

Nová úloha na laboratorním modelu helikoptéry CE150 v prostředí MATLAB

New laboratory in MATLAB for
CE150 helicopter model

Vilém Frolka

Bakalářská práce
2005



Univerzita Tomáše Bati ve Zlíně
Fakulta technologická

Univerzita Tomáše Bati ve Zlíně
Fakulta technologická
Institut řízení procesů a aplikované informatiky
akademický rok: 2004/2005

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Vilém FROLKA**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Automatické řízení a informatika**

Téma práce: **Nová úloha na laboratorním modelu helikoptéry
CE150 v prostředí MATLAB**

Zásady pro vypracování:

1. Provedte literární rešerši popisující MATLAB a Real-Time Toolbox.
2. Realizujte připojení modelu k počítači a uveďte popis tohoto zapojení.
3. Ověřte regulaci jednoduchým regulátorem.
4. Vytvořte návod k laboratorní úloze.

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. MATLAB: Function Reference, Version 7, MathWorks Inc., Natick, USA, 2004.
2. MATLAB: Getting Started With MATLAB, Version 7, MathWorks Inc., Natick, USA, 2004.
3. MC101 Motor Controller Card: User's Manual. Humusoft, s.r.o., Praha, 1992.
4. Real-Time Toolbox for MATLAB. User's Manual. Humusoft, s.r.o., Praha, 1999.

Vedoucí bakalářské práce:

Ing. Karel Perůtka

Institut řízení procesů a aplikované informatiky

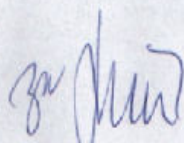
Datum zadání bakalářské práce:

21. února 2005

Termín odevzdání bakalářské práce:

3. června 2005

Ve Zlíně dne 17. února 2005



prof. Ing. Josef Šimoník, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel institutu

Děkuji vedoucímu bakalářské práce Ing. Karlu Perůtkovi za odborné rady a podněty, které mi poskytoval během řešení bakalářské práce.

Souhlasím s tím, že s výsledky mé práce může být naloženo podle uvážení vedoucího bakalářské práce a ředitele ústavu. V případě publikace budu uveden jako spoluautor.

Prohlašuji, že jsem na celé bakalářské práci pracoval samostatně a použitou literaturu jsem citoval.

Ve Zlíně, 27. 05. 2005

.....

Vilém Frolka

ABSTRAKT

Abstrakt česky

Hlavním cílem bakalářské práce bylo vytvořit novou laboratorní úlohu na laboratorním modelu helikoptéry CE150 v prostředí MATLAB. Za tímto účelem byla provedena literární rešerše v oblasti MATLABu, Real – Time toolboxu, karty MC101 a samočinně se nastavující regulátory. Dále bylo ověřeno připojení helikoptéry k PC. Pro vytvoření návodu bylo třeba provést identifikaci a výběr vhodné metody regulace. Závěrem je pak přiložen návod k nové laboratorní úloze.

Abstrakt ve světovém jazyce

New laboratory in MATLAB for CE150 helicopter model was the main goal of this thesis. Firstly, literary background research from the areas of MATLAB, Real-Time toolbox, MC101 technological card and self-tuning controllers was done for this purpose. This is followed by the verification of the connection between helicopter and PC. Next, it was necessary to identify the model and choose the appropriate method of control. Finally, the instructions of the new laboratory are enclosed as an appendix of the thesis.

OBSAH

ÚVOD.....	8
1 TEORETICKÁ A STUDIJNÍ ČÁST	9
1.1 MATLAB	9
1.1.1 Co je MATLAB.....	9
1.1.2 Co MATLAB nabízí.....	9
1.1.3 Systém MATLABu	10
1.1.4 Toolboxy	11
1.1.5 Simulink	11
1.1.6 Struktura MATLABu	12
1.1.7 Start MATLABu.....	13
1.1.8 Proměnné a funkce	13
1.2 REAL TIME TOOLBOX.....	14
1.2.1 Úvod	14
1.2.2 Vlastnosti.....	15
1.2.3 Načtení hardwarových ovladačů	15
1.2.4 Definování časovačů	16
1.2.5 Komunikace časovačů	17
1.2.6 Spouštění a zastavování časovačů	17
1.2.7 Příkazová řádka	18
1.3 KARTA MC-101	18
1.3.1 Popis produktu.....	18
1.3.2 Parametry karty	18
1.4 CE 150 HELICOPTER.....	19
1.4.1 Laboratorní model	19
1.4.2 Model helikoptéry	20
1.4.3 Popis systému	20
1.4.4 Specifikace	20
1.5 SAMOČINNĚ SE NASTAVUJÍCÍ REGULÁTORY	21
1.5.1 Úvod	21
1.5.2 Identifikace procesu.....	21
2 EXPERIMENTÁLNÍ ČÁST.....	23
2.1 POPIS PŘIPOJENÍ MODELU K POČÍTAČI.....	23
2.1.1 Popis připojení modelu k PC.....	23
2.1.2 Kabel k motorům	23
2.1.3 Měřicí kabel.....	24
2.1.4 Řídící kabel.....	25
2.1.5 Kompletní zapojení helikoptéry	25
2.2 IDENTIFIKACE MODELU	27
2.2.1 Postup identifikace	27
2.2.2 Provedení identifikace	28
2.2.3 Statické charakteristiky	28
2.3 ADAPTIVNÍ ŘÍZENÍ	31
2.3.1 Adaptivní regulátory.....	32
2.4 ŘÍZENÍ LABORATORNÍHO MODELU.....	33
2.4.1 Regulátor založený na minimalizaci lineárně-kvadratického kritéria	33
2.4.2 Modifikace Ziegler – Nichols používající Takahashi regulátor	35
ZÁVĚR.....	36
SEZNAM POUŽITÉ LITERATURY	38
SEZNAM OBRÁZKŮ.....	39

SEZNAM TABULEK	40
SEZNAM PŘÍLOH.....	41

ÚVOD

Úkolem této práce je ověření regulace jednoduchým regulátorem a vytvoření laboratorní úlohy. Vzhledem k tomu, že laboratorní model helikoptéry CE150 má dva motory, byla provedena identifikace každého motoru zvlášť. Pro zpřesnění identifikace v horizontálním směru byly realizovány měření pro 5 pozic helikoptéry. Naměřená data jsou zpracována do statické charakteristiky, abychom zjistili vhodnou oblast pro linearizaci.

Vzhledem k vazbě mezi hlavní vrtulí a zadní vrtulí byl použit pro řízení adaptivní regulátor zvlášť pro každý motor. V tomto je totožná i laboratorní úloha. Studenti budou zkoušet různé adaptivní regulátory na model a sledovat kvalitu regulace.

Model helikoptéry je propojen se spojovacím modulem a ten je spojen s kartou MC101 umístěnou v počítači. K řízení a sběru dat z helikoptéry slouží Real Time Toolbox, který pracuje pod programem MATLAB a předává informace z karty MC101 do MATLABu v reálném čase. Díky tomuto programu je velmi jednoduché měřit a ovládat helikoptéru. Použijeme-li program Simulink, pak vytvoříme schéma pro měření či regulaci velmi elegantně a to pouhým přetáhnutím bloků a jejich pospojováním.

1 TEORETICKÁ A STUDIJNÍ ČÁST

1.1 MATLAB

1.1.1 Co je MATLAB

Název vznikl z anglického *matrix laboratory*. MATLAB je vysoce výkonný jazyk pro technické výpočty. Je určen pro výpočty, modelování, návrh algoritmů, simulaci, analýzu dat, měření a zpracování signálu, návrhy řídicích a komunikačních systémů.

MATLAB je interaktivní systém, jehož základní prvek dat je pole, které nevyžaduje stanovení rozměrů. To nám dovoluje řešit mnoho technických problémů, zvláště v maticovém a vektorovém vyjádření. [1]

1.1.2 Co MATLAB nabízí

- rychlé výpočetní jádro
- otevřený a rozšiřitelný systém
- 2D a 3D působivá grafika
- konfigurovatelné uživatelské rozhraní MATLAB Desktop
- velké množství aplikačních knihoven
- programovací jazyk 4. generace
- objektové programování
- integrace s jazykem Java
- podpora vícerozměrných polí a uživatelsky definovaných datových struktur
- interaktivní nástroje pro tvorbu grafického uživatelského rozhraní
- interaktivní průvodce importem dat
- zvukový vstup a výstup, animace
- komunikace s externím přístrojovým vybavením
- výpočetní jádro pro programy psané ve Fortran a jazyce C
- distribuce nezávislých uživatelských aplikací: překlad do jazyka C, runtime modul, WWW technologie
- rozsáhlá tištěná i hypertextová on-line dokumentace

1.1.3 Systém MATLABu

Systém je tvořen z následujících částí [1]:

Vývojové prostředí

Je tvořeno soustavou nástrojů a prostředků, které pomáhají používat funkce a soubory. Obsahuje MATLAB desktop, příkazové okno, okno historie příkazů, okno pracovního prostředí a okno aktuálního adresáře.

Výpočetní jádro

Obsahuje rozsáhlou sbírku výpočetních algoritmů a funkcí jako jsou suma, sinus, cosinus, komplexní výpočty, inverzní matice, Fourierova transformace a několik set dalších.

Jazyk MATLABu

MATLAB obsahuje maticový jazyk na vysoké úrovni s funkcemi, strukturou dat, vstupy a výstupy. To umožňuje rychle vytvořit obsáhlé a komplikované programy.

Grafika

MATLAB má rozsáhlé možnosti zobrazování vektorů a matic jako obrazce. To zahrnuje rozsáhlé funkce pro dvojrozměrnou a trojrozměrnou grafiku, dále funkce pro zpracování obrazu, animace a prezentace grafiky. MATLAB dále umožňuje plné přizpůsobení grafického výstupu z vytvořených aplikací.

Externí rozhraní MATLABu

Je to knihovna, která dovoluje psát programy v jazyce C a Fortran, které fungují společně s MATLABem. To zahrnuje možnost volání programů z MATLABu, použití MATLABu jako výpočetní jádro a taky jej lze použít pro čtení a zápis MAT-souborů.

1.1.4 Toolboxy

Jádro MATLABu je možné rozšířit o problémově orientované funkce řešící specializovaný okruh problémů. Tyto funkce jsou soustředěné do toolboxu. Díky systému Handle Graphics dovolují tyto toolboxy využívat značně sofistikované matematické postupy i uživatelům nepříliš zblhlým v matematice. Umožní uživateli jednoduše zadat jeho data, vybrat metodu a spustit výpočet. Zde je uvedeno pár vybraných toolboxů rozšiřujících standardních instalací MATLABu.

- Real-time toolbox – toolbox pro řízení laboratorních modelů a průmyslových strojů pomocí PC počítače přes technologickou kartu nebo sériové rozhraní
- Polynomial toolbox – toolbox pro symbolické řešení diofantických rovnic a práce s polynomy
- Self-tuning Control Simulink Library – knihovna bloků pro Simulink pro simulaci diskrétních samočinně se nastavujících regulátorů
- CONTSID toolbox – slouží pro v čase spojitou identifikaci měřených nebo regulovaných systémů

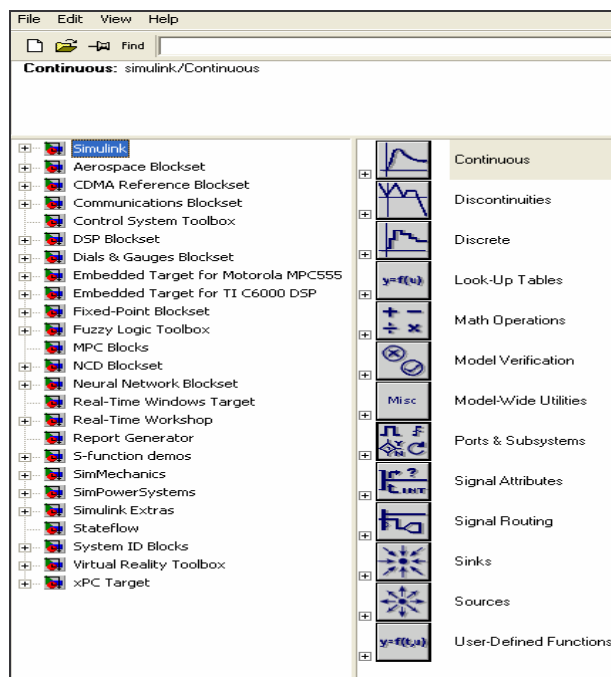
1.1.5 Simulink

Je určen pro simulaci dynamického systému pokud známe jeho matematický popis. S jeho pomocí lze určit časové průběhy výstupních veličin v závislosti na časovém průběhu vstupních veličin a na počátečním stavu.

Simulink využívá grafický zápis. Z příslušné knihovny si myší přetahujeme bloky a poté je pouze pospojujeme a máme sestavený model.

Jako vstupy lze používat bloky generující základní typy signálů. Při použití Real Time Toolboxu můžeme použít jako vstupní data přímo měřené veličiny v reálném čase.

