

# Prediktivní řízení modelu vrtulníku na NI CompactRio

## Diplomová práce

*Studijní program:*

N2612 Elektrotechnika a informatika

*Studijní obor:*

Automatické řízení a inženýrská informatika

*Autor práce:*

**Bc. Dominik Paulů**

*Vedoucí práce:*

Ing. Lukáš Hubka, Ph.D.

Ústav mechatroniky a technické informatiky





## Zadání diplomové práce

# Prediktivní řízení modelu vrtulníku na NI CompactRio

*Jméno a příjmení:* Bc. Dominik Paulů  
*Osobní číslo:* M18000158  
*Studijní program:* N2612 Elektrotechnika a informatika  
*Studijní obor:* Automatické řízení a inženýrská informatika  
*Zadávací katedra:* Ústav mechatroniky a technické informatiky  
*Akademický rok:* 2020/2021

### Zásady pro vypracování:

1. Sestavte matematický popis a simulační model vrtulníku. Provedte validaci simulačního modelu, který budete využívat pro návrh prediktivního regulátoru.
2. Navrhněte standardní prediktivní regulátor založený na modelu pro řízení vrtulníku CE150. Regulátor umožní ovládní azimutu i elevace. Implementujte regulátor na CompactRio a ověřte funkčnost.
3. Zkoumejte a popište chování regulátoru. V případě nedostatečných vlastností základní varianty regulátoru zkoumejte možnosti zlepšení jeho vlastností.
4. Navrhněte explicitní verzi prediktivního regulátoru pro řízení vrtulníku CE150, implementujte jej na CompactRio a ověřte funkčnost.
5. Diskutujte vlastnosti a použitelnost obou typů regulátoru.

*Rozsah grafických prací:*  
*Rozsah pracovní zprávy:*  
*Forma zpracování práce:*  
*Jazyk práce:*

dle potřeby dokumentace  
40–50  
tištěná/elektronická  
Čeština



### **Seznam odborné literatury:**

- [1] TRAVIS, Jeffrey a Jim KRING. LabVIEW for everyone: graphical programming made easy and fun. 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2007. ISBN 0131856723.
- [2] National Instruments: Building a Real-Time System With NI Hardware and Software [online]. 2019 [cit. 2019-02-21]. Dostupné z: <http://www.ni.com/white-paper/4040/en/>.
- [3] KARER, G., ZUPANČIČ, B., Modelling and identification of a laboratory helicopter, Proc. 5th Mathmod conferency, 2006.
- [4] MACIEJOWSKI, Jan Marian. Predictive control: with constraints. New York: Prentice Hall, 2002. ISBN 978-0201398236.
- [5] ALESSIO A. and A. BEMPORAD. A Survey on Explicit Model Predictive Control. In: Magni L., Raimondo D.M., Allgöwer F. (eds) Nonlinear Model Predictive Control. Lecture Notes in Control and Information Sciences, vol 384. Springer, Berlin, Heidelberg. 2009. ISBN 978-3-642-01093-4.

*Vedoucí práce:*

Ing. Lukáš Hubka, Ph.D.  
Ústav mechatroniky a technické informatiky

*Datum zadání práce:*

9. října 2020

*Předpokládaný termín odevzdání:*

17. května 2021

prof. Ing. Zdeněk Plíva, Ph.D.  
děkan

L.S.

doc. Ing. Milan Kolář, CSc.  
vedoucí ústavu

## Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědom toho, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má diplomová práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

16. května 2021

Bc. Dominik Paulů

# Prediktivní řízení modelu vrtulníku na NI CompactRio

## Abstrakt

Cílem této diplomové práce je návrh implicitního a explicitního prediktivního regulátoru pro model vrtulníku. Jádrem systému je v kombinaci s LabVIEW programovatelný kontrolér CompactRIO od společnosti National Instruments. V první části práce jsou představeny charakteristické vlastnosti modelu vrtulníku a jeho limitace.

Následuje kapitola zabývající se popisem kontroléru CompactRIO a návrhem struktury řídicí aplikace. Dále je uveden teoretický rozbor stavové prediktivní a explicitní regulace, z kterého byly implementovány MPC regulátory pro SISO i MIMO systémy. Následují praktické realizace těchto regulátorů.

## Klíčová slova:

National Instruments, LabVIEW, CompactRIO, řízení procesu, Real-Time, prediktivní řízení, vícerozměrové systémy

# Predictive Control of the Helicopter Model Realized on the NI CompactRio

## Abstract

The main goal of this diploma thesis is to design an implicit and explicit form of model predictive controller for a helicopter model. The core of the system is CompactRIO controller, which is programmable using LabVIEW. The first part of the work presents the characteristic properties of the helicopter.

The following chapter describes the CompactRIO controller and design of the application structure. The next topic deals with theoretical analysis of state predictive and explicit control from which were MPC controllers implemented. The practical implementations of these controllers for helicopter model follow.

## Key words:

National Instruments, LabVIEW, CompactRIO, process control, Real-Time, predictive control, multidimensional control systems

## Poděkování

Rád bych poděkoval Ing. Lukášovi Hubkovi, Ph.D. za cenné rady, věcné připomínky a vstřícnost při konzultacích. Dále bych poděkoval své rodině a blízkým za podporu během studia.

# Obsah

Seznam zkratk . . . . .	10
Seznam symbolů . . . . .	11
Seznam obrázků . . . . .	13
<b>Úvod</b>	<b>16</b>
<b>1 Model vrtulníku CE 150</b>	<b>18</b>
<b>2 CompactRIO</b>	<b>20</b>
2.1 FPGA . . . . .	21
2.2 I/O moduly . . . . .	22
2.3 FGPA Interface mód . . . . .	22
2.4 CRIO SCAN mód . . . . .	23
2.5 Hybridní mód . . . . .	24
2.6 LabVIEW RT . . . . .	24
2.7 Návrh struktury řídicí aplikace pro CompactRIO . . . . .	26
<b>3 Teorie identifikace dynamických systémů</b>	<b>30</b>
3.1 Identifikace metodou fyzikální analýzy . . . . .	30
3.2 Experimentální identifikace . . . . .	31
3.3 Identifikační proces . . . . .	31
<b>4 Identifikace modelu vrtulníku</b>	<b>33</b>
4.1 Identifikace subsystému hlavní vrtule – elevace . . . . .	34
4.2 Identifikace subsystému zadní motor – azimut . . . . .	38



4.3	Křížová vazba Hlavní motor – azimut . . . . .	40
4.4	Křížová vazba zadní motor – elevace . . . . .	42
<b>5</b>	<b>Popis a diskretizace nalezeného systému</b>	<b>44</b>
5.1	Diskretizace SISO systémů . . . . .	44
5.2	Diskretizace MIMO systému . . . . .	46
<b>6</b>	<b>Historie prediktivního řízení</b>	<b>48</b>
6.1	MPC v průmyslu . . . . .	49
<b>7</b>	<b>Prediktivní řízení založené na stavovém popisu</b>	<b>51</b>
7.1	Odvození rozšířeného modelu soustavy . . . . .	51
7.2	Predikce stavových a výstupních proměnných . . . . .	52
7.3	Optimalizace . . . . .	54
7.4	Prediktivní řízení se stavovou estimací . . . . .	56
7.5	Prediktivní řízení s omezeními . . . . .	57
<b>8</b>	<b>Návrh MPC pro SISO systém hlavní vrtule</b>	<b>62</b>
8.1	Varianta s analytickým výpočtem . . . . .	62
8.2	Varianta s kvadratickým programováním . . . . .	66
<b>9</b>	<b>Návrh MPC pro SISO systém zadní vrtule</b>	<b>68</b>
9.1	Varianta s analytickým výpočtem . . . . .	68
9.2	Varianta s kvadratickým programováním . . . . .	70
<b>10</b>	<b>Návrh MPC pro MIMO systém vrtulníku</b>	<b>72</b>
<b>11</b>	<b>Explicitní prediktivní řízení</b>	<b>78</b>
<b>12</b>	<b>Návrh explicitního MPC pro SISO systém hlavního motoru</b>	<b>81</b>
<b>13</b>	<b>Návrh explicitního MPC pro SISO systém zadního motoru</b>	<b>86</b>
	<b>Závěr</b>	<b>91</b>

Seznam použité literatury	95
Dodatek A Identifikace	96
Dodatek B Návrh implicitního MPC	97
Dodatek C Návrh explicitního MPC	99
Dodatek D Struktura aplikace VI	100
Dodatek E Inicializace VI	101
Dodatek F Řídicí část VI	102
Dodatek G Prediktivní regulátor VI	103
Dodatek H Stavový pozorovatel VI	104
Dodatek I Logování dat VI	105
Dodatek J Vytvoření logu VI	106
Dodatek K Data k zápisu VI	107
Dodatek L Shutdown VI	108
Dodatek M Grafy analytické verze MIMO MPC	109

## Seznam zkratek

<b>DMA</b>	Přímý přístup do paměti
<b>DMC</b>	Dynamic Matrix Control
<b>FM</b>	Fakulta mechatroniky, informatiky a mezioborových studií Technické univerzity v Liberci
<b>FPGA</b>	Programovatelné hradlové pole
<b>HEICON</b>	Hierarchical Constraint Control
<b>HMI</b>	Rozhraní mezi člověkem a strojem
<b>LabVIEW</b>	Laboratory Virtual Instrument Engineering Workbench
<b>LQR</b>	Lineárně kvadratický regulátor
<b>MAC</b>	Model Algorithmic Control
<b>MATLAB</b>	Matrix Laboratory
<b>MIMO</b>	System s více vstupy a výstupy
<b>MPC</b>	Prediktivní řízení založené na modelu
<b>MPHC</b>	Model Predictive Heuristic Control
<b>PAC</b>	Programovatelný automatický kontrolér
<b>PC</b>	Osobní počítač
<b>PID</b>	Proporcionálně integračně derivační regulátor
<b>PLC</b>	Programovatelný logický automat
<b>OS</b>	Operační systém
<b>QDMC</b>	Quadratic Dynamic Matrix Control
<b>QUADPROG</b>	Funkce kvadratického programování
<b>RMPCT</b>	Robust Model Predictive Control Technology
<b>RTOS</b>	Operační systém reálného času
<b>RT</b>	Real-Time
<b>SISO</b>	System s jedním vstupem a výstupem
<b>TUL</b>	Technická univerzita v Liberci
<b>USB</b>	Univerzální sériová sběrnice

## Seznam symbolů

$\mathbf{0}_m$	nulová matice o rozměrech $m \times m$
$\mathbf{1}_m$	jednotková matice o rozměrech $m \times m$
$\mathbf{A}$	matice vnitřních vazeb systému - systémová matice
$\mathbf{B}$	matice vazeb vstupů na stavy systému
$\mathbf{C}$	matice vazeb stavů systému na výstupy - výstupní matice
$\mathbf{C}_o$	matice říditelnosti
$\mathbf{D}$	matice přímých vazeb mezi vstupy a výstupy - matice převodu
$d(k)$	model stavové poruchy
$\mathbf{F}, \Phi$	dvojice matic využity pro predikci
$\mathbf{G}(s)$	Přenosová funkce systému
$J$	kritérium implicitního MPC
$J_e$	kritérium explicitního MPC
$J_{id}$	kvadratické integrální kritérium kvality
$\mathbf{K}_{mpc}$	zesílení stavové zpětné vazby při využití estimace
$\mathbf{K}_{ob}$	zesílení estimátoru
$\mathbf{K}_y$	zesílení stavové zpětné vazby vzhledem k referenci
$\mathbf{K}_x$	zesílení stavové zpětné vazby vzhledem k $\Delta x$
$\mathbf{M}, \gamma$	dvojice matic definují omezení
$m$	počet vstupů
$n_1$	počet stavů
$N_c$	horizont řízení
$N_p$	horizont predikce
$\mathbf{Ob}$	matice pozorovatelnosti
$p(k)$	model výstupní poruchy
$q$	počet výstupů
$\bar{\mathbf{R}}$	váhová matice implicitního MPC
$\mathbf{R}_s$	matice referenčních signálů
$S$	váha regulační odchylky

$s_v$	pomocná proměnná uvolňující omezení
$T$	váha přírůstků akční veličiny
$\Delta U$	vektor optimálních akčních zásahů
$\Delta u^{\min}, \Delta u^{\max}$	minimální a maximální limity pro přírůstek akčního zásahu
$U(s)$	Laplaceův obraz vstupního signálu
$u(k)$	akční veličina
$u^{\min}, u^{\max}$	minimální a maximální limity pro akční zásah
$w(k), r(k)$	žádaná veličina, referenční signál
$x(k)$	stavový vektor
$\hat{x}(k)$	estimovaný stavový vektor
$\tilde{x}(k)$	chyba estimace stavové vektoru
$x(k_i + m k_i)$	predikovaná hodnota stavového vektoru v čase $m$
$Y$	vektor predikovaných výstupního signálu
$Y(s)$	Laplaceův obraz výstupů
$y(k)$	regulovaná veličina
$y(k_i + m k_i)$	predikovaná hodnota regulované veličiny v čase $m$
$y^{\min}, y^{\max}$	minimální a maximální limity výstupy

## Seznam obrázků

1.1	Model vrtulníku CE 150 [2]	18
2.1	Princip SCAN módu [8]	23
2.2	Jitter [6]	25
2.3	Struktura řídicí aplikace	26
2.4	Náhled uživatelského prostředí	27
2.5	Vývojový diagram deterministické smyčky	28
4.1	Model vícerozměrové soustavy vrtulníku	33
4.2	Měření statické charakteristiky hlavní vrtule	34
4.3	Statická charakteristika hlavní vrtule	35
4.4	Výřez statické charakteristiky hlavní vrtule	35
4.5	Hlavní motor – měření dat	36
4.6	Hlavní motor – identifikace	37
4.7	Hlavní motor – validace	37
4.8	Zadní motor – měření dat	38
4.9	Zadní motor – identifikace	39
4.10	Zadní motor – validace	39
4.11	Vazba hlavní motor – azimut – měření dat	40
4.12	Vazba hlavní motor – azimut – identifikace	41
4.13	Vazba hlavní motor – azimut – validace	41
4.14	Vazba zadní motor – elevace – měření dat	42
4.15	Vazba zadní motor – elevace – identifikace	43
4.16	Vazba zadní motor – elevace – validace	43

7.1	Blokové schéma prediktivní regulace s pozorovatelem [15]	57
8.1	Porovnání regulace hlavního motoru analyticky	64
8.2	Regulace hlavního motoru analyticky - maximální elevace	64
8.3	Regulace hlavního motoru analyticky s poruchou	65
8.4	Zadní motor – průběh poruchy analyticky	65
8.5	Regulace hlavního motoru quadprog – maximální elevace	66
8.6	Porovnání regulace hlavního motoru quadprog	67
8.7	Regulace hlavního motoru quadprog s poruchou	67
9.1	Regulace zadního motoru analyticky	69
9.2	Regulace zadního motoru analyticky s poruchou	69
9.3	Hlavní motor – průběh poruchy analyticky	70
9.4	Regulace zadního motoru quadprog	70
9.5	Regulace zadního motoru quadprog s poruchou	71
9.6	Hlavní motor – průběh poruchy quadprog	71
10.1	MIMO regulace quadprog na konstantní hodnoty	73
10.2	MIMO regulace quadprog průběh akčních zásahů	74
10.3	MIMO regulace quadprog proměnná elevace	75
10.4	MIMO regulace quadprog proměnný azimut	75
10.5	MIMO regulace quadprog obě proměnné	76
10.6	MIMO regulace quadprog obě proměnné průběh akčních zásahů	77
10.7	MIMO regulace analyticky obě proměnné	77
11.1	Vývojový diagram exportu modelu [19]	80
12.1	Regulace hlavního motoru explicitní – srovnání	82
12.2	Regulace hlavního motoru explicitní - maximální elevace	83
12.3	Regulace hlavního motoru explicitní bez poruchy	84
12.4	Regulace hlavního motoru explicitní s poruchou	84
12.5	Zadní motor – průběh poruchy explicitně	85

13.1	Regulace zadního motoru explicitní v1 . . . . .	87
13.2	Regulace zadního motoru – akční zásahy . . . . .	88
13.3	Regulace zadního motoru explicitní s poruchou v1 . . . . .	88
13.4	Hlavní motor – průběh poruchy explicitní . . . . .	89
13.5	Regulace zadního motoru explicitní s poruchou v2 . . . . .	90



## Úvod

S rozvojem automatizace a robotiky zároveň rostou i požadavky na kvalitu a rychlost řízení, kde je v současné době trendem omezení působení lidského faktoru. V praxi se lze nejčastěji setkat s PLC systémy, kde většina z nich umožňuje vcelku snadnou implementaci klasického či modifikovaného PID regulátoru. Tento typ je pro mnohé aplikace dostačující, avšak při řízení složitých vícerozměrových soustav se mohou dostavit problémy. Z tohoto důvodu začaly vznikat moderní a sofistikované řídicí struktury, kam se řadí i prediktivní řízení založené na modelu.

Počátky prediktivního řízení se datují přibližně do 60. let 20. století, kde se Rudolf E. Kalman začal zabývat hledáním optimálního řešení akčních zásahů. Na těchto základech byl navržen lineární kvadratický regulátor LQR. Následně se začaly objevovat první verze různých prediktivních algoritmů. Pojem prediktivního řízení totiž obsahuje širokou škálu algoritmů, kde jednotlivé rozdíly mohou být ve struktuře použitého modelu a penalizační funkci.

Mezi výhody těchto regulátorů patří především schopnost pracovat se zadanými technologickými omezeními a univerzální postup návrhu pro SISO i MIMO systémy. Prediktivní regulátor je navíc možné využít pro mnoho druhů procesů s jednoduchou dynamikou až po nestabilní a neminimálně fázové systémy a vícerozměrové soustavy. Naopak mezi nevýhody lze zařadit požadavek na vyšší výpočetní výkon než v případě PID regulátoru, ovšem se současným trendem neustálého zvyšování výpočetního výkonu pomalu i tento problém odpadá. Alternativou k řešení této problematiky tak může být explicitní prediktivní regulátor, kde je optimalizace vyřešena offline. Největší úskalí při návrhu prediktivního regulátoru tak spočívá v nutnosti dostatečně přesného modelu řízené soustavy.

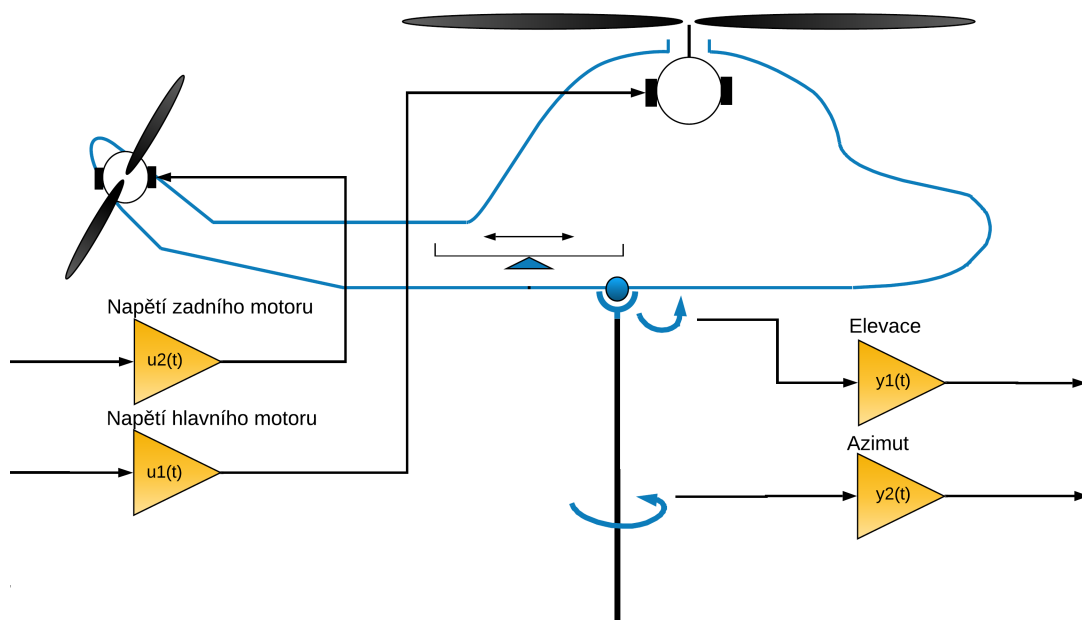
Cílem této práce je vytvoření implicitní a explicitní verze MPC a její následné ověření na fyzikálním modelu. Po seznámení s vlastnostmi modelu následuje tentýž úkol s programovatelným kontrolérem reálného času CompactRIO, jenž bude využit pro ovládání modelu vrtulníku, archivaci dat a komunikaci s vizualizací na standardním PC. Na základě těchto poznatků bylo přistoupeno k nalezení matematického modelu soustavy, kde bylo využito experimentálních metod identifikace, konkrétně spojitě, parametrické offline metody. K těmto účelům byl vytvořen identifikační skript v prostředí MATLAB.

Následně je možné přistoupit k návrhu implicitního a explicitního MPC na základě identifikovaných přenosů. Nejprve je navržena implicitní varianta, jež využívá jak analytickou, tak i verzi s kvadratickým programováním. Další část je zaměřena na explicitní variantu MPC s využitím Hybrid Toolboxu. Je zde představena a využita možnost export modelu z MATLABu do prostředí LabVIEW. Všechny navržené regulátory jsou otestovány na reálné soustavě modelu vrtulníku a následně jsou diskutovány a provedeny možné změny, které by vedly ke zkvalitnění regulačních pochodů.

# 1 Model vrtulníku CE 150

Model laboratorního vrtulníku CE 150 je jedním z unikátních produktů určených k teoretickému studiu a praktickému zkoumání základních či pokročilých principů řízení. To zároveň zahrnuje oblasti:

- modelování dynamických systémů
- identifikace
- analýza naměřených dat
- návrh klasických či moderních regulačních struktur



Obrázek 1.1: Model vrtulníku CE 150 [2]

Model se skládá z těla vrtulníku, na němž jsou umístěny dva stejnosměrné motory a servomechanismus. Motory ovládají signálem 0–10 V hlavní, respektive zadní vrtuli. Servomechanismus slouží k posunu přídatné zátěže, čímž dochází ke změně rozložení váhy. Umístění posuvné zátěže bylo zvoleno jako statické ve vzdálenosti  $l = 7$  cm na základě výsledků dle [1]. V této poloze byly naměřeny nejlepší vlastnosti z hlediska velikosti vstupního napětí a linearity statické charakteristiky. Tělo vrtulníku je spojeno s podstavcem, umožňuje tedy dva stupně volnosti:

- rotace kolem horizontální osy
- rotace kolem vertikální osy

Model tak může být popsán jako systém se dvěma vstupy:

- $u_1$  – napětí hlavního motoru
- $u_2$  – napětí zadního motoru

a dvěma výstupy:

- $y_1$  – úhel elevace
- $y_2$  – úhel azimut

Výstup  $y_1$  odpovídá úhlu mezi svislou a podélnou osou těla vrtulníku, zatímco výstup  $y_2$  označuje úhel v horizontální rovině mezi podélnou osou těla vrtulníku a jeho nulovou polohou. Oba tyto úhly jsou snímány pomocí inkrementálních rotačních čidel a převedeny na unifikovaný signál 0 – 10 V, který odpovídá aktuální poloze. Model vrtulníku umožňuje rozsah  $\pm 45^\circ$  v elevaci a  $\pm 135^\circ$  v azimutu. [3]

## 2 CompactRIO

CompactRIO je výkonný, rekonfigurovatelný vestavěný systém, který se skládá ze třech hlavních komponent:

- Operační systém reálného času
- FPGA
- Průmyslové I/O moduly

Procesor reálného času nabízí spolehlivé chování a vyniká v analýze a operacích s plovoucí desetinnou čárkou, zatímco FPGA exceluje v menších úlohách, které vyžadují přesné časování a vysokorychlostní logiku. Aplikace běžící na CompactRIO často využívají HMI, které operátorovi poskytují uživatelské rozhraní ke sledování stavu systému a nastavování parametrů. [4]

LabVIEW je graficky orientované programovací prostředí, určené k vývoji sofistikovaných řídicích systémů, jež využívá grafické ikony a vodiče, čímž vytváří vývojový diagram. Prostředí nabízí integraci s velkým množstvím hardwarových zařízení a poskytuje mnoho vestavěných knihoven pro realizaci řídicích struktur, analýzu dat a vizualizaci. [5]

Real-time kontrolér obsahuje procesor, který deterministicky vykonává aplikace reálného času a nabízí tak záznam dat, komunikaci s perifériemi a sledování vykonávání programu. Dále je možné využít vestavěný USB, RS232 a ethernet port. Operační systém reálného času je schopen vykonávat programy se specifickými časovými požadavky, které jsou důležité pro mnoho průmyslových či vědeckých projektů. Klíčovou součástí potřebnou k vytvoření aplikace reálného času je speciální operační

system, proto je standardní PC s klasickým OS nepřijatelný. Může zde totiž v kterémkoliv čase dojít ke zpoždění vykonávání programu vlivem např. probíhajících antivirových kontrol a úloh na pozadí, aktualizací a mnoho dalších. Pro programy, které potřebují k vykonání určitý čas, tak tato zpoždění mohou způsobit selhání systému. Z tohoto důvodu je RTOS navržen tak, aby spouštění programu proběhlo s přesným časováním. RTOS je tak vhodné použít v následujících případech:

- Vykonávání iterací s přesným časováním
- Detekce splnění termínu dokončení
- Rozvrhování priorit pro různé části programu
- Vykonání úloh s garancí splnění nejhoršího případu

Kromě přesného časování RTOS může být využit v aplikacích, kde je vyžadován spolehlivý chod bez přerušení v řádech měsíců či roků. To je důležité pro aplikace, kde prostoje jsou velice nákladné. Hardwarové vybavení je taktéž navrženo tak, aby bylo odolné v nevlídných podmínkách. Každá série výrobků od National Instruments využívá jeden ze tří možných operačních systémů reálného času – VxWorks, NI Linux Realtime, Pharlap. LabVIEW poskytuje určitou úroveň abstrakce, díky které je možné se vyhnout detailům RTOS. [6]

## 2.1 FPGA

Konfigurovatelné FPGA je hlavní částí architektury tohoto vestavěného systému. Je přímo spojeno se vstupy a výstupy pro získání vysokorychlostního přístupu k jednotlivým modulům a umožňuje tak jejich časování, spouštění a synchronizaci. Jelikož je každý modul připojen přímo s FPGA místo pomocí sběrnice, je možné dosáhnout téměř nulové latence. FPGA dále umožňuje programům přístup k I/O s jitterem menším než 500 ns. Dále je pak možné přímo programovat toto FPGA a přizpůsobit tak systém požadovaným potřebám. Vzhledem k vysoké rychlosti FPGA je možné jej využít při návrhu řídicích systémů, výkonných regulačních smyček a filtraci signálu.

Tímto způsobem je možné vykonat více než 20 regulačních PID smyček zároveň rychlostí 100 kHz. [6]

## 2.2 I/O moduly

Vstupně výstupní moduly obsahují ochranné izolační prvky, převodníky a vestavěné konektory. V současné době existuje více než 70 modulů pro CompactRIO, což umožňuje připojení různorodých senzorů a akčních členů. Dále je možné využít výrobky třetích stran nebo pomocí NI 9951 Development kitu vyvinout modul podle vlastních specifikací. Mezi hlavní moduly patří např.: měření proudu, napětí, odporu, teploty, akcelerometry, reléové jednotky a univerzální vstupy a výstupy. [7]

## 2.3 FPGA Interface mód

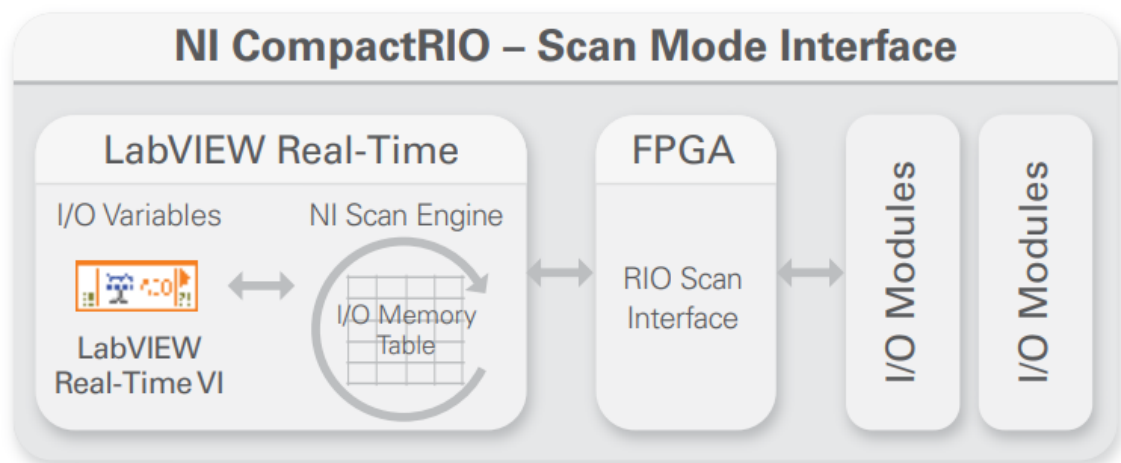
S využitím LabVIEW FPGA Interface módu je možné dosáhnout plného výkonu, které CompactRIO nabízí. Pomocí LabVIEW FPGA lze implementovat vlastní časování a spouštění, zpracování signálu a analýzy, vytvářet vlastní protokoly a přistupovat ke vstupům a výstupům v maximální rychlosti. Při komunikaci mezi FPGA VI a RT VI je možné odesílat data pomocí DMA FIFO. Tento způsob umožňuje vysokorychlostní přenos.

Stejně jako řídicí systémy založené na procesorech, tak i FPGA systémy je vhodné použít pro řídicí aplikace, kde se využívá analogových procesů, diskrétní logiky a stavových automatů. Nicméně systémy založené na FPGA se výrazně liší od systémů využívající pouze procesory. Pokud tedy aplikace využívá některé z níže uvedených požadavků, je vhodné se zamyslet, zda nezvolit LabVIEW FPGA mód. [6]

- Maximální výkon a spolehlivost
- Vysokorychlostní získávání/generování dat
- Uživatelská volba časování/spouštění/synchronizace
- Coprocessing

## 2.4 CRIO SCAN mód

V tomto módu je možné programovat pouze procesor reálného času. Tento režim poskytuje předdefinované FPGA, které periodicky skenuje I/O a umísťuje je do paměťového prostoru, který je přístupný pro Labview RT. Scan mód je vhodný pro aplikace, které vyžadují přístup k I/O v taktu stovek Hertz. Některé průmyslové řídicí a monitorovací aplikace jsou založeny pouze na aktuálních hodnotách na vstupních/výstupních kanálech. Nebere se tak v potaz časová historie dat, tudíž se neporovnávají aktuální hodnoty s předešlými a také se neměří rychlost změn dat. Často se nevyžaduje rychlost vyšší než 500 Hz. Technologie RIO SCAN umožňuje přístup k I/O bez nutnosti psát FPGA kód. Když kontrolér přistupuje k I/O pomocí SCAN interface, jsou automaticky detekovány moduly a hodnoty jsou umístěny do paměťové tabulky. Blokové schéma principu SCAN módu:



Obrázek 2.1: Princip SCAN módu [8]

Před použitím tohoto módu by měl být zvážena požadovaná rychlost smyčky a počet kanálů. I přesto však tento mód poskytuje několik benefitů. [8]

- Jednodušší programování
- Automatická detekce I/O modulů
- Diagnostika a ladění programu



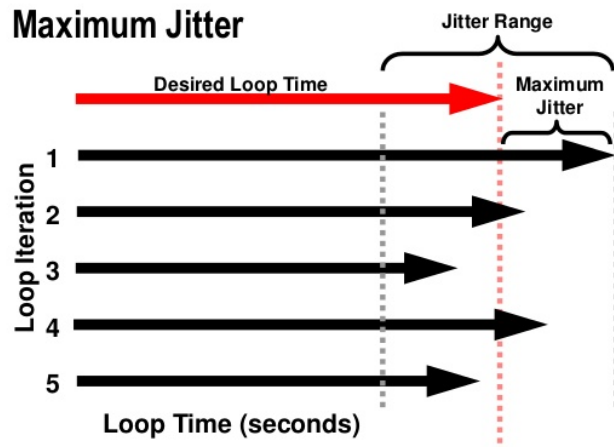
## 2.5 Hybridní mód

S tímto přístupem je možné vybrat moduly, ke kterým se bude přistupovat pomocí FPGA, a moduly, u kterých bude použito RIO Scan rozhraní. Je možné tak využít programování FPGA k obsluze vysokorychlostních operací, které SCAN mód nepodporuje. Tento mód má i svoje limitace. Jednou z nich je výrazné zvýšení času kompilace, jelikož kompilátor musí zkombinovat bitfile vytvořený v RIO SCAN rozhraní a bitfile, který vznikl na FPGA. Dalším problémem může být počet DMA kanálů, které je možné použít v FPGA kódu, jelikož SCAN mód ke své funkci potřebuje dva kanály. Většina FPGA obvodů má tři, takže je pak možné využít pouze jeden kanál pro realizaci FPGA kódu. [9]

## 2.6 LabVIEW RT

Po zvolení programovacího módu je vhodné vytvořit kostru aplikace. Jednotlivé procesy je žádoucí rozdělit na tzv. SubVI, které zpřehledňují aplikaci a usnadňují tak vývoj. Každé SubVI tak může obsahovat obsluhu jednoho či více procesů. Po návrhu struktury následuje implementace jednotlivých procesů.

