářů do prohledávaných cest
7 addpath([pwd, '/simulink']);
8 addpath([pwd, '/struc_toolbox']);
9 addpath([pwd, '/GUI']);
10 addpath([pwd, '/default_values']);
11 addpath([pwd, '/stored_systems']);
12
13 %% Nastavení výchozích hodnot
14 run('Joy_SetDefault.m')
15 run('Porucha_SetDefault.m')
16 run('Regulator_SetDefault.m')
17 run('Soustava_SetDefault.m')
18 run('ZadanaHodnota_SetDefault.m')
19 run('Zobrazeni_SetDefault.m')
20 STOPsim = 0;
21
22 %% Spuštění GUI
23 run('HlavniMenu')
```

3.4.1 Skripty pro nastavení výchozích hodnot

Tyto skripty provádí prostou inicializaci proměnných na výchozí hodnoty. Výchozí hodnoty pro každou proměnnou jsou uvedeny v příloze A. Pro demonstraci je popsán skript 3.2 `Joy_SetDefault`, který inicializuje proměnné pro konfiguraci vstupního zařízení.

Výpis 3.2: Kód skriptu Joy_SetDefault.m

```

1
2 % Nastavení Joy.ID na základě aktuální platformy
3 if ispc
4     Joy.ID = 1; %Na PC s OS MS Windows lze předpokládat,
5         %že herní kontrolér bude připojen jako ID1
6         %(první připojený)
7 elseif isunix
8     Joy.ID = 2; %Na PC s OS GNU/Linux lze předpokládat
9         % že herní kontrolér bude připojen jako ID2
10        %(první připojený)
11 elseif ismac
12     Joy.ID = 2; % chování na počítačích MAC není ověřeno
13 else
14     error('Platform□not□supported')
15 end
16
17
18 %% Nastavení konfiguračních proměnných
19 % v rámci vnořených struktur
20 Joy.Osa.A = 1;      % Pro subsystém A
21 Joy.Osa.B = 2;      % Pro subsystém B
22
23 Joy.Reverzovat.A = 0;
24 Joy.Reverzovat.B = 0;
25
26 Joy.Filtrovat.A = 0;
27 Joy.Filtrovat.B = 0;
28
29 Joy.Filtr.NUM.A = 1;
30 Joy.Filtr.NUM.B = 1;
31 Joy.Filtr.DEN.A = [1e-5 0.5 1];
32 Joy.Filtr.DEN.B = [1e-5 0.5 1];

```

3.4.2 Grafické uživatelské rozhraní (GUI)

Program MATLAB umožňuje 2 přístupy pro tvorbu uživatelského rozhraní.

1. `guide` - průvodce pro tvorbu GUI - výsledkem je grafické rozhraní skládající se ze souboru grafiky (.fig) a funkční části (.m)
2. `appdesigner` - prostředí pro tvorbu uživatelských aplikací - výsledkem je MATLAB aplikace (.mlapp)

Prostředí `appdesigner` je v současné době vyvíjené moderní prostředí pro tvorbu aplikací, které v budoucnu plně nahradí `guide`. Umožňuje jednoduše vytvářet integrované aplikace, které lze sdílet jako jeden balík. Koneční uživatelé si poté mohou aplikaci nainstalovat do MATLABu na svém PC. [9]

Průvodce tvorbou GUI `guide` je starší způsob tvorby grafického rozhraní, který bude brzy odstraněn. Oproti `appdesigner` nabízí širší podporu, lepší stabilitu. Výsledné GUI vytvořené pomocí `guide` se skládá ze souborem popisující grafiku (.fig) a MATLAB funkce (.m), která zajišťuje funkcionalitu. Zavoláním této funkce dojde ke spuštění GUI. [9]

Já jsem po zvážení vlastností jednotlivých přístupů zvolil pro tvorbu GUI první přístup, tedy pomocí `guide`, a to zejména proto, že aplikace z prostředí `appdesigner` z důvodu aktuálního vývoje nejsou kompatibilní pro editaci mezi verzemi MATLABu a v současné době nabízí menší podporu od komunity.

GUI bude předvedeno na příkladu obrazovky konfigurace vstupu. Vynechány jsou některé automaticky generované části kódu. Aby bylo umožněno spouštět simulaci nezávisle na GUI, není prováděno předávání parametrů při volání a návratu z funkce.

Pro zajištění kompatibility se staršími verzemi Simulinku je model ukládán do souboru `main.slx` v poslední použité verzi a následně vyexportován do souboru `main_R2015B.slx` ve verzi pro Simulink 8.6 (MATLAB r2015b). Při otevírání modelu nebo spouštění simulace je využito výjimek, jak znázorňuje úsek kódu 3.5.

Výpis 3.3: Kód funkce KonfiguraceVstupu.m

```

1 PomStruc = evalin('base', 'Joy');
2 % Struktura "Joy" je přenesena z workspace base
3 % do workspace GUI jako "PomStruc"
4 ...
5 assignin('base', 'ZalohaJoy', PomStruc);
6 % Struktura "PomStruc" je zapsána
7 % do workspace base jako "ZalohaJoy"
8
9 set(handles.edit1, 'String', PomStruc.ID);
10 % Do editovatelného textového pole je zapsána
11 % hodnota reflektující skutečně nastavenou hodnotu
12 set(handles.popupmenu2, 'Value', PomStruc.Osa.A);
13 % Do rolovacího výběru je zapsána hodnota
14 % reflektující skutečně nastavenou hodnotu
15
16 ...
17
18 function pushbutton6_Callback(hObject, eventdata, handles)
19 % Akce po stisku tlačítka "Back"
20
21 evalin('base', 'Joy_□=□ZalohaJoy;');
22 % Ve workspace base je zkopírována struktura
23 % "ZalohaJoy" do "Joy"
24 evalin('base', 'clear_□ZalohaJoy');
25 % Proměnná "ZalohaJoy" ve workspace base je smazána
26 close(KonfiguraceVstupu)
27 % Okno KonfiguraceVstupu je uzavřeno
28
29 ...
30
31 function edit1_Callback(hObject, eventdata, handles)
32 % Akce po zapsání hodnoty do editovatelného pole

```

Výpis 3.4: Kód funkce KonfiguraceVstupu.m

```

1 PomStruc = evalin('base', 'Joy');
2 % Struktura "Joy" je přenesena z workspace base
3 % do workspace GUI jako "PomStruc"
4 PomStruc.ID = str2double(get(hObject,'String'));
5 % Zapsaná hodnota do textového pole je převedena
6 % z textového řetězce na číslo a zapsána
7 % do proměnné PomStruc.ID
8 assignin('base', 'Joy', PomStruc);
9 % Struktura "PomStruc" je zapsána do workspace
10 % base jako "Joy" s aktualizovanou hodnotou ID
11
12
13 function edit1_CreateFcn(hObject, eventdata, handles)
14
15 if ispc && isequal(get(hObject,'BackgroundColor'),...
16     get(0,'defaultUicontrolBackgroundColor'))
17     set(hObject,'BackgroundColor','white');
18 end
19
20 ...
21
22 function pushbutton7_Callback(hObject, eventdata, handles)
23 % Akce při stisku tlačítka "Uložit"
24
25 evalin('base', 'clear ZalohaJoy');
26 % Proměnná "ZalohaJoy" ve workspace base je smazána
27 close(KonfiguraceVstupu);
28 % Okno KonfiguraceVstupu je zavřeno

```

Výpis 3.5: Ukázka užití výjimky

```
1  try
2      sim('main.slx', 10);
3  catch ME
4      warning('main.slx is a newer version of the simulink
5  than it is installed!')
6      warning('Trying to open main_R2015B.slx')
7      sim('main_R2015B.slx', 10);
8  end
```

3.4.3 Struc toolbox

Výstupy ze simulátoru jsou uloženy ve formátu struktury s časem. Pro pohodlné zpracování naměřených výsledků jsem vytvořil struc toolbox s následujícími funkcemi pro zpracování takového formátu.

[OutputStruct1, OutputStruct2] = CutStruc(Structure, varargin)

[OutputStruct1, OutputStruct2] = CutStruc(Structure, tc)

Funkce rozřízne vstupní signál na dva (OutputStruct1 a OutputStruct2). Structure je struktura s časem a tc je časový okamžik řezu.

[OutputStruct1, OutputStruct2] = CutStruc(Structure, t1, t2)

Vyřízne ze signálu Structure okno ohraničené časovými okamžiky t1 a t2. Vyříznuté okno je vráceno jako OutputStruct1 a zbylý signál jako OutputStruct2.

[Delay] = DelayDet(Struc1, Struc2, varargin)

Funkce se volá DelayDet(Struc1, Struc2, Show). Argument Struc1 je první vstupní signál, Struc2 je druhý vstupní signál zpožděný na signál Struc1. Show je volitelný argument, pokud platí Show == 1, funkce zároveň vykreslí graf korelace vstupních signálů. Funkce určí časový posun při nejvyšší hodnotě korelace mezi signály Struc1 a Struc2 a vrátí ho jako Delay.

[] = PlotStruc(Structure, varargin)

Volání PlotStructure(Structure, Figure). Vykreslí graf struktury s časem Structure do aktuálního "figure" okna, přitom nepřepisuje stávající grafy. Volitelný argument Figure obsahuje číslo "figure" okna, které má být nastaveno jako aktuální. Pokud

`Figure == 0` vykresluje se do aktuálního okna, pokud `Figure == -1` otevře se nové "figure" okno.

[Aligned] = StrucAlign(Struc)

Zarovná signál `Struc` tak, že vychází z bodu 0, 0 a vrátí do `Aligned`. Vhodné například pro zarovnání vyříznutých časových oken.

[IAE] = StrucIAE(Structure)

Vypočítá integrál (sumu) absolutní hodnoty signálu `Structure`, neboli určí hodnotu usměrněného lineárního integrálního kritéria kvality regulace ze signálu regulační odchylky podle vztahu 1.6.

[InterpolatedStruc] = StrucInterp(Struc, Time)

Interpoluje strukturu s časem `Struc` na konstantní časový krok daný argumentem `Time`. Vhodné pro přepočítání struktury zaznamenané při dynamické velikosti časového kroku na konstantní vzorkovací periodu.

[ISE] = StrucISE(Structure)

Vypočítá integrál (sumu) druhé mocniny hodnoty signálu `Structure`, neboli určí hodnotu kvadratického integrálního kritéria kvality regulace ze signálu regulační odchylky podle vztahu 1.7.

[ITAE] = StrucITAE(Structure)

Vypočítá hodnotu ITAE kritéria kvality regulace ze signálu regulační odchylky `Structure` podle vztahu 1.8.

[ITSE] = StrucITSE(Structure)

Vypočítá hodnotu ITSE kritéria kvality regulace ze signálu regulační odchylky `Structure` dle vztahu 1.9.

StrucSteps(StrucW, StrucY, varargin)

Funkce vyhledá jednotkové skoky v signálu `StrucW`, na základě kterých nanormuje a vrátí úseky signálu `StrucY`. Funkce se volá `StrucSteps(StrucW, StrucY, t0, t1)`, kde `StrucW` je signál, ve kterém budou vyhledány skoky, `StrucY` je signál, ze kterého budou vyřezány úseky. Argumenty `t0` a `t1` jsou volitelné. Argument `t0`

určuje délku vyříznutí signálu před skokem (výchozí 1) a t_1 určuje délku vyříznutí signálu po skoku (výchozí 4). Vyříznuté úseky jsou vráceny jako pole struktur s časem `Steps`.

[Equalized] = StrucZOHTE(Struc1, Struc2)

Tvarovač nultého řádu signálu `Struc1`. Doplní vzorkovací okamžiky podle signálu `Struc2`. Využití například pro sjednocení vzorkovacích okamžiků signálu w při náhodných pulzech jako řídicí signál a řízené veličiny y .

ClearMeasured

Nejedná se o funkci, ale o skript, zobrazí signály w a y pro oba subsystémy a následně smaže všechna naměřená data.

3.5 Popis Simulink modelu

Simulink model sestává z univerzální části a dvou totožných částí reprezentující subsystemy A a B.

3.5.1 Univerzální část

Tvoří ji blok "RealTime pacer", blok "To Workspace", který přenáší proměnné do workspace base MATLABu, zároveň zajišťuje zobrazení, prostřednictvím MATLAB funkce Zobrazeni, a vstup z GUI. Blok "To Workspace" je zobrazen na obrázku C.1 a C.2. Dalším blokem univerzální části je blok "Vstup", ve kterém je modifikován signál ze vstupního zařízení. Přesněji je volena konkrétní osa herního zařízení, provedena nebo neprovedena reverzace (násobení -1) a filtrace. Blok "Vstup" je zobrazen na obrázku C.3.

3.5.2 Subsystemy A a B

Oba subsystemy jsou totožné, proto bude popis dále věnován pouze subsystemu A. Architektura subsystemu A je znázorněna na zjednodušeném schématu 3.8 případně na úplném schématu C.4 a odpovídá regulační smyčce viz 1.3.

Generátor žádané hodnoty/Generátor poruchové veličiny

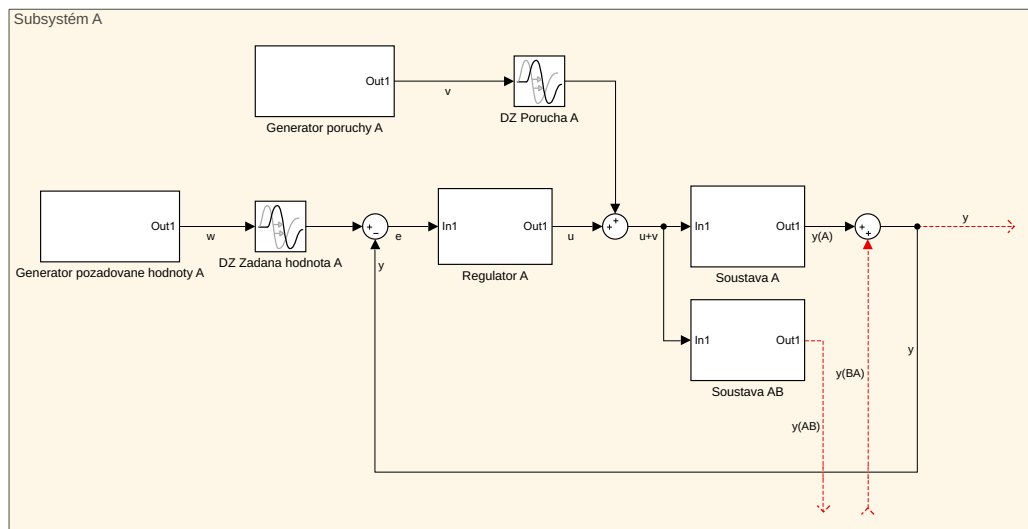
Architektura generátoru žádané hodnoty je shodná s architekturou generátoru poruchy viz C.5. V tomto bloku dojde k výběru jednoho z generovaných signálů.

