

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

**REALIZACE ŘÍDICÍHO SYSTÉMU DVOUROTOROVÉHO
LABORATORNÍHO SYSTÉMU V PROSTŘEDÍ LABVIEW**

Bc. Milan Zapletal

Diplomová práce
2014

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Milan Zapletal**
Osobní číslo: **I12521**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Řízení procesů**
Název tématu: **Realizace řídicího systému dvourotorového laboratorního systému v prostředí LabVIEW**
Zadávací katedra: **Katedra řízení procesů**

Z á s a d y p r o v y p r a c o v á n í :

Cíl práce:

Cílem práce je teoretický návrh a následná softwarová implementace řídicího systému upoutaného dvourotorového laboratorního systému v prostředí LabVIEW. Součástí práce bude identifikace uvedeného systému a simulace regulačních pochodů se zvoleným regulačním algoritmem v prostředí Matlab - Simulink.

Teoretická část:

Tvorba modelu na základě experimentální parametrické identifikace. Návrh vybraných regulačních algoritmů a nalezení jejich optimálních parametrů.

Implementační část:

Vytvoření skriptů v Matlabu a modelů v Simulinku. Realizace propojení laboratorní úlohy s PC s využitím měřicí karty NI USB 6008/9. Tvorba aplikace ve vývojovém prostředí LabVIEW.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

- [1] HAVLÍČEK, L. Modelování a řízení vícerozměrové soustavy [disertační práce]. Pardubice: Univerzita Pardubice, 2010. Fakulta elektrotechniky a informatiky. 89 s.
- [2] DORF, R. C.; BISHOP, R. H. Modern Control Systems. 10 vyd. New Jersey: Prentice Hall, 2005. 881 s. ISBN 0-13-127765-0.
- [3] BALÁTĚ, J. Automatické řízení. 2 vyd. Praha: BEN - technická literatura, 2004. 663 s. ISBN 80-7300-148-9.
- [4] KUPKA, L.; JANEČEK, J. Matlab & Simulink: řešené příklady. Lanškroun: TG tisk a SOŠ a SOU, 2007. 224 s. ISBN 978-80-239-9532-9.

Vedoucí diplomové práce:

Ing. Libor Kupka, Ph.D.

Katedra řízení procesů

Datum zadání diplomové práce:

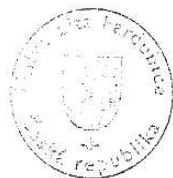
24. října 2013

Termín odevzdání diplomové práce:

16. května 2014



- prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2013

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 12. 5. 2014

Milan Zapletal

Poděkování

Na tomto místě bych rád poděkoval vedoucímu práce panu Ing. Liboru Kupkovi, Ph.D., za vstřícnost, pomoc a mnoho konzultací při vedení celé práce. Dále bych rád poděkoval panu Ing. Liboru Havlíčkovi, Ph.D., za cenné rady ohledně dvourotorového laboratorního systému.

V Pardubicích dne 12. 5. 2014

Milan Zapletal

Anotace

Cílem diplomové práce je vytvořit řídicí aplikaci v prostředí v LabVIEW, která je propojena s dvourotorovým laboratorním systémem pomocí komunikační karty NI USB 6009. Řídicí aplikace se skládá ze dvou částí. První část slouží k manuálnímu ovládní laboratorního systému a ukládá veškerá potřebná experimentální měřená data. Druhá část obsahuje implementovaný algoritmus řízení v podobě regulátorů PID, který umožňuje automatickou regulaci. Manuální a automatický režim je přepínatelný. Obsahem práce je také experimentální identifikace dvourotorového systému a tvorba matematických modelů v podobě přenosových funkcí, které jsou základem pro návrh PID regulátorů.

Klíčová slova

Dvourotorový laboratorní systém, LabVIEW, PID regulátor, experimentální identifikace

Title

Realization of the control system of the twin-rotor laboratory system in Labview

Annotation

The aim of the thesis is to create a control application in LabVIEW environment, connected with a twin-rotor laboratory system using communication card NI USB 6009. The Control application has two parts. The first part is used to manually control the laboratory system, and it stores all the necessary experimental measured data. The second part contains the control algorithm implemented in the form of a PID controller that enables automatic regulation. The manual and automatic modes are switchable. The thesis also deals with the experimental identification of a twin-rotor system and the creation of mathematical models in the form of transfer functions which are the basis for the design of PID controllers.

Keywords

Twin-rotor laboratory system, LabVIEW, PID controller, experimental identification

Obsah

Seznam zkratk	9
Seznam symbolů	10
Seznam obrázků	11
Seznam tabulek	13
Úvod	14
1 Základní informace o dvourotorovém systému	16
1.1 Konstrukce laboratorního modelu	16
1.2 Rozsahy úhlů vychýlení ramene ve vertikální a horizontální rovině.....	17
1.3 Pohony rotorů a jejich řízení.....	17
1.4 Získání informací o úhlech natočení.....	18
1.5 Předpoklady o chování systému	19
2 Řízení a regulace	20
2.1 Regulátory PID	21
2.2 Rozdělení soustavy MIMO na samostatné větve SISO	22
3 Měřicí karta NI USB 6008/9	25
4 Vývojové prostředí LabVIEW	27
4.1 Filozofie LabVIEW - Virtual Instrument	27
4.2 Vnitřní část – blokový diagram	28
4.3 Vnější část – přední panel	29
4.4 Datové typy.....	30
5 Oživení laboratorního systému	32
5.1 Propojení s počítačem	32
5.2 Měření a ukládání dat	37
5.3 Převod binárního souboru do MATLAB souboru	40
6 Experimentální identifikace dvourotorového systému	42
6.1 Vliv ocasního rotoru na úhel vychýlení hlavního rotoru	42
6.2 Experimentální identifikace hlavního rotoru	43
6.2.1 Omezení pracovního rozsahu, analýza dynamiky systému	44
6.2.2 Tvorba aproximačního modelu	47
6.3 Vliv hlavního rotoru na úhel vychýlení ocasního rotoru	50
6.4 Experimentální identifikace ocasního rotoru	51
6.4.1 Tvorba aproximačního modelu	52

7	Návrh regulátorů	54
7.1	Omezení skutečných regulačních pochodů rotorů.....	54
7.2	Regulátory navržené metodou Ziegler-Nichols	54
7.2.1	Regulátor ZN pro hlavní rotor	55
7.2.2	Regulátor ZN pro ocasní rotor	57
7.2.3	Simulace regulačních pochodů ZN.....	58
7.3	Regulátory navržené aplikací simplexové optimalizace.....	61
8	Implementace regulace v LabVIEW.....	65
8.1	Testování regulace na laboratorním systému.....	67
8.1.1	Vyhodnocení regulace hlavního rotoru.....	67
8.1.2	Vyhodnocení regulace ocasního rotoru	69
	Závěr	71
	Literatura	73

Seznam zkratk

PID	Proporcionálně-integračně-derivační
PI	Proporcionálně-integrační
USB	Universal Serial Bus
NI	National Instruments
URO	Uzavřený regulační obvod
MIMO	Multi In Multi Out
SISO	Single In Single Out
A/D	Převodník analogovo-digitální
KŘP	Katedra řízení procesů
VI	Virtual Instrument
ZN	Ziegler-Nichols
OPT	Optimalizovaný

Seznam symbolů

$w(t)$	Žádaná hodnota
$e(t)$	Regulační odchylka
$u(t)$	Akční veličina
$y(t)$	Regulovaná veličina
r_0	Zesílení regulátoru
T_i	Integrační konstanta regulátoru
T_d	Derivační konstanta regulátoru
T_k	Kritická perioda
r_{0k}	Kritické zesílení
e_{\max}	Maximální regulační odchylka
d_1	Porucha na výstupu
d_2	Porucha na výstupu
f	Optimalizační kritérium pro hledání parametrů modelu
J	Optimalizační kritérium pro hledání parametrů regulátoru
k	Relativní překmit
t_r	Doba regulačního pochodu
κ	Penalizace akční veličiny

Seznam obrázků

Obrázek 1.1 – Dvourotorový laboratorní model upoutané helikoptéry (HAVLÍČEK, 2010)..	16
Obrázek 1.2 – Omezení ve vertikálním směru (vlevo) a v horizontálním směru (vpravo)	17
Obrázek 1.3 – Ilustrace optického kruhového inkrementálního snímače (HAVLÍČEK, 2010)	18
Obrázek 1.4 – Blokové schéma dvourotorového systému (HORÁK, 2011).....	19
Obrázek 2.1 – Uzavřený regulační obvod	20
Obrázek 2.2 – Standardní zapojení PID regulátoru	21
Obrázek 2.3 – Blokové schéma decentralizovaného regulačního obvodu (MACHÁČEK, 2008).....	23
Obrázek 2.4 – Blokové schéma rozděleného regulačního obvodu s poruchami na výstupu....	24
Obrázek 3.1 – Měřicí karta NI USB 6009	25
Obrázek 4.1 – Blokový diagram (vlevo), čelní panel (vpravo)	27
Obrázek 4.2 – Paleta funkcí v blokovém diagramu.....	28
Obrázek 4.3 – Paleta ovládacích prvků v předním panelu	29
Obrázek 4.4 – Datové typy	30
Obrázek 5.1 – Paleta NI – Data Acquisition	32
Obrázek 5.2 – 1. část programu – přístup na hardware měřicí karty	33
Obrázek 5.3 – 2. část programu – výběr dat z Task signálu, ovládání a měření	34
Obrázek 5.4 – 3. část programu – čistící rutina	35
Obrázek 5.5 – Ukončení programu.....	36
Obrázek 5.6 – Čelní panel aplikace pro oživení laboratorního systému	37
Obrázek 5.7 – Část programu, která ukládá data.....	38
Obrázek 5.8 – Postupné vykonání operací při ukládání měřených dat.....	39
Obrázek 5.9 – Čelní panel rozšířeného programu s ukládáním dat	40
Obrázek 5.10 – Program na převod dat	41
Obrázek 5.11 – Čelní panel programu na převod dat	41
Obrázek 6.1 - Statická charakteristika úhlu vychýlení hlavního rotoru	43
Obrázek 6.2 – Upravená statická charakteristika omezeného pracovního rozsahu.....	45
Obrázek 6.3 – 3 přechodové charakteristiky jako reakce na jednotkové skoky v „lineární“ oblasti, posunuté do počátku souřadnicového systému	46
Obrázek 6.4 – Zapouzdření všech částí systému do jednoho bloku	48
Obrázek 6.5 – Přenos aproximující soustavu.....	48
Obrázek 6.6 – Zobrazení reakce na jednotkový skok reálného systému a aproximačního modelu	49
Obrázek 6.7 – Porovnání statických charakteristik	50
Obrázek 6.8 – Přechodová charakteristika ocasního rotoru	52
Obrázek 6.9 - Zobrazení reakce na jednotkový skok reálného systému a aproximačního modelu	53
Obrázek 7.1 – Relé ve zpětné vazbě, hlavní rotor	55
Obrázek 7.2 – Kritické kmity s relé ve zpětné vazbě, hlavní rotor.....	56
Obrázek 7.3 – Relé ve zpětné vazbě, ocasní rotor	57
Obrázek 7.4 – Kritické kmity s relé ve zpětné vazbě, ocasní rotor	57

Obrázek 7.5 – Simulace URO hlavního rotoru.....	59
Obrázek 7.6 – Simulace URO ocasního rotoru.....	59
Obrázek 7.7 – Simulace regulačního pochodu hlavního rotoru, regulátor PID, ZN	60
Obrázek 7.8 – Simulace regulačního pochodu hlavního rotoru, regulátor PI, ZN	60
Obrázek 7.9 – Simulace regulačního pochodu ocasního rotoru, regulátor PID, ZN	60
Obrázek 7.10 – Simulace regulačního pochodu ocasního rotoru, regulátor PI, ZN	61
Obrázek 7.11 – Optimalizační schéma (KUPKA, 2013).....	62
Obrázek 7.12 – Simulace regulačního pochodu hlavního rotoru, regulátor PID, OPT	63
Obrázek 7.13 – Simulace regulačního pochodu hlavního rotoru, regulátor PI, OPT	63
Obrázek 7.14 – Simulace regulačního pochodu ocasního rotoru, regulátor PID, OPT	64
Obrázek 7.15 – Simulace regulačního pochodu ocasního rotoru, regulátor PI, OPT.....	64
Obrázek 8.1 – Implementace regulátoru v LabVIEW	65
Obrázek 8.2 – Čelní panel kompletní aplikace pro regulaci laboratorního systému	66
Obrázek 8.3 – Skutečný regulační pochod hlavního rotoru, regulátor PI, OPT	68
Obrázek 8.4 – Skutečný regulační pochod hlavního rotoru, regulátor PI, ZN	68
Obrázek 8.5 – Skutečný regulační pochod ocasního rotoru, regulátor PI, OPT.....	69

Seznam tabulek

Tabulka 2.1 – Výpočet parametrů regulátorů metodou Ziegler-Nichols.....	22
Tabulka 6.1 – Vliv ocasního rotoru na rozsah úhlu vychýlení hlavního rotoru	42
Tabulka 7.1 – Nalezené parametry regulátorů pro hlavní rotor.....	56
Tabulka 7.2 – Nalezené parametry regulátorů pro ocasní rotor	58
Tabulka 7.3 – Parametry regulátorů hlavního rotoru ZN a OPT	62
Tabulka 7.4 – Parametry regulátorů ocasního rotoru ZN a OPT.....	62
Tabulka 7.5 – Vyhodnocení simulací regulačních pochodů.....	63

Úvod

Laboratorní modely se konstruují proto, aby bylo možné experimentovat a získávat zkušenosti s řízením těchto modelů, které se svými vlastnosti více či méně blíží reálným soustavám. Experimentování na reálných soustavách se provádí zřídka, protože je to finančně a časově náročné, někdy také nebezpečné. Pokud je vyvíjeno nějaké složité zařízení, začíná se na laboratorním modelu a na základě laboratorního modelu je konstruováno požadované zařízení.

Zkoumáním laboratorního modelu se získají informace o jeho chování a vytvoří se matematický model, který popisuje dynamické vlastnosti modelu (soustavy) a zkoumaná soustava se posune do prostředí simulací. Simulace jsou dnes téměř všude využívány, protože existují sofistikované simulační nástroje a v současné době také podpora výkonné výpočetní techniky. Matematický model lze získat s různou přesností. Je snahou získat co nejpřesnější matematický model, ale většinou není možné získat takový model, který plně odpovídá analyzované soustavě a popisuje celý pracovní rozsah. Přesnější model se nalezne matematicko-fyzikální analýzou, která dokáže zahrnout i nelinearity, ale stále je nutno při postupu nějaké skutečnosti zjednodušovat, protože různé malé vlivy (například tření součástí) se dají inženýrsky zanedbat. Dalším a nejvíce využívaným přístupem je experimentální identifikace soustavy, která je jednodušší, ale přináší méně přesný model, hlavně v případě složitějších soustav. Například zkoumáním přechodové charakteristiky, jakožto odezvy soustavy na jednotkový skok, lze zjistit její dynamické vlastnosti a následně získat parametry pro tvorbu aproximačního matematického modelu v podobě přenosové funkce. Pro urychlení identifikace a vyhodnocování většího množství dat dnes slouží různé numerické optimalizační metody, které umožní nalézt parametry přenosu programově.

Po tvorbě matematického modelu následuje návrh řízení. Při použití aproximačního modelu, získaného experimentální identifikací, se nezíská tak přesný model, aby bylo možno použít složitějších řídicích algoritmů. Proto je vhodné využít regulátor, u kterého se předpokládá skutečnost, že řízení bude fungovat i navržením na méně přesném aproximačním modelu. Takovouto podmínku splňuje hojně využívaný PID regulátor, který vykazuje robustní chování, což je schopnost provádět úspěšnou stabilní regulaci i při různých druzích poruch. Robustní regulátor ale většinou nemá tak kvalitní regulační pochod, jako jiné druhy algoritmů navržených s použitím přesnějšího modelu, ale s robustním PID je zajištěna stabilita URO. Pokud je regulační pochod v simulacích uspokojivý a stabilní, otestuje se na laboratorním modelu. Regulační pochod, simulovaný na matematickém modelu, může být uspokojivý, ale pro řízení laboratorního modelu může mít navržené řízení horší vlastnosti, případně může být nestabilní. V tomto případě je potřeba parametry regulátoru upravit tak, aby byla regulace laboratorního modelu uspokojivá.

Cílem této diplomové práce je vytvořit aplikaci v prostředí LabVIEW, která bude řídit laboratorní model upoutaného dvourotorového systému, přičemž spojení mezi řídicím systémem a laboratorním modelem zajistí měřicí karta NI USB 6009. Aplikace bude mít implementovány PID regulátory, jejichž návrhu bude předcházet experimentální identifikace laboratorního modelu a tvorba matematických modelů v podobě přenosových funkcí. K vyhodnocování experimentálních dat, tvorbě přenosových funkcí a návrhu regulátorů, bude využito prostředí MATLAB a SIMULINK.

V teoretické části práce bude nejprve popsána konstrukce modelu, rozsahy úhlů vychýlení ramene v horizontálním a vertikálním směru, pohony rotorů a jejich vlastnosti. S tím souvisí uvedení principu měření úhlů vychýlení ramene. Bude stručně popsán návrh PID regulátorů a uvedeny podmínky, za kterých bude dvourozměrný systém upoutané helikoptéry rozdělen na dvě samostatné soustavy a regulován dvěma samostatnými regulátory. Z těchto zjednodušení se bude vycházet při návrhu řízení.

Je vhodné se stručným popisem zaměřit na měřicí kartu NI USB 6009, pomocí které bude laboratorní model propojen s řídicím systémem a také stručně popsat prostředí LabVIEW, aby čtenář pochopil základní filozofii a programová schémata. Podrobněji budou rozebrány součásti (funkční bloky), které souvisí s řízením dvourotorového systému. Znalost významu jednotlivých bloků je důležitá pro pochopení programového schématu.

V praktické části bude nejprve vytvořena aplikace, která oživí laboratorní model a bude sloužit k ručnímu ovládní, pro vyzkoušení chování modelu. Následně bude aplikace upravena tak, aby bylo možné ukládat experimentálně změřená data, v tomto případě přechodové charakteristiky, které budou základem pro tvorbu přenosových funkcí. Systém bude proměřen tak, aby se vyhodnocením získaných přechodových charakteristik mohly vytvořit experimentální modely v podobě přenosových funkcí. Měřením se budou také zkoumat případné nelinearity a omezení rozsahu řízení (například pomocí statické charakteristiky), které souvisí s vhodným nalezením přenosové funkce. Tvorba přenosových funkcí z přechodových charakteristik bude prováděna v prostředí MATLAB a SIMULINK.

Poté bude následovat návrh regulátorů, opět v prostředí MATLAB a SIMULINK, a budou stanovena omezení, která je nutné zavést, pro úspěšnou regulovatelnost upoutané helikoptéry.

Po simulacích a vybrání uspokojivého regulačního pochodu budou regulátory implementovány do aplikace v LabVIEW, pomocí které bude provedeno testování na laboratorním modelu.

1 Základní informace o dvourotorovém systému

Laboratorní systém je realizován jako upoutaná helikoptéra, která modeluje skutečnou helikoptéru. Soustava je systém se dvěma vstupy a dvěma výstupy, jedná se tedy o systém MIMO. Vstupem do soustavy jsou řídicí napětí, která ovládají otáčky motorů, pohybující rotory. Výstupem soustavy je vertikální a horizontální úhel natočení modelu opět v podobě napěťových úrovní.

1.1 Konstrukce laboratorního modelu

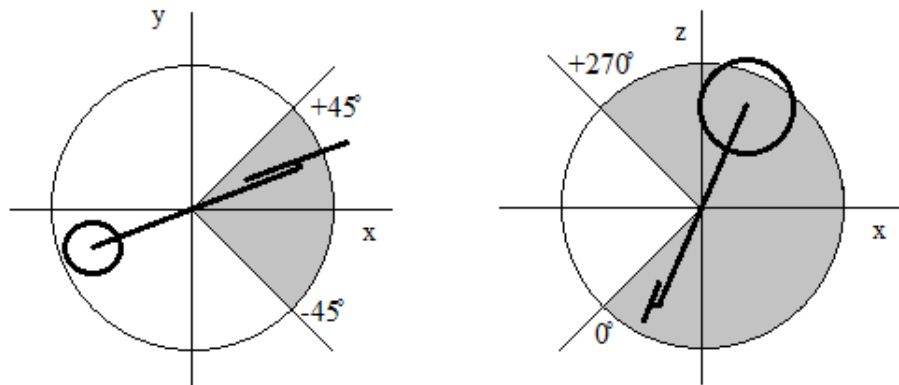
Pohyblivá část upoutané helikoptéry je rameno, které je spojeno s pevnou základnou pomocí ložisek umožňujících jeho otáčení. Na koncích ramene jsou umístěny dva vzájemně na sebe kolmé rotory, které pohánějí stejnosměrné motory. Hlavní rotor zdvihá model ve vertikálním směru (vertikální úhel vychýlení), ocasní rotor otáčí modelem v horizontálním směru (horizontální úhel vychýlení). Vychylování v horizontálním směru je možné po směru i proti směru hodinových ručiček. Zhruba uprostřed ramene je umístěno protizávaží, které jednak simuluje hmotnost helikoptéry a jednak vyvažuje celý model. Model je zkonstruován tak, že osy otáčení v obou směrech vychází ze stejného bodu, právě v místě otočného spojení. Laboratorní model je zobrazen na obrázku 1.1.



Obrázek 1.1 – Dvourotorový laboratorní model upoutané helikoptéry (HAVLÍČEK, 2010)

1.2 Rozsahy úhlů vychýlení ramene ve vertikální a horizontální rovině

Model upoutané helikoptéry má vzhledem ke konstrukci omezení rozsahů úhlů vychýlení. Ve vertikálním směru je otáčení kolem středové osy omezeno v rozsahu -45° až $+45^\circ$, v horizontálním směru 0° až $+270^\circ$. Hraničním místem omezených rozsahů je takzvaný doraz. Doraz je mechanická zarážka, která limituje rozsah vychýlení. Pro představu je na obrázku 1.2 zobrazen pracovní rozsah systému ve vertikální a horizontální rovině, při pohledu z boku a ze shora. Větší z kružnic naznačuje hlavní rotor.



Obrázek 1.2 – Omezení ve vertikálním směru (vlevo) a v horizontálním směru (vpravo)

1.3 Pohony rotorů a jejich řízení

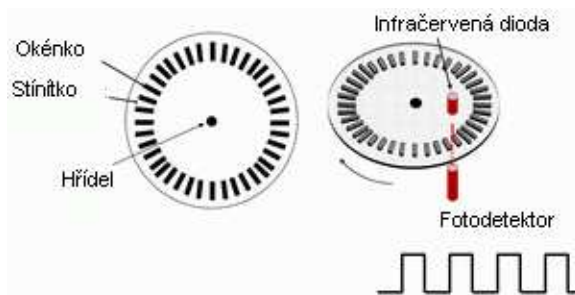
Akčními členy systému jsou dva stejnosměrné motory, jejichž otáčky jsou řízeny stejnosměrným napětím $0 - 5$ V, avšak každý motor rozdílným způsobem. Pohon hlavního rotoru helikoptéry zdvihá, směr otáčení hřídele motoru je jednostranný (po směru hodinových ručiček) a motor svým výkonem překonává gravitační sílu. To znamená směr pohybu ramene „nahoru“. Při poklesu otáček rameno ve vertikální poloze začne klesat, protože motor svým výkonem nestačí překonávat gravitační sílu. To znamená směr „dolů“. Napětíový rozsah pro řízení otáček je $0 - 5$ V.

Pohon ocasního rotoru s helikopterou otáčí „doprava“ i „doleva“. To znamená, že motor ocasního rotoru se točí po směru i proti směru hodinových ručiček. Tato vlastnost je jen výsadou laboratorního modelu, ocasní rotor reálné helikoptéry toto neumožňuje. Z důvodu realizovatelnosti vychylování ramene „doprava“ a „doleva“, je napětíový rozsah pro řízení pohonu rozdělen na dvě části. Řídicí napětí je $0 - 2,5$ V pro otáčení modelu doleva“. Pro otáčení „doprava“ je to $2,5 - 5$ V. Tímto provedením je o polovinu snížena přesnost řízení otáček, ale stále je dostačující. Při řídicím napětí 0 V má rotor po směru hodinových ručiček maximální otáčky, zvyšováním napětí otáčky klesají. Při dosažení $2,5$ V se rotor zastaví. V rozmezí $0 - 2,5$ V tedy rotor vychyluje rameno k dorazu 0° , „doleva“. Dalším zvyšováním napětí v rozmezí $2,5 - 5$ V se otáčky motoru zvyšují, ale proti směru hodinových ručiček. V tomto rozmezí rotor vychyluje rameno k dorazu 270° , „doprava“. Vyhodnocení řídicího napětí a tím změny polarity napájení ocasního motoru obstarává elektronická řídicí jednotka s mikroprocesorem.

Znalost těchto řídicích napěťových rozsahů obou motorů je důležitá pro správnou identifikaci systému a návrh regulátorů. V dalším textu práce jsou obě celé soustavy a jejich aerodynamické vlastnosti v podobě rameno/motor/rotor nazývány jen jako „rotor“ (hlavní rotor, ocasní rotor), aby bylo na každou soustavu zjednodušeně nahlíženo jako na jeden objekt.

1.4 Získání informací o úhlech natočení

Nezbytným údajem pro řízení je získávání informací o aktuální poloze jednotlivých rotorů. Model je vybaven dvěma kruhovými inkrementálními optickými snímači, které zajišťují informaci o úhlu natočení. Počítání úhlu zajišťuje elektronická řídicí jednotka s mikroprocesorem ATmega8. Osa optického snímače je spojena s osou ramene, otáčení stínítka je tedy přímo úměrné natočení modelu. Fotodetektor je pevně spojen s konstrukcí. Snímání probíhá v tzv. diferenciálním režimu, díky němuž elektronická jednotka dokáže také určit směr otáčení ramene v horizontálním i vertikálním směru. Ilustrace optického snímače je na obrázku 1.3.



Obrázek 1.3 – Ilustrace optického kruhového inkrementálního snímače (HAVLÍČEK, 2010)

Jelikož řízení probíhá analogově v rozsahu 0 – 5 V, tak i měření polohy probíhá stejným způsobem. Na rozsah -45° až $+45^\circ$ připadá 0 – 5 V, na rozsah 0° až $+270^\circ$ připadá také 0 – 5 V. To by znamenalo, že vertikální přesnost je $5/90$ V a horizontální přesnost je $5/270$ V. Z toho také plyne, že přesnost získání informace o poloze je ve vertikálním směru větší. V reálných podmínkách je přesnost polohy maximálně 1° , která je vymezena vlastnostmi inkrementálního snímače. Znalost přesnosti a rozsahů je opět důležitá pro návrh řízení. Napěťové údaje odpovídají aktuální poloze. Proto je jednodušší při návrhu regulátorů pracovat jen s napětím a ve výsledné aplikaci případně provést přepočet na úhel, vynásobením napěťového údaje konstantou, která činí 18 v případě hlavního rotoru a 54 v případě ocasního rotoru.

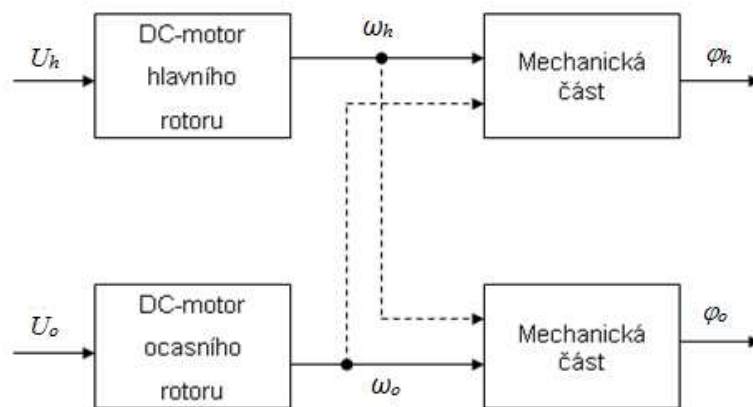
Před začátkem měření je nutná kalibrace čidel. Po zapnutí a nastartování řídicí jednotky je potřeba otočit modelem k oběma vertikálním i horizontálním dorazům, na kterých jsou umístěny kontakty. V okamžiku kontaktu na dorazu řídicí jednotka změří aktuální napětí odpovídající počáteční poloze, například u ocasního rotoru, signálu 0 V odpovídá 0° , u druhého kontaktu v ideálním případě 5V odpovídá 270° . Po dotyku dorazů je kalibrace hotova a následné dotyky kontaktů na dorazech nemají žádný vliv na měření.

