

# **Prediktivní řízení s omezením vstupních a výstupních veličin**

Predictive control with constrains of input and output values

Bc. Jiří Hubáček

---

Diplomová práce  
2008



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav řízení procesů

akademický rok: 2007/2008

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jiří HUBÁČEK**

Studijní program: **N 2807 Chemické a procesní inženýrství**

Studijní obor: **Automatizace a řídicí technika**

Téma práce: **Prediktivní řízení s omezením vstupních a výstupních veličin**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Navrhněte prediktivní řídicí algoritmus ve smyslu GPC pro systém popsany diskretní přenosovou funkcí 2 řádu.
3. Řídicí algoritmus doplňte o omezení vstupní a výstupní veličiny.
4. Simulačně ověřte navržené algoritmy v prostředí Matlab/Simulink.
5. Navržený algoritmus ověřte při řízení vybraného laboratorního modelu v reálném čase.

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **Maciejowski, J. M.: Predictive Control with Constraints, Prentice Hall, 2002.**
2. **Camacho, E. F. and C. Bordons: Model Predictive Control, Springer-Verlag London, 1999.**
3. **Kanjilal, P. P.: Adaptive Prediction and Predictive Control, IEE Control Engineering Series, 1995.**
4. **Mikleš, J., Fikar, M.: Modelovanie, identifikácia a riadenie procesov 2. Identifikácia a optimálne riadenie. Slovenská technická univerzita v Bratislavě, 2004.**
5. **Giesl, P.: Prediktivní řízení laboratorních modelů. Diplomová práce, UTB ve Zlíně, FAI, 2006.**

Vedoucí diplomové práce: **prof. Ing. Vladimír Bobál, CSc.**  
Ústav řízení procesů

Datum zadání diplomové práce: **22. února 2008**

Termín odevzdání diplomové práce: **21. května 2008**

Ve Zlíně dne 22. února 2008

prof. Ing. Vladimír Vašek, CSc.  
*děkan*



prof. Ing. Petr Dostál, CSc.  
*ředitel ústavu*

## **ABSTRAKT**

Prvním cílem této diplomové práce bylo vypracovat literární rešerši na problematiku týkající se prediktivního řízení bez i s omezením vstupních a výstupních veličin.

Dalším cílem bylo vytvoření prediktivního řídicího algoritmu ve smyslu GPC pro systém popsany diskretní přenosovou funkcí 2. řádu v programovém prostředí Matlab a dále takto vytvořený algoritmus rozšířit o omezení vstupní a výstupní veličiny.

Posledním cílem bylo tyto vytvořené algoritmy ověřit simulačně v prostředí Matlab/Simulink a následně je ověřit při řízení vybraného laboratorního modelu v reálném čase.

Klíčová slova: Prediktivní řízení, Matlab, Simulink, omezení vstupních a výstupních veličin

## **ABSTRACT**

The first goal of this master thesis was elaboration of a retrieval concerning problems of both constrained and unconstrained predictive control.

The next goal was design of a GPC control algorithm for a model of the controlled process which is described by second order discrete transfer function. Further task was to modify the control algorithm so that it could handle constraints of input and output signals of the controlled process. The task was to implement the algorithm into Matlab.

The last goal was to verify the designed algorithms both by simulation in the Matlab/Simulink environment and by real time control of a chosen laboratory model.

Keywords: Predictive control, Matlab, Simulink, constrains of input and output values

Děkuji vedoucímu své diplomové práce prof. Ing. Vladimírovi Bobálovi, CSc. za odborné vedení, podnětné připomínky a rady udílené při vypracování práce. Dále děkuji doc. Ing. Marku Kubalčíkovi, Ph.D. za konzultace a rady ohledně prediktivního řízení.

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

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

Ve Zlíně, 3. 6. 2008

.....

Jiří Hubáček

## OBSAH

ÚVOD.....	7
<b>I</b> <b>TEORETICKÁ ČÁST</b> .....	<b>9</b>
1 <b>HISTORIE AUTOMATIZACE A PREDIKTIVNÍHO ŘÍZENÍ</b> .....	<b>10</b>
2 <b>ÚVOD DO PREDIKTIVNÍHO ŘÍZENÍ</b> .....	<b>11</b>
3 <b>STRUKTURA A PRINCIP PREDIKTIVNÍHO ŘÍZENÍ</b> .....	<b>12</b>
4 <b>SLOŽKY PREDIKTIVNÍHO ŘÍZENÍ</b> .....	<b>14</b>
4.1    MODELÝ.....	14
4.1.1    Impulsní funkce.....	14
4.1.2    Přechodová funkce .....	16
4.1.3    Přenosová funkce .....	17
4.1.4    ARX model .....	18
4.1.5    Stavový popis .....	19
4.1.6    Jiné modely .....	21
4.2    ÚČELOVÁ FUNKCE.....	21
4.3    OMEZUJÍCÍ PODMÍNKY .....	24
4.3.1    Tvrdá omezení.....	25
4.3.2    Měkká omezení .....	27
<b>5</b> <b>ODVOZENÍ A REALIZACE PREDIKTIVNÍHO ŘÍZENÍ</b> .....	<b>29</b>
5.1    ODVOZENÍ PREDIKTORU .....	29
5.2    VÝPOČET OPTIMÁLNÍHO ŘÍZENÍ.....	32
5.3    ODVOZENÍ PREDIKTORU .....	34
<b>6</b> <b>IDENTIFIKACE</b> .....	<b>38</b>
6.1    METODA NEJMENŠÍCH ČTVERCŮ.....	38
6.2    REKURZIVNÍ METODA NEJMENŠÍCH ČTVERCŮ.....	39
<b>II</b> <b>PRAKTICKÁ ČÁST</b> .....	<b>42</b>
7 <b>POPIS LABORATORNÍHO MODELU AMIRA DR300</b> .....	<b>43</b>
8 <b>URČENÍ HORIZONTŮ PREDIKTIVNÍHO ŘÍZENÍ</b> .....	<b>46</b>
8.1    STATICÁ CHARAKTERISTIKA .....	46
8.2    PŘECHODOVÁ CHARAKTERISTIKA .....	48
9 <b>IDENTIFIKACE LABORATORNÍHO MODELU AMIRA DR300</b> .....	<b>54</b>
10 <b>SIMULACE A REÁLNÉ MĚŘENÍ PREDIKTIVNÍHO ŘÍZENÍ BEZ</b> <b>OMEZENÍ</b> .....	<b>60</b>

10.1	SIMULACE PREDIKTIVNÍHO ŘÍZENÍ BEZ OMEZENÍ NA MODELU ZÍSKANÉM IDENTIFIKACÍ.....	60
10.2	PREDIKTIVNÍ ŘÍZENÍ BEZ OMEZENÍ NA REÁLNÉM MODELU.....	63
10.3	SIMULACE PREDIKTIVNÍHO ŘÍZENÍ S OMEZENÍM NA MODELU ZÍSKANÉM IDENTIFIKACÍ.....	66
10.4	PREDIKTIVNÍ ŘÍZENÍ S OMEZENÍM NA REÁLNÉM MODELU .....	70
	<b>ZÁVĚR.....</b>	<b>73</b>
	<b>ZÁVĚR V ANGLIČTINĚ.....</b>	<b>74</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>75</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>77</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>79</b>
	<b>SEZNAM TABULEK.....</b>	<b>81</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>82</b>

## ÚVOD

V současné době je výpočetní technika důležitým aspektem pro úspěch člověka v téměř všech odvětvích lidského snažení, obzvláště důležitou roli hraje v komplexním vývoji informačních technologií. Používá se převážně v oblastech, kde je důležité zpracovávání dat, které by bylo klasickými metodami časově a technicky náročné. Díky použití výpočetní techniky v těchto oblastech se tedy dosahuje snížení celkových nákladů. Mezi tyto oblasti patří zejména grafické simulace průběhu procesu na základě vložených informací, dále pak složité výpočetní operace, rozsáhlé databáze dat a jiné operace, které by bez pomoci programů navržených pro tyto účely byly dnes již snad neřešitelné. V dnešní době nelze již pokládat softwarové vybavení výpočetní techniky jen za pomocné nástroje při zpracovávání dat, ale jejich úroveň je již tak velká, že jsou schopny samostatného řešení zadaných úloh, jejich kontroly a návrhu, provozu, řízení a ovládání [10], [11].

Jedním z oborů, kde výpočetní technika hraje obzvláště důležitou roli, je automatizace a řízení. V některých fázích dokonce zcela zastupuje lidský faktor. Jejím použitím dochází k usnadnění návrhu regulátorů a jejich parametrů, mezi které patří volba typu regulátoru a jeho struktury, atd. Výpočetní technika má své nezastupitelné místo i při průběžné analýze, simulaci, monitorování atd. [10], [11].

V oblasti automatizace a řízení se využívá velké množství programů sloužící např. k regulaci měřené soustavy, simulaci regulačního pochodu, atd. Jedním z nejpoužívanějších programů, které tyto operace provádí, je bezesporu MATLAB. Tento program obsahuje velké množství nástrojů, pomocí kterých je schopen na základě vložených dat provádět numerické či grafické řešení zadaných úloh, identifikaci systému, jeho řízení, optimalizaci a další [10], [11].

Mezi nejrozšířenější regulátory v oblasti automatizace a řízení patří bezesporu PID regulátory a to zejména díky jejich jednoduchosti. V praxi se objevují jak v analogové tak i v číslicové formě. Existuje u nich spousta nedostatků, obzvláště u systému s dopravním zpožděním, při kompenzaci poruch, apod. Rozvoj číslicové techniky umožnil návrh nových struktur regulátorů, což vedlo k zlepšení této situace [12].

Stávající algoritmy byly však stále navrhovány a odvozovány pro spojité systémy, ale postupem času se díky výzkumu objevily i algoritmy odvozené pro diskrétní systémy.



A právě modelování procesů v diskrétní formě umožnilo předpovídat budoucí hodnoty výstupu procesu. Tato strategie je považována za předchůdce současného prediktivního řízení, které umožňuje plné využití predikce výstupu systému pro řízení v reálném čase [12].

Tato práce se podrobněji zabývá problematikou týkající se prediktivního řízení s omezením vstupních a výstupních veličin v oblasti automatického řízení a nezbytnou průběžnou identifikací dynamicky se měnícího přenosu soustavy. Poslední částí je tvorba programu v programovém prostředí MATLABu sloužícího pro prediktivní řízení reálných systémů a dále otestování tohoto programu na reálném modelu v laboratořích.