Regulátor

V bloku regulátoru viz C.6 se přepíná zvolený regulátor. Výstup zvoleného regulátoru je pak omezen na rozsah od -1 do 1 a zesílen. Následuje blok "Omezení regulátoru" s nelinearitami C.7.

Soustava

Všechny soustavy (A, B, AB, BA) mají shodnou architekturu obsahující přenosovou funkci soustavy a dopravní zpoždění soustavy viz C.8.



Obr. 3.8: Zjednodušené schéma subsystému A

3.5.3 Virtuální světy

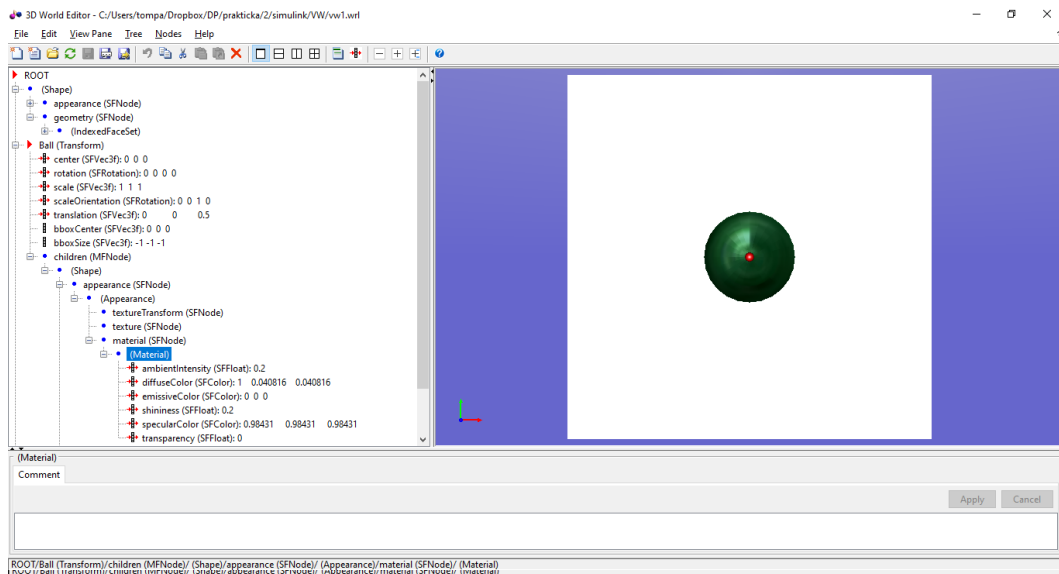
Pro co nejpřirozenější zobrazení byly vytvořeny tři virtuální světy.

- Virtual World 1 - znázorňující regulační odchylku pohybující se kuličkou
- Virtual World 2 - znázorňující požadovanou a skutečnou hodnotu pomocí pohybujících se kuliček
- Virtual World 3 - pohyb letounu

Virtuální světy jsou vytvořeny pomocí 3D World Editor u integrovaného přímo v MATLABu (pro editaci lze využít jakýkoli 3D editor podporující formát VRML nebo X3D [9]) s využitím již předpřipravených modelů. Každý virtuální svět má svůj blok v Simulink modelu, ve kterém dochází k přípravě dat pro virtuální svět a samotný blok VR Sink pro volání virtuálního světa.

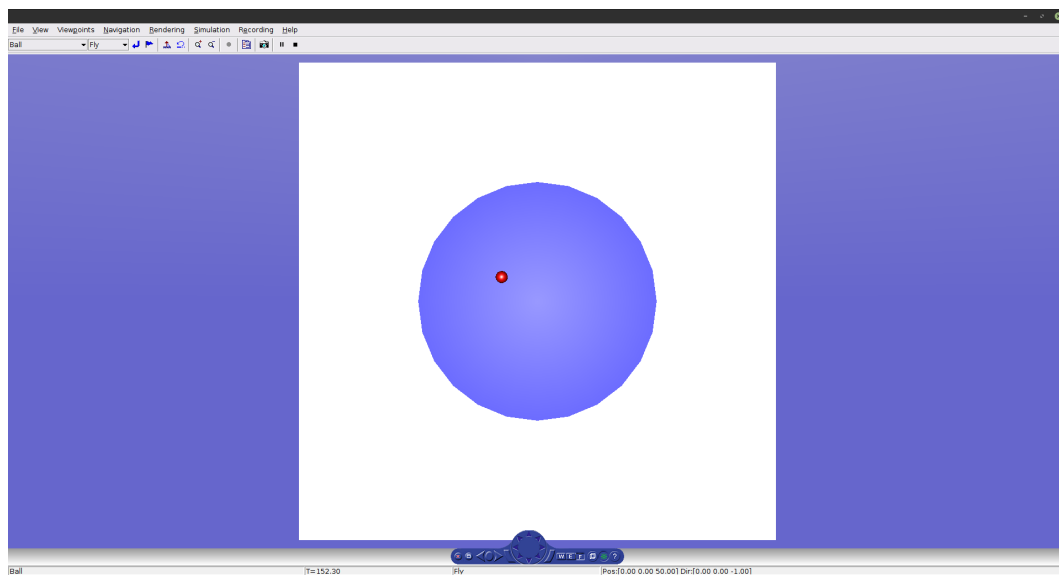
Virtual World 1

Virtuální svět 1 (Virtual World 1) zobrazuje regulační odchylku e prostřednictvím pohybující se červené kuličky. Cílem lidského operátora je působit svými akčními zásahy tak, aby červená kulička byla co nejbližší středu znázorněného zele-



Obr. 3.9: Okno 3D World Editoru, při editaci virtuálního světa 1

ným kuželem. Pohyb ve vodorovné ose je dán subsystémem A a pohyb ve svislé ose je dán subsystémem B.



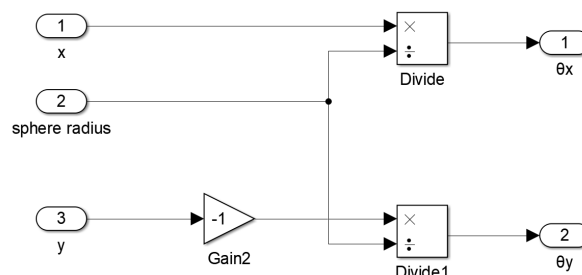
Obr. 3.10: Virtual World 1

Virtuální svět 1 se skládá z 5 prvků

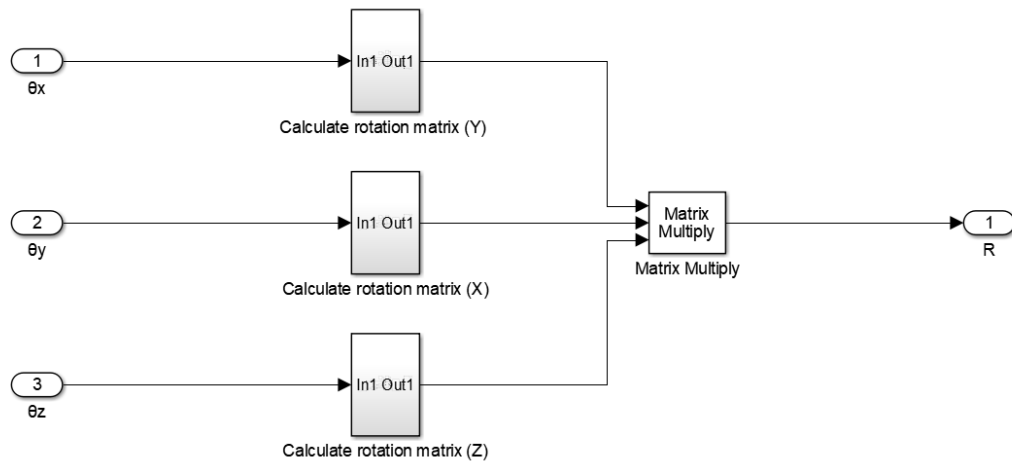
- Bílý podklad
- Červená kulička
- Viewpoint - nastavení pohledu

- Pozadí
- Kužel znázorňující střed - v rámci připomínek z ověřovacího měření byla změněna barva kuželu ze zelené na světle modrou

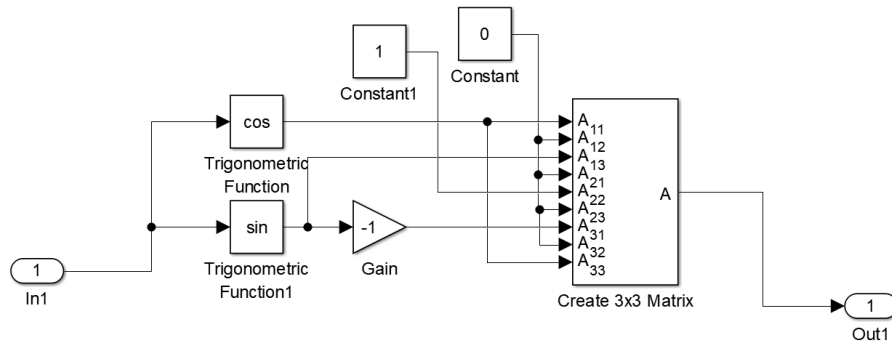
Blok virtuálního světa v Simulink modelu obsahuje řetězec přepočtu polohy kuličky na úhel jejího natočení, přepočtení úhlu natočení na matici rotace a výpočet vektoru SFRotation pro VR Sink (virtuální svět). Tento řetězec způsobuje realistické kutálení kuličky po povrchu. VR Signal Expander vytváří vektor translace SFVec3F pro VR Sink (virtuální svět).



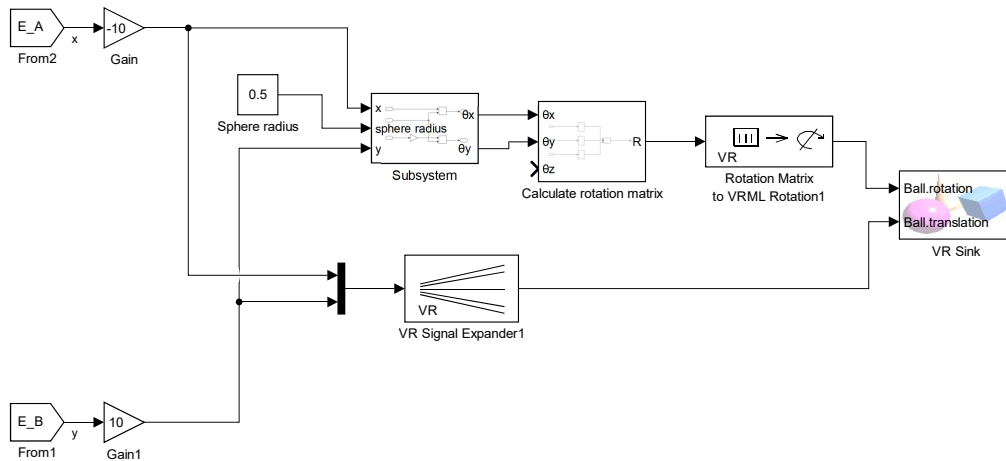
Obr. 3.11: Přepočtení polohy kuličky na úhel jejího natočení



Obr. 3.12: Výpočet matice rotace



Obr. 3.13: Výpočet matice rotace kolem osy Y



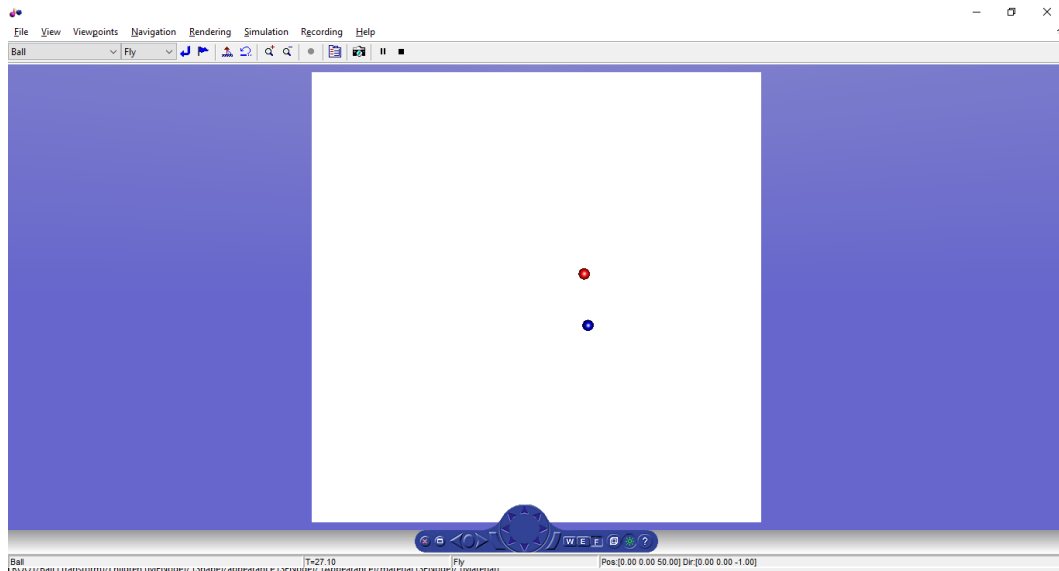
Obr. 3.14: Blok VW1

Virtual World 2

Virtuální svět 2 (Virtual World 2) zobrazuje žádanou hodnotu w prostřednictvím modré kuličky a skutečnou hodnotu regulované veličiny y prostřednictvím červené kuličky. Cílem lidského operátora je působit svými akčními zásahy tak, aby se červená kulička držela co nejtěsněji modré, v ideálním případě ji překrývala. Stejně jako ve virtuálním světě 1 je pohyb ve vodorovné ose dán subsystemem A a pohyb

ve svislé ose je dán subsystemem B.

Na základě připomínek z ověřovacího měření byla poloha červené kuličky v ose z posunuta a desetinu jednotek výše. Díky tomu má červená kulička vyšší prioritu při zobrazení a nedochází k nepříjemnému přeblikávání barev.