1.5 Předpoklady o chování systému

Skutečná helikoptéra otáčkami hlavního rotoru řídí výšku a naklápěním listů hlavního rotoru řídí směr pohybu. Ocasní rotor zde slouží jen jako kompenzátor poruchy, která vzniká vlivem odstředivé síly hlavního rotoru a způsobila by roztočení celé helikoptéry, nestabilitu a následný pád. Ocasní rotor působí proti roztočení helikoptéry a udržuje jí v klidu.

Laboratorní model otáčkami hlavního rotoru řídí pouze výšku. Ocasní rotor zde může sloužit ke změně směru, ale také k potlačení poruchy, protože hlavní rotor svou odstředivou silou, stejně jako u skutečné helikoptéry, vychyluje model i v horizontálním směru. Z hlediska automatizace se jedná o systém MIMO, protože změna otáček hlavního rotoru (výška) způsobuje v klidovém stavu ocasního rotoru změnu polohy ocasního rotoru (směr). Naopak, změna otáčení ocasního rotoru (směr) v klidovém stavu hlavního rotoru ovlivňuje polohu hlavního rotoru (výška).

Na obrázku 1.4 je blokové schéma systému s naznačenou křížovou vazbou vzájemně se ovlivňujících členů. Kde U_h a U_o jsou řídicí napětí hlavního a ocasního motoru, ω_h a ω_o jsou jejich otáčky, φ_h a φ_o jsou úhly natočení ve vertikálním a horizontálním směru.



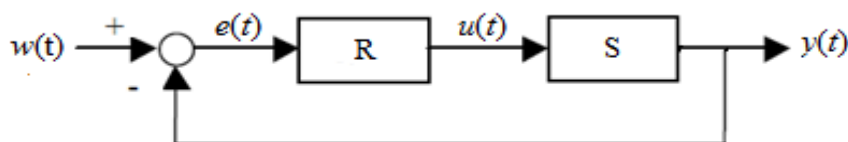
Obrázek 1.4 – Blokové schéma dvourotorového systému (HORÁK, 2011)

Reálné helikoptéry, analogicky i modely, vykazují nelineární chování v pracovním rozsahu. Nalezení parametrů regulace je tím pádem ztíženo a při navrhování regulátorů je nutno postupovat opatrně a s omezeným využitím lineárních přístupů, vždy v určitém pracovním bodě. V této fázi seznamování se s dvourotorovým systémem je tedy předpokládáno jeho nelineární chování. Kvůli nelinearitě je také ztíženo využití přístupů pro MIMO systémy a je potřeba využít spíše jednodušší typy regulátorů, které si poradí s velkými poruchami a nedokonalostí experimentálního modelu, vzniklou právě kvůli nelinearitám. Zde je prostor pro PID regulátory, které budou vhodně regulovat a vhodně potlačovat poruchy.

2 Řízení a regulace

Řízení je proces, při kterém řízená soustava reaguje na nějaké pokyny. Řízení se rozděluje na zpětnovazební a bez zpětné vazby. Řízení bez zpětné vazby může být použito jen v případě, že nehrozí nestabilita. Například v případě přivedení kapaliny do nádrže a znalosti času, za který se naplní, lze za tuto dobu přívod odpojit. Zpětnovazební řízení je například sledování hladiny nádrže a otevírání/zavírání kohoutku podle stavu hladiny. To ale za účasti člověka.

Regulace je proces, kdy se řízení provádí automaticky, bez účasti člověka. K tomu slouží zařízení zvané regulátor. Zapojení regulátoru a soustavy může být realizováno jako otevřený regulační obvod anebo uzavřený regulační obvod se zpětnou vazbou. Právě zpětná vazba je nejdůležitějším aspektem regulačního obvodu a právě ta umožňuje automatickou regulaci. Na obrázku 2.1 je zobrazen uzavřený regulační obvod. Na vstup obvodu je přivedena žádaná hodnota $w(t)$ a od ní se odečítá regulovaná veličina $y(t)$. Jejich rozdíl se nazývá regulační odchylka $e(t)$, která je přivedena na vstup regulátoru R. Regulátor vytvoří akční zásah $u(t)$, který působí na regulovanou soustavu S.



Obrázek 2.1 – Uzavřený regulační obvod

Regulace má 3 základní účely:

- Sledování žádané hodnoty $w(t)$.
- Potlačení poruch.
- Stabilizace soustavy v případě její přirozené nestability.

Ve všech požadavcích regulace hraje roli následující vztah, který je výsledkem zpětné vazby:

$$e(t) = w(t) - y(t) \quad (2.1)$$

Je-li regulační odchylka $e(t)$ rovna nule, výstup soustavy $y(t)$ se ustálil na žádané hodnotě $w(t)$. Změnou žádané hodnoty se změní výstup soustavy a tím regulační odchylka. Regulátor zapůsobí na soustavu a ustálí jí na nové žádané hodnotě.

Nutností úspěšného řízení je stabilita uzavřeného regulačního obvodu. K vyhodnocení kvality regulačního pochodu existuje množství kritérií (např. doba regulace, velikost regulační odchylky, kvadratická kritéria regulační odchylky). Na základě těchto kritérií lze rozlišit mezi nastavením parametrů regulátorů a zvolit ten nejlepší.

2.1 Regulátory PID

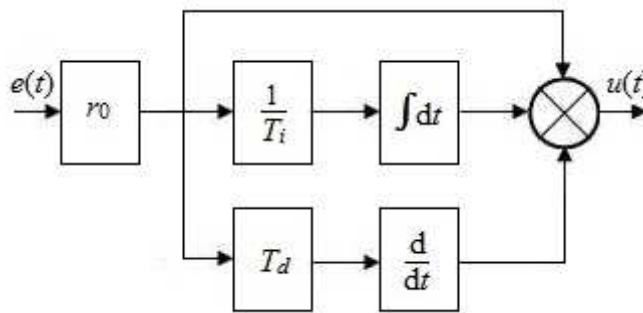
PID regulátor je složen z 3 částí:

- Proporcionální složky – zesílení regulátoru.
- Integrovační složky – její vliv způsobuje nulovou regulační odchylku.
- Derivační složky – urychluje regulační děj, ale musíme ji volit opatrně.

PID regulátory existují ve 3 formách (sériové zapojení, paralelní zapojení, standardní zapojení). Nejjednodušší zapojení regulátoru je standardní zapojení, které je charakterizováno rovnicí:

$$u(t) = r_0 \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (2.2)$$

kde konstanty r_0 , T_i a T_d jsou hledané parametry regulátoru, které se spočítají s použitím různých metod.



Obrázek 2.2 – Standardní zapojení PID regulátoru

Existuje celá řada metod, kterými lze nalézt parametry regulátoru. Mezi nejzákladnější metody nastavení parametrů regulátoru pro nekmitavé soustavy, patří analýza přechodové charakteristiky regulované soustavy a určení časových konstant (doba náběhu a doba průtahu) a zesílení soustavy, ze kterých se vypočítají parametry regulátoru.

Velmi oblíbenou a nejnámější metodou, která je také využita v experimentální části této práce, je vypočítání parametrů metodou Ziegler-Nichols z kritických hodnot regulované soustavy. Kritickými hodnotami regulované soustavy jsou kritické zesílení r_{0k} a kritická perioda T_k . Jsou to hodnoty, na kterých soustava kmitá konstantními kmity s konstantní amplitudou na hranici stability. Tyto hodnoty lze získat dvěma způsoby.

- Zapojení P regulátoru ve zpětné vazbě – postupnými změnami zesílení P regulátoru se ručně nalezne zesílení R, na kterém se uzavřený regulační obvod dostane na mez stability a rozkmitá se s konstantní amplitudou.
- Zapojení relé ve zpětné vazbě – relé se přepíná (tvoří obdélníkový průběh) a vybudí soustavu, která začne kmitat na mezi stability.

Z průběhu kmitů s konstantní amplitudou a konstantní periodou se získá přímo pouze kritická perioda T_k . Kritické zesílení r_{0k} je potřeba přepočítat pomocí kritické amplitudy e_{\max} , která se získá z průběhu kmitů. K přepočtu na kritické zesílení r_{0k} slouží vztah:

$$r_{0k} = \frac{4R}{\pi e_{\max}} \quad (2.3)$$

kde R je nalezené zesílení ve zpětné vazbě.

K vypočítání skutečných parametrů regulátorů slouží vztahy uvedené v tabulce 1.

Tabulka 2.1 – Výpočet parametrů regulátorů metodou Ziegler-Nichols

Regulátor	r_0	T_i s	T_d s
P	$0,5r_{0k}$		
PI	$0,45r_{0k}$	$0,85T_k$	
PID	$0,6r_{0k}$	$0,5T_k$	$0,125T_k$

Metoda relé ve zpětné vazbě (nebo P regulátor ve zpětné vazbě) je použitelná pro soustavy minimálně II. řádu. Někdy je problém rozkmitat soustavu II. řádu, pokud se přechodová charakteristika blíží soustavě I. řádu. Tato metoda funguje velice dobře od III. řádu soustavy. Metoda Ziegler-Nichols se považuje za poměrně agresivní metodou nastavení parametrů s překmitem. Na některých soustavách může způsobit nestabilitu, proto může nalezení parametrů touto metodou sloužit jako záchytný bod a následně parametry upravit pomocí experimentů, aby se dosáhlo stability uzavřeného regulačního obvodu.

2.2 Rozdělení soustavy MIMO na samostatné větve SISO

Soustava se dvěma vstupy a dvěma výstupy má obecnou přenosovou funkci (matici):

$$G_s(s) = \begin{pmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{pmatrix} \quad (2.4)$$

kde $G_s(s)$ Přenos soustavy
 $G_{11}(s)$ Přenos prvního vstupu a výstupu (vertikální rotor)
 $G_{22}(s)$ Přenos druhého vstupu a výstupu (horizontální rotor)
 $G_{12}(s)$ a $G_{21}(s)$ Přenosy vyjadřující vzájemnou interakci mezi prvky soustavy

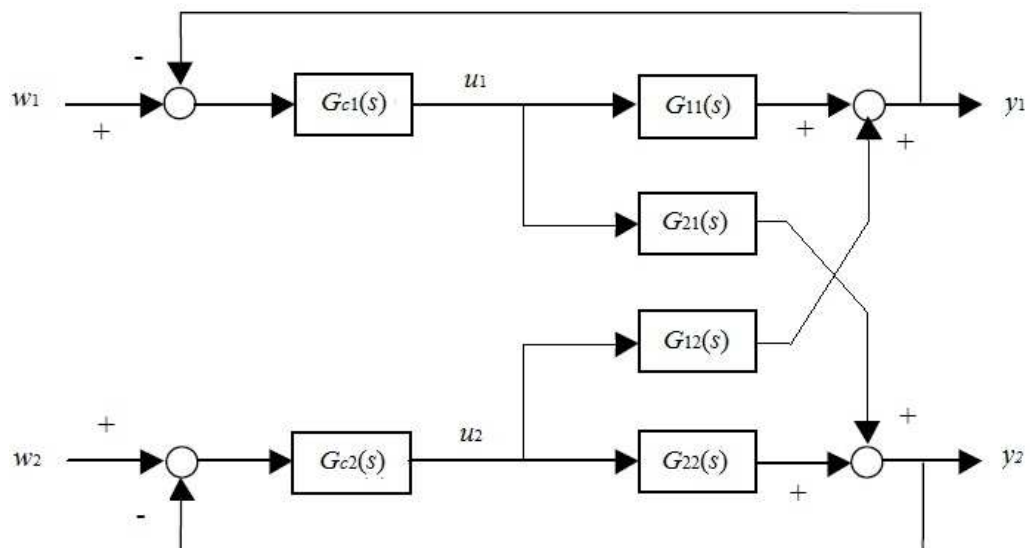
Takovouto soustavu lze řídit decentralizovaným způsobem řízení. Tento způsob využívá k řízení každého prvku soustavy vlastní regulátor (dva samostatné regulátory).

Decentralizovaný regulátor má přenosovou funkci (diagonální matici):

$$G_c(s) = \begin{pmatrix} G_{c1}(s) & 0 \\ 0 & G_{c2}(s) \end{pmatrix} \quad (2.5)$$

kde $G_c(s)$ Přenos decentralizovaného regulátoru
 $G_{c1}(s)$ a $G_{c2}(s)$ Přenos jednotlivých regulátorů

Regulátor, který řídí vlastní prvek soustavy, svým akčním zásahem ovlivňuje i stav druhého prvku (například změna otáček ocasního rotoru mírně změní úhel natočení hlavního rotoru). Na obrázku 2.3 je znázorněno blokové schéma uzavřeného decentralizovaného regulačního obvodu, kde w_1 a w_2 jsou žádané hodnoty, u_1 a u_2 jsou akční veličiny, y_1 a y_2 jsou regulované veličiny. V případě, že vzájemná interakce mezi prvky je značná, musí se brát tato skutečnost v úvahu. Na obrázku 2.3 jsou viditelné vzájemné interakce.



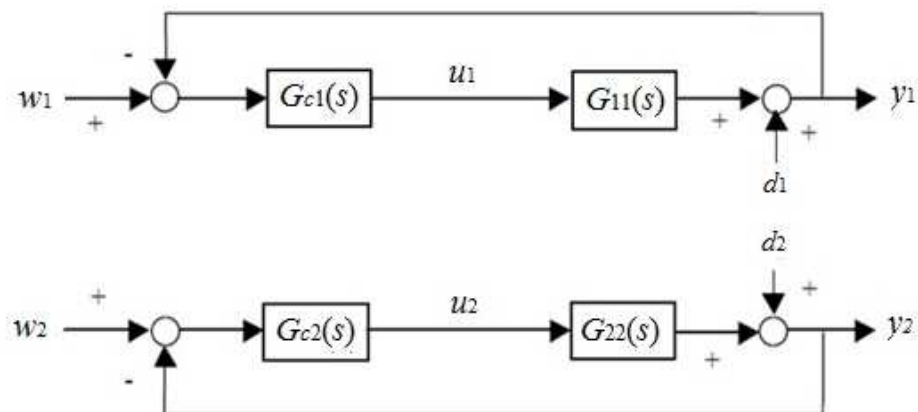
Obrázek 2.3 – Blokové schéma decentralizovaného regulačního obvodu (MACHÁČEK, 2008)

Z obrázku 2.3 je zřejmé, že pro návrh parametrů prvního regulátoru G_{c1} se musí uvažovat přenosová funkce mezi prvním vstupem a prvním výstupem, která obsahuje i druhou větev obvodu včetně regulátoru G_{c2} (pro $w_2 = 0$). Totéž platí i pro návrh druhého regulátoru. (MACHÁČEK, 2008)

Pokud je interakce mezi jednotlivými prvky velká a lze ji měřit, lze pomocí dalších regulátorů (ve vedlejších větvích) odstranit vliv křížových vazeb. Tyto regulátory zasahují opačným znaménkem (odečtením akční veličiny křížové vazby od hlavní větve akční veličiny) a vykrývají vlivy křížových vazeb. Tento postup je ale složitý. Identifikace takové soustavy a nastavování maticového regulátoru by mohlo být v rozsahu další diplomové práce,

proto je v této práci nutné zavést zjednodušení. Jsou-li interakce menší, lze je zanedbat a vytvořit diagonální přenosovou matici s nulovými prvky na vedlejší diagonále. Aby se co nejméně projevil vlivy interakcí, je potřeba nastavit regulátory s pomalejším regulačním pochodem. Pomalý regulační pochod je pak méně náchylný na poruchy a hledá se takové nastavení parametrů, kdy celý uzavřený regulační obvod zůstane stabilní i přes malé interakce.

Experimentální model v podobě přenosových funkcí je pouze aproximací s malou přesností. Počítá se s tím, že řízení s použitím PID regulátorů si s nedokonalostmi modelu poradí a stabilita URO bude zachována. Vyjmutím přenosových funkcí vzájemné interakce ($G_{12}(s)$ a $G_{21}(s)$) se z decentralizovaného popisu stanou dvě samostatné soustavy, ale umožní jednodušší návrh regulátorů. Při návrhu regulátorů se sice zanedbá vzájemná interakce, ale poruchové vlivy jsou simulovány přivedením poruchy na výstup soustavy. Na obrázku 2.4 je znázorněno blokové schéma rozděleného systému na dvě samostatné části s přivedenými poruchami d_1 a d_2 na výstup systému. Poruchy na výstupu sice nevykazují takové chování, jako kdyby se dva regulátory svými vnitřními vazbami vzájemně ovlivňovali, ale umožní nalézt takové parametry regulátorů, které by mohly být schopny odstraňovat poruchy. Při zachování pomalejších regulačních pochodů URO zůstane stabilní.



Obrázek 2.4 – Blokové schéma rozděleného regulačního obvodu s poruchami na výstupu

3 Měřicí karta NI USB 6008/9

K získávání dat a řízení systému je využita měřicí karta firmy National Instruments NI USB 6008/9. Zde je odpověď na to, proč jsou rozsahy pro měření a ovládání z dvourotorového systému v rozmezí 0 – 5 V. Karta je k řídicímu počítači připojena přes USB sběrnici, přes kterou se zároveň napájí. Napětí USB sběrnice je 5 V, napájecí napětí pro kartu je dané napětím sběrnice. Proto také A/D převodník v kartě má rozlišení 5 V.

Karta má 8 analogových vstupů a 2 analogové výstupy. Dále karta disponuje 8 digitálními vstupy a 8 digitálními výstupy. Pro operace s dvourotorovým systémem jsou využity 2 analogové výstupy pro ovládání motorů a 2 analogové vstupy, které obsahují informace z inkrementálních snímačů.

Napětí analogových vstupů je převáděno na digitální signál s rozlišením 12 nebo 14 bitů, které lze nastavit přístupem na kartu pomocí dodaného softwaru. Rozlišení výstupů je pouze 12 bitů. Toto rozlišení plně postačuje požadavkům, protože přesnost snímačů je 1°. Rozlišení 12 bitů umožňuje 8096 kroků z 5 V. Pro dvourotorový systém by bylo postačující rozlišení 256 kroků, což odpovídá 8 bitům, protože maximální rozsah je 270° pro ocasní rotor. Maximální vzorkovací frekvence karty je 48 kHz u modelu 6009 (10 kHz u modelu 6008), což bohatě postačuje k operacím s dvourotorovým systémem. Napěťový výstup karty umožňuje měnit hodnoty rychlostí maximálně 150 Hz (řídicí signál se může změnit maximálně 150 krát za sekundu).



Obrázek 3.1 – Měřicí karta NI USB

Tato měřicí karta patří mezi jedny z nejlevnějších řešení pro měření a řízení od firmy NI, ale k laboratorním úlohám na KŘP je takřka ideální, protože má výhodu přenositelnosti mezi stolními počítači nebo notebooky. Výhoda přenositelnosti a připojitelnosti přes USB k jakémukoliv počítači by mohla zastínit nevýhodu napájení a rozlišení A/D převodu 5 V.

Karta a její ovladače jsou kompatibilní téměř se všemi verzemi operačního systému Windows. Ovladače jsou dodávány spolu s kartou nebo je lze stáhnout z webu výrobce. Jedná se ale o velmi velký balík dat, ve kterém výrobce zahrnuje podporu k obrovskému množství hardwarových prostředků jeho produkce. Na druhou stranu instalací tohoto balíku se nainstaluje aplikace NI Device Monitor, která umožňuje s kartou pracovat stylem Plug&Play. To znamená, že není potřeba restart systému. Stačí kartu jednoduše připojit a Device Monitor si kartu sám najde. Dokonce vytvoří tréninkové spojení a spustí další aplikaci, pomocí které lze snadno zajistit správné propojení a fungování karty, nastavování výstupů, pojmenování výstupů nebo pojmenování samotné karty v případě, že je k počítači karet připojeno více. Tato aplikace se nazývá Test Panels. Nicméně k další práci s kartou je vhodné aplikaci Test Panels vypnout, aby se uvolnil přístup na hardware (samotnou kartu) z jiného softwarového produktu.

S těmito kartami se na KŘP pracuje s různými laboratorními systémy. Je realizováno propojení s prostředím MATLAB, ale MATLAB je náročný na hardware počítače a řízení v reálném čase je možné provádět jen s velmi malým vzorkováním, nejvíce 10 Hz. K plynulému běhu počítače ale spíše 1 Hz. Takto malé vzorkování je pro současné průmyslové aplikace příliš malé. Zvyšujícím se vzorkováním počítač nestíhá vypočítávat regulaci a dochází ke zpoždění, případně kolapsu programu nebo operačního systému. Takovýto kolaps by měl v průmyslu katastrofální následky.

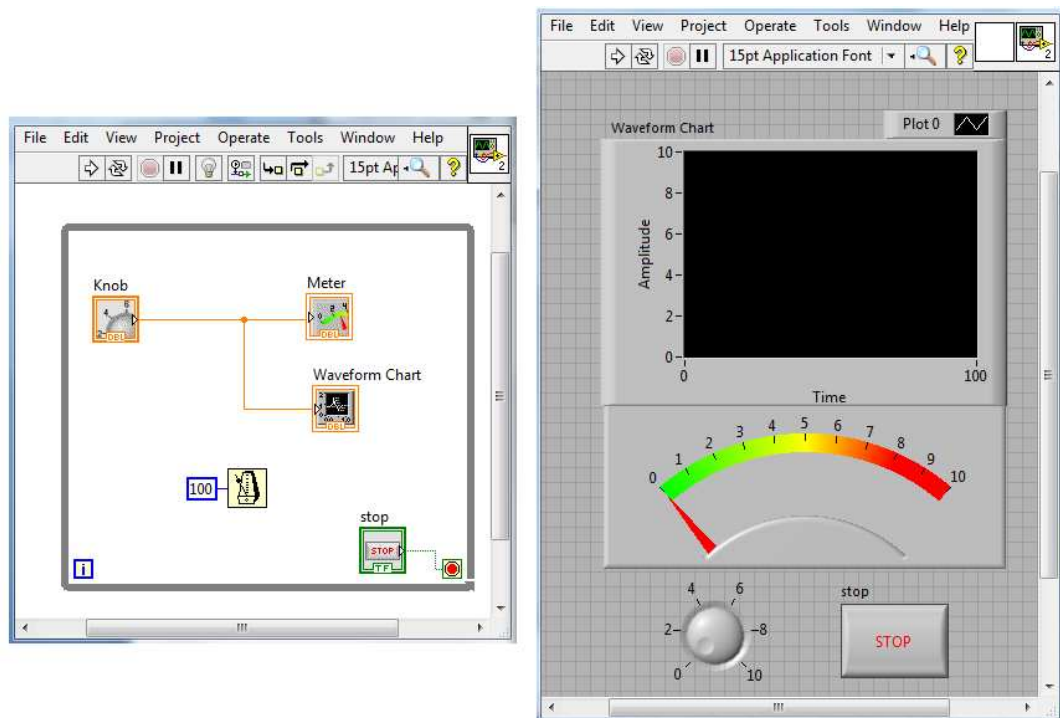
Pro řízení dvourotorového systému bylo vybráno průmyslové vývojové prostředí LabVIEW, také od firmy NI, které umožňuje pracovat se vyšším počtem vzorků než MATLAB. Podle výrobce by program, vytvořený v grafickém jazyce G, měl mít vlastnosti programu napsaného v jazyce C. Tím je myšlena optimalizace překladu a rychlost běhu programu. Proto je vhodnější na aplikaci v reálném čase použít místo MATLABu prostředí LabVIEW.

4 Vývojové prostředí LabVIEW

LabVIEW je profesionální nástroj firmy National Instruments, který lze využívat k řešení obrovského množství problémů. Své uplatnění nalezne jak v elektrotechnice, zpracování signálů, tak samozřejmě i v automatizaci a řízení. V LabVIEW lze vytvořit rozsáhlé řídicí systémy a pod systémy, které spolu komunikují. Základní verze obsahuje prvky, které jsou použitelné spíše pro simulace a prvky pro řešení jednodušších problémů různých oborů. Dokoupením a doinstalováním takzvaných Toolboxů lze docílit získání dalších nástrojů a předdefinovaných řešení pro daný obor. Firma NI podporuje velké množství vlastních i cizích měřicích přístrojů a z vývojového prostředí umožňuje programátorům přístup na všemožný hardware, dokonce i komunikaci přes internet.

4.1 Filozofie LabVIEW - Virtual Instrument

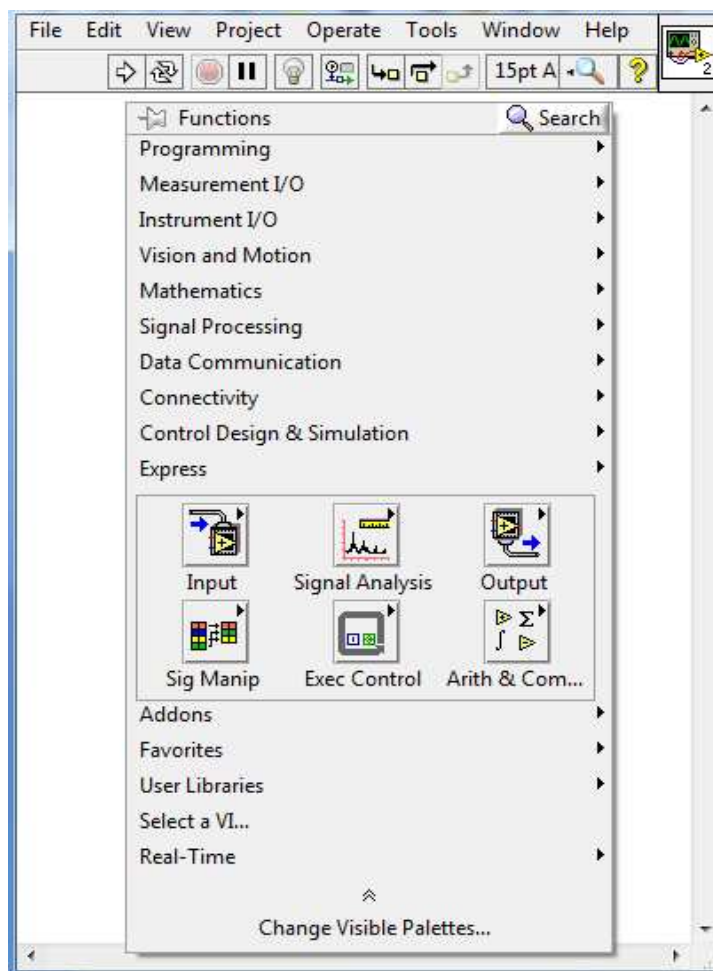
Jak již naznačuje název, v LabVIEW se vytváří aplikace a projekty, kdy výsledná aplikace vypadá jako měřicí přístroj nebo ovládací panel, takzvaný Virtual Instrument, dále VI. Tvorba aplikace je rozdělena na vnitřní část blokového diagramu (Block Diagram), která po vytvoření aplikace není vidět a vnější část předního panelu (Front Panel), která slouží uživateli. Je to podobné jako u různých programovacích jazyků, kdy kód programu samozřejmě uživatele nezajímá, uživatel komunikuje s programem přes formuláře a uživatelská prostředí. Blokovaný diagram a přední panel spolu komunikují. Přední panel dodává dovnitř data, které zadá uživatel, zevnitř jsou data (například z připojeného měřicího přístroje) odesílána do zobrazovačů a indikátorů na předním panelu. Na obrázku 4.1 je pro ilustraci zobrazen příklad blokového diagramu a jeho předního panelu, kde je umístěno ovládání a zobrazování.