## **I. TEORETICKÁ ČÁST**

## 1 HISTORIE AUTOMATIZACE A PREDIKTIVNÍHO ŘÍZENÍ

Počátky automatizace v novodobé historii se opírají o empirické metody a datují se zejména od zavedení odstředivého regulátoru otáček parního stroje Jamesem Wattem v roce 1784. Ve stejném století v Rusku Polzunov zrealizoval regulaci výšky hladiny v kotlích. Tato disciplína se neustále více matematizuje obzvláště v druhé polovině 19. století a to zejména díky E.J.Routhovi (1831-1907), A.Hurwitzovi (1858-1919) a A.M.Ljapunovovi (1857-1918), který se zasloužil o vypracování obecné teorie stability. V první půli 20. století vypracovali H.Nyquist a H.W.Bode frekvenční metody analýzy systémů, J.G.Ziegler a N.B.Nichols [14] efektivní metody nastavení regulátorů. Obrovský rozvoj teorie řízení nastal po druhé světové válce, kdy se efektivní, optimální a bezpečné řízení stalo součástí nejen průmyslových projektů, ale i nezastupitelnou složkou v oblasti strategických zbraní, navigace a ovládání leteckých a kosmických strojů. Významných osobností v tomto období je spousta, např. R.Bellman (dynamické programování - 1956), L.P.Pontrjagin (princip optimálnosti), R.E.Kalman (filtrace a predikce) [13].

Právě do tohoto období po druhé světové válce se datuje vznik prediktivního řízení. MPC (Model Predictive Control) [15] technologie se v praxi objevuje daleko dříve, než vznikla první publikace na toto téma. Pravděpodobně prvním, kdo v praxi použil algoritmus prediktivního řízení, byl Propoi v roce 1963. A až o 15 let později, tedy v roce 1978, byla prvně publikována popsaná aplikace prediktivního řízení (Richalet a kol.). K predikci byla použita impulsní posloupnost a optimální vstupy byly vypočítány iterativním způsobem, založeným na heuristickém přístupu. V roce 1980 představili Cutler a Remarker algoritmus pro mnohorozměrné systémy pod zkratkou DMC (Dynamic Matrix Control). Pro řešení optimalizačních problémů zde byla použita metoda nejmenších čtverců. Oba výše uvedené algoritmy v sobě nezahrnovaly omezení vstupních a výstupních veličin. O pár let později v roce 1983 se objevuje algoritmus QDMC (Quadratic Dynamic Matrix Control), který již řeší omezení těchto veličin, avšak není stále použitelný pro nestabilní systémy a to díky tomu, že používá přechodové a impulsní posloupnosti. V roce 1988 byl popsán vztah mezi MPC a stavovým popisem v SMOC (Shell Multivariable Optimizing Controller). Právě SMOC dokáže pomocí Kalmanova filtru odhadnout neměřitelné poruchy. V dnešní době se používá velmi oblíbený GPC algoritmus [6], [7], který využívá přenosovou funkci a je také vhodný pro adaptivní prediktivní řízení. [12]

## 2 ÚVOD DO PREDIKTIVNÍHO ŘÍZENÍ

MPC získává stále větší popularitu díky tomu, že poskytuje systematický přístup k řízení průmyslových procesů s omezeními na vstupu, výstupu i jiných proměnných v řídicím programu.

MPC patří do skupiny, které používají model procesu k predikci budoucích odezev systému. Používáním této predikce jsou tyto algoritmy schopny vypočítat optimální trajektorii vstupních hodnot v každém okamžiku periody vzorkování na základě minimalizace účelové funkce na horizontu predikce, která vyjadřuje požadavky, které jsou kladené na regulační pochod. Výstupem tohoto optimalizačního problému je optimální řešení na horizontu predikce.

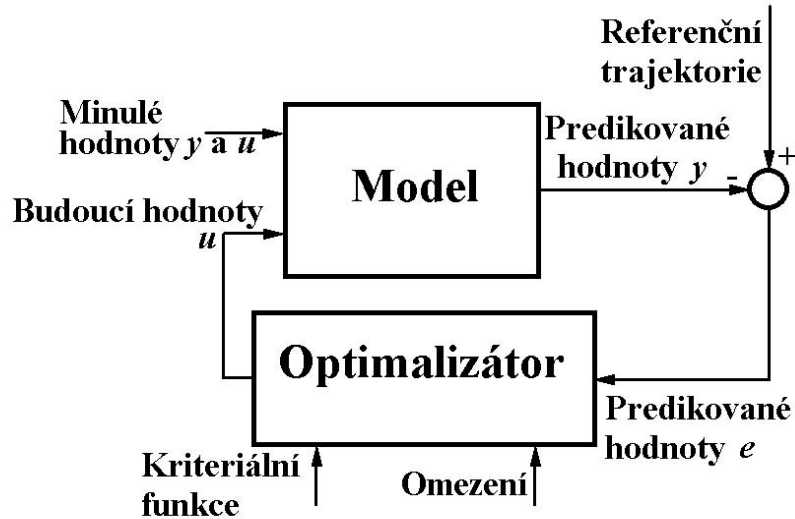
Pod pojmem prediktivní řízení si lze představit velké množství metod, které mají několik společných základních rysů:

- Již dopředu je známa trajektorie žádané veličiny,
- matematický model systému se používá na predikci budoucího akčního zásahu,
- výpočet posloupnosti budoucích akčních zásahů zahrnuje i minimalizaci účelové funkce s budoucími hodnotami přírůstků akčního zásahu,
- pouze první akční zásah je realizovaný a celý postup minimalizace funkcionálu se opakuje v další periodě vzorkování, což se nazývá jako strategie pohyblivého horizontu.

V poslední době se prediktivní řízení začíná hodně rozvíjet ani ne tak kvůli tomu, že je možné uvažovat s omezeními akční a regulované veličiny již při návrhu regulátoru, ale spíše díky tomu, že MPC algoritmy jsou při řízení procesu mnohostranně použitelné a robustní. Ve srovnání s klasickými PID regulátory je kvalita řízení u prediktivního řízení vyšší. Lze tyto MPC algoritmy také použít na neminimálně fázové, nestabilní a mnohorozměrné systémy, či systémy s dopravním zpožděním včetně řízení systémů nelineárních [5]. Jejich rozšíření je také možné zásluhou výkonnějšího hardwaru, který umožňuje řešení optimalizačních problémů přímo “online” a tím je implementovat přímo v praxi [2], [3], [4], [8].

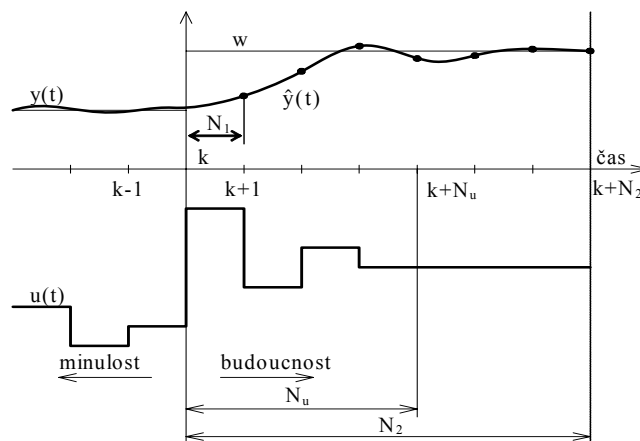
### 3 STRUKTURA A PRINCIP PREDIKTIVNÍHO ŘÍZENÍ

Základní struktura prediktivního řízení je zobrazena na obrázku 3.1:



Obrázek 3.1 Základní struktura systému prediktivního řízení

Princip prediktivního řízení je uveden na obrázku 3.2:



Obrázek 3.2 Princip prediktivního řízení

a je následující:

- Model řízeného procesu je explicitní součástí regulátoru a je používán na predikci  $N$  budoucích výstupů procesu  $\hat{y}$ ; tyto predikce jsou vypočítány vzhledem k informacím dostupným do časového okamžiku  $k$  a vzhledem k neznámé trajektorii akčních zásahů, kterou je třeba určit,

- trajektorie budoucích akčních zásahů je určena z řešení optimalizačního problému, který obsahuje vhodnou účelovou funkci a omezení vstupních a výstupních veličin; v účelové funkci jsou zahrnuty budoucí hodnoty výstupu, žádané veličiny a akčních zásahů,
- pro řízení procesu je však použit jen první člen trajektorie akčního zásahu, která byla počítána v předešlém kroku; celý postup se periodicky opakuje v každém kroku periody vzorkování, což se nazývá jako strategie pohyblivého horizontu [2], [3], [4], [8].

## 4 SLOŽKY PREDIKTIVNÍHO ŘÍZENÍ

### 4.1 Modely

V oblasti prediktivního řízení lze používat libovolný model procesu, u kterého je však podmínkou, aby dostatečně vystihoval dynamické vlastnosti procesu. U lineárních modelů lze, bez použití omezení, vypočítat trajektorii akčních zásahů, a proto se v dnešní době také nejčastěji používají. Model procesu je potřebný právě pro výpočet predikce budoucích hodnot na výstupu. Některé modely v sobě přímo obsahují modely poruch, v jiných se předpokládá, že poruchy jsou konstantní. [8]

Následně jsou uvedeny některé z těchto modelů.

#### 4.1.1 Impulsní funkce

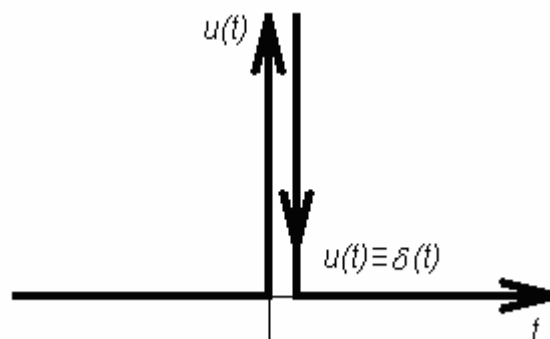
Je to model procesu založený na posloupnosti impulsní funkce, což je vlastně odezva systému na vstupní jednotkový (Diracův) impuls při nulových počátečních podmínkách systému.

Jednotkový (Diracův) impuls se označuje jako  $\delta(t)$  a je definován

$$\delta(t) = 0 \quad \text{pro } t \neq 0 \quad (4.1)$$

$$\int_{-\infty}^{\infty} \delta(t) dt = 1 .$$

Graficky:



Obrázek 4.1 Jednotkový (Diracův) impuls

Jednotkový (Diracův) impuls je idealizovaná funkce, která je fyzikálně nerealizovatelná.