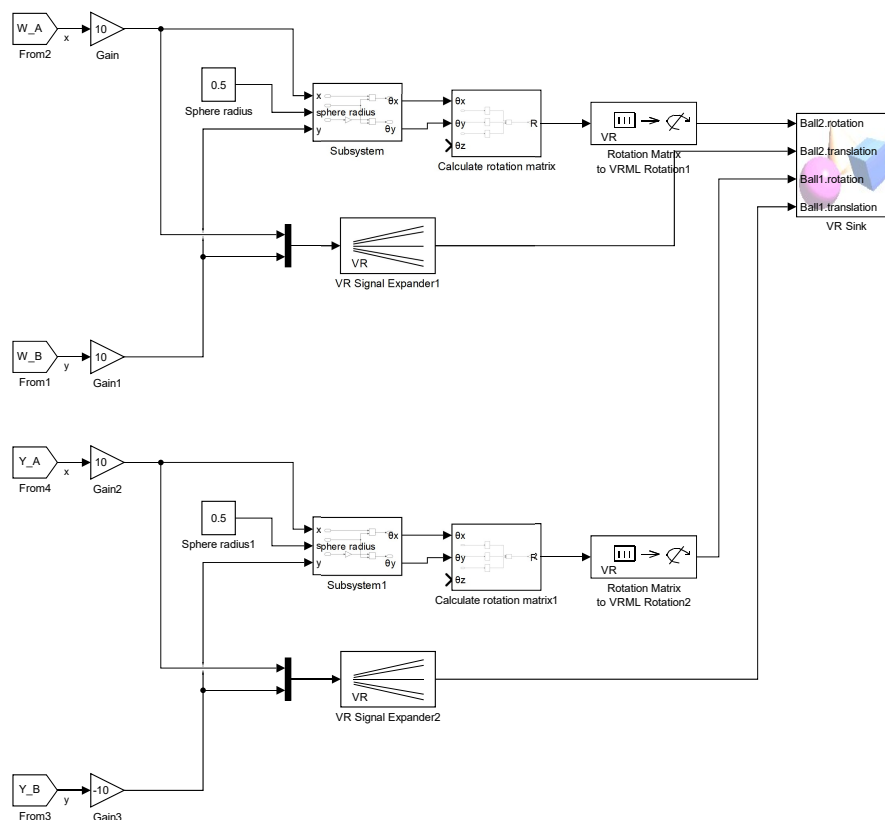


Obr. 3.15: Virtual World 2

Virtuální svět 2 se skládá z 5 prvků:

- Bílý podklad
- Modrá kulička
- Červená kulička
- Viewpoint - nastavení pohledu
- Pozadí

Podobně jako v bloku virtuálního světa 1 je i v bloku virtuálního světa 2 použit řetězec obstarávající realistické kutálení kuličky po povrchu. Tento řetězec je použit pro modrou i červenou kuličku. Pro vytvoření vektoru translace je opět pro obě kuličky použit VR Signal Expander.



Obr. 3.16: Blok VW2

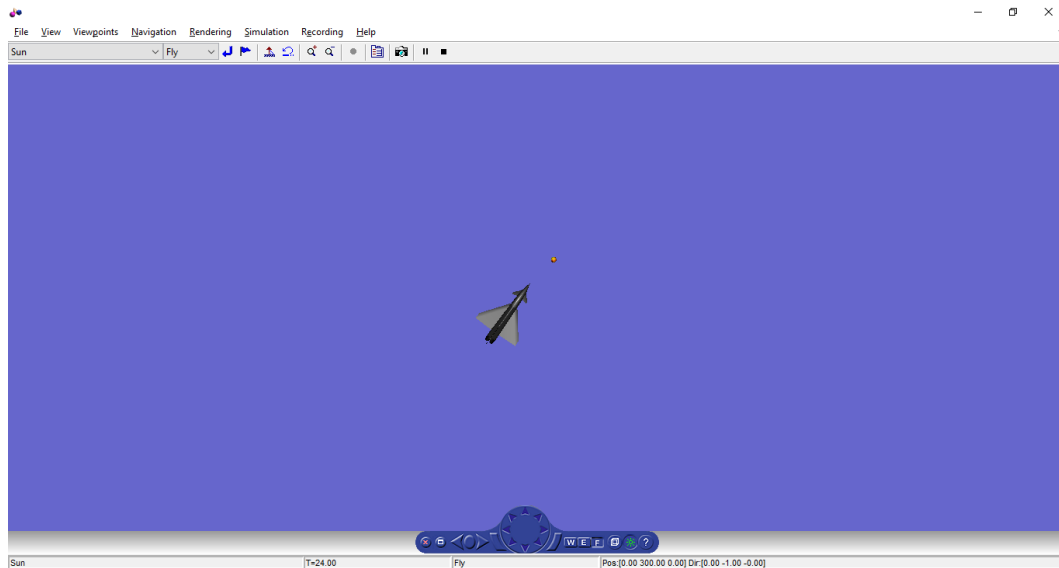
Virtual World 3

Ve virtuálním světě 3 (Virtual World 3) je cílem sledovat kutálející se kuličku letounem tak, aby vzdálenost byla co nejmenší. V ideálním případě by kulička měla být uprostřed, uvnitř letounu. Podsystem A určuje natočení letounu nebo směru kutálení kuličky, podsystem B pak rychlost. Protože úloha nezapadá do koncepce simulátoru, nelze vyhodnocovat chyby z regulačních smyček, ale do workspace MATLABu je do proměnné BallPos, respektive AircraftPos zapisována struktura s časem s informací o poloze kuličky, respektive letadla. Tyto výstupy již lze dále vyhodnocovat.

Protože nelze vyhodnocovat chybu z regulačních smyček, ta odpovídá rozdílu natočení kuličky a letadla nebo rozdílu rychlostí koule a letadla, není tedy směrodatná vzhledem k cíli úkolu, není možné použít matematické regulátory (PID, přenos) pro tuto úlohu. K dispozici jsou 4 pohledy (ze středu letadla, před letadlem, z koule směrem vzad a na scénu shora).

Virtuální svět 3 se skládá ze 7 prvků:

- Viewpoint - střed letadla
- Viewpoint - před letadlem
- Viewpoint - koule směrem vzad

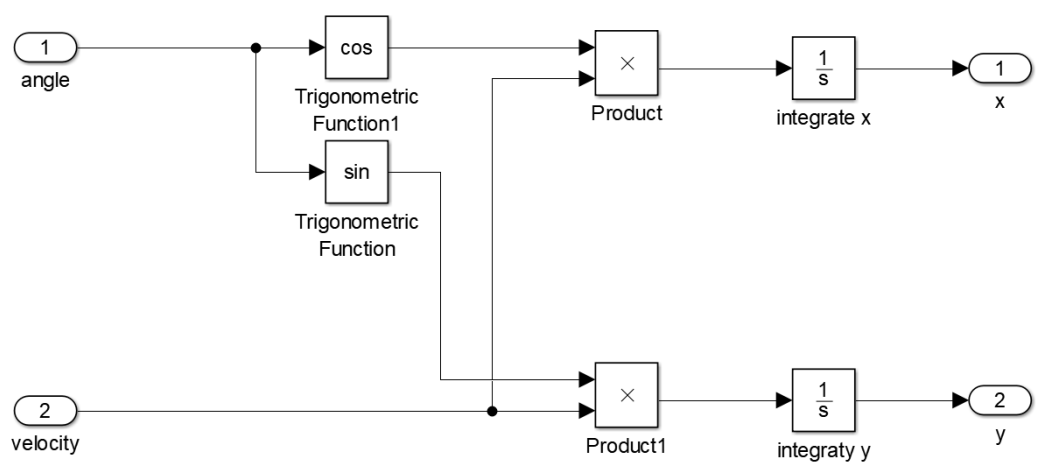


Obr. 3.17: Virtual World 3

- Viewpoint - shora
- Pozadí
- Kulička
- Letadlo

Informace o úhlu a rychlosti je přepočtena a integrována na polohu zadanou pomocí souřadnic x a y blokem `angleVelocity2xy`. Podobně jako v předchozích virtuálních světech je použit řetězec pro realizaci realistického kutálení kuličky a VR Signal Expander pro výpočet vektoru translace kuličky a letadla. Aktuální poloha kuličky a letadla je zapisována do workspace MATLABu bloky `To Workspace` jako struktura s časem do proměnných `BallPos` a `AircraftPos`. Kromě polohy a natočení letadla a kuličky jsou také podobným postupem počítány polohy a natočení Viewpointů.

Schéma bloku virtuálního světa 3 viz v příloze (C.9 a C.10).



Obr. 3.18: Přepočet úhlu a rychlosti na polohu xy

4 Základní měření a vyhodnocení

4.1 Popis měření

Za účelem ověření funkčnosti, použitelnosti a demonstrace užití jsem provedl základní ověřovací měření. Ověřovací měření trvalo pro každého operátora asi 45 minut a zahrnovalo 13 scénářů. Před měřením každého scénáře bylo operátorovi sděleno o jaký přenos a žádanou hodnotu se jedná. Měření každého scénáře pak trvalo minimálně 90 sekund, z toho 30 sekund bylo vyhrazeno pro seznámení se soustavou a přímo následovalo 60 sekund užitečného měření. Při testování byl jako analogový vstup použit volant s pedály Genius SpeedWheel 3 MT. Natočení volantu bylo vstupem subsystému A a stisk pedálu vstupem subsystému B.

Před začátkem měření bylo položeno testovaným operátorům několik otázek:

1. Váš věk?
2. Věnujete se gamingu (akčním hrám)? (1-5, 1 - nehraji akční videohry, 5 - hraji velmi často)
3. Jak často řídíte automobil? (1-5, 1 - nejsem řidič, 5 - řídím velmi často)
4. Máte znalosti z oboru teorie řízení?
5. Požil jste před testováním alkohol nebo jiné omamné látky?

Tyto otázky jsou namířeny na faktory, které dle mého předpokladu mohou ovlivňovat odezvu lidského operátora. Odpovědi operátorů na tyto otázky jsou v tabulce 4.1.

Operátor Otázka	1	2	3	4	5	6	7
1 - věk	23	22	24	25	21	20	24
2 - gaming	2	5	2	3	1	1	4
3 - řízení automobilu	4	5	3	5	2	3	4
4 - znalost teorie řízení	ANO	NE	ANO	ANO	NE	NE	ANO
5 - alkohol	NE	NE	NE	NE	NE	NE	NE

Tab. 4.1: Odpovědi testovaných operátorů na položené otázky

4.1.1 Testovací scénáře

Pro měření bylo zvoleno 13 testovacích scénářů zobrazených v tabulce 4.2. Zvolené přenosy soustavy byly inspirovány na základě reálných systémů viz 1.1

Scénář	Přenos soustavy	Zobrazení	Žádaná hodnota	Vstup
1	1	VW2	náhodné pulzy	volant
2	1	VW2	náhodné pulzy	pedály
3	1	VW2	náhodné pulzy	pedály
4	1 (A i B)	VW2	náhodné pulzy	volant a pedály
5	1	VW2	superponované sinusoidy	volant
6	$\frac{1}{p}$	VW2	náhodné pulzy	volant
7	$\frac{1}{p}$	VW2	superponované sinusoidy	volant
8	$\frac{1}{p^2}$	VW2	náhodné pulzy	volant
9	$\frac{1}{p^2}$	VW2	superponované sinusoidy	volant
10	$\frac{1}{p(2p+1)}$	VW2	náhodné pulzy	volant
11	$\frac{1}{p(2p+1)}$	VW2	superponované sinusoidy	volant
12	$\frac{1}{p(2p+1)}$	VW1	superponované sinusoidy	volant
13	$\frac{1}{p(2p+1)}$	scope	superponované sinusoidy	volant

Tab. 4.2: Testovací scénáře

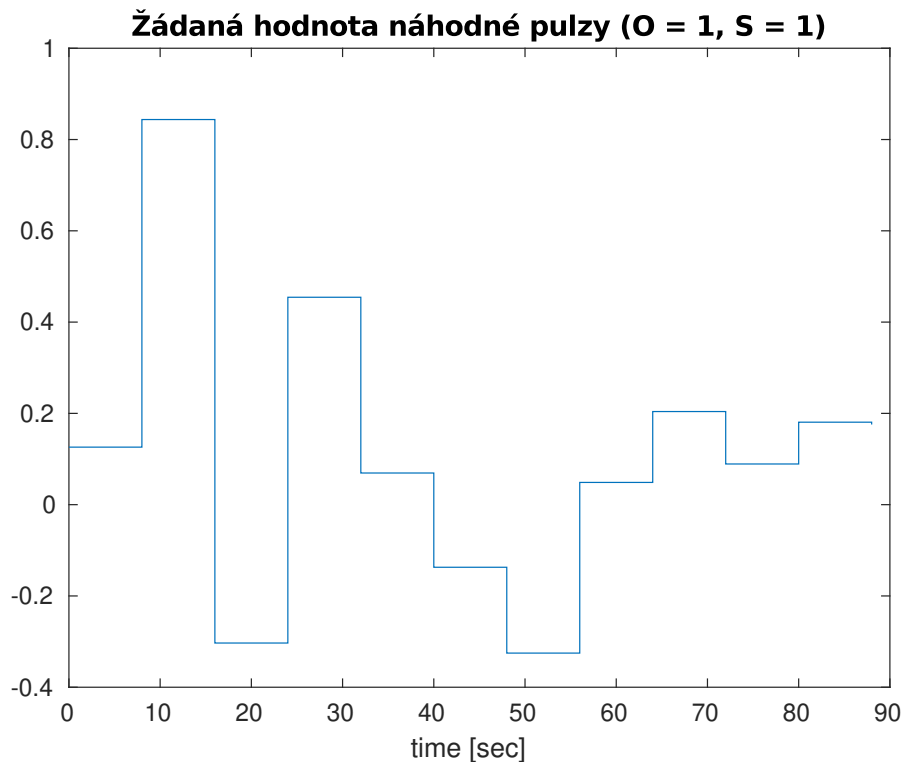
Cílem těchto scénářů je porovnat chování lidského operátora při řízení různých soustav, dále pak porovnat vliv vstupního zařízení a různé citlivosti levé a pravé nohy při stisknutí pedálu (scénáře 1, 2, 3). Schopnost člověka soustředit se na dvě úlohy najednou (scénáře 1, 2, 3, 4). Vliv způsobu zobrazení (scénáře 11, 12, 13).