Při návrhu aplikace reálného času je možné vybrat ze dvou úrovní spolehlivosti. První úroveň garantuje, že programy běží v určitém taktu bez přerušení. Operační systém reálného času, který je použit v CompactRIO, poskytuje tuto úroveň bez nutnosti využití jakýkoliv speciálních programovacích technik. Je možné tak implementovat procesy pomocí běžných cyklů While s časováním, druhá úroveň spolehlivosti zahrnuje determinismus a jitter. Determinismus popisuje, jak proces reaguje na vnější události, nebo provádí operace během stanoveného limitu, kdežto jitter vyjadřuje kolísání rychlosti odezvy viz graf 2.2.



Obrázek 2.2: Jitter [6]

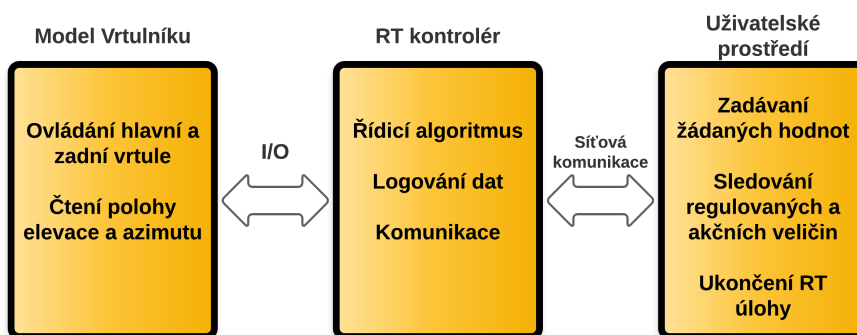
CompactRIO pomáhá snížit toto kolísání, což může být požadavek v některých specifických oblastech. Existují dvě možnosti návrhu deterministických aplikací:

1. Implementace kritických úloh pomocí Labview FPGA
  - Jelikož FPGA běží bez využití OS, je možné garantovat, že kritické úlohy neselžou.
2. Implementace kritických úloh pomocí Timed Loop v Labview RT
  - Timed loop je komplexnější verze While cyklu, jenž je navržena tak, aby úlohy byly vykonávány deterministicky s minimálním jitterem.

Běžně se v aplikacích vyskytují jak deterministické, tak i nedeterministické úlohy. Deterministické procesy se musí pokaždé dokončit včas, proto využívají vyhrazené zdroje k včasnému dokončení. Rozdělení deterministických procesů od ostatních poskytuje dostatek těchto zdrojů. Například pokud řídicí aplikace pořizuje data v pravidelných intervalech a následně je ukládá na disk, je nutné deterministicky snímat data v daných intervalech, nicméně ukládání dat je nedeterministická úloha, jelikož operace se soubory mají nepredikovatelné doby odezev, které závisí na druhu hardwaru a dostupnosti jejich prostředků. Proto je vhodné použít Timed loop pro měření dat a While loop pro ukládání do souboru. [6]

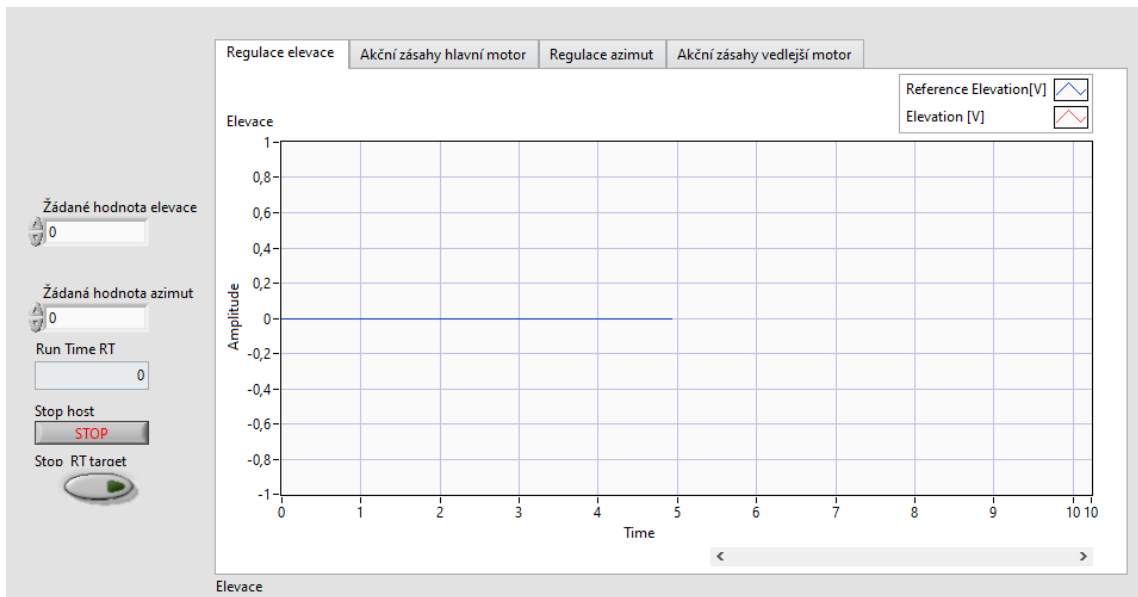
## 2.7 Návrh struktury řídicí aplikace pro CompactRIO

K vývoji aplikace pro CompactRIO je využito programovací prostředí LabVIEW. Cílem je sestavit řídicí algoritmus založený na prediktivním kontroléru, respektive explicitním prediktivním kontroléru. Aplikace má zajistit regulaci nejen jednotlivých SISO systémů vrtulníku ale i komplexní řízení v obou regulovaných veličinách, tedy elevaci a azimutu. Dále bude vytvořeno uživatelské prostředí, které umožní základní ovládání a vizualizaci. Součástí návrhu je i archivace dat systému na lokální disk CompactRIO. Struktura řídicí aplikace je zobrazena na obr. 2.3.



Obrázek 2.3: Struktura řídicí aplikace

Uživatelské rozhraní bude spouštěno na osobním počítači. Úkolem aplikace je komunikovat s programem na RT targetu, jenž zajišťuje sběr dat z vrtulníku a vykonává řídicí algoritmus založený na prediktivním řízení. Skrze prostředí bude možné zadávat jednotlivé žádané veličiny směrem k RT aplikaci a případně i její běh zastavit. Naopak směrem z CompactRIO budou odesílána data o vypočítaných akčních zásazích a aktuálně naměřených výstupech. Tyto údaje pak budou graficky prezentovány. Na obr. 2.4 je zobrazena vizuální podoba uživatelské aplikace.



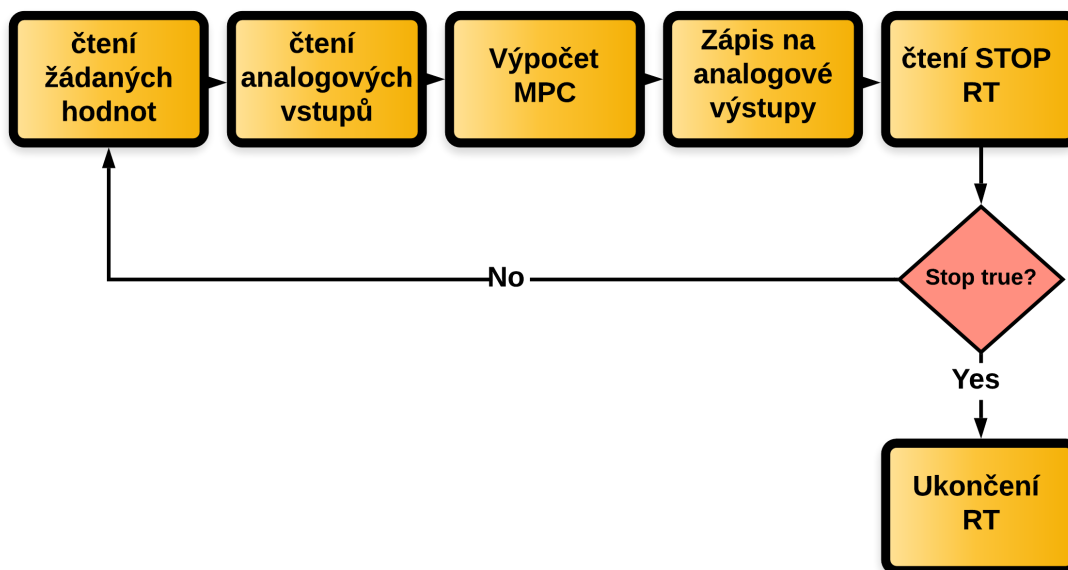
Obrázek 2.4: Náhled uživatelského prostředí

Řídicí aplikaci pro RT target lze rozdělit na 4 hlavní části:

- Inicializační část
- Regulace
- Záznam dat
- Ukončení aplikace

V inicializační části jsou založeny a inicializovány proměnné potřebné ke komunikaci s uživatelskou aplikací. Dále je pak inicializován prediktivní regulátor, který na základě stavového popisu předpočítá všechny potřebné parametry. Součástí je i výpočet zesílení stavového pozorovatele. Poslední nezbytný úkol je příprava souboru k logování záznamů. Logování bude probíhat do souboru typu `.txt`, jelikož poskytuje dobrý kompromis z hlediska přenositelnosti, velikosti souboru a jednoduchosti zápisu. V inicializační části je tedy vytvořen soubor, případně při shodě názvu souboru je navázáno na předcházející záznam. V druhé části probíhají dva paralelní procesy – regulace a ukládání dat. Deterministická úloha je v tomto případě cyklicky vykonávající se regulační smyčka s MPC regulátorem, jelikož musí být zaručeno

její včasné dokončení. K tomuto účelu je vhodné použít strukturu Timed Loop. Na následujícím diagramu 2.5 je znázorněn průběh deterministické smyčky.



Obrázek 2.5: Vývojový diagram deterministické smyčky

Nedeterministickou úlohou je v tomto případě archivace dat. Zde není bezpodmínečně nutné dodržet přesné časování. Z tohoto důvodu postačí použít `While loop`. Data log je aktualizován se stejnou periodou, jakou probíhá regulační smyčka `Timed loop`, přesto nesmí ovlivnit determinismus aplikace. Způsob ukládání dat je asynchronní a je využito vnitřní buffer, aby nedošlo ke ztrátě dat. Log obsahuje následující parametry: žádaná hodnota elevace a azimutu, regulovaná veličina elevace a azimutu, akční veličina napětí hlavního a vedlejší motor a aktuální čas. Dále také probíhá nedeterministická úloha síťové komunikace s uživatelským rozhraním, jež je spuštěné na standardním PC s OS Windows. Zde je zobrazen přehled hlavních úloh probíhajících na RT targetu.

- Výpočet akčního zásahu pomocí MPC
- Monitoring, zda nedošlo k požadavku na ukončení RT úlohy
- Případné ukončení úlohy
- Odesílání a přijímání požadavků z uživatelské aplikace

- Logování dat do souboru

Poslední část má zajistit správné ukončení úlohy, tedy nastavit všechny výstupy CompactRIO na bezpečné hodnoty, aby byla eliminována možnost zranění. Dále je též potřeba korektně ukončit logování dat a uzavřít soubor.

## 3 Teorie identifikace dynamických systémů

Pro úspěšný návrh řídicího algoritmu je vhodné, v některých případech i nezbytné, znát matematický model daného systému. Model umožňuje zkoumat vlastnosti a chování při situacích, které by byly v reálném světě těžko realizovatelné či velice nákladné. Model je však pouze abstrakcí reálného systému, jelikož při tvorbě je vhodné vyloučit nepodstatné vlivy a vlastnosti, které by model pouze zkomplikovaly a analýza by pak byla obtížně proveditelná. Model však musí obsahovat všechny charakteristické vlastnosti reálného systému. Ztotožnění modelu a reálného systému je cílem identifikace.

K identifikaci zkoumaného systému lze přistoupit ze dvou různých směrů. Analytický přístup využívá matematicko-fyzikální analýzu, empirický přístup naopak využívá experimentálních metod.

### 3.1 Identifikace metodou fyzikální analýzy

Matematický model je sestaven ze znalosti fyzikálních zákonů, které probíhají ve sledovaném objektu. Na jejich základě často vznikne soustava diferenciálních rovnic, které vede na stavový popis. Většinou tak vzniká model typu white-box, kdy je známá nejen struktura, ale i fyzikální děje probíhající uvnitř systému. Tento způsob však vyžaduje dokonalou znalost technologie, ve kterém systém pracuje, proto může být tento přístup nepoužitelný pro komplexní systémy či procesy se složitými fyzikálními interakcemi. Výhodou však je souvislost mezi parametry modelu a parametry reálné soustavy. Získané modely také mají většinou širší oblast platnosti než modely z experimentální identifikace. [10]

## 3.2 Experimentální identifikace

Tento způsob identifikace využívá naměřených dat při vhodném experimentu. Je ovšem nutné počítat s tím, že reálná data obsahují šum a mohou být ovlivněna poruchami, offsety či drifts. Matematický model je určen vyhodnocením závislosti výstupních veličin na vstupních. Model tak vyjadřuje vnější popis systému, tzv. black-box model, kde struktura není předem definována a musí tak být zvolena.

Souvislost mezi parametry modelu a konstrukčními parametry reálné soustavy není zřejmá. Tento typ také neumožňuje náhled do vnitřní struktury a většinou má svoji platnost pouze v okolí zvoleného pracovního bodu. Výhodou je pak relativně jednoduché sestavení a v mnohých případech je to jediná možnost popisu. [10]

## 3.3 Identifikační proces

Při výběru identifikační metody je nutné zvážit druh vstupního signálu a matematického modelu a také způsob vyhodnocování naměřených odezev. Dále je vhodné do procesu volby metody zahrnout dostupné apriorní informace o systému, předchozí zkušenosti a podmínky identifikace. Zvolená metoda tak musí umožňovat vyšetření dynamických vlastností systému při působení budicího signálu. Způsob identifikace, který využívá deterministické budicí signály je vhodný pouze tehdy, pokud je úroveň šumu malá vzhledem k úrovni testovacího signálu. V opačném případě je vhodné použít stochastické metody identifikace. Proces identifikace lze rozdělit do několika kroků. [11]

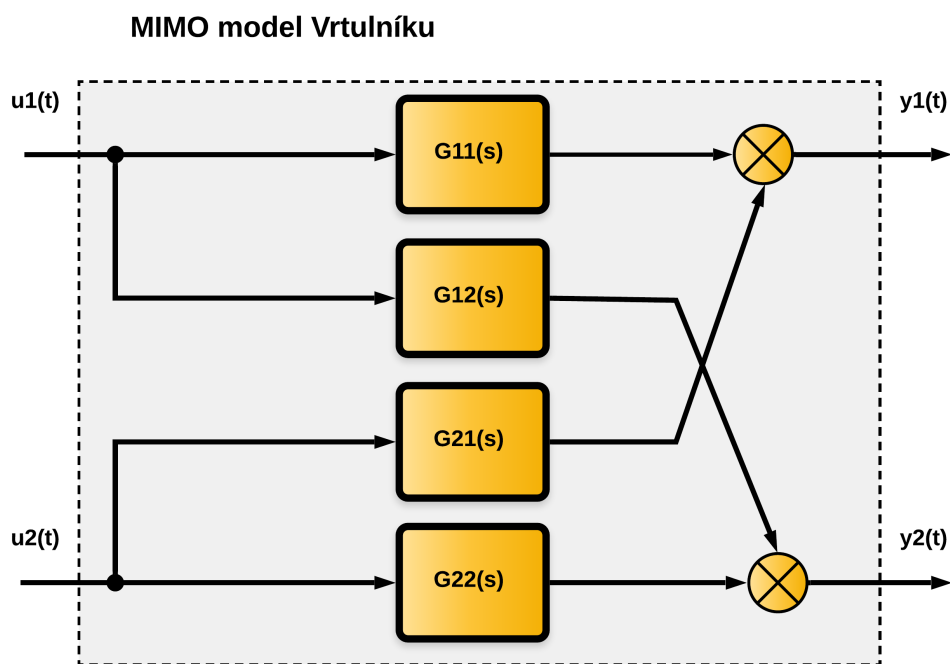
1. Návrh a realizace měření na reálné soustavě, volba pracovního bodu
2. Výběr úseku, na kterém bude provedena identifikace
3. Výběr struktury matematického modelu
4. Odhad parametrů modelu na základě měření a zvoleném kritériu
5. Verifikace modelu



Existuje několik možností, jak ověřit správnost zvoleného modelu. Intuitivně se nabízí možnost porovnat výstup reálné soustavy a modelu. Tato metoda však nedává informace o říditelnosti či pozorovatelnosti ani zda je zvolena správná struktura modelu. Další možný způsob je založen na frekvenčním přístupu, kdy se porovnávají frekvenční charakteristiky modelu a reálné soustavy.

## 4 Identifikace modelu vrtulníku

Z celkového pohledu se jedná o tzv. MIMO systém, tedy systém s více vstupními a výstupními veličinami. Každá z těchto vstupních veličin může ovlivňovat současně více výstupních. Vzájemnými interakcemi mezi vstupy a výstupy vznikají složité vazby a souvislosti. MIMO systém lze popsat pomocí soustavy diferenciálních rovnic, stavovým popisem nebo u časově invariantních lineárních soustav pomocí přenosových matic.



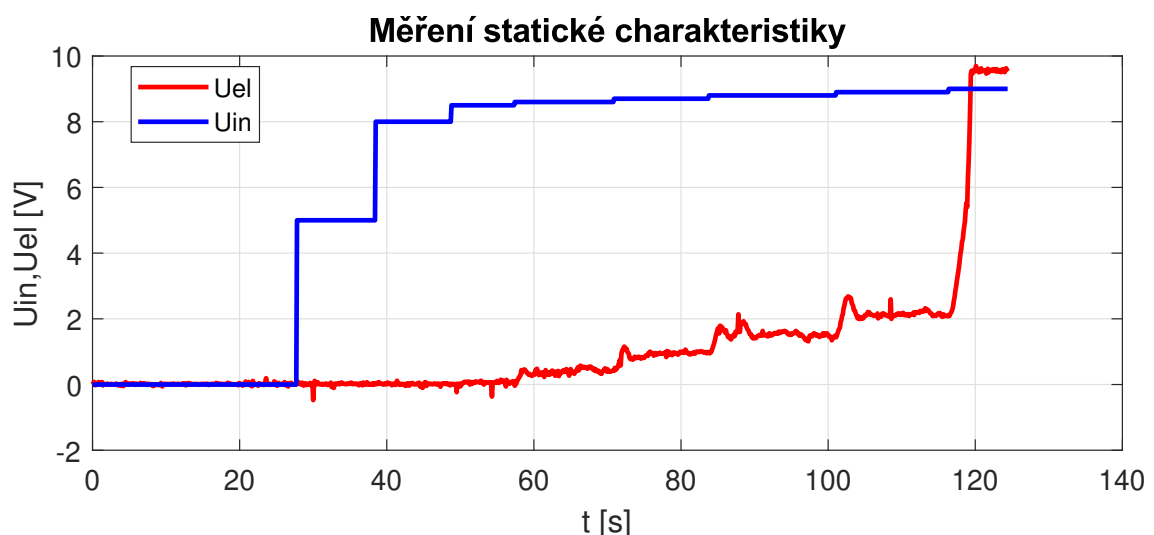
Obrázek 4.1: Model vícerozměrové soustavy vrtulníku

Před identifikací je vhodné provést dekompozici systému na relativně samostatné podsystémy, které jsme schopni identifikovat. Laboratorní model je možné rozdělit na jednotlivé systémy, jelikož umožňuje aretaci v elevaci či azimutu, čímž by mělo dojít k částečnému potlačení okolních vlivů. Jednotlivé subsystémy dle grafu 4.1 jsou:

- $G_{11}(s)$ : vazba napětí hlavní motor – elevace
- $G_{12}(s)$ : vazba napětí hlavní motor – azimut
- $G_{21}(s)$ : vazba napětí vedlejší motor – elevace
- $G_{22}(s)$ : vazba napětí vedlejší motor – azimut

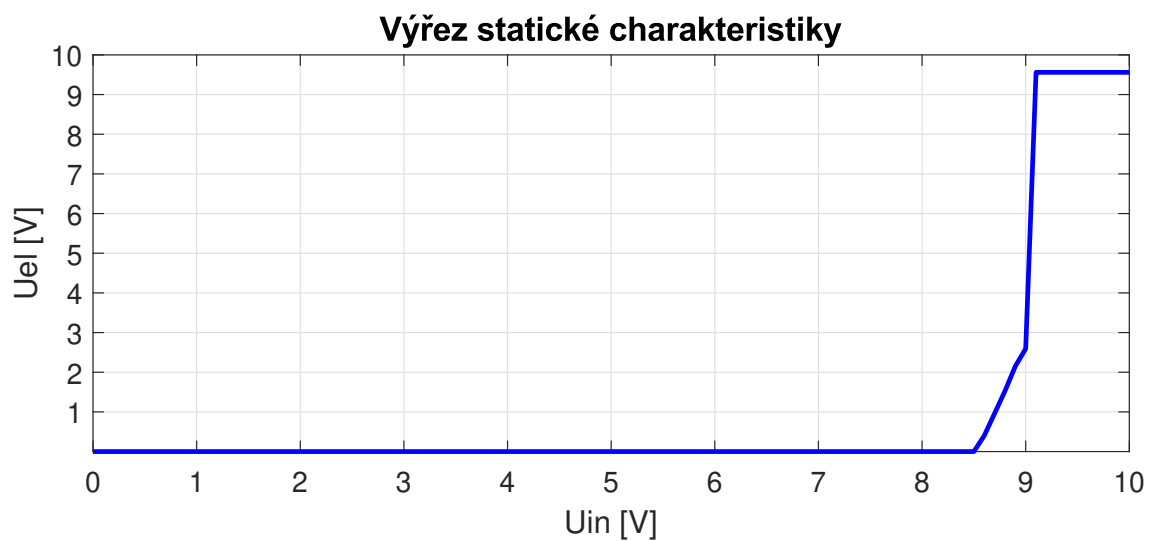
## 4.1 Identifikace subsystému hlavní vrtule – elevace

V první řadě je nutné ověřit statické vlastnosti identifikované soustavy. Tyto vlastnosti vyjadřují závislost výstupní veličiny na vstupní veličině po odeznění přechodového děje, má tedy smysl pouze u stabilních systémů. Navržená struktura měření spočívala v postupných skocích vstupní veličiny o dostatečné délce, tak aby došlo k ustálení výstupu.

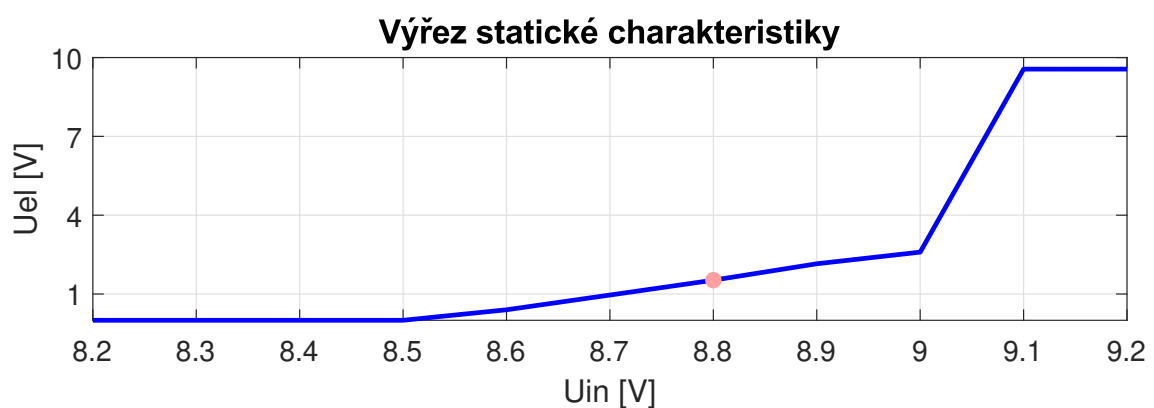


Obrázek 4.2: Měření statické charakteristiky hlavní vrtule

Z grafu měření 4.2, respektive statické charakteristiky 4.3 plyne, že subsystém hlavní vrtule – elevace je z celkového hlediska nelineární systém. Další významnou vlastností systému je, že změna elevace z nulové polohy nastává při hodnotách vstupního napětí okolo 8,5 V. Pokud je naopak vstupní napětí vyšší než 9 V, dochází k překlopení vrtulníku do maximální polohy. I když se jedná o nelineární systém, tak i zde je možné najít lineární část vhodnou pro řízení. V této části byl zvolen pracovní bod  $U = 8,8$  V, ve kterém byla provedena spojitá identifikace.



Obrázek 4.3: Statická charakteristika hlavní vrtule

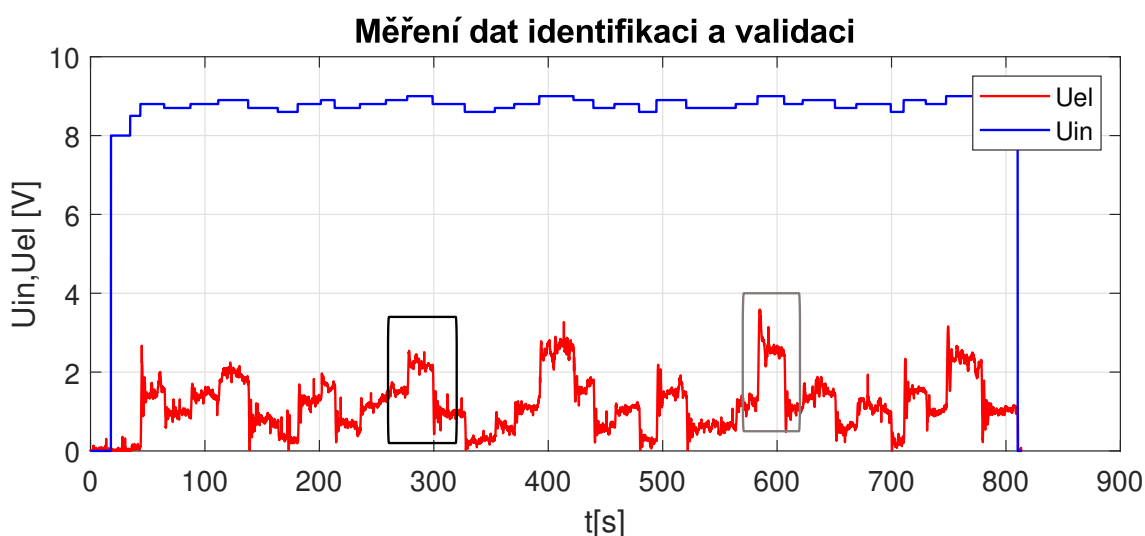


Obrázek 4.4: Výřez statické charakteristiky hlavní vrtule

Měření dat k identifikaci dynamických vlastností bylo provedeno v okolí pracovního bodu, kde byly prováděny malé skoky vstupní veličiny. Naměřená data byla rozdělena na identifikační a validační sadu. Identifikace byla provedena pomocí skriptu v MATLABu, který využíval funkci `fminsearch`. Struktura hledaného systému je ve formě přenosové funkce:

$$G(s) = \frac{K}{a_n s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \dots + 1} \quad (4.1)$$

Na naměřených datech 4.5 byla vybrána vhodná část k identifikaci (černý výběr) a validaci (šedý výběr).



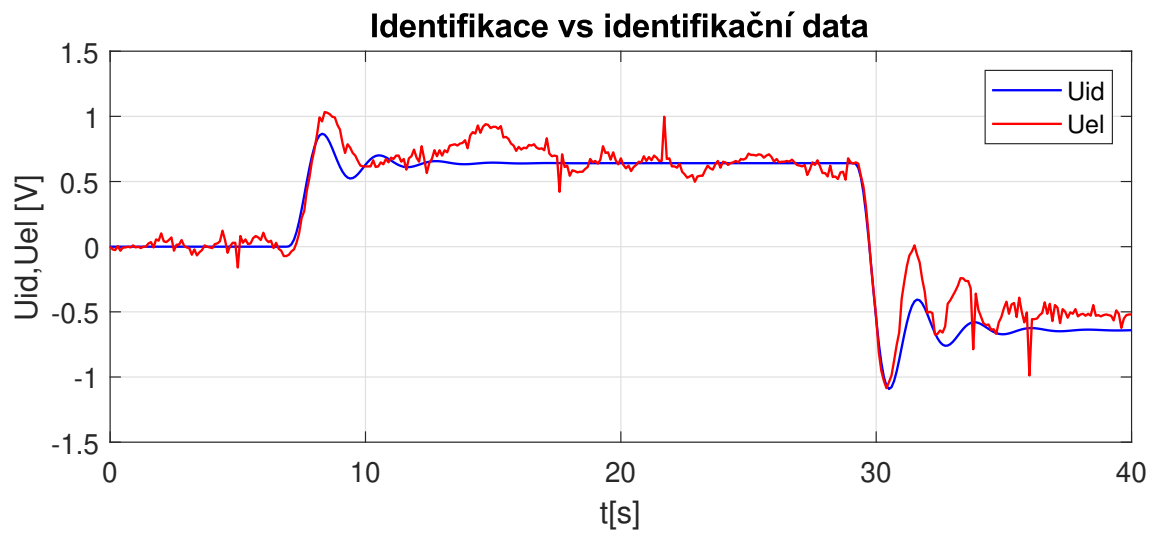
Obrázek 4.5: Hlavní motor – měření dat

Nejpřesněji popisuje dynamiku subsystému hlavní motor – elevace přenos 3. řádu. K numerickému vyčíslení shody všech modelů bylo využito kvadratické kritérium:

$$J_{id} = \sum_{k=1}^N [y_{model}(k) - y_{data}(k)]^2 \quad (4.2)$$

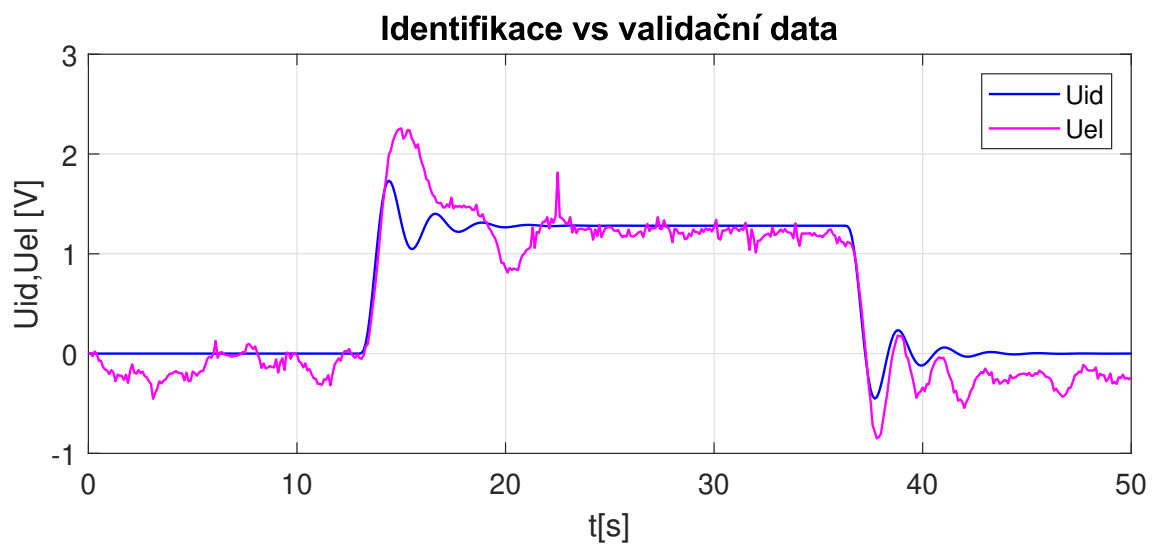
Hodnota tohoto kritéria dosahovala hodnoty 7,0576. Procentuální shoda byla 75,05 %. Celkový přenos má následující podobu:

$$G_{11}(s) = \frac{5,607}{0,03742s^3 + 0,1652s^2 + 0,4561s + 1} \quad (4.3)$$



Obrázek 4.6: Hlavní motor – identifikace

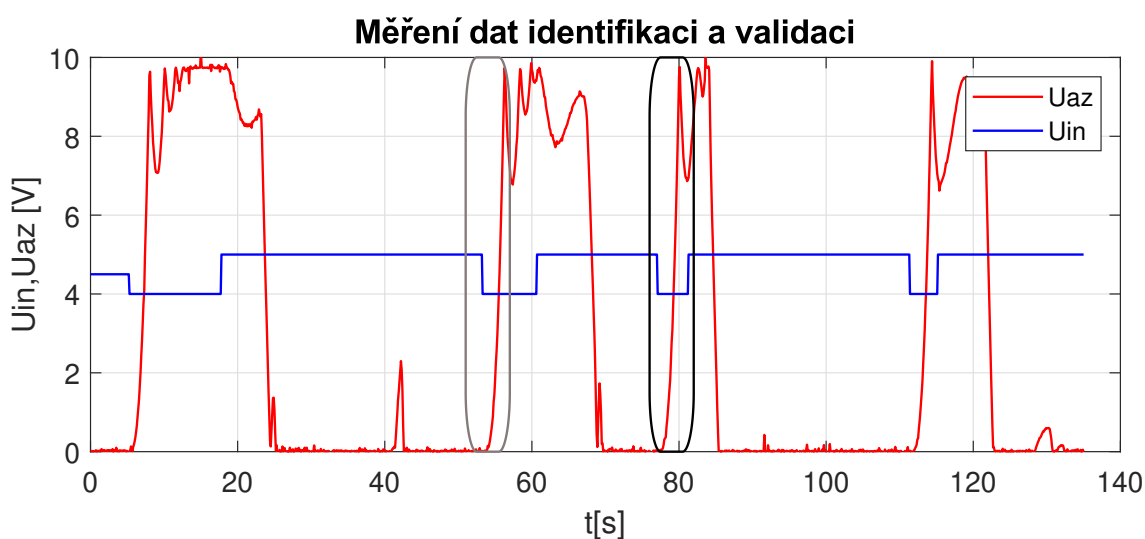
Nalezený přenos byl následně otestován na validačních datech. Zde dosahovala procentuální shoda modelu podobné hodnoty, konkrétně 67,78 %. Hodnota kritéria  $J_{id}$  byla rovna 31,4892.