Tento model je z praktických důvodů omezen jen na několik prvních členů a označuje se jako FIR (Finite Impulse Response). Vztah mezi vstupem a výstupem je popsán rovnicí

$$y(k) = \sum_{i=1}^N h_i u(k-i) = H(z^{-1})u(k). \quad (4.2)$$

kde:

- $H(z^{-1}) = h_1 z^{-1} + h_2 z^{-2} + \dots + h_n z^{-n}$  - je polynom  $n$ -tého stupně,
- $y(k)$  - je výstup systému,
- $u(k)$  - je vstup systému,
- $h_i$  - jsou koeficienty impulsní posloupnosti,
- $z^{-1}$  - je operátor zpětného posuvu, pro který platí:

$$z^{-1}y(k) = y(k-1). \quad (4.3)$$

Je zřejmé, že tento model může být použit jen pro stabilní systémy a jen pro prvních pár členů této posloupnosti. Predikce v čase  $k+j$  je pak dána vztahem

$$\hat{y}(k-j|k) = \sum_{i=0}^N h_i u(k+j-i|k) = \hat{H}(q^{-1})u(k+j|k). \quad (4.4)$$

Pro MIMO systémy s  $m$  výstupy lze tuto rovnici napsat ve tvaru

$$y_m(k) = \sum_{i=1}^p \sum_{l=0}^n h_i^{l,m} u^l(k-i). \quad (4.5)$$

kde:

- $p$  - je počet vstupů systému,
- $u^l(-)$  - je  $l$ -tý vstup systému,
- $h^{l,m}$  - je sekvence impulsní posloupnosti  $l$ -tého vstupu do  $m$ -tého výstupu systému.

Tento model se často používá díky jeho jednoduchosti, snadné identifikaci, která nevyžaduje znalost struktury systému, a možnosti jednoduchého zachycení i neminimálně fázového chování.



Nevýhody tohoto modelu jsou:

- je potřeba vysoká hodnota  $N$  (50),
- může popisovat jen stabilní procesy.

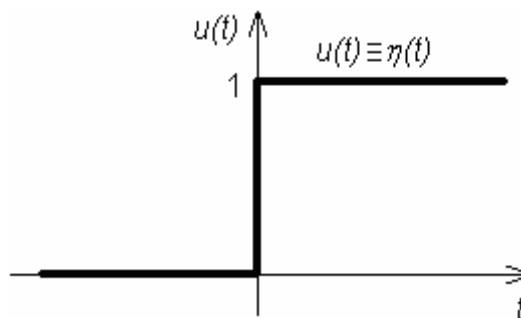
#### 4.1.2 Přejchodová funkce

Je to model procesu založený na posloupnosti přechodové funkce, což je vlastně odezva systému na vstupní jednotkový (Heavisideův) skok při nulových počátečních podmínkách systému.

Jednotkový (Heavisideův) skok se označuje jako  $\eta(t)$  a je definován:

$$\begin{aligned} \eta(t) &= 0 \quad \text{pro } t < 0 \\ \eta(t) &= 1 \quad \text{pro } t \geq 0. \end{aligned} \tag{4.6}$$

Graficky:



Obrázek 4.2 Jednotkový (Heavisideův) skok

Tento model je hodně podobný předchozímu FIR modelu a sdílí s ním stejné výhody i nevýhody. Pro stabilní systémy se pro popis výstupu používá diskretní přechodová funkce omezená na několik prvních členů

$$y(k) = \sum_{i=1}^N g_i \Delta u(k-i) = G(z^{-1})(1-z^{-1})u(k), \tag{4.7}$$

kde:

- $G(z^{-1}) = g_1 z^{-1} + g_2 z^{-2} + \dots + g_n z^{-n}$  - je polynom  $n$ -tého stupně,
- $y(k)$  - je výstup systému,
- $u(k)$  - je vstup systému,
- $g_i$  - jsou koeficienty přechodové posloupnosti,

- $\Delta u(k) = u(k) - u(k - i)$  - jsou přírůstky vstupu systému,
- $z^{-1}$  - je operátor zpětného posuvu, pro který platí:

$$z^{-1}y(k) = y(k - 1). \quad (4.8)$$

Je zřejmé že

$$\Delta y(k) = \sum_{i=0}^{\infty} \Delta g_i \Delta u(k - i) = \sum_{i=0}^{\infty} h_i \Delta u(k - i), \quad (4.9)$$

kde:

- $\Delta g_i = g_i - g_{i-1}$ ,
- $h_i = g_i - g_{i-1}$ .

Protože stabilní systém můžeme uvažovat pouze pokud  $\Delta g_i \rightarrow 0$  pro  $i \rightarrow \infty$  a proto lze výstup systému aproximovat jako

$$\hat{y}(k) = y_0 + \sum_{i=0}^N \Delta g_i \Delta u(k - i) = y_0 + \hat{G}(z^{-1})(1 - z^{-1})u(k), \quad (4.10)$$

kde:

$$\hat{G}(z^{-1}) = g_1 z^{-1} + g_2 z^{-2} + \dots + g_n z^{-n}.$$

Predikce budoucích signálů je pak dána vztahem

$$\hat{y}(k + j|k) = \sum_{i=0}^N g_i \Delta u(k + j - i|k). \quad (4.11)$$

#### 4.1.3 Přenosová funkce

Tento model se využívá například v metodě GPC (Generalized Predictive Control), která je velmi rozšířená a oblíbená. Výstup je svázán se vstupem pomocí vztahu

$$A(z^{-1})y(k) = B(z^{-1})u(k), \quad (4.12)$$

z čehož vyplývá vztah pro výstup ve tvaru

$$y(k) = \frac{B(z^{-1})}{A(z^{-1})}u(k) = G(z^{-1})u(k). \quad (4.13)$$

kde:

- $B(z^{-1}) = b_0 + b_1 z^{-1} + \dots + b_{nb} z^{-nb}$  - je polynom čitatele přenosu systému,

-  $A(z^{-1}) = a_0 + a_1 z^{-1} + \dots + a_{n_a} z^{-n_a}$  - je polynom jmenovatele přenosu systému.

Z předchozích vztahů je zřejmé, že predikce založená na tomto modelu je dána jako

$$\hat{y}(k+j|k) = \frac{B(z^{-1})}{A(z^{-1})} u(k+j|k). \quad (4.14)$$

Velkou výhodou tohoto modelu je jeho platnost i pro nestabilní systémy. V praxi se častěji používá s rozšířením o model poruchy.

Nevýhodou je naopak problém, že musíme znát stupně polynomů čitatele i jmenovatele přenosu, tedy stupně polynomů  $A(z^{-1})$  a  $B(z^{-1})$ .

#### 4.1.4 ARX model

Tento model je velmi vhodný díky tomu, že popisuje stochastické vlastnosti systému. Pro ARX model s měřitelnou poruchou platí

$$y(k) + \sum_{i=1}^{n_a} a_i y(k-i) = \sum_{i=1}^{n_b} b_i u(k-i) + \sum_{i=1}^{n_d} d_i v(k-i) + e_s(k). \quad (4.15)$$

kde:

- $y(k)$  - je výstup systému,
- $u(k)$  - je vstup systému,
- $v(k)$  - je porucha systému,
- $e_s(k)$  - je bílý šum.

Vektory minulých hodnot vstupu, výstupu a poruchy jsou

$$\begin{aligned} \tilde{y} &= [y(k-n_a) \dots y(k-1)]^T \\ \tilde{u} &= [u(k-n_a) \dots u(k-1)]^T \\ \tilde{v} &= [v(k-n_a) \dots v(k-1)]^T, \end{aligned} \quad (4.16)$$

a vektory budoucích hodnot vstupu, výstupu a poruchy jsou

$$\begin{aligned} \bar{y} &= [\hat{y}(k) \dots \hat{y}(k+N-1)]^T \\ \bar{u} &= [\hat{u}(k) \dots \hat{u}(k+N-1)]^T \\ \bar{v} &= [\hat{v}(k) \dots \hat{v}(k+N-1)]^T, \end{aligned} \quad (4.17)$$

kde:

- $\hat{y}(k)$  - je predikovaná hodnota  $y(k)$  v čase  $k$ ,

pak predikce výstupu je dána vztahem

$$\bar{y} = A_p^{-1}(-A_t \tilde{y} + B_t \tilde{u} + D_t \tilde{v} + B_p \bar{u} + D_p \bar{v}), \quad (4.18)$$

kde matice  $A_p$ ,  $A_t$ ,  $B_p$ ,  $B_t$ ,  $D_p$  a  $D_t$  jsou definovány jako

$$\begin{aligned} [A_t | A_p] &= \left[ \begin{array}{cccc|cccccc} a_n & . & . & . & a_1 & 1 & 0 & . & . & . & . & 0 \\ 0 & a_n & . & . & . & a_1 & 1 & 0 & . & . & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & 0 & a_n & . & . & a_1 & 1 & 0 & . & 0 \\ 0 & . & . & . & 0 & a_n & . & . & a_1 & 1 & . & 0 \\ 0 & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & 0 & . & . & a_n & . & . & a_1 & 1 \end{array} \right] \\ [B_t | B_p] &= \left[ \begin{array}{cccc|cccccc} b_n & . & . & . & b_1 & b_0 & 0 & . & . & . & . & 0 \\ 0 & b_n & . & . & . & b_1 & b_0 & 0 & . & . & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & 0 & b_n & . & . & b_1 & b_0 & 0 & . & 0 \\ 0 & . & . & . & 0 & b_n & . & . & b_1 & b_0 & . & 0 \\ 0 & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & 0 & . & . & b_n & . & . & b_1 & b_0 \end{array} \right] \\ [D_t | D_p] &= \left[ \begin{array}{cccc|cccccc} d_n & . & . & . & d_1 & d_0 & 0 & . & . & . & . & 0 \\ 0 & d_n & . & . & . & d_1 & d_0 & 0 & . & . & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & 0 & d_n & . & . & d_1 & d_0 & 0 & . & 0 \\ 0 & . & . & . & 0 & d_n & . & . & d_1 & d_0 & . & 0 \\ 0 & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & 0 & . & . & d_n & . & . & d_1 & d_0 \end{array} \right]. \quad (4.19) \end{aligned}$$

#### 4.1.5 Stavový popis

Pomocí diskrétního stavového popisu lze model procesu napsat jako

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k). \end{aligned} \quad (4.20)$$

Predikce výstupu pro tento model je

$$\bar{\mathbf{y}} = \bar{\mathbf{P}}\mathbf{x}(k) + \bar{\mathbf{H}}\bar{\mathbf{u}}. \quad (4.21)$$

kde:

- $x(k)$  - je počáteční stav,
- vektory  $\bar{y}$ ,  $\bar{u}$  -  $\bar{y} = [y(k)^T \dots y(k+N-1)^T]^T$ ,
- $\bar{u} = [u(k)^T \dots u(k+N-1)^T]^T$ ,
- matice  $\bar{P}$ ,  $\bar{H}$  -  $\bar{P} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix}$ ,
- $\bar{H} = \begin{bmatrix} D & & & \\ CB & D & & \\ \vdots & & \ddots & \\ CA^{N-2}B & \dots & CB & D \end{bmatrix}$ .