Jako žádanou hodnotu jsem zvolil náhodné pulzy, které představují jednotkový skok, po normalizaci lze tedy porovnat přechodové charakteristiky regulace. Vzorovací perioda generátoru náhodných pulzů byla 8 s pro subsystém A a 12 s pro subsystém B. Periody vzorkování byly voleny tak, aby nabídly lidskému operátorovi dostatek času pro ustálení po regulačním ději a také, aby docházelo ke změnám žádané hodnoty v jednom subsystému, ale také v obou subsystémech zároveň.

Druhým signálem žádané hodnoty byla superpozice sinusoid o amplitudě 0.5 a frekvenci 0.02 Hz a 0,037 Hz pro subsystém A a 0,013 Hz a 0,045 Hz pro subsystém B. Frekvence jsem volil tak, aby odpovídaly předpokládaným regulačním schopnostem člověka a výsledný signál byl aperiodický. Pro subsystém A a subsystém B jsem volil rozdílné signály, aby výsledná žádaná hodnota nevykreslovala tvar Lissajousových obrazců.

Při scénáři 2, kdy vstupním zařízením byly pedály, nebyli operátoři upozorněni, kterou nohou mají obsluhovat levý pedál, ale všichni operátoři zvolili levou nohu. Při scénáři 3 jsem operátory vyzval, aby pro obsluhu levého pedálu použili druhou nohu, než v předchozím scénáři (pravou). Pro čtvrtý scénář si operátoři mohli zvolit, kterou nohu pro obsluhu levého pedálu použijí. Všichni operátoři kromě 7. operátora

zvolili levou nohu jako ve scénáři 2. Operátor 7 zvolil pravou, jako ve scénáři 3.



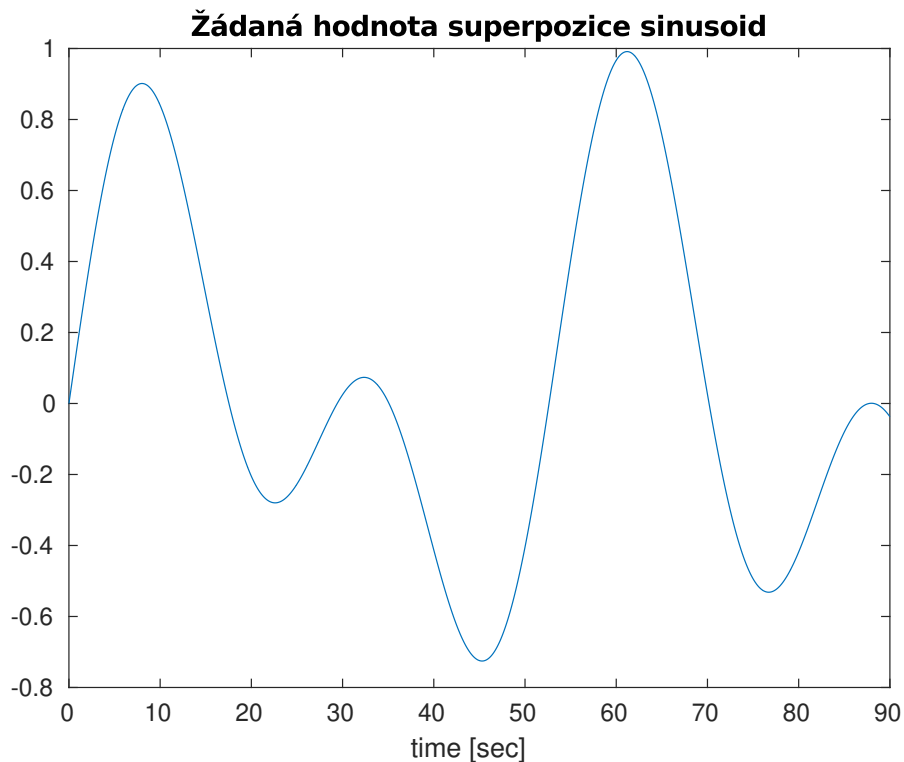
Obr. 4.1: Příklad žádané hodnoty náhodných pulzů

4.1.2 Vstupní zařízení

Jako vstupní zařízení byl použit volant s pedály Genius SpeedWheel 3 MT. Volant lze otáčet v rozsahu -180° až $+180^\circ$, přičemž je operátorovi kladen odpor pružinami vracující volant do nulové polohy. Výstupní signál je převáděn 8 bitovým analogově digitálním převodníkem. V okolí nulové polohy je znatelné pásmo necitlivosti. Pedály jsou 2, výstupní signál je opět 8 bitový. Levý pedál má vyšší prioritu (při stisku obou pedálů najednou je zjistitelná pouze poloha levého pedálu). U obou pedálů je v okolí nulové polohy pásmo necitlivosti a v okolí koncové polohy výstupní signál saturuje.

4.2 Zpracování dat

Z naměřených signálů jsem pomocí funkce `CutStruc` vyřízl užitečný časový úsek, signál v čase $t = 30$ s až $t = 90$ s. Scénáře, kde jako žádaná hodnota figurují náhodné pulzy jsem zpracoval funkcí `StrucSteps`.



Obr. 4.2: Žádaná hodnota superpozice sinusoid

Při zpracování dat jsem zjistil, že pro scénář 2 operátora 6 chybí naměřená data.

4.2.1 Vyhodnocení kvality regulace

Pro scénáře využívající jako žádanou hodnotu náhodné pulzy byly posouzeny kritéria kvality regulace v časové oblasti.

Překmit

Operátor Scénář	1	2	3	4	5	6	7
1	113 369	108 143	112 217	153 4 260	114 335	85 685	101 136
2	139 3 354	127 1 434	236 11 653	234 32 856	130 518	- -	180 4 700
3	94 570	240 16 604	49 4 000	148 4 346	96 3 371	136 528	127 1 433
4A	82 2 085	106 84	95 124	95 400	109 700	96 126	111 78
4B	157 3 352	136 359	136 1 906	148 4 346	213 33 774	165 5 192	100 27
6	110 103	100 17	100 122	110 70	103 40	96 39	103 180
8	166 2 220	104 445	151 2 638	150 1 495	208 30 200	79 3 180	283 20 707
10	115 153	98 52	90 156	115 112	69 239	77 867	115 773

Tab. 4.3: Průměrný překmit v % | rozptyl

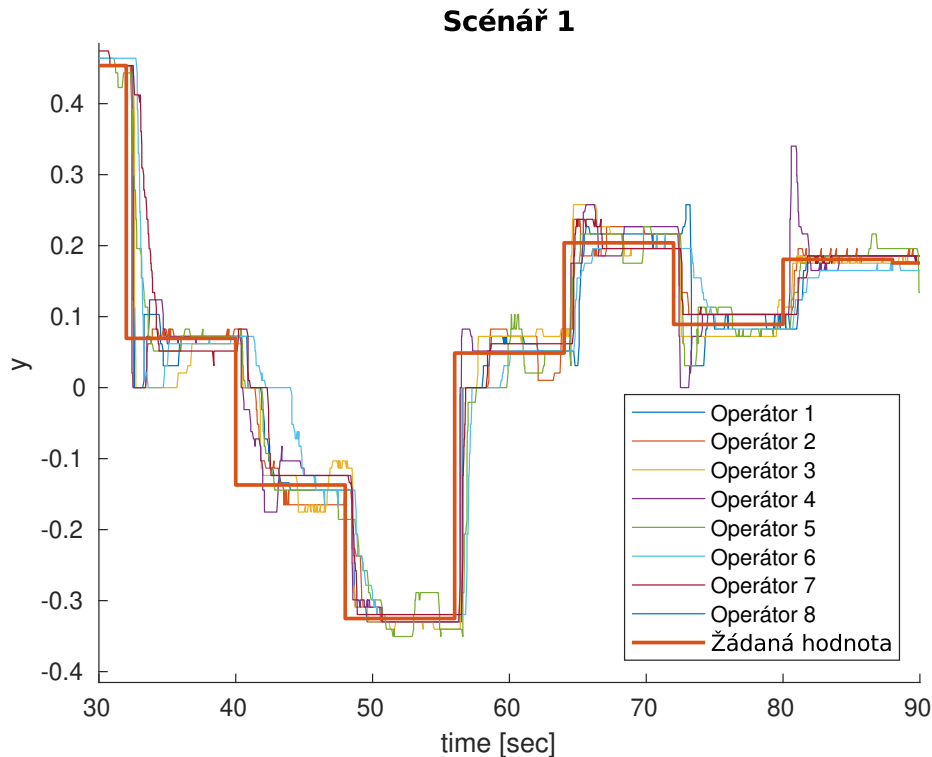


Obr. 4.3: Volant Genius SpeedWheel 3 MT [12]

Z tabulky 4.3 je patrné, že nejlépe se operátorům dařilo regulovat soustavu ze scénáře 6 s přenosem $F_S = \frac{1}{p}$, tedy čistý integrátor. Naopak nejhůře dopadla regulace dvojitého integrátoru ze scénáře 8, $F_S = \frac{1}{p^2}$. Na základě porovnání rozptylu lze posoudit, že operátoři 2 a 7 podávali nejstabilnější výsledky. Také je patrné, že při řízení pedály jsou rozptyly řádově větší, operátorům se systém hůře řídí, než při řízení stejného systému volantem.

Integrální kritéria

Kritéria, která vážují chybu časem jsem vyhodnotil pouze pro scénáře, u kterých byla žádaná hodnota náhodné pulzy. Pomocí funkce `StrucSteps` byly nalezeny skoky, na které jsem aplikoval ITAE a ISAE kritérium. Výsledná hodnota je součet hodnot



Obr. 4.4: Užitečná část signálu, scénář 1

pro jednotlivé skoky v signálu.

Pro další zpracování jsem za účelem zachování co největší porovnatelnosti chybějící hodnotu pro operátora 6 scénář 2 nahradil průměrnou hodnotou ostatních operátorů pro tento scénář.

Z porovnání scénářů podle integrálních kritérií v tabulce 4.8 je jasně patrné, že regulace pomocí pedálů, konkrétně scénáře 2, 3, 4B je pro lidského operátora značně náročnější a regulace méně kvalitní. Je také patrné zhoršení v případě, kdy se má operátor soustředit na dvě úlohy zároveň. Hodnota kritérií 4A a 4B je i více než dvojnásobná oproti scénáři 1 a 2. Zajímavé je porovnání scénářů 2 a 3, kdy pro scénář 2 je větší hodnota ISE než IAE a pro scénář 3 je hodnota kritérií opačná. Může to znamenat, že operátoři při obsluze levého pedálu pravou nohou působí citlivěji.

Při porovnání různých způsobů zobrazení (scénáře 11, 12, 13) je jeho vliv pozorovatelný. Nejvíce operátorům vyhovuje zobrazení VW1, kdy je jim zobrazována kulička, jejíž poloha odpovídá regulační chybě. V případě zobrazení Scope je jasné podstatné zhoršení regulace oproti zobrazení ve virtuální realitě. To může být způsobeno nižším rozlišením při zobrazení a při zobrazení více signálů (w , e , y) mají operátoři tendenci zaměňovat řízený a ostatní signály.

Operátor Scénář	1	2	3	4	5	6	7
1	48	46	56	48	92	99	90
2	139	123	167	158	167	–	159
3	175	134	232	129	167	175	183
4A	215	90	168	117	126	164	165
4B	365	171	355	218	468	225	359
5	111	92	179	164	117	122	94
6	138	81	135	84	125	159	90
7	85	78	93	78	90	140	76
8	299	232	212	292	769	383	554
9	195	113	312	185	908	193	232
10	258	207	287	151	228	215	273
11	192	107	130	160	147	116	157
12	125	79	90	67	172	152	135
13	206	157	233	186	372	240	280

Tab. 4.4: Integrál absolutní chyby (IAE)

Operátor Scénář	1	2	3	4	5	6	7
1	6	4	6	5	14	16	16
2	42	24	64	33	36	–	48
3	41	29	76	26	39	45	61
4A	38	10	25	15	19	24	24
4B	89	31	90	56	183	53	109
5	6	5	34	12	8	7	5
6	27	18	33	18	29	34	17
7	6	4	7	5	5	19	4
8	65	34	35	55	391	153	124
9	25	8	76	26	694	28	37
10	48	48	60	26	50	49	63
11	18	8	9	19	12	7	15
12	6	3	3	3	12	12	8
13	27	13	26	25	62	38	49

Tab. 4.5: Integrál čtverce chyby (ISE)

Operátor Scénář	1	2	3	4	5	6	7
1	534	445	394	582	549	758	472
2	7 587	6 841	8 967	9 670	7 803	–	8 151
3	9 704	7 726	12 390	6 483	9 393	8 370	8 671
4A	1 958	898	1 486	1 010	982	1 542	1 203
4B	18 380	10 090	20 320	11 490	23 460	11 950	20 210
6	1 080	898	1 486	491	863	943	652
8	1 956	2 227	1 722	1 779	3 043	1 834	5 336
10	1 778	1 327	2 197	1 191	1 952	1 759	1 915

Tab. 4.6: ITAE kritérium

Operátor Scénář	1	2	3	4	5	6	7
1	359	185	228	349	270	460	232
2	2 314	1 527	3 283	3 188	1 673	–	2 399
3	2 991	2 389	4 134	1 522	2 498	2 193	2 718
4A	1 720	577	1 070	546	545	1 050	832
4B	4 974	2 161	5 610	3 802	11 540	3 152	7 457
6	665	330	802	349	660	623	438
8	1 727	1 650	1 128	1 421	4 201	1 902	11 190
10	1 416	962	1 601	873	1 948	1 437	1 417

Tab. 4.7: ITSE kritérium

Při vstupním signálu superponovaných sinusoid se z řízených soustav operátorům dařilo nejlépe řídit soustavu ze scénáře 7 s přenosem $F_S = \frac{1}{p}$, hůře si vedli při řízení soustavy ze scénáře 1 s přenosem $F_S = 1$. Ještě hůře si vedli při řízení soustavy ze scénáře 11 ($\frac{1}{p(2p+1)}$) a nejhůře s velkým odstupem dopadlo řízení soustavy ze scénáře 9 ($\frac{1}{p^2}$).