Obrázek 4.7: Hlavní motor – validace

## 4.2 Identifikace subsystému zadní motor – azimut

Jelikož má tento subsystém astatický charakter, není u něj možné měřit statické vlastnosti, pouze jen ty dynamické. Tento typ systému je charakteristický tím, že nedochází k ustálení výstupu, ten se trvale mění až do dosažení mezního stavu daného konstrukcí zařízení. Opět byla provedena série měřicích experimentů a získaná data rozdělena na validační (šedý výběr) a identifikační (černý výběr) část.



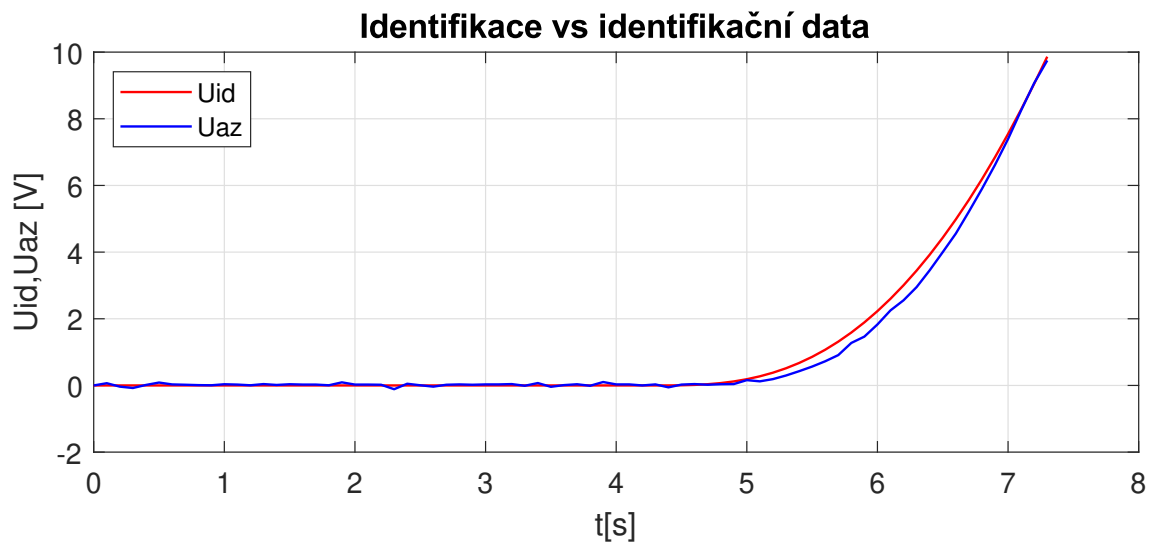
Obrázek 4.8: Zadní motor – měření dat

Struktura hledaného přenosu byla ve tvaru:

$$G(s) = \frac{K}{a_n s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \dots + s} \quad (4.4)$$

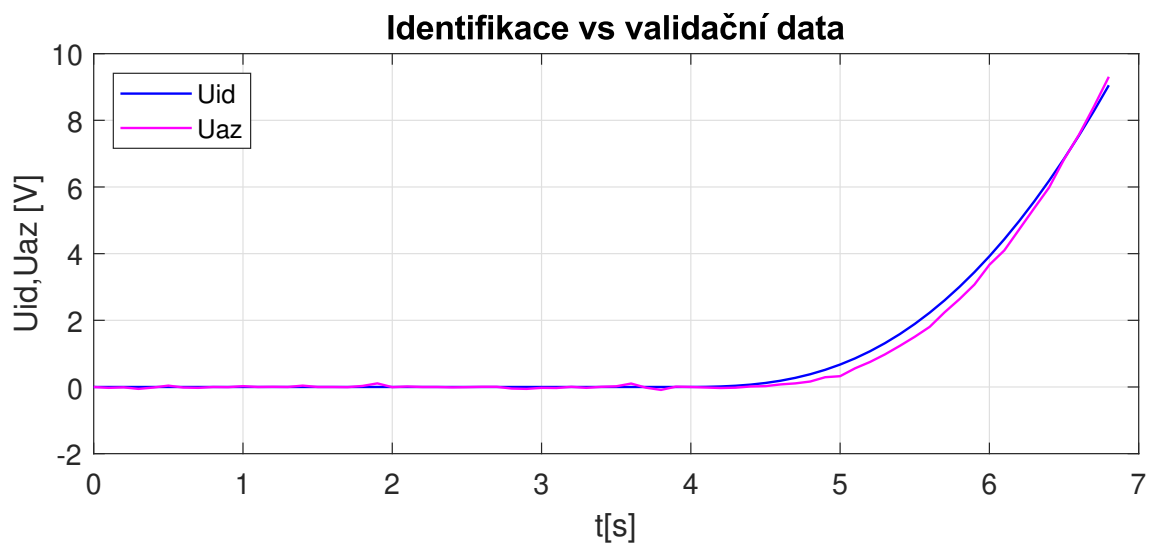
Tudíž systém s astatickem 1. řádu. Byly vyzkoušeny i struktury modelu s astatickem vyšších řádů, které ovšem nepřinesly zlepšení shody modelu. Nalezený přenos má následující podobu:

$$G_{22}(s) = \frac{-53,53}{14,25s^3 + 11,6s^2 + s} \quad (4.5)$$



Obrázek 4.9: Zadní motor – identifikace

Hodnota kritéria  $J_{id}$  byla 1,897. Procentuální shoda dosahovala 95,24 %. Opět byl nalezený model porovnán vzhledem k validačním datům. Zde kritérium  $J_{id}$  mělo hodnotu 2,5246 a procentuální shoda byla 94,58 %.

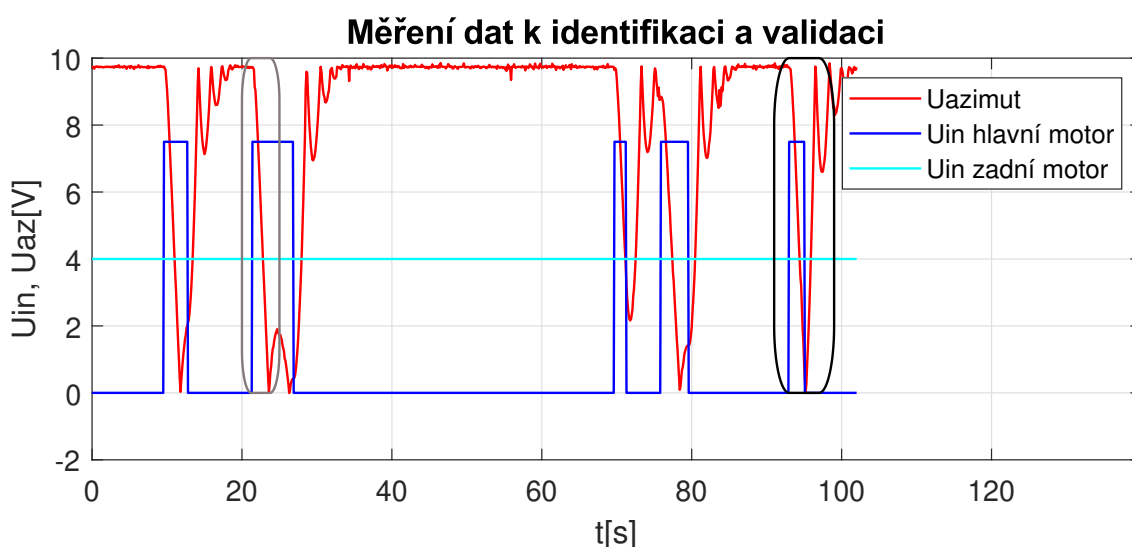


Obrázek 4.10: Zadní motor – validace



### 4.3 Křížová vazba Hlavní motor – azimut

Jelikož laboratorní model umožňuje aretaci v elevaci, respektive azimutu, lze tak poměrně snadno provést měření křížových vazeb. V tomto případě byla provedena fixace v elevaci, následně byly provedeny skoky vstupního napětí hlavního motoru. Napětí na vedlejším motoru bylo po celou dobu měření konstantní 4 V. Pokud by byla hodnota větší než 5 V, zadní vrtule by vytvářela stejně orientovaný moment, z toho důvodu je nutné se pohybovat pod touto hranicí. U naměřených dat opět došlo na rozdělení na validační a identifikační sadu.



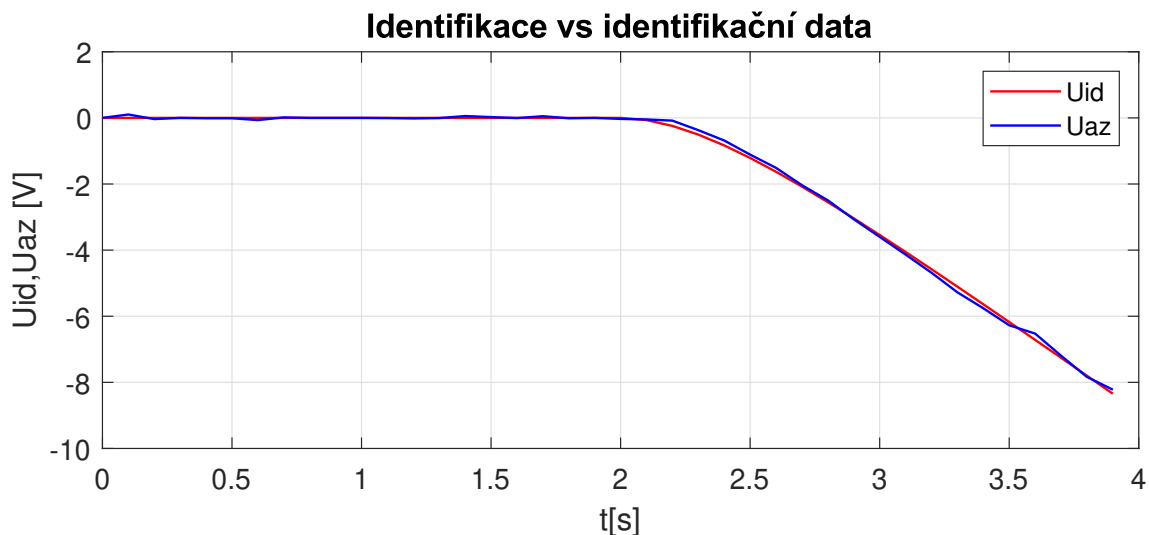
Obrázek 4.11: Vazba hlavní motor – azimut – měření dat

Z naměřených dat plyne, že se opět jedná o systém s astatickým charakterem. Struktura hledaného přenosu byla stejná jako u předchozího astatického systému, tedy:

$$G(s) = \frac{K}{a_n s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \dots + s} \quad (4.6)$$

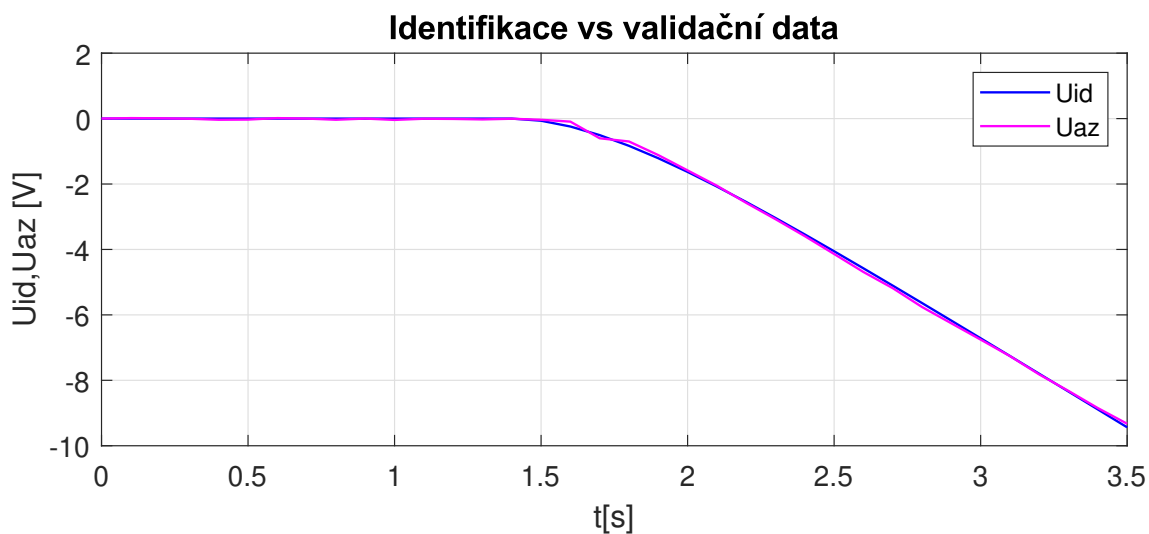
Byly vyzkoušeny i další struktury a s vyšším řádem astaticismu, které i v tomto případě nepřinesly přesnější shodu. Nalezený obrazový přenos má tvar:

$$G_{12}(s) = \frac{-0,7326}{0,3833s^2 + s} \quad (4.7)$$



Obrázek 4.12: Vazba hlavní motor – azimut – identifikace

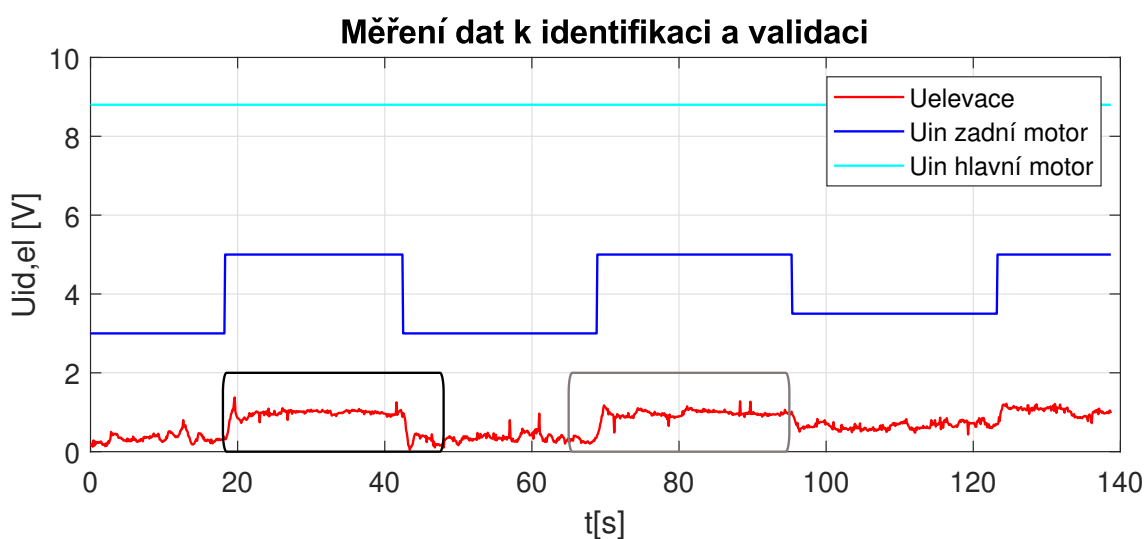
Kritérium  $J_{id}$  nabylo hodnoty 0,2507, v procentuálním vyjádření se jedná o shodu 97,02 %. Shoda modelu proti verifikačním datům byla 98,02 %. Kritérium  $J_{id}$  mělo hodnotu 0,1373.



Obrázek 4.13: Vazba hlavní motor – azimut – validace

## 4.4 Křížová vazba zadní motor – elevace

Při měření této křížové vazby byl zafixován pohyb v azimutu. Napětí na hlavním motoru bylo nastaveno na hodnotu pracovního bodu, tedy 8,8 V. Následně pak byly prováděny skoky na zadním motoru, které byly rozdílné jak ve velikosti, tak i délce trvání. Z grafu měření 4.14 byla opět vybrána data vhodná k identifikaci (černý výběr), respektive k validaci (šedý výběr).



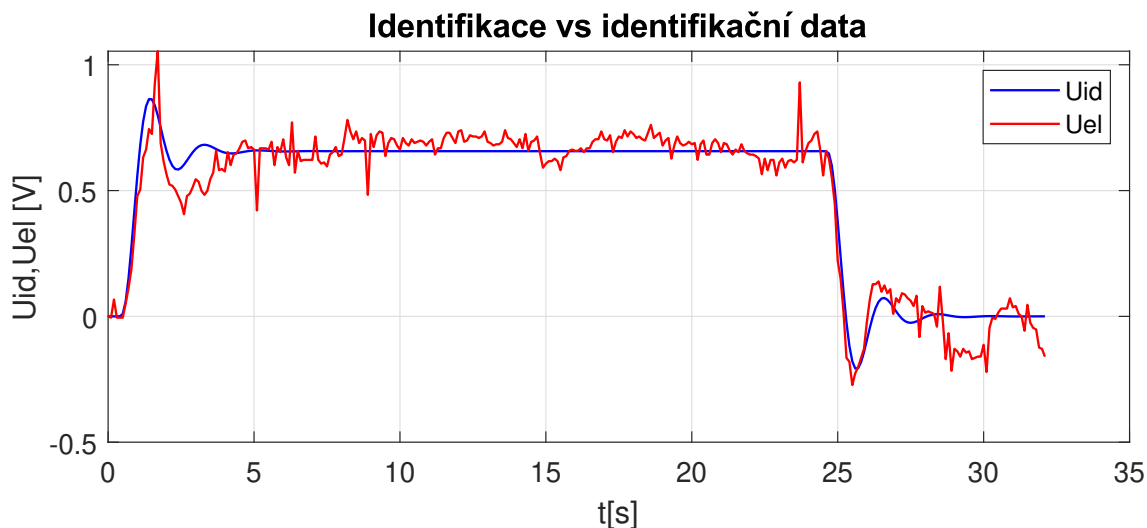
Obrázek 4.14: Vazba zadní motor – elevace – měření dat

Struktura hledaného přenosu byla:

$$G(s) = \frac{K}{a_n s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \dots + 1} \quad (4.8)$$

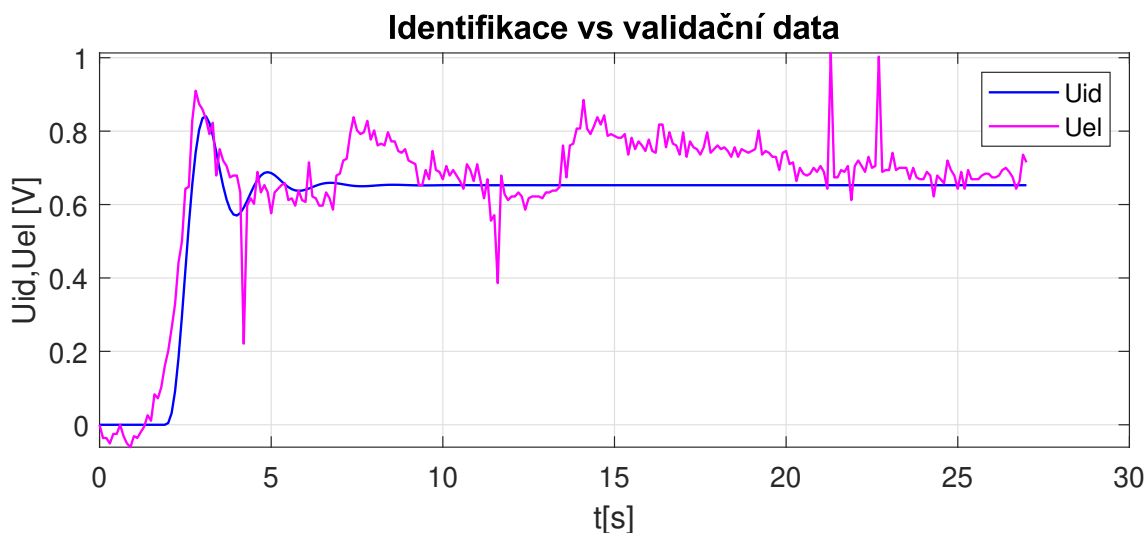
kde nejvěrnější shodu poskytoval systém třetího řádu. Hodnota kritéria  $J_{id}$  byla 1,8297, procentuální shoda 75,79 %. Nalezený přenos:

$$G_{21}(s) = \frac{0,3013}{0,0217s^3 + 0,1148s^2 + 0,4234s + 1} \quad (4.9)$$



Obrázek 4.15: Vazba zadní motor – elevace – identifikace

Shoda modelu proti verifikačním datům byla 56,53 %, kritérium  $J_{id}$  mělo hodnotu 2,1549.



Obrázek 4.16: Vazba zadní motor – elevace – validace

## 5 Popis a diskretizace nalezeného systému

Tento laboratorní model lze díky aretaci v kloubech také provozovat jako dva SISO systémy: hlavní motor – elevace nebo vedlejší motor – azimut. Nejprve tak byly tyto dva subsystémy diskretizovány samostatně a následně byl pro ně implementován MPC regulátor. Po úspěšném otestování řízení těchto dvou SISO systémů následovala diskretizace MIMO systému a nalezení MPC regulátoru, který by umožňoval komplexní řízení obou veličin.

### 5.1 Diskretizace SISO systémů

Pro realizaci vybraného typu prediktivního řízení je nutné přenosové funkce převést na stavový popis. V prostředí MATLAB je možné využít příkaz `tf2ss`, ovšem existují i jiné možnosti. Je také důležité si uvědomit, že převod do stavového prostoru není jednoznačný. Stavový popis pro systém hlavní vrtule – elevace:

$$\mathbf{A} = \begin{bmatrix} -4,415 & -12,19 & -26,72 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \mathbf{C} = [0 \ 0 \ 149,9] \quad (5.1)$$

$$\mathbf{D} = \mathbf{0}$$

Stavový popis pro systém zadní vrtule – azimut:

$$\mathbf{A} = \begin{bmatrix} -0,814 & -0,070 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \mathbf{C} = [0 \ 0 \ -3,756] \mathbf{D} = \mathbf{0} \quad (5.2)$$

Dalším nezbytným krokem je současné spojité stavové modely převést vhodnou periodou vzorkování na diskrétní. V MATLABu je možné využít příkaz `c2d`, který implicitně převádí spojité systém na diskrétní pomocí tvarovače 0. řádu. Při hledání vhodné vzorkovací periody je nutné splnit Nyquistovu podmínku, tedy že je nutné vzorkovat minimálně dvojnásobnou frekvencí, než je požadovaná šířka pásma. Pravidlo vzorkování dle [12] pak uvádí následující rovnici:

$$\frac{1}{T_s} = 30 \times \text{šířka pásma systému} \quad (5.3)$$

Na základě těchto poznatků byla stanovena vzorkovací perioda 10 ms. Jednotlivé matice diskrétního stavové popisu pro model hlavní motor – elevace mají následující hodnoty.

$$\mathbf{A} = \begin{bmatrix} 0,9562 & -0,1205 & -0,2614 \\ 0,00978 & 0,9994 & -0,0013 \\ 4,927e-5 & 0,00999 & 1 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} 0,00978 \\ 4,927e-5 \\ 1,648e-7 \end{bmatrix}; \quad (5.4)$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 149,8 \end{bmatrix}; \mathbf{D} = \mathbf{0}$$

Diskrétní stavový popis systému vedlejší motor – azimut:

$$\mathbf{A} = \begin{bmatrix} 0,9919 & -0,00069 & 0 \\ 0,009959 & 1 & 0 \\ 4,986e-5 & 0,01 & 1 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} 0,00995 \\ 4,986e-5 \\ 1,663e-7 \end{bmatrix}; \quad (5.5)$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & -3,756 \end{bmatrix}; \mathbf{D} = \mathbf{0}$$

Před realizací stavového MPC je vhodné, respektive nutné zkontrolovat pozorovatelnost a říditelnost jednotlivých soustav. V MATLABu je možné otestovat říditelnost příkazem `ctrb`, jejímž návratovým parametrem je matice:

$$\mathbf{Co} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A^2B} & \dots & \mathbf{A^{n-1}B} \end{bmatrix} \quad (5.6)$$

Matici pozorovatelnosti je možné získat příkazem `obsv`. Struktura je následující:

$$\mathbf{Ob} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \quad (5.7)$$

Aby byla soustava říditelná, respektive pozorovatelná, je nutné, aby hodnota matice  $\mathbf{Co}$  a  $\mathbf{Ob}$  byla rovna počtu stavů. V tomto případě jsou již systémy v minimální realizaci.

## 5.2 Diskretizace MIMO systému

V případě MIMO modelu lze výstup  $\mathbf{Y}(s)$  vyjádřit pomocí přenosové matice:

$$\mathbf{Y}(s) = \mathbf{G}(s) \cdot \mathbf{U}(s) \quad (5.8)$$

Přenosová matice pro model vrtulníku o dvou vstupech a dvou výstupech má následující strukturu:

$$\mathbf{G}(s) = \begin{bmatrix} G_{11}(s) & G_{21}(s) \\ G_{12}(s) & G_{22}(s) \end{bmatrix} \quad (5.9)$$

Vektor výstupu  $\mathbf{Y}(s)$  a vektor vstupu  $\mathbf{U}(s)$  musí mít taktéž odpovídající rozměr:

$$\mathbf{Y}(s) = \begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} \quad \mathbf{U}(s) = \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \quad (5.10)$$

Přenosová matice s nalezenými subsystemy má tedy podobu:

$$\mathbf{G}(s) = \begin{bmatrix} \frac{5,607}{0,03742s^3+0,1652s^2+0,4561s+1} & \frac{0,3013}{0,0217s^3+0,1148s^2+0,4234s+1} \\ \frac{-0,7326}{0,3833s^2+s} & \frac{-53,53}{14,25s^3+11,6s^2+s} \end{bmatrix} \quad (5.11)$$

K nalezení minimální realizace lze využít příkaz `minreal`, jehož návratová hodnota je stavový popis pouze s říditelnými a pozorovatelnými módy. Matice diskrétního stavového popisu pak mají následující tvar:

$$\begin{aligned}
 \mathbf{A} &= \begin{bmatrix} 0 & 0 & 0 & -2,804 & 0 & 0 & 0 & 0 & 0 & 2,856 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2,609 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2,136 & 2,342 & 0,509 & -2,1692 & 0 & 0 & 0 & 0 & 0,499 & 2,070 \\ 0 & 0 & 0 & 0 & -5,269 & -4,858 & 2,868 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0,814 & -0,070 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -2,173 & -2,382 & 0,499 & 2,207 & 0 & 0 & 0 & 0 & 0,492 & -2,246 \end{bmatrix} \\
 \mathbf{B} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ -2,804 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 2 \\ 0 & 0 \\ 2,853 & 0 \end{bmatrix} \\
 \mathbf{C} &= \begin{bmatrix} 0 & 4,683 & 0 & 0 & 0 & 0 & 0,950 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1,911 & 0 & 0 & 0 & 0 & 0 & -1,878 \end{bmatrix} \\
 \mathbf{D} &= \mathbf{0}
 \end{aligned} \tag{5.12}$$

Matice systému  $\mathbf{A}$  má v tomto případě rozměr  $10 \times 10$ , matice řízení  $\mathbf{B}$   $10 \times 2$  a výstupní matice  $\mathbf{C}$   $2 \times 10$ . Matice  $\mathbf{D}$  popisující přímou vazbu vstupů na výstupy je dle předpokladů nulová.



## 6 Historie prediktivního řízení

Vývoj moderních metod řízení a regulace sahá až k roku 1960, kdy se Rudolf E. Kalman začal zabírat hledáním optimálního řešení. Následně pak byl navržen lineární kvadratický regulátor (LQR), který optimalizuje podmínkami neomezenou kvadratickou penalizační funkci stavů a vstupů. LQR regulátory mají významné stabilizační účinky vzhledem k využití nekonečného horizontu. Na druhou stranu jeho vliv v průmyslu není tak významný, jelikož v jeho formulaci nejsou zohledněny nelinearity a omezení reálného systému.

Ve druhé polovině 70. let se začaly objevovat první verze prediktivního řízení v procesním průmyslu. Mezi nejzásadnější patřil Model Predictive Heuristic Control (MPHC) od J. Richaleta a Dynamic Matrix Control (DMC) od Cutlera a Remakera. MPHC byl později přejmenován na Model Algorithmic Control (MAC). Základem pro oba algoritmy bylo využití dynamického modelu procesu k predikování účinků budoucích akčních zásahů, které byly získány minimalizací predikční chyby vzhledem k zadaným omezením. Optimalizace je pak opakována každou vzorkovací periodou s aktuálními naměřenými vzorky. Počáteční verze MPC neměly automaticky stabilizující účinek, ovšem v případě stabilní soustavy a vhodně zvoleného horizontu je pak stability dosaženo. Softwarové řešení pro MPHC byla pojmenováno jako IDCOM, což je zkratka pro identifikaci a ovládání. Tento přístup využíval jako model impulzní odezvu a kvadratickou penalizační funkci na konečném horizontu predikce.

Algoritmus DMC byl představen v roce 1979. Zde je v roli modelu přechodová charakteristika. Počáteční verze algoritmů DMC a IDCOM se považují za první generaci technologie MPC. Druhou generaci představuje Quadratic Dynamic Matrix Control (QDMC) vyvinutý v roce 1983, který k řešení optimalizační úlohy

s omezeními využíval kvadratické programování. Tato omezení pak byla definována sadou lineárních nerovnic. Přestože QDMC poskytoval systematický přístup z hlediska začlenění vstupních a výstupních omezení, neexistoval zde jednoznačný postup v případě výskytu neexistujícího řešení.

Ve druhé polovině 80. let Shell přišel s Shell Multivariable Optimizing Controller (SMOC), který spojoval stavový prostor a prediktivní řízení. Model v podobě stavového popisu tak dokázal popsat široké množství dynamických procesů včetně poruchových jevů. V roli stavového pozorovatele vystupuje Kalmanův filtr, který estimuje stavové proměnné a neměřené poruchové veličiny. V roce 1998 došlo ke vzniku DMC+ kombinací algoritmů SMCA a DMC. Tato generace je čtvrtou a zároveň aktuální generací technologie MPC a již standardně obsahuje doplňky vylepšené technologie identifikace, víceúrovňová optimalizace apod. K této generaci se ještě řadí algoritmy Hierarchical Constraint Control (HEICON), RMPC od společnosti Honeywell a PCT od Profimatic. V roce 1995 vznikl díky kooperaci firem Honeywell a Profimatic algoritmus RMPCT, tedy Robust Model Predictive Control Technology. [13]

## 6.1 MPC v průmyslu

V současnosti existuje několik společností, které se zabývají komerční výrobou a vývojem prediktivních regulátorů. Zde je seznam těch nejvýznamnějších společností a jejich hlavních produktů:

- DMC corp. : Dynamic Matrix Control (DMC)
- Adezva : PFC, IDCOM
- Honeywell Profimatic : RMPCT, PCT
- Setpoint Inc : SMCA, IDCOM-M
- Treiber Controls : OPC
- SCAP : APCS

Největší podíl na trhu drží DMC corp. s 26 % aplikací, o zbytek se dělí ostatní společnosti přibližně stejným dílem. Většina aplikací je v oblasti rafinerií, jelikož se jedná o původní pole působnosti těchto regulátorů. Dalším významným zastupitelem je oblast petrochemického a chemického průmyslu. Ostatní oblasti, kde se lze s prediktivní regulací setkat, je především potravinářský průmysl, automotive, letecký a kosmický průmysl. [14]

## 7 Prediktivní řízení založené na stavovém popisu

K formulaci prediktivního zákona řízení je nutné znát matematický popis soustavy, jenž je reprezentován stavovým popisem.

$$\begin{aligned}\mathbf{x}_m(k+1) &= \mathbf{A}_m \mathbf{x}_m(k) + \mathbf{B}_m \mathbf{u}(k) \\ \mathbf{y}_m(k+1) &= \mathbf{C}_m \mathbf{x}_m(k)\end{aligned}\tag{7.1}$$

v obecné podobě má stavový popis přímou vazbu mezi vstupním signálem  $\mathbf{u}(k)$  a výstupním signálem  $\mathbf{y}(k)$

$$\mathbf{y}_m(k) = \mathbf{C}_m \mathbf{x}_m(k) + \mathbf{D}_m \mathbf{u}(k)\tag{7.2}$$

Nicméně je možné předpokládat, že vstup  $\mathbf{u}(k)$  nemůže ovlivnit výstup  $\mathbf{y}(k)$  ve stejném okamžiku, tudíž je možné považovat  $\mathbf{D}=\mathbf{0}$ .