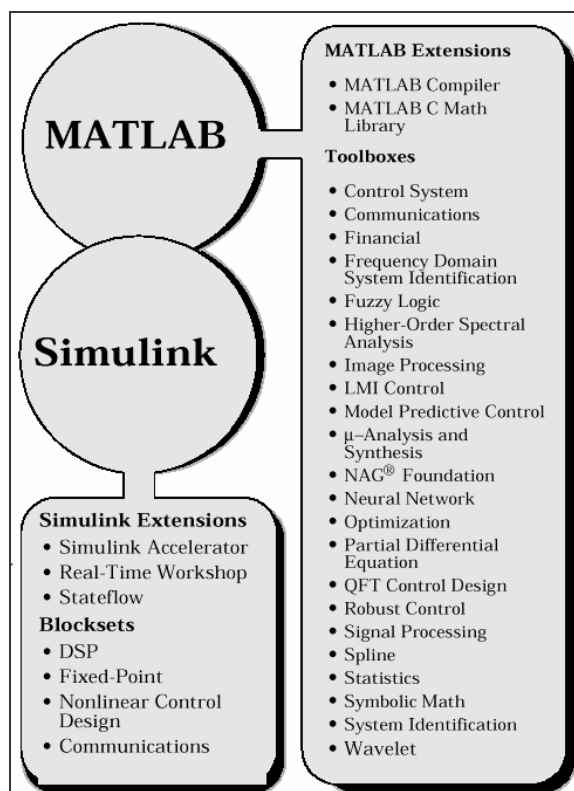
Výsledky simulace se nejčastěji zobrazují graficky – zobrazení typu osciloskop nebo XY graf. Další možností výstupu je přenesení proměnných do pracovního prostoru MATLABu nebo uložení do souboru. Základní okno je zobrazeno na obr. 1. [5]



Obr. 1. Okno Simulinku

1.1.6 Struktura MATLABu

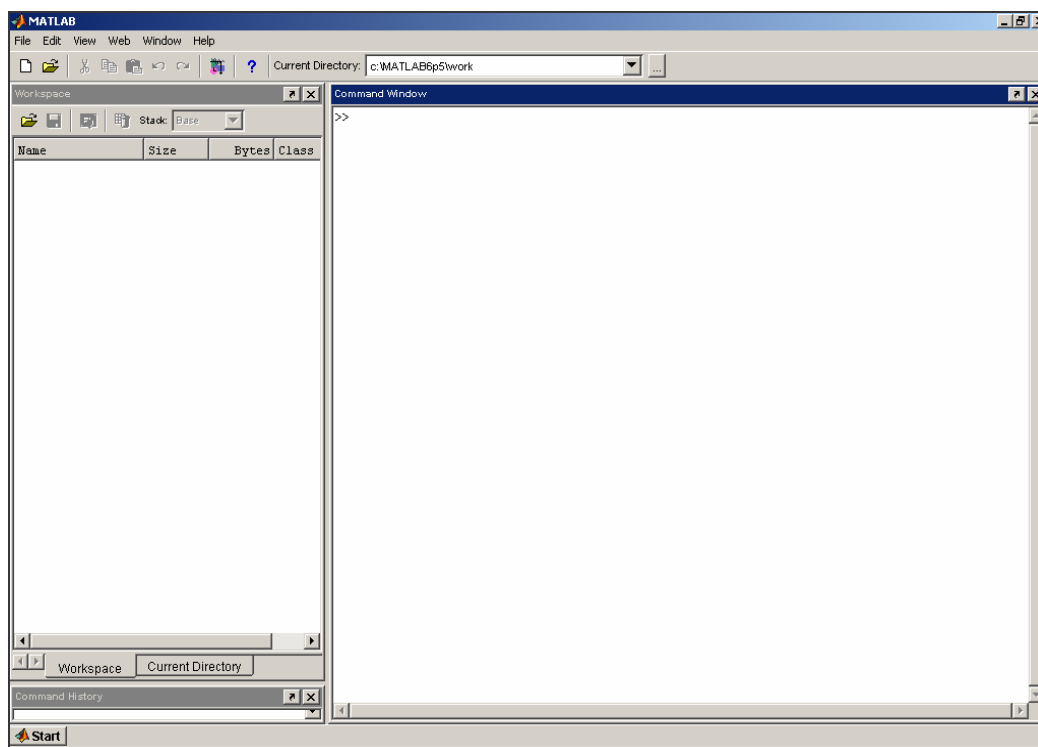
[1]



Obr. 2. Struktura MATLABu

1.1.7 Start MATLABu

MATLAB spouštíme dvojklikem na ikonu MATLAB na pracovní ploše. Po spuštění se nám zobrazí okno na Obr. 3. Do okna *Command Window* píšeme příkazy, v okně *Workspace* jsou zobrazeny všechny proměnné. V případě kliknutí na *Current Directory* se otevře okno, kde se zobrazí aktuální adresář s příslušnými podadresáři a soubory. [2]



Obr. 3. Hlavní okno MATLABu

1.1.8 Proměnné a funkce

Proměnné v MATLABu musí mít název začínajícím písmenem a ten může být až 31 znaků dlouhý. Rozlišují se velká a malá písmena. Můžeme použít libovolný název, ale některé názvy mají dopředu přiřazenou hodnotu (*pi*, *ans*, ...).

Proměnné jsou zjednodušeně řečeno dvojího typu – komplexní matice nebo řetězec znaků. Proměnné není třeba deklarovat – vytváří se automaticky v potřebné velikosti při prvním použití. Po vytvoření proměnná existuje do ukončení programu nebo do jejího zrušení příkazem *clear* a jejím názvem. Příkaz *clear* použitý bez názvu smaže všechny proměnné.

Při práci můžeme vytvořit velké množství proměnných navíc s různou velikostí. Příkaz *who* vypíše seznam existujících proměnných a příkaz *whos* vypíše také seznam existujících proměnných ale navíc i s jejich velikostmi.

Hodnoty, které proměnné zastupují, jsou v paměti počítače uloženy ve formátu pohyblivé řadové čárky. V případě, že je vypisujeme na obrazovku musí být konvertovány z tohoto formátu do řetězce číslic. Parametry této konverze můžeme nastavit příkazem *format*.

Hlavní síla MATLABu je ve funkcích. Funkce je obvykle složitější sada příkazů, která jeden nebo více vstupních parametrů (proměnných nebo konstant) zpracuje do jednoho nebo více výstupních parametrů (proměnných) podle určitého předpisu (algoritmu). Funkce mohou být jednoduché (např. sinus) nebo složité (vyhledávání minima funkce více proměnných). Funkce mohou být přímo součástí jádra MATLABu nebo ve formě předpisu, který je uložen v samostatném souboru na disku s příponou *.m*. Takovéto funkce se nazývají m-funkce. Součástí standardní instalace MATLABu je velké množství m-funkcí organizovaných v adresářích sdružujících m-funkce podle jejich zaměření. Seznam adresářů, kde jsou m-funkce uloženy se vytváří při instalaci. [2]

1.2 Real Time Toolbox

1.2.1 Úvod

Real Time Toolbox je nástavba programu Matlab, která slouží jako uživatelské prostředí pro měření a regulaci na různých typech zařízení v reálném čase. Umožňuje okamžitý přístup k vnějším analogovým a digitálním signálům téměř bez znalosti hardwarové problematiky. Je možno používat buď tradiční styl Matlabu — příkazovou řádku, nebo grafické uživatelské rozhraní, nebo výkonnou knihovnu systému Simulink. Lze experimentovat na vnějších signálech přímo z Matlabu, bez nutnosti přenášet data ze specializovaných programů pro sběr dat, bez nutnosti kompilace nebo jiných složitých a dlouho trvajících činností. To je ideální pro zpracování signálu, návrhy řídicích systémů a podobné úlohy. [4]

1.2.2 Vlastnosti

Real Time Toolbox vytváří z počítače měřicí ústřednu tím, že umožňuje práci s vnějšími signály v reálném čase. Umožňuje provádět sběr dat se zajištěným pravidelným vzorkováním a jejich další zpracování. Je založen na jádře reálného času a ovladačích přídatných karet pro vstup a výstup signálů. [4]

- nejnovější verze 3.12 běží pod Microsoft Windows 98 , Windows 2000 , Windows XP a MATLAB R13 / MATLAB R14
- 32 nezávislých časovačů, 256 vstupních a 256 výstupních kanálů
- vzorkovací frekvence až do 66 kHz bez nutnosti používat externí zdroje hodinového signálu
- knihovna bloků pro systém Simulink umožňuje používat většinu funkcí přímo z blokového schématu
- úrovně spouštěné ukládání dat umožňuje zachytit jednorázové děje
- zpracování dat on-line pomocí diskretních polynomiálních filtrů
- on-line řízení rychlých systémů vestavěnými lineárními regulátory
- podpora generování periodických signálů na analogových výstupních kanálech
- automatické vytváření skriptových M-souborů z graficky zadaných povelů
- možnost přístupu k více přídatným zařízením zároveň
- možnost změny parametrů ovladačů hardware za běhu a možnost změnit jejich implicitní nastavení
- asynchronní přístup přímo na vstupní a výstupní porty

1.2.3 Načtení hardwarových ovladačů

Pro vzájemnou komunikaci mezi jádrem reálného času a kartou zajišťující spojení počítače s měřeným objektem je nutné na začátku samotné práce provést načtení vstupního ovladače. Tím se zpřístupní karta pro další ovládání.

Příkaz provádějící výše popisovanou činnost je *rtload*. Po zadání v pracovním prostoru MATLABu se vytvoří okno *Select a hardware driver*. Zde se v příslušném adresáři na disku vyhledají ovladače hardwarových zařízení — myši a karty. Podle názvu karty provedeme výběr souboru shodného jména. Otevře se okno ovladače, kde je možné zapsat

hardwarovou adresu, zatrhnout možnosti digitálního vstupu, výstupu a DMA používaného při měření velmi rychlých signálů.

Ve spodní části okna se nacházejí tlačítka *OK*, *Reset*, *Save* a *Cancel*. *OK* potvrdí správnost zadaných údajů, *Save* uloží nové data na disk, *Reset* provede návrat k uloženým hodnotám a *Cancel* zruší načtený ovladač. Zpravidla pro obsluhu karty jsou už všechny údaje optimálně nadefinované. Ověřit načtený ovladač lze pomocí příkazu *rtwhc*. V MATLABu se potom zobrazí podrobnosti o kartě. Následné odstranění ovladače provede příkaz *rtunload*. [4]

1.2.4 Definování časovačů

Další nutnou podmínkou pro komunikaci s měřeným objektem je definování procesů časovačů. Časovače nejsou jen prostředky Real Time Toolboxu, ale tvoří základní aktivní prvky bez kterých by komunikace byla obtížná. Děje, které řídí časovače probíhají v synchronní podobě tj. v časových intervalech.

Příkaz definující časovače je *rtdef*. Vykonáním se objeví okno, kde je možné nastavit příslušné vlastnosti.

Nejdříve se definuje jeho číslo (*Select timer to be defined*), typ (*Select timer type*), číslo kanálu (*Timer channels*) a perioda vzorkování (*Timer period*). Číslo slouží k identifikaci časovače a vyznačuje jeho prioritu, typ zahrnuje několik nabídek. Na začátku si zadáme volbu *Input*. *Input* je vstupní proces, který čte data z měřeného objektu.

Je-li vstupní ovladač pomocí *rtload* načten, je možné doplnit další volbu v políčku *Timer channels*. Jedná se časovací kanál, který základním způsobem komunikuje s měřeným objektem. Vstupní kanál s označením *I* odpovídá vstupnímu časovacímu kanálu karty (*AD0*) a tak dále. To platí i pro výstupní kanály. Očíslování vstupních a výstupních kanálů karty je nezávislé.

Posledním krokem je nastavení periody vzorkování. Vzorkování určuje, jak často budou měřené objekty čteny a v jakých časových intervalech se na měřené objekty bude zapisovat. Volba periody je velmi důležitým parametrem. Závisí na vlastnostech regulované soustavy, ale standardně je nastavena na *0,1 s*. Jednotkami jsou sekundy.

Kliknutím na tlačítko *OK* je proces nadefinován.

Postup zopakujeme pro volbu *output*, která charakterizuje výstupní proces posílající data na měřený objekt. Jakmile toto máme nastaveno, můžeme ověřit skutečnost procesů zadáním příkazu *rtwho*. Pokud chcete odstranit časovače, proveďte příkaz *rtclear*. Tím budou vymazány všechny vymezené objekty a ponechán jen vstupní ovladač. [4]

1.2.5 Komunikace časovačů

Příkazy, které používáme jsou *rtrd* a *rtwr*. Jejich použitím se otevře okno prvního nalezeného časovače. V případě nutnosti upřesnit typ napíšeme *rtrd (2)*. Číslo 2 určuje časovač 2, číslo 1 časovač 1 atd. podle množství nadefinovaných časovačů.

Po nadefinování vstupního a výstupního časovače můžete otevřít dvě okna. Na první pohled jsou identická. Rozdíl je v barevném označení u vstupního typu, kde vstupní hodnota je žlutá na tmavě zeleném pozadí, kdežto hodnota výstupního typu může být ovládána jezdcem. Dále obsahují ovládací prvky časovacích vlastností, několik tlačítek a spustitelný ovládací rámeček.

Tlačítko *Run* spouští (stav start) a zastavuje (stav stop) příslušný časovač. Kliknutím na rámeček dojde k jeho zaškrtnutí, ale nejsou vidět žádné změny v časovacím okně. Podívat se na ně můžeme stiskem tlačítka *Animate*. Po stlačení uvidíme přibývajících počet ticků a hodnotu respektive výšku úrovně signálu.

Změna hodnoty signálu se provádí výstupním jezdcem, který se nastaví do vhodných polí. [4]

1.2.6 Spouštění a zastavování časovačů

Příkazy sloužící pro daný účel jsou *rtstart* a *rtstop*.