Pro stav je rovnice predikce obdobná

$$\bar{x} = Px(k) + H\bar{u}. \quad (4.22)$$

kde:

- $x(k)$  - je počáteční stav,
- vektory  $\bar{x}$ ,  $\bar{u}$  -  $\bar{x} = [x(k+1)^T \dots x(k+N)^T]^T$ ,
- $\bar{u} = [u(k)^T \dots u(k+N-1)^T]^T$ ,
- matice  $P$ ,  $H$  -  $P = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}$ ,
- $H = \begin{bmatrix} B & & & \\ AB & B & & \\ \vdots & & \ddots & \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}$ .

Výhodou stavového popisu je stejný zápis jednorozměrných a mnohorozměrných systémů. Na druhou stranu je však potřeba pozorovatel stavu. [12]

#### 4.1.6 Jiné modely

Již dříve bylo řečeno, že je možné v prediktivním řízení použít libovolné modely. Dosti často se také používají spojité modely, které bývají reprezentovány popisem pomocí soustavy diferenciálních rovnic. Jejich velkou nevýhodou jsou ovšem vysoké nároky na simulaci a optimalizaci. V poslední době se začínají používat i jiné modely jako například neuronové sítě či popisy pomocí fuzzy logiky. V těchto případech bývá model používán buďto přímo a nebo jsou s ním generovány jen některé charakteristiky procesu, jako například skoková či impulsní odezva systému.

### 4.2 Účelová funkce

Standardní účelová funkce obsahuje kvadratické členy přírůstků řízení na konečném horizontu do budoucnosti a kvadratické členy regulační odchylky

$$J = E \sum_{i=N_1}^{N_2} [\delta(i)\hat{y}(k+i) - w(k+i)]^2 + \sum_{i=1}^{N_U} [\lambda(i)\Delta u(k+i-1)]^2, \quad (4.23)$$

kde:

- $\hat{y}(k+i)$  - je predikovaný výstup  $i$  kroků do budoucnosti v závislosti na informacích dostupných do času  $k$ ,
- $w(k+i)$  - je posloupnost žádané veličiny,
- $\Delta u(k+i-1)$  - je posloupnost budoucích přírůstků akčních zásahů, které mají být vypočítány.

Implicitně se předpokládají omezení na přírůstky akčních zásahů ve tvaru

$$\Delta u(k+i-1) = 0, \quad N_U < i < N_2. \quad (4.24)$$

Účelová funkce má obecně následující parametry:

- Parametry  $N_1$ ,  $N_2$  a  $N_U$  se nazývají minimální, maximální a řídicí horizont.  $N_1$  a  $N_2$  určují interval v budoucnosti, ve kterém je potřeba sledovat trajektorii žádané veličiny.  $N_1$  volíme minimálně  $T_d + 1$ , kde  $T_d$  je přibližná hodnota dopravního zpoždění. Nastavením dostatečně velkého  $N_1$  se lze vyhnout problémům s neminimálně fázovými soustavami. Hodnota  $N_2$  by měla pokrývat důležitou oblast přechodové charakteristiky, proto se většinou volí podobná času  $T_{90}$ , který vytyčuje úsek, kdy přechodová

charakteristika přejde z 10% na 90% své ustálené hodnoty. Využitím řídicího horizontu  $N_U$  se snižuje výpočtová náročnost metody. Parametry  $\delta(i)$  a  $\lambda(i)$  jsou sekvence, kterými lze ovlivňovat budoucí chování řízené soustavy. Většinou se volí jako konstanty a nebo ve formě exponenciálních vah.

- Předpokládá se, že trajektorie žádané hodnoty regulované veličiny  $w(k+i)$  je známá. Pokud tomu tak není, tak se obvykle uvažuje konstantní a rovna aktuální hodnotě žádané veličiny. Další možností je použití trajektorie, která plynule přechází z aktuální hodnoty výstupu na konečnou hodnotu žádané veličiny, například jako filtr prvního řádu.

$$\begin{aligned} w(k) &= y(k), \\ w(k+i) &= \alpha w(k+i-1) + (1-\alpha)w^\infty, \end{aligned} \quad (4.25)$$

kde:

- $\alpha$  - určuje plynulost trajektorie, jestliže  $\alpha \rightarrow 0$ , je trajektorie nejrychlejší, pokud  $\alpha \rightarrow 1$ , je trajektorie nejpomalejší.

Stejný efekt může být dosažen použitím polynomu  $P(z^{-1})$ . Výstup sleduje trajektorii  $\frac{1}{P}w$ . Filtr odpovídající předešlé trajektorii prvního řádu je dán vztahem

$$P(z^{-1}) = \frac{1 - \alpha z^{-1}}{1 - \alpha}. \quad (4.26)$$

Cílem prediktivního řízení spočítat posloupnost budoucích hodnot změny akčního zásahu  $[\Delta u(k), \Delta u(k+1), \dots]$  tak, aby bylo minimalizováno kritérium (4.23). Pro další práce je nutné kritérium (4.23) převést do maticové podoby.

Výstup modelu lze spočítat jako součet volné odezvy  $y_0$  a nucené odezvy  $y_n$  modelu. Volná odezva je závislá pouze na minulých hodnotách akčního zásahu a výstupu soustavy, protože se předpokládá, že budoucí hodnota akčního zásahu je konstantní, tedy změny akčního zásahu jsou nulové. Na rozdíl od volné odezvy bere nucená odezva v potaz právě nenulové budoucí změny akčního zásahu. Pro lineární i obecně nelineární systém není problém volnou odezvu vypočítat. Nucenou odezvu lze spočítat jako součin matice  $\mathbf{G}$  (Jacobián model) a vektoru budoucích změn akčního zásahu  $\tilde{\mathbf{u}}$ , který dopředu známe

$$\mathbf{y} = \mathbf{y}_n + \mathbf{y}_0 = \mathbf{G}\tilde{\mathbf{u}} + \mathbf{y}_0. \quad (4.27)$$

Účelová funkce (4.23) se převede do maticové podoby

$$J = (\mathbf{y} - \mathbf{w})^T (\mathbf{y} - \mathbf{w}) + \lambda \tilde{\mathbf{u}}^T \tilde{\mathbf{u}}. \quad (4.28)$$

Do rovnice (4.28) se dosadí rovnice (4.27), čímž se získá

$$J = (\mathbf{G}\tilde{\mathbf{u}} + \mathbf{y}_0 - \mathbf{w})^T (\mathbf{G}\tilde{\mathbf{u}} + \mathbf{y}_0 - \mathbf{w}) + \lambda \tilde{\mathbf{u}}^T \tilde{\mathbf{u}}, \quad (4.29)$$

kde:

- $\mathbf{y}$  - je vektor předikovaných výstupů systému -  $\mathbf{y}^T = [\hat{y}(k+1) \hat{y}(k+2) \dots \hat{y}(k+N_2)]$ ,
- $\tilde{\mathbf{u}}$  - je vektor změn akčních zásahů -  $\tilde{\mathbf{u}}^T = [\Delta u(k) \Delta u(k+1) \dots \Delta u(k+N_2-1)]$ ,
- $\mathbf{w}$  - vektor budoucích hodnot žádané veličiny-  $\mathbf{w}^T = [w(k+1) w(k+2) \dots w(k+N_2)]$ ,
- $\mathbf{y}_0$  - je volná odezva systému -  $\mathbf{y}_0^T = [y_0(k+1) y_0(k+2) \dots y_0(k+N_2)]$ ,
- $\mathbf{G}$  - je Jacobián model (parciální derivace modelu podle jeho proměnných), pro kauzální systémy se jedná o trojúhelníkovou matici, pro lineární systémy je tvar matice  $\mathbf{G}$  znám

$$\mathbf{G} = \begin{bmatrix} g_1 & 0 & 0 & \dots & 0 \\ g_2 & g_1 & 0 & \dots & 0 \\ g_3 & g_2 & g_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{N_2} & g_{N_2-1} & g_{N_2-2} & \dots & g_{N_2-N_U+1} \end{bmatrix}. \quad (4.30)$$

Koeficienty  $g_j$  jsou prvky odezvy modelu na jednotkový skok a jejich hodnotu můžeme zjistit ze vztahu

$$g_j = -\sum_{i=1}^j a_i g_{j-i} + \sum_{i=0}^j b_i \quad j = 0, 1, \dots, N_2 - 1, \quad (4.31)$$

kde:

- $g_k = 0$  pro každé  $k < 0$ ,
- hodnoty  $a_i$  a  $b_i$  jsou koeficienty lineárního modelu soustavy v diskretním tvaru

$$G(z) = \frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + \dots}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + \dots}. \quad (4.32)$$



Účelovou funkci (4.29) dále převedeme do maticového tvaru používaného pro metody kvadratického programování

$$J = \frac{1}{2} \tilde{\mathbf{u}}^T \mathbf{H} \tilde{\mathbf{u}} + \mathbf{b}^T \tilde{\mathbf{u}} + \mathbf{f}_0, \quad (4.33)$$

kde:

- $\mathbf{H} = 2(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})$ ,
- $\mathbf{b}^T = 2(\mathbf{y}_0 - \mathbf{w})^T \mathbf{G}$ ,
- $\mathbf{f}_0 = (\mathbf{y}_0 - \mathbf{w})^T (\mathbf{y}_0 - \mathbf{w})$ .