### 7.1 Odvození rozšířeného modelu soustavy

$$\begin{aligned}\mathbf{x}_m(k+1) - \mathbf{x}_m(k) &= \mathbf{A}_m[\mathbf{x}_m(k) - \mathbf{x}_m(k-1)] + \mathbf{B}_m[\mathbf{u}(k) - \mathbf{u}(k-1)] \\ \Delta \mathbf{x}_m(k+1) &= \mathbf{x}_m(k+1) - \mathbf{x}_m(k) \\ \Delta \mathbf{x}_m(k) &= \mathbf{x}_m(k) - \mathbf{x}_m(k-1) \\ \Delta \mathbf{u}(k) &= \mathbf{u}(k) - \mathbf{u}(k-1)\end{aligned}\tag{7.3}$$

Z výrazů 7.3 je pak možné určit následující rovnici:

$$\Delta \mathbf{x}_m(k+1) = \mathbf{A}_m \Delta \mathbf{x}_m(k) + \mathbf{B}_m \Delta \mathbf{u}(k)\tag{7.4}$$

Nový stavový vektor je následně definován:

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}_m(k)^T & \mathbf{y}(k) \end{bmatrix}^T \quad (7.5)$$

$$\begin{aligned} \mathbf{y}(k+1) - \mathbf{y}(k) &= \mathbf{C}_m[\mathbf{x}_m(k+1) - \mathbf{x}_m(k)] = \mathbf{C}_m\Delta\mathbf{x}_m(k+1) = \\ &= \mathbf{C}_m\mathbf{A}_m\Delta\mathbf{x}_m(k) + \mathbf{C}_m\mathbf{B}_m\Delta\mathbf{u}(k) \end{aligned} \quad (7.6)$$

$$\underbrace{\begin{bmatrix} \Delta\mathbf{x}_m(k+1) \\ \mathbf{y}(k+1) \end{bmatrix}}_{\mathbf{x}(k+1)} = \underbrace{\begin{bmatrix} \mathbf{A}_m & \mathbf{O}_m^T \\ \mathbf{C}_m\mathbf{A}_m & \mathbf{1} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{x}_m(k) \\ \mathbf{y}(k) \end{bmatrix}}_{\mathbf{x}(k)} + \underbrace{\begin{bmatrix} \mathbf{B}_m \\ \mathbf{C}_m\mathbf{B}_m \end{bmatrix}}_{\mathbf{B}} \Delta\mathbf{u}(k) \quad (7.7)$$

$$\mathbf{y}(k) = \underbrace{\begin{bmatrix} \mathbf{O}_m & \mathbf{1} \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} \Delta\mathbf{x}_m(k) \\ \mathbf{y}(k) \end{bmatrix}}_{\mathbf{x}(k)} \quad (7.8)$$

Matice  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  tvoří takzvaný rozšířený model soustavy, který je dále využit při návrhu prediktivního regulátoru. Dalším krokem je výpočet predikovaného výstupu soustavy na základě budoucího řídicího signálu. [15]

## 7.2 Predikce stavových a výstupních proměnných

Budoucí řídicí trajektorie je označena:

$$\Delta\mathbf{u}(k_i), \Delta\mathbf{u}(k_i + 1), \dots, \Delta\mathbf{u}(k_i + N_c - 1) \quad (7.9)$$

kde  $N_c$  je horizont řízení, který určuje počet parametrů budoucí řídicí trajektorie. S dostupnou informací  $\mathbf{x}(k_i)$  jsou budoucí stavové proměnné predikované po  $N_p$  vzorků, kde  $N_p$  označuje horizont predikce.  $N_p$  je dále také velikost optimalizačního okna. Budoucí stavové proměnné lze označit:

$$\mathbf{x}(k_i + 1|k_i), \mathbf{x}(k_i + 2|k_i), \dots, \mathbf{x}(k_i + m|k_i), \dots, \mathbf{x}(k_i + N_p|k_i) \quad (7.10)$$

kde  $(k_i + m|k_i)$  je predikovaná hodnota stavové proměnné v čase  $k_i + m$  na základě stavu v čase  $k_i$ . Dle rozšířeného stavového popisu je možné budoucí stavové proměnné

postupně vypočítat.

$$\begin{aligned}
\mathbf{x}(k_i + 1|k_i) &= \mathbf{A}\mathbf{x}(k_i) + \mathbf{B}\Delta\mathbf{u}(k_i) \\
\mathbf{x}(k_i + 2|k_i) &= \mathbf{A}\mathbf{x}(k_i + 1) + \mathbf{B}\Delta\mathbf{u}(k_i + 1) \\
&\vdots \\
\mathbf{x}(k_i + N_p|k_i) &= \mathbf{A}^{N_p}\mathbf{x}(k_i) + \mathbf{A}^{N_p-1}\mathbf{B}\Delta\mathbf{u}(k_i) + \mathbf{A}^{N_p-2}\mathbf{B}\Delta\mathbf{u}(k_i + 1) \\
&+ \dots + \mathbf{A}^{N_p-N_c}\mathbf{B}\Delta\mathbf{u}(k_i + N_c - 1)
\end{aligned} \tag{7.11}$$

Predikovaný výstup lze po substituci zapsat:

$$\begin{aligned}
\mathbf{y}(k_i + 1|k_i) &= \mathbf{C}\mathbf{A}\mathbf{x}(k_i) + \mathbf{C}\mathbf{B}\Delta\mathbf{u}(k_i) \\
\mathbf{y}(k_i + 2|k_i) &= \mathbf{C}\mathbf{A}^2\mathbf{x}(k_i) + \mathbf{C}\mathbf{A}\mathbf{B}\Delta\mathbf{u}(k_i) + \mathbf{C}\mathbf{B}\Delta\mathbf{u}(k_i + 1) \\
\mathbf{y}(k_i + 3|k_i) &= \mathbf{C}\mathbf{A}^3\mathbf{x}(k_i) + \mathbf{C}\mathbf{A}^2\mathbf{B}\Delta\mathbf{u}(k_i) + \mathbf{C}\mathbf{A}\mathbf{B}\Delta\mathbf{u}(k_i + 1) + \mathbf{C}\mathbf{B}\Delta\mathbf{u}(k_i + 2) \\
&\vdots \\
\mathbf{y}(k_i + N_p|k_i) &= \mathbf{C}\mathbf{A}^{N_p}\mathbf{x}(k_i) + \mathbf{C}\mathbf{A}^{N_p-1}\mathbf{B}\Delta\mathbf{u}(k_i) + \mathbf{C}\mathbf{A}^{N_p-2}\mathbf{B}\Delta\mathbf{u}(k_i + 1) \\
&+ \dots + \mathbf{C}\mathbf{A}^{N_p-N_c}\mathbf{B}\Delta\mathbf{u}(k_i + N_c - 1)
\end{aligned} \tag{7.12}$$

Následně lze definovat vektory:

$$\mathbf{Y} = [\mathbf{y}(k_i + 1|k_i)^T, \mathbf{y}(k_i + 2|k_i)^T, \mathbf{y}(k_i + 3|k_i)^T, \dots, \mathbf{y}(k_i + N_p|k_i)^T]^T \tag{7.13}$$

$$\Delta\mathbf{U} = [\Delta\mathbf{u}(k_i)^T, \Delta\mathbf{u}(k_i + 1)^T, \Delta\mathbf{u}(k_i + 2)^T, \dots, \Delta\mathbf{u}(k_i + N_c - 1)^T]^T \tag{7.14}$$

V případě systému s jedním vstupem a výstupem je délka vektoru  $\mathbf{Y}$  rovna  $N_p$  a délka vektoru  $\Delta\mathbf{U}$  je  $N_c$ . U MIMO soustavy je předpokladem, že systém má  $m$  vstupů,  $q$  výstupů a  $n_1$  stavů, a dále také počet výstupů je menší nebo roven počtu vstupů, tedy  $q \leq m$ . Pokud by byl počet výstupů větší než počet vstupů, nelze předpokládat, že bude možné ovládat jednotlivé výstupy nezávisle s nulovou regulační odchylkou. Rovnice 7.11 a 7.12 lze vyjádřit v kompaktní formě. [15]

$$\mathbf{Y} = \mathbf{F}\mathbf{x}(k_i) + \Phi\Delta\mathbf{U} \tag{7.15}$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \\ \vdots \\ \mathbf{CA}^{N_p} \end{bmatrix} \quad \Phi = \begin{bmatrix} \mathbf{CB} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CA}^2\mathbf{B} & \mathbf{CAB} & \mathbf{CB} & \dots & \mathbf{0} \\ \vdots & & & & \\ \mathbf{CA}^{N_p-1}\mathbf{B} & \mathbf{CA}^{N_p-2}\mathbf{B} & \mathbf{CA}^{N_p-3}\mathbf{B} & \dots & \mathbf{CA}^{N_p-N_c}\mathbf{B} \end{bmatrix} \quad (7.16)$$

### 7.3 Optimalizace

Cílem prediktivního řízení je přiblížit predikovaný výstup co nejlíže referenčnímu signálu, který předpokládáme konstantní po celou délku optimalizačního horizontu. Tento přístup je pak využit při výpočtu optimálního  $\Delta\mathbf{U}$ , jelikož je minimalizována chybová funkce mezi žádanou hodnotou a predikovaným výstupem.

$$\mathbf{R}_s^T = \underbrace{\begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}}_{N_p} \mathbf{r}(k_i) = \overline{\mathbf{R}}_s \mathbf{r}(k_i) \quad (7.17)$$

Penalizační funkce:

$$J = (\mathbf{R}_s - \mathbf{Y})^T (\mathbf{R}_s - \mathbf{Y}) + \Delta\mathbf{U}^T \overline{\mathbf{R}} \Delta\mathbf{U} \quad (7.18)$$

První člen výrazu slouží k minimalizaci chyby mezi predikovaným výstupem a referenčním signálem. Druhý člen pak zohledňuje velikost  $\Delta\mathbf{U}$ .  $\overline{\mathbf{R}}$  je diagonální matice ve tvaru  $\overline{\mathbf{R}} = \mathbf{r}_w \mathbf{I}_{N_c \times N_c}$ , kde  $\mathbf{r}_w > \mathbf{0}$  je použito jako ladící parametr. V případě, kdy je  $\mathbf{r}_w = \mathbf{0}$ , lze pak penalizační funkci interpretovat, že nechceme brát v potaz, jak velké  $\Delta\mathbf{U}$  může být. V opačném případě, tedy kdy  $\mathbf{r}_w$  bude vysoké, penalizační funkce představuje situaci, kdy bereme ohledy na velikost  $\Delta\mathbf{U}$ . K nalezení optimálního  $\Delta\mathbf{U}$ , které minimalizuje kritérium  $J$  s využitím vztahu 7.15 lze vyjádřit:

$$J = (\mathbf{R}_s - \mathbf{F}\mathbf{x}(k_i))^T (\mathbf{R}_s - \mathbf{F}\mathbf{x}(k_i)) - 2\Delta\mathbf{U}^T \Phi^T (\mathbf{R}_s - \mathbf{F}\mathbf{x}(k_i)) + \Delta\mathbf{U}^T (\Phi^T \Phi + \overline{\mathbf{R}}) \Delta\mathbf{U} \quad (7.19)$$

První derivace penalizační funkce:

$$\frac{\partial J}{\partial \Delta\mathbf{U}} = -2\Phi^T (\mathbf{R}_s - \mathbf{F}\mathbf{x}(k_i)) + 2(\Phi^T \Phi + \overline{\mathbf{R}}) \Delta\mathbf{U} \quad (7.20)$$

Analytické minimum lze získat:

$$\frac{\partial J}{\partial \Delta \mathbf{U}} = 0 \quad (7.21)$$

Optimální řídicí signál je pak dán:

$$\Delta \mathbf{U} = (\Phi^T \Phi + \bar{\mathbf{R}})^{-1} \Phi^T (\mathbf{R}_s - \mathbf{F} \mathbf{x}(k_i)) \quad (7.22)$$

za předpokladu, že existuje inverze členu  $(\Phi^T \Phi + \bar{\mathbf{R}})$ , který je v rámci optimalizace nazýván Hessova matice. Jak již bylo odvozeno, tak vektor optimálních parametrů  $\Delta \mathbf{U}$  v čase  $k_i$  je nalezen řešením rovnice 7.22, kde člen  $(\Phi^T \Phi + \bar{\mathbf{R}})^{-1} \Phi^T \mathbf{R}_s$  odpovídá změně žádané veličiny, zatímco člen  $-(\Phi^T \Phi + \bar{\mathbf{R}})^{-1} \Phi^T \mathbf{F}$  odpovídá stavové zpětné vazbě. Přestože vektor  $\Delta \mathbf{U}$  obsahuje členy  $\Delta \mathbf{u}(k_i)$ ,  $\Delta \mathbf{u}(k_i + 1)$ ,  $\Delta \mathbf{u}(k_i + 2)$ , ... tak vzhledem k použití principu ustupujícího horizontu je využit pouze první člen, zatímco ostatní jsou ignorovány. Následující periodu vzorkování je sestaven nový stavový vektor pro výpočet další sekvence řídicího signálu. Tento opakující se proces je pak nazván jako ustupující horizont.

$\Phi^T \Phi$  má rozměr  $mN_c \times mN_c$ ,  $\Phi^T \mathbf{F}$  má rozměr  $mN_c \times n$  a součin  $\Phi^T \bar{\mathbf{R}}_s$  je roven posledním  $q$  sloupcům  $\Phi^T \mathbf{F}$ . Váhová matice  $\bar{\mathbf{R}}$  je bloková matice s  $m$  bloky a její rozměr je stejný jako  $\Phi^T \Phi$ . Referenční signál je dán :

$$\mathbf{r}(k_i) = [r_1(k_i), r_2(k_i), \dots, r_q(k_i)] \quad (7.23)$$

a je tvořen  $q$  vstupními signály. Na základě principu ustupujícího horizontu je k optimálnímu řídicímu signálu použito prvních  $m$  členů  $\Delta \mathbf{U}$ .

$$\begin{aligned} \Delta \mathbf{u}(k_i) &= \begin{bmatrix} \mathbf{1}_m & \mathbf{0}_m & \dots & \mathbf{0}_m \end{bmatrix} (\Phi^T \Phi + \bar{\mathbf{R}})^{-1} [\Phi^T \bar{\mathbf{R}}_s \mathbf{r}(k_i) - \Phi^T \mathbf{F} \mathbf{x}(k_i)] = \\ &= \mathbf{K}_y \mathbf{r}(k_i) - \mathbf{K}_{\text{mpc}} \mathbf{x}(k_i) \end{aligned} \quad (7.24)$$

$\mathbf{1}_m$  a  $\mathbf{0}_m$  jsou jednotkové, respektive nulové matice o rozměru  $m \times m$ , kde  $\mathbf{K}_y$  je první člen výrazu  $(\Phi^T \Phi + \bar{\mathbf{R}})^{-1} \Phi^T \bar{\mathbf{R}}_s$  a  $\mathbf{K}_{\text{mpc}}$  je první řádek výrazu  $(\Phi^T \Phi + \bar{\mathbf{R}})^{-1} \Phi^T \mathbf{F}$ .  
[15]



## 7.4 Prediktivní řízení se stavovou estimací

Při návrhu prediktivního regulátoru je předpokládáno, že informace  $\mathbf{x}(\mathbf{k}_i)$  je dostupná v čase  $k_i$ , z čehož plyne, že všechny stavové proměnné jsou měřitelné. Ve většině reálných případů tomu tak není, ať už technologických či fyzikálních důvodů. Tento problém odstraňuje stavový pozorovatel, který na základě naměřených dat dokáže estimovat stavové proměnné. Pozorovatel navíc může pracovat jako filtr šumu. Pozorovatel je založen na matematickém modelu soustavy.

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (7.25)$$

Následně je možné tento model využít k estimaci stavových proměnných  $\hat{\mathbf{x}}(k)$ .

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) \quad (7.26)$$

Tento princip by byl funkční pouze v případě stabilní soustavy a správného odhadu počátečních podmínek. Chyba estimace  $\tilde{\mathbf{x}}(k)$  je dána:

$$\begin{aligned} \tilde{\mathbf{x}}(k) &= \mathbf{x}(k) - \hat{\mathbf{x}}(k) \\ \tilde{\mathbf{x}}(k+1) &= \mathbf{A}[\mathbf{x}(k) - \hat{\mathbf{x}}(k)] = \mathbf{A}\tilde{\mathbf{x}}(k) \end{aligned} \quad (7.27)$$

Pokud  $\mathbf{A}$  má vlastní čísla uvnitř jednotkové kružnice, pak  $\|\tilde{\mathbf{x}}(k)\| \rightarrow 0$  pokud  $k \rightarrow \infty$ , což znamená, že estimovaná hodnota  $\tilde{\mathbf{x}}(k)$  konverguje k  $\mathbf{x}(k)$ . Naopak, pokud  $\mathbf{A}$  má jedno či více vlastních čísel mimo jednotkovou kružnici, jedná se o nestabilní systém pak  $\|\tilde{\mathbf{x}}(k)\| \rightarrow \infty$  pokud  $k \rightarrow \infty$ , což znamená, že estimovaná hodnota nekonverguje k  $\mathbf{x}(k)$ . Dalším problémem takto zavedeného estimátoru je u stabilních soustav rychlost konvergence  $\|\tilde{\mathbf{x}}(k)\| \rightarrow 0$ , která je dána polohou pólů soustavy. Například pokud jsou póly soustavy blízko počátku komplexní roviny, tak pak chyba  $\|\tilde{\mathbf{x}}(k)\|$  konverguje rychleji k nule než v případě vzdálenějších pólů. Zlepšit estimaci lze zavedením korekčního členu a zpětné vazby. Rovnici lze pak zapsat:

$$\hat{\mathbf{x}}(k+1) = \underbrace{\mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k)}_{\text{Model}} + \underbrace{\mathbf{K}_{\text{ob}}[\mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}(k)]}_{\text{Korekce}} \quad (7.28)$$

kde  $\mathbf{K}_{ob}$  je zesílení pozorovatele a jeho volbou lze ovlivnit rychlost konvergence. Matice  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  pocházejí z rozšířeného modelu soustavy. Kritérium k minimalizaci je pak modifikováno do podoby:

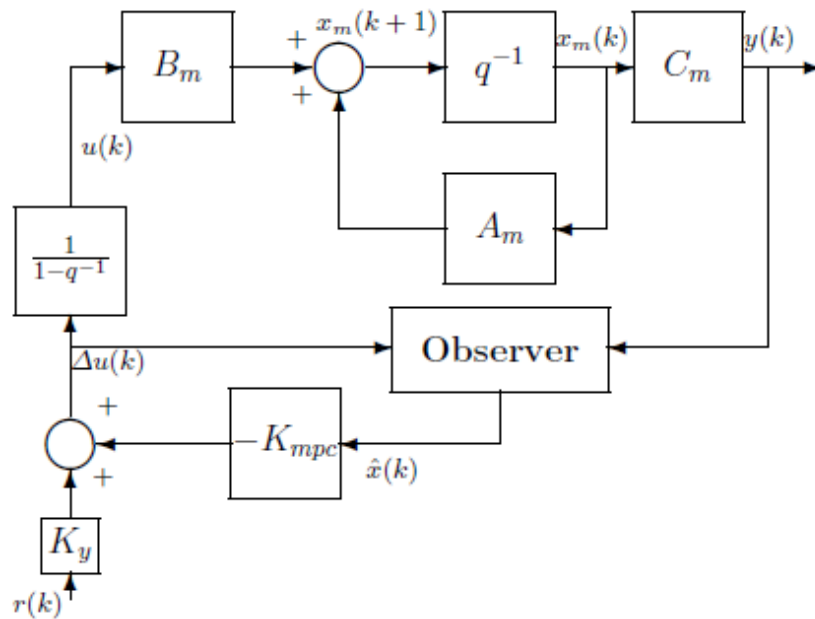
$$J = (\mathbf{R}_s - \mathbf{F}\hat{\mathbf{x}}(k_i))^T (\mathbf{R}_s - \mathbf{F}\hat{\mathbf{x}}(k_i)) - 2\Delta\mathbf{U}^T \Phi^T (\mathbf{R}_s - \mathbf{F}\hat{\mathbf{x}}(k_i)) + \Delta\mathbf{U}^T (\Phi^T \Phi + \bar{\mathbf{R}}) \Delta\mathbf{U} \quad (7.29)$$

Optimální řešení  $\Delta\mathbf{U}$  je ve tvaru:

$$\Delta\mathbf{U} = (\Phi^T \Phi + \bar{\mathbf{R}})^{-1} \Phi^T (\mathbf{R}_s - \mathbf{F}\hat{\mathbf{x}}(k_i)) \quad (7.30)$$

Aplikace principu ustupujícího horizontu vede dle [15] na optimální řešení  $\Delta\mathbf{u}$ :

$$\Delta\mathbf{u}(k_i) = \mathbf{K}_y \mathbf{r}(k_i) - \mathbf{K}_{mpc} \mathbf{x}(k_i) \quad (7.31)$$



Obrázek 7.1: Blokové schéma prediktivní regulace s pozorovatelem [15]

## 7.5 Prediktivní řízení s omezeními

V této kapitole je představena oblast prediktivního řízení, kde se vyskytují omezení. Dále jsou popsány druhy omezení a jejich vliv na kvalitu řízení a na následné řešení optimalizační úlohy pomocí kvadratického kritéria. V praxi se lze nejběžněji

setkat s omezeními vyskytujícími se na akční veličině, regulované veličině a v případě stavového popisu i na stavové proměnné.

### 1. Omezení na velikosti akčního zásahu

- Jedná se o jedno z nejpoužívanějších omezení, vyskytuje se totiž reálně na každém fyzickém akčním členu. Příkladem může být ventil, který nelze otevřít na více než 100 %. Tento typ je zástupcem tzv. tvrdého omezení.

Lze ho zapsat:

$$u^{\min} \leq u(k) \leq u^{\max} \quad (7.32)$$

### 2. Omezení na přírůstku akční veličiny

- Jedná se o další tvrdé omezení na akční veličině. V tomto případě je omezena rychlost změn. To lze aplikovat například v případech, kdy se ventil nedokáže otevírat rychleji, než je maximální rychlost daná výrobcem. Lze ho zapsat:

$$\Delta u^{\min} \leq \Delta u(k) \leq \Delta u^{\max} \quad (7.33)$$

### 3. Omezení na velikosti výstupní veličiny

- Tímto omezením lze stanovit pracovní rozsah systému. Lze ho popsat:

$$y^{\min} \leq y(k) \leq y^{\max} \quad (7.34)$$

Patří do skupiny měkkých omezení, je tak možné využít pomocnou proměnnou  $s_v > 0$  tak, aby platil vzorec:

$$y^{\min} - s_v \leq y(k) \leq y^{\max} + s_v \quad (7.35)$$

Stejný princip lze použít i pro omezení stavových proměnných. V tomto případě dojde k uvolnění omezení na výstupní veličině, omezení na akční veličině mohou být zachována a úloha nalezení řídicího signálu má řešení. Tento nalezený signál se pak co nejvíce blíží k zadaným omezením.

V případě vícerozměrových systémů se definují omezení pro každý vstup samostatně. U horních limitů přírůstků akční veličiny:

$$[u_1^{\max}, u_2^{\max}, \dots, u_m^{\max}] \quad (7.36)$$

obdobně u dolních limitů:

$$[u_1^{\min}, u_2^{\min}, \dots, u_m^{\min}] \quad (7.37)$$

kde každé omezení je definováno:

$$\begin{aligned} \Delta u_1^{\min} &\leq \Delta u_1(k) \leq \Delta u_1^{\max} \\ \Delta u_2^{\min} &\leq \Delta u_2(k) \leq \Delta u_2^{\max} \\ &\vdots \\ \Delta u_m^{\min} &\leq \Delta u_m(k) \leq \Delta u_m^{\max} \end{aligned} \quad (7.38)$$

Dalším krokem je zavedené omezení ve formě lineárních nerovností ztotožnit se stávajícím výpočtem  $\Delta U$ . Jelikož k výpočtu optimálního řešení bude využito kvadratické programování, je nutné omezení rozdělit na dvě samostatné části, jednu týkající se dolních limitů a druhou horních limitů. V případě omezení na velikosti přírůstků akčních zásahů:

$$\Delta U^{\min} \leq \Delta U \leq \Delta U^{\max} \quad (7.39)$$

po separaci:

$$\begin{aligned} -\Delta U &\leq \Delta U^{\min} \\ \Delta U &\leq \Delta U^{\max} \end{aligned} \quad (7.40)$$

Kde  $\Delta U^{\min}$  a  $\Delta U^{\max}$  jsou sloupcové vektory s  $N_c$  prvky  $\Delta u^{\min}$  a  $\Delta u^{\max}$ . Obdobným způsobem lze definovat i ostatní omezení včetně výstupního a omezení na velikosti akčního zásahu. Obvykle jsou omezení zavedena na všechny budoucí vzorky a vyjádřena s využitím vektoru  $\Delta U$ . Omezení na velikosti akčního zásahu lze

vyjádřit:

$$\begin{bmatrix} u(k_i) \\ u(k_i + 1) \\ u(k_i + 2) \\ \vdots \\ u(k_i + N_c - 1) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} u(k_i - 1) + \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 1 & \dots & 0 \\ \vdots & & & \ddots & \\ 1 & 1 & \dots & 1 & 1 \end{bmatrix} \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i + 1) \\ \Delta u(k_i + 2) \\ \vdots \\ \Delta u(k_i + N_c - 1) \end{bmatrix} \quad (7.41)$$

Tuto rovnici lze vyjádřit v kompaktní formě využívající člen  $\Delta \mathbf{U}$ :

$$\begin{aligned} -(C_1 u(k_i - 1) + C_2 \Delta \mathbf{U}) &\leq -\mathbf{U}^{\min} \\ (C_1 u(k_i - 1) + C_2 \Delta \mathbf{U}) &\leq -\mathbf{U}^{\max} \end{aligned} \quad (7.42)$$

Kde  $\mathbf{U}^{\min}$  a  $\mathbf{U}^{\max}$  jsou opět sloupcové vektory s  $N_c$  prvky obsahující  $u^{\min}$  respektive  $u^{\max}$ . Stejně se definují limity pro velikost akčních zásahů, a pokud je to požadováno, tak i pro výstupní a stavové veličiny.

Výstupní omezení lze s využitím  $\Delta \mathbf{U}$  vyjádřit následovně:

$$\mathbf{Y}^{\min} \leq \mathbf{F}\mathbf{x}(k_i) + \Phi \Delta \mathbf{U} \leq \mathbf{Y}^{\max} \quad (7.43)$$

Následně je pak možné definovat prediktivní řídicí zákon s přítomností tvrdých omezení, který minimalizuje kritérium  $J$ :

$$J = (\mathbf{R}_s - \mathbf{F}\mathbf{x}(k_i))^T (\mathbf{R}_s - \mathbf{F}\mathbf{x}(k_i)) - 2\Delta \mathbf{U}^T \Phi^T (\mathbf{R}_s - \mathbf{F}\mathbf{x}(k_i)) + \Delta \mathbf{U}^T (\Phi^T \Phi + \bar{\mathbf{R}}) \Delta \mathbf{U} \quad (7.44)$$

a podléhá následujícím omezením:

$$\begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \mathbf{M}_3 \end{bmatrix} \Delta \mathbf{U} \leq \begin{bmatrix} \mathbf{N}_1 \\ \mathbf{N}_2 \\ \mathbf{N}_3 \end{bmatrix} \quad (7.45)$$

kde:

$$\begin{aligned} \mathbf{M}_1 &= \begin{bmatrix} -\mathbf{C}_2 \\ \mathbf{C}_2 \end{bmatrix}; \mathbf{N}_1 = \begin{bmatrix} -\mathbf{U}^{\min} + \mathbf{C}_1 \mathbf{u}(k_i - 1) \\ -\mathbf{U}^{\max} - \mathbf{C}_1 \mathbf{u}(k_i - 1) \end{bmatrix}; \mathbf{M}_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}; \\ \mathbf{N}_2 &= \begin{bmatrix} -\Delta \mathbf{U}^{\min} \\ -\Delta \mathbf{U}^{\max} \end{bmatrix}; \mathbf{M}_3 = \begin{bmatrix} -\Phi \\ \Phi \end{bmatrix}; \mathbf{N}_3 = \begin{bmatrix} -\mathbf{Y}^{\min} + \mathbf{F}\mathbf{x}(k_i) \\ \mathbf{Y}^{\max} - \mathbf{F}\mathbf{x}(k_i) \end{bmatrix} \end{aligned} \quad (7.46)$$

Výraz 7.45 lze zapsat do tvaru:

$$\mathbf{M}\Delta\mathbf{U} \leq \gamma \quad (7.47)$$

Jelikož penalizační funkce  $J$  je kvadratická a omezení ve formě lineárních nerovnic, úloha nalezení optimálního řídicího signálu se stává standardní problematikou kvadratického programování. [15] Obvykle je v optimalizaci rozhodovací proměnná označena  $x$ , penalizační funkce a omezení pak mají následující podobu:

$$\begin{aligned} J &= \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ \mathbf{M}\mathbf{x} &\leq \gamma \end{aligned} \quad (7.48)$$

Nyní je možné ztotožnit proměnné  $\mathbf{Q}$ , respektive  $\mathbf{c}$  z výrazu 7.44.

$$\begin{aligned} \mathbf{Q} &= 2(\Phi^T\Phi + \bar{\mathbf{R}}) \\ \mathbf{c} &= 2\Phi^T(\mathbf{F}\mathbf{x} - \mathbf{R}_s) \end{aligned} \quad (7.49)$$

Zbývající člen 7.50 je konstantní a je možné ho ignorovat v průběhu optimalizace.

$$(\mathbf{R}_s - \mathbf{F}\mathbf{x})^T(\mathbf{R}_s - \mathbf{F}\mathbf{x}) \quad (7.50)$$

## 8 Návrh MPC pro SISO systém hlavní vrtule

Tato kapitola se věnuje návrhu prediktivního regulátoru pro systém s jedním vstupem a jedním výstupem, konkrétněji pro subsystém hlavní vrtule – elevace. Prediktivní regulátor byl navržen dle předcházející kapitoly, dále musel být využit stavový pozorovatel, jelikož se zde vyskytují stavy, které vzhledem ke konstrukci zařízení nelze v současné konfiguraci měřit. Z přechodové charakteristiky tohoto subsystému byl určen horizont predikce. Vhodná perioda vzorkování byla odvozena ze vztahu 5.3 v předchozích kapitolách. Pomocí  $\bar{\mathbf{R}}$  lze pak měnit odezvu prediktivního regulátoru, konkrétněji v penalizační funkci zohledňuje velikost akčních zásahů. V obou variantách regulátoru byla testována šířka pracovního rozsahu, tedy pro jak velkou žádanou hodnotu elevace je tento systém ještě stabilní. Dále byla vyzkoušena odezva na sérii skokových vstupních signálů a vyhodnocena kvalita regulace. Na závěr bylo otestována odolnost regulátoru proti poruchám, které byly generovány vlivem zadního motoru.

### 8.1 Varianta s analytickým výpočtem

Tato varianta regulátoru využívá pro výpočet optimálního vektoru  $\Delta\mathbf{U}$  vztah 7.30. Tudíž tato varianta při svém výpočtu nebere v potaz omezení, která se na systému mohou vyskytovat. Vlastnosti těchto omezení lze však následně definovat pomocí příkazů `if-else`. Z hlediska přístupu lze pak tento postup považován za suboptimální. Tedy nejprve je vypočten signál  $\Delta\mathbf{U}$  a až následně je ohodnocena jeho validita vzhledem k zadaným omezením a případně je pak upraven.