V tabulce 4.9 je vypočtena průměrná hodnota jednotlivých integrálních kritérií pro scénáře, ve kterých operátor řídil volantem (1, 4A, 5 až 13). Nejlepším operátorem byl operátor 2, který se často věnuje hraní videoher. Nejhůře si vedl operátor 5. Pořadí operátorů podle ISE kritéria je v tabulce 4.10. Parametr ztráta vyjadřuje, o kolik procent je operátor horší než první operátor v pořadí.

Tabulka 4.11 obsahuje porovnání průměrné hodnoty jednotlivých integrálních kritérií pro scénáře, ve kterých operátor řídil pedály (2, 3 a 4B). Nejlépe si opět vedl operátor 2, naopak nejhůře si na rozdíl od předchozího případu vedl operátor 5. V tabulce 4.12 je opět zobrazeno pořadí operátorů.

Kritérium Scénář	IAE	ISE	ITAE	ITSE
1	68,4	9,6	533,4	246,2
2	645,6	354,1	8 170	2 367
3	675,9	345,6	8 963	2 635
4A	149,3	22,1	7 871	905,6
4B	1 008	521,1	16 558	3 152
5	125,6	11,0	–	–
6	116	25,1	795,6	522,4
7	91,4	7,1	–	–
8	391,6	122,4	2 556	3 317
9	305,4	127,7	–	–
10	231,3	49,1	1 731	1 379
11	144,1	12,6	–	–
12	117,1	6,7	–	–
13	239,1	34,3	–	–

Tab. 4.8: Porovnání scénářů podle integrálních kritérií

Kritérium Operátor	IAE	ISE
1	144,0	20,9
2	98,6	11,9
3	145,8	24,2
4	117,8	16,1
5	242,0	99,7
6	160,2	31,0
7	165,1	27,8

Tab. 4.9: Porovnání operátorů podle integrálních kritérií při řízení volantem

Střední dopravní zpoždění

Dopravní zpoždění pro scénáře 1 až 4, 6, 8, 10 jsem odečetl přímo z grafu. Pro určení dopravního zpoždění ve scénářích 5, 7, 9, 11 až 13 byla užita korelační analýza implementována funkcí `DelayDet` mezi žádanou hodnotou w a výslednou hodnotou y .

Z tabulky 4.13 je patrné, že pro scénáře 5, 7, 9, 11 až 13 je dopravní zpoždění velmi nízké a pro scénář 9 vychází dopravní zpoždění dokonce záporné. Tento jev je zapříčiněn způsobem řízení, kdy operátoři měli tendenci předbíhat žádanou hodnotu,

Pořadí	Operátor	ISE	Ztráta [%]
1.	2	11,9	0
2.	4	16,1	35
3.	1	20,9	75
4.	3	24,2	103
5.	7	27,8	134
6.	6	31,0	160
7.	5	99,7	736

Tab. 4.10: Pořadí operátorů podle nejnižší hodnoty ISE kritéria (volant)

Kritérium Operátor	IAE	ISE
1	226,3	57,3
2	142,7	28,3
3	251,4	76,7
4	168,6	38,5
5	267,3	86,1
6	183,9	46,4
7	233,4	72,7

Tab. 4.11: Porovnání operátorů podle integrálních kritérií při řízení pedály

Pořadí	Operátor	ISE	Ztráta [%]
1.	2	28,3	0
2.	4	38,5	36,1
3.	6	46,4	64,1
4.	1	57,3	102,7
5.	7	72,7	157,0
6.	3	76,7	171,4
7.	5	86,1	204,5

Tab. 4.12: Pořadí operátorů podle nejnižší hodnoty ISE kritéria (pedály)

na základě odhadu jejího budoucího vývoje. Příklad průběhu signálu pro operátora 7 a scénář 9, kdy je předbíhání žádané hodnoty nejvíce patrné je na obrázku 4.5.

Operátor Scénář	1	2	3	4	5	6	7
1	0,68	0,45	0,51	0,47	0,59	1,12	0,62
2	0,88	0,83	0,54	0,85	0,64	–	1,10
3	1,03	0,33	1,14	0,47	0,47	0,50	0,50
4A	1,60	0,65	0,88	0,67	0,70	1,08	0,89
4B	0,73	0,77	0,79	0,78	1,01	0,72	0,98
5	1,00	0,40	0,10	1,00	0,60	1,30	0,90
6	1,05	0,64	0,89	0,67	0,89	0,96	0,75
7	0,00	0,10	0,00	0,40	0,20	0,00	0,00
8	1,12	0,65	0,78	0,92	1,63	1,04	0,70
9	-0,20	0,00	-0,15	-0,40	0,30	-0,10	-0,60
10	0,88	0,75	0,75	0,80	2,12	1,08	0,91
11	0,00	0,30	0,20	0,80	0,60	0,08	0,00
12	0,50	0,00	0,30	0,30	1,40	0,30	0,30
13	0,02	0,50	0,30	0,70	0,20	0,70	1,30

Tab. 4.13: Střední dopravní zpoždění pro jednotlivé operátory a scénáře [s]

4.3 Identifikace parametrů modelu lidského operátora

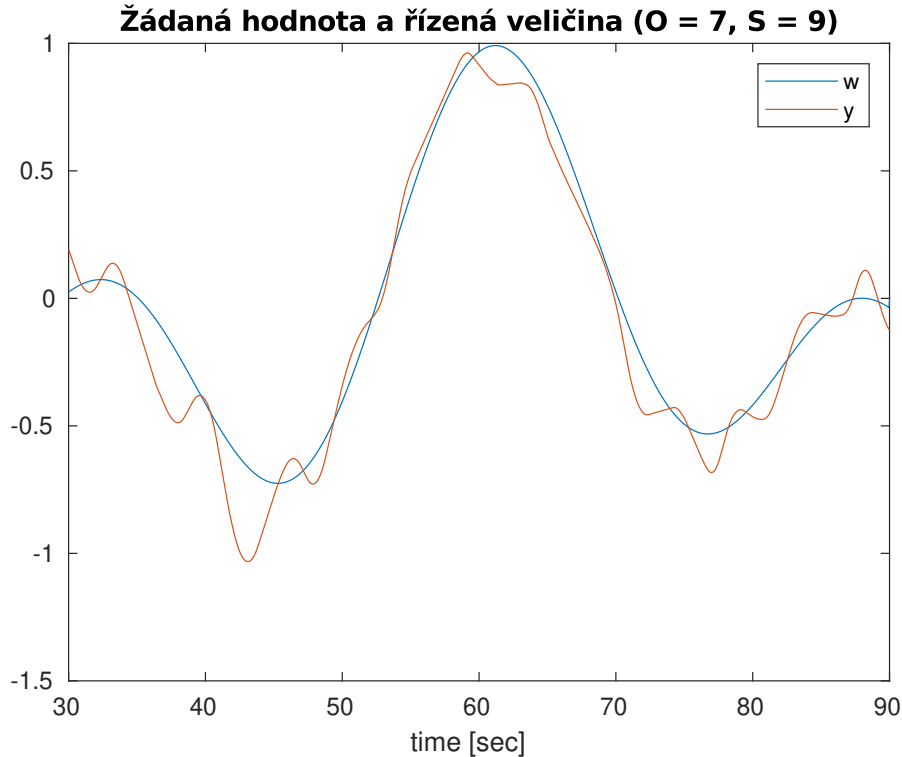
Pro identifikaci jsem zvolil precision model 1.4, který by měl ze zmíněných modelů nejdokonaleji odpovídat odezvě lidského operátora. Parametry modelu jsem identifikoval pomocí MATLAB nástroje `ident`.

Tvar identifikovaného precision modelu:

$$F_R(p) = K_R \frac{T_D p + 1}{(T_I p + 1)(T_N p + 1)} e^{-\tau_D p} \quad (4.1)$$

Model jsem identifikoval pro scénář 1 na základě přenosu otevřené smyčky. Žádanou hodnotou w byly náhodné pulzy, přenos soustavy $F_S = 1$. Zaznamenaná hodnota řídicí veličiny w byla doplněna pro všechny vzorkovací okamžiky řízené veličiny y funkcí `StrucZOHE` a oba zaznamenané průběhy byly převedeny prostřednictvím interpolační funkce `StructInterp` na shodnou konstantní vzorkovací periodu $T_S = 0.01$ s. Zesílení bylo nastavené jako konstantní o velikosti 1, dopravní zpoždění τ_D bylo dosazeno z tabulky 4.13. Model jsem identifikoval jako procesní se dvěma póly, jednou nulou a dopravním zpožděním

Identifikovaný model operátora 1 při řešení scénáře 1:



Obr. 4.5: Žádaná hodnota a řízená veličina ($O = 7, S = 9$)

$$F_R(p) = 1 \frac{0,66p + 1}{(1,38p + 1) + (0,02p + 1)} e^{-0,68p} \quad (4.2)$$

Typické hodnoty parametrů modelu 4.2 jsou: [4]

- K Závisí na aplikaci a poměru výstupního a vstupního signálu, reprezentuje operátorovy zvyklosti.
- T_N Typicky 0,05 až 0,2 s, udává setrvačné zpoždění neuromuskulárního systému operátora.
- T_I Typicky 0,1 až jednotky sekund. Setrvačná časová konstanta, souvisí s prováděním rutin a stereotypů.
- T_D Typicky 0,2 až jednotky sekund. Prediktivní časová konstanta, souvisí se schopností předvídat budoucí vývoj.
- τ_D Typicky 0,5 až 1 s. Dopravní zpoždění nervového systému.

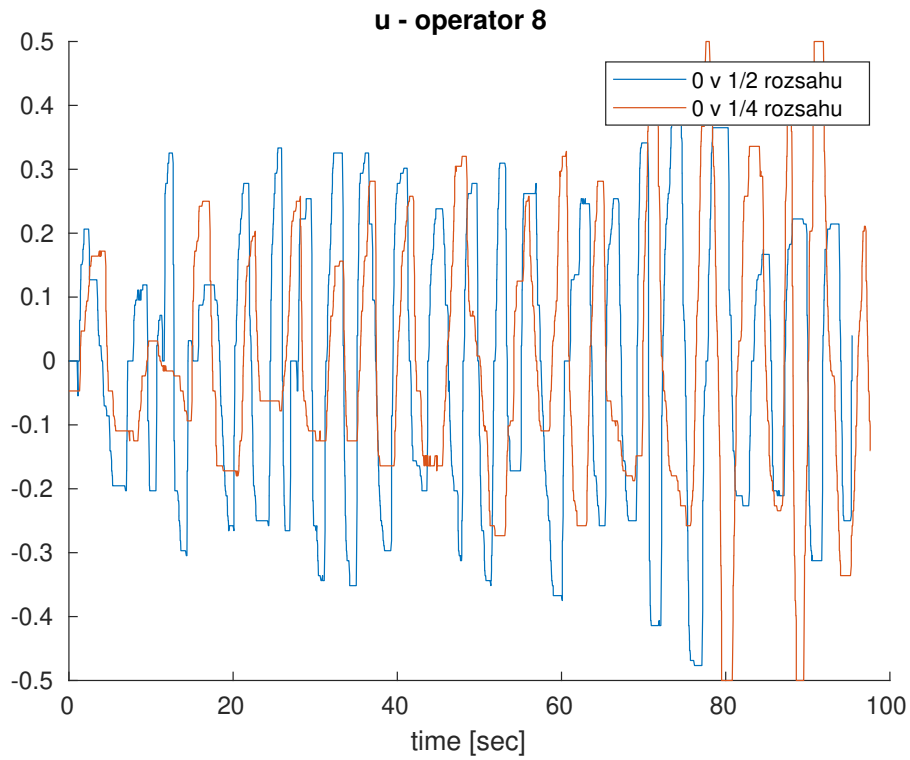
Identifikované parametry modelu pro ostatní operátory jsou uvedeny v tabulce 4.14.

Operátor	K_R	T_I	T_N	T_D	τ_D	FIT [%]
1	1	1,38	0,02	0,66	0,68	69,5
2	1	1,19	1,15	0,49	0,45	65,8
3	1	1,84	1,59	3,03	0,51	65,7
4	1	1,34	0,38	1,26	0,47	71,2
5	1	1,54	1,53	0,89	0,59	68,7
6	1	1,33	1,13	1,24	1,12	71,0
7	1	1,13	1,13	1,92	0,62	76,2

Tab. 4.14: Identifikované parametry precision modelu operátorů pro scénář 1

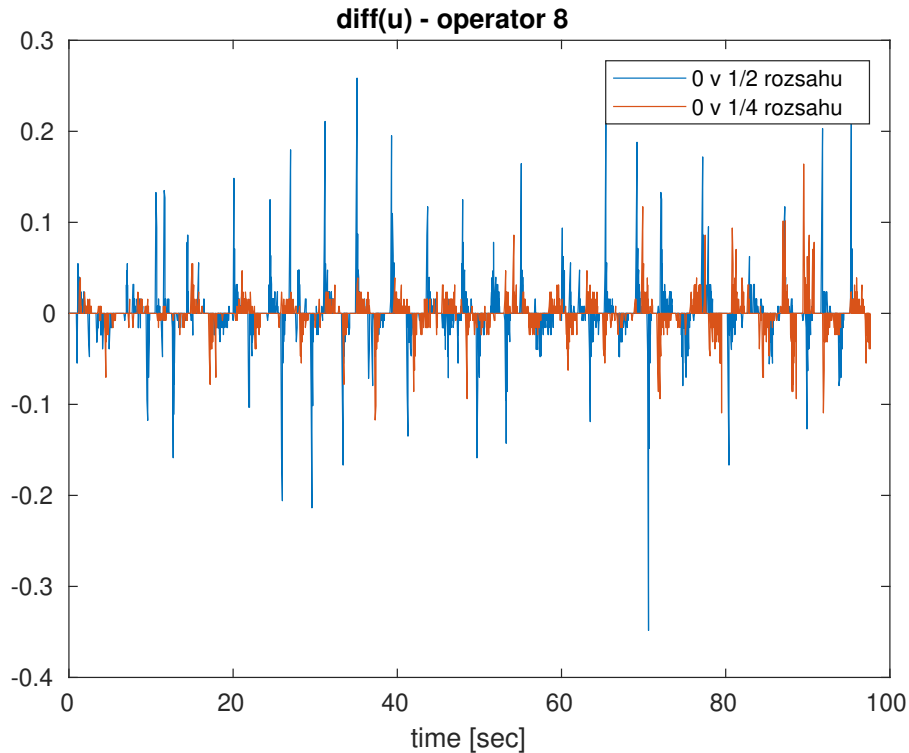
Identifikované parametry T_I a T_D operátorů odpovídají výše zmíněnému typickému rozsahu. Parametr T_N vychází pro většinu operátorů větší než je jeho typická hodnota. Může to být způsobené zahrnutím dynamiky vstupního zařízení (volantu) do modelu. Dopravní zpoždění τ_D pro operátory 2 a 4 je mírně nižší než typická hodnota je pravděpodobně způsobeno silným soustředěním na řídicí úlohu a lze předpokládat, že při delším testovacím scénáři by τ_D vzrostlo do typického rozsahu. Naopak pro operátora 6 je hodnota dopravního zpoždění τ_D vyšší. Myslím si, že je to zapříčiněné delším rozhodováním o vhodném akčním zásahu.