Člen  $\mathbf{f}_0$  v rovnici (4.33) není závislý na  $\tilde{\mathbf{u}}$ , proto na řešení optimalizační úlohy nemá žádný vliv a tudíž jej lze z rovnice vypustit, čímž se získá výsledný tvar kritéria pro optimalizaci

$$J = \frac{1}{2} \tilde{\mathbf{u}}^T \mathbf{H} \tilde{\mathbf{u}} + \mathbf{b}^T \tilde{\mathbf{u}}. \quad (4.34)$$

### 4.3 Omezující podmínky

V praxi se lze velmi často setkat s omezeními (constraints). Často to jsou omezení fyzická ať už u senzorů, či akčních členů, nebo technologická či jiná omezení daného procesu. Nejčastěji se omezují vstupní veličiny, které mohou pracovat pouze v určitém rozsahu hodnot (například průtoky tekutiny nemohou být záporné, nebo ventil nemůže být otevřen více jak na 100%). Existují však i určitá doporučení a limity pro výstupní veličiny procesů (například s ohledem na životní prostředí či bezpečnost na pracovištích apod.). A právě velkou předností prediktivního řízení je schopnost pracovat s omezeními vstupních i výstupních veličin, což mělo vliv na rozšíření MPC v průmyslu, protože velké množství průmyslových procesů je řízeno na hodnotách blízkých se omezujícím podmínkám, nebo přímo na nich. Z tohoto faktu vyplývá požadavek na optimální řízení v rámci omezení [1], [8].

Nejjednodušším a v praxi zároveň nepoužívanějším způsobem aplikace omezení je analytické řešení optimalizační úlohy bez omezení a následná aplikace omezení, čemuž se říká “saturace na výsledek“. Toto řešení je jednoduché, avšak nezaručí optimální řízení

podle zvolených kritérií. Další nevýhodou je, že takováto omezení lze aplikovat pouze na veličiny, které vystupují z algoritmu optimalizace (akční zásah a změna akčního zásahu).

Vhodnějším řešením je tedy řešit danou optimalizační úlohu již s danými omezeními. Tento přístup umožní nejen omezení výstupních veličin daného optimalizačního algoritmu, ale také výstupní veličiny soustavy a při použití stavového modelu dokonce i omezení jednotlivých vnitřních stavů systému.

Omezení rozlišujeme jako:

- tvrdá omezení (hard constraints) jsou fyzická omezení procesů, jejichž hranice v prostoru řešení nelze za žádných okolností překročit,
- měkká omezení (soft constraints) jsou ta, která mohou být za určitých podmínek porušena.

#### 4.3.1 Tvrdá omezení

Nejčastěji používaná omezení jsou:

- omezení změny akčního zásahu      -  $\Delta u_{\min} \leq u(k) - u(k-1) \leq \Delta u_{\max}$ ,
- omezení akčního zásahu              -  $u_{\min} \leq u(k) \leq u_{\max}$ ,
- omezení výstupní veličiny           -  $y_{\min} \leq y(k) \leq y_{\max}$ ,
- omezení překmitu                      -  $y(k+j) \leq w(k)$ ,
- zajištění monotónního průběhu výstupní veličiny
  - neklesající                      -  $y(k+j) \leq y(k+j+1) \quad \forall y(k) < w(k)$ ,
  - nerostoucí                        -  $y(k+j) \geq y(k+j+1) \quad \forall y(k) > w(k)$ .

Všechna omezení lze vyjádřit jedinou maticovou nerovnicí

$$A\mathbf{u} \leq \mathbf{b}. \quad (4.35)$$

Omezení změny akčního zásahu lze vyjádřit jako

$$\begin{aligned}
 & \Delta u(k) \leq \Delta u_{\max} & \Delta u(k) \geq \Delta u_{\min} \\
 & & -\Delta u(k) \leq -\Delta u_{\min} \\
 & \begin{bmatrix} 1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \\ \vdots & \vdots & & \ddots \end{bmatrix} \mathbf{u} \leq \begin{bmatrix} \Delta u_{\max} \\ \Delta u_{\max} \\ \Delta u_{\max} \\ \vdots \end{bmatrix} & \begin{bmatrix} -1 & 0 & 0 & \cdots \\ 0 & -1 & 0 & \cdots \\ 0 & 0 & -1 & \\ \vdots & \vdots & & \ddots \end{bmatrix} \mathbf{u} \leq \begin{bmatrix} -\Delta u_{\min} \\ -\Delta u_{\min} \\ -\Delta u_{\min} \\ \vdots \end{bmatrix} \\
 & \mathbf{I} \mathbf{u} \leq \Delta \mathbf{u}_{\max} & -\mathbf{I} \mathbf{u} \leq -\Delta \mathbf{u}_{\min}
 \end{aligned} \quad (4.36)$$

Omezení akčního zásahu lze vyjádřit jako

$$\begin{aligned}
 & u(k) \leq u_{\max} & u(k) \geq u_{\min} \\
 & u(k-1) + \Delta u(k) \leq u_{\max} & -u(k-1) - \Delta u(k) \leq -u_{\min} \\
 & \Delta u(k) \leq u_{\max} - u(k-1) & -\Delta u(k) \leq -u_{\min} + u(k-1) \\
 & \begin{bmatrix} 1 & 0 & 0 & \cdots \\ 1 & 1 & 0 & \cdots \\ 1 & 1 & 1 & \\ \vdots & \vdots & & \ddots \end{bmatrix} \mathbf{u} \leq \begin{bmatrix} u_{\max} - u(k-1) \\ u_{\max} - u(k-1) \\ u_{\max} - u(k-1) \\ \vdots \end{bmatrix} & \begin{bmatrix} -1 & 0 & 0 & \cdots \\ -1 & -1 & 0 & \cdots \\ -1 & -1 & -1 & \\ \vdots & \vdots & & \ddots \end{bmatrix} \mathbf{u} \leq \begin{bmatrix} -u_{\min} + u(k-1) \\ -u_{\min} + u(k-1) \\ -u_{\min} + u(k-1) \\ \vdots \end{bmatrix} \\
 & \mathbf{T} \mathbf{u} \leq \mathbf{u}_{\max} - \mathbf{u}_{k-1} & -\mathbf{T} \mathbf{u} \leq -\mathbf{u}_{\min} + \mathbf{u}_{k-1}
 \end{aligned}$$

Omezení výstupu soustavy lze vyjádřit jako

$$\begin{aligned}
 & y(k) \leq y_{\max} \\
 & \mathbf{G} \mathbf{u} + \mathbf{y}_0 \leq \mathbf{y}_{\max} \\
 & \mathbf{G} \mathbf{u} \leq \mathbf{y}_{\max} - \mathbf{y}_0 \\
 & \begin{bmatrix} g_1 & 0 & 0 & \cdots & 0 \\ g_2 & g_1 & 0 & \cdots & 0 \\ g_3 & g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{N_2} & g_{N_2-1} & g_{N_2-2} & \cdots & g_{N_2-N_U+1} \end{bmatrix} \mathbf{u} \leq \begin{bmatrix} y_{\max} - y_0(k+1) \\ y_{\max} - y_0(k+2) \\ y_{\max} - y_0(k+3) \\ \vdots \\ y_{\max} - y_0(k+N_2) \end{bmatrix} \\
 & & y(k) \geq y_{\min} \\
 & -\mathbf{G} \mathbf{u} - \mathbf{y}_0 \leq -\mathbf{y}_{\min} \\
 & -\mathbf{G} \mathbf{u} \leq -\mathbf{y}_{\min} + \mathbf{y}_0 \\
 & \begin{bmatrix} -g_1 & 0 & 0 & \cdots & 0 \\ -g_2 & -g_1 & 0 & \cdots & 0 \\ -g_3 & -g_2 & -g_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -g_{N_2} & -g_{N_2-1} & -g_{N_2-2} & \cdots & -g_{N_2-N_U+1} \end{bmatrix} \mathbf{u} \leq \begin{bmatrix} -y_{\min} + y_0(k+1) \\ -y_{\min} + y_0(k+2) \\ -y_{\min} + y_0(k+3) \\ \vdots \\ -y_{\min} + y_0(k+N_2) \end{bmatrix} \\
 & & (4.38)
 \end{aligned}$$

Omezení změny akčního zásahu (4.36), akčního zásahu (4.37) a výstupu systému (4.38) se dosadí do nerovnice (4.35)

$$\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \\ \mathbf{T} \\ -\mathbf{T} \\ \mathbf{G} \\ -\mathbf{G} \end{bmatrix} \mathbf{u} \leq \begin{bmatrix} \Delta \mathbf{u}_{max} \\ -\Delta \mathbf{u}_{min} \\ \mathbf{u}_{max} - \mathbf{u}_{k-1} \\ -\mathbf{u}_{min} + \mathbf{u}_{k-1} \\ \mathbf{y}_{max} - \mathbf{y}_0 \\ -\mathbf{y}_{min} + \mathbf{y}_0 \end{bmatrix}. \quad (4.39)$$

Maticе  $\mathbf{I}$  (jednotková),  $\mathbf{T}$  (dolní trojúhelníková) a  $\mathbf{G}$  jsou čtvercového rozměru  $N_U$ . Na pravé straně nerovnice jsou sloupcové vektory délky  $N_U$ .

### 4.3.2 Měkká omezení

V některých případech lze použít i tzv. měkká omezení, kdy je omezované veličině dovoleno překročit stanovené meze o určitou toleranci  $\varepsilon$ . Děje se tak v případech, kdy není možné nalézt optimální řešení v rámci tvrdých omezení, nebo v případech, kdy je možno tvrdá omezení z hlediska technologického a bezpečnostního překročit. Posunutí tvrdých omezení výstupu soustavy uvolněním horní meze o  $\varepsilon_{max}$  a spodní meze o  $\varepsilon_{min}$  je vyjádřeno následujícími nerovnostmi

$$\begin{aligned} y(k) &\leq y_{max} + \varepsilon_{max} \\ \mathbf{G}\mathbf{u} + \mathbf{y}_0 &\leq \mathbf{y}_{max} + \boldsymbol{\varepsilon}_{max} \\ \mathbf{G}\mathbf{u} &\leq \mathbf{y}_{max} + \boldsymbol{\varepsilon}_{max} - \mathbf{y}_0 \end{aligned}$$

$$\begin{bmatrix} \mathbf{g}_1 & 0 & 0 & \cdots & 0 \\ \mathbf{g}_2 & \mathbf{g}_1 & 0 & \cdots & 0 \\ \mathbf{g}_3 & \mathbf{g}_2 & \mathbf{g}_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{g}_{N_2} & \mathbf{g}_{N_2-1} & \mathbf{g}_{N_2-2} & \cdots & \mathbf{g}_{N_2-N_U+1} \end{bmatrix} \mathbf{u} \leq \begin{bmatrix} y_{max} + \varepsilon_{max} - y_0(k+1) \\ y_{max} + \varepsilon_{max} - y_0(k+2) \\ y_{max} + \varepsilon_{max} - y_0(k+3) \\ \vdots \\ y_{max} + \varepsilon_{max} - y_0(k+N_2) \end{bmatrix}$$

$$\begin{aligned} y(k) &\geq y_{min} + \varepsilon_{min} \\ -\mathbf{G}\mathbf{u} - \mathbf{y}_0 &\leq -\mathbf{y}_{min} - \boldsymbol{\varepsilon}_{min} \\ -\mathbf{G}\mathbf{u} &\leq -\mathbf{y}_{min} - \boldsymbol{\varepsilon}_{min} + \mathbf{y}_0 \end{aligned}$$

$$\begin{bmatrix} -\mathbf{g}_1 & 0 & 0 & \cdots & 0 \\ -\mathbf{g}_2 & -\mathbf{g}_1 & 0 & \cdots & 0 \\ -\mathbf{g}_3 & -\mathbf{g}_2 & -\mathbf{g}_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\mathbf{g}_{N_2} & -\mathbf{g}_{N_2-1} & -\mathbf{g}_{N_2-2} & \cdots & -\mathbf{g}_{N_2-N_U+1} \end{bmatrix} \mathbf{u} \leq \begin{bmatrix} -y_{min} - \varepsilon_{min} + y_0(k+1) \\ -y_{min} - \varepsilon_{min} + y_0(k+2) \\ -y_{min} - \varepsilon_{min} + y_0(k+3) \\ \vdots \\ -y_{min} - \varepsilon_{min} + y_0(k+N_2) \end{bmatrix}. \quad (4.40)$$

Podobně lze odvodit i ostatní omezení a zahrnout je do nerovnice (4.35).