Jedno z hlavních omezení, které se na této soustavě vyskytuje, je omezení na velikosti akční veličiny, jelikož model vrtulníku pracuje v jeho těsné blízkosti. V tabulce 8.1 jsou uvedeny parametry MPC regulátoru.

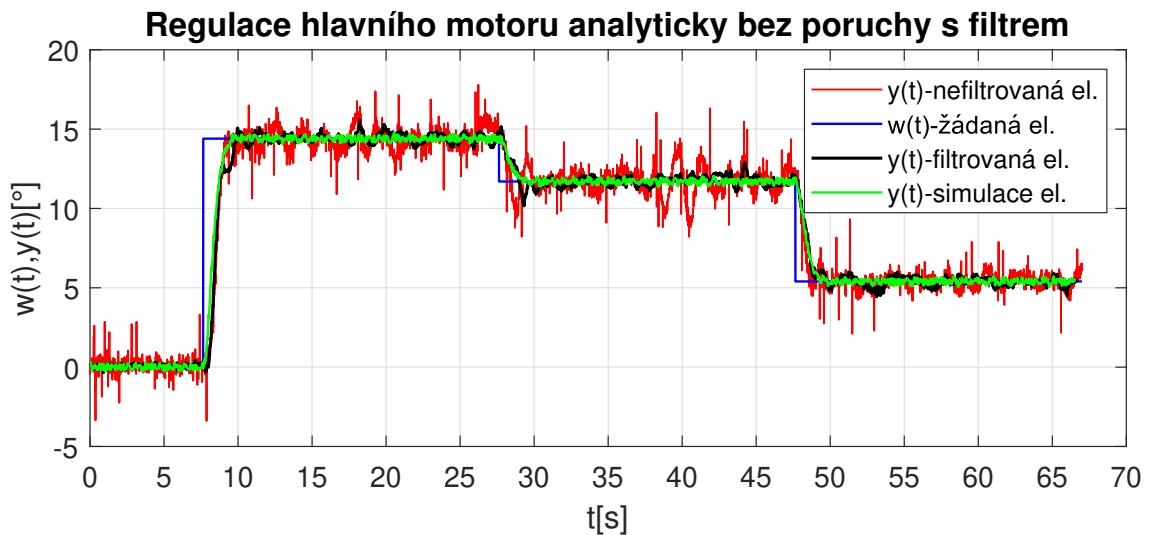
Perioda vzorkování	0,01 s
Horizont predikce	450
Horizont řízení	2
Váha akčních zásahů	0,001
Výchozí hodnota akční veličiny	5 V
Omezení na velikost akčního zásahu	0–10 V
Typ a délka filtru	Medián délka 2

Tabulka 8.1: Parametry MPC hlavní motor

Z průběhu statické charakteristiky 4.3 plyne, že se zde vyskytuje široké pásmo vstupní necitlivosti. Tento úsek zapříčiní, že zvyšující se vstupní napětí nemá vliv na výstupní napětí udávající úhel elevace. V tomto případě dochází ke změně elevace až při napětí přibližně 8,5 V.

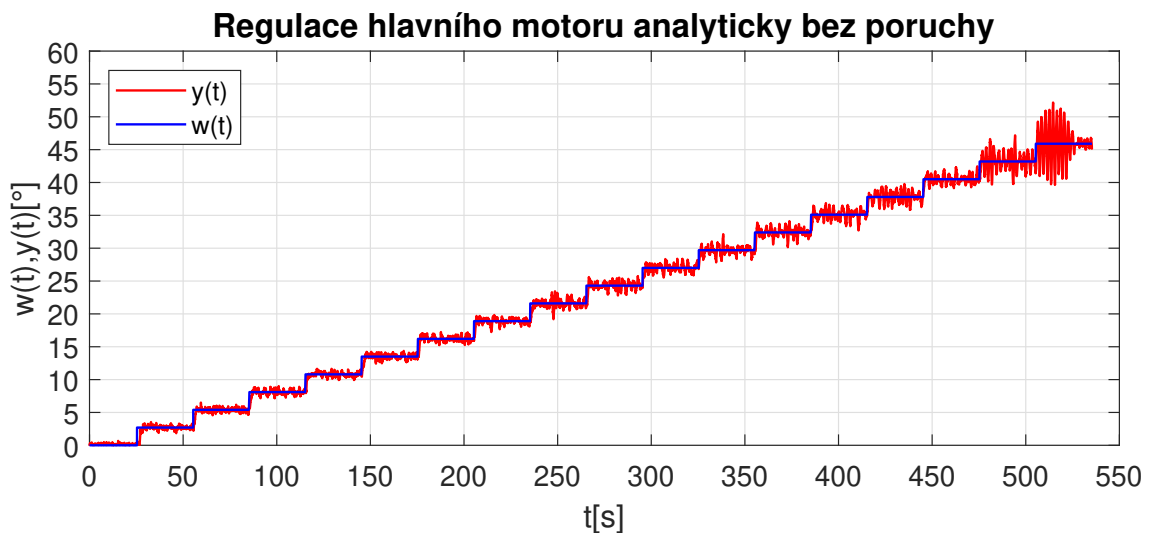
Toto nevhodné chování bylo zmírněno tím, že počáteční hodnota akčního zásahu byla změněna z 0 V na 5 V. Byl tak zmenšen úsek, kdy dochází k postupnému růstu akčního zásahu do pracovního bodu a tím je zmenšeno i vyskytující se zpoždění. Dále byl použit filtr na signál elevace. Byly vyzkoušeny různé struktury a jejich délky. Uspokojivého výsledku dosáhl Medián filtr o délce 2 prvky z generické knihovny LabVIEW. Jeho aplikací došlo ke snížení šumu bez vlivu na vlastní dynamiku systému. Na grafu 8.1 jsou zobrazeny regulační pochody porovnávající filtrovaný a nefiltrovaný signál elevace a taktéž i výstup ze simulace v MATLABu. Tato uvedená vylepšení byla následně využita ve všech dalších experimentech. Integrální kritérium kvality regulace mělo u nefiltrovaného signálu hodnotu  $J = 249,79$ , kdežto u filtrovaného signálu došlo ke zlepšení na hodnotu  $J = 216,41$ . V procentuálním vyjádření je to zlepšení o 16 %.





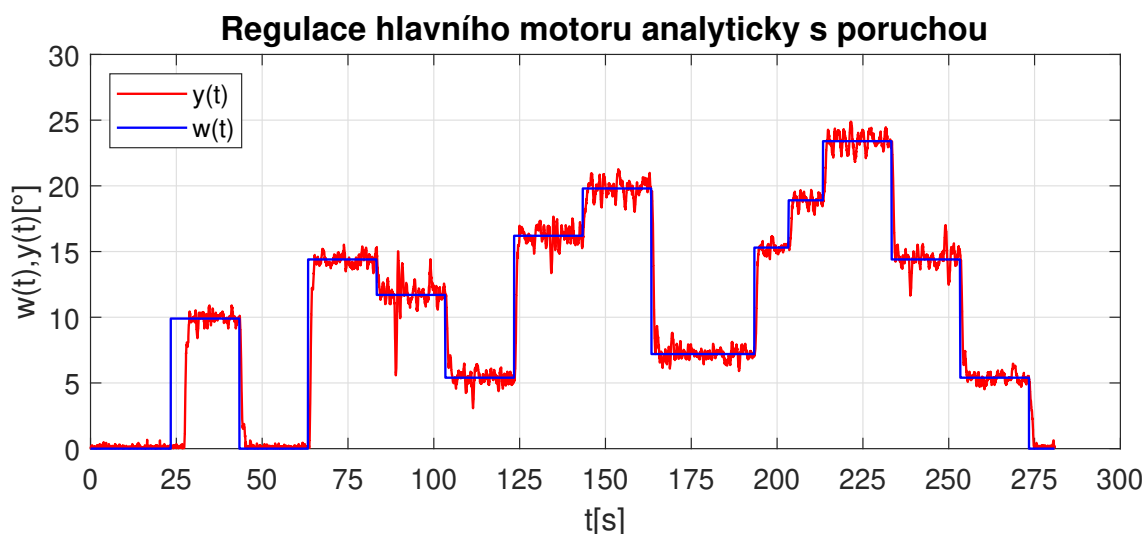
Obrázek 8.1: Porovnání regulace hlavního motoru analyticky

Dalším krokem bylo otestovat, v jakých mezích žádaných veličin je možné stabilně řídit elevaci. Prediktivní regulátor využívá lineární stavový model, který má však svou platnost pouze v okolí zvoleného pracovního bodu. Jelikož je tento systém výrazně nelineární, nelze předpokládat, že lze systém stabilně řídit v celém rozsahu  $0-90^\circ$ . Graf 8.2 ukazuje, že oblast, ve které je možné řídit elevaci je přibližně  $0-50^\circ$ . V tomto případě mělo kritérium kvality hodnotu 601,31.

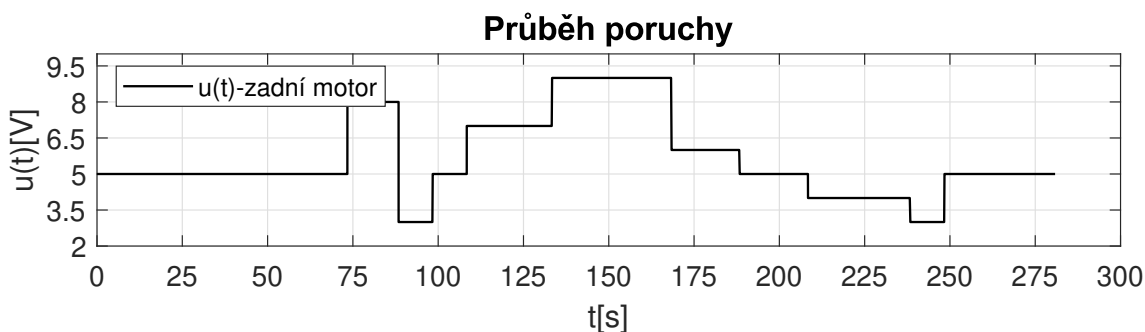


Obrázek 8.2: Regulace hlavního motoru analyticky - maximální elevace

Dále byla otestována odolnost proti působícím poruchám, respektive schopnost odstranit vliv poruchy. Žádaná hodnota byla tvořena postupnými skoky v rozmezí 0–50°, tedy v oblasti, kde je regulace stabilní. Porucha byla tvořena od druhého subsystému zadní vrtule a její průběh měl různou velikost a délku trvání. Z grafu plyne, že MPC dokáže vyregulovat vstupující poruchu a i nadále pracovat bez trvalé regulační odchylky. Dále je z grafu 8.3 vidět, že dle předpokladů má největší vliv na regulaci porucha ve svém přechodovém ději, při konstantní hodnotě již vliv není tak výrazný. Hodnota kritéria J dosahovala hodnoty 1 229,6.



Obrázek 8.3: Regulace hlavního motoru analyticky s poruchou

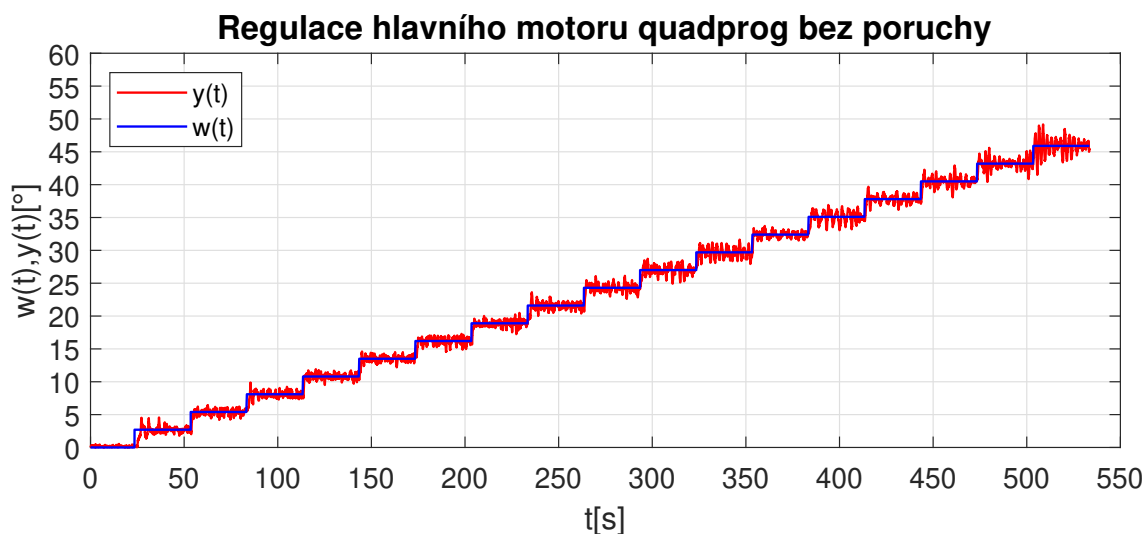


Obrázek 8.4: Zadní motor – průběh poruchy analyticky

## 8.2 Varianta s kvadratickým programováním

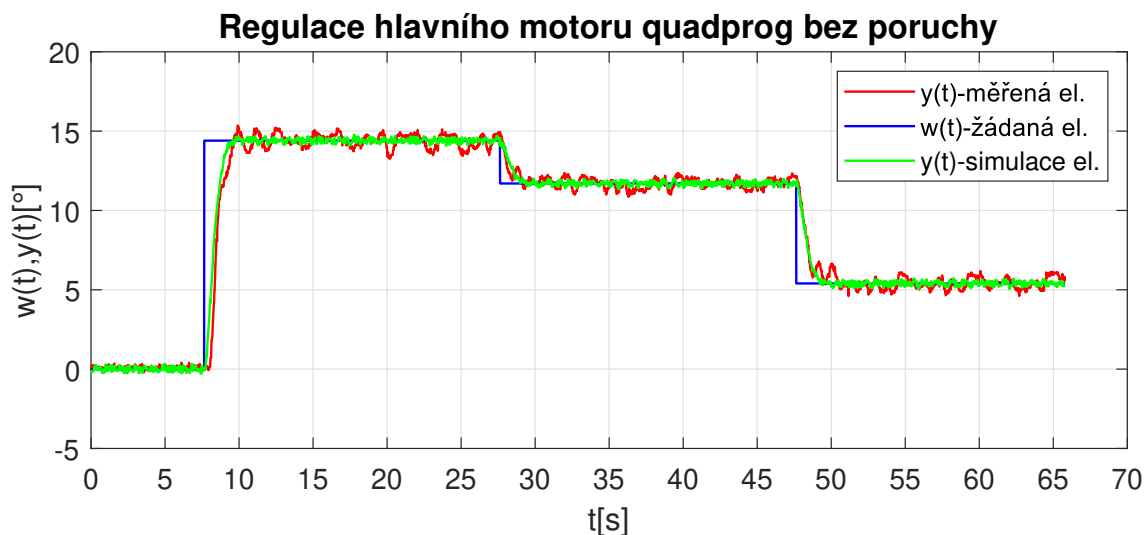
V této variantě je pro nalezení optimálního vektoru  $\Delta U$  využito kvadratické programování. Předností této varianty je, že výpočet optimálního vektoru  $\Delta U$  probíhá vzhledem k zadaným omezením tak, aby penalizační funkce 7.44 nabývala svého extrému, v tomto případě minima. Za nevýhodu lze určitě považovat vysokou výpočetní náročnost funkce kvadratického programování, obzvláště pokud je využita v procesech s malou periodou vzorkování. Opět bylo využito pouze omezení vzhledem k velikosti akčních zásahů na 0–10 V. Parametry MPC zůstaly stejné jako v předchozí variantě viz tabulka 8.1.

Na grafu 8.5 je zobrazen průběh regulace, kde opět bylo cílem zjistit maximální rozsah elevace. Je možné pozorovat, že s využitím kvadratického programování došlo k lehkému zvýšení rozsahu oproti předchozí variantě. Hodnota kritéria kvality byla 365,3.



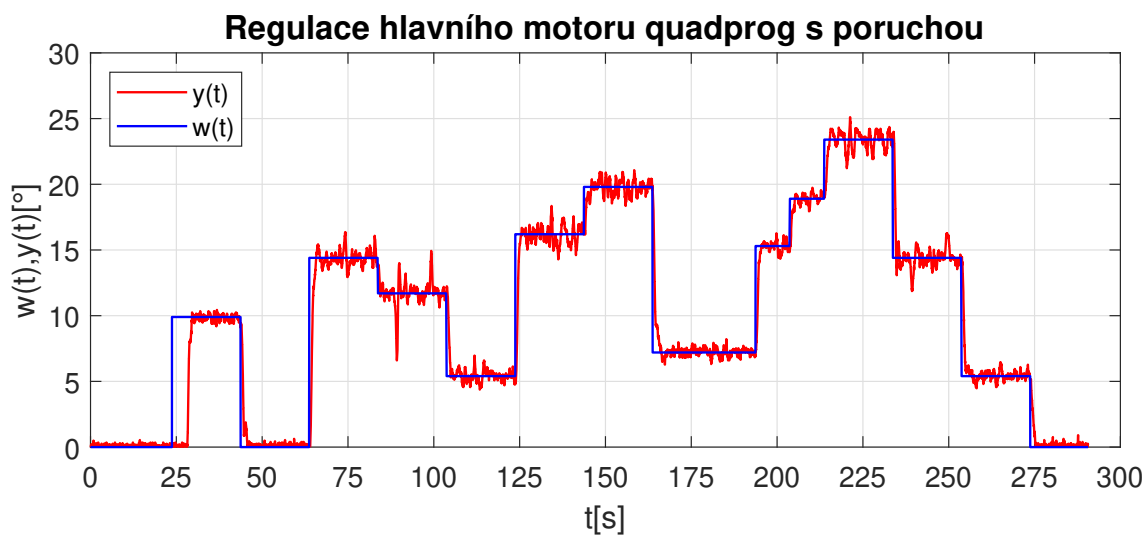
Obrázek 8.5: Regulace hlavního motoru quadprog – maximální elevace

Dále byla otestována celková schopnost regulátoru sledovat zadanou trajektorii. Žádaná hodnota je tvořena posloupností skoků o různé velikosti. Na grafu 8.6 je vidět, že vrtulník dokáže sledovat zadanou trajektorii, okolo které mírně osciluje. Amplituda těchto kmitů je však pouze v jednotkách v stupňů. Dále je vidět shodující se regulační pochod reálného systému a simulace. Hodnota kritéria  $J_{id}$  byla 214,58.



Obrázek 8.6: Porovnání regulace hlavního motoru quadprog

V posledním testu bylo cílem zjistit schopnost MPC kontroléru vyregulovat vstupující poruchu. Ta byla opět tvořena od druhého subsystému zadní vrtule. Její průběh je zobrazen na grafu 8.4. I v této variantě dokáže MPC regulátor pracovat bez trvalé regulační odchylky. Průběh regulačního pochodu je zobrazen na grafu 8.3. Hodnota kritéria  $J_{id}$  byla 1274,3.



Obrázek 8.7: Regulace hlavního motoru quadprog s poruchou

## 9 Návrh MPC pro SISO systém zadní vrtule

V této kapitole byl navržen MPC regulátor pro druhý subsystém, tedy pro vazbu mezi zadním motorem a azimutem. Byly opět vyzkoušeny dvě varianty regulátorů, tedy verze s analytickým postupem a verze, která k nalezení optimálního přírůstku  $\Delta \mathbf{u}$  využívá funkci kvadratické programování. I v případě tohoto systému musel být využit stavový pozorovatel, jelikož se zde vyskytují neměřitelné stavy. Systém zadní vrtule je charakteristický tím, že nulové otáčky jsou při napětí 5 V. Při poklesu, respektive růstu nad tuto hodnotu dochází ke změně směru otáčení vrtule a tím i ke změně směru vrtulníku v azimutu.

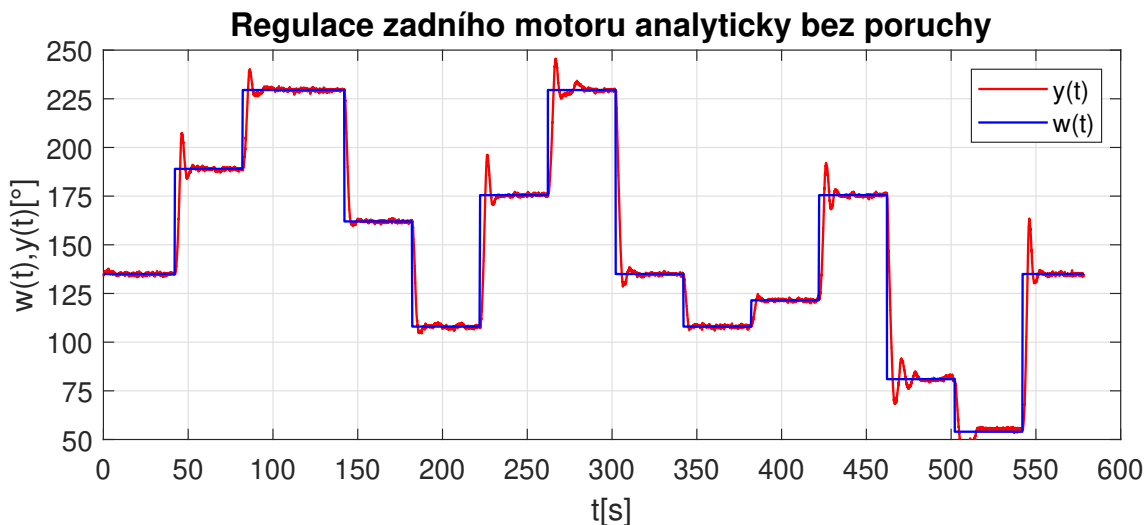
### 9.1 Varianta s analytickým výpočtem

Velikost horizontu predikce tak byla ponechána na stejné hodnotě jako v případě systému hlavní vrtule. Opět se zde vyskytuje omezení na velikosti akčního zásahu 0–10 V. Použité parametry jsou zobrazeny v tabulce 9.1.

Perioda vzorkování	0,01 s
Horizont predikce	450
Horizont řízení	2
Váha akčních zásahů	0,01
Výchozí hodnota akční veličiny	5 V
Omezení na velikost akčního zásahu	0–10 V
Typ a délka filtru	Medián délka 2

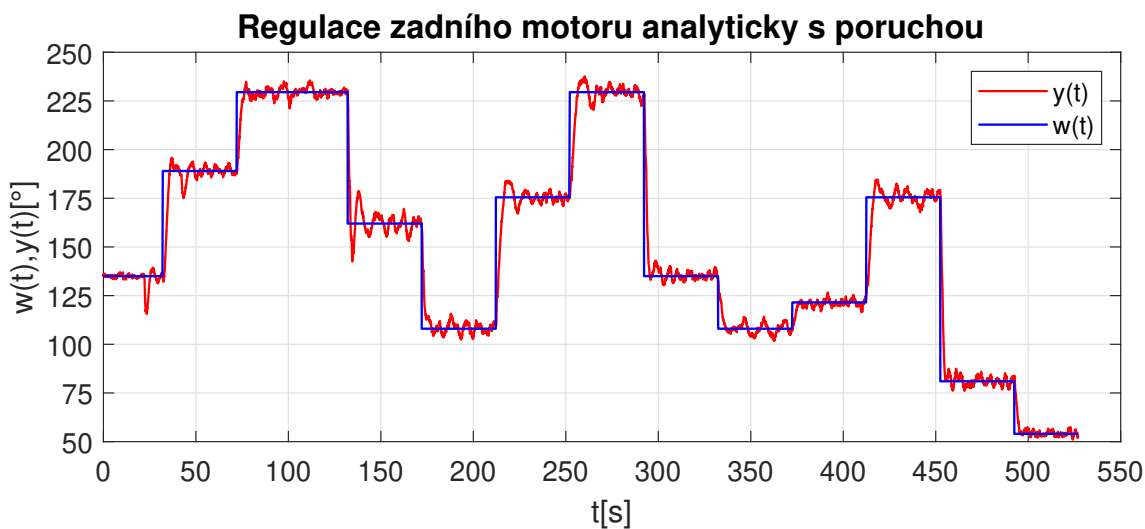
Tabulka 9.1: Parametry MPC zadní motor

Graf 9.1 zobrazuje průběh regulace systému zadní vrtule. Z grafu plyne, že tento systém je možné provozovat v celém rozsahu azimutu, tedy přibližně 270°.

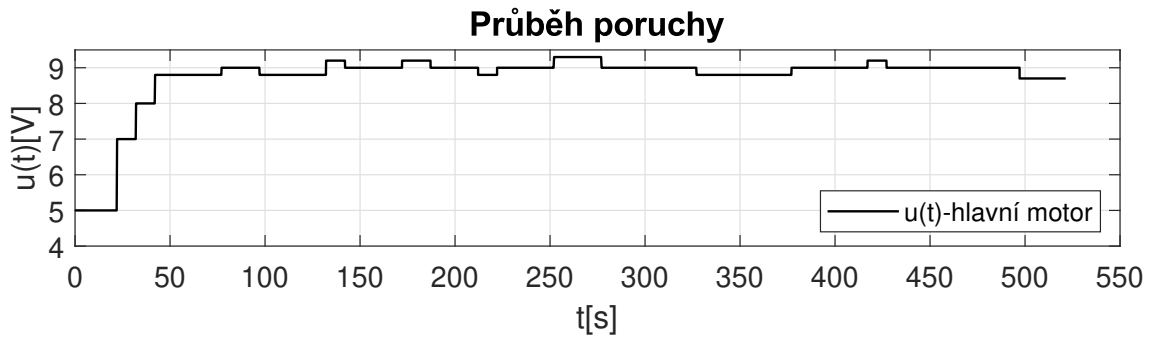


Obrázek 9.1: Regule zadního motoru analyticky

Dále pak byl navržený MPC regulátor otestován, pokud je zavedena porucha od systému hlavní vrtule. Na Grafu 9.2 je zobrazen regulační pochod, kdy byl průběh poruchy dynamický. Cílem bylo částečně simulovat chování systému, když má vrtulník nenulovou elevaci. Napětí přiváděné na hlavní motor se tak pohybovalo v rozmezí 8–9,2 V. Průběh poruchy je zobrazen na grafu 9.3.



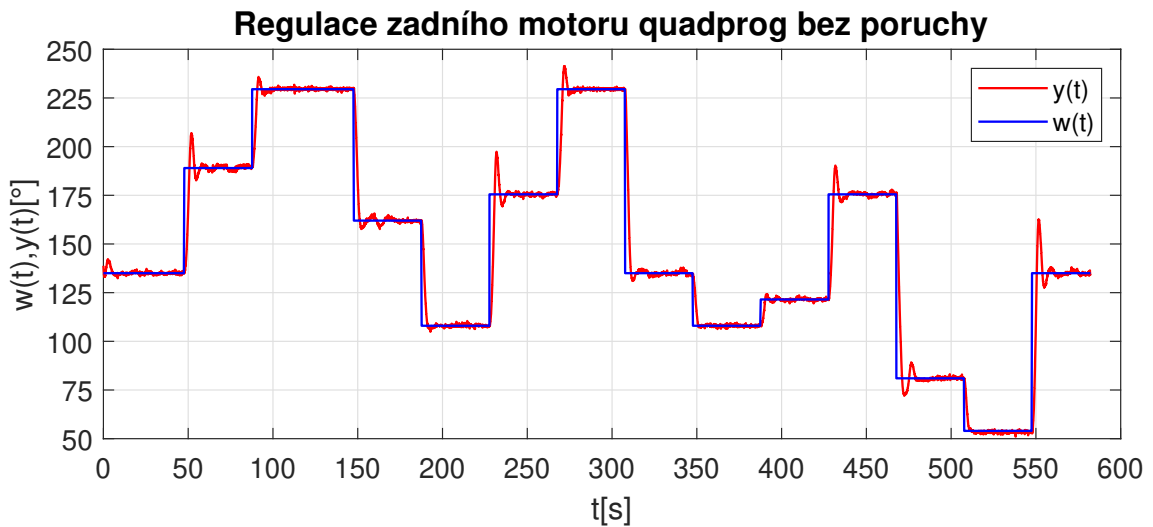
Obrázek 9.2: Regule zadního motoru analyticky s poruchou



Obrázek 9.3: Hlavní motor – průběh poruchy analyticky

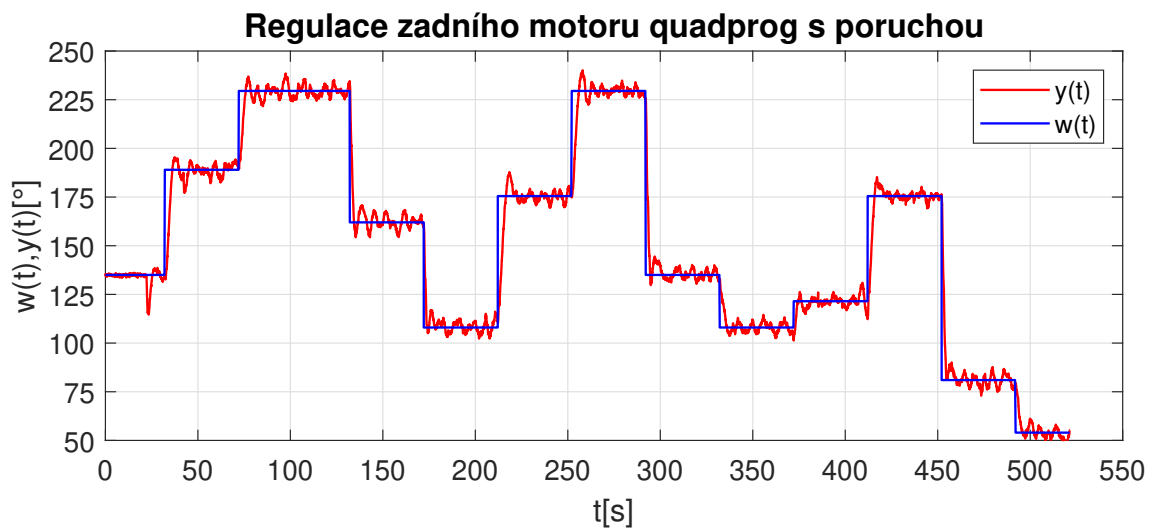
## 9.2 Varianta s kvadratickým programováním

Parametry a omezení MPC regulátoru byly ponechány stejné jako v předchozí variantě, viz tabulka 9.1. Graf 9.4 zobrazuje průběh regulace, pokud je akční zásah řešením optimalizačního problému pomocí kvadratického programování. Opět je možné systém provozovat v plném rozsahu azimutu. Z hlediska kvality regulace je situace podobná variantě s analytickou verzí.

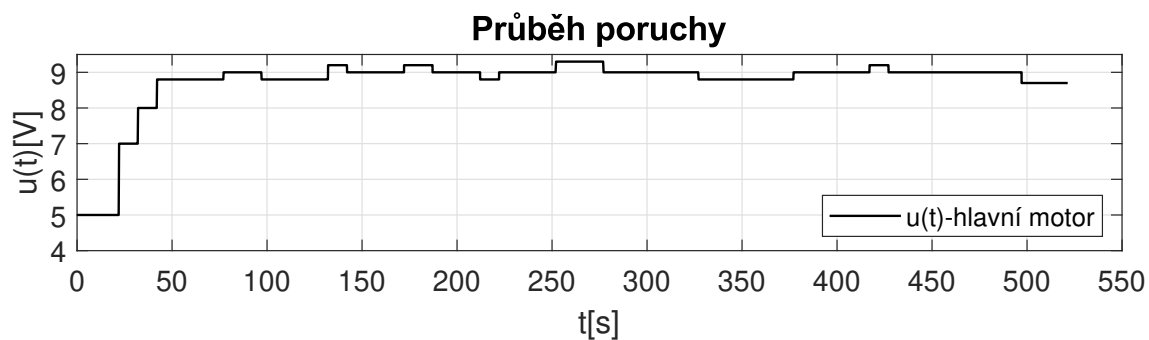


Obrázek 9.4: Regulace zadního motoru quadprog

Navržený MPC regulátor byl pak opět vyzkoušen z hlediska potlačení poruch, kde byla opět zavedena porucha ze subsystému hlavního motoru. Průběh poruchy byl stejný jako v analytickém případě viz graf 9.6. Regulační pochod je zobrazen na grafu 9.5.