Pro scénáře 6, 8 a 10 se nepodařilo model identifikovat, protože, jak je patrné například z grafu akčního zásahu u pro scénář 8 operátora 1 (4.6), lidský operátor se při těchto scénářích uchyluje spíše k bang-bang regulaci a osciluje řízenou veličinou y v okolí žádané hodnoty w .



Obr. 4.7: Akční zásahy operátora 8 při řízení systému $F_S = \frac{1}{p^2}$

Tento jev je ještě lépe patrný na grafu derivace akčního zásahu operátora (4.8, který vypovídá o plynulosti operátorových zásahů.



Obr. 4.8: Derivace akčních zásahů operátora 8 při řízení systému $F_S = \frac{1}{p^2}$

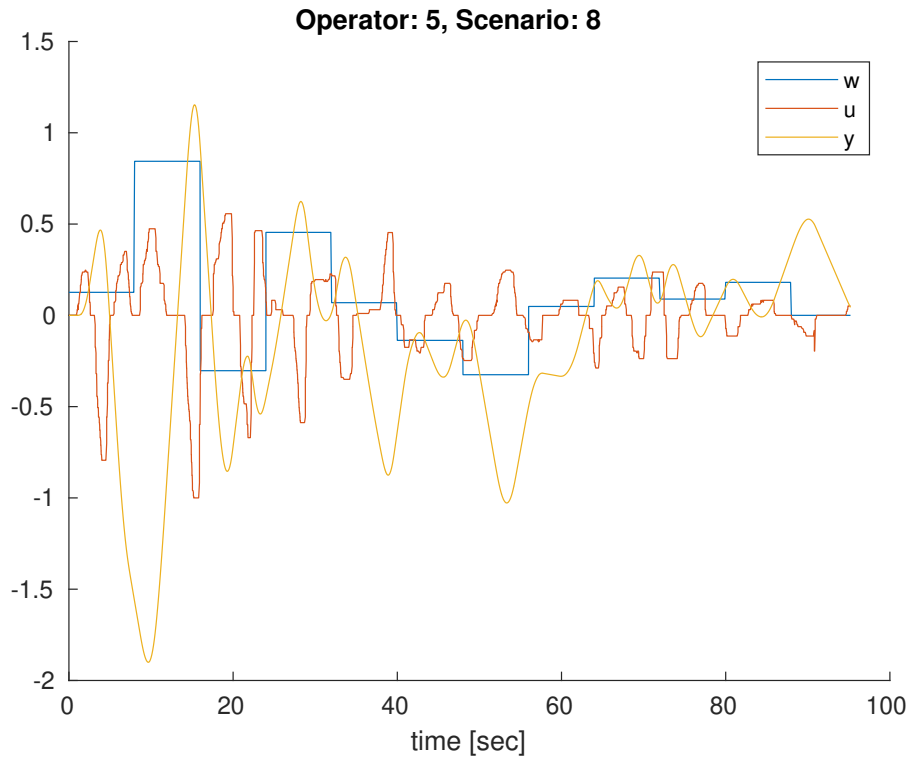
4.4 Závěr z měření

Cílem základního měření bylo ověřit použitelnost simulátoru pro měření odezvy lidského operátora a demonstrovat jeho použití a možné zpracování dat. Pomocí simulátoru se podařilo provést měření dat a ty následně zpracovat. Kvantifikovat kvalitu regulace pro jednotlivé operátory a scénáře a pro scénář 1 se podařilo identifikovat parametry modelu operátora.

Z naměřených průběhů je patrné, že někteří operátoři ztráceli kontrolu nad soustavou, například operátor 5 při scénáři 8 (4.9), zatímco jiným se dařilo soustavu řídit například operátor 2 a scénář 8 (4.10). Operátoři, ale (kromě soustavy $F_S = 1$) volili přístup bang-bang regulátoru.

Na základě ověřovacího měření byly stanoveny tyto připomínky, které byly zpracovány:

- Při zobrazení VW2 dochází k nepříjemnému přeblikávání barev u kuliček v případě, že se překrývají.
- Náhodné pulzy mají stále stejný průběh.
- Superponované sinusoidy mají stále stejný tvar a operátor si může jejich průběh zapamatovat.



Obr. 4.9: Scénář 8, Operátor 5, $F_S = \frac{1}{p^2}$

- Při zobrazení VW1 má kužel příliš tmavou barvu.

Z doplňujících měření vyplývá, že pásmo necitlivosti a vratné pružiny mají vliv na akční zásah operátora. Pro měření by tedy mělo být voleno takové vstupní zařízení, které nebude mít silné vratné pružiny nebo jiný vliv na operátora.

Závěr

V rámci diplomové práce byla shrnuta problematika identifikace modelu chování lidského operátora a představeny základní modely lidského operátora.

Byl navrhnut a implementován simulátor dynamického TITO systému s možností nelinearit. Veškeré parametry simulátoru (testovací scénáře) jsou nastavitelné z grafického uživatelského rozhraní. Rozhraní pro styk s operátorem může být tvořeno libovolným analogovým herním zařízením na straně vstupu, výstup je zobrazován na monitoru volitelným způsobem. Simulátor dále umožňuje porovnání s matematickým regulátorem. Pro zpracování naměřených dat byl vytvořen StrucToolbox.

Bylo provedeno základní měření zahrnující 13 scénářů se čtyřmi různými přenosy soustavy, ukázka zpracování dat a identifikace parametrů modelu lidského operátora. Měření ověřilo funkčnost a použitelnost implementovaného simulátoru, dále byly stanoveny podmínky pro zlepšení, které byly zapracovány. Z měření lze usoudit, že lidský operátor se v některých příkladech chová jako bang-bang regulátor. Tento jev byl podroben bližšímu zkoumání, které pomocí doplňujícího měření identifikovalo příčiny vedoucí k tomuto chování.

Z důvodu malého vzorku operátorů není smysluplné provést statistické zpracování naměřených průběhů vzhledem k datům z dotazníků.

Vytvořený simulátor může posloužit k dalšímu zkoumání odezvy lidského operátora, zejména zkoumání adaptace člověka na řízenou dynamiku, testování limitů lidského operátora při řízení různých dynamik v kontextu žádané hodnoty a poruchové veličiny. Simulátor dále také umožní identifikovat model řízení lidského operátora pro různé scénáře a zkoumat vlivy rozhraní mezi simulátorem a člověkem. Výsledky provedeného měření mohou pomoci především pro další volbu parametrů simulátoru.

Literatura

- [1] OBŠÍLOVÁ, Lucie, Marie HAVLÍKOVÁ a Zdeněk BRADÁČ. Neuromuskulární systém člověka. *Elektrorevue: Časopis pro elektrotechniku*. 2016, **2016**(4), 124-128. DOI: -. ISSN 1213-1539.
- [2] MCRUER, Duane T. a Ezra S. KRENDEL. *MATHEMATICAL MODELS OF HUMAN PILOT BEHAVIOR: AGARDograph No. 188*. Paris, France: NTIS, 1974, 83 s. AD 775 905. Dostupné také z: <https://apps.dtic.mil/dtic/tr/fulltext/u2/775905.pdf>.
- [3] JALOVECKÝ, Rudolf. Člověk v systému řízení letu letounu. In: *Měření, diagnostika, spolehlivost palubních soustav letadel: sborník příspěvků ... odborného semináře, Brno ...* Brno: Univerzita obrany, 2008, s. 67-74. ISBN 978-80-7231-555-0.
- [4] JIRGL, Miroslav. *Analýza modelů chování pilota při řízení letu letounu*. Brno, 2016, 142 s. Dizertační práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřicí techniky. Vedoucí práce Doc. Ing. Zdeněk Bradáč, Ph.D.
- [5] TAKAGI, T a M SUGENO. DERIVATION OF FROM HUMAN CONTROL RULES FUZZY OPERATOR'S ACTIONS CONTR. *IFAC Symposium on Fuzzy Information: Knowledge Representation and Decision Analysis*. Marseille, 1983, **1983**, 55-60. DOI: 10.1016/S1474-6670(17)62005-6. ISSN 1474-6670.
- [6] BLAHA, Petr a Petr VAVŘÍN. *Řízení a regulace I: Základy regulace lineárních systémů - spojité a diskrétní*. Brno: Vysoké učení technické, 2009.
- [7] AZARIA, M. a D. HERTZ. Time delay estimation by generalized cross correlation methods. *IEEE Transactions on Acoustics, Speech, and Signal Processing* [online]. 1984, **32**(2), 280-285 [cit. 2019-11-06]. DOI: 10.1109/TASSP.1984.1164314. ISSN 0096-3518. Dostupné z: <http://ieeexplore.ieee.org/document/1164314/>