## 5 ODVOZENÍ A REALIZACE PREDIKTIVNÍHO ŘÍZENÍ

V této kapitole bude odvozeno prediktivní řízení ve smyslu metody GPC, která je založena na vstupně-výstupním modelu procesu.

### 5.1 Odvození prediktoru

Většina stochastických SISO (Single Input Single Output) systémů může být popsáno ARMAX (CARMA – Controlled Autoregressive Moving-Average) modelem ve tvaru

$$A(z^{-1})y(k) = z^{-d}B(z^{-1})u(k) + C(z^{-1})e_s(k), \quad (5.1)$$

kde:

- $u(k)$  - je řídicí posloupnost systému řízení,
- $y(k)$  - je výstupní posloupnost systém řízení,
- $e_s(k)$  - je bílý šum,
- $A, B, C$  - jsou polynomy ve tvaru

$$- A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{na}z^{-na},$$

$$- B(z^{-1}) = b_1z^{-1} + b_2z^{-2} + \dots + b_{nb}z^{-nb},$$

$$- C(z^{-1}) = 1 + c_1z^{-1} + c_2z^{-2} + \dots + c_{nc}z^{-nc}.$$

Pro průmyslové aplikace, ve kterých jsou nestacionární poruchy, což je většina z nich, je vhodnější použít integrovaný model CARIMA (Controlled Autoregressive Integrated Moving-Average). Prvním krokem při odvození prediktivního řízení je výpočet optimálního predikátoru výstupní veličiny. Pro odvození tohoto predikátoru bude použit jednorozměrný CARIMA model ve tvaru

$$A(z^{-1})y(k) = B(z^{-1})u(k-1) + \frac{C(z^{-1})}{\Delta}e_s(k). \quad (5.2)$$

V rovnici (5.2) vystupuje akční veličina ve tvaru  $u(k-1)$ , protože  $u(k)$  je neznámá veličina, která má být vypočítána. Jelikož je tato rovnice uvažována  $j$  kroků do budoucnosti, je proto vynásobena členem  $z^j$  a zároveň je podělena polynomem  $A$ . Těmito operacemi se získá

$$\hat{y}(k+j) = \frac{B(z^{-1})}{A(z^{-1})}u(k+j-1) + \frac{C(z^{-1})}{\Delta A(z^{-1})}e_s(k+j). \quad (5.3)$$

Poslední člen rovnice (5.3) obsahuje zároveň minulé i budoucí hodnoty neměřitelné poruchy  $e_s$ . Je možností je rozdělit polynomiálním dělením člena  $\frac{C(z^{-1})}{\Delta A(z^{-1})}$  a oddělením prvních  $j$  členů s kladnými mocninami  $z$ . Vznikne tím rovnice

$$\frac{C(z^{-1})}{\Delta A(z^{-1})} = E_j(z^{-1}) + z^{-j} \frac{F_j(z^{-1})}{\Delta A(z^{-1})}, \quad (5.4)$$

kde:

- polynom  $E_j$  - je řádu  $j-1$ .

Dosazením pravé strany (5.4) zpět do (5.3) vznikne

$$\hat{y}(k+j) = \frac{B(z^{-1})}{A(z^{-1})}u(k+j-1) + E_j(z^{-1})e_s(k+j) + \frac{F_j(z^{-1})}{\Delta A(z^{-1})}e_s(k). \quad (5.5)$$

Poslední člen rovnice (5.5) obsahuje pouze současnou hodnotu poruchy  $e_s$ , kterou lze vypočítat ze vztahu

$$e_s(k) = \frac{\Delta A(z^{-1})}{C(z^{-1})}y(k) - \frac{B(z^{-1})}{C(z^{-1})}\Delta u(k-1), \quad (5.6)$$

a dosazením zpět do rovnice (5.5) se získá

$$\begin{aligned} \hat{y}(k+j) &= \frac{B(z^{-1})}{A(z^{-1})}u(k+j-1) + E_j(z^{-1})e_s(k+j) + \frac{F_j(z^{-1})}{C(z^{-1})}y(k) - \frac{F_j(z^{-1})B(z^{-1})}{\Delta A(z^{-1})C(z^{-1})}\Delta u(k-1) = \\ &= \left[ \frac{B(z^{-1})}{\Delta A(z^{-1})} - z^{-j} \frac{F_j(z^{-1})B(z^{-1})}{\Delta A(z^{-1})C(z^{-1})} \right] \Delta u(k+j-1) + \frac{F_j(z^{-1})}{C(z^{-1})}y(k) + E_j(z^{-1})e_s(k+j) = \\ &= \frac{B(z^{-1})}{C(z^{-1})} \left[ \frac{C(z^{-1})}{\Delta A(z^{-1})} - z^{-j} \frac{F_j(z^{-1})}{\Delta A(z^{-1})} \right] \Delta u(k+j-1) + \frac{F_j(z^{-1})}{C(z^{-1})}y(k) + E_j(z^{-1})e_s(k+j) \quad (5.7) \end{aligned}$$

Jestliže se dosadí pravá strana rovnice (5.4) za podíl  $\frac{C(z^{-1})}{\Delta A(z^{-1})}$  do členu obsahujícího

$\Delta u(k+j-1)$  v rovnici (5.7), vznikne

$$\hat{y}(k+j) = \frac{B(z^{-1})}{C(z^{-1})} \left[ E_j(z^{-1}) + z^{-j} \frac{F_j(z^{-1})}{\Delta A(z^{-1})} - z^{-j} \frac{F_j(z^{-1})}{\Delta A(z^{-1})} \right] \Delta u(k+j-1) + \frac{F_j(z^{-1})}{C(z^{-1})} y(k) + E_j(z^{-1}) e_s(k+j) \quad (5.8)$$

Opět se oddělí současné a budoucí přírůstky řízení od minulých pomocí polynomiálního dělení

$$\frac{B(z^{-1})E_j(z^{-1})}{C(z^{-1})} = G_j(z^{-1}) + z^{-j} \frac{\Gamma_j(z^{-1})}{C(z^{-1})}, \quad (5.9)$$

a dostaneme výsledný tvar pro budoucí hodnoty výstupu procesu

$$\hat{y}(k+j) = G_j(z^{-1}) \Delta u(k+j-1) + \frac{\Gamma_j(z^{-1})}{C(z^{-1})} \Delta u(k-1) + \frac{F_j(z^{-1})}{C(z^{-1})} y(k) + E_j(z^{-1}) e_s(k+j) \quad (5.10)$$

Pro dosažení optimálního predikátoru výstupu  $y(k+j)$  na základě údajů známých do času  $k$  se zanedbá poslední člen obsahující budoucí hodnoty poruch

$$\hat{y}(k+j) = G_j(z^{-1}) \Delta u(k+j-1) + \frac{\Gamma_j(z^{-1})}{C(z^{-1})} \Delta u(k-1) + \frac{F_j(z^{-1})}{C(z^{-1})} y(k). \quad (5.11)$$

Pokud se pro součet posledních dvou členů rovnice (5.11) zavede vztah

$$y_0(k+j) = \frac{\Gamma_j(z^{-1})}{C(z^{-1})} \Delta u(k-1) + \frac{F_j(z^{-1})}{C(z^{-1})} y(k), \quad (5.12)$$

pak lze rovnici (5.11) zapsat ve tvaru

$$\hat{y}(k+j) = G_j(z^{-1}) \Delta u(k+j-1) + y_0(k+j). \quad (5.13)$$

Z rovnic (5.4) a (5.9) lze vidět, že pro odvození  $j$ -krokového prediktoru je potřeba vyřešit dvě rovnice polynomiálního dělení (neboli Diofantické rovnice) ve tvaru

$$C(z^{-1}) = E_j(z^{-1}) \Delta A(z^{-1}) + z^{-j} F_j(z^{-1}), \quad (5.14)$$

$$B(z^{-1}) E_j(z^{-1}) = G_j(z^{-1}) C(z^{-1}) + z^{-j} F_j(z^{-1}). \quad (5.15)$$

Pro lepší celkové pochopení prediktoru je vhodné analyzovat rovnici (5.13) a členy, které obsahuje. Nejdříve je nutné uvažovat, že všechny budoucí přírůstky akčních zásahů jsou nulové, díky čemuž zůstane řízení zůstane konstantní na současné hodnotě. Z rovnice (5.13) vyplývá



$$\hat{y}(k+j) = y_0(k+j). \quad (5.16)$$

Lze vidět, že člen  $y_0$  je volnou odezvou soustavy, jestliže vstup zůstává konstantní na poslední aplikované hodnotě  $u(k-1)$ . Podobně lze uvažovat, že proces je v čase  $k$  v ustáleném stavu a lze taky předpokládat, že ustálený stav je nulový. Z tohoto předpokladu vyplývá, že volná odezva  $y_0(k+j)$  je taktéž nulová. Jestliže na proces působí v čase  $k$  jednotková skoková změna akčního zásahu, tak výstup procesu je daný na základě (5.13) jako

$$\begin{aligned} \hat{y}(k+j) &= G_j(z^{-1})\Delta u(k+j-1) = \\ &= g_{j,0}\Delta u(k+j-1) + g_{j,1}\Delta u(k+j-2) + \dots + g_{j,j-1}\Delta u(k) = \\ &= g_{j,j-1}. \end{aligned} \quad (5.17)$$

Je zřejmé, že polynom  $G_j(z^{-1})$  obsahuje koeficienty diskrétní přechodové funkce. Lze to také ukázat, pokud je rovnice (5.4) uvažována vynásobena  $\frac{B(z^{-1})}{C(z^{-1})}$  a použita společně se vztahem (5.9), pak lze získat

$$\begin{aligned} \frac{B(z^{-1})}{\Delta A(z^{-1})} &= \frac{B(z^{-1})E_j(z^{-1})}{C(z^{-1})} + z^{-j} \frac{B(z^{-1})F_j(z^{-1})}{\Delta A(z^{-1})C(z^{-1})} = \\ &= G_j(z^{-1}) + z^{-j} \frac{\Gamma_j(z^{-1})}{C(z^{-1})} + z^{-j} \frac{B(z^{-1})F_j(z^{-1})}{\Delta A(z^{-1})C(z^{-1})}, \end{aligned} \quad (5.18)$$

z čehož lze vidět, že  $G_j$  je podílem polynomů  $\frac{B(z^{-1})}{\Delta A(z^{-1})}$ .