Obrázek 4.1 – Blokovaný diagram (vlevo), čelní panel (vpravo)

4.2 Vnitřní část – blokový diagram

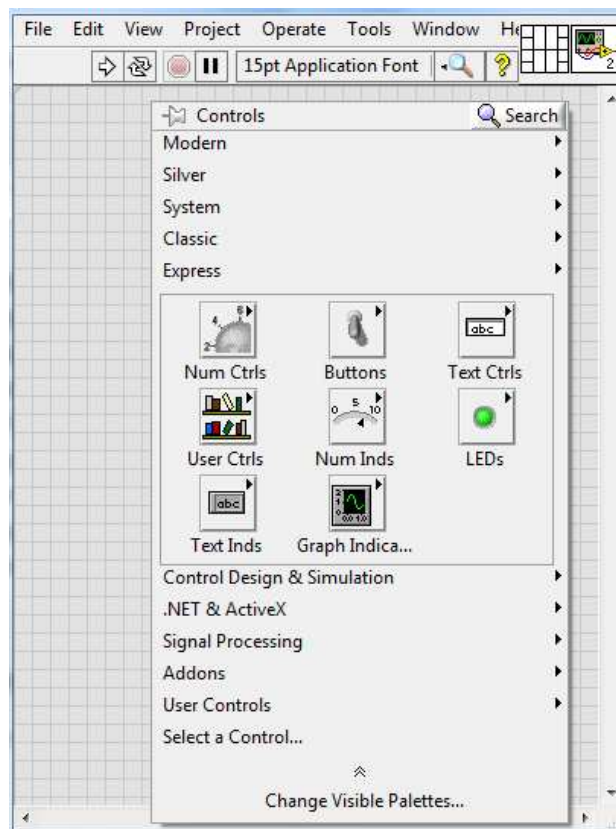
Vnitřní část slouží pro konstrukci programu. Konstrukcí je myšleno to, že program se nepíše strukturovaným textem, ale skládá se spojováním funkčních bloků pomocí „drátových spojů“ do blokových diagramů splňujících požadovanou funkčnost. Funkční bloky mohou reprezentovat například softwarový přístup na hardware nebo jsou to matematické či logické funkce. Každý funkční blok je umístěn v knihovně funkcí. Knihovny jsou logicky rozděleny na určité okruhy funkcí a funkční bloky jsou vybírány ze seznamu a následně umisťovány do blokového diagramu. Programátor přistupuje k funkcím pomocí palety *Functions* v okně Block Diagram, která se objeví po kliknutí pravým tlačítkem myši. Tento systém programování se nazývá jazyk G, od slova Graphic, což znamená grafické programování. Je to vlastně nadstavba nad strukturovaným textem. Pod každým blokem se skrývá jeho naprogramovaná funkčnost a po dokončení blokového diagramu je při spuštění program standardně přeložen vlastním kompilátorem. Výrobce udává, že rychlost běhu programu se po přeložení přibližuje rychlosti programů napsaných v jazyce C. To je velkou výhodou, protože skládání blokových diagramů silně spoří programátorův čas. Využité bloky a jejich význam budou popsány až v praktické části tvorby aplikace.



Obrázek 4.2 – Paleta funkcí v blokovém diagramu

4.3 Vnější část – přední panel

Přední panel slouží jako interface mezi programem a uživatelem. Na přední panel se umisťují ovládací prvky (tlačítka, různé potenciometry) a zobrazující prvky (grafy, diody). Jak již bylo zmíněno, přední panel s blokovým diagramem komunikuje způsobem, že „drátovými spoji“ jsou prvky předního panelu a blokového diagramu spojeny. Prvky předního panelu jsou totiž reprezentovány i jako bloky v blokovém diagramu a vnitřní struktura, která například měří nebo počítá nějaký problém, zobrazí vypočtený výsledek přivedeným „drátovým spojem“ na prvek předního panelu. Přední panel také obsahuje knihovny ovládacích prvků, které se nazývají *Controls*. Programátor k nim přistupuje kliknutím pravým tlačítkem myši do prostoru okna Front Panel.

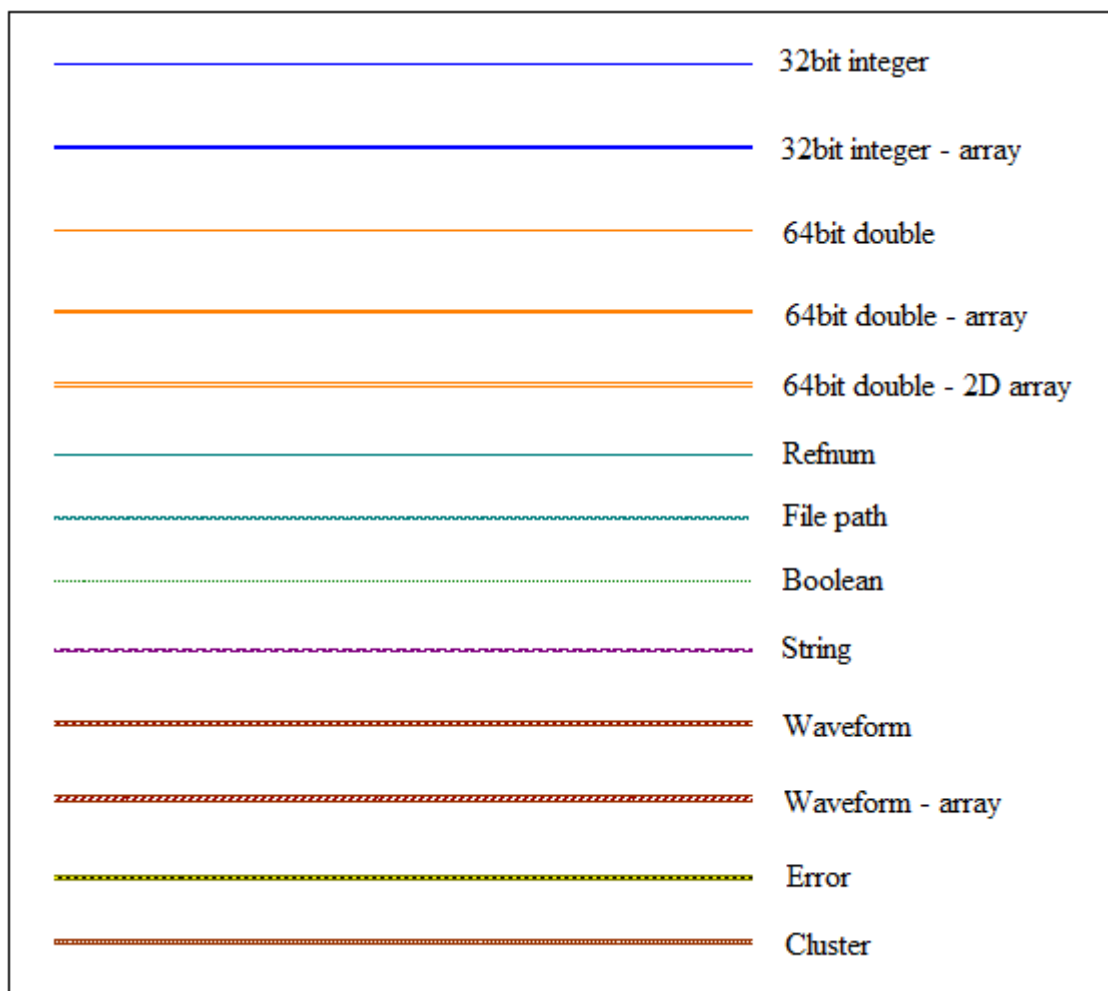


Obrázek 4.3 – Paleta ovládacích prvků v předním panelu

4.4 Datové typy

LabVIEW, stejně jako veškeré programovací jazyky, rozlišuje různé datové typy. Navíc obsahuje řadu speciálních datových typů, které lze příhodně využívat při práci s velkým množstvím dat, při měření nebo řízení. Datové typy, které si lze dobře představit, jsou např. integer nebo double. Hůře si lze představit datový typ cluster. Cluster je buňka, která může zároveň obsahovat prvky různých datových typů. Lze vytvořit i jejich kombinace, jako je například vektor clusterů. Dalšími speciálními a velmi důležitými datovými typy jsou waveform a error.

Jednotlivé bloky se dají propojit různými vodiči (wires). Právě každý vodič reprezentuje daný datový typ a každý blok lze propojit takovým datovým typem, se kterým umí pracovat. V LabVIEW jsou datové typy rozlišeny barevnými vodiči. Na obrázku 4.4 jsou zobrazeny datové typy, které jsou využity při řízení dvourotorového systému. Existuje řada dalších datových typů, ale není potřeba je všechny zmiňovat.



Obrázek 4.4 – Datové typy

Popis jednotlivých datových typů:

- 32bit integer – 32 bitové číslo, modrá barva.
- 32bit integer array – pole 32 bitových čísel, zesílená modrá barva.
- 64bit double – 64 bitové číslo, oranžová barva.
- 64bit double array – pole 64 bitových čísel, zesílená oranžová barva.
- 64bit double 2D array – 2D pole 64 bitových čísel, dvojitá čára, oranžová barva.
- Refnum – speciální datový typ, identifikátor souboru, světle modrá barva.
- Boolean – logický datový typ true/false, zelená barva.
- String – řetězec, důležitý datový typ pro reálné řízení, růžová barva.
- Waveform – speciální datový typ. Vícerozměrné pole obsahující vektor času, vektor měřených dat (velikost podle měřených veličin), hnědá barva.
- Waveform array – pole waveform datových typů, už složitější konstrukce. Z takto složené struktury je potřeba správně vybírat vektor dat, hnědá barva.
- Cluster – struktura, která může obsahovat elementy různých datových typů i polí, hnědá barva.
- Error – speciální cluster obsahující 3 prvky. Využíván v real-time měření a řízení, obsahuje real-time informace o stavu hardwaru a správném průběhu měření. Důležitý je první prvek, logická hodnota true/false. V případě přepnutí logické hodnoty (podle nastavení) ukončí běh programu. Žlutá barva.

LabVIEW umožňuje ve většině případů vytvořit ze všech možných datových typů jejich pole a vícerozměrná pole. Na obrázku 4.4 je pouze výběr těch datových typů, které se vyskytují v programech na identifikaci a řízení systému.

V seznamu datových typů není rozlišeno, jestli se jedná například o 32 bitové číslo se znaménkem nebo bez znaménka. To, jestli bude číslo jen kladné nebo záporné, lze nastavit ve vlastnostech daného bloku.

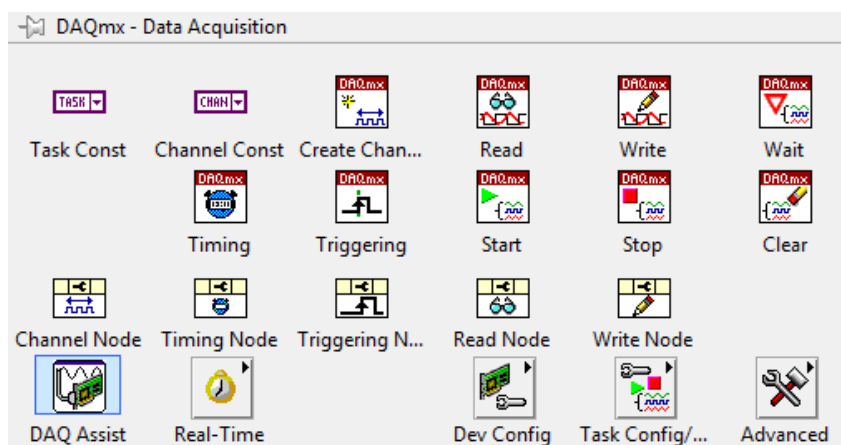
5 Oživení laboratorního systému

V druhé kapitole je dvourotorový laboratorní systém popsán z hlediska pozorování v nečinném stavu a z dostupné literatury jsou vybrána důležitá fakta, jejichž znalost je nezbytná pro tvorbu řídicího programu. Následujícím úkolem je systém oživit, zajistit komunikaci s řídicím počítačem a otestovat správné propojení. K získání informací o dynamice je nejen důležité ovládání a pozorování, ale právě měření odezev a ukládání změřených dat, která dále poslouží k návrhu regulátorů.

V rámci tvorby těchto programů je vysvětlen jejich princip a také použité bloky, aby opět čtenář porozuměl funkci programu a neviděl schéma jako změť barevných čar.

5.1 Propojení s počítačem

Propojení s počítačem je realizováno pomocí karty NI USB 6009. V LabVIEW jsou ke komunikaci s měřicí kartou speciální bloky, které ji reprezentují. Programátor tedy na kartu nahlíží jako na skupinu bloků. Jako v každém bloku, i v blocích reprezentujících skutečný hardware, se skrývají naprogramované struktury, které zajišťují komunikaci počítače a měřicí karty. Tyto bloky nalezneme v paletě *Functions > Measurement I/O > NI-DAQmx*, v blokovém diagramu.

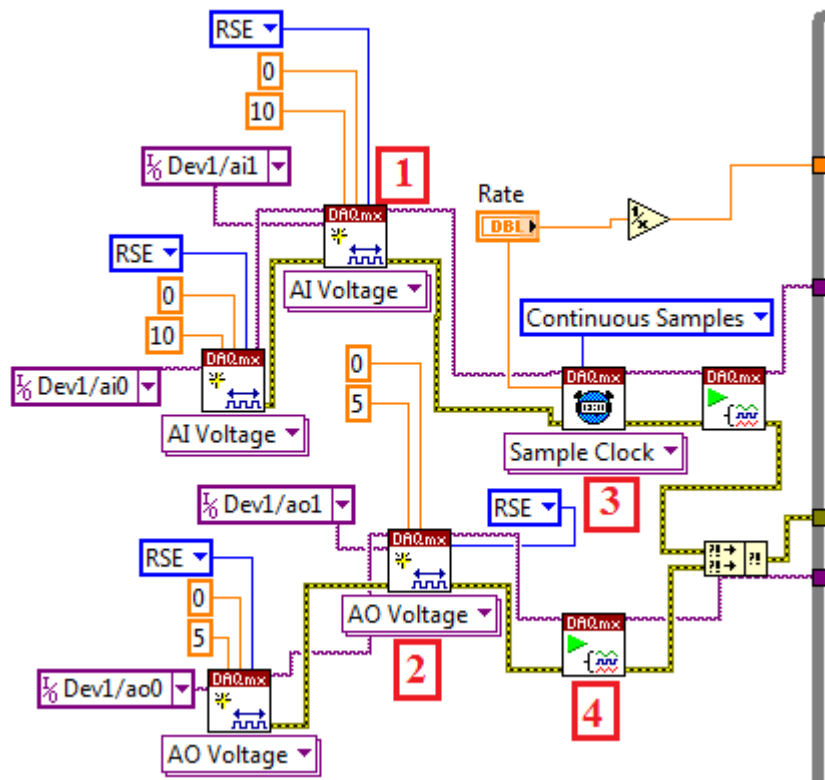


Obrázek 5.1 – Paleta NI – Data Acquisition

Pomocí prvků palety *NI-DAQmx* tedy přivedeme data z karty do prostředí LabVIEW. Dalším nezbytným prvkem pro reálné ovládání a měření je prvek *While Loop*. Je alternativou k běžné řídicí struktuře v programovacích jazycích, cyklu *while*. U cyklu *while* předem nedefinujeme počet opakování, proto je vhodný do aplikace pracující v reálném čase, kdy je potřeba vyhodnocovat jednotlivé vzorky dat, ale samozřejmě není znám jejich počet, protože aplikace běží, dokud není přerušena chybou nebo zastavena uživatelem. V LabVIEW cyklus *While Loop* reprezentuje rámeček, který obsáhne všechny prvky, které jsou jeho součástí. Vše, co leží v rámečku *While Loop*, je nadále cyklicky vyhodnocováno. Počet opakování řídí vzorkovací frekvence, jejíž volbu si musí programátor rozmyslet. Měřicí karta umožňuje vzorkovat až 48 kHz, ale v řídicím programu je nastavena na 1KHz, což je dostačující pro běžný počítač s Windows. Počet cyklů se dá zaznamenávat do proměnné „i“, která je součástí

cyklu a lze s ní pracovat při výpočtu doby měření v ovládací aplikaci. Cyklus *While Loop* nalezneme v paletě *Functions > Programming > Structures > While Loop*.

Na následujících obrázcích je zobrazeno blokové schéma ovládacího programu, rozděleného na část pro nastavení komunikace s kartou a část cyklickou. Všimněme si cyklické části, která je součástí *While Loop*.



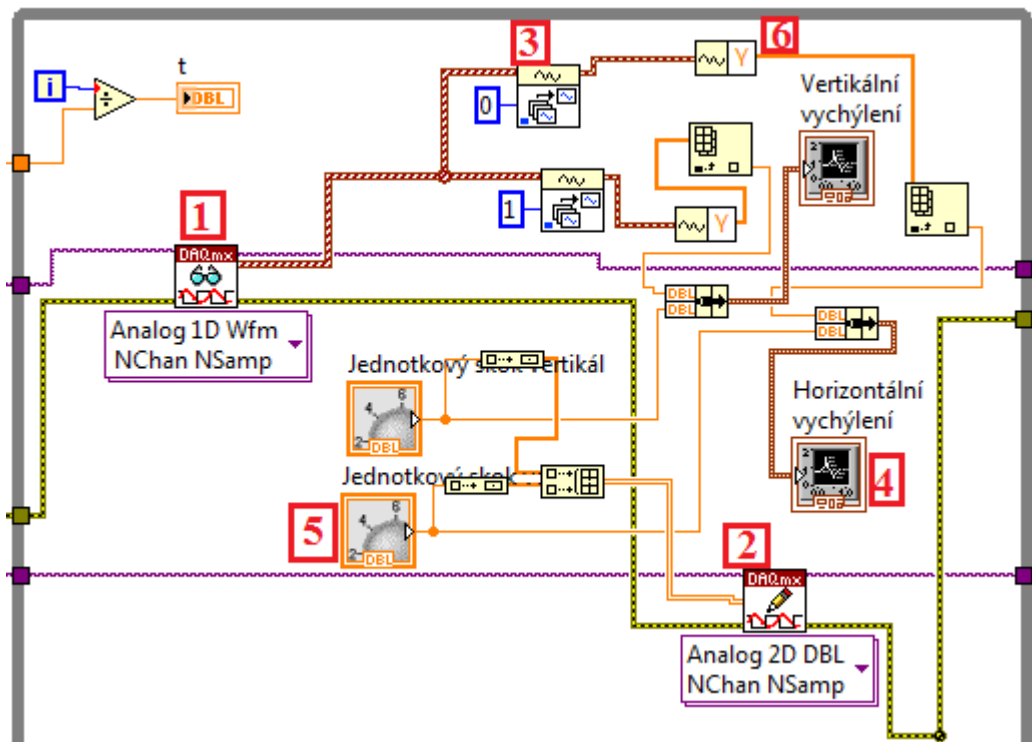
Obrázek 5.2 – 1. část programu – přístup na hardware měřicí karty

Popis vybraných nejdůležitějších bloků 1. části programu:

1. DAQ mx Create Channel – blok, který zastupuje konfiguraci měřicí karty. Dalo by se říci blok, ve kterém určíme, co je přivedeno zvnějšku do karty. Karta má více vstupů, musí se vybrat ten aktuální. Pokud je k LabVIEW připojeno více karet, musí se rozlišit i mezi nimi.
2. DAQ mx Create Channel – stejný blok jako č. 1, ale nyní zastupuje výstup z měřicí karty. Dalo by se říci blok, který definuje, který ovládací signál se odešle do laboratorního systému.
3. DAQ mx Timing – blok, který tvoří z analogového signálu signál digitální. Na jeho vstup je přivedena vzorkovací frekvence (proměnná Rate), v tomto případě 1KHz.
4. DAQ mx Start Task – blok, který vytvoří takzvaný *Task*, specifikovaný v blocích č. 1 resp. 2.

Celá komunikace probíhá pomocí takzvaného *Task* signálu. Je reprezentován růžovým vodičem, tedy jedná se o řetězec dat. Pro práci s kartou je nutné dodržet spojení těchto bloků, protože série těchto bloků vytváří přístup na hardware karty. Není úplně pravda, že digitalizaci provádí blok *DAQ mx Timing*, protože digitalizaci a úpravu signálu provádí celý řetězec bloků.

Největší problém byl odladit tuto sérii bloků. Lze si jednoduše představit, že se vyberou bloky *DAQ mx Create Channel* a z každého bude vycházet jeden měřený signál, se kterým se bude dále pracovat. Toto v LabVIEW není možné, protože vytvoření *Task* znamená přístup na hardware karty a ten je možný pouze jeden jediný, tedy kanál lze vytvořit jen jeden. Proto jsou bloky vstupů a výstupů zapojeny vedle sebe a propojeny stejným růžovým signálem. Jsou zvláště zřetězeny bloky vstupů (měření polohy rotorů) a bloky výstupů (ovládací signály obou rotorů). Všechny měřené signály jsou zřetězené v jednom *Task* signálu a dále je potřeba z nich vybrat ten správný. To je již součástí druhé části blokového diagramu.



Obrázek 5.3 – 2. část programu – výběr dat z *Task* signálu, ovládání a měření

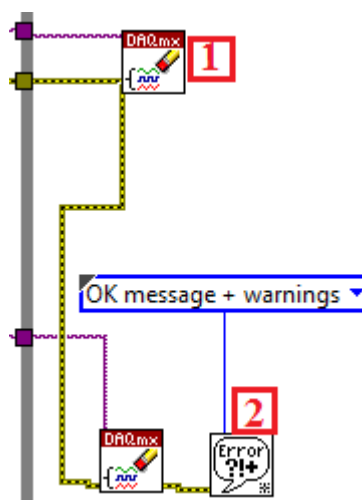
Tato část programu cyklicky vybírá data z řetězce a v reálném čase je zobrazuje ve dvou grafech, zároveň převádí data z ovládacích prvků a připravuje je na výstup. Znalosti vzorkovací frekvence a počtu opakování lze vypočítat dobu běhu programu.

Popis vybraných nejdůležitějších bloků 2. části programu:

1. *DAQ mx Read* – na základě přivedeného *Task* signálu čte z karty data a vytváří z nich pole waveform signálů. Tyto waveformy obsahují měřená data.

2. DAQ mx Write – na základě přivedeného Task signálu posílá data na kartu; je důležité přivést všechny ovládací signály najednou, zde je problém vyřešen vložím ovládacího signálu do 2D pole.
3. Index Waveform Array – blok, který se nalezne v paletě *Functions > Programming > Waveform*. Vybírá jednotlivé Waveform signály z pole Waveformů.
4. Waveform chart – graf, který má pevně danou časovou osu. Je na něj přiveden řídicí signál a měřené vychýlení rotoru, který řídí. Oba signály jsou zabaleny do datového typu cluster, se kterým blok pracuje. Vkládá se na předním panelu v paletě *Controls > Graph Indicators > Graph*.
5. Knob – ovládací prvek, přirovnatelný k softwarovému potenciometru. Má nastavitelný rozsah (v tomto případě 0-5V). Také se vkládá na předním panelu v paletě *Controls > Numeric Controls > Knob*.
6. Get Waveform Components – blok, který rozebere waveform signál a vybere z něj komponenty. V tomto případě byl vybrán přímo komponent pole, které obsahuje měřená data.

Poslední částí je potřeba ukončit Task a vyčistit kanál, aby se dal program spustit znovu. To je důležité také v případě, pokud by na měřicí kartu přistupovalo více programů, které na sebe vzájemně čekají. Tyto bloky již nejsou součástí cyklu *While Loop*, protože nepracují cyklicky, ale jednorázově vyčistí kanál po skončení programu.



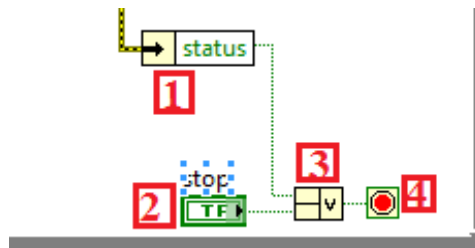
Obrázek 5.4 – 3. část programu – čistící rutina

Popis bloků 3. části programu:

1. DAQ mx Clear Task – vyčistí Task a uvolní kanál.
2. General Error Handler – dohlíží nad správnou funkcí programu a určuje logickou hodnotu chyb true/false. V případě chyby se program zastaví a ukáže chybové hlášení.

Ve všech částech programu se vyskytuje žlutý error signál. Je propojen s celým řetězcem bloků, které přistupují na hardware karty. Je to cluster obsahující 3 pole. Důležité pro tuto aplikaci je první pole, které obsahuje boolean signál, určující true/false stavu programu. Stav true/false řídí blok *General Error Handler*. Tento signál, pokud nastane chyba, může ukončit běh programu, pokud je přiveden na ukončovací tlačítko, které je součástí cyklu *While Loop*.

Program je také zapotřebí nějakým způsobem ukončit manuálně. K tomu je využito standardní tlačítko, které při stisknutí změní svou logickou hodnotu. Tlačítko je tedy typu boolean. Problém ukončení programu tlačítkem nebo chybou je vyřešen logickou funkcí „or“, => tlačítko nebo chyba. Vyhodnocení zastavení programu je také uvnitř cyklu *While Loop*.



Obrázek 5.5 – Ukončení programu

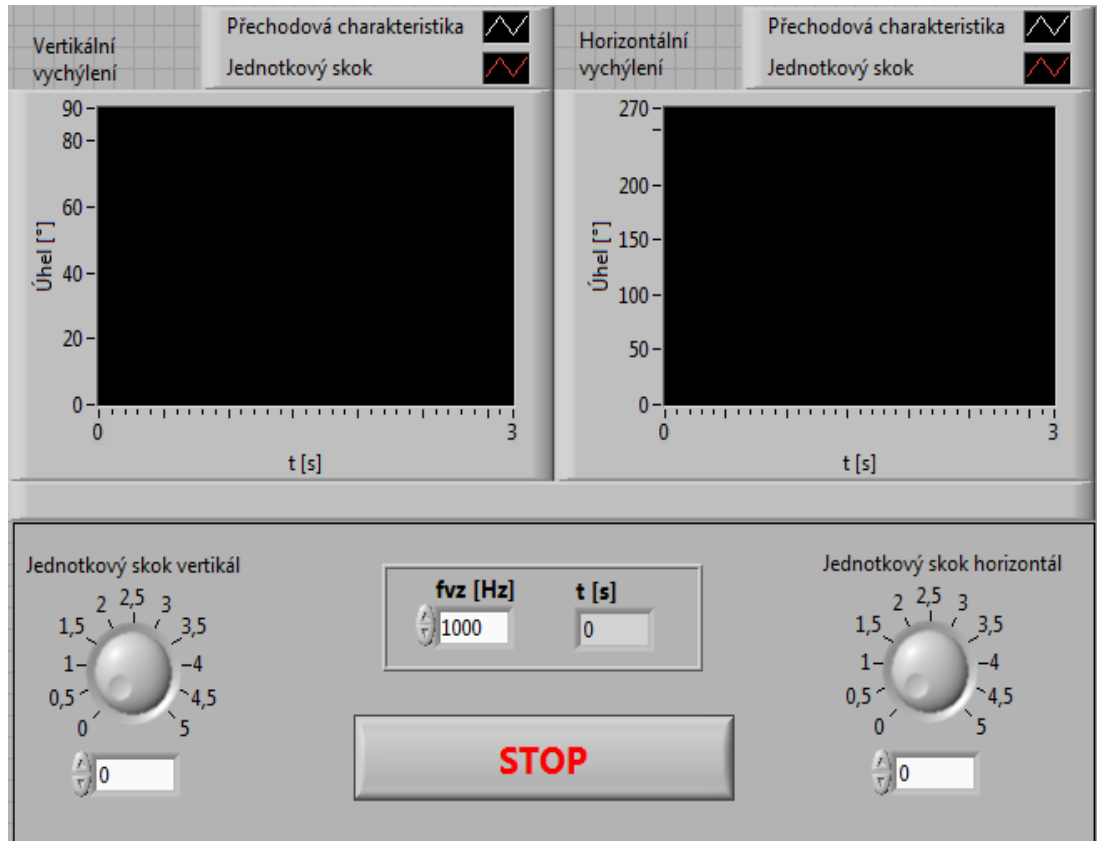
Popis bloků ukončení programu:

1. Unbundle – blok, který vybere z clusteru první pole logických hodnot
2. Tlačítko – standardní tlačítko, které má buď logickou hodnotu true nebo false. Vkládá se na předním panelu v paletě *Controls > Buttons > OK Button*.
3. Compound Arithmetic – blok, který zajistí funkci „or“. Blok umí i ostatní logické operace, stačí jej nastavit na tu požadovanou.
4. Přepínač, který je součástí cyklu *While Loop*. Je nastavitelný, ukončení běhu programu může být spuštěno jak hodnotou true, tak hodnotou false. Zde je nastaven na hodnotu „stop if true“.