- [8] HAVLÍKOVÁ, Marie. *Diagnostika systémů s lidským operátorem*. Brno, 2009. Dizertační práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřicí techniky. Vedoucí práce Doc. Ing. Zdeněk Malec, CSc.
- [9] *MATLAB Documentation* [online]. Natick (Massachusetts): The MathWorks, 2019 [cit. 2019-12-24]. Dostupné z: <https://www.mathworks.com/help/index.html>

- [10] Metody identifikace systémů z přechodových charakteristik. *Fakulta strojní - VŠB-TU Ostrava: sylaby a elektronické učebnice* [online]. Ostrava: VŠB-TU Ostrava, 2006 [cit. 2020-03-02]. Dostupné z: <http://books.fs.vsb.cz/Identifikace/str/metody.html>
- [11] *MID - Modelování a identifikace* [online]. Brno: Petr BLAHA, 2008 [cit. 2020-03-03]. Dostupné z: <https://sites.google.com/site/modelovaniaidentifikace/>
- [12] Volant Genius SpeedWheel 3 MT, PC. In: *EURONICS* [online]. [cit. 2020-03-21]. Dostupné z: <https://www.euronics.cz/image/product/800x800/158567.jpg>

Seznam příloh

A	Detailní struktura uložení konfiguračních proměnných a výchozí hodnoty	80
B	Detailní popis adresářové struktury	87
C	Detailní simulink schémata	90

A Detailní struktura uložení konfiguračních proměnných a výchozí hodnoty

MATLAB base workspace

— Joy	Konfigurace vstupního zařízení
— ID = 1 (MS windows) nebo 2 (GNU/Linux)	ID vstupního zařízení
— Osa	Volba použité osy vstupního zařízení
— A = 1	Pro subsystém A
— B = 2	Pro subsystém B
— Reverzovat	Reverzovat osu
— A = 0	
— B = 0	
— Filtrovat	Filtrovat vstupní signál
— A = 0	
— B = 0	
— Filtr	Přenos filtru vstupního signálu
— NUM	Čitatele koeficientů filtru
— A = 1	
— B = 1	
— DEN	Jmenovatelové koeficienty filtru
— A = [1e-5 0.5 1]	
— B = [1e-5 0.5 1]	
— Porucha	Konfigurace generátoru poruchové veličiny
— Zpozdeni	Zpoždění mezi zobrazením a vstupem do systému
— A = 0	
— B = 0	
— Volba	Volba možnosti poruchové veličiny
— A = 1	
— B = 1	
— NahodnaFaze ...	Příznak pro použití náhodné fáze sinusovek nebo náhodného seedu pro náhodné pulzy
— A = 0	
— B = 0	
— Pulzy	Konfigurace obdélníkového signálu
— Amplituda	Amplituda obdélníkového signálu (peak-to-peak)
— A = 1	
— B = 1	
— Perioda	Perioda obdélníkového signálu
— A = 1	
— B = 1	
— Strida	Střída obdélníkového signálu v %
— A = 50	
— B = 50	
— Nahodne	Konfigurace náhodného signálu
— Amplituda	Amplituda náhodného signálu (peak-to-peak)

└─ A = 0	
└─ B = 0	
FR .4	Regulátor zadaný přenosem
└─ A = 2.5984*tf([1], [0.55 1])	
└─ B = 2.5984*tf([1], [0.55 1])	
Zesileni	Zesílení regulátoru (za nelinearitami) .4 A = 1
└─ B = 1	
OmezeniStrmosti	Konfigurace omezení strmosti (směrnice)
└─ Rostouci	Omezení růstu (maximální směrnice)
└─ A = Inf	
└─ B = Inf	
└─ Klesajici	Omezení klesání (minimální směrnice)
└─ A = -Inf	
└─ B = -Inf	
Hystereze	Konfigurace hystereze
└─ Sirka	Šířka hystereze
└─ A = 0	
└─ B = 0	
Necitlivost	Konfigurace pásma necitlivosti
└─ Start	Začátek pásma necitlivosti
└─ A = 0	
└─ B = 0	
Omezeni	Konfigurace omezení (saturace)
└─ Upper	Horní mez
└─ A = 1	
└─ B = 1	
└─ Lower	Dolní mez
└─ A = -1	
└─ B = -1	
DopravniZpozdeni	Dopravní zpoždění regulátoru
└─ A = 0	
└─ B = 0	
Soustava	Konfigurace soustavy
└─ FS	Přenos soustavy
└─ A = tf([1], [1 1])	Přenos soustavy A -> A
└─ B = tf([1], [1 1 1])	Přenos soustavy B -> B
└─ AB = tf(0)	Přenos soustavy A -> B
└─ BA = tf(0)	Přenos soustavy B -> A
└─ DopravniZpozdeni	Dopravní zpoždění soustavy
└─ A = 0	
└─ B = 0	
└─ AB = 0	
└─ BA = 0	
ZadanaHodnota	Konfigurace generátoru žádané hodnoty
└─ Zpozdeni	Zpoždění mezi zobrazením a vstupem do systému
└─ A = 0	

└─ B = 0	
Volba	Volba možnosti poruchové veličiny
└─ A = 4	
└─ B = 4	
NahodnaFaze ...	Příznak pro použití náhodné fáze sinusovek nebo náhodného seedu pro náhodné pulzy
└─ A = 0	
└─ B = 0	
Pulzy	Konfigurace obdélníkového signálu
└─ Amplituda	Amplituda obdélníkového signálu (peak-to-peak)
└─ A = 1	
└─ B = 1	
└─ Perioda	Perioda obdélníkového signálu
└─ A = 1	
└─ B = 1	
└─ Strida	Střída obdélníkového signálu v %
└─ A = 50	
└─ B = 50	
Nahodne	Konfigurace náhodného signálu
└─ Amplituda	Amplituda náhodného signálu (peak-to-peak)
└─ A = 1	
└─ B = 1	
└─ PeriodaVzorkovani	Perioda vzorkování náhodného signálu
└─ A = 1	
└─ B = 1	
└─ Seed	Výchozí hodnota náhodného generátoru
└─ A = 23341	
└─ B = 23341	
Rampa	Konfigurace lineárně narůstajícího signálu
└─ Strmost	Strmost (směrnice) lineárně narůstajícího signálu
└─ A = 1	
└─ B = 1	
└─ Pocatek	Počáteční čas (bias) lineárně narůstajícího signálu
└─ A = 1	
└─ B = 1	
Sin1	Konfigurace superponovaných sinusoid - 1. sinusoida
└─ Amplituda	Amplituda (peak-to-peak) sinusoidy
└─ A = 0,25	
└─ B = 0,25	
└─ Frekvence	Frekvence sinusoidy
└─ A = 0,5	
└─ B = 0,7	
└─ Bias	Stejnosečná složka (bias) sinusoidy
└─ A = 0	
└─ B = 0	
└─ Faze	Fázový posuv sinusoidy

	A = 0	
	B = 0	
	Sin2	Konfigurace superponovaných sinusoid - 2. sinusoida
	Amplituda	Amplituda (peak-to-peak) sinusoidy
	A = 0,25	
	B = 0,25	
	Frekvence	Frekvence sinusoidy
	A = 0,2	
	B = 0,3	
	Bias	Stejnoseměrná složka (bias) sinusoidy
	A = 0	
	B = 0	
	Faze	Fázový posuv sinusoidy
	A = 0	
	B = 0	
	Sin3	Konfigurace superponovaných sinusoid - 3. sinusoida
	Amplituda	Amplituda (peak-to-peak) sinusoidy