Okna mají podobný tvar s jediným rozdílem v příkazové řádce. Každé okno se skládá ze zaškrťávacích políček pro příslušný typ časovače. Nadefinované časovače příkazem *rtdef* jsou vysvíceny. Kliknutím na některé políčko se proces spustí nebo zastaví.

Volba *Auto Apply* má přímou vazbu na políčka. Zatrhnutím nebo odstraněním fajfky se okamžitě provede zvolená akce. Pokud není *Auto Apply* zapnuto, akce nejsou uvedeny do činnosti. Stav je však pamatován a vykonán až po stlačení tlačítka *Apply*. Tím se dají synchronně spouštět nebo zastavovat mnohonásobné procesy s několika časovači. Tlačítko *Revert* nuluje políčka a *All* spouští naráz všechny procesy. [4]

1.2.7 Příkazová řádka

Okna zjednodušují práci, jsou snadno ovladatelná, ale méně produktivní. Pro začátek však plně postačují. Ve spodní části každého okna se nachází příkazová řádka tzv. *Command line*, která poskytuje přehled o zápisu vybrané formy příkazu. Má za úkol seznámit uživatele i s jinou formou ovládání, kterou lze využít později u automatických úloh. [4]

1.3 Karta MC-101

1.3.1 Popis produktu

Karta MC-101 je určena pro snímání a řízení rychlostí, směrů a zrychlení u krokových motorů. Zároveň mohou být řízeny 3 motory při použití jedné karty. Každý ze 3 kanálů obsahuje časovač 8253 řízený běžným mikrořadičem 8031. Jakmile se odešle příkaz z počítače na kartu, mikrořadič 8031 jej převezme a stará se o řízení motoru, zatímco počítač je volný pro další operace. Mikrořadič má paměť na 10 příkazů.

U každého kanálu mohou být změněny parametry na modulaci šířky impulsů pro kontrolu stejnosměrných motorů, tepelných systémů atd.

Karta ještě obsahuje 8-mi bitový digitální vstupní a výstupní port pro použití v dalších aplikacích. [3]

1.3.2 Parametry karty

Umožňuje nezávislé operace u 3 kanálů s těmito parametry [3]:

- programovatelné kroky od 1 do 8191 impulsů za sekundu nebo 0,3 až 2700 otáček za sekundu u motoru s 200 kroky za otáčku
- programovatelné zrychlení a zpomalení od 100 do 25000 impulsů za sekundu
- počet kroků za jeden příkaz od -32768 do 32767
- nezávislé pohotovostní vstupy pro okamžité zastavení
- nezávislá volba krokování pro každý kanál
- měření času pomocí krystalu

1.4 CE 150 Helicopter

1.4.1 Laboratorní model

Celý laboratorní model je složen z těchto částí [6]:

- počítač s kartou MC 101
- spojovací modul
- helikoptéra

Model helikoptéry je zobrazen na obr. 4. a spojovací modul na obr. 5.



Obr. 4. Model helikoptéry



Obr. 5. Spojovací modul

1.4.2 Model helikoptéry

Laboratorní model helikoptéry je systém se dvěma měnitelnými vstupy a dvěma měřenými výstupy, který slouží ke studiu dynamických systémů a pro ověření teoretických modelů. Je určen pro číslicové řízení průmyslovým kontrolorem nebo IBM PC. Hardwarové a softwarové rozhraní podporují identifikaci a řízení v reálném čase pomocí programu MATLAB. [6]

1.4.3 Popis systému

Systém se sestává z těla, které nese dvě vrtule poháněné stejnosměrnými motory. Systém má dva stupně volnosti. Oba poziční úhly (horizontální a vertikální) jsou ovlivněny otáčkami vrtulí. Stejnosměrné motory jsou poháněné výkonovými zesilovači používající modulaci šířek impulsů. Oba úhly jsou měřitelné. Pro změnu těžiště je na helikoptěře umístěn servomotor, který posunuje malé závaží. Matematický model vrtulníku je MIMO 2x2 systém. [6]

1.4.4 Specifikace

Helikoptéra:

- hlavní vrtule – 180/100 přímo poháněná hlavním motorem s permanentními magnety na 12V, 0 až 5 A a 12000 otáček
- vedlejší vrtule – 150/100 přímo poháněná zadním motorem s permanentními magnety na 12V, 0 až 2 A a 7100 otáček
- motor na změnu těžiště – 5 V a 120mA

Spojovací modul:

- získávání úhlů pomocí přírůstkové pulzní logiky
- napájení střídavým napětím 240 V

1.5 Samočinně se nastavující regulátory

1.5.1 Úvod

Oblast adaptivního řízení procházela v uplynulých letech významným rozvojem. Cílem tohoto přístupu bylo řešit problematiku návrhu regulátorů v případech, kdy charakteristiky řízeného procesu nejsou dostatečně známy, nebo se v jeho průběhu mění. Postupem času vznikly různé směry řešení této problematiky. Jedním z úspěšných a perspektivních směrů jsou tzv. samočinně se nastavující regulátory (STC — Self-Tuning Controllers).

Základní myšlenka STC je založena na průběžné identifikaci vhodného modelu řízeného procesu s následnou syntézou regulátoru. [7]

1.5.2 Identifikace procesu

V adaptivním řízení je úloha identifikace stejně důležitá jako role syntézy regulátoru. Identifikace pro adaptivní řízení má ovšem svá specifika, která vedou k tomu, že se v převážné míře odhadují parametry regresního modelu a používá se metoda nejmenších čtverců. Zabýváme-li se identifikací určité soustavy, měli bychom postupovat podle následujícího schématu [7]:

1. Příprava identifikačního experimentu. Vybíráme nejvhodnější vstupní (budící) signál jako kompromis mezi teoreticky optimálním vybuzením a tím, co lze aplikovat z hledisek technologie. Průběh identifikačního experimentu lze sledovat, lze jej přerušit a upravit vstupní signál.
2. Data naměřená při experimentu lze uchovat a následně zpracovávat různými metodami a různými modely, filtrovat apod.
3. Získané parametry modelu lze verifikovat na jiných vzorcích dat.
4. Identifikační experiment lze opakovat, eventuálně již s využitím znalostí získaných předchozími experimenty.

5. Lze testovat či verifikovat podmínky pro nestrannost odhadů.

Naproti tomu, při průběžné identifikaci pro adaptivní řízení musíme vycházet z následujících podmínek:

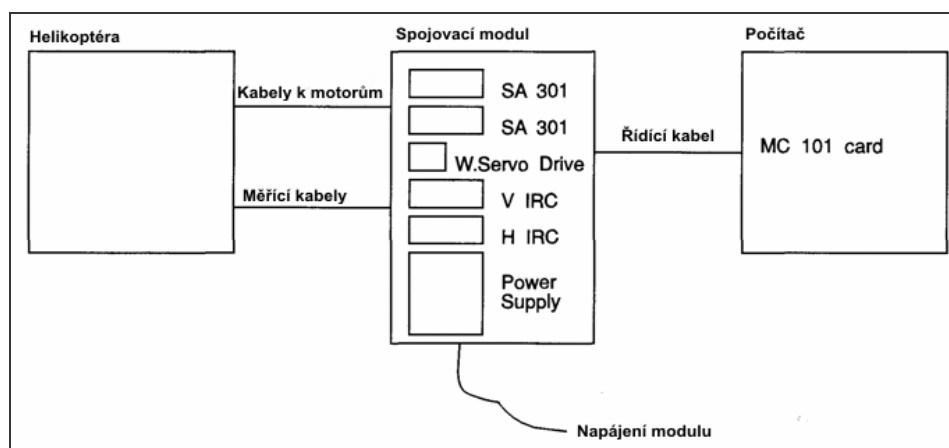
- 1.) Data (vstupy) jsou generovány zpětnovazebním regulátorem.
- 2.) Cílem regulátoru je kompenzovat poruchy, stabilizovat proces. To jsou okolnosti, které zhoršují možnosti identifikace parametrů.
- 3.) Identifikační proces u adaptivního řízení trvá velmi dlouho. Proto lze jen stěží předpokládat konstantnost odhadovaných parametrů. Metody odhadování časově proměnných parametrů jsou nezbytné.
- 4.) Identifikace musí být funkční za různých pracovních podmínek soustavy.
- 5.) Strukturu identifikovaného modelu obvykle nelze v průběhu pochodu měnit.
- 6.) Identifikační algoritmus musí být numericky spolehlivý a dostatečně rychlý.

2 EXPERIMENTÁLNÍ ČÁST

2.1 Popis připojení modelu k počítači

2.1.1 Popis připojení modelu k PC

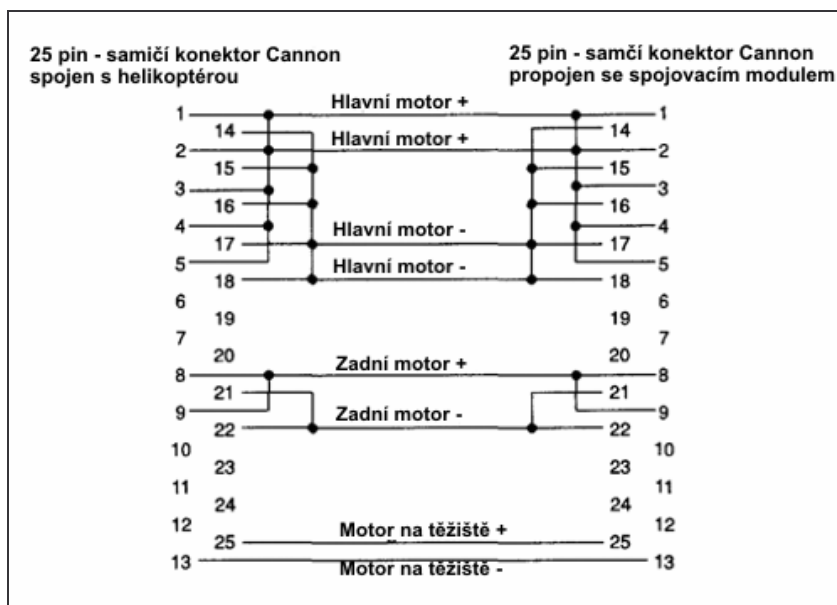
Na obr.6. je schématický popis zapojení. Helikoptéra je spojena dvěma kabely se spojovacím modulem. První kabel slouží k napájení motorů a druhý kabel je měřicí. Spojovací modul je spojen s počítačem prostřednictvím jednoho kabelu, který je zapojen do karty MC 101 Card.



Obr. 6. Popis zapojení modelu k PC

2.1.2 Kabel k motorům

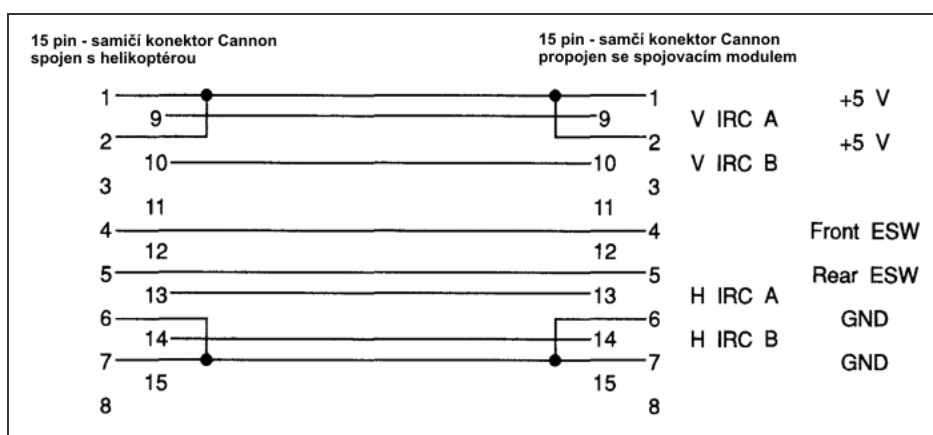
Na obr.7. je znázorněn popis zapojení napájecích kabelů k motorům. Do helikoptéry je zapojen 25 pin – samičí konektor Cannon a do spojovacího modulu 25 pin – samčí konektor Cannon. Hlavní motor helikoptéry je napájen 4 vodiči. Zadní motor a motor pro změnu těžiště jsou napájeny po 2 vodičích.



Obr. 7. Kabel k motorům

2.1.3 Měřicí kabel

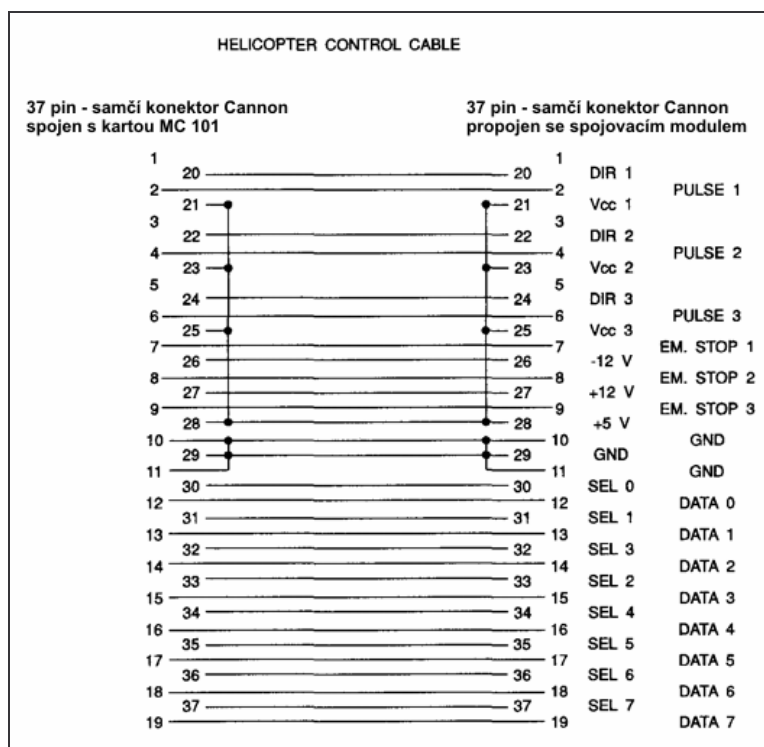
Na obr.8. je znázorněn měřicí kabel pro snímání otáček hlavní a zadní vrtule a také pro polohu těžiště. Do helikoptéry je připojen 15 pin – samičí konektor Canon a do spojovacího modulu 15 pin – samčí konektor Canon.