Obrázek 9.5: Regulace zadního motoru quadprog s poruchou



Obrázek 9.6: Hlavní motor – průběh poruchy quadprog



## 10 Návrh MPC pro MIMO systém vrtulníku

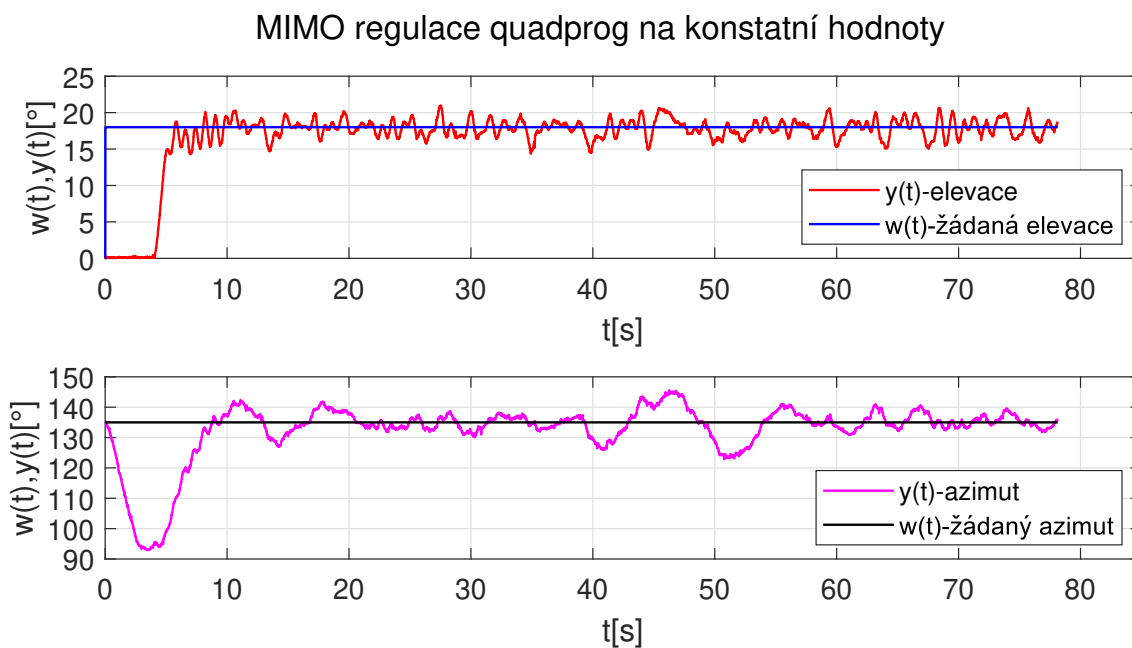
Tato kapitola se věnuje návrhu MPC regulátoru pro MIMO systém vrtulníku. Regulátor byl navržen na základě diskrétního stavového modelu 5.12. Velkou výhodou stavového popisu je vcelku snadný přechod mezi SISO a MIMO systémy, tudíž základní struktura regulátoru použita v předchozích jednorozměrných systémech mohla být zachována a pouze byla rozšířena na vícerozměrový systém. Byly vyzkoušeny obě varianty výpočtu řídicího signálu, tedy s využitím kvadratického programování a analytickým výpočtem. Jelikož analytická varianta nemá v praxi tak velké uplatnění jako verze s postupnou optimalizací využívající kvadratické programování, nejsou výsledky této verze uvedeny v plném rozsahu.

Vzhledem k vysoké výpočetní náročnosti funkce `quadprog` byl využit stejný způsob filtrování vstupních signálů jako u SISO systémů. Tedy filtrace probíhá uvnitř řídicí smyčky s periodou 10 ms. Naopak výhodou představuje jednoduché kladení různých druhů omezení, která jsou respektována již při výpočtu optimálních řídicích signálů. Parametry MPC využívající kvadratické programování jsou shrnuty v tabulce 10.1 a byly získány na základě několika desítek experimentů, kde byly vyzkoušeny různé kombinace parametrů horizontu řízení a predikce, vzorkovací periody a váhy akčních zásahů. Dle obdobných testů byly odvozeny i parametry regulátoru pro SISO systémy.

Perioda vzorkování	0,01 s
Horizont predikce	600
Horizont řízení	2
Váha akčních zásahů	0,001
Výchozí hodnota akčních veličin	5 V
Omezení na velikost akčních zásahů	0–10 V
Omezení na velikost přírůstku akčních zásahů	$\pm 1,5$ V
Typ a délky filtrů	Medián délka 2

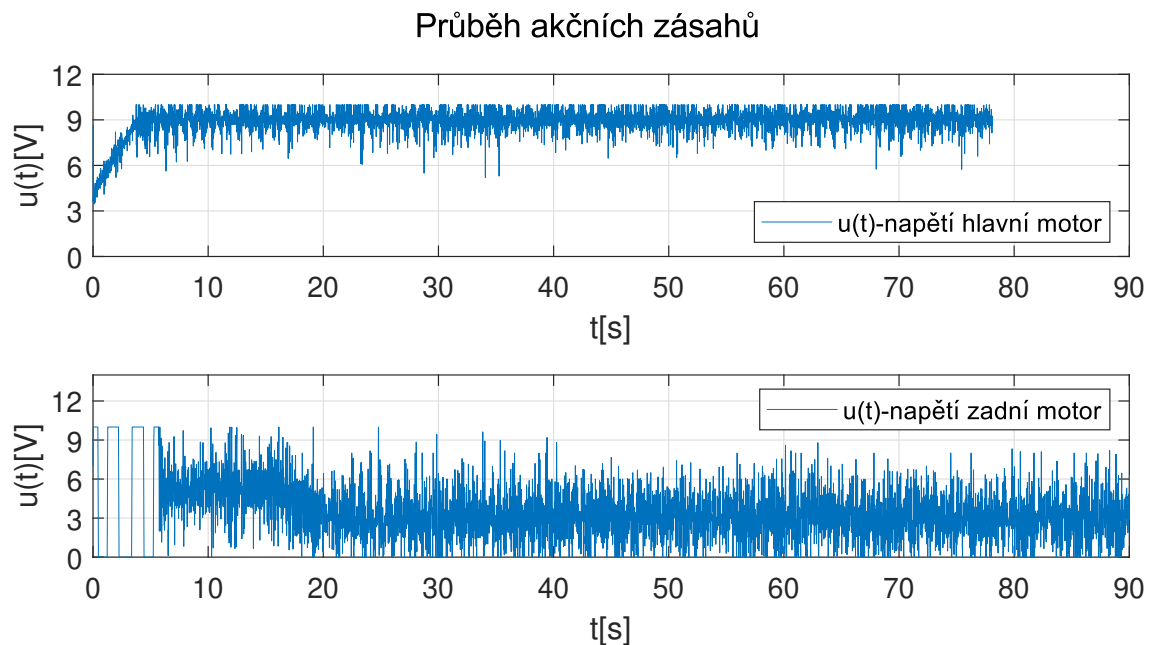
Tabulka 10.1: Parametry quadprog MIMO MPC

Na grafu 10.1 je možné pozorovat, že se v případě elevace stále vyskytuje zpoždění, které však bylo zmenšeno změnou výchozí velikosti akční veličiny. Nyní se tak toto zpoždění pohybuje okolo 5 sekund. V elevaci dochází k lehkým zákmitům v jednotkách stupňů kolem žádané hodnoty. V případě azimutu jsou tyto kmity co do velikosti výraznější, přesto je regulace stabilní a vrtulník je schopen sledovat obě žádané veličiny. Integrální kritérium v elevaci dosahovalo hodnoty 1 903,6 a v azimutu 1 046,6.



Obrázek 10.1: MIMO regulace quadprog na konstantní hodnoty

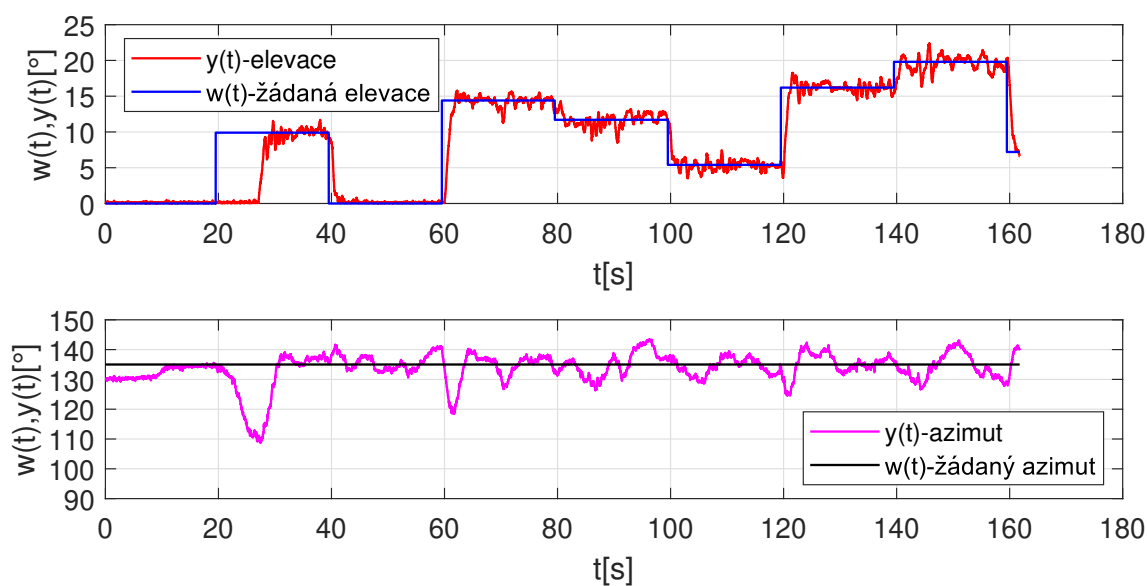
Z průběhu akčních zásahů 10.2 je možné pozorovat posun výchozího bodu akční veličiny. Hodnota napětí na hlavním motoru je před změnou požadované veličiny 5 V a po skokové změně žádané veličiny toto napětí roste na potřebné hodnoty ke splnění cílů regulace. Taktéž dochází k aktivaci zadaných omezení. V případě elevace je to především horní limit 10 V. Naopak u azimutu je to především spodní limit 0 V.



Obrázek 10.2: MIMO regulace quadprog průběh akčních zásahů

Další způsob testování regulační smyčky spočíval ve skokových změnách žádané hodnoty elevace a konstantní žádanou hodnotou azimutu. Opět je možné pozorovat zpoždění při prvotní změně žádané veličiny elevace. Při dalším skoku z nulové polohy je regulace již bez zpoždění. Z vizuálního hlediska lze kvalitu regulace posoudit obdobně jako v předchozím případě. Opět dochází k malým kmitům jak v případě elevace, tak i azimutu, přesto regulace pracuje bez trvalé regulační odchylky. Integrovaná kritéria pro regulaci elevace mělo hodnotu 1 549,5 a pro regulaci azimutu 1 103,6.

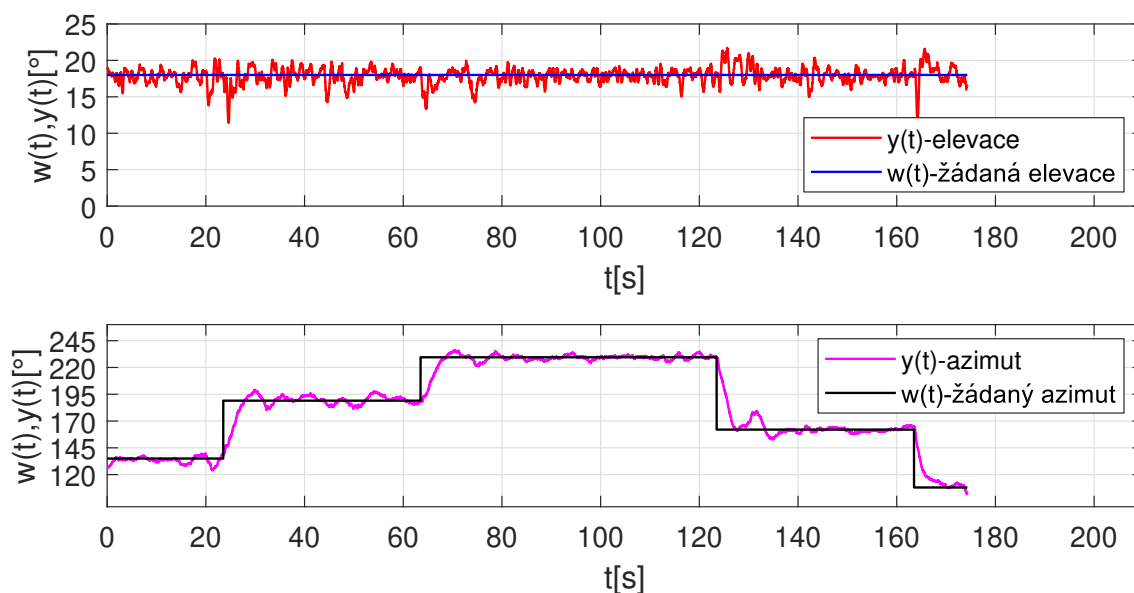
### MIMO regulace quadprog proměnná elevace



Obrázek 10.3: MIMO regulace quadprog proměnná elevace

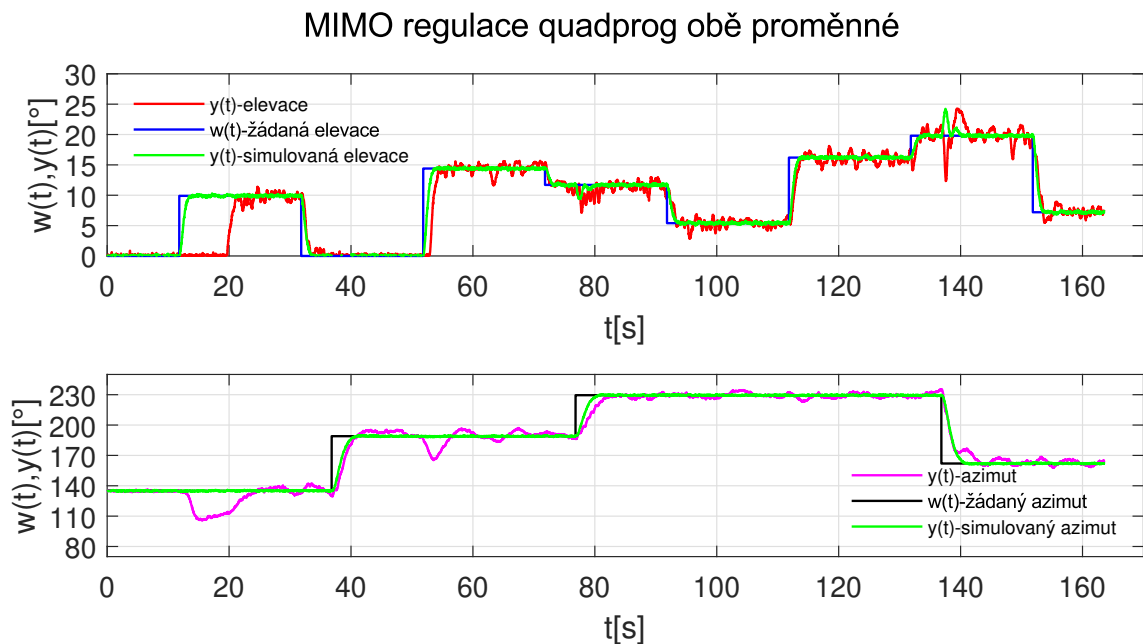
Další test byl založen na opačné situaci, kdy žádaná hodnota elevace byla konstantní a skokově měnící se byla žádaná hodnota azimutu. I s touto situací se MPC regulátor vypořádal bez problémů a opět bylo umožněno řízení bez trvalých odchylek. Hodnota kritéria v elevaci byla 281,6 a v azimutu 2 259,6.

### MIMO regulace quadprog proměnný azimut



Obrázek 10.4: MIMO regulace quadprog proměnný azimut

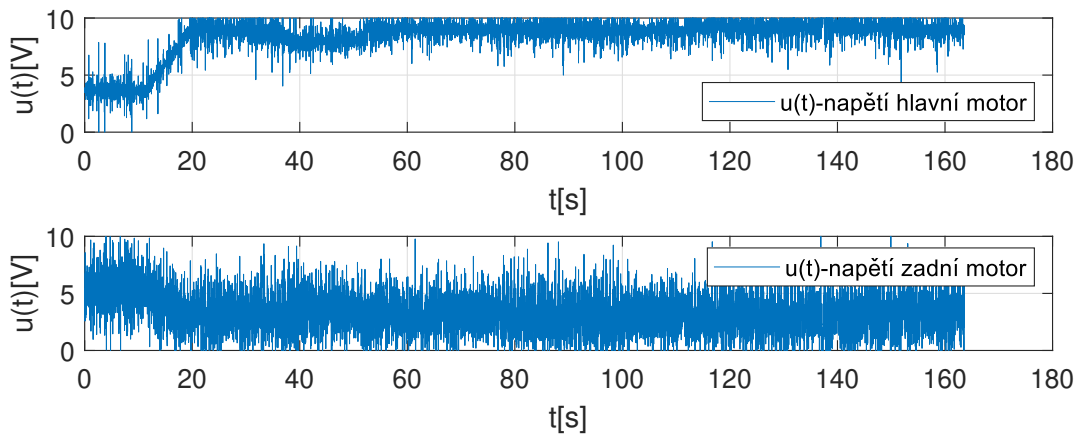
Na závěr byl vyzkoušen způsob regulace, který měl zajistit sledování skokově proměnných žádaných veličin elevace a azimutu. Na grafu 10.5 je zobrazeno porovnání reálného a simulovaného regulačního pochodu. V případě simulace se zde nevyskytuje prvotní zpoždění způsobené růstem akčního zásahu do pracovního bodu. U reálné soustavy nastávají opět lehké zákmity kolem žádané veličiny na rozdíl od simulace. Kritérium J mělo hodnotu 1 836,5 v elevaci a 2 648,8 v azimutu.



Obrázek 10.5: MIMO regulace quadprog obě proměnné

Akční zásahy jsou podobné předcházejícím situacím a ukazují, že jak napětí hlavního, tak i zadního motoru při regulaci často dosahuje hodnot, při kterých jsou aktivována příslušná omezení. Napětí hlavního motoru se podle předpokladů často pohybuje na hranici horního omezení, přesto i ve výjimečných situacích je aktivováno spodní omezení. V případě zadního motoru jsou aktivována horní i dolní omezení. Průběhy napětí na jednotlivých motorech mají ve všech uvedených případech značný rozptyl, což jistě zvyšuje jejich opotřebení. Byly provedeny pokusy o zlepšení této situace s využitím filtru akčních zásahů nebo s prodloužením periody vzorkování. Obě tyto varianty vedly k jednoznačnému zhoršení regulačního pochodu a následné destabilizace.

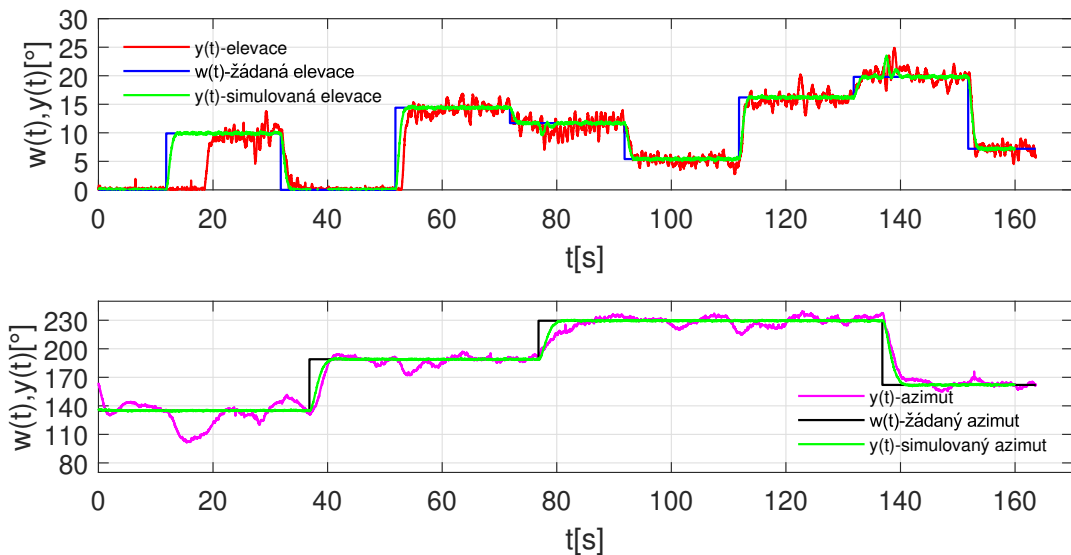
Průběh akčních zásahů



Obrázek 10.6: MIMO regulace quadprog obě proměnné průběh akčních zásahů

Pro ilustraci je na grafu 10.7 zobrazena analytická verze MPC, jenž využívá stejné parametry regulátoru. Modifikací je úprava filtrace vstupních signálů, které jsou nyní měřeny v mnohem rychlejší smyčce s periodou 1 ms, která umožňuje použití větší délky filtru bez ovlivnění stability regulace. Bylo vyzkoušeno více druhů filtrů např. Medián, Moving Avarage, IIR filtr. Nejlepších výsledků dosáhl medián filtr o délce 10 prvků. Tyto filtrované signály jsou pak předávány řídicí smyčce, která pracuje se standardní periodou 10 ms. Tato změna by se měla projevit dalším zklidněním regulačního pochodu.

MIMO regulace analyticky obě proměnné



Obrázek 10.7: MIMO regulace analyticky obě proměnné

## 11 Explicitní prediktivní řízení

Prediktivní řízení založené na modelu (MPC) našlo své uplatnění před více než 30 lety, jelikož dokáže řídit vícerozměrové systémy a zároveň při svém výpočtu respektuje zadané technická a technologická omezení na řízené soustavě. MPC obvykle ke svému řešení využívá techniku online kvadratického programování, kvůli jehož vyšší výpočetní náročnosti je využíván především v procesech s pomalejší dynamikou, kde není vyžadována malá perioda vzorkování. Další nevýhodou tohoto přístupu může být nesprávně provedený výpočet kvadratického programování či jeho chyba, což může skrývat bezpečnostní riziko.

Explicitní MPC umožňuje vyřešit problém optimalizace offline pro daný rozsah požadovaných pracovních podmínek. Pomocí technik multiparametrického programování, explicitní MPC vypočítává optimální řídicí signál offline jako funkci stavů a referenčních vektorů, takže online operace je pak redukována na volbu, který z regionů bude aktivní. Tímto dochází k úspoře výpočetního výkonu. [17]

Problémem je nyní nalezení příslušné funkce. Jedná se o obdobný přístup jako u klasické varianty MPC, která využívá minimalizaci kvadratického kritéria. V případě standardního MPC je nejčastěji využito kvadratické programování, kdežto u explicitního MPC se využívá multiparametrického kvadratického programování, jelikož řešení je hledáno pro všechny možné stavy  $\mathbf{x}$ . Výsledkem je afinní funkce  $\Gamma(\mathbf{x})$ , která představuje řídicí zákon:

$$\Gamma(\mathbf{x}) = \begin{cases} \mathbf{F}_1 \mathbf{x} + \mathbf{G}_1 & ; \mathbf{x} \in \mathbf{R}_1 \\ \vdots & \\ \mathbf{F}_R \mathbf{x} + \mathbf{G}_R & ; \mathbf{x} \in \mathbf{R}_R \end{cases} \quad (11.1)$$

kde  $\mathbf{R}_i \subset \mathbf{R}^{n_x}; i = 1 \dots R$

$\mathbf{R}$  jsou označovány kritické regiony a členy  $\mathbf{F}$  a  $\mathbf{G}$  jsou v nich konstantní. Tyto regiony rozdělují stavový prostor  $\mathbf{x}(t)$  na jednotlivé části, v nichž je definována spojitá afinní funkce 11.1. [16] Postup výpočtu akčního zásahu explicitního regulátoru lze shrnout do následujících bodů:

- Estimace současného stavu systému skrze měření
- Identifikace regionu, ke kterému aktuální stav  $\mathbf{x}$  náleží
- Vyhledání parametrů  $\mathbf{F}$  a  $\mathbf{G}$  identifikovaný region
- Výpočet funkce  $\mathbf{F}\mathbf{x} + \mathbf{G}$

K implementaci explicitního prediktivního regulátoru byl využit Hybrid Toolbox [17] a možnost exportu modelu z prostředí MATLAB/SIMULINK. Tento toolbox využívá k zajištění sledování reference následující kritérium:

$$J_e = \sum_{k=0}^{N-1} [y(t+k|k) - r(t)]^T S [y(t+k|k) - r(t)] + \Delta u(t+k)^T \Delta u(t+k) + \rho \epsilon^2 \quad (11.2)$$

Postup návrhu explicitního regulátoru pomocí Hybrid Toolboxu vychází z konstrukce klasického MPC. Nejprve je tedy nutné navrhnout standardní (implicitní) MPC a následně převést do explicitní formy. Návrh implicitního regulátoru pak vyžaduje tyto parametry:

- Stavový popis soustavy
- Perioda vzorkování
- Horizont predikce
- Horizont řízení
- Omezení akční a přírůstku akční veličiny
- Váhy regulační odchylky a akční veličiny



Takto vytvořený implicitní MPC regulátor je výchozím bodem pro návrh explicitního regulátoru. Zde je nutné specifikovat rozsah stavových proměnných, což může být obtížné, když stav nekorresponduje s fyzickým parametrem. V tomto případě je vhodné provést několik simulačních příkladů s typickými průběhy žádaných a poruchových hodnot a sledovat rozsah stavových proměnných. Dále je nutné určit rozsahy referenčního signálu. V tuto chvíli je možné vytvořit explicitní prediktivní regulátor. Nevýhodou je v tomto případě exponenciální závislost počtu regionů na počtu vstupních parametrů. Tuto vlastnost lze částečně kompenzovat spojením regionů se stejnými hodnotami zesílení.

Nyní je možné vytvořený objekt explicitního prediktivního regulátoru otevřít v prostředí SIMULINK. Pro export modelu je nutné nainstalovat softwarové nástroje, jimiž jsou NI Veristand Model Framework, Veristand a Model Interface Toolkit. Tyto programy musí být nainstalovány jednak na straně RT targetu, tak i na vývojovém PC. Dále musí být v SIMULINKu zvolen WIND RIVER GNU kompilér, jenž vygeneruje soubory kompatibilní s LabVIEW, respektive CompactRIO. [18] Následující diagram 11.1 zobrazuje průběh návrhu modelu. Oranžově jsou označeny kroky v MATLABu, zeleně naopak kroky nezbytné v LabVIEW.



Obrázek 11.1: Vývojový diagram exportu modelu [19]

## 12 Návrh explicitního MPC pro SISO systém hlavního motoru

V případě testování explicitního regulátoru bylo postupováno obdobně jako v případě implicitní verze. Nejprve byl tedy navržený regulátor otestován v simulacích a následně vyzkoušen na reálné soustavě. Následovala fáze ladění regulátoru, kdy pomocí vah bylo cílem dosáhnout lepších výsledků.

K návrhu explicitního regulátoru pro systém hlavního motoru byl využit stavový popis popsáný rovnicemi 12.1,12.2. Tento model je na rozdíl od standardní verze rozšířen o přídatný stav, který představuje konstantní výstupní poruchu. Tato modifikace umožňuje řízení bez trvalé regulační odchylky i v případech, kdy na soustavu působí zmíněná porucha, vyskytuje se nelinearita či identifikovaný systém není přesnou aproximací reálné soustavy. [20]

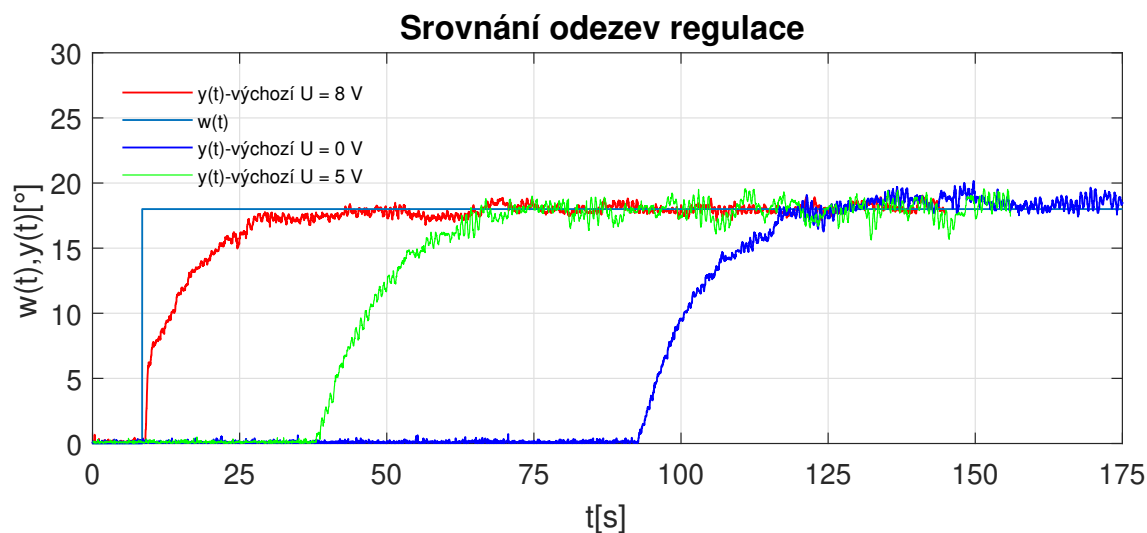
$$\underbrace{\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{p}(k+1) \end{bmatrix}}_{\mathbf{x}_{\text{aug}}(k+1)} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}}_{\mathbf{A}_{\text{aug}}} \underbrace{\begin{bmatrix} \mathbf{x}(k) \\ \mathbf{p}(k) \end{bmatrix}}_{\mathbf{x}_{\text{aug}}(k)} + \underbrace{\begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}}_{\mathbf{B}_{\text{aug}}} \mathbf{u}(k) \quad (12.1)$$

$$\mathbf{y}(k) = \underbrace{\begin{bmatrix} \mathbf{C} & \mathbf{G}_p \end{bmatrix}}_{\mathbf{C}_{\text{aug}}} \underbrace{\begin{bmatrix} \mathbf{x}(k) \\ \mathbf{p}(k) \end{bmatrix}}_{\mathbf{x}_{\text{aug}}(k)} \quad (12.2)$$

Standardně je voleno  $\mathbf{G}_p = \mathbf{1}$ .

Vzhledem k vysoké vstupní necitlivosti systému hlavní vrtule byly opět provedeny úpravy, které tento jev potlačují. Na grafu 12.1 je vidět, že pokud je cílem okamžitá reakce systému, je vhodné zvolit počáteční hodnotu akčního zásahu v bodě 8 V, což koresponduje s průběhem statické charakteristiky 4.3.

Pokud je zvolen počáteční akční zásah v bodě 5 V, tak se zde již vyskytuje dopravní zpoždění přibližně 35 sekund. V případě výchozí akčního zásahu 0 V je již toto zpoždění okolo 90 sekund. Z těchto důvodů byla využita výchozí hodnota 8 V, která zaručuje reakci bez dopravního zpoždění. Přehled parametrů explicitního regulátoru hlavního motoru je shrnut v 12.1.