Nyní je vysvětlena struktura programu a zbývá se zaměřit na čelní panel. Na čelním panelu jsou umístěny dva prvky zobrazovačů. Jak již bylo zmíněno, jedná se o typ *Waveform Chart*. Každý graf zobrazuje jednu řídicí a jednu měřenou veličinu. Každý rotor tedy má svůj zobrazovač úhlu natočení. Informace v podobě napětí 0 – 5 V je potřeba přepočítat na úhel. Toho se dosáhne vynásobením aktuálního měřeného vzorku konstantou, která rozšíří rozmezí 0 – 5 V na rozsah od -45° do $+45^\circ$ pro vertikální vychýlení a 0° až 270° pro horizontální vychýlení. Pro zjednodušení je rozsah horizontálního rozsahu programu upraven na 0° až 90° , aby bylo možné snadno zároveň zobrazovat i řídicí veličinu.

Na grafu je tedy zobrazována žádaná hodnota $w(t)$ a výstupní veličina $y(t)$, které jsou obě přepočítány na úhel vychýlení. Časová osa je omezena na zobrazování 3 sekund, což stačí na zobrazení přechodu z jednoho stavu do druhého stavu. Graf si musí pamatovat všechny vzorky po dobu 3 sekund a poté je přepíše daty dalších 3 sekund. Bylo by možné nastavit delší zobrazení, ale pak by nebyl viditelný tak detailní průběh přechodů a navíc by si musel

program pamatovat o mnoho více dat a tím by zatěžoval systém. Pod zobrazovači jsou umístěny dva ovládací potenciometry *Knob*, které lze ovládat myší, nebo do pole pod nimi zadat přesnou požadovanou hodnotu. Mezi potenciometry je umístěno pole pro zadávání vzorkovací frekvence a indikátor doby běhu programu. Pod nimi je umístěno tlačítko na ukončení programu. Na obrázku 5.6 je zobrazen čelní panel. Celý program je uživateli skrytý a uživatel pracuje jen s čelním panelem.



Obrázek 5.6 – Čelní panel aplikace pro oživení laboratorního systému

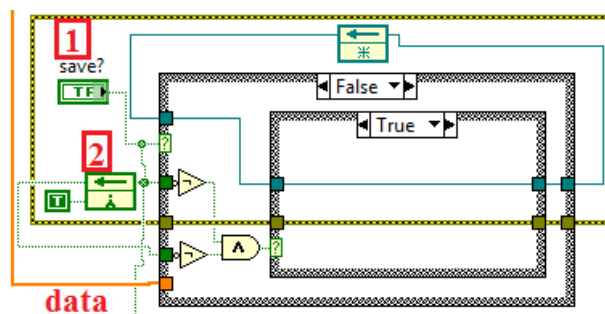
5.2 Měření a ukládání dat

Pro získání informací o dynamice systému nestačí jen pozorovat přechodové charakteristiky, ale je potřeba je ukládat k pozdějšímu vyhodnocení. Proto je nutné program rozšířit o tuto funkci. Veškeré prvky předchozího programu jsou nezměněny a je přidána část, která ukládá data. Celé rozšíření ukládání dat je vloženo do cyklu *While Loop* k předchozím prvkům, protože se ukládá vše, co je právě měřeno či ovládáno. Je potřeba ukládat obě žádané hodnoty $w(t)$ a výstupní přechodové charakteristiky $y(t)$. K těmto čtyřem veličinám je vhodné ještě ukládat vektor času přechodového děje, který poslouží jako časová osa při vyhodnocování průběhů. Požadavkem je tedy ukládat pět vektorů do nějakého souboru. V každé iteraci běhu cyklu je uložena aktuální hodnota všech veličin a vložena do vektoru, který má pět položek. Každým krokem se k uloženým pěti položkám přidají další, čímž vznikne dvourozměrné pole, které je ukládáno do souboru.

Proto přibude další programová struktura, která se nazývá „Case“. Tato struktura se dá přirovnat k větvení *If – Else* standardního programovacího jazyka. *Case* struktura je opět

rámeček, kde bloky umístěné v něm jsou součástí této struktury. *Case* strukturu lze řídit booleovskými *True/False* signály, kterými se přepíná obsah rámečku. Ve stavu *True* může obsahovat *Case* struktura úplně jiné bloky než v případě *False*. *Case* struktura se od *If – Else* začíná lišit v případě, že je zapotřebí přepínat více možností. To už se ale podobá programové struktuře *switch – case* standardního programovacího jazyka. Na *Case* lze přivádět například celá kladná čísla od 1 do 10 a každá hodnota strukturu přepne do jiného stavu s jinými bloky. *Case* struktura se vkládá z palety *Functions > Programming > Structures > Case Structure* v blokovém diagramu. Tato struktura je při ukládání dat využita pro spouštění a vypínání ukládání. Po spuštění programu nelze ihned začít ukládat data, protože se musí všechno nejdříve ustálit a poté měřit a ukládat. Systém Windows musí na začátku spustit program, který je náročný a mohlo by ze začátku dojít k chybě v reálném čase. Navíc celá elektronika pracuje se zpožděním, proto je vhodné před měřením pár sekund počkat. Ve chvíli zahřívacího běhu programu *Case* struktura pracuje s prázdným obsahem, po přivedení logické hodnoty ze spouštěcího tlačítka se přepne na bloky pracující se souborem a začnou se ukládat data.

Data se ukládají do formátu „.scl“. Soubor je nejprve nutné vytvořit, pak ukládat data a nakonec uzavřít. Problém je vyřešen dvěma vnořenými *Case* strukturami se zpožďovačem řídicího booleovského signálu tlačítka, čímž se vytvoří čtyři stupně řízeného vybírání funkčních bloků. Dále je využita ještě jedna *Case* struktura pro měření doby ukládání dat, která spustí zobrazování času ukládání také po stisknutí tlačítka. Na obrázku 5.7 je zobrazena část programu starající se o ukládání dat.



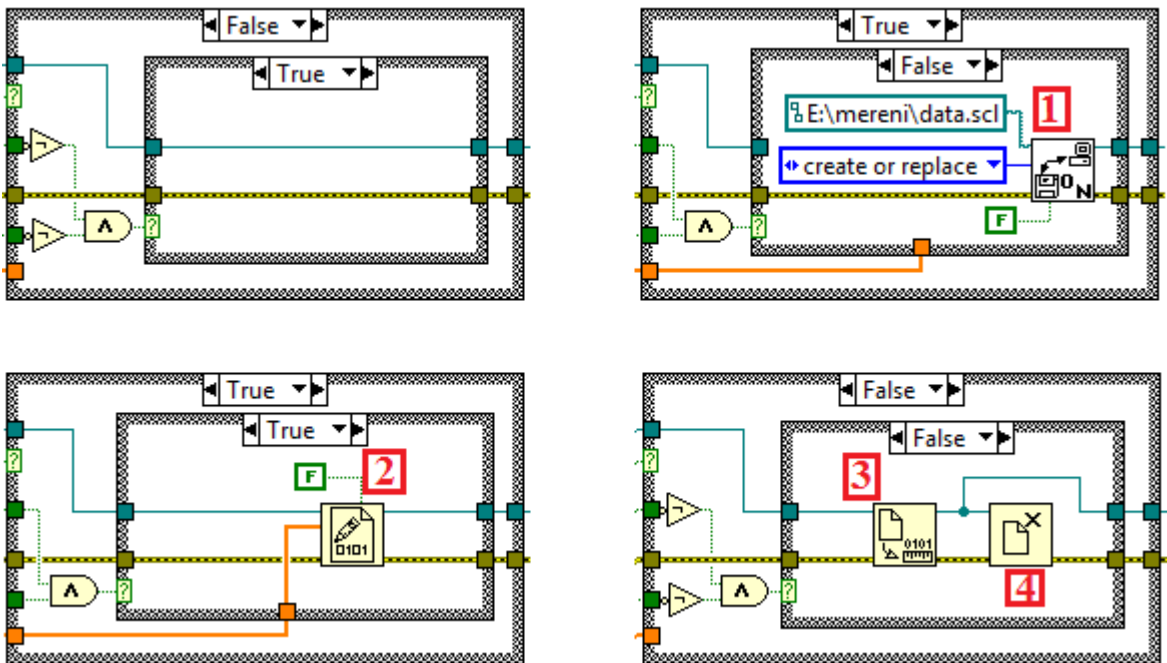
Obrázek 5.7 – Část programu, která ukládá data

Popis základních prvků ukládání dat:

1. Tlačítko, které spouští ukládání dat.
2. Feedback Node – speciální blok, který zpožďuje přivedený signál o 1 hodnotu. Zde má svou významnou funkci, která bude popsána dále.

Pokud je na spouštěcím tlačítku logická 0 (není spuštěné), vnější *Case* má *False*, vnitřní má *True*, kde není žádný blok. Po stisknutí tlačítka logická 1 přepne vnější *Case* na *True*. Díky zpožďovači logického signálu (*Feedback Node*) je na vnitřní *Case* po dobu jedné iterace přiváděna logická 0, čímž se přepne na *True* a bloky v ní vytvoří soubor. V další iteraci má vnitřní *Case* hodnotu *False* a začíná ukládat data. Po vypnutí tlačítka je opět na vnější *Case* přiváděna logická 0 a má hodnotu *False*. Zpožďovač přivádí po dobu jedné

iterace logickou 1 a v této jedné iteraci má vnitřní *Case* hodnotu *False*, při které uzavírá soubor. Poté se vnitřní *Case* přepne na hodnotu *True*, která neobsahuje žádné bloky. Během těchto kroků se uloží změřená data.

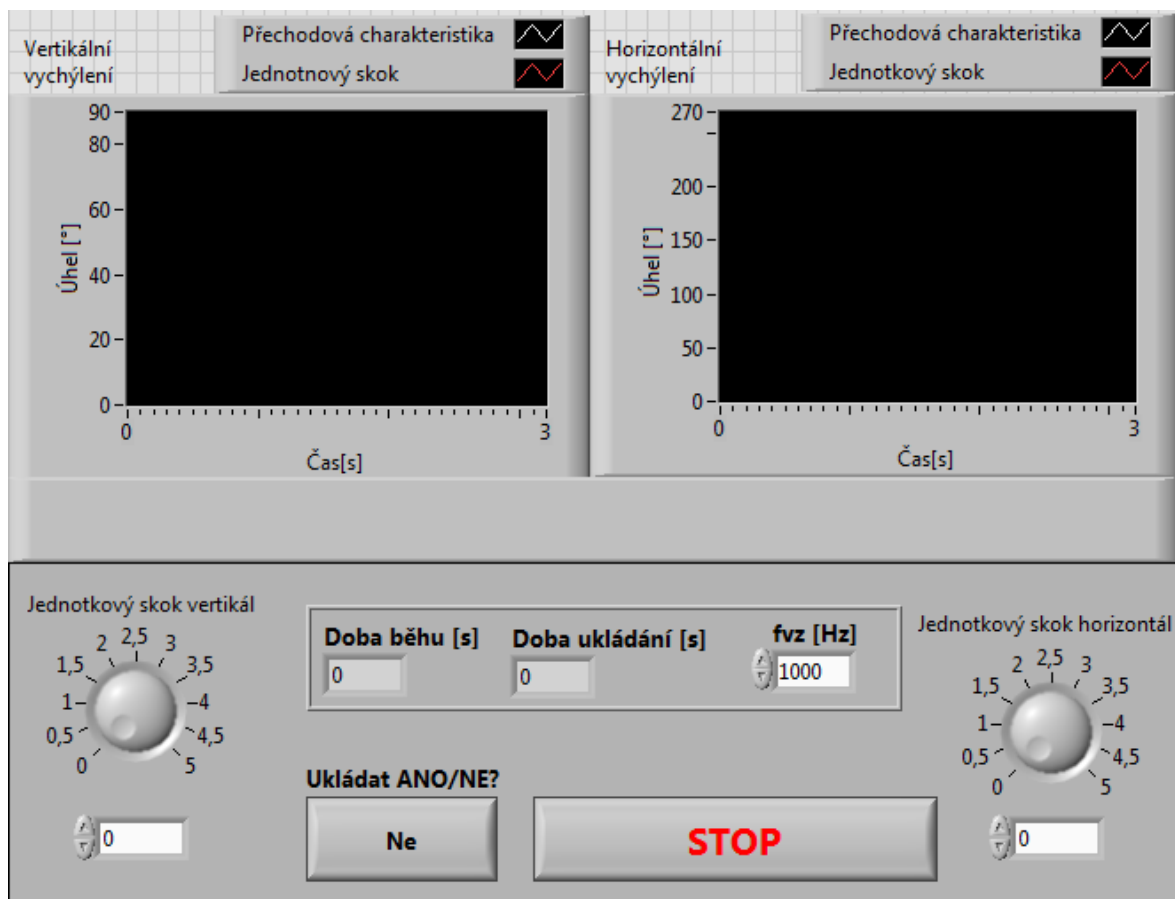


Obrázek 5.8 – Postupné vykonání operací při ukládání měřených dat

Popis důležitých bloků v *Case* strukturách ukládajících data:

1. Open_Create_Replace File – blok, který vytváří soubor ve složce, ke které má nadefinovanou cestu.
2. Write To Binary File – blok, který zapisuje data do předem otevřeného souboru.
3. Get Size File – blok, který zjišťuje velikost souboru a je předřazen bloku pro ukončení ukládání.
4. Close Binary File – uzavření souboru.

Všechny bloky jsou propojeny speciálním *Refnum* světle modrým signálem. Protože práci s ukládáním dat vykonává několik bloků, musí si vzájemně předávat informace. K tomu právě slouží v LabVIEW tento datový typ. Data jsou ukládána v binárním souboru a je potřeba je převést do takového formátu, se kterým pracuje MATLAB, aby bylo možno data použít k návrhu regulátorů.

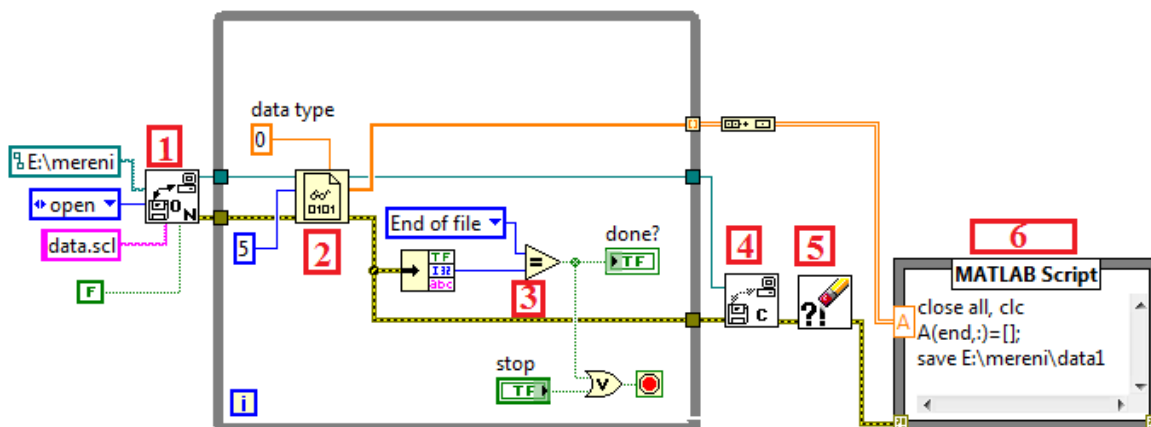


Obrázek 5.9 – Čelní panel rozšířeného programu s ukládáním dat

5.3 Převod binárního souboru do MATLAB souboru

Nejprve je důležité zmínit jednu důležitou vlastnost LabVIEW, která prozatím nebyla potřeba. LabVIEW umožňuje volat jádro MATLABu k jednoduchým operacím, jako je například vynášení grafů, proto je možné využívat vlastností a skriptů MATLABu přímo v programech blokových diagramů. Je zapotřebí mít nainstalovaný software MATLAB, protože při volání skriptu psaného v kódu MATLABu se zavolá jeho script server, ten přeloží a vyhodnotí skript a vrátí výsledky zpět do LabVIEW. K zavolání MATLABu slouží programová struktura *Matlab Script* v paletě *Functions > Mathematics > Script Nodes > MATLAB Script*.

Program na převod dat je navržen tak, že nejdříve otevře binární soubor a zavolá MATLAB pomocí *MATLAB Script*, do kterého v jednotlivých iteracích posílá data z otevřeného binárního souboru. Opět je využita programová struktura *While Loop*, protože na začátku převodu dat není známo, jak je soubor velký. *MATLAB Script* si vytváří (vícerozměrné) pole z přijímaných dat a v každé iteraci své pole zvětší o další prvek. Na konci se vytvořené pole dat uloží do nového souboru, který je definován cestou ve skriptu. Je zde také využit error signál, který lze použít jako indikátor konce souboru a jeho vyhodnocením proces převodu ukončit.



Obrázek 5.10 – Program na převod dat

Popis důležitých bloků v programu na převod dat:

1. Open_Create_Replace File – blok, který otevírá binární soubor na zadané adrese.
2. Read From Binary File – blok, která čte data z binárního souboru.
3. Porování informace, jestli nastal konec souboru.
4. Close File – uzavření souboru.
5. Clear Errors – vyčištění error signálu. Tuto operaci je vhodné provést stejně jako u měření, aby se odstranily případné budoucí chyby při opětovném spouštění programu.
6. Matlab Script – rámeček, ve kterém je psán skript v jazyce MATLAB. Zde skript přijímá data ze struktury *While Loop* a ukládá do vytvořeného pole. Struktura má přímo vstup určený na přijímání jednorozměrného i vícerozměrných polí dat. Po skončení čtení je pole uloženo do souboru zadaného názvu a adresy.

Program se po skončení převodu ukončí sám, v případě cyklické chyby jej uživatel ukončí tlačítkem stop. Na čelním panelu je indikátor ukončeného převodu signalizací „done“.



Obrázek 5.11 – Čelní panel programu na převod dat

6 Experimentální identifikace dvourotorového systému

Tato kapitola se zabývá experimentální identifikací a nalezením aproximačních přenosových funkcí obou rotorů. Při identifikaci systému je zohledňována interakce mezi rotory a nelinearita systému.

Před samotnou identifikací každého rotoru byly změřeny velikosti interakcí mezi rotory, se kterými je nutné při následném návrhu řízení počítat (určení velikosti poruch d_1 a d_2 , viz kapitola 3.2, obrázek 2.4). Při identifikaci byly využity předpoklady a informace z kapitoly 2, které byly potvrzeny a rozšířeny podrobnějším zkoumáním. Měřením byly předpoklady, kterými jsou nelinearita soustavy a vzájemné ovlivňování polohy rotorů, potvrzeny.

6.1 Vliv ocasního rotoru na úhel vychýlení hlavního rotoru

Hlavní rotor má rozsah vychýlení v ideálním případě -45° až 45° (pro zjednodušení 0° až 90°). Jelikož je ale úhel vychýlení hlavního rotoru ovlivňován ocasním rotorem, ne vždy je možné vychýlit hlavní rotor v uvedeném rozsahu. Pro identifikaci velikosti interakce bylo měření rozsahu úhlu vychýlení hlavního rotoru provedeno ve třech částech:

1. Ocasní rotor vypnutý.
2. Ocasní rotor na maximálních otáčkách po směru hodinových ručiček.
3. Ocasní rotor na maximálních otáčkách proti směru hodinových ručiček.

V každé části měření byl rozsah úhlu vychýlení hlavního rotoru omezen vlivem otáček ocasního rotoru. Odstředivá síla ocasního rotoru působí na hlavní rotor a nedovolí zůstat v ideálním rozsahu 0° až 90° . Při vypnutém ocasním rotoru se dokonce skutečné vychýlení také liší od ideálních 0° až 90° , vlivem závaží na modelu. Zkoumáním těchto třech částí interakce je možné určit velikost poruchy d_1 . V následující tabulce jsou uvedeny skutečné rozsahy vychýlení vlivem interakce (poruchy).

Tabulka 6.1 – Vliv ocasního rotoru na rozsah úhlu vychýlení hlavního rotoru

Interakce ocasního rotoru	Rozsah úhlu vychýlení, $^\circ$
Ideální rozsah	0 - 90
Ocasní rotor vypnutý	10 - 90
Ocasní rotor po směru maximum	0 - 72
Ocasní rotor proti směru maximum	20 - 90

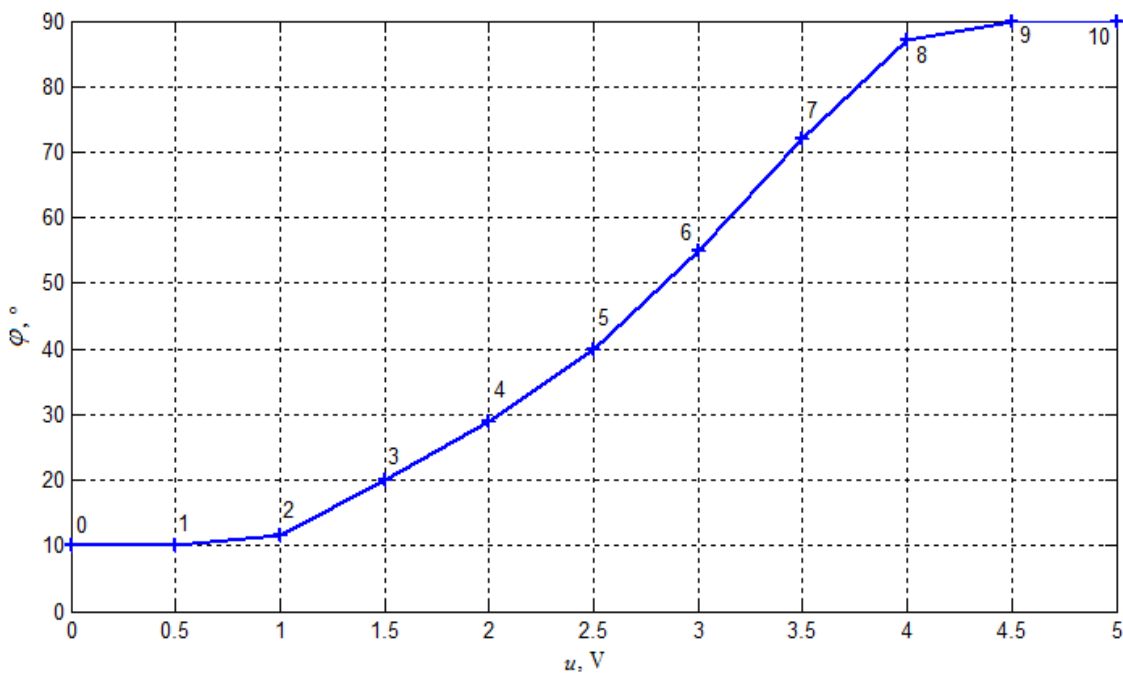
Z tabulky 2 lze nyní určit velikost maximální poruchy, která se vyskytuje při zapnutém ocasním rotoru po směru hodinových ručiček, která činí 18° . Při zapnutém rotoru proti směru hodinových ručiček je velikost poruchy 10° . Průměrná velikost poruchy z těchto dvou hodnot bude využita při návrhu regulátoru pro hlavní rotor a je maximální poruchou, která může ovlivňovat hlavní rotor. Její hodnota je $0,7$ V. Pro úspěšnou regulaci je potřeba otáčky ocasního rotoru omezit, aby se hlavní rotor vychýlil v co největším úhlu a nebyl tolik ovlivňován ocasním rotorem.

Dalo by se říci, že ocasní rotor svými otáčkami hlavnímu rotoru přidává nebo ubírá offset. Velkým kolísáním otáček ocasního rotoru se hlavní rotor rozkmitá a celý laboratorní model se

houpe. Proto je nutné v návrhu regulátoru pro ocasní rotor zajistit pomalejší regulační děj, který zajistí stabilitu a také se musí počítat s tím, že konstrukce modelu nedovolí řídit hlavní rotor v ideálním rozsahu a reálný pracovní rozsah úhlu musí být snížen.

6.2 Experimentální identifikace hlavního rotoru

Jelikož je předpokládán nelineární systém, prvním krokem je sestavení statické charakteristiky. Ideální pracovní rozsah pro vychýlení hlavního rotoru byl rozdělen na 10 bodů, po kroku 0,5 V tak, aby byl proměřen celý rozsah úhlu vychýlení hlavního rotoru a byl zajištěn dostatek dat pro tvorbu statické charakteristiky, která zobrazuje ustálené stavy. Pro tvorbu statické charakteristiky byla vybrána část s vypnutým ocasním rotorem, protože v této části se podařilo hlavní rotor vychýlit v největším rozsahu, tudíž pohled na statickou charakteristiku bude o systému vypovídat nejvíce.



Obrázek 6.1 - Statická charakteristika úhlu vychýlení hlavního rotoru

Z obrázku 6.1 je patrné, že se jedná o nelineární systém, protože s rostoucím napětím na vstupu úhel vychýlení neroste lineárně, není zde přímková závislost. Směrnice průběhu se v různých bodech mění, tím pádem zesílení soustavy se také mění. Tímto jsou teoretické předpoklady o systému potvrzeny a při návrhu regulátoru je nutné s touto skutečností počítat. Metody návrhu regulátorů fungují a jsou prováděny na lineárních modelech. Aby bylo možné řídit nelineární soustavu lineárním regulátorem, existuje několik přístupů, které je možno využít:

1. Linearizace statické charakteristiky – aplikací matematických postupů lze upravit stávající nelineární průběh charakteristiky a vytvořit přímkovou závislost, která bude vyhovovat celému rozsahu (soustava bude mít v celém pracovním rozsahu stejné zesílení) a teprve na tuto závislost navrhnout regulátor. Nevýhodou linearizace je

zavedení chyb, odchylek od skutečného průběhu statické charakteristiky, které jsou tím větší, čím je větší vzdálenost od pracovního bodu, ve kterém je provedena linearizace.