Obr. 8. Měřicí kabel

2.1.4 Řídící kabel

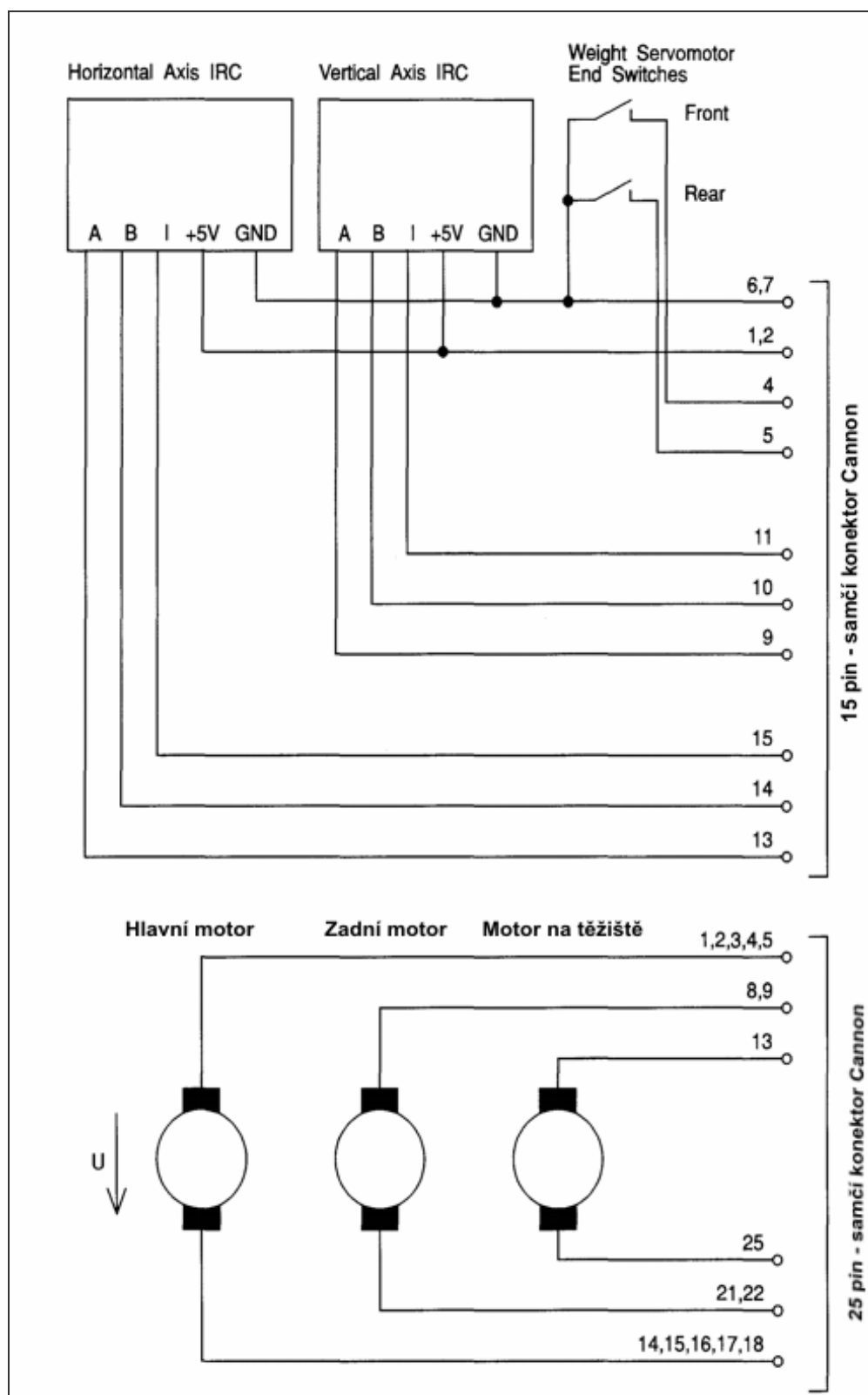
Obrázek 9. znázorňuje propojení karty MC101, která je umístěna v počítači se spojovacím modulem. Do karty je přiveden 37 pin – samčí konektor Cannon a spojovací modul je připojen 37 – pin samčím konektorem Cannon. Po tomto kabelu získáváme informace z helikoptéry (otáčky vrtulí, těžiště) a řídíme helikoptéru.



Obr. 9. Řídící kabel

2.1.5 Kompletní zapojení helikoptéry

Na obr. 10. je znázorněno kompletní zapojení helikoptéry a spojení se spojovacím modulem. Jak již bylo zmíněno výše do helikoptéry vedou 2 kabely. Jeden na ovládání motorů a druhý na získávání informací.

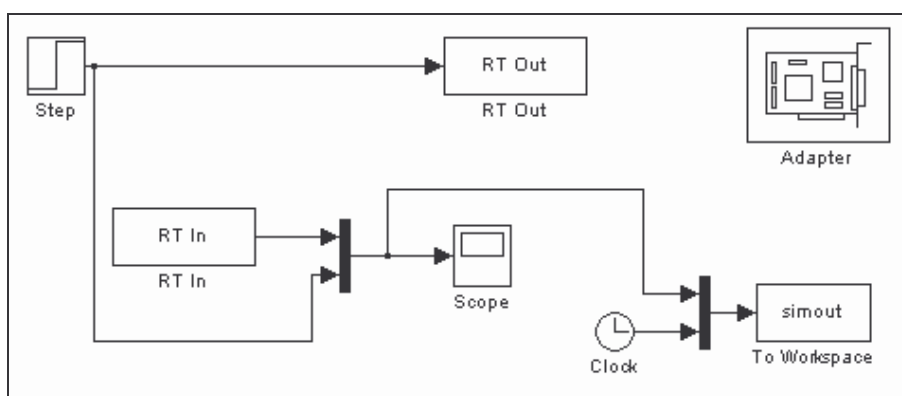


Obr. 10. Kompletní zapojení helikoptéry

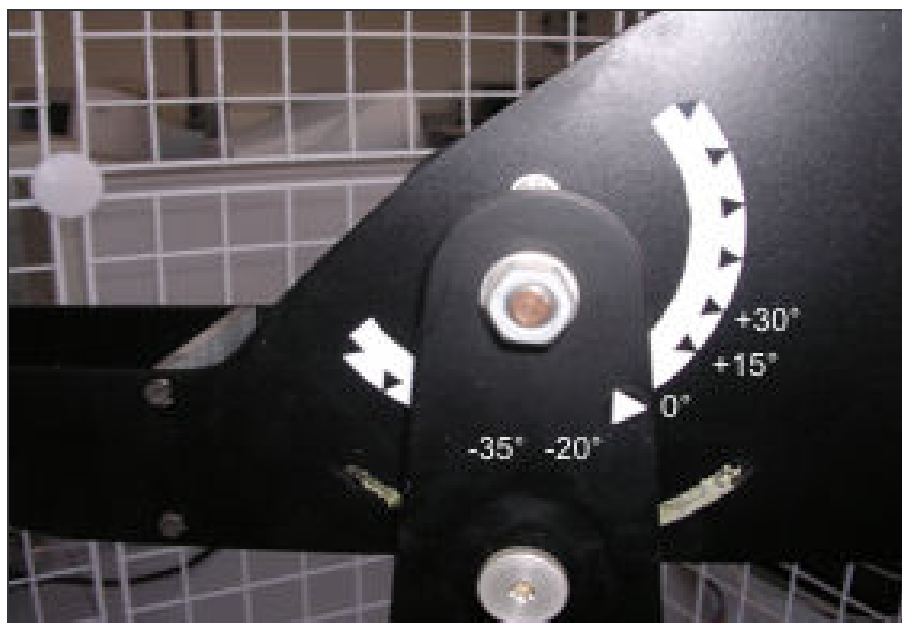
2.2 Identifikace modelu

2.2.1 Postup identifikace

K identifikaci jsem použil Simulinkové schéma na obr.11. Prováděl jsem identifikaci pro vertikální směr a poté pro horizontální směr. Pro identifikaci vertikálního směru, jsem si helikoptéru aretoval tak, aby se nemohla otáčet v horizontálním směru. Pro identifikaci horizontálního směru, jsem aretoval vertikální směr v 5-ti úhlech. Detail aretace vertikálního směru je na obr. 12.



Obr. 11. Simulinkové schéma pro identifikaci



Obr. 12. Detail nastavení náklonu helikoptéry

2.2.2 Provedení identifikace

- nastavení parametrů bloku *Adapter*:
RTLOAD ('MC101HEL' [opt , adress , DMARQ])
 opt = frekvence otáček (může se měnit)
 adress = adresa karty MC101 – musí být 784
 DMARQ = 0
 př. **RTLOAD ('MC101HEL' [10000 , 784 , 0])**
- nastavení parametrů bloku *RT OUT*: záleží na motoru, který ovládám. **Nastavení je podle tabulky II.**
- nastavení parametrů bloku *RT IN*: záleží na směru, který měřím. **Nastavení je podle tabulky I.**

Dále v bloku *STEP* se nastavuje skok žádané veličiny v čase. Poté se spustí měření tlačítkem *PLAY* na nástrojové liště. Po proběhnutí měření jsou výsledky uloženy v proměnné „*simout*“ v prostředí MATLAB. Průběh je možný vykreslit kliknutím na blok *SCOPE*.

Tabulka I. Výstupy z helikoptéry

Číslo kanálu		Akční zásah
1.	Hlavní motor	<-1 , 1>
2.	Zadní motor	<-1 , 1>
4.	Motor na ovládání těžiště	<-1 , 1>

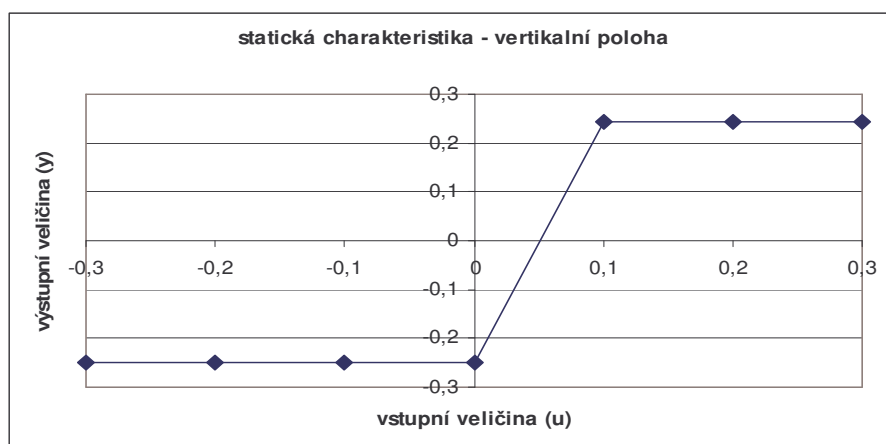
Tabulka II. Vstupy do helikoptéry

Číslo kanálu		Hodnota výstupu
1.	Horizontální pozice	<-1 , 1>
2.	Vertikální pozice	<-1 , 1>

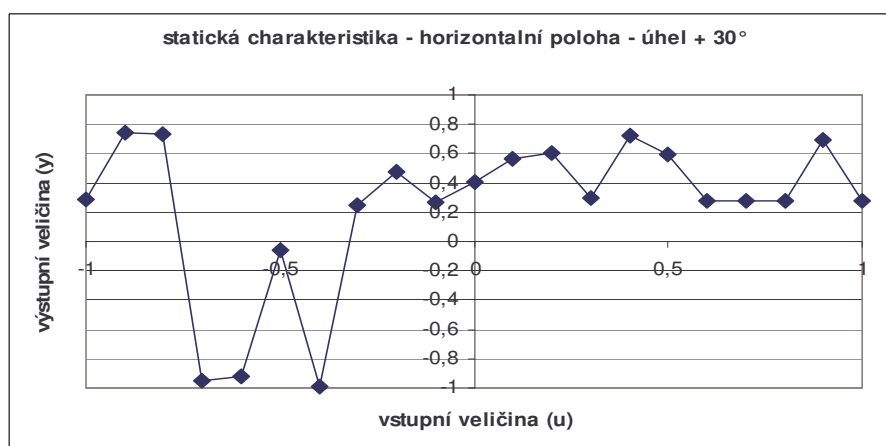
2.2.3 Statické charakteristiky

Vstupní veličiny jsem zadával v hodnotách <-1 , 1> po kroku 0,1 pro horizontální směr a hodnoty <-0.3 , 0.3> po kroku 0,1 pro vertikální směr. Hodnoty pro vertikální směr jsou dané fyzikálními parametry modelu.