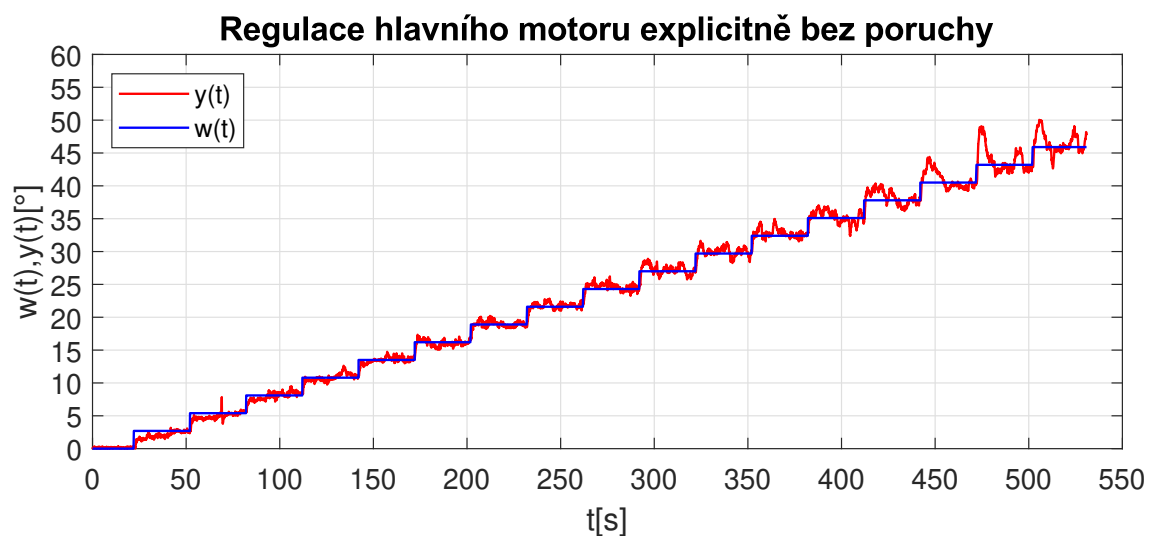


Obrázek 12.1: Regulace hlavního motoru explicitní – srovnání

Perioda vzorkování	0,015 s
Horizont predikce	180
Horizont řízení	4
Váha regulační odchylky	10
Váha akční veličiny	0,1
Výchozí hodnota akční veličiny	8 V
Omezení na velikost akčního zásahu	0–10 V
Omezení na velikost přírůstku akčních zásahů	$\pm 1,5$ V
Typ a délka filtru	Medián délka 2
Počet vygenerovaných regionů	275

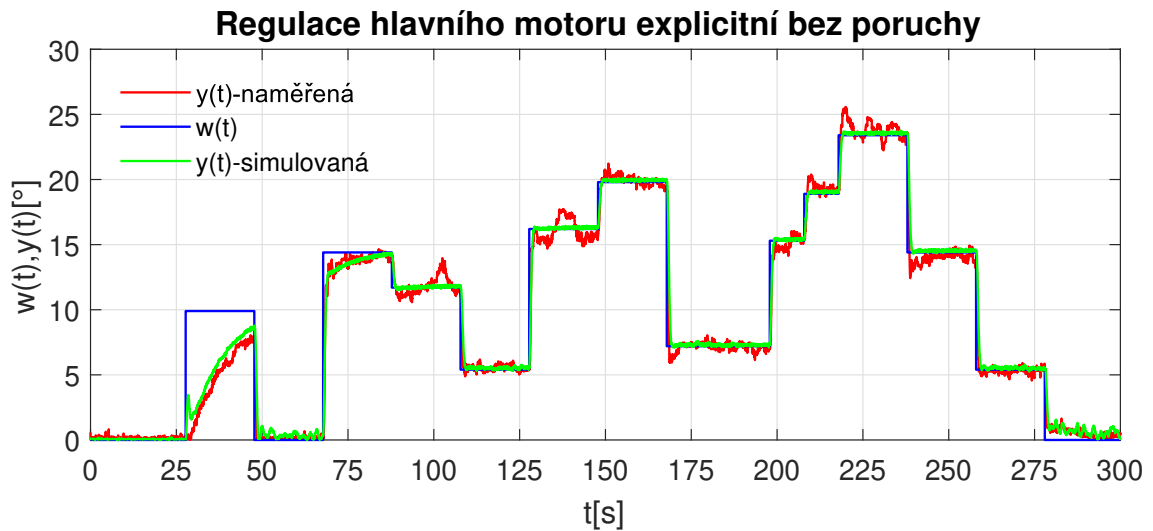
Tabulka 12.1: Parametry explicitního MPC hlavní motor

Stejně jako v případě implicitní verze, tak i v explicitní variantě regulátoru byla vyzkoušena maximální hodnota elevace, ve které lze vrtulník stabilně ovládat. Žádaná hodnota byla tvořena postupnými skoky o délce 30 sekund a amplitudě 3°. Z grafu 12.2 plyne, že vrtulník je v případě explicitní varianty regulátoru schopen řídit svůj pohyb přibližně ve stejném rozsahu jako implicitní verze, tedy 0–45°. Dále je vidět, že co se týče kvality sledování referenční trajektorie, tak regulátor podle předpokladů vyniká především v okolí pracovního bodu, což je okolo 25°. S rostoucí žádanou hodnotou se pak vlivem nelinearit a odchylek modelu vyskytují oscilace.



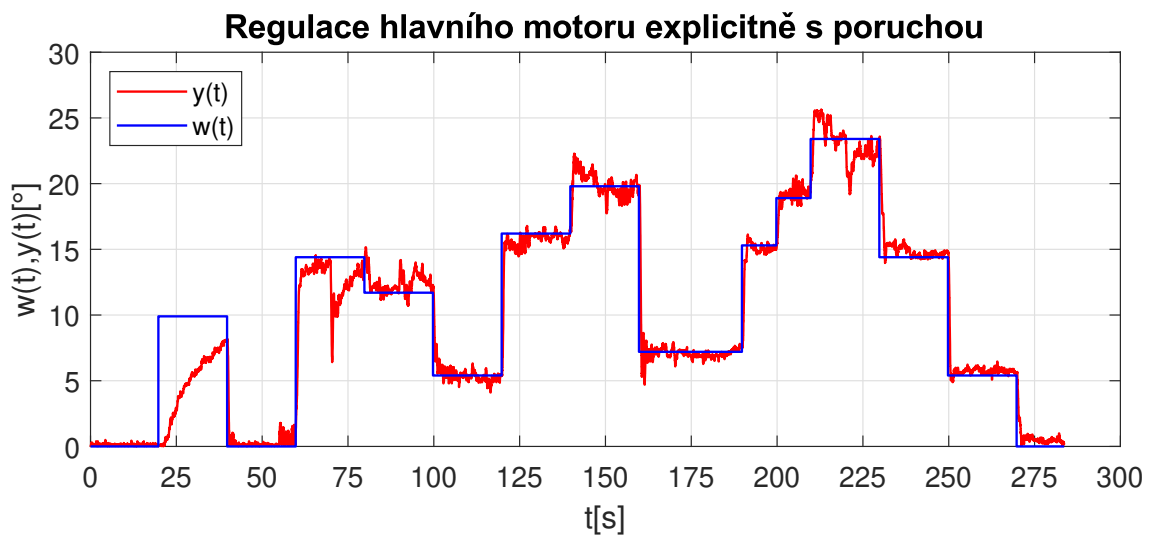
Obrázek 12.2: Regulace hlavního motoru explicitní - maximální elevace

Dále byla otestována celková schopnost regulátoru sledovat zadanou trajektorii. Žádaná hodnota je v tomto případě opět tvořena posloupností skoků o rozdílné velikosti i směru. Z grafu 12.3 plyne, že při prvotním skoku v 25 sekundách je vlivem kombinací faktorů nízké žádané hodnoty a opatrného nastavení regulátoru nedosaženo požadované hodnoty včas. Tato vlastnost je zachycena i v simulaci, kde v roli regulované soustavy stojí systém nalezený identifikací, jenž je na výstupu doplněn o poruchový signál typu bílý šum. S následujícími referenčními signály dokáže regulátor pracovat bez trvalé regulační odchylky. Je nutné dodat, že celková kvalita a průběh regulace je velmi závislý především na vzorkovací periodě a taktéž i na horizontu predikce.

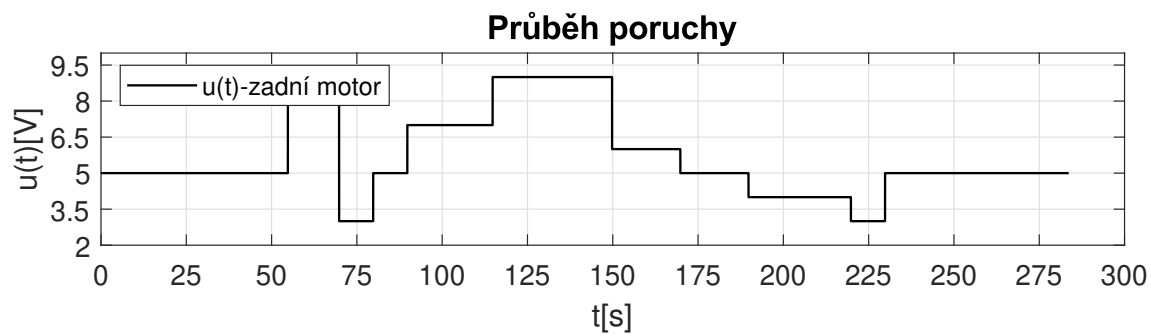


Obrázek 12.3: Regulace hlavního motoru explicitní bez poruchy

Poslední test systému hlavní vrtule spočíval ve vyregulování vstupující poruchy. Ta byla stejně jako v případě implicitního regulátoru tvořena systémem zadní vrtule. I v této variantě je regulátor schopný poruchu potlačit a pracovat tak i nadále bez trvalé regulační odchylky viz 12.4. Průběh poruchy je zobrazen na 12.5.



Obrázek 12.4: Regulace hlavního motoru explicitní s poruchou



Obrázek 12.5: Zadní motor – průběh poruchy explicitně

## 13 Návrh explicitního MPC pro SISO systém zadního motoru

Tato kapitola se věnuje návrhu explicitního regulátoru pro systém zadní vrtule, který je význačný svým astatickým charakterem. Opět bylo postupováno systematicky, kdy byl nejprve navržený regulátor otestován v simulacích a následně pak použit při řízení reálného systému. Následující fáze se pak věnovala zlepšení dosažených výsledků.

Při řízení této soustavy musel být použit jiný typ rozšířeného stavového popisu, jelikož v předcházející variantě 12.1 a 12.2 nebylo možné dosáhnout uspokojivých výsledků. Důvodem může být vstupující porucha v jiné části regulační smyčky. Řešení poskytuje stavový popis 13.1 a 13.2, který využívá stavovou poruchu. [20] V tabulce 13.1 je zobrazen přehled všech parametrů explicitního regulátoru. Opět je vhodné připomenout, že i v případě návrhu regulátoru pro tento systém hraje stěžejní roli volba vzorkovací periody v kombinaci s horizontem predikce. Souhrn použitých parametrů explicitního regulátoru je v 13.1.

$$\underbrace{\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{d}(k+1) \end{bmatrix}}_{\mathbf{x}_{\text{aug}}(k+1)} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{G}_d \\ \mathbf{0} & \mathbf{1} \end{bmatrix}}_{\mathbf{A}_{\text{aug}}} \underbrace{\begin{bmatrix} \mathbf{x}(k) \\ \mathbf{d}(k) \end{bmatrix}}_{\mathbf{x}_{\text{aug}}(k)} + \underbrace{\begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}}_{\mathbf{B}_{\text{aug}}} \mathbf{u}(k) \quad (13.1)$$

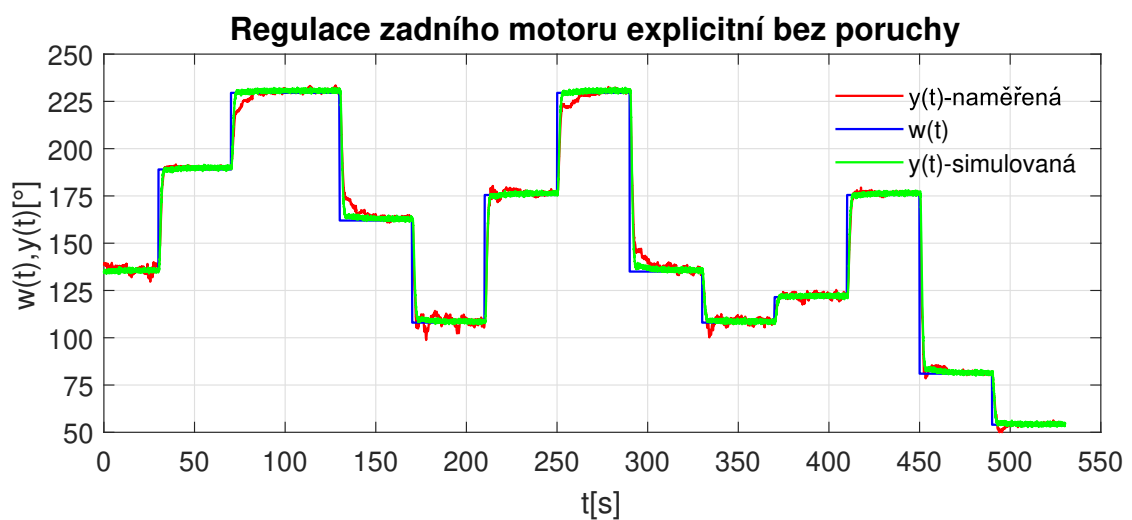
$$\mathbf{y}(k) = \underbrace{\begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix}}_{\mathbf{C}_{\text{aug}}} \underbrace{\begin{bmatrix} \mathbf{x}(k) \\ \mathbf{d}(k) \end{bmatrix}}_{\mathbf{x}_{\text{aug}}(k)} \quad (13.2)$$

Standardně je voleno  $\mathbf{G}_d = \mathbf{B}$ .

Perioda vzorkování	0,03 s
Horizont predikce	100
Horizont řízení	4
Váha regulační odchylky	10
Váha akční veličiny	0,1
Výchozí hodnota akční veličiny	5 V
Omezení na velikost akčního zásahu	0-10 V
Typ a délka filtru	Medián délka 2
Počet vygenerovaných regionů	52

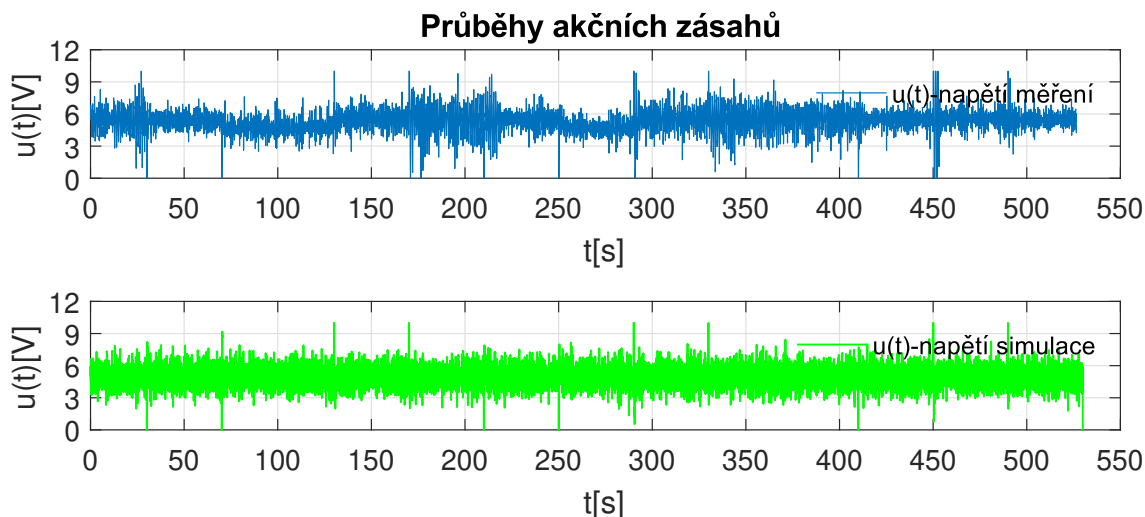
Tabulka 13.1: Parametry explicitního MPC zadní motor

Nejprve byla opět vyzkoušena regulace bez přítomnosti poruchy. Z grafu 13.1 plyne, že simulovaný a skutečný průběh se od sebe nijak zásadně neliší. I v tomto případě byl u simulované varianty v roli regulované soustavy model nalezený identifikací, jehož výstup byl doplněn o poruchový signál typu bílý šum. Dále je zřejmé, že regulátor pracuje bez trvalé regulační odchylky a je možné tento systém provozovat v celém jeho rozsahu, tedy  $\pm 135^\circ$ . Dále je z průběhu akčních zásahů 13.2 možné pozorovat, že pokud na systém nepůsobí silná porucha, tak se napětí na zadním motoru pohybuje okolo klidové hodnoty 5 V.



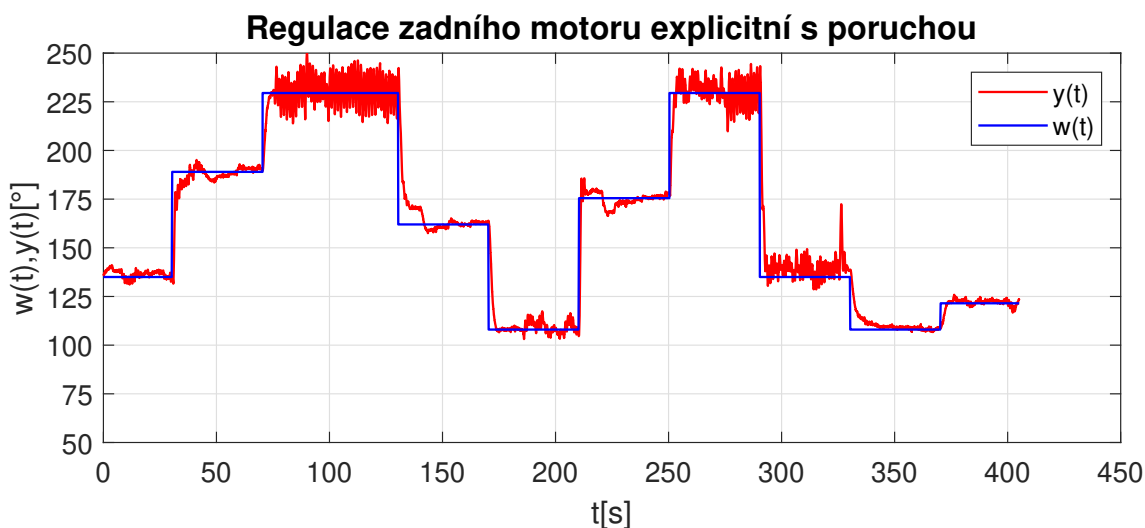
Obrázek 13.1: Regulace zadního motoru explicitní v1



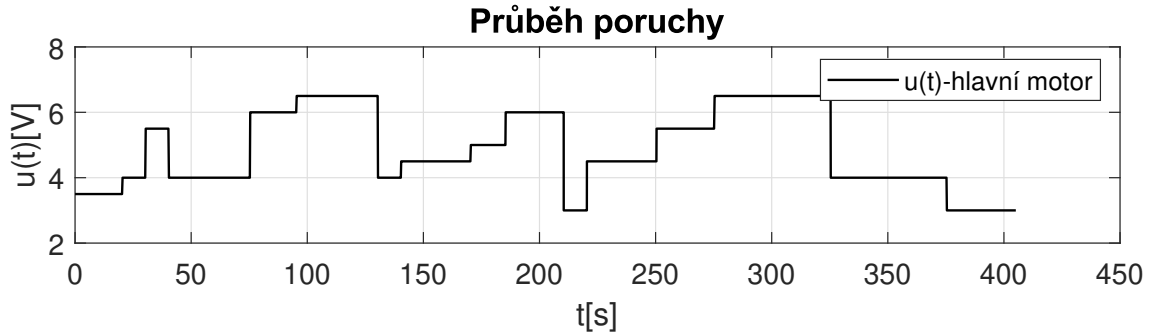


Obrázek 13.2: Regulace zadního motoru – akční zásahy

V rámci dalšího měření byl otestován vliv poruchy, která byla opět zavedena od druhého subsystému hlavní vrtule. Z naměřených průběhů 13.3 je patrné, že pokud je napětí na hlavním motoru pod úrovní 6 V, je regulace kvalitní, bez významných zákmitů. Naopak pokud je toto napětí vyšší než zmíněná hranice, tak se již objevují silné kmity kolem žádané hodnoty. Ty lze pozorovat zejména v čase 65–120 sekund, respektive 250–290 sekund. I přesto je však regulátor schopen pracovat bez trvalé regulační odchylky. Průběh poruchy je zachycen na grafu 13.4.



Obrázek 13.3: Regulace zadního motoru explicitní s poruchou v1



Obrázek 13.4: Hlavní motor – průběh poruchy explicitní

Vzhledem k vcelku velkým oscilacím, které nastávají při výrazné vstupující poruše, bylo dalším cílem zlepšit toto chování. Z tohoto důvodu byl využit jiný typ stavového popisu popsany rovnicemi 13.3 a 13.4. Tento typ je v podstatě kombinací stavových popisů využitých pro hlavní a zadní motor a využívá tak oba druhy možných poruch, tedy stavovou a výstupní. [20]

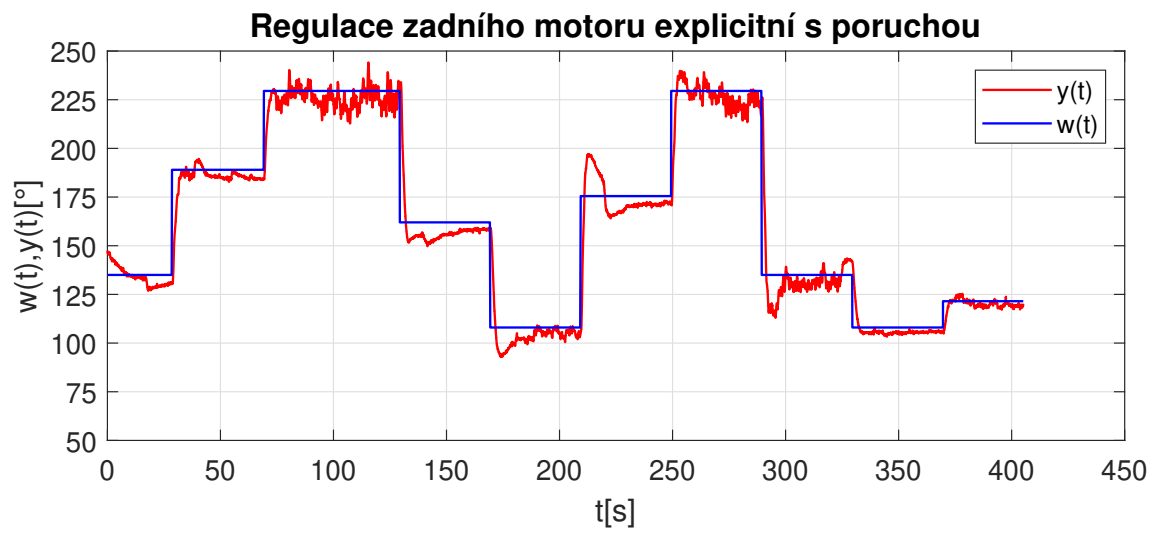
Dále byly vyzkoušeny i jiné parametry regulátoru, ale vzhledem k silné závislosti na vhodné vzorkovací periodě bylo nastavení dle 13.1 vyhodnoceno jako vyhovující.

$$\underbrace{\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{d}(k+1) \\ \mathbf{p}(k+1) \end{bmatrix}}_{\mathbf{x}_{\text{aug}}(k+1)} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{G}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}}_{\mathbf{A}_{\text{aug}}} \underbrace{\begin{bmatrix} \mathbf{x}(k) \\ \mathbf{d}(k) \\ \mathbf{p}(k) \end{bmatrix}}_{\mathbf{x}_{\text{aug}}(k)} + \underbrace{\begin{bmatrix} \mathbf{B} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}}_{\mathbf{B}_{\text{aug}}} \mathbf{u}(k) \quad (13.3)$$

$$\mathbf{y}(k) = \underbrace{\begin{bmatrix} \mathbf{C} & \mathbf{0} & \mathbf{G}_p \end{bmatrix}}_{\mathbf{C}_{\text{aug}}} \underbrace{\begin{bmatrix} \mathbf{x}(k) \\ \mathbf{d}(k) \\ \mathbf{p}(k) \end{bmatrix}}_{\mathbf{x}_{\text{aug}}(k)} \quad (13.4)$$

Standardně je voleno  $\mathbf{G}_d = \mathbf{B}$  a  $\mathbf{G}_p = \mathbf{1}$ .

Z důvodu srovnání s předchozí variantou byl využit stejný průběh poruchy a také i referenční signál. Na první pohled je patrné, že došlo ke zkvalitnění regulačního pochodu. V oblasti, kde se v předchozím případě vyskytovaly výrazné kmity, došlo ke snížení jejich amplitud.



Obrázek 13.5: Regulace zadního motoru explicitní s poruchou v2

## Závěr

Výsledkem této práce je úspěšné splnění všech bodů zadání. V první řadě jsem se seznámil s programovatelným kontrolérem CompactRIO. Následovalo získání teoretických znalostí o prediktivním řízení založeném na modelu v implicitní a explicitní variantě. Explicitní verze je charakteristická svými nižšími výpočetními nároky, jelikož problém optimalizace je řešen offline pomocí multiparametrického kvadratického programování. Během řízení se pak pouze vyhodnocuje aktuální stav systému, ke kterému náleží určitý akční zásah. Naopak u implicitní verze je optimalizace řešena opakovaně online, nejčastěji s využitím kvadratického programování.

Použitelnost především implicitní verze regulátoru byla v dřívějších dobách omezena na pomalejší soustavy, které nevyžadují vysokou frekvenci vzorkování. S neustálým vývojem elektroniky lze v dnešní době stabilně řídit systémy se vzorkovací periodou v jednotkách milisekund. V případě explicitní verze se vzorkování může pohybovat i v řádech nanosekund. V praxi je nejčastěji prediktivních regulátorů využito v oblasti rafinerií a chemického průmyslu, jelikož pro tyto oblasti byly původně konstruovány. V současnosti je však jejich uplatnění rozšířeno na oblast automotivy, kde je lze využít pro regulaci volnoběžných otáček, kontrolu trakce, aktivního zavěšení apod.

Nejprve jsem navrhl implicitní verzi MPC pro SISO systémy vrtulníku. Následně došlo k rozšíření, které umožňuje řízení i MIMO systému. Z tohoto důvodu byl prediktivní regulátor založen na stavovém popisu, jelikož poskytuje snadný přechod mezi jednorozměrovými a vícerozměrovými systémy. Dále byl MPC navíc vyvíjen ve dvou variantách, kde jedna ke svému výpočtu využívala analytický přístup a druhá naopak optimalizaci pomocí kvadratického programování.

Navržené regulátory jsem nejprve simuloval v prostředí LabVIEW. Po úspěšné verifikaci následovala tvorba RT aplikace pro CompactRIO a ověření regulace na reálné soustavě. U systému hlavního motoru hraje výraznou roli nelinearita zapříčiněná necitlivostí při nízkých hodnotách napětí. V tomto případě dochází k roztočení motoru při hodnotě 4 V a ke změně elevace až při napětí 8,5 V. Jelikož je akční zásah limitován na 0–10 V, tak k řízení systému zbývá pouze omezené pásmo. K odstranění zpoždění, které je zapříčiněné touto vlastností, jsem změnil výchozí hodnotu akčního zásahu z 0 V na 5 V. Dále byl použit filtr na signálu elevace. Dobré vlastnosti poskytoval medián filtr o délce 2 prvky, který dokázal vyhladit vyskytující se šum a tím zároveň zklidnit průběh akčních zásahů. V oblasti zvoleného pracovního bodu je regulace kvalitní, se zvyšující elevací se vlivem nelinearit začínají vyskytovat mírné oscilace, přesto lze systém hlavní vrtule provozovat přibližně v rozsahu 0–50°, jak analytické, tak i verzi s kvadratickým programováním. Z hlediska potlačení vstupující poruchy, která byla tvořena druhým systémem zadní vrtule, tak regulátor dokáže i nadále pracovat bez trvalé regulační odchylky.

Obdobným způsobem byl testován systém zadní vrtule, který je typický svým astatickým charakterem. Ke zlepšení vlastností regulátoru byl opět využit medián filtr na signálu azimutu. V tomto případě je možné systém provozovat v celém rozsahu 270° v obou variantách regulátoru. Vstupující porucha byla analogicky tvořena druhým systémem a z hlediska jejího potlačení dokázaly oba regulátory pracovat bez trvalé regulační odchylky.

Následovala komplexní regulace vícerozměrového systému vrtulníku, kde měla být zajištěna součinnost obou motorů. V tomto případě jsem musel využít i omezení na přírůstcích akčních zásahů, čímž došlo ke zklidnění regulace během skokových změn žádaných veličin. Taktéž byl lehce pozměněn horizont predikce, který poskytuje kompromis mezi kvalitou řízení a výpočetní náročností. Navržený MPC MIMO regulátor dokáže zajistit stabilní sledování obou žádaných veličin, a to verzi s kvadratickým programováním a překvapivě i v analytické variantě, kterou lze svým přístupem považovat za suboptimální.

V poslední části jsem provedl úspěšný návrh explicitních regulátorů pro jednotlivé SISO systémy, kde jsem využil možnosti exportu modelu z prostředí MATLAB do LabVIEW. Nejprve bylo nutné rozšířit stávající stavový popis o dodatečný stav představující výstupní poruchu, jelikož systém pracoval s trvalou regulační odchylkou. Ta byla zapříčiněna již zmiňovanou vysokou vstupní necitlivostí. Problém zpoždění počáteční odezvy byl vyřešen obdobně jako u implicitní varianty s tím rozdílem, že počáteční hodnota akčního zásahu byla 8 V. V oblasti pracovního bodu lze dosažené výsledky považovat za kvalitní, s vyšší elevací dochází ke stejnému chování jako u implicitní varianty. Z hlediska regulovatelného rozsahu elevace je situace opět podobná standardnímu MPC. Vstupující porucha je i v této verzi opět vyregulována a nevyskytuje se tak trvalá regulační odchylka.

U systému zadního motoru jsem musel využít jiný druh rozšířeného popisu, jelikož předchozí varianta nedosahovala uspokojivých výsledků. Rozdíl v těchto popisech je především, jakým způsobem je interpretována poruchová veličina. Po této úpravě bylo možné zajistit regulace v celém rozsahu a bez trvalé regulační odchylky. V případě, kdy je do systému zavedena porucha, tak při její vyšší úrovni jsou pozorovány nezanedbatelné kmity. K jejím odstranění byla využita další verze stavového popisu, která kombinuje předešlé varianty. Výsledkem je jednoznačné zklidnění regulace při působení poruchy.

Úspěšně se mi tak podařilo implementovat implicitní a explicitní regulátory pro jednotlivé systémy vrtulníku. V případě implicitní verze je dále zajištěna regulace vícerozměrového systému, a to jak ve variantě s kvadratickým programováním, tak i s analytickým výpočtem. Naopak se mi nepodařilo navrhnout explicitní MPC pro vícerozměrový systém. Z velké části to přisuzuji závislosti jednotlivých subsystémů na vzorkovací periodě a horizontech predikce a řízení. Rozšíření explicitního regulátoru na použitelné řízení MIMO soustavy vrtulníku by mohlo představovat další pokračování této práce. Případně dalším námětem by mohla být realizace nelineárního či gain scheduling MPC. K tomu by bylo možné využít v práci představenou možnost exportu modelu z prostředí MATLAB.

## Literatura

- [1] PAULŮ, Dominik. *Řízení modelu vrtulníku pomocí PID regulátorů*. Liberec, 2018. Bakalářská práce. Technická univerzita v Liberci. Vedoucí práce Lukáš Hubka.
- [2] HUMUSOFT. CE 150 helicopter model: Educational Manual. *Dipartimento di Ingegneria dell'informazione e scienze matematiche* [online]. [cit. 2021-05-05]. Dostupné z:  
[http://www.dii.unisi.it/control/cmr/altro/heli\\_ce150\\_manual.pdf](http://www.dii.unisi.it/control/cmr/altro/heli_ce150_manual.pdf)
- [3] KARER, G. a B. ZUPANČIČ. Modelling and identification of a laboratory helicopter. In: *Proceedings 5th MATHMOD Vienna, February 2006*. Ljubljana.
- [4] Mnoho podob technologie NI RIO. *DPS elektronika od A po Z* [online]. 2011 [cit. 2021-5-5]. Dostupné z:  
<https://www.dps-az.cz/vyvoj/id:6122/mnoho-podob-technologie-ni-rio>
- [5] VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. *Začínáme s LabVIEW*. Praha: BEN - technická literatura, 2008. ISBN 978-80-7300-245-9.
- [6] LabVIEW for CompactRIO Developer's Guide. In: *Engineer Ambitiously - NI* [online]. [cit. 2021-05-05]. Dostupné z:  
<http://www.ni.com/cs-cz/shop/compactrio/compactrio-developers-guide.html>
- [7] Analog and Digital I/O With CompactRIO: The Ultimate Multipurpose Controller. *Engineer Ambitiously - NI* [online]. 2019 [cit. 2021-5-15]. Dostupné z: <https://www.ni.com/cs-cz/innovations/white-papers/16/analog-and-digital-i-o-with-compactrio-the-ultimate-multipurpos.html>
- [8] Understanding NI CompactRIO Scan Mode. *Engineer Ambitiously - NI* [online]. 2020 [cit. 2021-5-15]. Dostupné z: <https://www.ni.com/cs-cz/innovations/white-papers/08/understanding-ni-compactrio-scan-mode.html>
- [9] How To Use Scan Engine and FPGA Simultaneously on a CompactRIO. *Engineer Ambitiously - NI* [online]. 2020 [cit. 2021-5-15]. Dostupné z: <https://www.ni.com/cs-cz/innovations/white-papers/01/How-to-use-scan-engine-and-FPGA-simultaneously.html>

- [10] VROŽINA, Milan, Zora JANČÍKOVÁ a Jiří DAVID. *Identifikace systémů: učební text* [online]. Ostrava: Vysoká škola báňská - Technická univerzita, 2012 [cit. 2021-5-15]. ISBN 978-80-248-2594-6. Dostupné z: <http://www.person.vsb.cz/archivcd/FMMI/IS/Identifikace%20systemu.pdf>
- [11] MODRLÁK, Osvald a Lukáš HUBKA. *Automatické řízení: učební text*. Liberec: Technická univerzita v Liberci, 2012. ISBN ISBN978-80-7372-850-2.
- [12] CANNON, Mark. *Discrete Systems* [online]. University of Oxford [cit. 2021-5-15]. Dostupné z: <https://www.coursehero.com/file/60971350/DCS-Lecture1pdf/>
- [13] RAGHU, Neha. *Model Predictive Control: History and Development* [online]. Patiala, 2013 [cit. 2021-5-15]. ISSN 2231-5381. Dostupné z: <http://ijettjournal.org/volume-4/issue-6/IJETT-V4I6P173.pdf>
- [14] CAMACHO, E. F. a Carlos BORDONS. *Model predictive control*. 2nd ed. London: Springer, c2007. Advanced textbooks in control and signal processing (Springer). ISBN 978-1-85233-694-3.
- [15] WANG, Liuping. *Model predictive control system design and implementation using MATLAB* [online]. New York: Springer, c2009 [cit. 2021-5-15]. Advances in industrial control. ISBN 978-1-84882-331-0.
- [16] KOPAL, Martin. *Explicitní prediktivní řízení a jeho aplikace* [online]. Liberec, 2014. Diplomová práce. Technická univerzita v Liberci. Vedoucí práce Lukáš Hubka.
- [17] ALESSIO, Alessandro a Alberto BEMPORAD. *A Survey on Explicit Model Predictive Control* [online]. Berlin: Springer, 2009 [cit. 2021-5-15]. ISBN 978-3-642-01094-1.
- [18] Compiling a Model from The MathWorks, Inc. Simulink® Software (Model Interface Toolkit). *Engineer Ambitiously - NI* [online]. 2017 [cit. 2021-5-15]. Dostupné z: [https://zone.ni.com/reference/en-XX/help/374160C-01/vsmithelp/mit\\_convert\\_model\\_to\\_dll/](https://zone.ni.com/reference/en-XX/help/374160C-01/vsmithelp/mit_convert_model_to_dll/)
- [19] Using Models from The MathWorks, Inc. Simulink® Software (Model Interface Toolkit). *Engineer Ambitiously - NI* [online]. 2017 [cit. 2021-5-15]. Dostupné z: [https://zone.ni.com/reference/en-XX/help/374160C-01/vsmithelp/mit\\_model\\_from\\_mdl/](https://zone.ni.com/reference/en-XX/help/374160C-01/vsmithelp/mit_model_from_mdl/)
- [20] ŠANTIN, Ondřej. *Influence of Model Uncertainty on Constraints Handling in Predictive Control* [online]. Praha, 2009 [cit. 2021-5-15]. Dostupné z: [https://wiki.control.fel.cvut.cz/mediawiki/images/3/3b/Dp\\_2009\\_santin\\_ondrej.pdf](https://wiki.control.fel.cvut.cz/mediawiki/images/3/3b/Dp_2009_santin_ondrej.pdf). Diplomová práce. České vysoké učení technické v Praze.