	A = 0,25	
	B = 0,25	
	Frekvence	Frekvence sinusoidy
	A = 0,3	
	B = 0,1	
	Bias	Stejnoseměrná složka (bias) sinusoidy
	A = 0	
	B = 0	
	Faze	Fázový posuv sinusoidy
	A = 0	
	B = 0	
	Sin4	Konfigurace superponovaných sinusoid - 4. sinusoida
	Amplituda	Amplituda (peak-to-peak) sinusoidy
	A = 0,25	
	B = 0,25	
	Frekvence	Frekvence sinusoidy
	A = 0,2	
	B = 0,05	
	Bias	Stejnoseměrná složka (bias) sinusoidy
	A = 0	
	B = 0	
	Faze	Fázový posuv sinusoidy
	A = 0	
	B = 0	
	Sin5 - Sin10	Konfigurace superponovaných sinusoid - 1. - 10. sinusoida
	Amplituda	Amplituda (peak-to-peak) sinusoidy
	A = 0	
	B = 0	
	Frekvence	Frekvence sinusoidy

		A = 0	
		B = 0	
	└─	Bias	Stejnoseměrná složka (bias) sinusoidy
		A = 0	
		B = 0	
	└─	Faze	Fázový posuv sinusoidy
		A = 0	
		B = 0	
└─	Zobrazeni	Konfigurace zobrazení	
	└─	Volba = 1	Volba formátu zobrazení
	└─	Chyba = true	Zobrazení regulační odchylky
	└─	ZadanaHodnota = false	Zobrazení žádané hodnoty
	└─	Porucha = false	Zobrazení poruchové veličiny
	└─	AkcniZasah = false	Zobrazení akčního zásahu
	└─	RegulovanaVelicina = false	Zobrazení regulované veličiny
	└─	ZasahRegulatoru = false	Zobrazení zásahu regulátoru (před nelinearitami)
	└─	ZobrazitCiselne = true	Zobrazení číselné hodnoty
	└─	Osy = [-1 1 -1 1]	Nastavení rozsahu os [xmin xmax ymin ymax]
	└─	Sim	Vytvoření proměnné pro předání parametrů simulinku

B Detailní popis adresářové struktury

```
/ ..... Kořenový adresář
├── default_values ..... Podadresář se skripty nastavující výchozí hodnoty
│   ├── Joy_SetDefault.m ..... Výchozí hodnoty vstupního zařízení
│   ├── Porucha_SetDefault.m ..... Výchozí hodnoty generátoru poruchové veličiny
│   ├── Regulator_SetDefault.m ..... Výchozí hodnoty regulátoru
│   ├── Soustava_SetDefault.m ..... Výchozí hodnoty soustavy
│   ├── TestVstupu.m ..... Hodnoty pro test vstupního zařízení
│   ├── ZadanaHodnota_SetDefault.m ..... Výchozí hodnoty generátoru žádané hodnoty
│   └── Zobrazeni_SetDefault.m ..... Výchozí hodnoty pro zobrazení
├── GUI ..... Podadresář se soubory grafického rozhraní
│   ├── KonfiguraceRegulator.fig
│   ├── PoruchaPulzy.m
│   ├── KonfiguraceRegulator.m
│   ├── PoruchaRampa.fig
│   ├── HlavniMenu.fig
│   ├── KonfiguraceSoustava.fig
│   ├── PoruchaRampa.m
│   ├── HlavniMenu.m
│   ├── KonfiguraceSoustava.m
│   ├── PoruchaSin.fig
│   ├── Info.fig
│   ├── KonfiguraceSystemu.fig
│   ├── PoruchaSin.m
│   ├── Info.m
│   ├── KonfiguraceSystemu.m
│   ├── UAMT_color_RGB_CZ.png
│   ├── KonfiguraceVstupu.fig
│   ├── UlozeniSystemu.m
│   ├── KonfiguraceFRA.fig
│   ├── KonfiguraceVstupu.m
│   ├── ZadanaHodnotaNahodne.fig
│   ├── KonfiguraceFRA.m
│   ├── KonfiguraceZadaneHodnoty.fig
│   ├── ZadanaHodnotaNahodne.m
│   ├── KonfiguraceFRB.fig
│   ├── KonfiguraceZadaneHodnoty.m
│   ├── ZadanaHodnotaPulzy.fig
│   ├── KonfiguraceFRB.m
│   ├── KonfiguraceZobrazeni.fig
│   ├── ZadanaHodnotaPulzy.m
│   ├── KonfiguracePIDA.fig
│   ├── KonfiguraceZobrazeni.m
│   ├── ZadanaHodnotaRampa.fig
│   └── KonfiguracePIDA.m
```



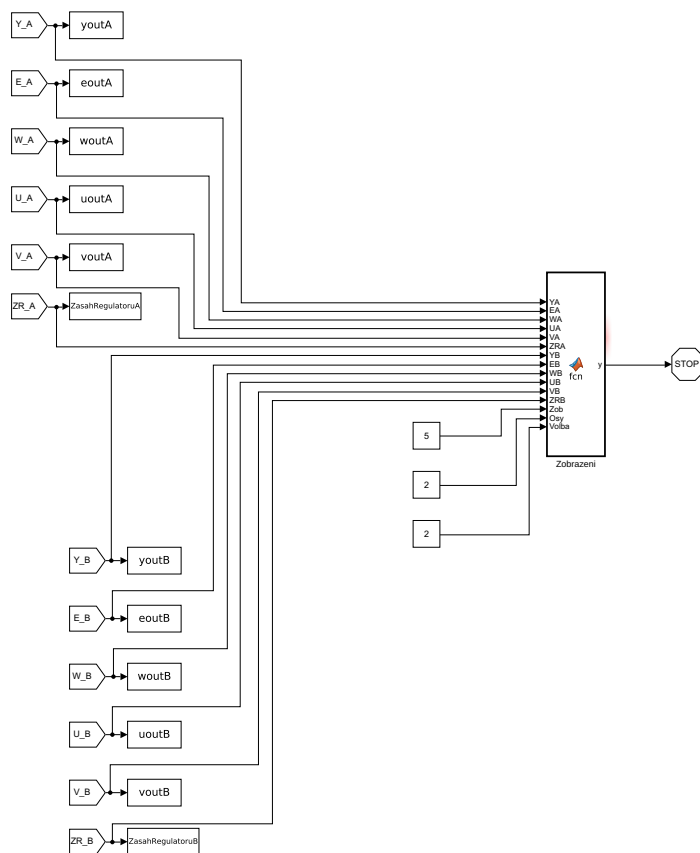
```

├─ NacteniSystemu.m
├─ ZadanaHodnotaRampa.m
├─ KonfiguracePIDB.fig
├─ ObrRegulacniObvod.bmp
├─ ZadanaHodnotaSin.fig
├─ KonfiguracePIDB.m
├─ PoruchaNahodne.fig
├─ ZadanaHodnotaSin.m
├─ KonfiguracePoruchy.fig
├─ PoruchaNahodne.m
├─ KonfiguracePoruchy.m
├─ PoruchaPulzy.fig
├─ simulink .....Podadresář se simulink modely
├─ VW .....Podadresář s virtuálními světy
│   └─ vw1.wrl
│     └─ vw2.wrl
│       └─ vw3.wrl
├─ main.slx ..... Model ve formátu poslední použité verze Simulinku
├─ main_R2015B.slx ..... Model ve formátu verze r2015b
├─ msfun_realtime_pacer.m
├─ odkomentovani.m
├─ odkomentovani_R2015B.m
├─ optimalizace.m
├─ optimalizace_R2015B.m
├─ realtime_pacer_lib.mdl
├─ slblocks.m
├─ stored_systems .....Podadresář s uloženými systémy
│   └─ 1
│     └─ 01-nahodnePulzy_tf(1).system
│       └─ 02-nahodnePulzy_tf((1),(1 0)).system
│         └─ 03-nahodnePulzy_tf((1),(1 0 0)).system
│           └─ 04-nahodnePulzy_tf((1),(2 1 0)).system
│             └─ Default.system
├─ stored_measurements .....Podadresář pro ukládání naměřených hodnot
│   └─ 1-09_03_20 ..... Podadresář s naměřenými hodnotami základního měření
├─ struc_toolbox ..... Podadresář s funkcemi pro práci se strukturami
│   └─ ClearMeasured.m
│     └─ CutStruc.m
│       └─ DelayDet.m
│         └─ PlotStruc.m
│           └─ StrucAlign.m
│             └─ StrucIAE.m
│               └─ StrucInterp.m
│                 └─ StrucISE.m
│                   └─ StrucTAE.m
│                     └─ Struc.ITSE.m

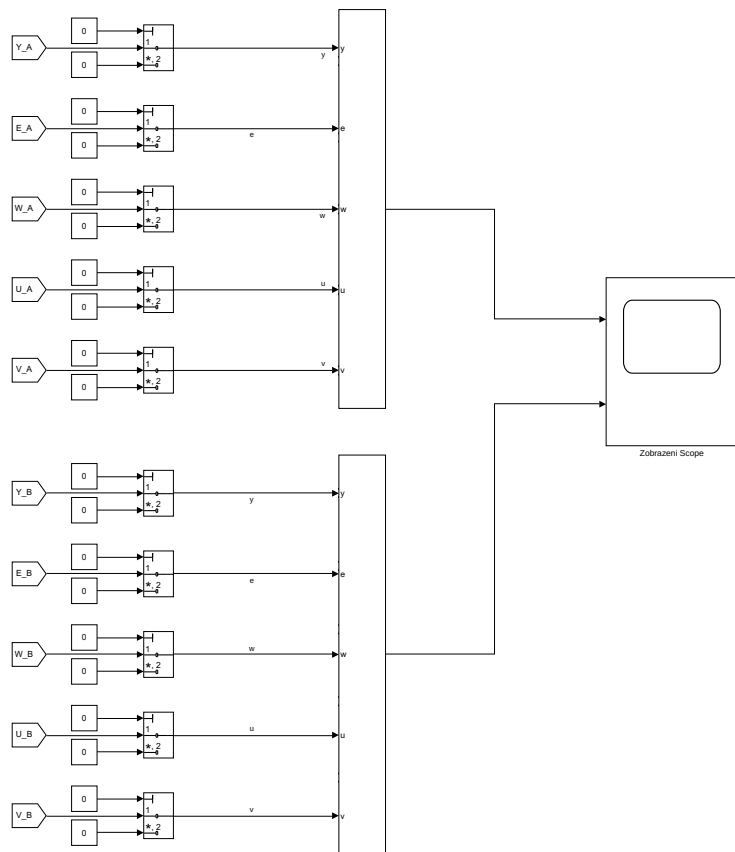
```

├─ StrucSteps.m
├─ StrucZOHE.m
└─ START.m

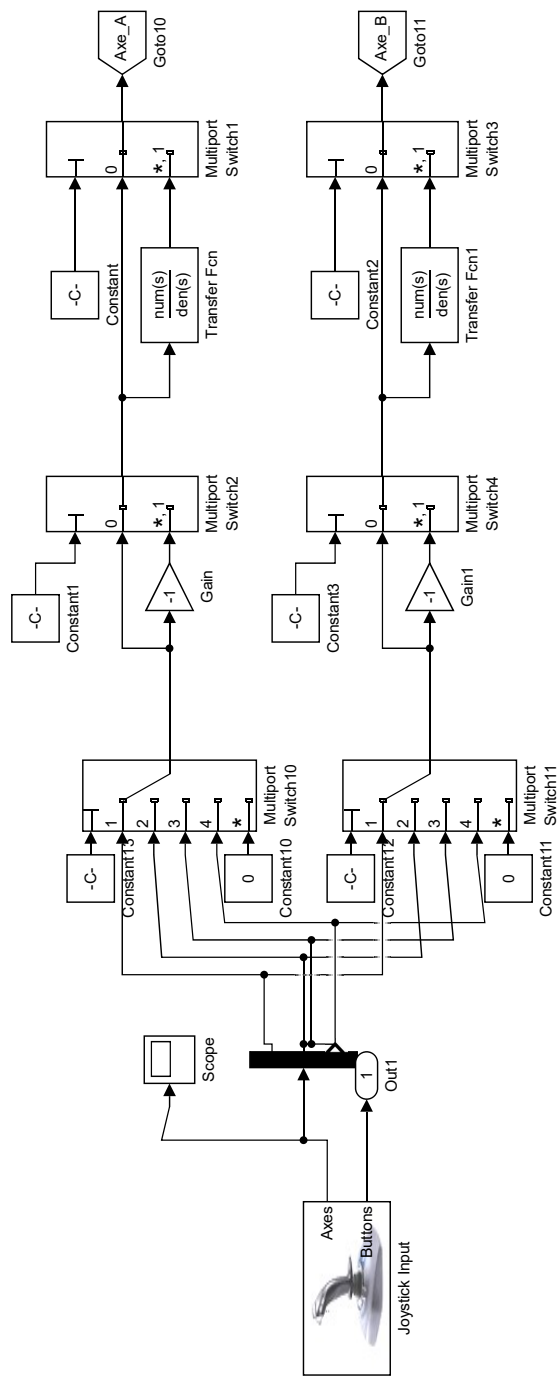
C Detailní simulink schémata



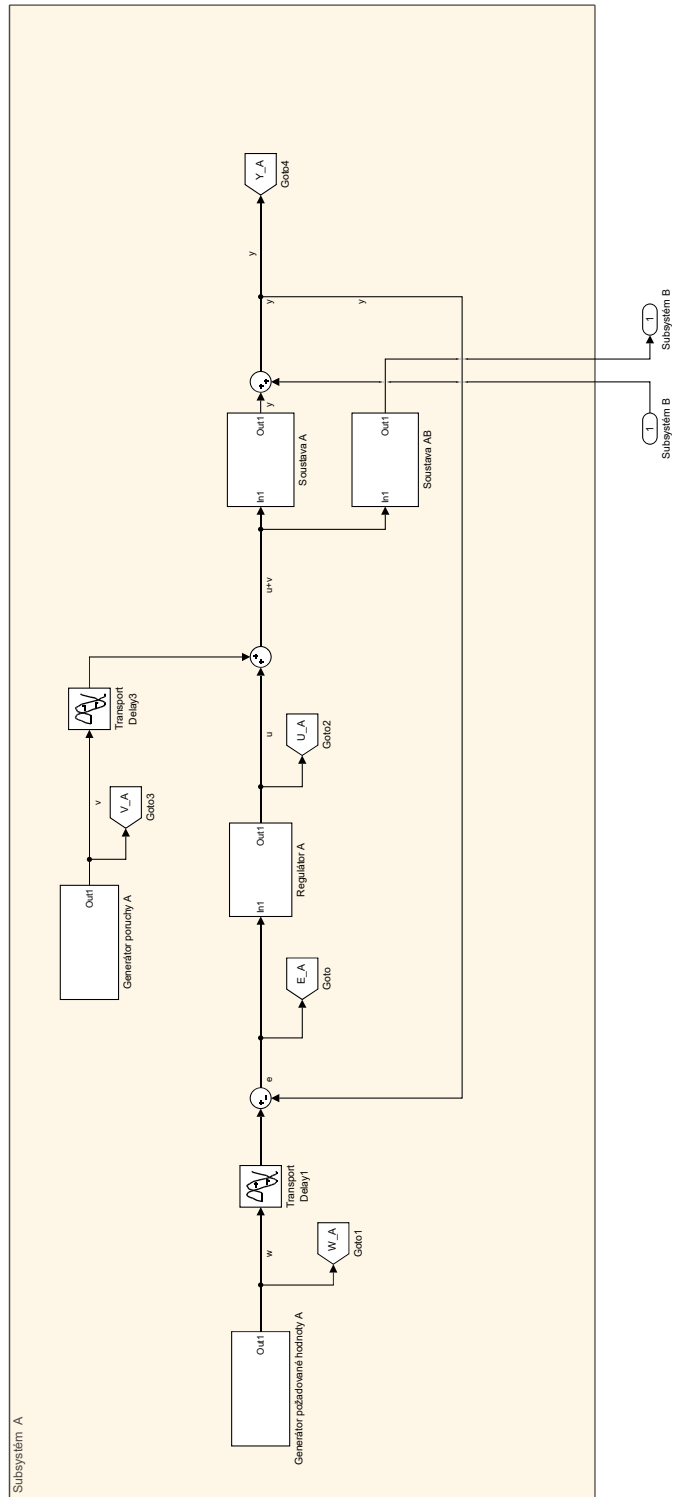
Obr. C.1: To Workspace část 1



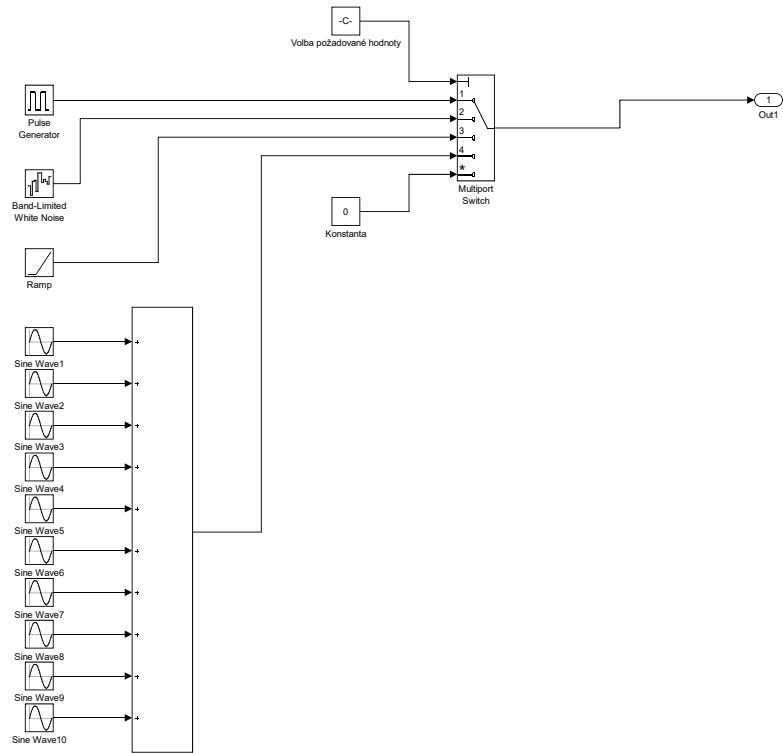
Obr. C.2: To Workspace část 2



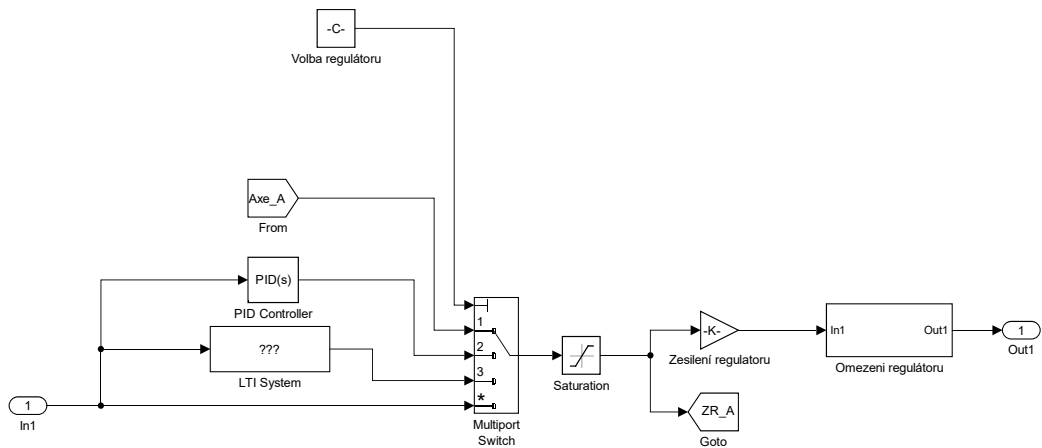
Obr. C.3: Vstup



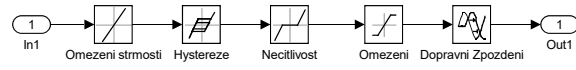
Obr. C.4: Subsystem A



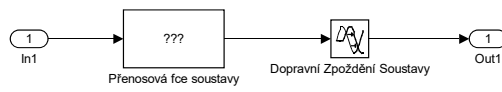
Obr. C.5: Generátor žádané hodnoty



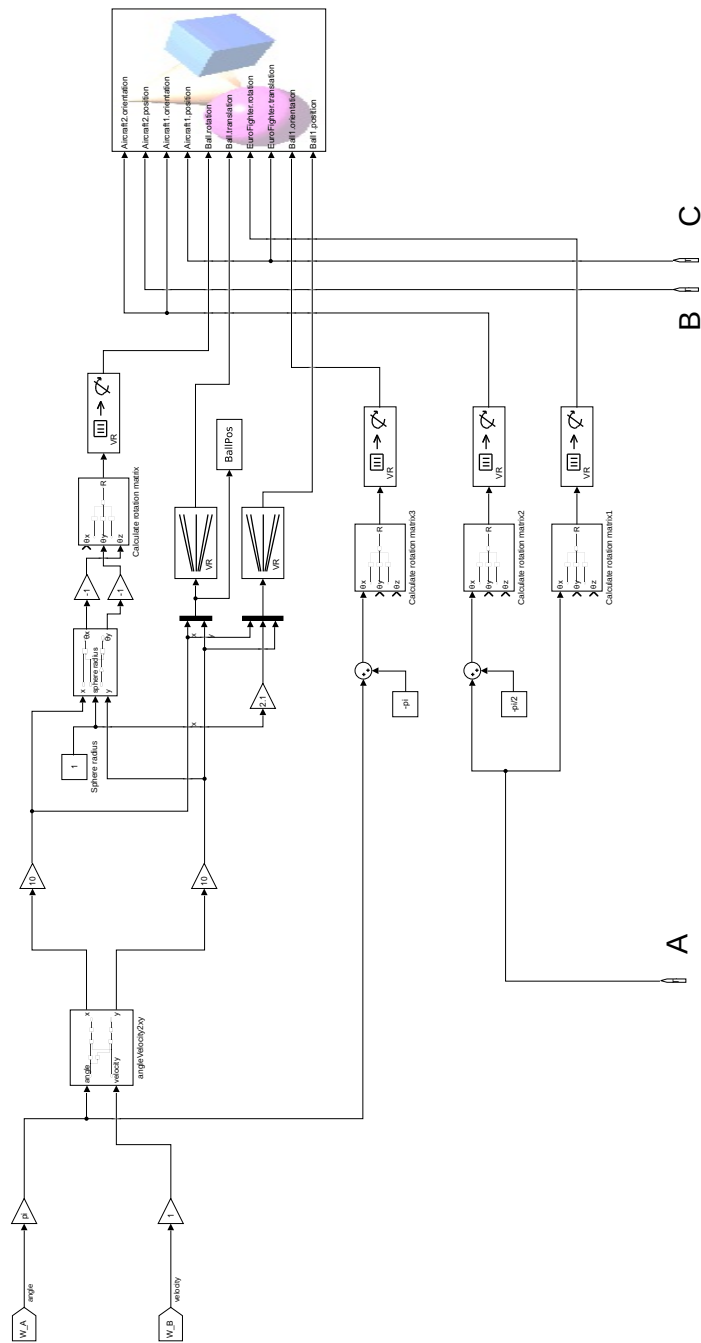
Obr. C.6: Regulátor



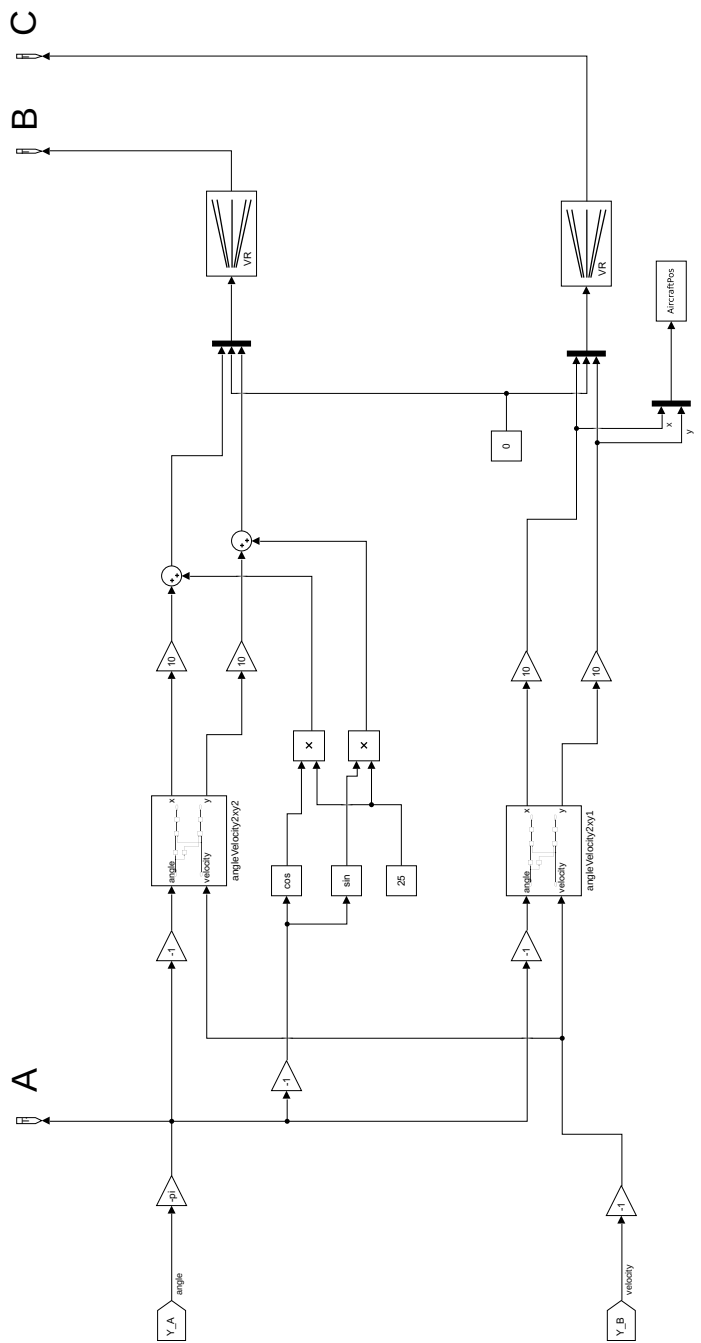
Obr. C.7: Omezení regulátoru



Obr. C.8: Soustava



Obr. C.9: Blok VW3 část 1



Obr. C.10: Blok VW3 část 2