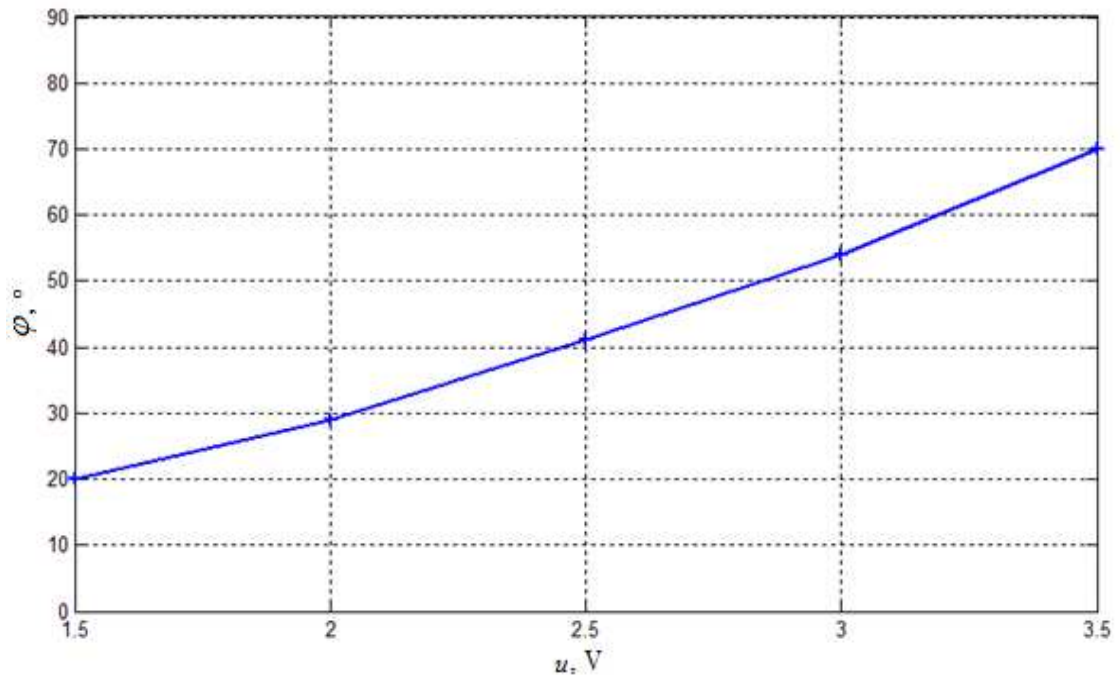
2. Gain Scheduling – pokud se statická charakteristika soustavy mění velmi značně a je požadavek na řízení v celém rozsahu, je možné ji rozdělit na malé dílčí pracovní rozsahy. Každý dílčí pracovní rozsah lze považovat za lineární, pokud je v okolí dílčího pracovního bodu lineární závislost. Pro každý dílčí pracovní rozsah je navržen vlastní regulátor, který musí splnit podmínku stability. Navržením více regulátorů pro několik dílčích pracovních bodů, které kopírují celý pracovní rozsah, lze regulovat celý pracovní rozsah přepínáním těchto regulátorů.
3. Omezení pracovního rozsahu na lineární část – je-li na nelineární statické charakteristice znatelná lineární část (nejčastěji motor, který se zpočátku rozbíhá pomaleji), lze navrhnout regulátor jen na ten pracovní rozsah, který je lineární. Regulace pak probíhá pouze v lineární části pracovního rozsahu.

Měřením bylo zajištěno 10 bodů pro zobrazení celého pracovního rozsahu pomocí statické charakteristiky. Z obrázku 6.1 lze vyčíst fakt, že několik částí statické charakteristiky, zhruba uprostřed pracovního rozsahu (mezi body 2 – 8), by se daly považovat za lineární. To znamená, že při zkoumání dynamiky systému pomocí přechodových charakteristik, jako reakcí na jednotkové skoky, by několik přechodových charakteristik mohlo být téměř shodných. Proto není nutné hledat aproximační přenosy všech přechodových charakteristik, ale stačí změřit reakci na jednotkový skok v jakékoliv části lineární oblasti a z přechodové charakteristiky vytvořit aproximační přenosovou funkci. Mezi body 0 – 1 a 9 – 10 systém vůbec nereaguje, jedná se tedy o experimentálně zjištěné minimum a maximum akční veličiny. Z toho plyne, že akční veličina při budoucím návrhu regulátoru musí být omezena v rozsahu 0,5 – 4,5 V, aby regulátor neměl snahu působit na soustavu neuskutečnitelně malými nebo velkými akčními zásahy. V bodech 1 – 2 a 8 – 9 je znatelná největší nelinearita. S přihlédnutím na fakt z kapitoly 6.1, která popisuje vliv ocasního motoru, je potřeba pracovní rozsah pro tvorbu přenosové funkce ještě snížit. Ve spodní části systém reaguje velmi zdlouhavě, v horní části zase může docházet k dorazům, jelikož se jedná o upoutanou helikoptéru. Proto bude vhodné omezit pracovní rozsah úhlu vychýlení hlavního rotoru na 20 – 80 ° stupňů.

6.2.1 Omezení pracovního rozsahu, analýza dynamiky systému

Pro potvrzení toho, že systém se chová téměř stejně ve vybrané lineární pracovní oblasti, je potřeba změřit v této oblasti několik přechodových charakteristik, které ilustrují podobnost v dynamice systému a také jeho linearitu. Zesílení soustavy by v lineární oblasti mělo být téměř stejné. Z obrázku 6.1 vyplývá, že 1,5 V akční veličiny odpovídá úhel 20°, což je počátek upraveného pracovního rozsahu. Pro vyhodnocení dynamiky v lineárním pracovním rozsahu byly změřeny 3 přechodové charakteristiky, jako reakce na jednotkový skok v rozsazích akční veličiny 1,5 – 2,5 V, 2 – 3 V, 2,5 – 3,5 V. Nejprve je na obrázku 6.2 zobrazena statická charakteristika ve vybraných hodnotách akční veličiny 1,5 V, 2 V, 2,5 V,

3 V a 3,5 V. Dynamika systému je tedy identifikována v lineárním pracovním rozsahu 20 – 70° úhlu vychýlení hlavního rotoru.

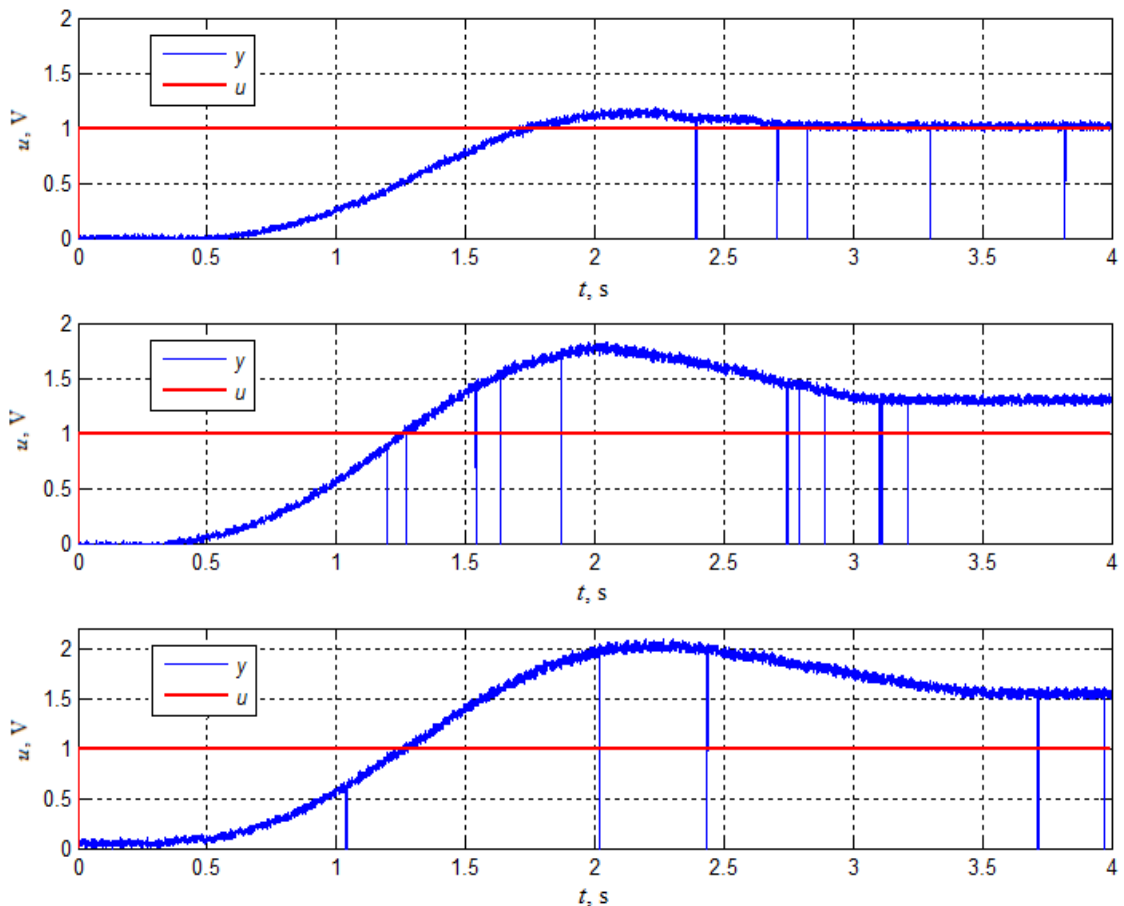


Obrázek 6.2 – Upravená statická charakteristika omezeného pracovního rozsahu

Podle obrázku 6.2 má zobrazený pracovní rozsah téměř lineární průběh. Při měření přechodových charakteristik se vyskytovaly problémy, protože rotoru občas kolísaly otáčky, takže ustalování na jednotlivých bodech se místy lišilo. To znamená, že při měření té samé přechodové charakteristiky docházelo k různým výsledkům a její průběh byl různý. Proto výsledný model může být méně přesný. Tyto problémy mohly být způsobeny řídicí jednotkou laboratorního systému nebo také řídicím počítačem, na kterém mohl operační systém řešit jiné úlohy a ovlivňovat měření.

Pro tvorbu aproximačního modelu je potřeba vybrat nejvhodnější přechodovou charakteristiku. Výhodou nyní je, že všechny 3 přechodové charakteristiky jsou dynamickou odezvou systému na jednotkový skok v „lineární“ oblasti. Proto aproximační přenos může být vytvořen na základě jedné ze 3 přechodových charakteristik. Pokud by model, vytvořený na základě jedné přechodové charakteristiky nevyhovoval, může být vytvořen další model z druhé nebo třetí přechodové charakteristiky. Předpokladem ale je, že všechny přechodové charakteristiky jsou téměř shodné. Nejvhodnější analyzovanou přechodovou charakteristikou by mohla být odezva na jednotkový skok 2 – 3 V, protože leží uprostřed a odchylky v oblasti počátku a konce pracovního rozsahu by mohly být nejmenší.

Na obrázku 6.3 jsou zobrazeny pod sebou všechny 3 přechodové charakteristiky posunuté do počátku souřadnicového systému, aby byla jasně viditelná odezva na jednotkový skok $u(t)$ s odezvou $y(t)$ a tak mohla být zkoumána shodnost průběhů pro budoucí model.



Obrázek 6.3 – 3 přechodové charakteristiky jako reakce na jednotkové skoky v „lineární“ oblasti, posunuté do počátku souřadnicového systému

Při pohledu na přechodové charakteristiky je vidět, že i když byla vybrána „lineární“ oblast, stále je i tato oblast lehce nelineární. První přechodová charakteristika je reakcí na skok 1,5 V – 2,5 V. Je viditelný pomalejší náběh a menší zesílení. Druhá přechodová charakteristika je reakcí na skok 2 V – 3 V, která leží uprostřed pracovního rozsahu a zde je viditelný rychlejší náběh a mírně větší zesílení soustavy. Třetí přechodová charakteristika je reakcí na skok 2,5 V – 3,5 V. Zde je náběh téměř shodný s druhou přechodovou charakteristikou, ale v této oblasti je znatelné ještě větší zesílení. Jako kompromis je vybrána druhá přechodová charakteristika, ležící uprostřed pracovního rozsahu a tím pádem možné odchylky by mohly růst podobně směrem k počátku i ke konci pracovního rozsahu. Jak již bylo zmíněno dříve, změřené průběhy jsou zatíženy chybou, kterou způsobují neidentifikované vlivy řídicí jednotky nebo řídicího počítače, proto při zkoumání dynamiky systému musí být počítáno s nepřesnostmi a výsledný model také nebude úplně odpovídat zkoumanému systému. Je zde snaha co nejlépe aproximovat zkoumanou soustavu.

6.2.2 Tvorba aproximačního modelu

Přechodová charakteristika má jev kmitavý. To znamená, že nelze použít jednodušších aproximačních metod, jako je například aproximace polynomem s násobnou časovou konstantou. Pro nalezení aproximačního přenosu ale velmi vhodně poslouží jedna z metod numerické optimalizace. V MATLABu je implementována takzvaná Simplexová optimalizační metoda, pomocí které lze najít odpovídající přenos, který aproximuje dynamické chování systému. Funkce, která volá optimalizační algoritmus má v MATLABu název *fminsearch()*. Tato optimalizační metoda hledá minimum funkce, na základě daného kritéria. Stručně řečeno, hledá se co nejlepší proložení průběhu přechodové charakteristiky racionální lomenou funkcí – přenosem, jehož jmenovatel je polynomem takového řádu, který nejlépe proloží skutečná data (je charakteristickým polynomem aproximované soustavy). Jako kritérium f se běžně využívá kvadrátu odchylek e reálných dat y od aproximace polynomem \hat{y} :

$$e = y - \hat{y} \quad (6.1)$$

$$f = e^2 \quad (6.2)$$

- kde e odchylka reálných dat od aproximace
 y reálná data dynamiky systému (naměřená přechodová charakteristika)
 \hat{y} data aproximace (aproximační přechodová charakteristika)
 f hodnota kritéria, která se předá funkci *fminsearch()*

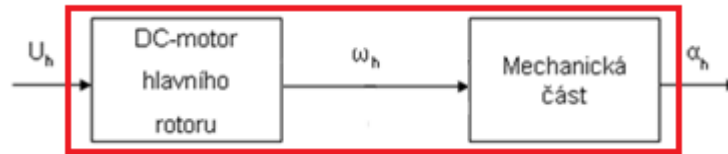
Celý proces probíhá krokově (iteračně). Kritérium obvykle obstarává nějaká uživatelská funkce s názvem například *minimum*, která je poté volána funkcí *fminsearch()* a je jedním z jejích parametrů *fminsearch(...,minimum,...)*. Uvnitř funkce *minimum*, obstarávající výpočet kritéria, probíhá v každém kroku srovnání reálných dat s aproximovanými a výpočet nové odchylky. Odchylka postupně klesá a až se dosáhne uspokojivého výsledku, proces skončí zobrazením přechodové charakteristiky reálných dat spolu s přechodovou charakteristikou aproximační.

Před začátkem optimalizace je nutné dodat skriptu následující informace:

- Vektory vstupů $u(t)$, výstupů $y(t)$ a vektor času t .
- Prvotní odhad parametrů zesílení K a parametrů charakteristického polynomu ($a_0, a_1, a_2, a_3 \dots$), podle řádu polynomu.
- Odhadovaný tvar aproximačního přenosu, tím také odhad řádu.

Jelikož se aproximovaná soustava skládá ze stejnosměrného motoru, který pohání hlavní rotor a to ještě přes relativně měkkou gumovou převodovku a celá tato soustava vychyluje rameno, je předpokládáno, že aproximace soustavy bude odpovídat minimálně polynomu řádu III, protože samotný stejnosměrný motor je soustava řádu II. Na obrázku 6.4 je zobrazeno „zapouzďení“ soustavy (dříve označené jako „rotor“) jejímž vstupem je řídicí

napětí a výstupem úhel vychýlení hlavního rotoru. Celá série mechanických spojitostí zvyšuje řád soustavy.

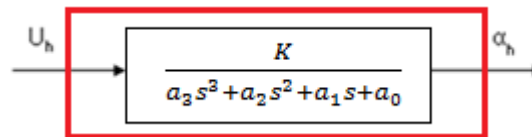


Obrázek 6.4 – Zapouzdření všech částí systému do jednoho bloku

Kmitavý průběh přechodové charakteristiky lze aproximovat přenosem, jehož jmenovatel je polynom s různými parametry, aby rozložení pólů mohlo vytvořit tlumený periodický děj. Přenos může vypadat takto:

$$G_s(s) = \frac{K}{a_3s^3 + a_2s^2 + a_1s + a_0} \quad (6.3)$$

Samozřejmě řád soustavy může být vyšší a v tom případě je potřeba polynom rozšířit. Na obrázku 6.5 je zobrazen aproximační přenos pro část systému na obrázku 6.4.



Obrázek 6.5 – Přenos aproximující soustavu

Uvedme nyní skript v MATLABu, který bude hledat aproximační přenos na základě reálných dat, odhadu parametrů a definovaném kritériu. Proměnné, ve kterých jsou uložena data, jsou definována jako globální, protože jsou využívány skriptem i uživatelskou funkcí *minimum*.

Hlavní skript:

```

global t y u
K=1; a3=1; a2=1; a1=1; a0=1;
x0=[K a3 a2 a1 a0];
x=fminsearch('minimum',x0,OPTIONS);
K=x(1); a3=x(2); a2=x(3); a1=x(4); a0=x(5);
sys=tf(K,[a3 a2 a1 a0]);
ys=lsim(sys,u,t);
plot(t,y);
hold on;
plot(t,ys);

```

%definice promennych
%vychozi situace, odhad parametru
%priprava vektoru odhadu
%optimalizace,fminsearch vraci
%vyber parametru
%model s nalezenymi parametry
%odezva na jednotkovy skok
%vykresleni realnych dat
%vykresleni aproximace

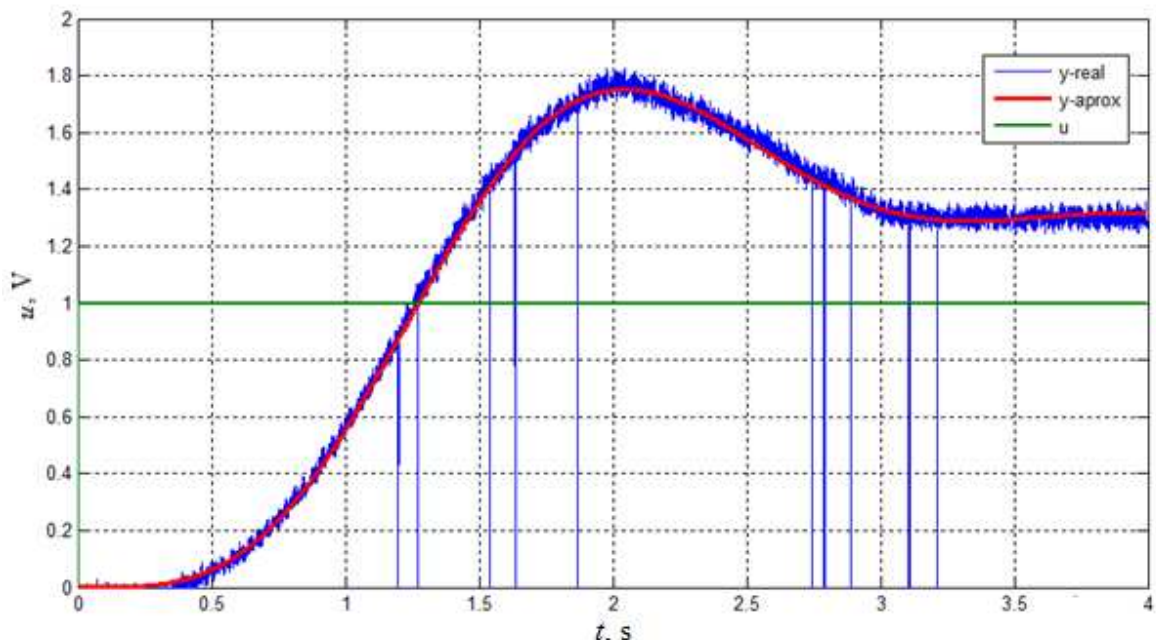
Uživatelská funkce *minimum*, která aplikuje kritérium:

```
function f=minimum(x)           %definice funkce
global y t u                    %definice promenných

K=x(1); a3=x(2); a2=x(3); a1=x(4); a0=x(5); %vyber parametru z vektoru
sys=tf(K,[a3 a2 a1 a0]);       %model uvnitr funkce
ys=lsim(sys,u,t);             %simulace uvnitr funkce
e=y-ys;                        %vypocet odchylky realita-aproximace
f=e'*e;                        %kvadrat odchylek, ucelova funkce pro fminsearch
```

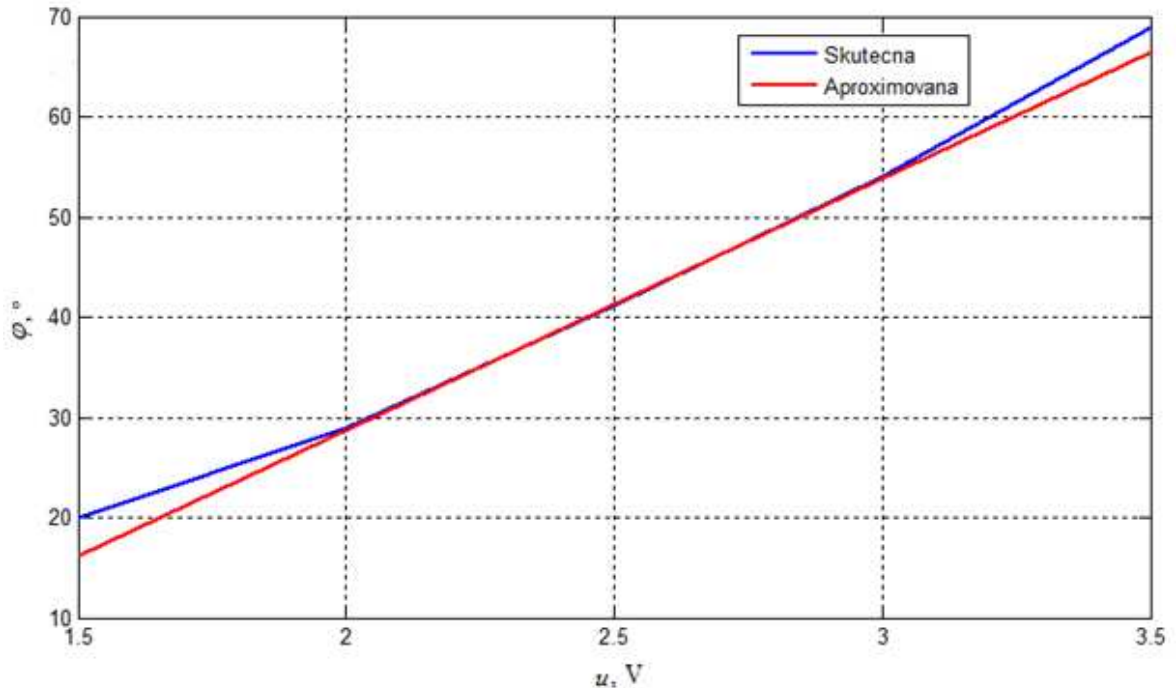
Skripty jsou částečně ilustrativní (obsahují komentáře), aby byl demonstrován princip výpočtu v MATLABu. Tím, že je předpokládán systém vyššího řádu, více záleží na tom, jak je nastaven odhad parametrů. Čím je řád vyšší, tím více různých výsledků může optimalizační metoda přinést a nemusí být uspokojivé. Při hledání co nejlepšího aproximačního přenosu bylo nutné provést několik desítek různých optimalizačních experimentů, než bylo dosaženo uspokojivého výsledku. Nejuspokojivějším proložením reálných dat je polynom IV. řádu. Použití polynomu III řádu nepřineslo tak dobrý výsledek. Na obrázku 6.6 je zobrazen průběh reálných dat a aproximovaný průběh. Výsledný model v podobě přenosu hlavního rotoru v „lineární“ oblasti je následující:

$$G_s(s) = \frac{1,2978}{0,0402s^4 + 0,1066s^3 + 0,5522s^2 + 0,724s^1 + 1} \quad (6.4)$$



Obrázek 6.6 – Zobrazení reakce na jednotkový skok reálného systému a aproximačního modelu

Aby bylo možné teoreticky rozhodnout o správnosti nalezeného modelu, je vhodné sestavit statickou charakteristiku s použitím modelu a zobrazit ji vedle statické charakteristiky reálného měření. Čím více se obě statické charakteristiky shodují (hlavně svým sklonem), tím vhodnější je nalezený model pro vybraný omezený pracovní rozsah. Na obrázku 6.7 je zobrazeno porovnání obou statických charakteristik.



Obrázek 6.7 – Porovnání statických charakteristik

Podle obrázku 6.7 je nalezený model pro daný lineární pracovní rozsah vhodný. Sklon obou křivek je podobný a je dobře viditelný fakt, že s nárůstem akční veličiny zesílení soustavy stoupá a zároveň s poklesem akční veličiny zesílení soustavy klesá. Návrh regulátoru bude proveden jen pro tento omezený pracovní rozsah. Výchozí pozice úhlu natočení před začátkem regulačních pochodů pak může být buď 20° a regulace může být prováděna směrem nahoru a pak až dolů, nebo se za výchozí pozici určí střed pracovního rozsahu a regulace může být prováděna oběma směry.

6.3 Vliv hlavního rotoru na úhel vychýlení ocasního rotoru

Zatímco hlavní rotor odpovídá statické soustavě a vlivy ocasního rotoru jdou změřit díky skutečnosti, že statická soustava se na vnější podnět ustálí, ocasní rotor je soustavou astatickou. Změnou otáček hlavního rotoru, který se pohybuje ve vertikálním směru, se ocasní rotor vychyluje ze své stabilní polohy (pokud nějakou má) a pohybuje se až k dorazu. Díky této skutečnosti lze obtížně změřit velikost poruchy. Navíc velikost poruchy se při různém vychýlení hlavního rotoru nemění stejně. Při malých otáčkách hlavního rotoru je poruchový vliv roven přibližně 0,5 V. Při zvyšování otáček nejprve poruchový vliv roste až na hodnotu 0,7 V a pak začne klesat pod 0,5 V. Toto je způsobeno soustavou napájecích kabelů, které

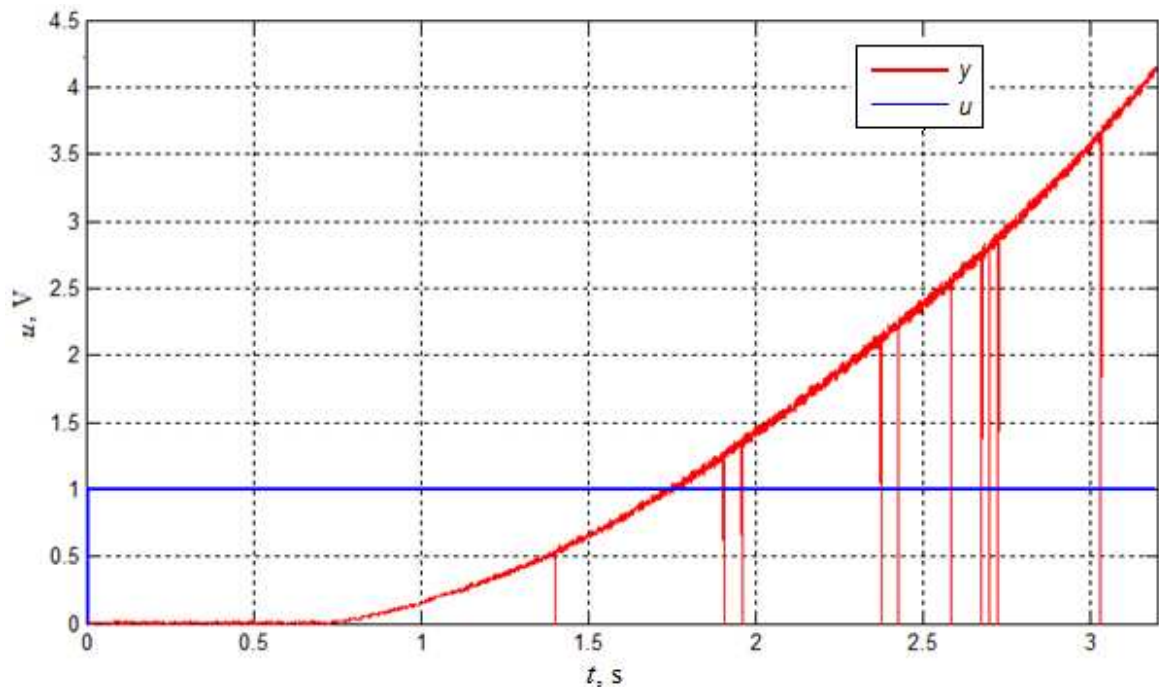
svým ohýbáním působí jako malá pružinka, která působí proti odstředivé síle hlavního rotoru a způsobuje neměřitelnou poruchu.