## A Identifikace

```
1 clc; close all; clear all
2 global tG uG yG
3 %% vstupni data
4 im_data=importdata('data_k_identifikaci.txt');
5 u_id=im_data.data(:,1); %napeti hlavni motor
6 y_id=im_data.data(:,3); %elevace
7 t_id=im_data.data(:,5); % cas
8 tG=t_id;
9 %% pracovní bod
10 y0=y_id(1);
11 u0=u_id(1);
12 yG=y_id-y0;
13 uG=u_id-u0; %posun pracovniho bodu
14 %% struktura obrazoveho řpenosu
15 K=6;
16 A=[0.03 0.4 0.9]; % pocatecni odhad
17 F=tf(K,[A 1]) % řpenosová funkce
18 x=[K A];
19 J_poc=critT_hlavni(x)
20 %% hldani optimalnich parametru
21 OPTIONS=optimset('MaxIter',600,'TolX',1e-6);
22 x=fminsearch('critT_hlavni',x,OPTIONS);
23 %% nalezeny prenos
24 J_kon=critT_hlavni(x)
25 KK=x(1);
26 AA=[x(2:end) 1];
27 FF=tf(KK,AA);
28
29
30
31
32 function J= critT_hlavni(x)
33 global tG yG uG
34 A=[x(2:end) 1];
35 sys=tf(x(1),A);
36 [yi , ti]=lsim(sys ,uG,tG);
37 J=sum((yG- yi) . *(yG- yi));
38 end
```

## B Návrh implicitního MPC

```
1  clc;close all;clear all
2  %% Definice systemu, diskretizace, minimalni realizace
3  s=tf('s')
4  Ts=0.01;
5  G11=tf(5.607,[0.03742 0.1652 0.4561 1]);
6  G12=tf(0.3313,[0.02179 0.1148 0.4234 1]);
7  G21=tf(-0.00249,[0.001303 0.003399 0]);
8  G22=tf(-53.53,[14.25 11.6 1 0]);
9  G=[G11 G12;G21 G22];
10 sys_discrete=c2d(G,Ts); %diskretizace systemu
11 sys=ss(sys_discrete);
12 sys=minreal(sys); % minimalni realizace
13 [Ap,Bp,Cp,Dp]=ssdata(sys);
14 %% parametry implicitniho MPC
15 Np=450; % horizont predikce
16 Nc=2; %horizont rizeni
17 lmd=0.01; % vaha akcnich zasahu
18 Q=3*eye(2);
19 R=1*eye(2); % kovariancni matice sumu
20 [kest,L,P] = kalman(sys,Q,R); % zesileni estimatoru
21 [m1,n1]=size(Cp);
22 [n1,n_in]=size(Bp);
23 A_e=eye(n1+m1,n1+m1);
24 A_e(1:n1,1:n1)=Ap;
25 A_e(n1+1:n1+m1,1:n1)=Cp*Ap; % rozsirena matice A
26
27 B_e=zeros(n1+m1,n_in);
28 B_e(1:n1,:)=Bp;
29 B_e(n1+1:n1+m1,:)=Cp*Bp; % rozsirena matice B
30
31 C_e=zeros(m1,n1+m1);
32 C_e(:,n1+1:n1+m1)=eye(m1,m1); % rozsirena matice C
33 n=n1+m1;
34 h(1:m1,:)=C_e;
35 F(1:m1,:)=C_e*A_e;
36 %inicializace matic Phi a F
37 for kk=m1+1:m1*Np-1
38     h(kk:kk+m1-1,:)=h(kk-m1:kk-1,:)*A_e;
39     F(kk:kk+m1-1,:)=F(kk-m1:kk-1,:)*A_e;
40 end
41 v=h*B_e;
```

```

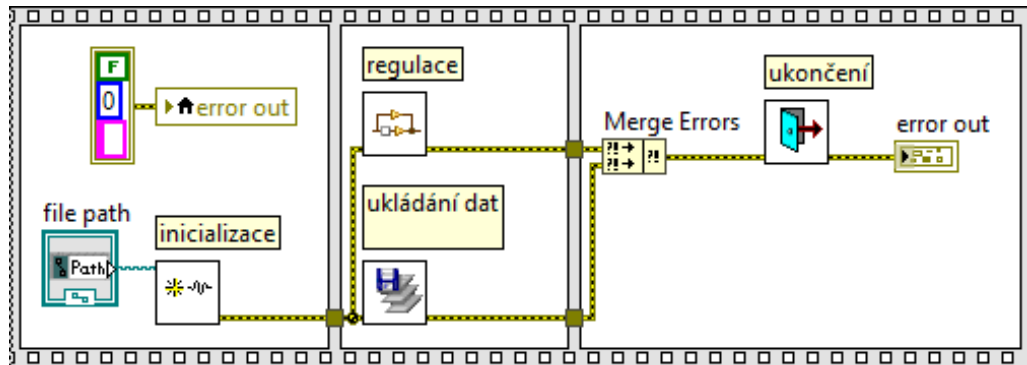
42 Phi=zeros(2*Np,2*Nc)
43 Phi(:,1:m1)=v;
44 j=1;
45 for i=m1+1:m1:m1*Nc
46     Phi(:,i:i+1)=[zeros(j*m1,1*m1);v(1:m1*Np-m1*j,1:m1)];
47     j=j+1;
48 end
49 Phi_Phi=Phi'*Phi;
50 Phi_F=Phi'*F;
51 Phi_R=Phi_F(:,end-m1+1:end);
52 Q=2*(Phi_Phi+lmd*eye(m1*Nc,m1*Nc)); % vypocet matice Q pro funkci ...
    quadprog
53
54 % zesileni MPC regulátoru
55 Ky=[eye(m1,m1) ...
    zeros(2,m1*(Nc-1))] * inv(Phi_Phi+lmd*eye(m1*Nc,m1*Nc)) * (Phi_R);
56 Kmpc=[eye(2,2) ...
    zeros(2,m1*(Nc-1))] * inv(Phi_Phi+lmd*eye(m1*Nc,m1*Nc)) * (Phi_F);

```

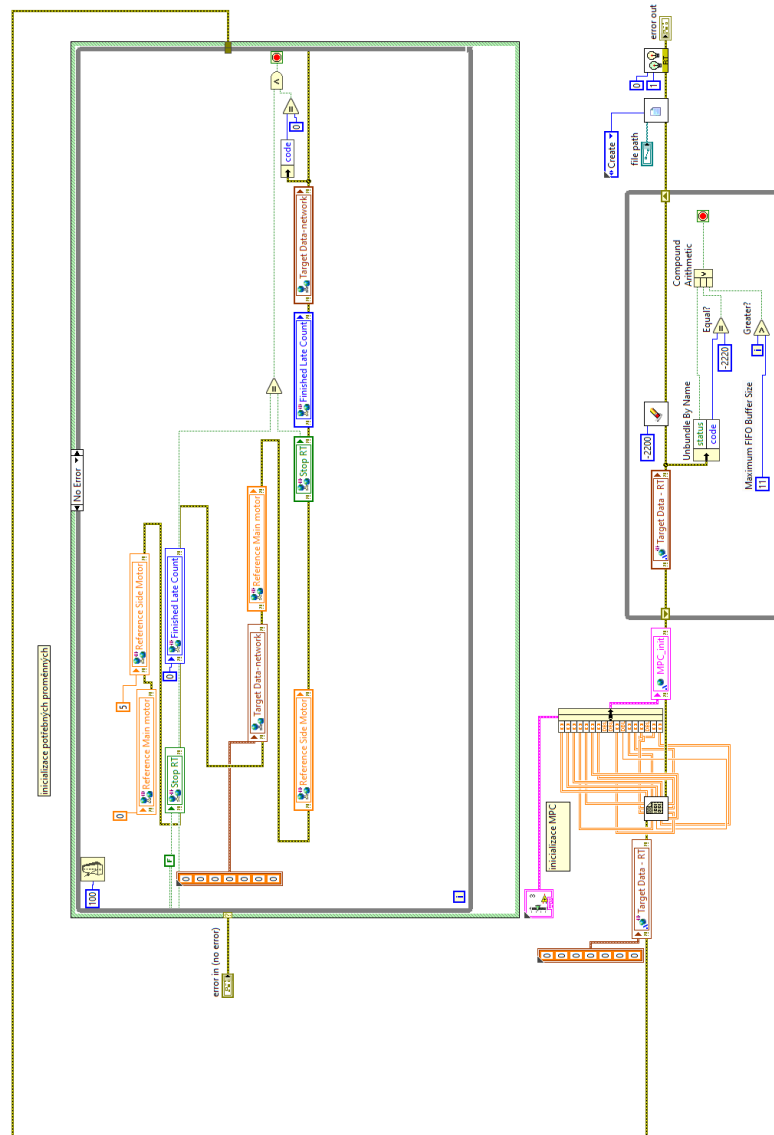
## C Návrh explicitního MPC

```
1 clear all; close all; clc;
2 %% Definice systemu, diskretizace
3 Ts=0.015; %vzorkovaci perioda
4 G=tf(5.607,[0.03742 0.1652 0.4561 1]);
5 sys_discrete=c2d(G,Ts); %diskretizace systému
6 sys=ss(sys_discrete);
7 sys=minreal(sys);
8 [Ap,Bp,Cp,Dp]=ssdata(sys);
9 %% parametry MPC
10 Aaug=[Ap zeros(3,1); 0 0 0 1]; %rozsirena matice A
11 Baug=[Bp;0]; %rozsirena matice B
12 Caug=[Cp 1]; %rozsirena matice C
13 sys_aug=ss(Aaug,Baug,Caug,0,Ts); %rozsireny stavovy popis
14 clear limits
15 limits.umin=-5;
16 limits.umax=5; %limity velikosti akcni veliciny
17 limits.dumin=-1.5;
18 limits.dumax=1.5; %limity velikosti prirutku akcni veliciny
19 clear cost
20 cost.S=10; %vaha regulacni odchylky
21 cost.T=0.1; %vaha prirutku akcni veliciny
22 cost.rho=Inf;
23 clear interval
24 interval.N=180; %horizont predikce
25 interval.Nu=4; %horizont rizeni
26 %% Návrh MPC regulátoru
27 Reg_imp=lincon(sys_aug,'track',cost,interval,limits); %implicitni verze
28 clear range
29 range.xmax=[10000;10000;10000;10000];
30 range.xmin=[-10000;-10000;-10000;-10000]; %rozsahy stavu
31 range.umax=limits.umax;
32 range.umin=limits.umin; %rozsahy akcnich velicin
33 range.refymax=10;
34 range.refymin=0;%rozsah referencniho signalu
35 clear options
36 options.verbose=1;
37 options.reltol=1e-6;
38 options.uniteeps=1e-3;
39 options.qpsolver='qpact';
40 Reg_exp=expcon(Reg_imp,range,options); %explicitni regulator
```

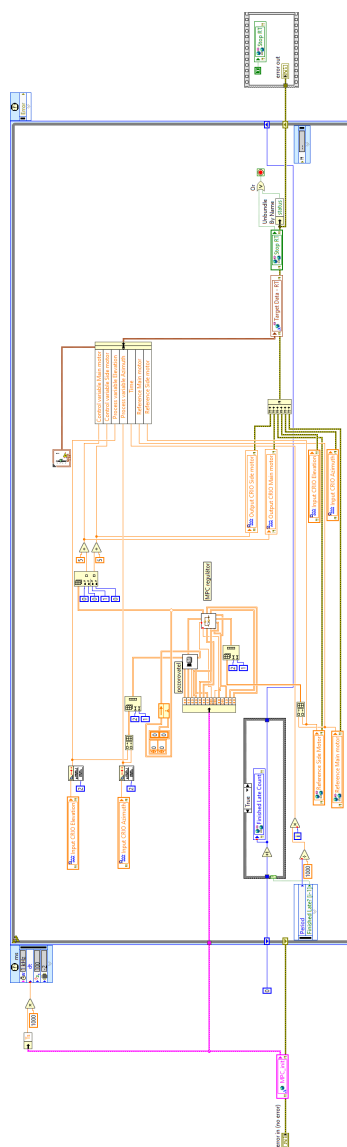
## D Struktura aplikace VI



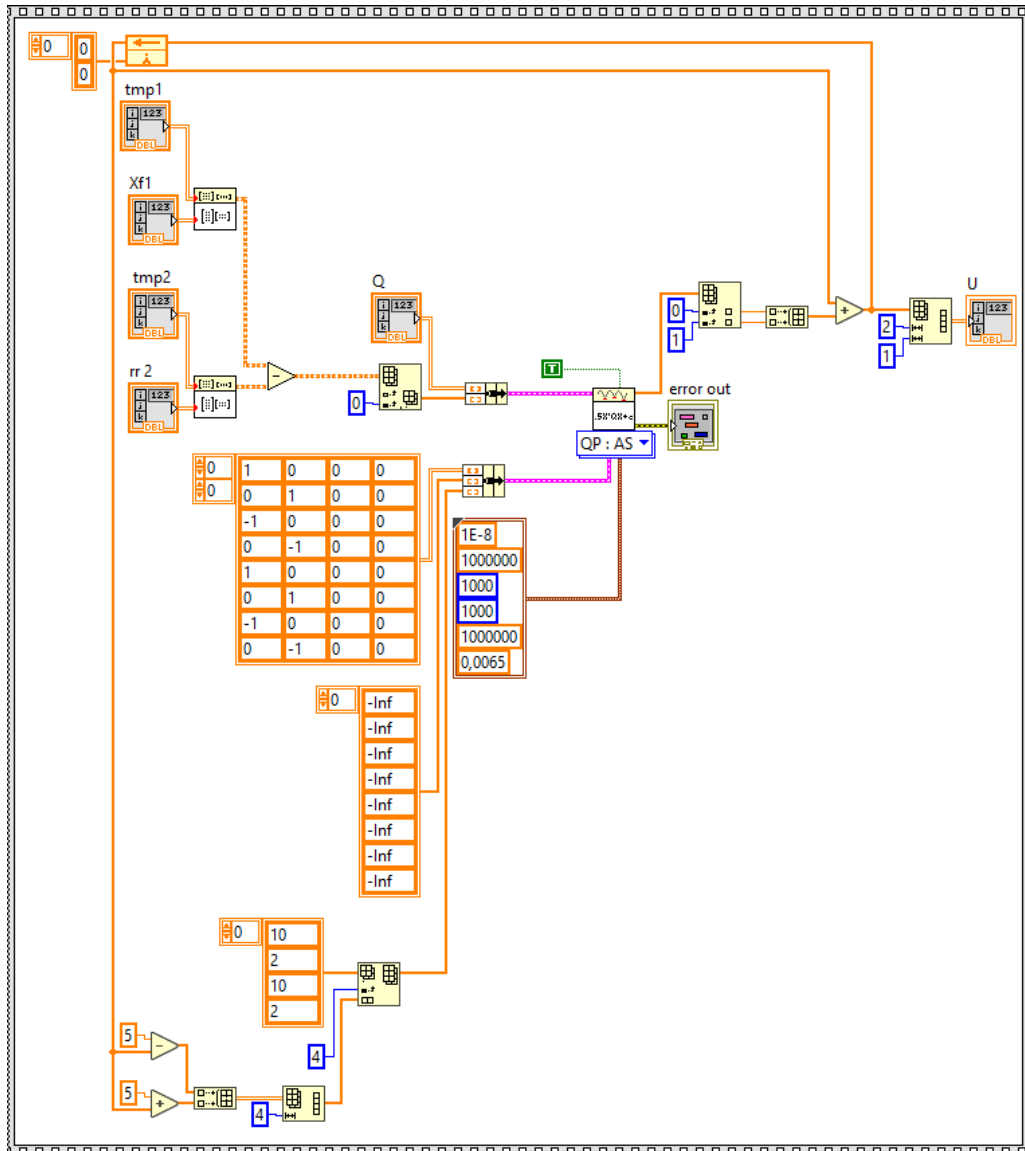
# E Inicializace VI



## F Řídicí část VI

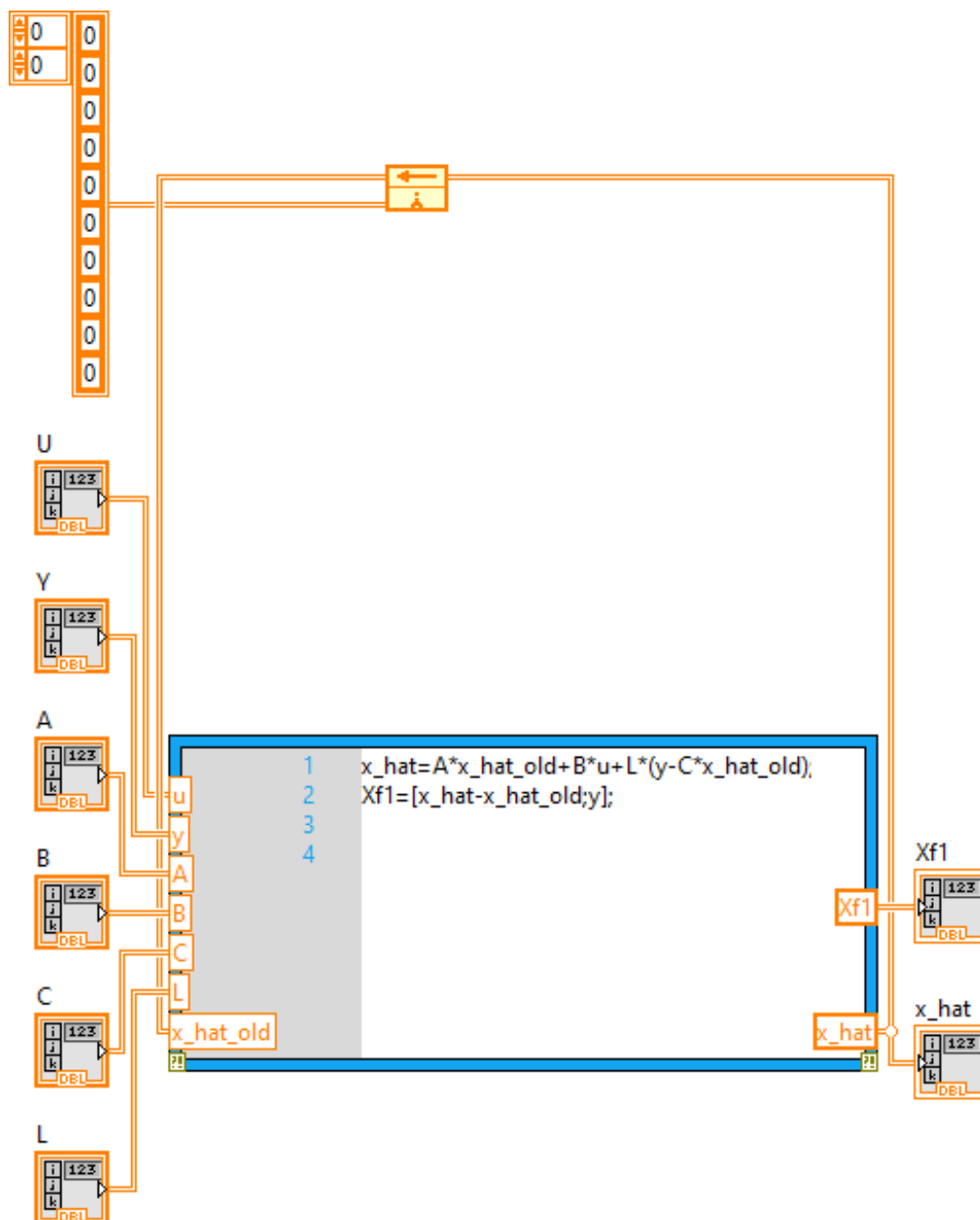


## G Prediktivní regulátor VI

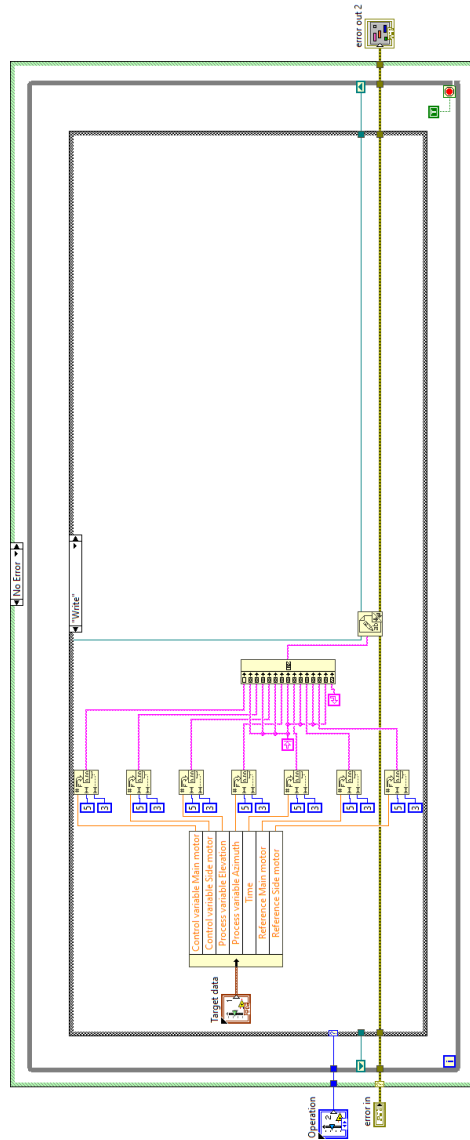




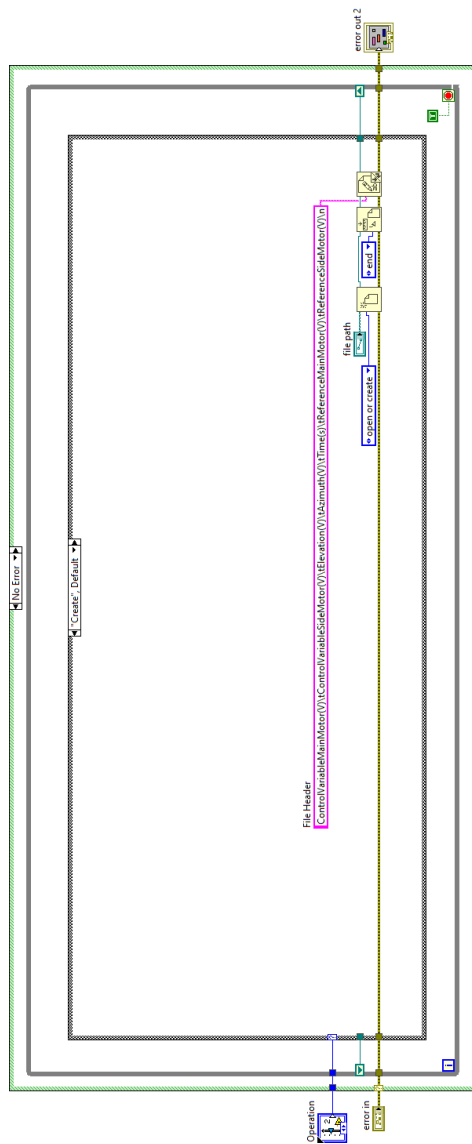
## H Stavový pozorovatel VI



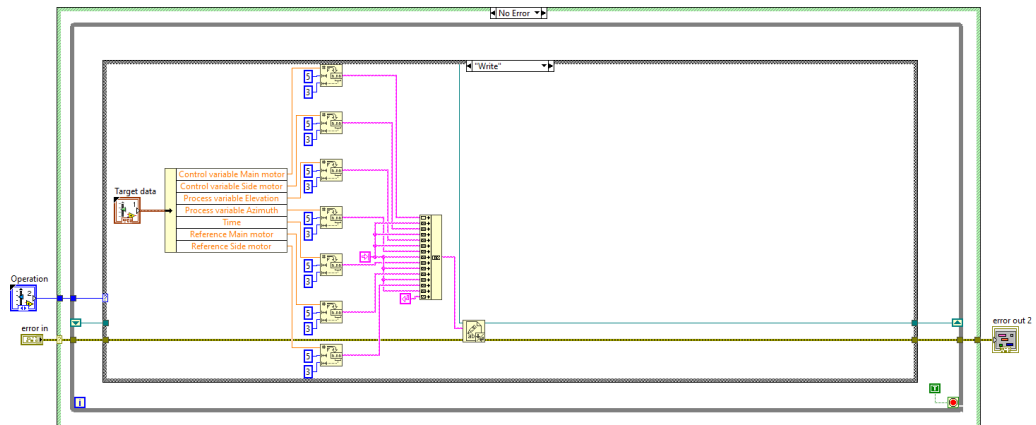
# I Logování dat VI



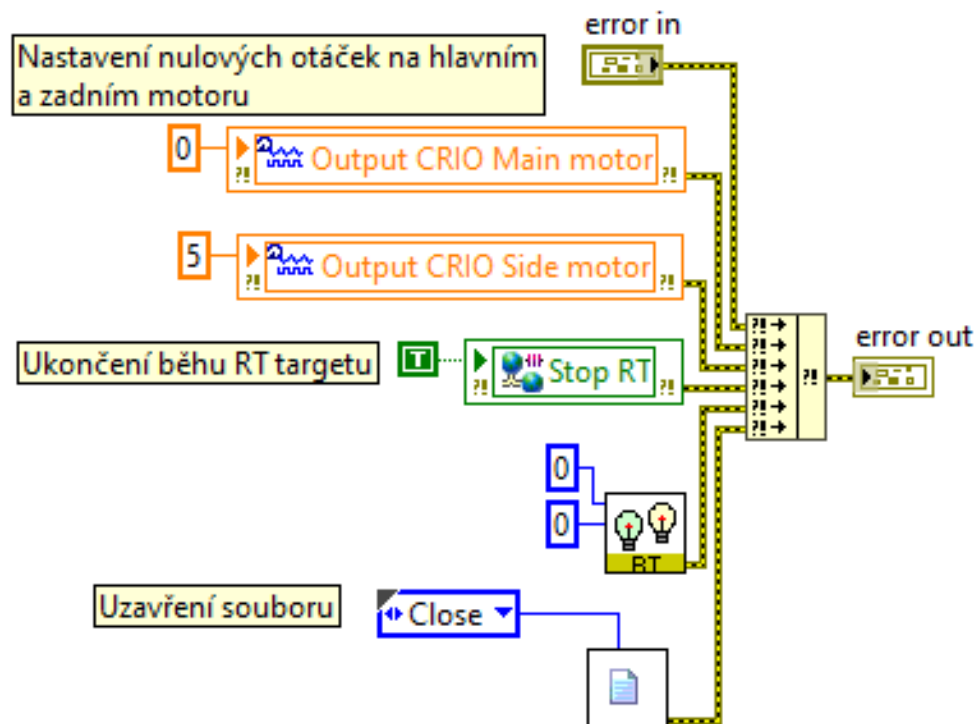
## J Vytvoření logu VI



## K Data k zápisu VI

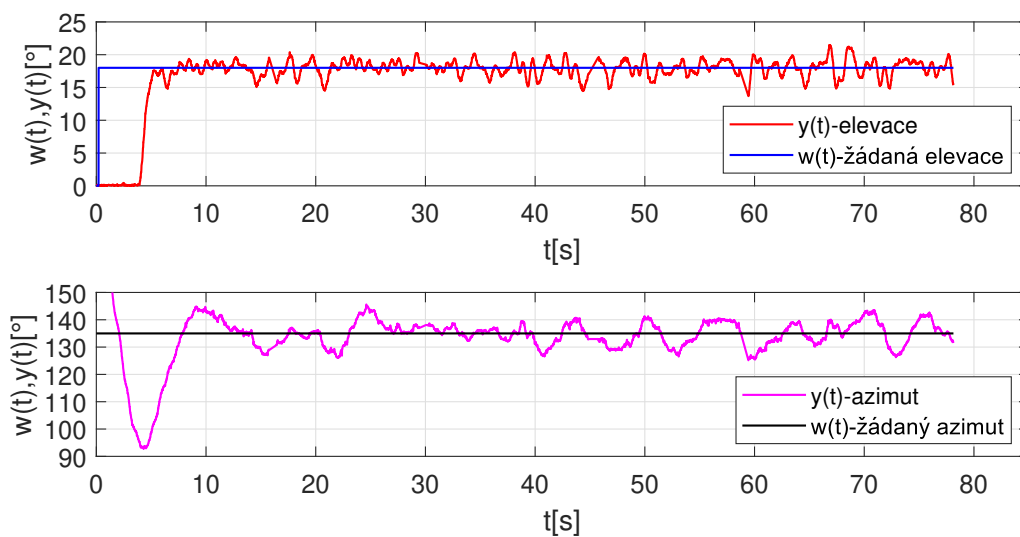


## L Shutdown VI

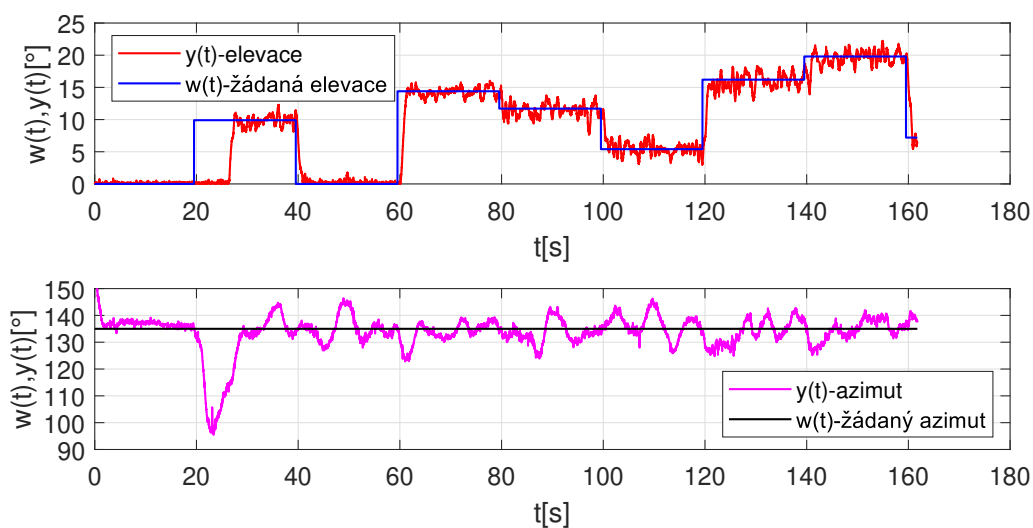


## M Grafy analytické verze MIMO MPC

MIMO regulace na konstantní hodnoty



MIMO regulace analyticky proměnná elevace



### MIMO regulace analyticky proměnný azimut