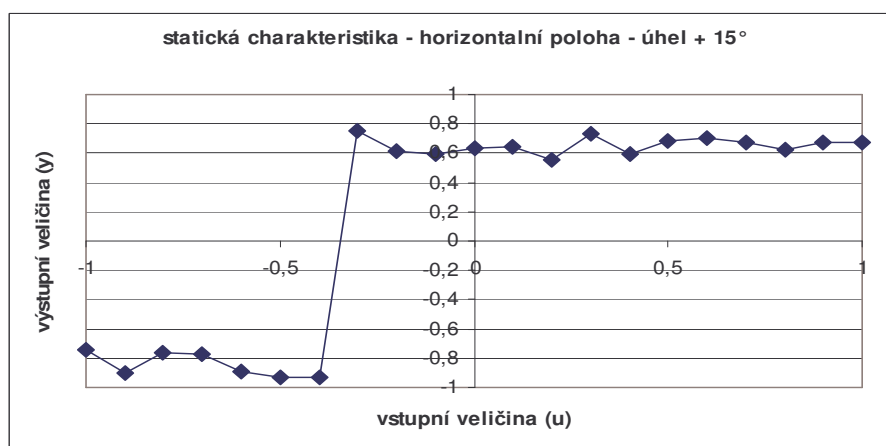
Na obrázku 13. je zobrazena statická charakteristika vertikální polohy. Na obrázcích 14 ÷ 18 jsou znázorněny statické charakteristiky horizontální polohy a to v úhlech : +30°, +15°, 0°, -20°, -35°.



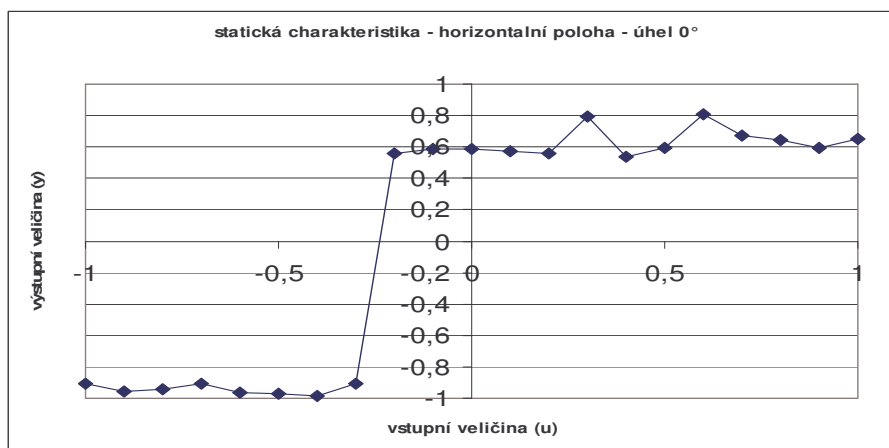
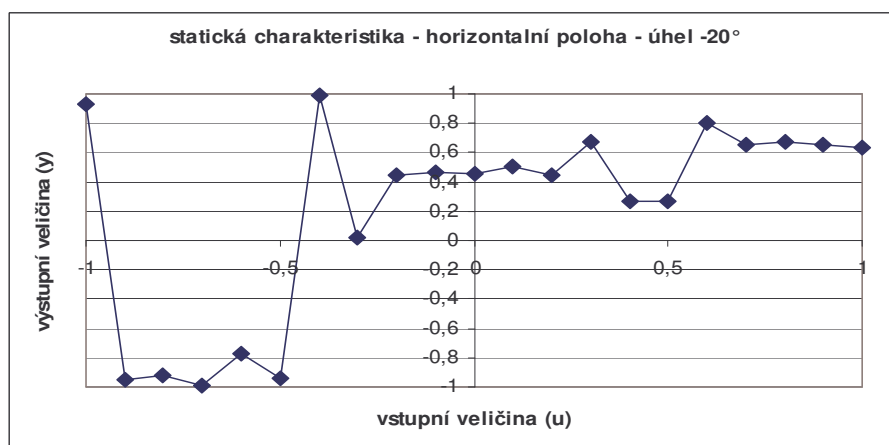
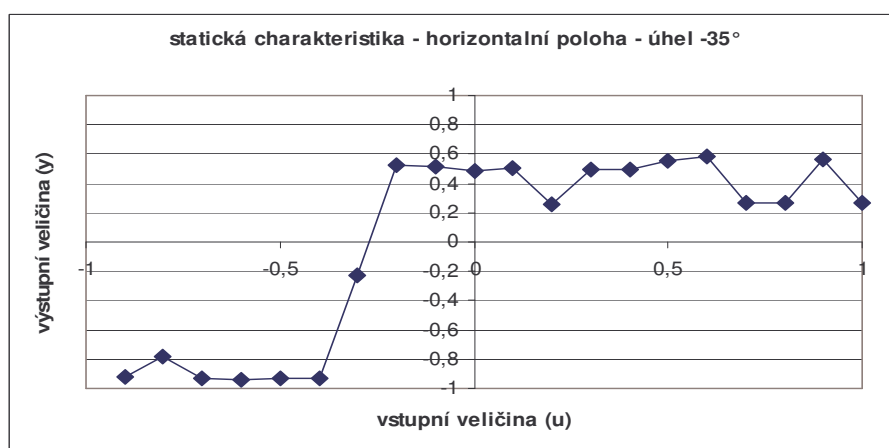
Obr. 13. Statická charakteristika – vertikální poloha



Obr. 14. Statická charakteristika – horizontální poloha – úhel + 30°



Obr. 15. Statická charakteristika – horizontální poloha – úhel + 15°

Obr. 16. Statická charakteristika – horizontální poloha – úhel 0° Obr. 17. Statická charakteristika – horizontální poloha – úhel -20° Obr. 18. Statická charakteristika – horizontální poloha – úhel -35°

2.3 Adaptivní řízení

Vzhledem k tomu, že statické charakteristiky jsou nelineární a hlavní a zadní vrtule nejsou nezávislé jsem se rozhodl použít adaptivní řízení k ovládání obou motorů najednou. Na Obr. 19. je znázorněno schéma adaptivního řízení podle [8]. Jsou zde 2 regulátory. Jeden pro hlavní motor a druhý pro zadní motor.

Při otevření bloku regulátoru zde nastavíme periodu vzorkování.

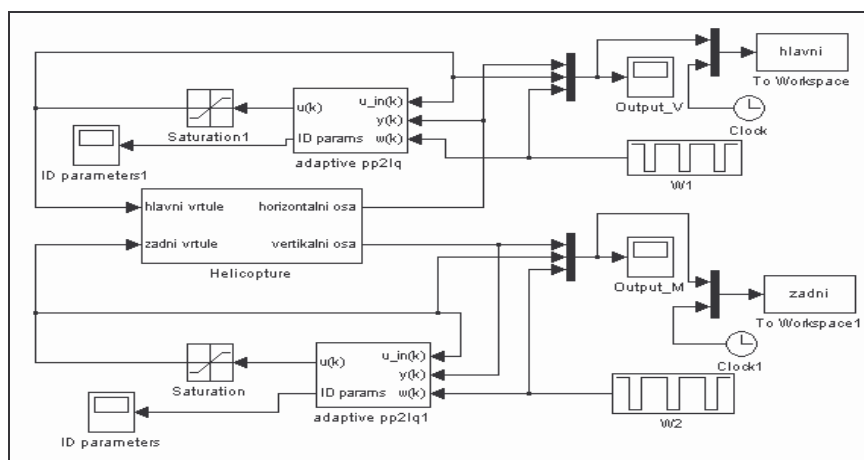
Žádané veličiny se zadávají do bloků:

- $W1$ – pro hlavní motor
- $W2$ – pro zadní motor

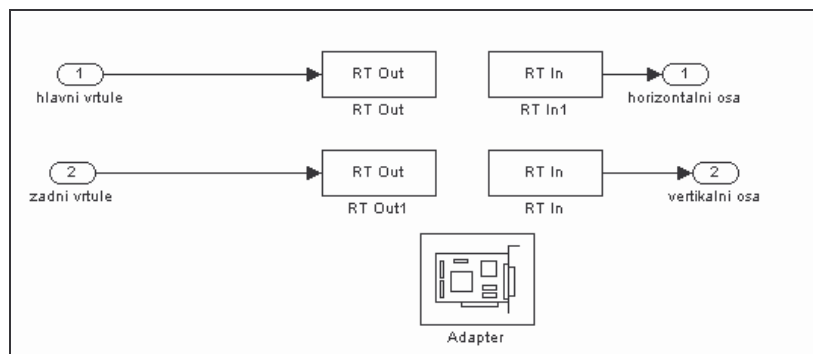
Do bloků *Saturation* a *Saturation1* se zadávají maximální akční zásahy podle tabulky I.

Vykreslení průběhů je možné otevřením bloků *Output_V* a *Output_M*. Naměřené hodnoty jsou v prostředí MATLAB pod názvem *hlavni* a *zadni*.

Na Obr. 20. je otevřen blok *Helicopter*. Zde se nastavují bloky *Adapter*, *RT In* a *RT Out* stejně jako v bodě 2.2.2, který je uveden výše.



Obr. 19. Schéma pro adaptivní řízení podle [8]



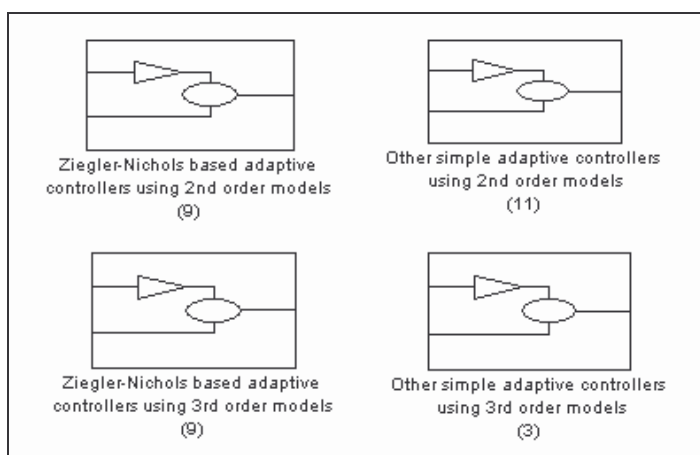
Obr. 20. Schéma bloku „Helicopter“ podle [8]

2.3.1 Adaptivní regulátory

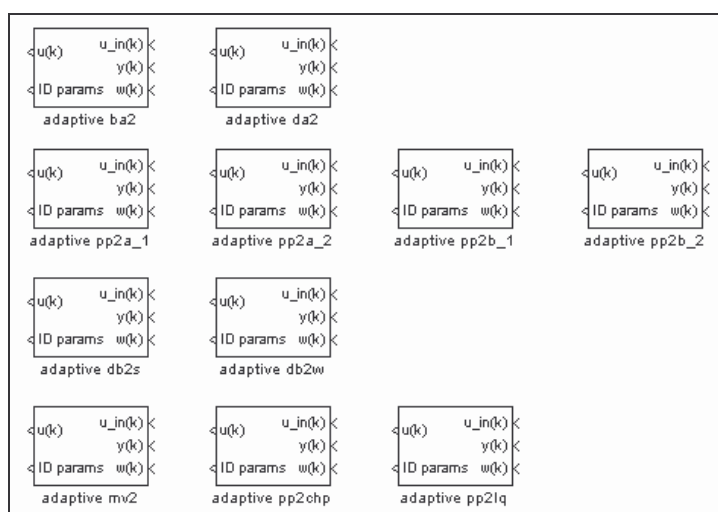
Pro výběr dalších regulátorů je třeba v prostředí MATLAB otevřít adresář STCTOOL a spustit soubor stcsl_std.mdl. STCTOOL [9] je dostupný na CD přiloženém k této práci nebo na internetových stránkách < <http://www.utb.cz/stctool>>. Poté se otevře nabídka na Obr. 21. Z této nabídky je možné zvolit regulátor používající model 2. nebo 3. řádu a dva typy regulátorů:

- regulátory používající modifikaci Ziegler – Nichols
- regulátory nepoužívající modifikaci Ziegler – Nichols

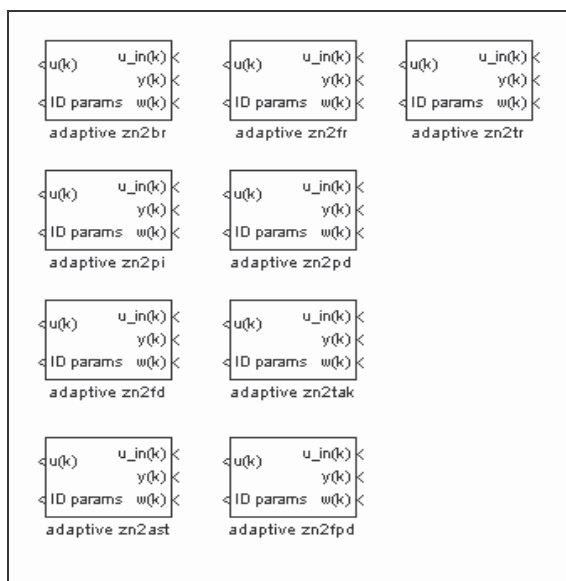
Na Obr. 22. jsou zobrazeny regulátory používající model 2. řádu s jinou modifikací než Ziegler – Nichols a na Obr. 23. jsou regulátory používající model 2. řádu s modifikací Ziegler – Nicholsovy metody. Zkratky jsou vysvětleny v příloze 3.