Pro řízení ocasního rotoru ale lze vliv hlavního rotoru využít, protože úhel vychýlení ocasního rotoru se při vypnutém hlavním rotoru teoreticky nemůže ustálit. Přivedením skokové akční veličiny se ocasní rotor vždy vychýlí z ustáleného stavu a dojde k naražení do dorazu. Vlivem konstrukce celého laboratorního modelu se úhel vychýlení ocasního rotoru dotýká dorazu i při vypnutém systému (rozložení hmotností). Přivedením akční veličiny se úhel natočení začne zvětšovat, až rameno dosáhne protějšího dorazu (viz kapitola 2.4). Pokud by úhel vychýlení ocasního ramene překročil požadovanou hodnotu, pohon ocasního rotoru by musel změnit směr otáčení a měl by snahu dostat ocasní rotor zpět na požadovanou hodnotu. Vlivem astatické setrvačnosti a faktu, že motor se musí zastavit, změnit směr otáčení a znovu se rozjet (viz kapitola 2.3), lze těžko dosáhnout požadovaného úhlu vychýlení. V určité situaci ale nastane případ, že odstředivá síla hlavního rotoru, působící proti pohonu ocasního rotoru, se vyrovná se silou pohonu ocasního rotoru a ocasní rotor se může ustálit na určené hodnotě.

Řídit samotný úhel natočení ramene ocasního rotoru je velmi obtížné a bez interakce hlavního rotoru to ani není možné. Spíše je vhodné určit výchozí úhel vychýlení horizontálního směru a držet jeho polohu vzhledem k měnícímu se úhlu hlavního rotoru (u reálné helikoptéry slouží ocasní rotor k udržení úhlu natočení celé helikoptéry, aby se nezačala otáčet kolem své osy). Regulátor ocasního rotoru tedy spíše reguluje poruchu. Díky tomu, že pro řízení hlavního rotoru byl vybrán daný rozsah, bude se vždy vycházet ze skutečnosti, že hlavní rotor v celém regulačním procesu bude aktivní. Vzhledem k náchylnosti ocasního rotoru je dobré snažit se použít regulátor s pomalým regulačním dějem i za cenu ztráty kvality regulačního pochodu, pokud bude zachována stabilita.

6.4 Experimentální identifikace ocasního rotoru

Ocasní rotor je nestabilní systém. Z toho plyne, že neexistuje statická charakteristika, která by ilustrovala nelinearity a tvar pracovního rozsahu. Jedinou možností, jak získat informace o chování takovéto soustavy, je přechodová charakteristika jako reakce na jednotkový skok. Přechodová charakteristika lze měřit při pohybu ocasního ramene od dorazu 0° k dorazu 270° nebo naopak a také při různém vychýlení hlavního rotoru, který zpomaluje nebo zrychluje přechodový děj ($0^\circ - 270^\circ$ odstředivá síla hlavního rotoru působí proti pohonu ocasního rotoru a zpomaluje přechodový děj, $270^\circ - 0^\circ$ odstředivá síla hlavního rotoru se sčítá s pohonem ocasního rotoru). Přechodová charakteristika pro analýzu dynamiky byla změřena v rozsahu $0^\circ - 270^\circ$, při vychýlení úhlu hlavního rotoru 45° (pohon hlavního rotoru drží konstantní otáčky a polohu), což je uprostřed pracovního rozsahu hlavního rotoru, kde byl hlavní rotor identifikován a v tomto pracovním rozsahu bude hlavní rotor působit na ocasní rotor. Přechodová charakteristika ocasního rotoru, jakožto reakce na jednotkový skok, je zobrazena na obrázku 6.8.



Obrázek 6.8 – Přechodová charakteristika ocasního rotoru

Měření probíhalo v rozsahu úhlu $0^\circ - 270^\circ$, což odpovídá napětí z čidla $0\text{ V} - 5\text{ V}$ (viz kapitola 2.4). V tomto rozsahu úhlu je akční veličinou pro pohon ocasního rotoru $2,5\text{ V} - 5\text{ V}$ (viz kapitola 2.3). Na hodnotě $2,5\text{ V}$ řídicího napětí pohon stojí, motor se začíná rozbíhat od $2,5\text{ V}$. Ve skutečnosti, při zapnutí hlavním rotoru v jeho pracovním rozsahu, se začíná ocasní rotor v horizontálním směru vychylovat až na hodnotě $3,1\text{ V}$ řídicího napětí. Skutečný skok pro analýzu dynamiky soustavy je proveden skokem $3,1\text{ V} - 4,1\text{ V}$, který odpovídá 1 V jednotkového akčního skoku.

6.4.1 Tvorba aproximačního modelu

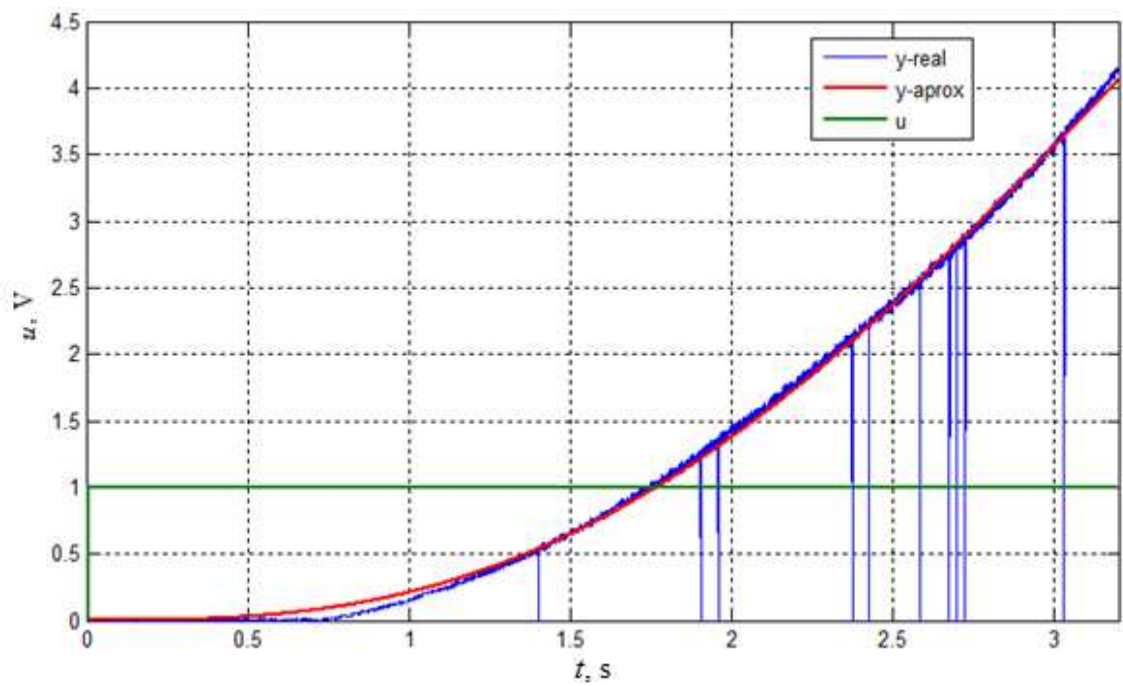
Pohon ocasního rotoru není připojen k rotoru přes žádnou gumovou převodovku, jako je to u hlavního rotoru, ale rotor je přímo na hřídeli motoru. Jelikož řád aproximačního modelu hlavního rotoru vyšel jako nejlepší řešení proložení polynomem IV. řádu, je předpokládáno, že soustava je nyní III. řádu (řád snížen právě o převodovku), o čemž dále přesvědčí nalezený model. Hledání modelu bylo provedeno s využitím aproximační přenosové funkce s atatismem I. řádu, opět s pomocí simplexové optimalizační metody (viz kapitola 7.2.2). Předpokládaný tvar přenosu ocasního rotoru je následující:

$$G_s(s) = \frac{K}{a_3s^3 + a_2s^2 + a_1s} \quad (6.5)$$

Nalezený aproximační přenos pomocí simplexové optimalizační metody:

$$G_s(s) = \frac{1,606}{1,127s^3 + 0,3803s^2 + 0,9681s} \quad (6.6)$$

Na obrázku 6.9 je viditelné úspěšné proložení reálné přechodové charakteristiky nalezeným přenosem.



Obrázek 6.9 - Zobrazení reakce na jednotkový skok reálného systému a aproximačního modelu

7 Návrh regulátorů

Při hledání parametrů regulátorů je k metodě Ziegler-Nichols přidána další metoda, aby bylo možné pomocí simulací porovnat regulační děje a vybrat nejlepší sadu parametrů regulátorů pro regulaci laboratorního modelu. Prvotní nalezení parametrů je provedeno Ziegler-Nicholsovou metodou a dalšími postupy jsou parametry dále upravovány. Pro vyhodnocení nejlepšího regulačního děje byla vybrána dvě kritéria, na základě kterých je vyhodnocována kvalita regulačních pochodů:

- Doba regulačního pochodu t_r – doba, za kterou regulovaná veličina $y(t)$ dosáhne 95% žádané hodnoty $w(t)$ v ustáleném stavu.
- Relativní překmit k – překmit $y(t)$ v %, vzhledem k $w(t)$.

7.1 Omezení skutečných regulačních pochodů rotorů

Hlavnímu rotoru byl pracovní rozsah omezen na lineární část, v rozmezí $20^\circ - 70^\circ$ (kapitola 6.2.1, obrázek 6.2). Žádaná hodnota $w(t)$ může být nastavena jen v tomto rozsahu. Regulace tedy probíhá s tím, že hlavní rotor je před začátkem regulace již roztočen přivedeným napětím 1,5 V odpovídajícím úhlu 20° . Porucha $d_1(t)$ je přivedena na výstup soustavy a její hodnota může činit až 0,7 V (průměr změřených poruch). To ale jen v případě, že by se otáčky ocasního rotoru při regulaci hlavního rotoru měnily velmi výrazně a skokově. Při pomalých regulačních dějích je porucha znatelně menší. Porovnání simulovaných průběhů se skutečnými je vždy v rozsahu $20^\circ - 70^\circ$ u laboratorního modelu a s upraveným rozsahem $0^\circ - 50^\circ$ v simulacích. Upravený rozsah je popsán v kapitole 7.2.3.

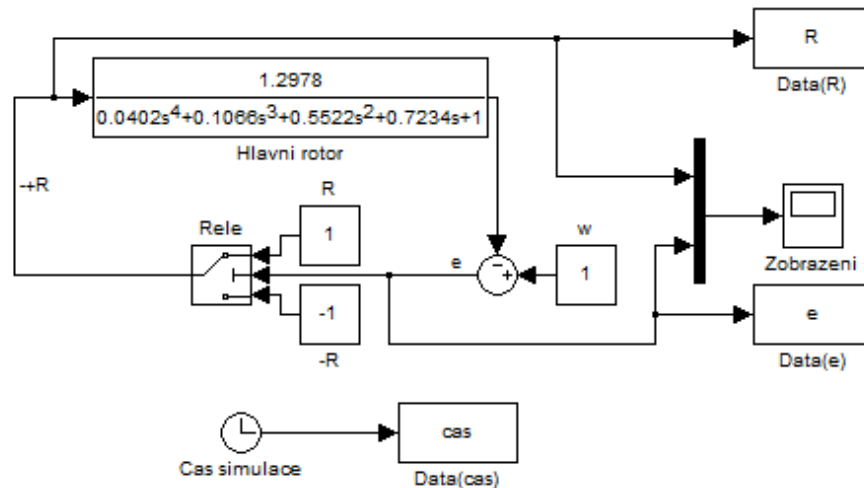
Pohon ocasního rotoru má jednu velmi nepříjemnou vlastnost a tou je již několikrát zmiňovaný fakt, že reakce na řídicí signál, který začíná od 2,5 V, je nulová. Reakce začíná zhruba od 3 V, protože mezi 2,5 V a 3 V je takzvaná mrtvá zóna. Mrtvá zóna je i při opačném smyslu otáčení pohonu, takže celková mrtvá zóna je při řídicím napětí 2 V – 3 V. Tím pádem nelze použít regulátor P, který by v případě malých žádaných hodnot vůbec nemohl zareagovat. Je nutné použít PI regulátor, jehož I složka postupně zvyšuje akční veličinu, až se akční veličina dostane z mrtvé zóny. V simulacích je mrtvá zóna simulována pomocí speciálního bloku. Mrtvá zóna je i u hlavního rotoru, ale tam je odstraněna výběrem regulace v omezeném rozsahu.

7.2 Regulátory navržené metodou Ziegler-Nichols

Nejprve je aplikována metoda Ziegler-Nichols (viz kapitola 3.1), což je výpočet parametrů regulátoru z kritických hodnot r_{0k} a T_k , získaných zapojením soustavy s relé ve zpětné vazbě. Takto získané parametry mohou, ale také nemusí na reálné soustavě fungovat tak dobře, protože Ziegler-Nicholsova metoda je poměrně agresivním nastavením parametrů a prvotně byla navržena pro chemický průmysl a to hlavně pro regulaci poruch. Pokud je regulace s takto nalezenými parametry neuspokojivá, mohou se tyto využít jako prvotní s následujícím ručním hledáním uspokojivějších parametrů.

7.2.1 Regulátor ZN pro hlavní rotor

Soustava aproximující hlavní rotor je IV. řádu, proto lze relé ve zpětné vazbě bez problému využít. Soustava byla rozkmitána přivedením konstanty zastupující jednotkový skok s hodnotou $w = 1$, viz obrázek 7.1. Relé reaguje na přivedenou odchylku $e(t)$ a jeho výstupem je konstanta $\pm R$, která je přivedena na vstup soustavy. V čase $t = 0$ se $e = w$ a relé se přepíná na hodnotu $-R$. Na hodnotě $e = 0$ se relé přepíná na hodnotu $+R$. Celý proces je periodický.



Obrázek 7.1 – Relé ve zpětné vazbě, hlavní rotor

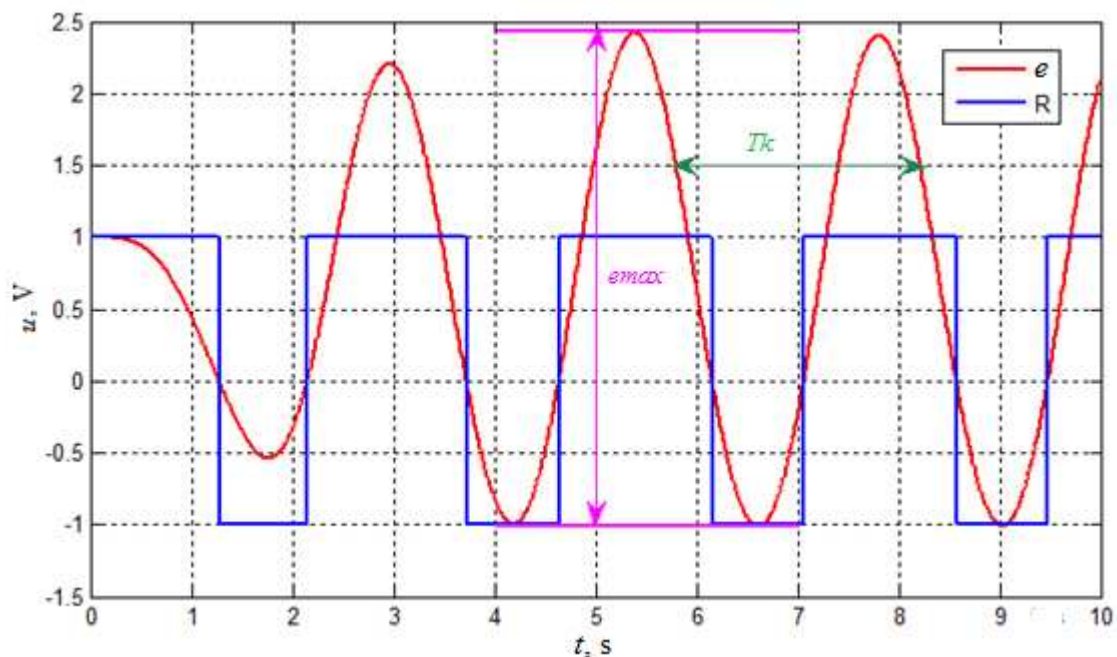
Z průběhu vstupu do relé e a výstupu z relé $\pm R$ je změřena kritická perioda T_k a kritická amplituda e_{max} . Kritické kmity jsou zobrazeny na obrázku 7.2. Změřené kritické hodnoty jsou následující:

$$\pm R = 1 \quad (7.1)$$

$$e_{max} = 3,4 \text{ V} \quad (7.2)$$

$$T_k = 2,55 \text{ s} \quad (7.3)$$

$$r_{0k} = \frac{4R}{\pi e_{max}} = 0,37 \text{ V} \quad (7.4)$$



Obrázek 7.2 – Kritické kmity s relé ve zpětné vazbě, hlavní rotor

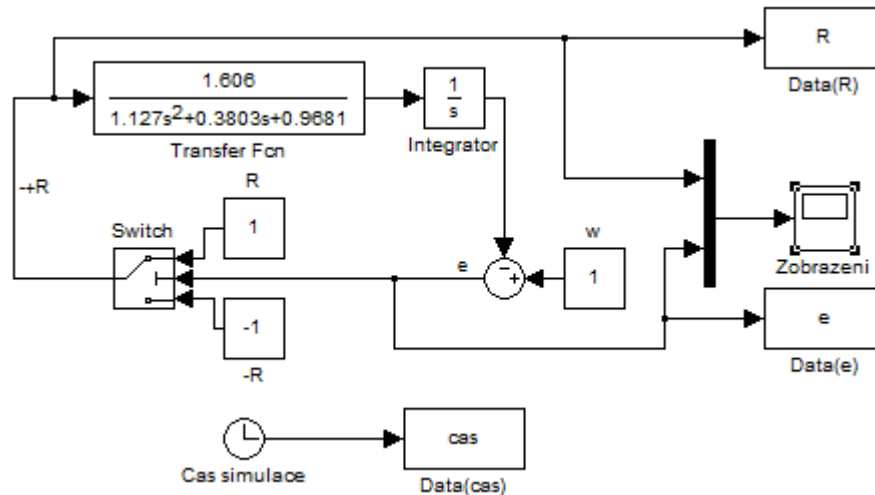
Pomocí kritických hodnot jsou vypočítány parametry regulátorů P, PI a PID. V tuto chvíli totiž není jasné, který z regulátorů bude nejvíce vyhovovat regulované soustavě, proto jsou vypočítány všechny. Výpočet parametrů vychází z tabulky 1, kapitoly 3.1. Jistá je jen nevhodnost regulátoru P, jelikož má trvalou regulační odchylku a je nevhodný k regulaci statické soustavy.

Tabulka 7.1 – Nalezené parametry regulátorů pro hlavní rotor

Regulátor	r_0	T_i , s	T_d , s
P	0,185		
PI	0,17	2,17	
PID	0,23	1,275	0,32

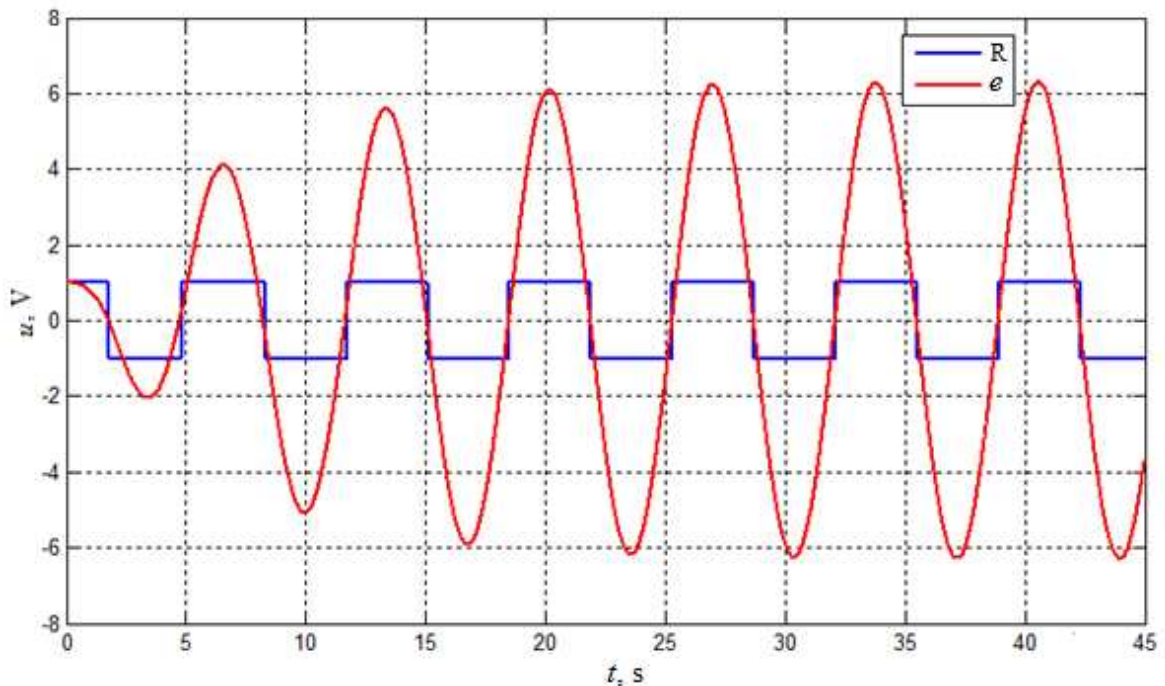
7.2.2 Regulátor ZN pro ocasní rotor

Ocasní rotor je také aproximován přenosem vyššího řádu, proto lze i v tomto případě využít zapojení relé ve zpětné vazbě s následným přepočtem parametrů Ziegler-Nichols metodou. Byl využit naprosto stejný postup, jako v kapitole 7.2.1. Rozdíl je jen v zapojení, kdy je na výstup přenosové funkce připojen blok integrace, který celému přenosu přidá řád, což odpovídá astatické soustavě.



Obrázek 7.3 – Relé ve zpětné vazbě, ocasní rotor

Z průběhu vstupu do relé e a výstupu z relé $\pm R$ je změřena kritická perioda T_k a kritická amplituda e_{max} . Kritické kmity jsou zobrazeny na obrázku 7.4.



Obrázek 7.4 – Kritické kmity s relé ve zpětné vazbě, ocasní rotor

Změřené kritické hodnoty jsou následující:

$$\pm R = 1 \quad (7.5)$$

$$e_{max} = 12,5 \text{ V} \quad (7.6)$$

$$T_k = 7 \text{ s} \quad (7.7)$$

$$r_{0k} = \frac{4R}{\pi e_{max}} = 0,101 \text{ V} \quad (7.8)$$

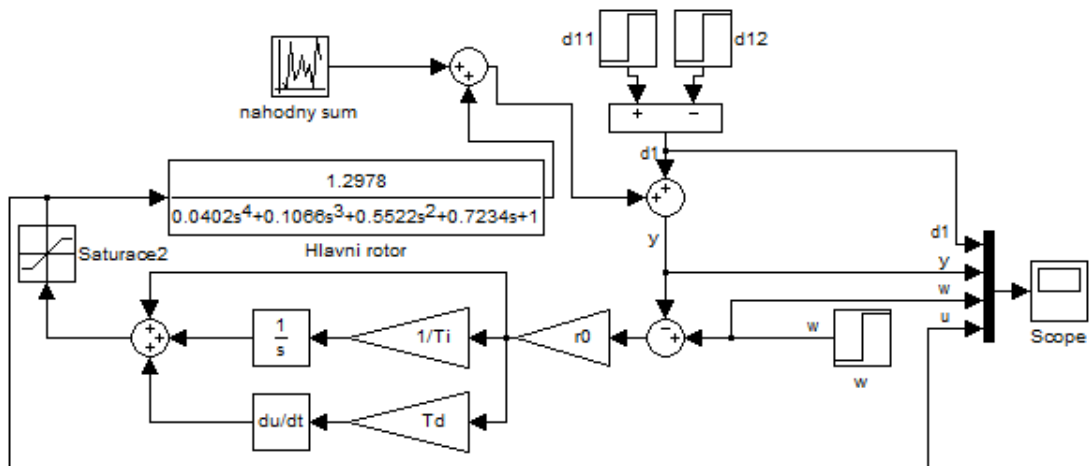
Pomocí kritických hodnot jsou vypočítány parametry regulátorů P, PI a PID. V tuto chvíli také není jasné, který z regulátorů bude nejvíce vyhovovat regulované soustavě, proto jsou vypočítány všechny. Výpočet parametrů vychází z tabulky 2.1. Na regulaci astatické soustavy by bylo možné použít regulátor P, ale z důvodu mrtvé zóny, viz kapitola 7.1, na soustavu ocasního rotoru použít nelze.

Tabulka 7.2 – Nalezené parametry regulátorů pro ocasní rotor

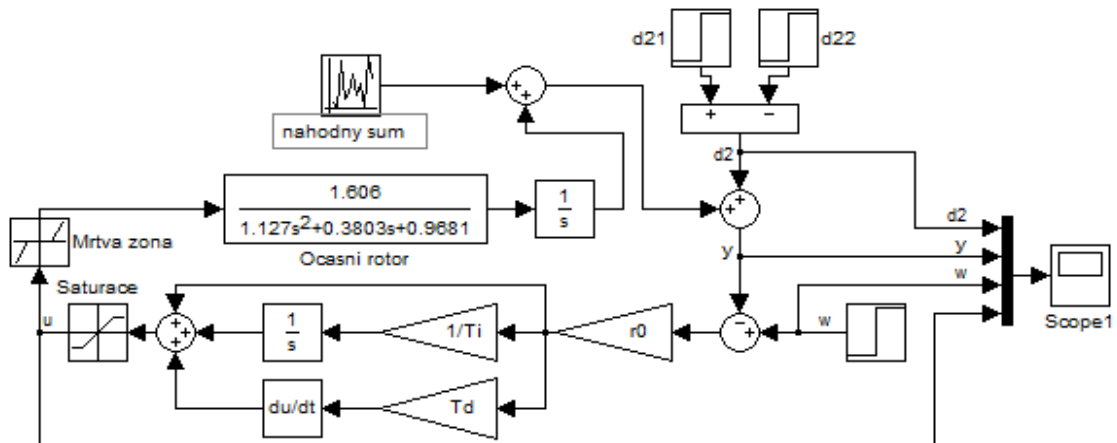
Regulátor	r_0	T_b , s	T_d , s
P	0,05		
PI	0,045	0,95	
PID	0,06	3,5	0,875

7.2.3 Simulace regulačních pochodů ZN

V simulacích regulačních pochodů obou rotorů jsou testovány regulátory PI a PID ve standardním zapojení (obrázek 2.2). Nejprve je simulována reakce na žádanou hodnotu a po ustálení, reakce na dvě po sobě jdoucí poruchy, z důvodu simulace interakce. V simulaci je potřeba dodržet reálné rozsahy akčních veličin, zapojením saturací do větví akčních veličin. Simulace hlavního rotoru vychází z omezeného rozsahu $20^\circ - 70^\circ$, který lze při simulaci brát jako rozsah $0^\circ - 50^\circ$. Omezení akční veličiny $u(t)$ v simulaci je proto místo původních $0 \text{ V} - 4,5 \text{ V}$ posunutá o $1,5 \text{ V}$ (20°), což odpovídá $-1,5 \text{ V}$ až $3,5 \text{ V}$. Simulace ocasního rotoru je simulována v rozsahu $-2,5 \text{ V}$ až $2,5 \text{ V}$, jde jen posun o $2,5 \text{ V}$, opět pro zjednodušení simulace. Na obrázku 7.5 je zobrazen URO hlavního rotoru, na obrázku 7.6 je zobrazen URO ocasního rotoru. Zapojení regulátoru PI je téměř stejné jako PID (absence větve s derivací), proto není uvedeno.