## 5.2 Výpočet optimálního řízení

Účelová funkce v GPC je dána vztahem

$$J = E \sum_{i=N_1}^{N_2} [\delta(i)\hat{y}(k+i) - w(k+i)]^2 + \sum_{i=1}^{N_U} [\lambda(i)\Delta u(k+i-1)]^2, \quad (5.19)$$

který byl již zmíněn v kapitole 4.2. Nejdříve pro jednoduchost předpokládejme, že  $N_1 = 1$ ,  $N_U = N_2$ ,  $\delta(i) = 1$ . Z těchto předpokladů vyplývá, že jsou potřebné všechny predikce až do času  $k+N_2$ . Dalším krokem je vytvoření vektorů predikce výstupu, volné odezvy, budoucích přírůstků akční veličiny a budoucí trajektorie žádané veličiny

$$\hat{\mathbf{y}}^T = [\hat{y}(k+1), \hat{y}(k+2), \dots, \hat{y}(k+N_2)], \quad (5.20)$$

$$\mathbf{y}_0^T = [y_0(k+1), y_0(k+2), \dots, y_0(k+N_2)], \quad (5.21)$$

$$\tilde{\mathbf{u}}^T = [\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+N_2-1)], \quad (5.22)$$

$$\mathbf{w}^T = [w(k+1), w(k+2), \dots, w(k+N_2)]. \quad (5.23)$$

Na vektorový zápis rovnice prediktoru (5.13) vytvoříme matici obsahující koeficienty přechodové charakteristiky

$$\mathbf{G} = \begin{pmatrix} \mathbf{g}_1 & 0 & \cdots & \cdots & 0 \\ \mathbf{g}_2 & \mathbf{g}_1 & 0 & \cdots & 0 \\ \vdots & \mathbf{g}_2 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \mathbf{g}_1 & 0 \\ \mathbf{g}_{N_2} & \cdots & \cdots & \cdots & \mathbf{g}_{N_2-N_U+1} \end{pmatrix}. \quad (5.24)$$

Při úvaze horizontu  $N_1$  lze vidět, že prvních  $N_1 - 1$  řádků matice  $\mathbf{G}$  musí být odstraněno. Podobně u horizontu  $N_U$  dochází k redukci počtu sloupců na  $N_U$ . Z čehož vyplývá, že matice  $\mathbf{G}$  má rozměry  $[N_2 - N_1 + 1 \times N_U]$ .

Prediktor ve vektorovém tvaru je dán vztahem

$$\hat{\mathbf{y}} = \mathbf{G}\tilde{\mathbf{u}} + \mathbf{y}_0, \quad (5.25)$$

a účelová funkce (5.19) je dána vztahem

$$\begin{aligned} J &= (\hat{\mathbf{y}} - \mathbf{w})^T (\hat{\mathbf{y}} - \mathbf{w}) + \lambda \tilde{\mathbf{u}}^T \tilde{\mathbf{u}} = \\ &= (\mathbf{G}\tilde{\mathbf{u}} + \mathbf{y}_0 - \mathbf{w})^T (\mathbf{G}\tilde{\mathbf{u}} + \mathbf{y}_0 - \mathbf{w}) + \lambda \tilde{\mathbf{u}}^T \tilde{\mathbf{u}}. \end{aligned} \quad (5.26)$$

Po roznásobení a úpravě se získá

$$\begin{aligned} J &= (\hat{\mathbf{y}} - \mathbf{w})^T (\hat{\mathbf{y}} - \mathbf{w}) + \lambda \tilde{\mathbf{u}}^T \tilde{\mathbf{u}} = \\ &= \tilde{\mathbf{u}}^T \mathbf{G}^T \mathbf{G} \tilde{\mathbf{u}} + \lambda \tilde{\mathbf{u}}^T \tilde{\mathbf{u}} + \tilde{\mathbf{u}}^T \mathbf{G}^T (\mathbf{y}_0 - \mathbf{w}) + (\mathbf{y}_0 - \mathbf{w})^T \mathbf{G} \tilde{\mathbf{u}} + (\mathbf{y}_0 - \mathbf{w})(\mathbf{y}_0 - \mathbf{w})^T \\ &= \tilde{\mathbf{u}}^T (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}) \tilde{\mathbf{u}} + \tilde{\mathbf{u}}^T \mathbf{G}^T (\mathbf{y}_0 - \mathbf{w}) + (\mathbf{y}_0 - \mathbf{w})^T \mathbf{G} \tilde{\mathbf{u}} + (\mathbf{y}_0 - \mathbf{w})(\mathbf{y}_0 - \mathbf{w})^T. \end{aligned} \quad (5.27)$$

Gradient  $\mathbf{g}$  a Hessova matice  $\mathbf{H}$  jsou definovány vztahy

$$\mathbf{g}^T = \mathbf{G}^T (\mathbf{y}_0 - \mathbf{w}), \quad (5.28)$$

$$\mathbf{H} = \mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}. \quad (5.29)$$

Tímto způsobem byla originální úloha optimálního řízení transformována na problém lineární algebry. Minimum účelové funkce (5.26) se získá tak, že se první

derivace  $\frac{\partial J}{\partial \tilde{\mathbf{u}}}$  položí rovna nule a tím se získá

$$0 = 2(\mathbf{G}\mathbf{G}^T + \lambda \mathbf{I})\tilde{\mathbf{u}} + 2\mathbf{G}^T (\mathbf{y}_0 - \mathbf{w}),$$

$$\begin{aligned}(\mathbf{G}\mathbf{G}^T + \lambda\mathbf{I})\tilde{\mathbf{u}} &= -\mathbf{G}^T(\mathbf{y}_0 - \mathbf{w}), \\ \tilde{\mathbf{u}} &= -(\mathbf{G}\mathbf{G}^T + \lambda\mathbf{I})^{-1}\mathbf{G}^T(\mathbf{y}_0 - \mathbf{w}).\end{aligned}\quad (5.30)$$

Tímto způsobem byla vypočtena celá trajektorie budoucích přírůstků akční veličiny.

Tímto způsobem byla vypočítána celá trajektorie budoucích přírůstků řízení. Pochopitelně se jedná o řízení v otevřeném regulačním obvodu. Na uzavření regulačního obvodu se aplikuje do procesu první element z  $\tilde{\mathbf{u}}$ , tj.  $\Delta u(k)$  a celý postup se opakuje v další periodě vzorkování  $k+1$ . Tato strategie výpočtu zákona řízení se nazývá *princip klouzavého horizontu* a je to jeden z hlavních pojmů v prediktivním řízení.

Když se označí první řádek matice  $(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T$  jako  $\mathbf{K}$ , potom lze první element z vypočítané trajektorie optimálního řízení vypočítat pomocí vztahu

$$\Delta u(k) = \mathbf{K}(\mathbf{w} - \mathbf{y}_0) \quad (5.31)$$

Tedy, když v budoucnosti není žádný rozdíl mezi volnou odezvou a žádanou hodnotou regulované veličiny, potom je vypočítaný přírůstek řízení nulový. Jestliže existují nenulové rozdíly v budoucnosti, potom je jim přírůstek řízení přímo úměrný faktorem  $\mathbf{K}$ .

Když se shrne celý postup, je důležité si všimnout, že pro výpočet jsou potřebné jen dvě charakteristiky procesu: volná odezva  $\mathbf{y}_0$ , která se mění v každé periodě vzorkování a přechodová charakteristika  $G(z)$ , kterou je v případě časově invariantních lineárních systémů potřebné vypočítat jen jednou. I Hessova matice, kterou je zapotřebí invertovat, je závislá jen na  $G(z)$  a tedy může být vypočítána dopředu. Vypočítané řízení je tedy závislé jen na váhovém součtu minulých vstupů a výstupů obsažených v členu  $\mathbf{y}_0$  a reprezentuje tedy lineární zákon řízení.