Obr. 21. Výběr regulátoru



Obr. 22. Adaptivní regulátory s jinou modifikací než modifikací Ziegler – Nicholsovy metody



Obr. 23. Adaptivní regulátory používající modifikací Ziegler – Nicholsovy metody

2.4 Řízení laboratorního modelu

Periodu vzorkování jsem zvolil 0,05s u obou regulátorů. Menší periodu vzorkování není možné zvolit kvůli omezeným parametrům karty MC101. Změna žádané veličiny musí být u obou motorů z minimální hodnoty na maximální a naopak. Při jiné hodnotě než mezní regulátory nejsou schopny motory řídit – velká perioda vzorkování. Žádaná hodnota pro vertikální pozici je $\langle -0.2, 0.2 \rangle$ a pro horizontální pozici $\langle -0.8, 0.8 \rangle$. Tyto hodnoty jsou dány fyzikálními parametry modelu (dorazy).

Pro regulaci jsem zvolil dva typy regulátorů:

1. pp2lq = regulátor založený na minimalizaci lineárně-kvadratického kritéria
2. zn2tak = modifikace Ziegler – Nichols používající Takahashi regulátor

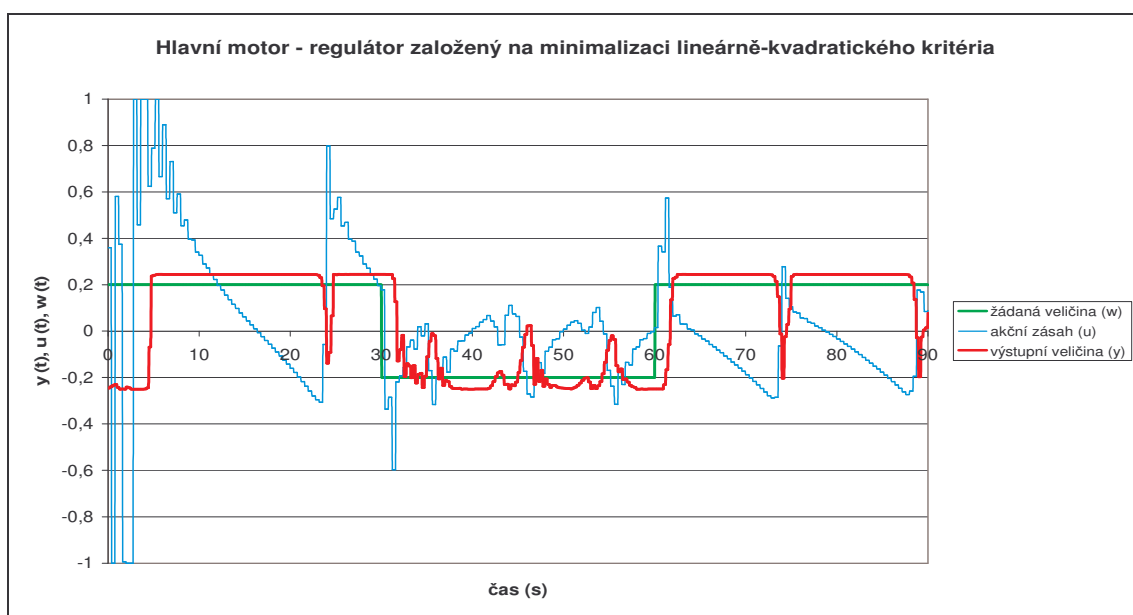
2.4.1 Regulátor založený na minimalizaci lineárně-kvadratického kritéria

Na Obr. 24 je zobrazena regulace hlavního motoru helikoptéry pomocí lineárně-kvadratického kritéria a na Obr. 25 regulace zadního motoru podle stejného kritéria.

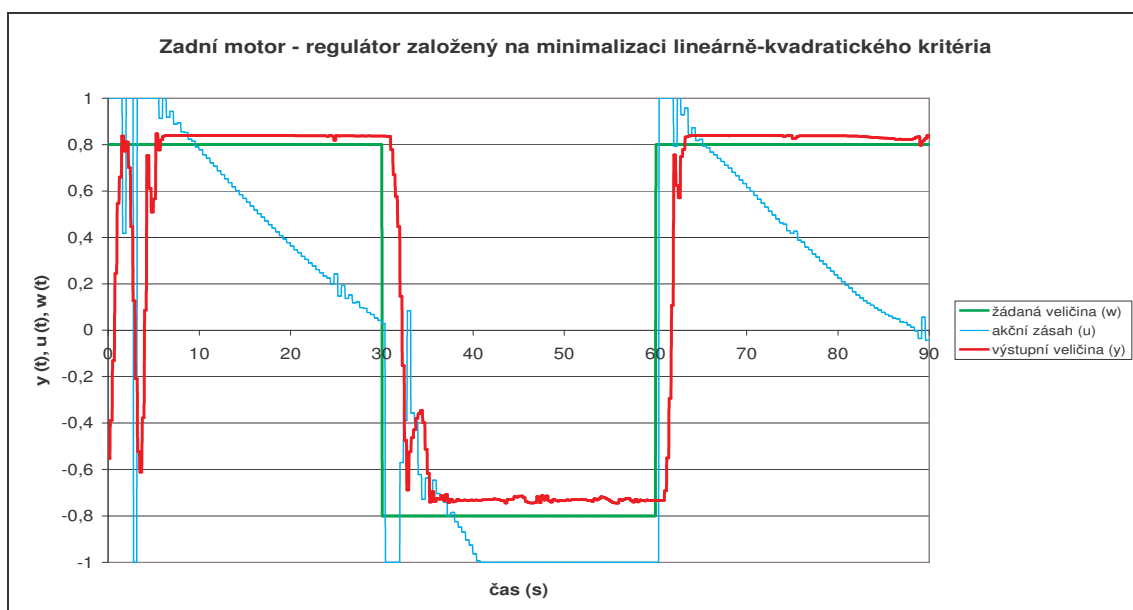
U obr. 24. regulátor řídí otáčky hlavní vrtule. Při žádané veličině 0,2 ji dosáhne s malým zpožděním, což je způsobeno počátečními podmínkami. Poté reguluje nad žádanou veličinu a neodstraní úplně regulační odchylku – velká perioda vzorkování.

Při žádané veličině $-0,2$ řídí otáčky velmi nekvalitně. Jsou zde velké překmity. Toto je způsobeno pravděpodobně provázaností obou motorů.

Na obr. 25. při žádané veličině $0,8$ se regulovaná veličina dostane na žádanou veličinu po jednom velkém podkmitu, což je způsobeno počátečními podmínkami. Při žádané veličině $-0,8$ následuje jeden překmit a poté se regulovaná veličina drží nad žádanou veličinou. Průběh lze upravit změnou konstant penalizace.



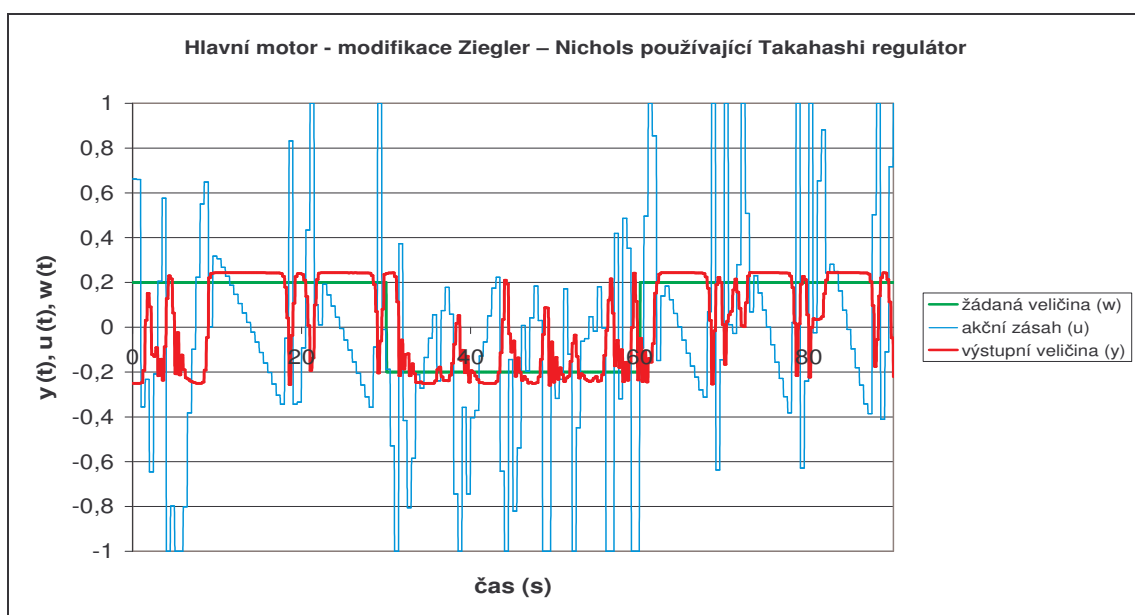
Obr. 24. Regulace hlavního motoru – minimalizace lineárně-kvadratického kritéria



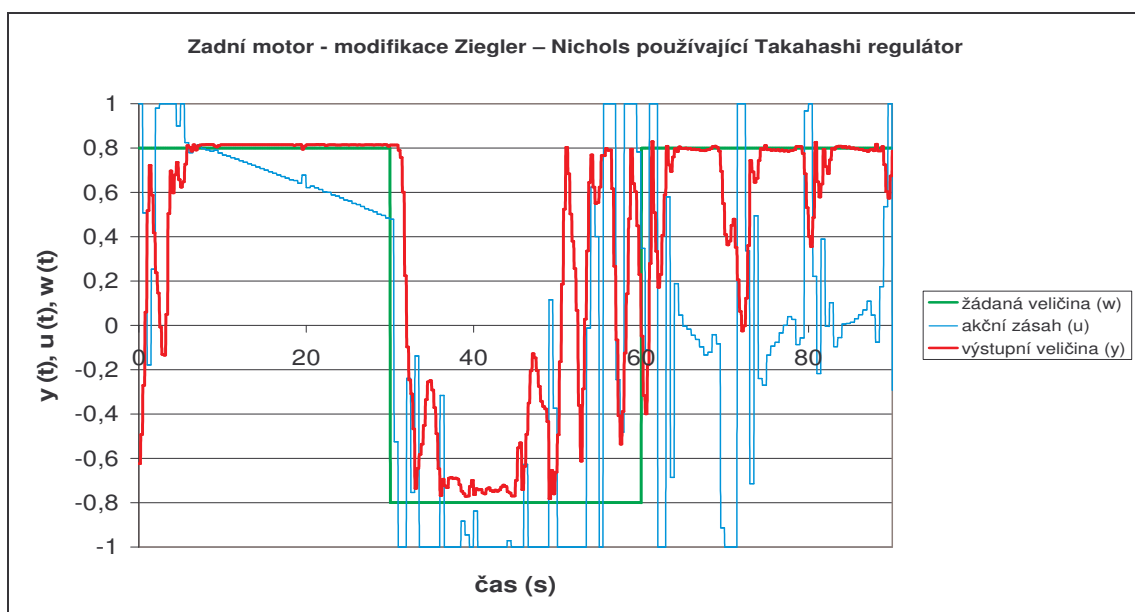
Obr. 25. Regulace zadního motoru – minimalizace lineárně-kvadratického kritéria

2.4.2 Modifikace Ziegler – Nichols používající Takahashi regulátor

Na Obr. 26 je uvedena regulace hlavního motoru pomocí Takahashi regulátoru. Ve srovnání s předchozí metodou se tento regulátor jeví jako méně vhodný kvůli velkým kmitům které jsou při obou žádaných veličinách. Na Obr. 27 je znázorněna regulace zadního motoru. Na žádanou veličinu 0,8 se dostane s jedním velkým podkmitem a poté se je regulovaná veličina téměř totožná s žádanou veličinou. Při skoku na žádanou veličinu - 0,8 se soustava rozkmitá. Při dalším skoku na žádanou veličinu 0,8 soustava stále kmitá. Nicméně snaha o náznak regulace je zřejmá.



Obr. 26. Regulace hlavního motoru – Takahashi regulátor



Obr. 27. Regulace zadního motoru – Takahashi regulátor

ZÁVĚR

Tato práce obsahuje v teoretické části přehled o programu Matlab, Simulink, Real Time Toolbox. Dále popis laboratorního modelu helikoptéry CE150 a popis adaptivních regulátorů.

V experimentální části se práce zabývá realizací připojení modelu k počítači. Helikoptéra je propojena se spojovacím modulem dvěma kabely. Jeden slouží k ovládání motorů a druhý pro měření. Spojovací modul je propojen jedním kabelem, který slouží jako řídicí kabel s počítačem, přesněji s kartou MC101 umístěnou v počítači.

Dále jsem prováděl identifikaci. Měření probíhalo jednou pro vertikální polohu a 5x pro horizontální polohu. To proto, že horizontální poloha jde aretovat v mnoha místech, ale výsledky jsou stejné. Vertikální polohu lze aretovat v 5 místech označených na helikoptěře a pokaždé vyjdou jiné výsledky. U horizontálního měření natočení jsem prováděl pro každý úhel náklonu 21 měření. U vertikálního měření stačilo provést 6 měření. Z těchto výsledků jsem vytvořil statickou charakteristiku. Z této charakteristiky je patrné, že systém je nelineární a bylo by velmi obtížné řídit obě vrtule najednou jednoduchým regulátorem. Navíc jsou vrtule spolu provázány. Proto jsem se rozhodl použít adaptivní regulátor pro řízení obou vrtulí zároveň.