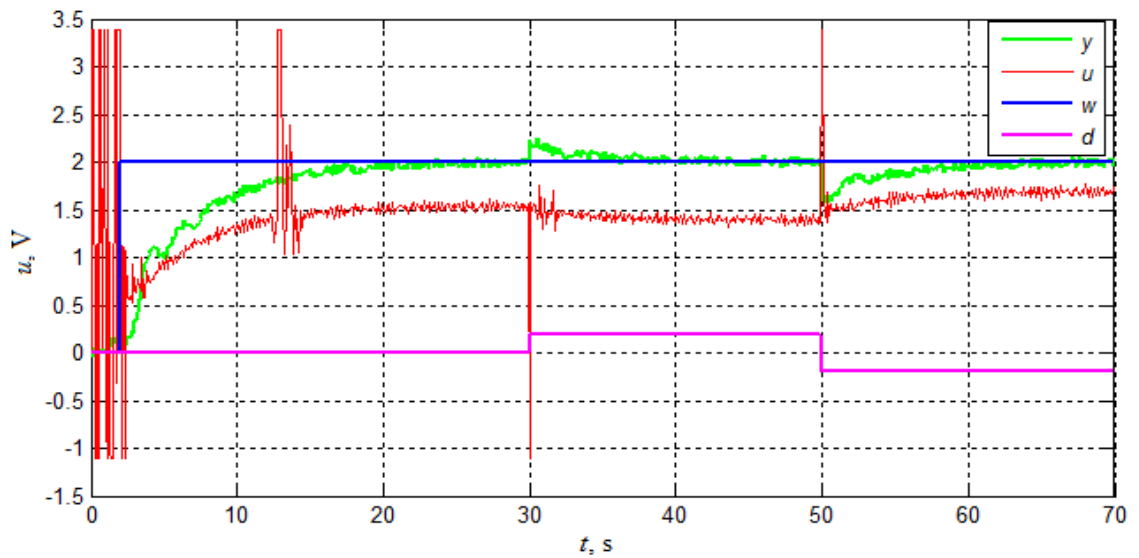
Obrázek 7.5 – Simulace URO hlavního rotoru



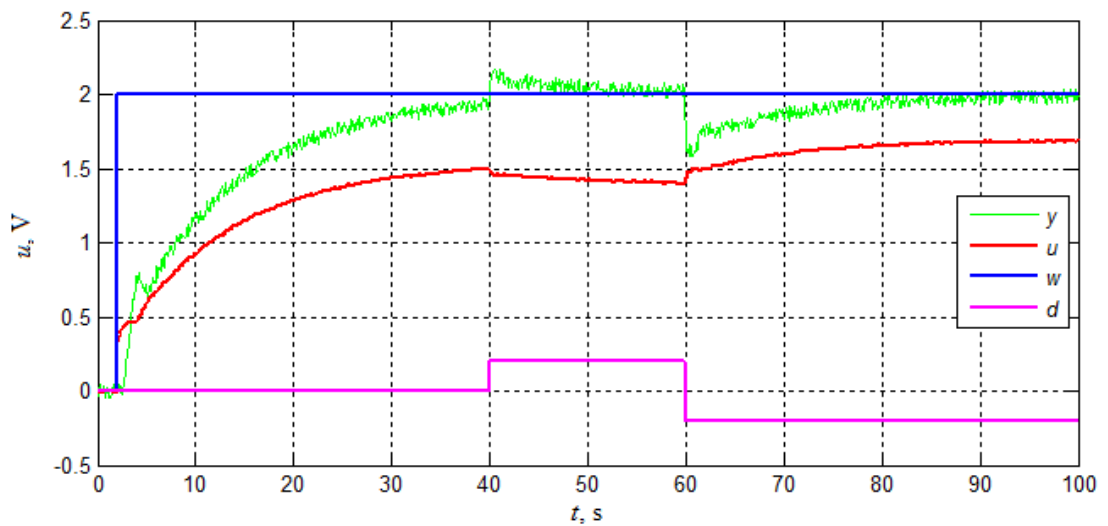
Obrázek 7.6 – Simulace URO ocasního rotoru

Aby bylo dosaženo co nejlepšího přiblížení k reálným podmínkám, k výstupu soustavy byl přidán náhodný šum, jako simulace zašuměného signálu z čidla měřícího úhel. Na obrázku 6.6 je viditelná úroveň šumu, jehož hodnota je 0,1 V, což je z hlediska celkového rozsahu 5 V relativně hodně. V případě takto zašuměného signálu by mohla derivační složka tvořit problémy generováním rychle se střídajících velkých akčních zásahů. Cílem je především plynulá a stabilní regulace, kvalita regulačního pochodu je až na druhém místě.

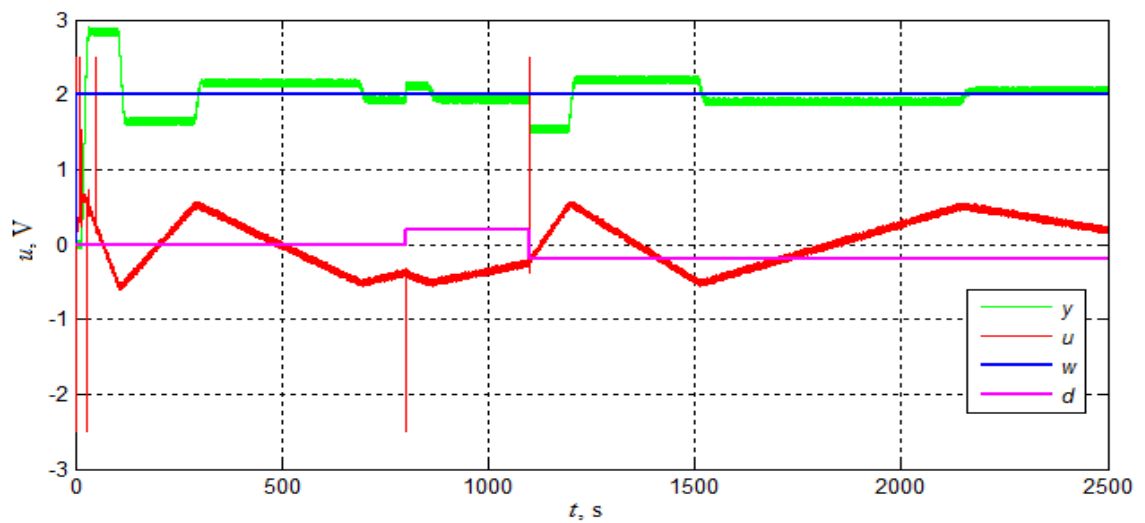
Srovnáním obrázků 7.7 a 7.8 lze usoudit, že pro regulaci hlavního rotoru je mnohem lepší využít regulátor PI. Doba regulačního pochodu je sice u PID sice kratší, ale regulátor PI netvoří tak nevyzpytatelné akční zásahy a jeho integrační charakter dokonce působí jako filtrace šumu superponovaného na měřené veličině. Stále se ale jedná o simulaci a až laboratorní model ukáže vhodnost těchto regulačních pochodů.



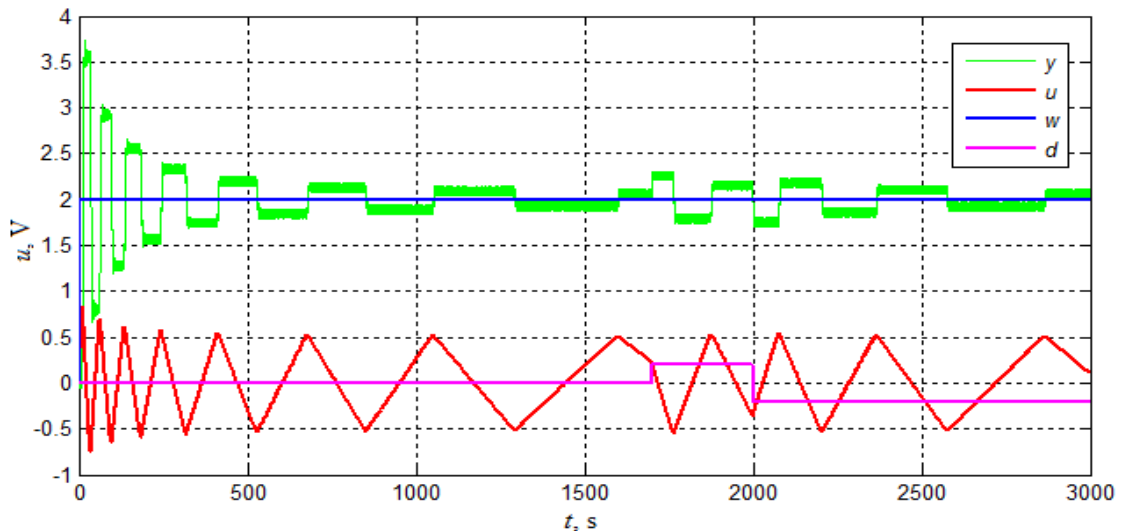
Obrázek 7.7 – Simulace regulačního pochodu hlavního rotoru, regulátor PID, ZN



Obrázek 7.8 – Simulace regulačního pochodu hlavního rotoru, regulátor PI, ZN



Obrázek 7.9 – Simulace regulačního pochodu ocasního rotoru, regulátor PID, ZN



Obrázek 7.10 – Simulace regulačního pochodu ocasního rotoru, regulátor PI, ZN

Regulační pochody ocasního rotoru (obrázek 7.9 a 7.10) s parametry nalezenými pomocí Ziegler-Nichols metody jsou nevhodné. Vlivem mrtvé zóny a velké setrvačnosti ocasního rotoru je doba regulačních pochodů obrovská. Regulační pochody s těmito parametry jsou stabilní, ale nepoužitelné. Nevhodný je regulátor PID i regulátor PI. K nalezení lepších parametrů je potřeba využít jinou metodu, která je uvedena v následující kapitole.

7.3 Regulátory navržené aplikací simplexové optimalizace

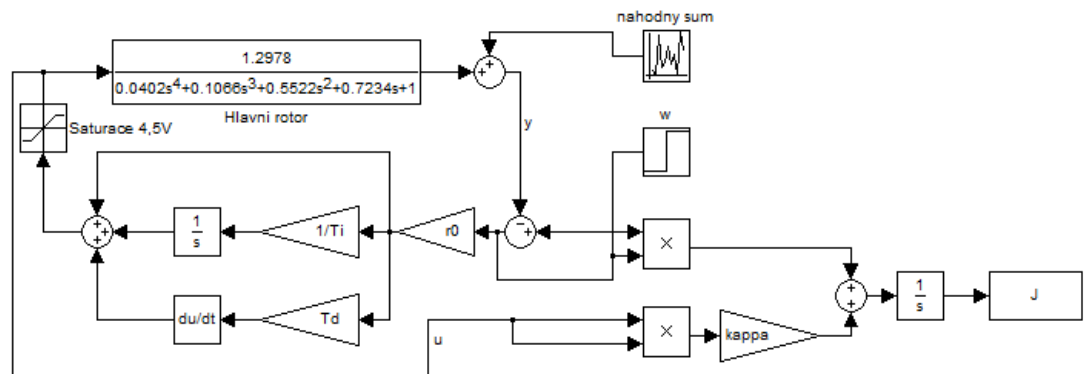
Další sofistikovanější metodou je aplikace numerické simplexové optimalizace, která lze (mimo hledání modelu soustavy) využít i pro nalezení parametrů regulátoru. Pro prvotní odhad parametrů jsou využity parametry vypočítané metodou Ziegler-Nichols. Samozřejmě při hledání optimalizovaných parametrů regulátoru zde hraje roli jiná účelová funkce, než u hledání modelu. Kritérium f (7.9), které se nazývá zobecněné kvadratické kritérium, záleží na regulační odchylce $e(t)$, akční veličině $u(t)$ a je zavedena penalizace κ (kappa), která dovoluje zasahovat do regulačního pochodu a ovlivňovat velikost překmitu. Velikost konstanty κ lze nastavit v rozmezí 0 – 1. Skript funguje na stejném principu jako v kapitole 6.2.2. Parametry regulátoru a výpočet kritéria jsou, na rozdíl od kapitoly 6.2.2, počítány v prostředí SIMULINK. Výsledné parametry jsou předávány do prostředí MATLAB, funkcí $fminsearch()$, která hledá minimum.

Zobecněné kvadratické kritérium (KUPKA, 2013):

$$f = \int_0^t \left\{ [e(t)]^2 + \kappa [u(t)]^2 \right\} dt \quad (7.9)$$

Optimalizace parametrů je provedena částečně v MATLAB skriptu a částečně v prostředí SIMULINK, jelikož je jednodušší kritérium vypočítat v SIMULINKu a v optimalizaci simulační schéma zavolat. Skript tedy spouští simulaci a hledá minimum funkce f s tím, že upravuje parametry T_i , T_d a r_0 a ze simulací jej vezme prostřednictvím

proměnné J . Optimalizační schéma je zobrazeno na obrázku 7.11. Jako příklad je uvedeno hledání parametrů PID regulátoru pro hlavní rotor. Postup hledání parametrů pro regulátor PI a regulátory pro ocasní rotor je analogický.



Obrázek 7.11 – Optimalizační schéma (KUPKA, 2013)

V regulačních pochodech byla penalizace volena tak, aby byly co nejplynulejší změny akční veličiny $u(t)$ a aby byla doba regulace co nejkratší. Ve většině případů vycházel nejlepší regulační pochod při $\kappa = 0,1$. V tabulce 7.3 a 7.4 jsou uvedeny parametry regulátorů nalezené pomocí optimalizační metody, kde jsou pro přehlednost uvedeny i parametry získané v kapitolách 7.2.1 a 7.2.2.

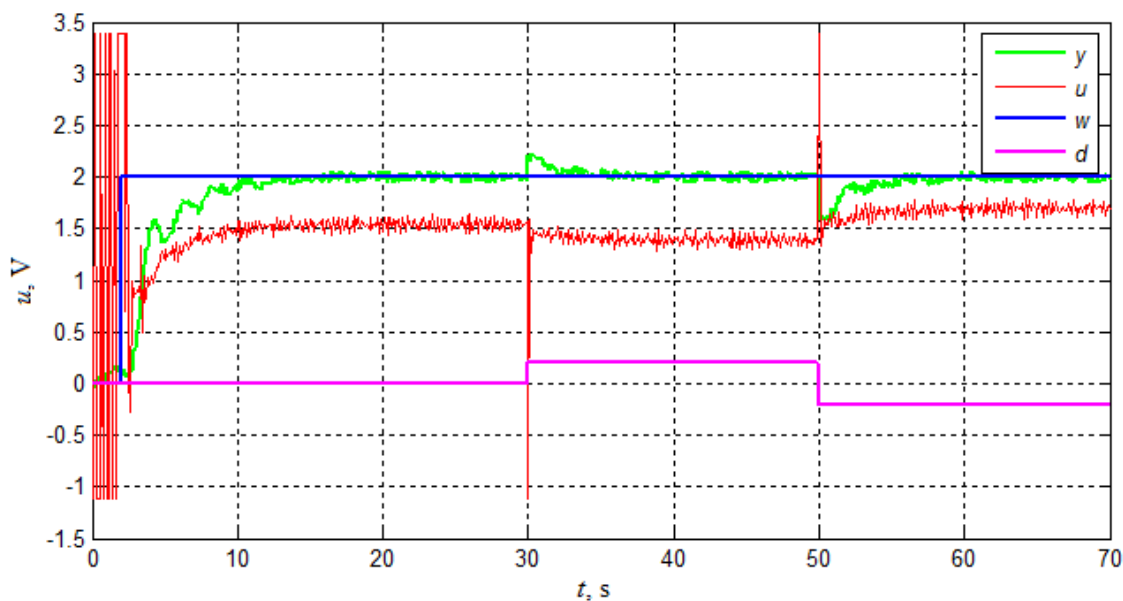
Tabulka 7.3 – Parametry regulátorů hlavního rotoru ZN a OPT

Regulátory ZN	r_0	T_i s	T_d s
PI	0,17	2,17	
PID	0,23	1,275	0,32
Regulátory OPT	r_0	T_i s	T_d s
PI	0,28	0,85	
PID	0,29	0,89	0,34

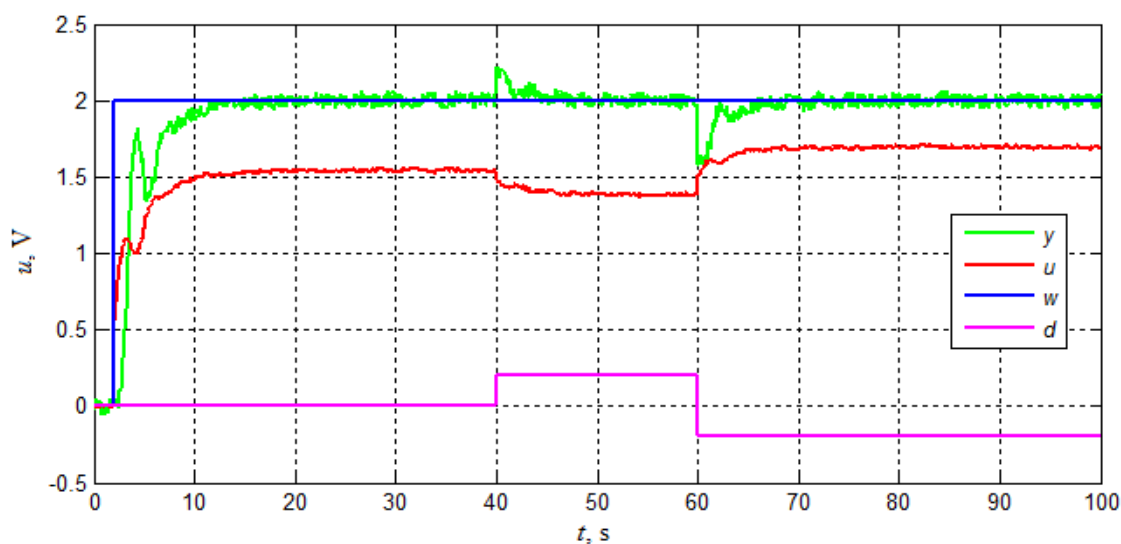
Tabulka 7.4 – Parametry regulátorů ocasního rotoru ZN a OPT

Regulátory ZN	r_0	T_i s	T_d s
PI	0,045	0,95	
PID	0,06	3,5	0,875
Regulátory OPT	r_0	T_i s	T_d s
PI	0,18	2,15	
PID	0,18	1,2	0,3

Na obrázku 7.12 a 7.13 jsou zobrazeny průběhy regulačních pochodů PID a PI regulátorů pro hlavní rotor. V obou případech, především z hlediska doby regulačního pochodu, jsou optimalizované parametry lepší, než parametry nalezené metodou Ziegler-Nichols. Porovnání dob regulačních pochodů je uvedeno v tabulce 7.5.



Obrázek 7.12 – Simulace regulačního pochodu hlavního rotoru, regulátor PID, OPT

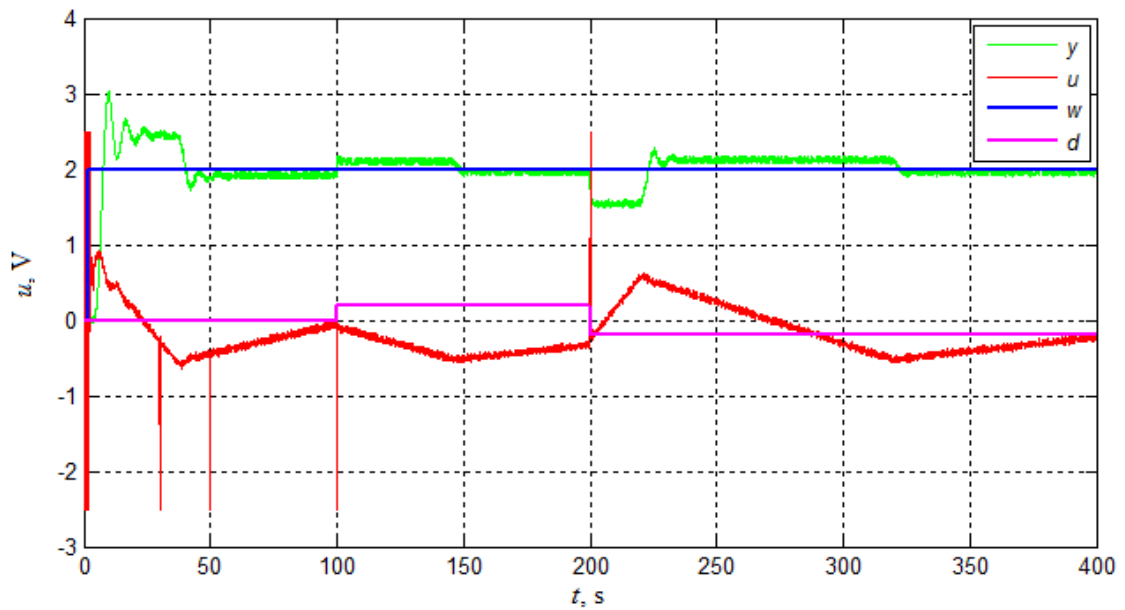


Obrázek 7.13 – Simulace regulačního pochodu hlavního rotoru, regulátor PI, OPT

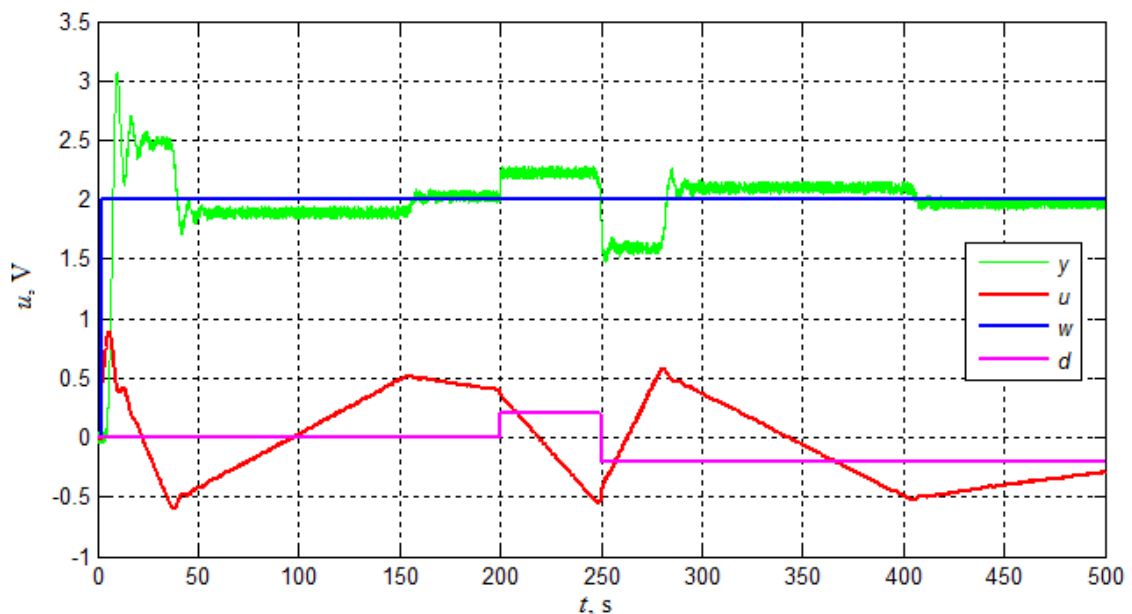
Na obrázku 7.14 a 7.15 jsou zobrazeny průběhy regulačních pochodů PID a PI regulátorů pro ocasní rotor. Zde je viditelná obrovská změna v délce regulačního pochodu, který je zkrácen z tisíců sekund na stovky sekund. Lze prohlásit, že pro řízení laboratorního systému jsou lepší parametry nalezené pomocí optimalizace. Podle simulovaného průběhu by bylo lepší vybrat regulátor PID, ale skutečná použitelnost je zjištěna při testování na laboratorním systému.

Tabulka 7.5 – Vyhodnocení simulací regulačních pochodů

	$t_r - PID, s$	$t_r - PI, s$	$k - PID, \%$	$k - PI, \%$
Hlavní – ZN	20	40	-	-
Hlavní – OPT	10	12	-	-
Ocasní – ZN	700	1300	45	80
Ocasní – OPT	60	160	50	50



Obrázek 7.14 – Simulace regulačního pochodu ocasního rotoru, regulátor PID, OPT



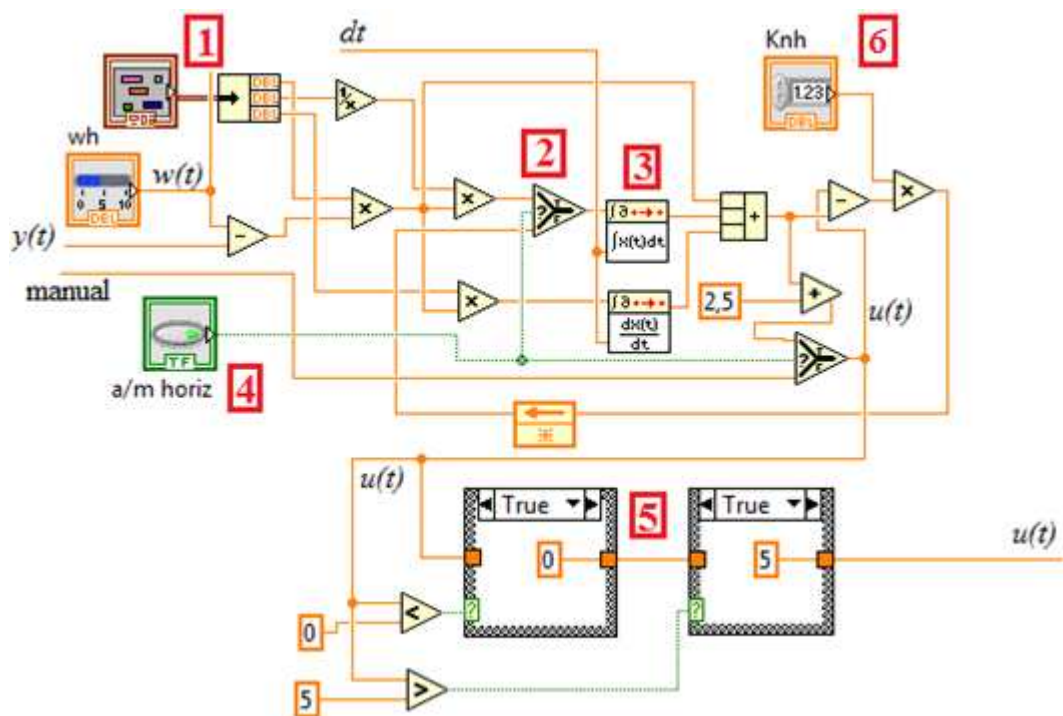
Obrázek 7.15 – Simulace regulačního pochodu ocasního rotoru, regulátor PI, OPT

Snahou bylo nastavit parametry regulátorů tak, aby bylo zabráněno velkým překmitům, které by mohly způsobit nestabilitu reálné soustavy. U hlavního rotoru lze takto parametry nastavit, ale několikrát zmiňovaná setrvačnost ocasního rotoru s mrtvou zónou nedovolí nastavit parametry tak, aby nedošlo k žádnému překmitu. To jedině v případě, že značně dlouhá doba regulačního pochodu by byla vyžadována a to v délce trvání desítek minut, což je prakticky nesmysl. V tabulce 7.5 jsou regulační pochody porovnány z hlediska doby regulačního pochodu a u ocasního rotoru t_r a relativního překmitu k . Ostatní kritéria, jako jsou například různé regulační plochy, nemají v této úloze potenciál k vyhodnocení optimálního regulačního pochodu. Dvurotorový laboratorní systém se chová velmi nevyzpytatelně a jedině testování ukáže vhodnost navržených regulátorů.