Řízení tohoto laboratorního modelu je obtížné v tom, že karta MC101 umístěná v počítači není schopna použít menší periodu vzorkování než 0,05s, což je u takto rychlého systému velmi velká perioda. Proto jsem musel zadávat žádané veličiny do krajních poloh – po dorazy. U vertikálního řízení (hlavní motor) je žádaná veličina $< -0.2, 0.2 >$. U horizontálního řízení (zadní motor) je žádaná veličina $< -0.8, 0.8 >$. V mém měření jsem použil dva regulátory:

- založený na minimalizaci lineárně-kvadratického kritéria
- modifikace Ziegler – Nichols používající Takahashi regulátor

U prvního regulátoru je patrný náznak regulace. Průběh řízení zadního motoru je celkem uspokojivý. Při řízení hlavního motoru kmitá při žádané hodnotě -0,2. Druhý regulátor se jeví pro regulaci jako méně vhodný. Při regulaci zadního motoru se při skoku žádané veličiny na -0,8 regulátor rozkmital. U regulace hlavního motoru kmital téměř v průběhu celého měření. V dalším bodě mé práce jsem navrhl laboratorní úlohy k tomuto modelu. Ta bude spočívat v identifikaci systému pro horizontální a vertikální směr. Poté stanovení

statické charakteristiky. Dále student bude řídit helikoptéru pomocí adaptivních regulátorů. Vybere 5 regulátorů s modifikací Ziegler – Nichols a dalších 5 regulátorů nepoužívající modifikaci Ziegler – Nichols. Tyto průběhy vynese do grafu a porovná kvalitu regulace.

SEZNAM POUŽITÉ LITERATURY

- [1] MATLAB: Function reference, Version 7, MathWorks Inc., Natick, USA, 2004.
- [2] MATLAB: Getting Started With MATLAB, Version 7, MathWorks Inc., Natick, USA, 2004.
- [3] MC101 Motor Controller Card: User's Manual. Humusoft, s.r.o., Praha, 1992.
- [4] Real-Time Toolbox for MATLAB. User's Manual. Humusoft, s.r.o., Praha, 1999.
- [5] DUŠEK, F.: MATLAB a Simulink – úvod do používání, Univerzita Pardubice, 2000.
- [6] Humusoft: CE150 Helicopter – popis produktu [online]. [cit.2005-5-2]. Dostupný z URL: < <http://www.humusoft.cz/models/ce150cz.htm> >.
- [7] Bobál, V., Böhm, J., Prokop, R., Fessl, J.: Praktické aspekty samočinně se nastavujících regulátorů, VUT v Brně, Vutium 1999.
- [8] Chalupa, P.: Discrete Decentralized Control Systems. Disertační práce, Univerzita Tomáše Bati ve Zlíně 2003.
- [9] Univerzita Tomáše Bati Zlín: Simulinková knihovna adaptivních regulátorů [online]. [cit.2005-5-2] Dostupný z URL: < <http://www.utb.cz/stctool> >.

SEZNAM OBRÁZKŮ

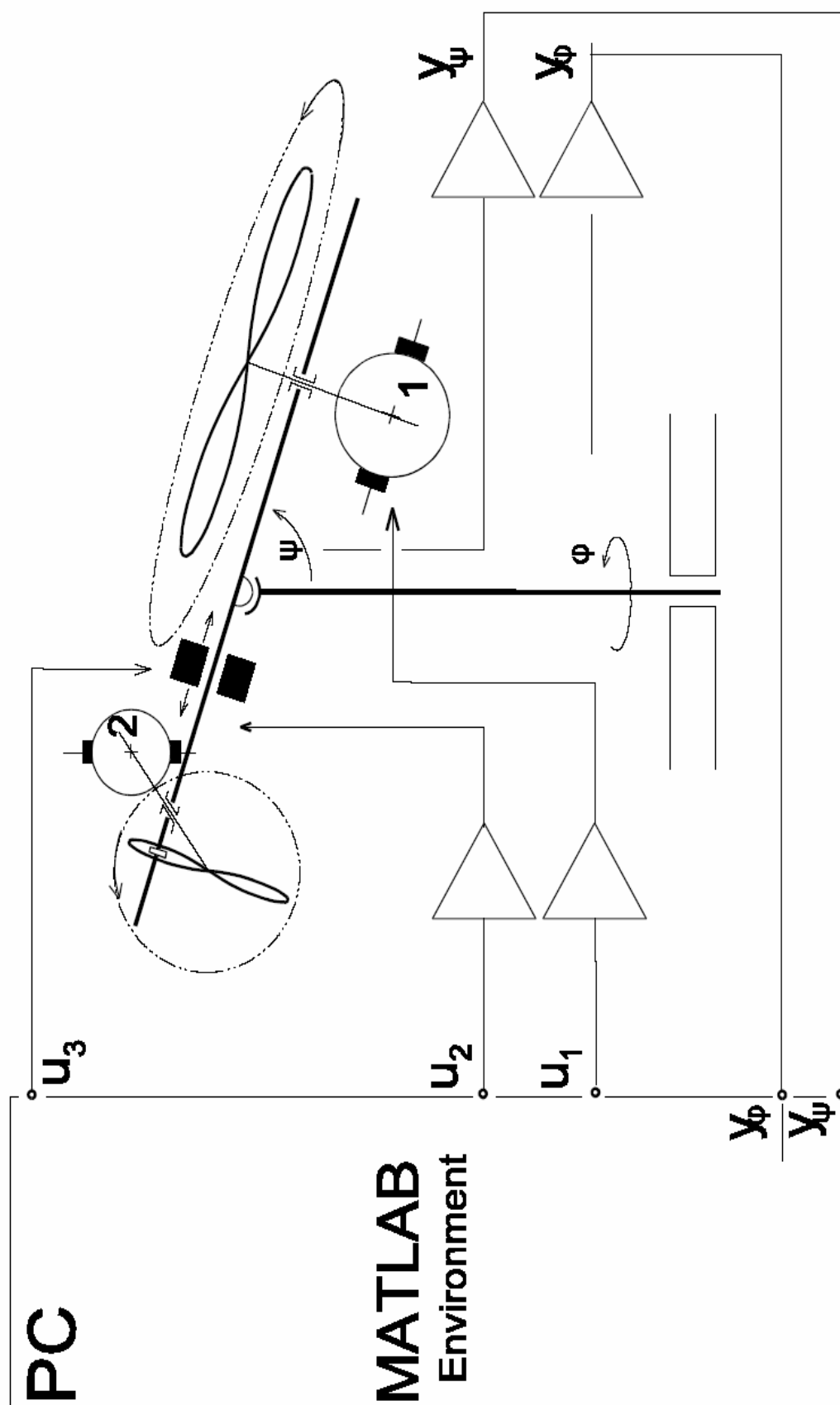
Obr. 1. Okno Simulinku.....	12
Obr. 2. Struktura MATLABu	12
Obr. 3. Hlavní okno MATLABu	13
Obr. 4. Model helikoptéry	19
Obr. 5. Spojovací modul	20
Obr. 6. Popis zapojení modelu k PC	23
Obr. 7. Kabel k motorům	24
Obr. 8. Měřicí kabel	24
Obr. 9. Řídící kabel.....	25
Obr. 10. Kompletní zapojení helikoptéry	26
Obr. 11. Simulinkové schéma pro identifikaci	27
Obr. 12. Detail nastavení náklonu helikoptéry	27
Obr. 13. Statická charakteristika – vertikální poloha	29
Obr. 14. Statická charakteristika – horizontální poloha – úhel + 30°	29
Obr. 15. Statická charakteristika – horizontální poloha – úhel + 15°	29
Obr. 16. Statická charakteristika – horizontální poloha – úhel 0°	30
Obr. 17. Statická charakteristika – horizontální poloha – úhel -20°	30
Obr. 18. Statická charakteristika – horizontální poloha – úhel -35°	30
Obr. 19. Schéma pro adaptivní řízení podle [8].....	31
Obr. 20. Schéma bloku „Helicopture“ podle [8]	31
Obr. 21. Výběr regulátoru.....	32
Obr. 22. Adaptivní regulátory s jinou modifikací.....	32
Obr. 23. Adaptivní regulátory používající.....	33
Obr. 24. Regulace hlavního motoru – minimalizace lineárně-kvadratického kritéria.....	34
Obr. 25. Regulace zadního motoru – minimalizace lineárně-kvadratického kritéria.....	34
Obr. 26. Regulace hlavního motoru – Takahashi regulátor.....	35
Obr. 27. Regulace zadního motoru – Takahashi regulátor	35

SEZNAM TABULEK

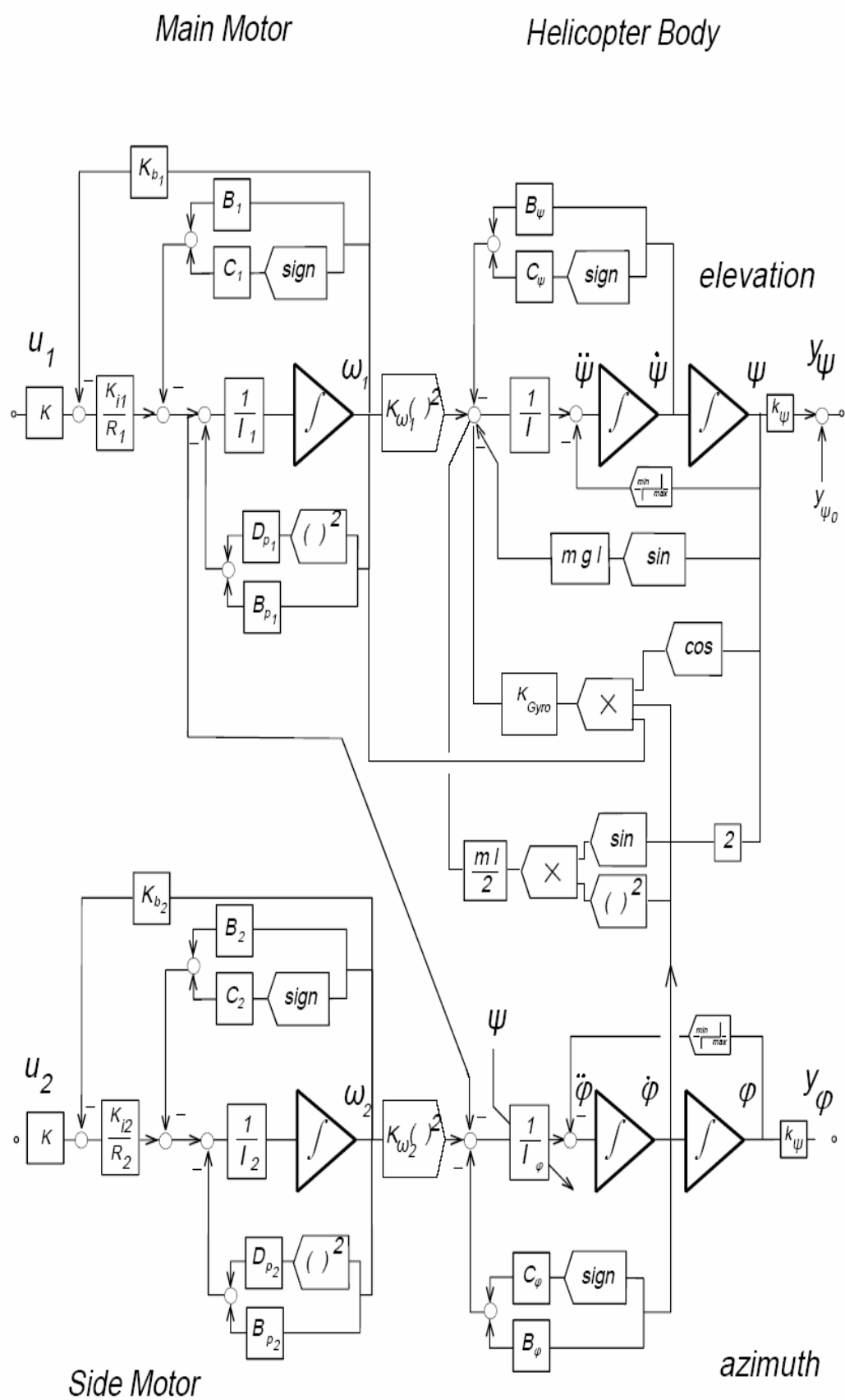
Tab. I	Výstupy z helikoptéry	27
Tab. II	Vstupy do helikoptéry	27

SEZNAM PŘÍLOH

- P 1 : Schématický diagram modelu helikoptéry
- P 2 : Teoretický model helikoptéry
- P 3 : Vysvětlení zkratk regulátorů
- P 4 : Obsah CD
- P 5 : Návrh laboratorní úlohy



P 2 : Teoretický model helikoptéry



P 3 : Vysvětlení zkratk regulátorů

Zkratka	Typ regulátoru
ba2	PID Bányász – Keviczky regulátor
da2	PID Dahlin regulátor
pp2a_1	PID regulátor metodou přiřazení pólů – modifikace A1
pp2a_2	PID regulátor metodou přiřazení pólů – modifikace A2
pp2b_1	PID regulátor metodou přiřazení pólů – modifikace B1
pp2b_2	PID regulátor metodou přiřazení pólů – modifikace B2
db2s	Dead-beat regulátor – silná verze
db2w	Dead-beat regulátor – slabá verze
mv2	Metoda minimálního rozptylu
pp2chp	Zobecněná metoda přiřazení pólů
pp2lq	Minimalizace lineárně - kvadratického kritéria
zn2br	PID používající zpětnou obdelníkovou metodu – modifikace ZN
zn2fr	PID používající dopřednou obdelníkovou metodu – modifikace ZN
zn2tr	PID používající lichoběžníkovou – modifikace ZN
zn2pi	PI regulátor – modifikace ZN
zn2pd	PD regulátor – modifikace ZN
zn2fd	PID regulátor používající filtraci derivační složky Tustinovou aproximací
zn2tak	PID Takahashi regulátor
zn2ast	PID Åström regulátor
zn2fpd	PID regulátor používající zpětnou obdelníkovou metodu s náhradou derivace čtyřbodovou diferencí

P 4 : Obsah CD

- bakalářská_práce obsahuje práci ve formátu .doc a .pdf

- ovladač_ke_kartě_MC101 obsahuje dva soubory s ovladači pro kartu MC101
- simulinkové_schéματα obsahují schémata pro identifikaci a adaptivní řízení
- statické_charakteristiky – tento adresář je rozdělen na statické charakteristiky podle polohy identifikace a úhlu naklonění. Jsou zde všechny identifikace
- stctool obsahuje adaptivní regulátory

P 5 : Návrh laboratorní úlohy

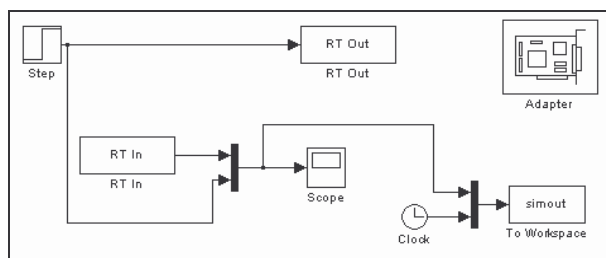
Na laboratorním modelu helikoptéry CE150 proveďte :

- a. Identifikaci

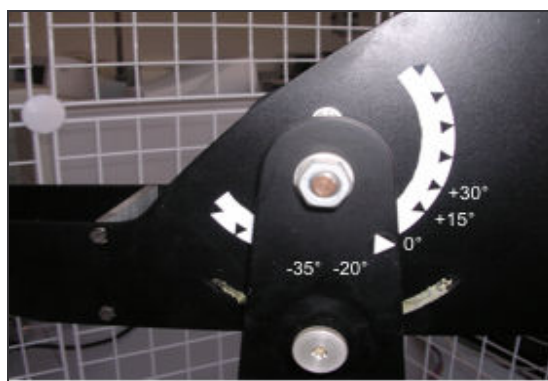
b. Vyberte nejvhodnější adaptivní regulátor.

Ad a.

Identifikaci proveďte podle Simulinkového schématu na Obr. 1 tak, že si zaaretujete jeden směr (horizontální, vertikální) a budete zadávat vstupní veličinu (u) v hodnotách $< -1, 1 >$ po kroku 0,1. Poté proveďte měření pro druhý směr. Při aretaci vertikální polohy si vyberte jeden úhel podle Obr. 2. Z výsledků zpracujte statickou charakteristiku.



Obr. 1. Simulinkové schéma pro identifikaci



Obr. 2. Aretace vertikální polohy

- nastavení parametrů bloku *Adapter*:
RTLOAD ('MC101HEL' [opt , adress , DMARQ])
 opt = frekvence otáček (může se měnit)
 adress = adresa karty MC101 – musí být 784
 DMARQ = 0
 př. **RTLOAD ('MC101HEL' [10000 , 784 , 0])**
- nastavení parametrů bloku *RT OUT*: záleží na motoru, který ovládám. **Nastavení je podle tabulky II.**
- nastavení parametrů bloku *RT IN*: záleží na směru, který měřím. **Nastavení je podle tabulky I.**

Dále v bloku *STEP* se nastavuje skok žádané veličiny v čase. Poté se spustí měření tlačítkem *PLAY* na nástrojové liště. Po proběhnutí měření jsou výsledky uloženy v proměnné „simout“ v prostředí MATLAB. Průběh je možný vykreslit kliknutím na blok *SCOPE*.

Tabulka III. Výstupy z helikoptéry

Číslo kanálu		Akční zásah
1.	Hlavní motor	$< -1, 1 >$
2.	Zadní motor	$< -1, 1 >$
4.	Motor na ovládání těžiště	$< -1, 1 >$

Tabulka IV. Vstupy do helikoptéry

Číslo kanálu		Hodnota výstupu
1.	Horizontální pozice	<-1 , 1>
2.	Vertikální pozice	<-1 , 1>

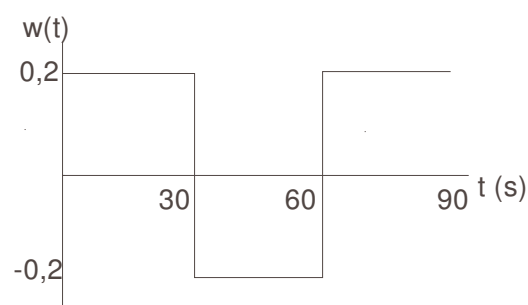
Ad b.

Pro řízení adaptivním regulátorem použijte Simulinkové schéma na Obr. 5.

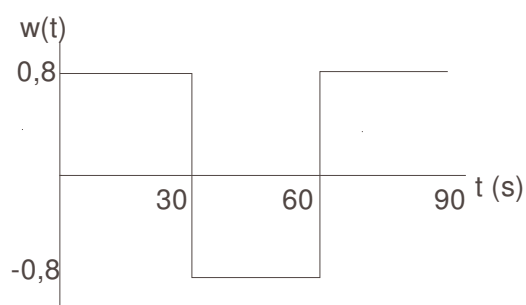
Při otevření bloku regulátoru se zde nastavuje perioda vzorkování – tu vhodně zvolte

Žádané veličiny se zadávají podle obrázků 3 pro W1 a 4 pro W2 do bloků:

- W1 – pro hlavní motor
- W2 – pro zadní motor



Obr. 3. Žádaná veličina hlavního motoru

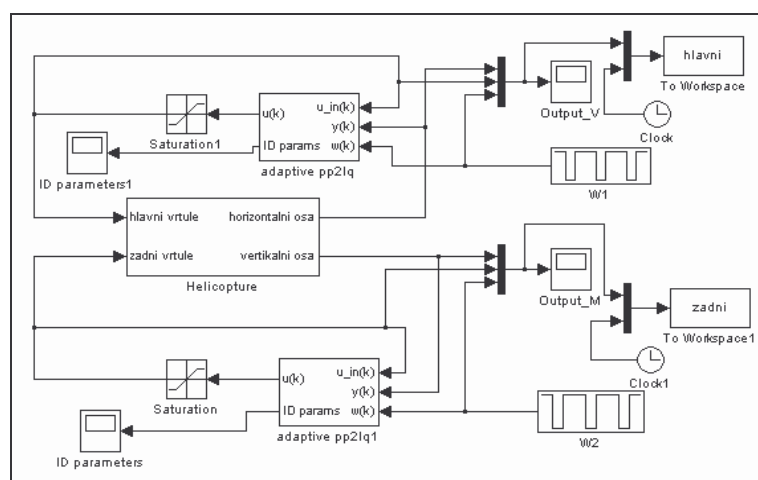


Obr. 4. Žádaná veličina zadního motoru

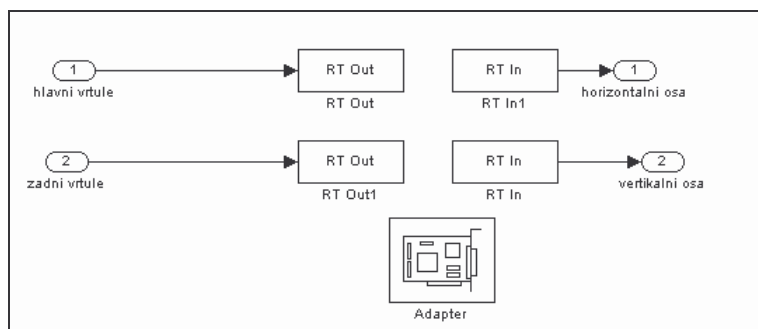
Do bloků *Saturation* a *Saturation1* se zadávají maximální akční zásahy podle tabulky I.

Vykreslení průběhů je možné otevřením bloků *Output_V* a *Output_M*. Naměřené hodnoty jsou v prostředí MATLAB pod názvem *hlavni* a *zadni*.

Na Obr. 6. je otevřen blok *Helicopter*. Zde se nastavují bloky *Adapter*, *RT In* a *RT Out* stejně jako v bodě a.



Obr. 5. Schéma pro adaptivní řízení

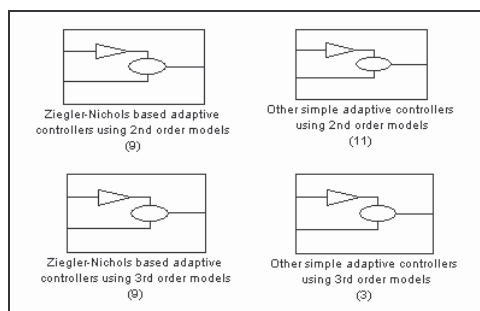


Obr. 6. Schéma bloku „Helicopture“

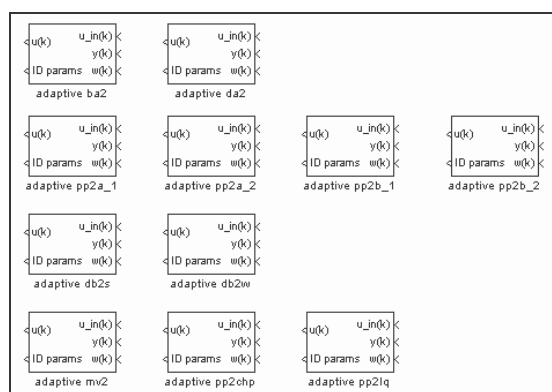
Pro výběr dalších regulátorů je třeba v prostředí MATLAB otevřít adresář STCTOOL a spustit soubor stcs1_std.mdl. STCTOOL je dostupný na internetových stránkách < <http://www.utb.cz/stctool> >. Poté se otevře nabídka na Obr. 7. Z této nabídky je možné zvolit buď 2 nebo 3 rozměrový model a dva typy regulátorů:

- regulátory používající modifikaci Ziegler – Nichols
- regulátory nepoužívající modifikaci Ziegler – Nichols

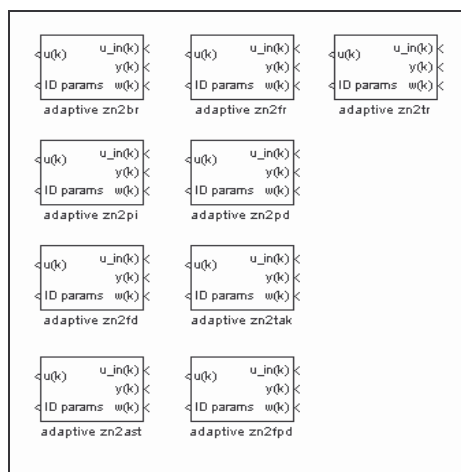
Na Obr. 8. jsou zobrazeny regulátory pro 2 rozměrový model s jinou modifikací než Ziegler – Nichols a na Obr. 9. jsou regulátory pro 2 rozměrový model používající modifikace Ziegler – Nichols.



Obr. 7. Výběr regulátoru



Obr. 8. Adaptivní regulátory s jinou modifikací než Ziegler - Nichols



Obr. 9. Adaptivní regulátory používající modifikace Ziegler – Nichols

Zkratka	Typ regulátoru
ba2	PID Bányász – Keviczky regulátor
da2	PID Dahlin regulátor
pp2a_1	PID regulátor metodou přiřazení pólů – modifikace A1
pp2a_2	PID regulátor metodou přiřazení pólů – modifikace A2
pp2b_1	PID regulátor metodou přiřazení pólů – modifikace B1
pp2b_2	PID regulátor metodou přiřazení pólů – modifikace B2
db2s	Dead-beat regulátor – silná verze
db2w	Dead-beat regulátor – slabá verze
mv2	Metoda minimálního rozptylu
pp2chp	Zobecněná metoda přiřazení pólů
pp2lq	Minimalizace lineárně - kvadratického kritéria
zn2br	PID používající zpětnou obdelníkovou metodu – modifikace ZN
zn2fr	PID používající dopřednou obdelníkovou metodu – modifikace ZN
zn2tr	PID používající lichoběžníkovou – modifikace ZN
zn2pi	PI regulátor – modifikace ZN
zn2pd	PD regulátor – modifikace ZN
zn2fd	PID regulátor používající filtraci derivační složky Tustinovou
zn2tak	PID Takahashi regulátor
zn2ast	PID Åström regulátor
zn2fpd	PID regulátor používající zpětnou obdelníkovou metodu s náhradou derivace

Vyberte si 5 regulátorů používající modifikace Ziegler – Nichols a 5 regulátorů s jinou modifikací než Ziegler – Nichols. Proveďte měření a výsledky všech regulátorů porovnejte. Vyneste do grafů statické charakteristiky a průběhy regulací.