8 Implementace regulace v LabVIEW

Aby bylo možné zjistit vhodnost nalezených parametrů z předchozí kapitoly, je potřeba otestovat regulaci na laboratorním dvourotorovém systému. Je potřeba aplikaci pro měření a získávání experimentálních dat rozšířit, aby umožňovala nejen měřit přechodové charakteristiky, ale také regulovat laboratorní systém. Aplikace má implementovány dva regulátory, každý pro jeden rotor. Aby se zachovala možnost ruční manipulace s laboratorním systémem, je vytvořen regulátor s možností přepínání manuálního a automatického režimu. Přepínání režimů je řešeno pomocí bez-nárazového přepínání, aby nedocházelo ke skokům akční veličiny při přepnutí. Díky umožnění ručního řízení, je možno zapnout regulaci jednoho rotoru (bude v automatickém režimu) a druhý nechat v manuálním režimu a generovat poruchy. Aplikace tedy umožňuje testování regulátorů provádět jak na změnu žádané hodnoty, tak na poruchy generované druhou větví. Přepínání režimů auto-manuál, včetně nastavení parametrů obou regulátorů, je umožněno na čelním panelu aplikace. Stejně jako v předchozí verzi aplikace i tato aplikace umožňuje ukládat veškerá potřebná data, kterými jsou průběhy všech veličin. Na obrázku 8.1 je zobrazen jeden z regulátorů implementovaný do LabVIEW. Druhý regulátor je totožný.

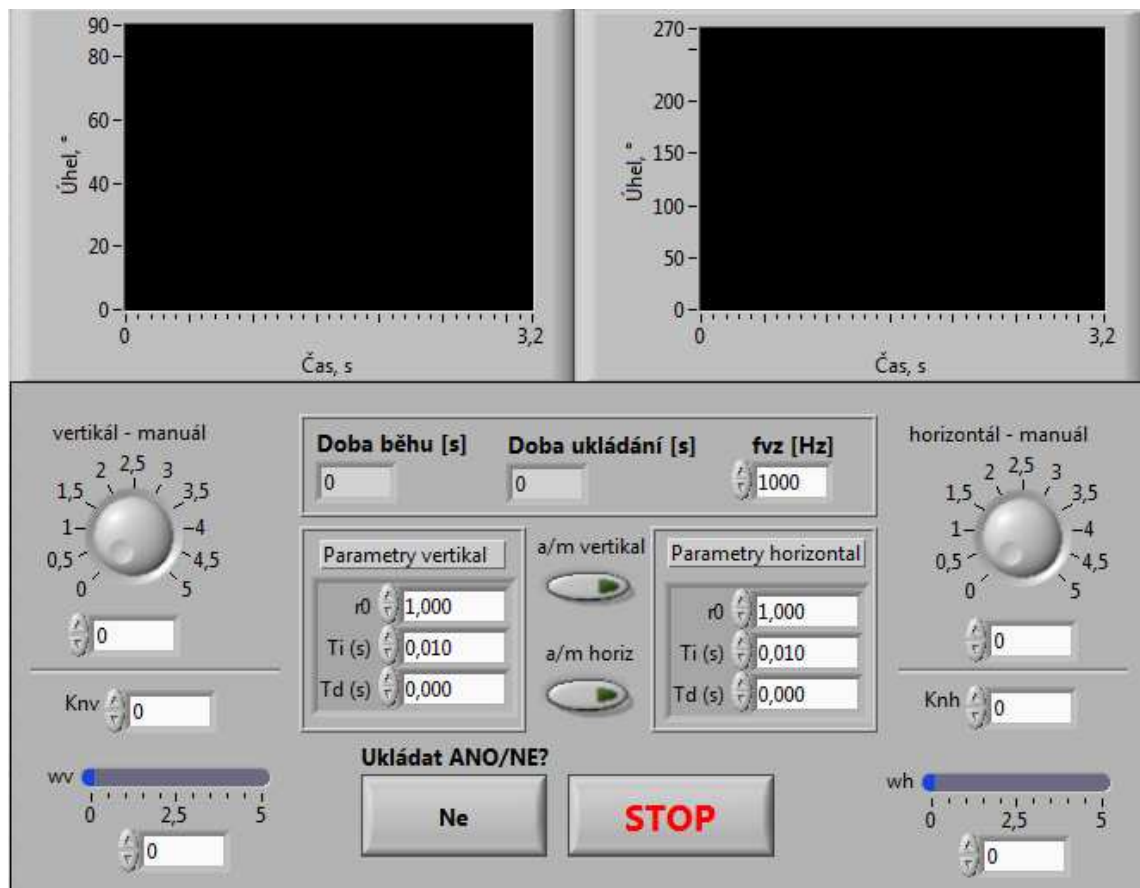
Může se zdát, že regulátor je v nějakém speciálním zapojení, ale i zde je použito standardní zapojení regulátoru, stejně jako v simulacích. Na obrázku 8.1 jsou vyznačeny procesní veličiny, které jsou přivedeny do regulátoru. Všechny veličiny se dále ukládají. Ukázka čelního panelu aplikace je zobrazena na obrázku 8.2.



Obrázek 8.1 – Implementace regulátoru v LabVIEW

Popis vybraných důležitých bloků implementované regulace:

1. PID Gains – struktura, do které uživatel zadává na čelním panelu parametry regulátoru. Následuje rozdělení struktury na jednotlivé konstanty.
2. Select – jedná se o přepínač, který se přepíná přivedenou logickou 0 nebo 1. V programu jsou tyto bloky dva a přepínají automatický a manuální režim.
3. Integral, Derivative – speciální bloky, které jsou zde jako integrátor a derivátor, zapojené ve standardním zapojení PID. K oběma blokům musí být přivedena časová diference dt , která závisí na nastavené vzorkovací frekvenci. Nedodržení správné diference způsobí špatně vypočtené akční zásahy.
4. Push Button – tlačítko, kterým se přepíná režim auto-manuál.
5. Case Structure – dvě case struktury v zapojení *If-else*, které zde slouží k omezení akčního zásahu, jelikož je povoleno pouze rozmezí 0 – 5 V. Pokud by regulátor (například vlivem derivační složky) vytvořil větší či menší akční zásah, bude omezen právě na tento napěťový rozsah.
6. Konstanta K_n – konstanta v rozsahu 0 – 1, určuje míru tlumení bez nárazového přepnutí.



Obrázek 8.2 – Čelní panel kompletní aplikace pro regulaci laboratorního systému

8.1 Testování regulace na laboratorním systému

Dvourotorový laboratorní systém je velmi citlivý a reálné průběhy se od simulace podstatně liší. Tato skutečnost byla očekávána, protože nalezené modely, které sloužily pro návrh regulátorů, jsou zatíženy chybou, z důvodu nelinearit a snahou aproximovat soustavy lineárním modelem. Hlavní rotor má sice rozdílný skutečný průběh regulačního pochodu oproti simulaci, ale nalezený regulátor úspěšně reguluje hlavní rotor v daném omezeném rozsahu a reguluje i poruchy, které vznikají otáčkami ocasního rotoru. Regulátor ocasního rotoru dokáže udržovat stabilitu soustavy, ale jen v případě, že změny žádané hodnoty jsou malé. Regulátor ocasního rotoru je spíše vhodný na regulaci malých poruch, které vznikají změnami otáček hlavního rotoru, jako u reálné helikoptéry.

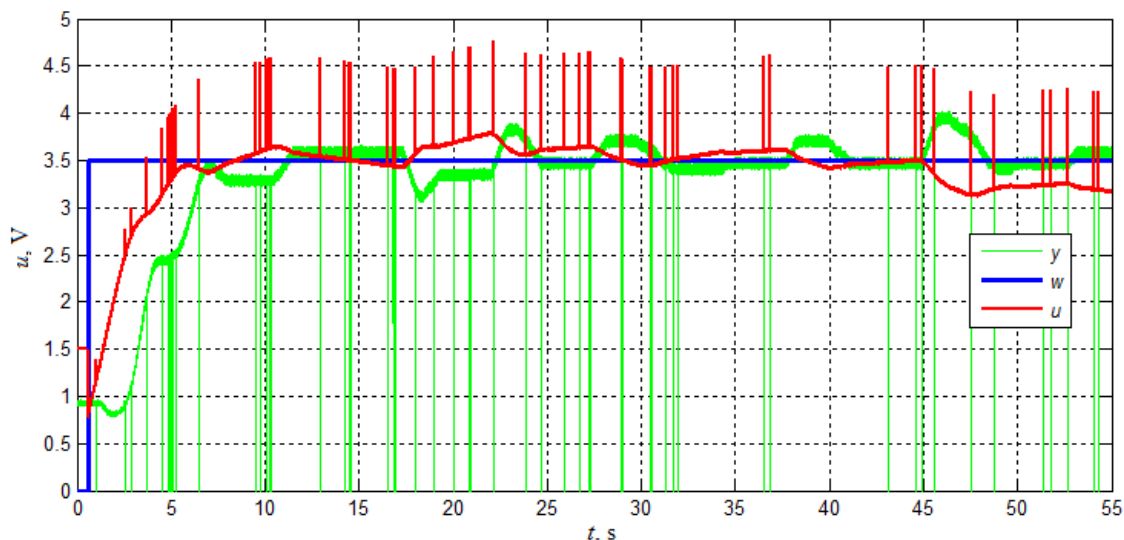
Testování regulačních pochodů probíhalo stejně jako v simulaci, vždy s odpojenou druhou větví. To znamená, že regulace hlavního rotoru probíhala s vypnutým regulátorem ocasního rotoru, který byl nastaven na manuální režim a sloužil k vytváření poruch. Totéž u měření ocasního rotoru. Ocasní rotor byl testován na regulaci poruch, které tvoří hlavní rotor změnami svých otáček. Regulace na žádanou hodnotu ocasního rotoru vykazuje nestabilitu, proto zůstane regulátor ocasního rotoru pouze pro regulaci poruch. Vypínání druhé větve plyne z faktu, že dvourotorový systém byl rozdělen na dvě SISO soustavy a interakce vzájemných vlivů rotorů jakožto MIMO systému, byly tvořeny poruchami v podobě manuálních změn.

Kvalita regulačních pochodů byla zkoumána jen na základní požadavky, jako je doba regulace a relativní překmit. Další podrobnější zkoumání regulačních pochodů nebylo provedeno, protože nalézt kvalitnější nastavení parametrů pro takto nevyzpytatelný systém není snadné a vyžadovalo by náročné experimentování a ruční hledání parametrů. Ocasní rotor se navíc nedá kvalitně řídit, protože mezi stabilitou a kvalitou je v tomto případě velmi tenká hranice. Snaha získat regulační pochod vyšší kvality rychle přechází do nestability. Proto je vhodnější nechat parametry regulátoru nalezené numerickou optimalizací, se kterými zůstává URO ocasního rotoru stabilní.

8.1.1 Vyhodnocení regulace hlavního rotoru

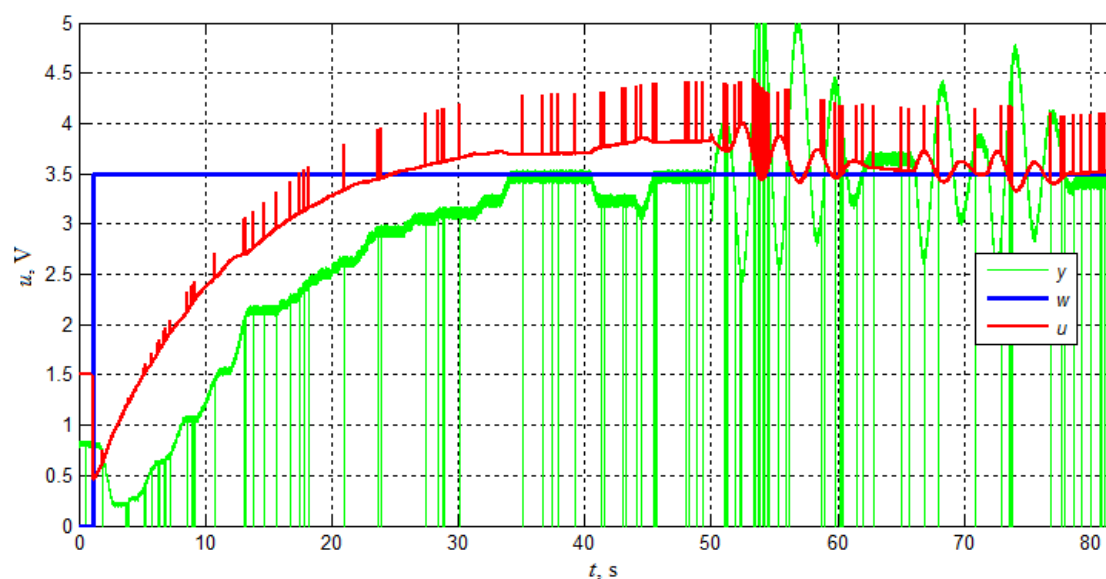
Regulace hlavního rotoru byla testována s nalezenými parametry pomocí obou metod, nejprve na žádanou hodnotu a poté na regulaci poruchy. Regulátory s parametry nalezenými oběma metodami stabilně regulují reakci na žádanou hodnotu. S regulací poruchy je to jiné, malou poruchu zvládají oba regulátory, ale větší poruchu zvládá stabilně regulovat pouze regulátor s parametry nalezenými metodou ZN, což je překvapivé. Byly vybrány dva regulační pochody, které zobrazují úspěšnou regulaci žádané hodnoty a poté regulaci malých poruch s regulátorem s optimalizovanými parametry (obrázek 8.3), a úspěšnou regulaci žádané hodnoty, malých poruch způsobených změnami otáček ocasního rotoru a také velkou poruchu ručním rozhoupáním hlavního rotoru, s regulátorem s parametry ZN (obrázek 8.4). Na první pohled je zřejmé, že regulátor s optimalizovanými parametry má kvalitnější regulační pochod, protože ustálení je již v čase $t = 12$ s, zatímco u regulátoru ZN je ustálení až v čase $t = 34$ s.

Pro regulaci byl použit pouze regulátor PI, protože derivační složka PID regulátoru způsobovala obrovské výkyvy akční veličiny, téměř v rozmezí 0 – 5V. Bylo testováno zapojení speciálního bloku v LabVIEW, pro filtraci akční veličiny, ale filtr nedokázal tyto vlivy odstranit, proto byl použit regulátor PI, který úspěšně reguluje žádanou hodnotu i poruchy.



Obrázek 8.3 – Skutečný regulační pochod hlavního rotoru, regulátor PI, OPT

Hlavní rotor byl nejprve vybuzen v manuálním režimu akční veličinou $u = 1,5$ V, z důvodu nastavení počátku omezeného regulovaného rozsahu. V čase $t = 2$ s byla přivedena žádaná hodnota $w = 3,5$ V. S jejím přivedením byla řídicí aplikace přepnuta do automatického režimu regulace. V čase $t = 12$ s se hlavní rotor ustálil na žádané hodnotě, $y = 3,5$ V. V časech $t = 17$ s, $t = 23$ s, $t = 27$ s, $t = 37$ s a $t = 45$ s, byly postupně přivedeny malé poruchy, vlivem změn otáček ocasního rotoru.



Obrázek 8.4 – Skutečný regulační pochod hlavního rotoru, regulátor PI, ZN

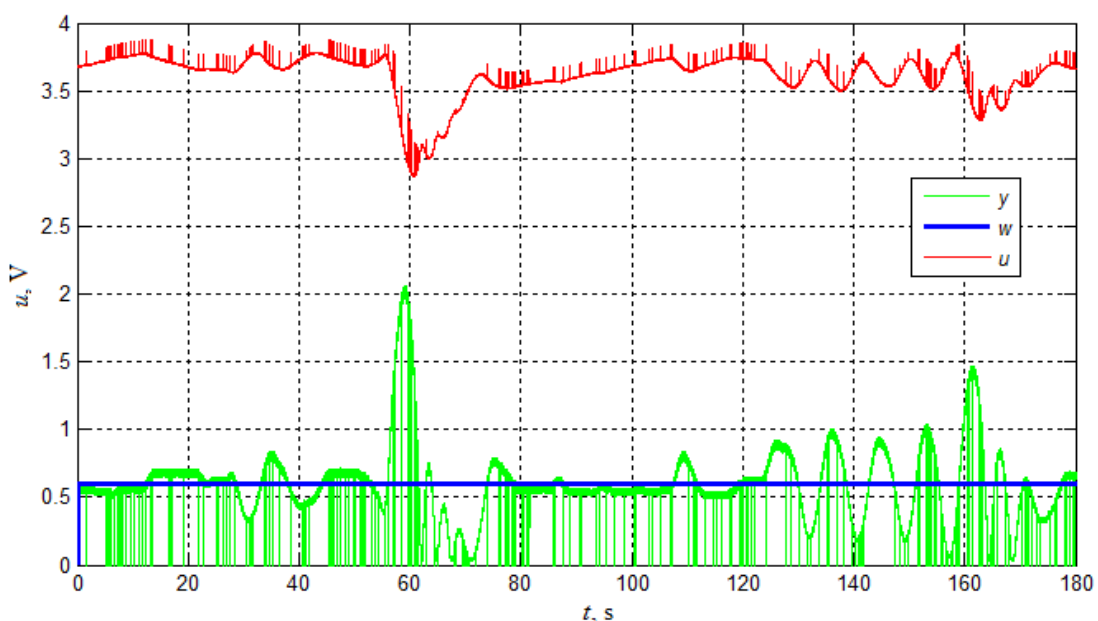
Hlavní rotor byl opět nejprve vybuzen v manuálním režimu akční veličinou $u = 1,5$ V, pro nastavení počátku omezeného regulovaného rozsahu. V čase $t = 2$ s byla přivedena žádaná hodnota $w = 3,5$ V. V čase $t = 40$ s a $t = 45$ s byly přivedeny malé poruchy, manuální změnou otáček ocasního rotoru. V čase $t = 50$ s a $t = 65$ s bylo manuálně drknuto do raménka laboratorního modelu. Regulátor s parametry ZN dokázal úspěšně stabilizovat hlavní rotor s přivedením razantní poruchy.

Pro vyhodnocení regulačních pochodů postačuje pouze údaj doby regulace, kdy se výstup soustavy ustálí na 95% žádané hodnoty. Oba regulační pochody jsou bez překmitu, stejně jako simulované regulační pochody.

8.1.2 Vyhodnocení regulace ocasního rotoru

Ocasní rotor byl testován také s parametry nalezenými pomocí obou metod. Zde je problém stability větší, soustava ocasního rotoru je velmi citlivá. Regulátor s parametry nalezenými pomocí metody ZN nebyl schopen stabilně regulovat ocasní rotor na žádanou hodnotu, dokonce nebyl schopen ani regulovat poruchu. Parametry nalezené metodou ZN jsou absolutně nevhodné. Regulátor nestihne zareagovat ani na malou změnu poruchy a soustava se rozkmitá.

Regulátor s parametry nalezenými optimalizační metodou má výsledky lepší než ZN, ale také nedokáže regulovat na žádanou hodnotu. Jakákoliv změna žádané hodnoty soustavu rozkmitá. Jak již bylo řečeno, mezi kvalitou a stabilitou je velmi tenká hranice a nalézt parametry regulátoru, které dokážou udržet stabilitu při reakci regulátoru na žádanou hodnotu, je téměř nadlidský úkol. Snahou bylo alespoň udržet ocasní rotor ve stabilní poloze, pokud dojde ke změnám otáček hlavního rotoru, který je v tomto případě zdroj poruch. Tohoto požadavku bylo dosaženo, pokud se nehledí na kvalitu regulačního pochodu. Nutno dodat ještě jeden fakt, že regulace na odstranění poruchy musí probíhat se zapnutým hlavním rotorem a pak měnit jeho otáčky. Pokud je hlavní rotor zastaven, nelze vůbec regulovat ocasní rotor, z důvodu setrvačnosti, která je odstředivou silou hlavního rotoru zmírněna.



Obrázek 8.5 – Skutečný regulační pochod ocasního rotoru, regulátor PI, OPT

Nejprve byl ocasní rotor ustálen manuálně, akční veličinou $u = 3,7$ V, při zapnutém hlavním rotoru (není zobrazeno). V čase $t = 0$ s byla soustava přepnuta na automatický režim po dobu 13 s, kde bylo zkoumáno úplné ustálení. V časech $t = 13$ s, $t = 25$ s byla přivedena malá porucha změnami otáček hlavního rotoru. V čase $t = 50$ s byla přivedena porucha větší změnou otáček hlavního rotoru. Soustava měla tendenci se rozkmitat a narazila do spodního dorazu, ale v čase $t = 80$ s došlo k ustálení. Poté byly přivedeny další malé poruchy, přičemž porucha v čase $t = 123$ s měla opět tendenci soustavu rozkmitat, ale opět byla schopna soustavu ustálit, za cenu velmi nekvalitního regulačního pochodu, který trval přibližně 60 s. Tento regulační pochod je nejlepší, kterého bylo dosaženo. Doba regulace se musí počítat na desítky s. Postupem návrhu regulátoru pomocí experimentální identifikace i s použitím numerické metody, nelze dosáhnout lepšího výsledku.

Závěr

V diplomové práci bylo postupováno podle zadání a dalo by se říci, že bylo dosaženo všech úkolů. Práce je sepsána tak, jak bylo postupováno v laboratoři a bylo snahou vždy po úspěšném experimentování všechny informace ihned sepsat, aby do práce nebyly zavedeny chyby.

Nejprve byl sestaven program v LabVIEW, pomocí kterého bylo možné pohybovat s laboratorním modelem a postupně se s ním seznamovat. Tento program představuje vlastně manuální ovládání, které bylo v konečném kompletním programu ponecháno, pro budoucí studenty, aby mohli také získat zkušenosti s ovládáním modelu. Při ručním ovládání modelu byla pozorována omezení, narážení ramene do dorazů, byla proměřována mrtvá pásma stejnosměrných motorů a celkově pozorovány interakce obou rotorů. V této fázi ale nebylo možné zjistit nic o dynamice laboratorního modelu, proto musel být ovládací program rozšířen o možnost ukládat experimentální data v podobě přechodových charakteristik obou rotorů, jako reakcí na jednotkový skok, na základě nichž byly tvořeny modely v podobě přenosových funkcí.

U hlavního rotoru byla nejprve získána statická charakteristika, která odhalila a potvrdila teoretické předpoklady o nelinearitě. V tuto chvíli nastala diskuse a hlubší seznamování se s laboratorním modelem, jejímž závěrem bylo omezit pracovní rozsah hlavního rotoru na lineární část, aby bylo možné použít lineárních postupů návrhu regulátoru. Hlavní rotor je statickou soustavou a nebyly problémy s návrhem regulátorů, které dokážou regulovat na žádanou hodnotu a také kompenzovat poruchové vlivy.

Ocasní rotor je astatickou soustavou, s velkou setrvačností a velkou citlivostí. Navíc způsob, jakým je rozlišen směr otáčení ocasního rotoru, pomocí rozdělení řídicího napětí na dvě části, představuje problém s mrtvým pásmem, které je způsobené konstrukcí stejnosměrného motoru a vyskytuje se u každého elektromotoru. Velká setrvačnost a mrtvá zóna způsobují fakt, že nalézt správné parametry regulátoru je téměř nadlidský úkol. Byly nalezeny uspokojivé parametry, ale pouze z hlediska stability a pouze pro regulaci poruchy.

Matematické modely, které byly sestaveny na základě přechodových charakteristik, jsou to nejlepší, co lze vytvořit pro co nejlepší nalezení parametrů regulátorů. Hlavní rotor byl aproximován statickým přenosem IV. řádu, ocasní rotor byl aproximován astatickým přenosem III. řádu. Oba modely byly nalezeny aplikací numerické optimalizace, z experimentálních přechodových charakteristik. Na základě těchto přenosových modelů byly navrženy regulátory metodou Ziegler-Nichols a také aplikací numerické optimalizace.

Navržené regulátory byly nejprve simulovány a bylo sledováno, zda je dodržen základní požadavek, kterým je stabilita regulačního pochodu a tím celého URO. Simulace prokázaly stabilitu a bylo možné se pustit do dalšího úkolu, kterým byla implementace regulátoru do původní aplikace pro ovládání a tím zjistit vhodnost nalezených regulátorů v praxi tj. přímým zkoumáním regulačních pochodů na laboratorním modelu.

Laboratorní model je soustavou MIMO, kde působí vzájemné interakce a navíc je nelineární. Hlavním a nejnáročnějším úkolem diplomové práce bylo vytvořit funkční a odladěnou aplikaci v LabVIEW, proto bylo rozhodnuto na soustavu nahlížet jako na dvě soustavy SISO, které na sebe vzájemně působí poruchami. K tomu, aby bylo možné toto

zjednodušení použít, je vždy regulován každý rotor zvlášť, aby nedocházelo k vzájemnému ovlivňování regulátorů.

Aby nedošlo k nedorozumění, v celé práci bylo pracováno v jednotkách V a to z důvodu snahy zachovat jednotnost. Regulovaná veličina obou rotorů je úhel, ale informace z optických snímačů je v jednotkách V a akční veličina také. Všechny průběhy (s výjimkou statické charakteristiky) jsou na ose Y také v jednotkách V. Mezi napětím z čidel a skutečným úhlem je spojitost. Pokud by byla potřeba zjistit úhel, stačí pouze informaci v jednotkách V vynásobit konstantou. Konstanta pro hlavní rotor je $5/90$, konstanta pro ocasní rotor je $5/270$, jedná se o poměr rozsahu napětí z čidla ku celkovému rozsahu úhlu vychýlení.

Tato práce je pro autora obrovským přínosem. Díky experimentování s reálnou soustavou získal cenné zkušenosti s identifikováním poměrně složitého laboratorního systému a s nastavováním regulátorů. V rámci práce autor vyzkoušel velmi mnoho teoretických znalostí z různých předmětů v průběhu studia, díky čemuž si mohl velmi mnoho informací spojit a posunout se dále v chápání problematiky oboru.

Řídící a ovládací aplikace bude poskytnuta KŘP, kde v budoucnu může sloužit pro rozšíření výuky a zahrnutí laboratorního systému upoutané helikoptéry do výuky předmětů zabývajících se laborováním.

Dále je možné na tuto diplomovou práci navázat a využít již zjištěných poznatků. Hlavním přínosem je aplikace v LabVIEW. Aplikace by mohla sloužit komukoliv, kdo bude pracovat s dvurotorovým systémem a například hledat přesnější model nebo navrhovat regulátor pomocí jiného algoritmu, který by mohl být do ní implementován.

Literatura

- BALÁTĚ, J. 2004. *Automatické řízení, 2. vydání*. Praha: BEN – Technická literatura. 663 s. ISBN 80-7300-148-9.
- DUŠEK, F. 2012. *Teorie řídicích systémů – přednášky*. [cit. 15. 3. 2014].
- HAVLÍČEK, L. 2010. *Modelování a řízení vícerozměrové soustavy [disertační práce]*. Pardubice: Univerzita Pardubice. Fakulta elektrotechniky a informatiky. 89 s.
- HORÁK, R. 2011. *Řízení vertikálního pohybu [disertační práce]*. Pardubice: Univerzita Pardubice. Fakulta elektrotechniky a informatiky. 55 s.
- KUPKA, L., JANEČEK, J. 2007. *Matlab & Simulink: řešené příklady*. Lanškroun: TG tisk a SOŠ a SOU. 224 s. ISBN 978-80-239-9532-9.
- KUPKA, L. 2013. *Softwarové prostředky pro řízení, přednáška 5. Optimalizace I*. [cit. 15. 3. 2014].
- MACHÁČEK, J. 2008. *Robustní řízení vícerozměrové soustavy*. [online]. [cit. 16. 1. 2014]. Dostupné na: http://pernerscontacts.upce.cz/12_2008/machacek.pdf
- NATIONAL INSTRUMENTS CORPORATION. 2014a. *Answers to Frequently Asked Questions about NI-DAQmx*. [online]. [cit. 6. 2. 2014]. Dostupné na: <http://www.ni.com/white-paper/3021/en/>
- NATIONAL INSTRUMENTS CORPORATION. 2014b. *Learn 10 Functions in NI-DAQ*. [online]. [cit. 6. 2. 2014]. Dostupné na: <http://www.ni.com/white-paper/2835/en/>
- VLACH, J., HAVLÍČEK, J., VLACH, M. 2008. *Začínáme s LabVIEW*. Praha: BEN – technická literatura. 247 s. ISBN 978-80-7300-245-9.