### 5.3 Odvození prediktoru

Pro objasnění výše uvedených principů odvození se uvažuje jednoduchý jednorozměrný proces popsáný CARIMA modelem. Jestliže je uvažován CARIMA model pro  $C(z^{-1})=1$ , lze proces popsat rovnicí ve tvaru

$$\Delta A(z^{-1})y(k) = B(z^{-1})\Delta u(k-1) + e_s(k). \quad (5.32)$$

Předpokládá se, že polynomy modelu jsou ve tvaru  $A(z^{-1})=1+a_1z^{-1}+a_2z^{-2}$ ;  $B(z^{-1})=b_1z^{-1}+b_2z^{-2}$ . Po dosazení jednotlivých polynomů do CARIMA modelu se získá

$$(1-z^{-1})(1+a_1z^{-1}+a_2z^{-2})y(k)=(b_1z^{-1}+b_2z^{-2})\Delta u(k)+e_s(k), \quad (5.33)$$

$$\left[1+(a_1-1)z^{-1}+(a_1-a_2)z^{-2}-a_2z^{-3}\right]y(k)=(b_1z^{-1}+b_2z^{-2})\Delta u(k)+e_s(k). \quad (5.34)$$

V časové reprezentaci se obdrží rovnice

$$y(k)=(1-a_1)y(k-1)+(a_1-a_2)y(k-2)+a_2y(k-3)+b_1\Delta u(k-1)+b_2\Delta u(k-2)+e_s(k). \quad (5.35)$$

Dále se uvažuje účelová funkce (4.23) s parametry  $N_1=1$ ,  $N_2=3$ ,  $N_u=2$ . Jestliže položíme  $e_s(k+i)=0$  získáme jednotlivé predikce výstupu ve tvaru

$$\begin{aligned} \hat{y}(k+1) &= (1-a_1)y(k)+(a_1-a_2)y(k-1)+a_2y(k-2)+b_1\Delta u(k)+b_2\Delta u(k-1) \\ \hat{y}(k+2) &= (1-a_1)y(k+1)+(a_1-a_2)y(k)+a_2y(k-1)+b_1\Delta u(k+1)+b_2\Delta u(k) \\ \hat{y}(k+3) &= (1-a_1)y(k+2)+(a_1-a_2)y(k+1)+a_2y(k)+b_1\Delta u(k+2)+b_2\Delta u(k+1) \end{aligned} \quad (5.36)$$

Na základě předpokladů o horizontech ( $N_u=2$ ) je člen  $\Delta u(k+2)$  rovný nule. Vyšší predikce lze vyjádřit na základě nižších predikcí. Pro praktické nasazení při řízení konkrétních procesů je žádoucí, aby bylo možno volit libovolně jednotlivé horizonty vzhledem k dynamice řízeného procesu. Predikce je výhodné v tomto případě počítat pomocí rekurzivního algoritmu. K sestavení těchto rekurzivních algoritmů lze přistupovat různým způsobem. Dále bude popsán způsob založený na přímém vyjádření vyšších predikcí pomocí predikcí nižších. V tomto případě je nutno manuálně spočítat postupným dosazováním matici volné odezvy  $y_0$  a matici dynamiky  $\mathbf{G}$  pro první tři predikce (nutný počet manuálně spočítaných predikcí závisí na řádu modelu procesu). Postupným dosazováním a úpravami se získají první tři predikce v následujícím tvaru

$$\begin{aligned} \begin{bmatrix} \hat{y}(k+1) \\ \hat{y}(k+2) \\ \hat{y}(k+3) \end{bmatrix} &= \begin{bmatrix} g_1 & 0 \\ g_2 & g_1 \\ g_3 & g_2 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \end{bmatrix} + \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} y(k) \\ y(k-1) \\ y(k-2) \\ \Delta u(k-1) \end{bmatrix} = \\ &= \begin{bmatrix} b_1 & 0 \\ b_1(1-a_1)+b_2 & b_1 \\ (a_1-a_2)b_1+(1-a_1)^2b_1+(1-a_1)b_2 & (1-a_1)b_1+b_2 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \end{bmatrix} + \\ &+ \begin{bmatrix} (1-a_1) & (a_1-a_2) & a_2 & b_2 \\ (1-a_1)^2+(a_1-a_2) & (1-a_1)(a_1-a_2)+a_2 & a_2(1-a_1) & (1-a_1)b_2 \\ (1-a_1)^3+2(1-a_1)(a_1-a_2)+a_2 & (1-a_1)^2(a_1-a_2)+a_2(1-a_1)+(a_1-a_2)^2 & a_2(1-a_1)^2+(a_1-a_2)a_2 & (1-a_1)^2b_2+(a_1-a_2)b_2 \end{bmatrix} \begin{bmatrix} y(k) \\ y(k-1) \\ y(k-2) \\ \Delta u(k-1) \end{bmatrix} \end{aligned} \quad (5.37)$$

Další postup rekurze lze rozdělit na rekurzivní výpočet dalších řádků matice volné odezvy a matice dynamiky. Na základě tří předchozích predikcí je opakovaně počítán následující řádek matice volné odezvy následujícím způsobem

$$\begin{aligned} p_{41} &= (1 - a_1)p_{31} + (a_1 - a_2)p_{21} + a_2p_{11} \\ p_{42} &= (1 - a_1)p_{32} + (a_1 - a_2)p_{22} + a_2p_{12} \cdot \\ p_{43} &= (1 - a_1)p_{33} + (a_1 - a_2)p_{23} + a_2p_{13} \end{aligned} \quad (5.38)$$

V následujícím kroku je první řádek vynechán a další predikce se počítá z posledních tří řádků včetně řádku vypočítaného v předchozím kroku. Cyklickým opakováním tohoto postupu lze snadno spočítat libovolný počet řádků této matice.

Rekurze matice dynamiky je podobná. V tomto případě stačí opakovaně počítat následující prvek prvního sloupce obdobně jako v předchozím případě a následující sloupce posunout způsobem patrným ze vzorce (5.37). Jestliže je zvolený řídicí horizont nižší než horizont predikce, je nutno pouze redukovat počet sloupců v matici dynamiky. Výpočet nového prvku je prováděn následujícím způsobem

$$g_4 = (1 - a_1)g_3 + (a_1 - a_2)g_2 + a_2g_1. \quad (5.39)$$

## Výpočet optimální sekvence řízení

### 1. Příklad bez omezení vstupního signálu

Nyní je cílem určit algoritmus řízení podle vztahu (5.31), ve kterém  $\mathbf{K}$  označuje první řádek matice

$$\mathbf{R} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1N_2} \\ r_{21} & r_{22} & r_{23} & \cdots & r_{2N_2} \end{bmatrix}. \quad (5.40)$$

Jedná se tedy o řádkový vektor

$$\mathbf{K} = [r_{11} \quad r_{12} \quad \cdots \quad r_{1N_2}], \quad (5.41)$$

jejíž prvky násobí řídicí veličinu  $w(k)$ . Koeficienty volné odezvy jsou rovny záporným hodnotám prvků řádkového vektoru

$$\mathbf{Q} = \mathbf{K} \mathbf{y}_0 = [q_{11} \quad q_{12} \quad q_{13} \quad q_{14}]. \quad (5.42)$$

Řídicí zákon je tedy dán ve tvaru

$$\Delta u(k) = -q_{11}y(k) - q_{12}y(k-1) - q_{13}y(k-2) - q_{14}\Delta u(k-1) + r_{11}w(k+1) + r_{12}w(k+2) + \cdots + r_{1N_2}w(k+N_2) \quad (5.43)$$

$$u(k) = \Delta u(k) + u(k-1). \quad (5.44)$$

## 2. Příklad s omezením vstupního signálu

Obecnou formulaci GPC s omezeními je možno napsat následovně

$$\min_{\Delta u} 2\mathbf{g}^T \Delta \mathbf{u} + \Delta \mathbf{u}^T \mathbf{H} \Delta \mathbf{u}, \quad (5.45)$$

vzhledem k  $A\Delta \mathbf{u} \leq \mathbf{b}$ .

Jednotlivé matice v omezující nerovnosti lze pro zadaný konkrétní případ definovat následujícím způsobem

$$A = \begin{bmatrix} \mathbf{T} \\ -\mathbf{T} \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \mathbf{1}u_{\min} - \mathbf{1}u(k-1) \\ -\mathbf{1}u_{\max} + \mathbf{1}u(k-1) \end{bmatrix}. \quad (5.46)$$

Pro řídicí horizont  $N_U = 2$  platí, pro vyšší horizonty je nutné matice a vektory rozšířit

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ -1 & 0 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \end{bmatrix} \geq \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_{\min} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} [u(k-1)] - \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_{\max} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} [u(k-1)]. \quad (5.47)$$

Tuto optimalizační úlohu je možno řešit pomocí kvadratického programování. V této práci byla pro její řešení použita funkce Matlabu Optimization Toolbox quadprog.

## 6 IDENTIFIKACE

### 6.1 Metoda nejmenších čtverců

Metoda nejmenších čtverců patří mezi metody regrese analýzy, které se používají pro vyšetřování statických i dynamických vztahů mezi veličinami ve vyšetřovaném objektu. Obecně je uvažován jednorozměrný stochastický proces, který je popsán modelem ARX. Pro vektor dat a vektor parametrů se předpokládá, že stupeň polynomu  $a$  se rovná stupni polynomu  $b$  a tento stupeň odpovídá řádu soustavy  $n$ . Dále se předpokládá, že stupeň polynomu  $d$  je nulový [9].

Pro vektor dat platí

$$\boldsymbol{\varphi}^T(k-1) = [-y(k-1) \quad -y(k-2) \quad \cdots \quad -y(k-n) \quad u(k-1) \quad u(k-2) \quad \cdots \quad u(k-n)], \quad (6.1)$$

a pro vektor parametrů platí

$$\boldsymbol{\theta}^T = [a_1 \quad a_2 \quad \cdots \quad a_n \quad b_1 \quad b_2 \quad \cdots \quad b_n]. \quad (6.2)$$

Lze pak generování výstupní veličiny  $y(k)$  v jednotlivých krocích znázornit maticovou rovnicí ve tvaru

$$\mathbf{y} = \mathbf{F}\boldsymbol{\theta} + \mathbf{e}, \quad (6.3)$$

kde:

-  $\mathbf{F}$  - je matice o rozměru  $(N-n, 2n)$  a má tvar

$$\mathbf{F} = \begin{bmatrix} -y(n) & -y(n-1) & \cdots & -y(1) & u(n) & u(n-1) & \cdots & u(1) \\ -y(n+1) & -y(n) & \cdots & -y(2) & u(n+1) & u(n) & \cdots & u(2) \\ \vdots & & & & & & & \vdots \\ -y(N-1) & -y(N-2) & \cdots & -y(N-n) & u(N-1) & u(N-2) & \cdots & u(N-n) \end{bmatrix}, \quad (6.4)$$

-  $\mathbf{y}$  - je vektor naměřených dat o rozměru  $(N-n)$  -  $\mathbf{y}^T = [y(n+1) \quad y(n+2) \quad \cdots \quad y(N)]$ ,

-  $\mathbf{e}$  - je chybový vektor o rozměru  $(N-n)$  -  $\mathbf{e}^T = [e(n+1) \quad e(n+2) \quad \cdots \quad e(N)]$ ,

kde:

-  $N$  - je počet naměřených dat.

Z rovnice 6.3 lze určit chybu

$$\mathbf{e} = \mathbf{y} - \mathbf{F}\boldsymbol{\theta}. \quad (6.5)$$

Parametry soustavy se získají na základě kritéria minimálního součtu kvadrátů chyby  $e$

$$J = e^T e = (y - F\Theta)^T (y - F\Theta). \quad (6.6)$$

Minimum tohoto kvadratického kritéria se získá pokud se první derivace tohoto kritéria podle parametrů  $\Theta$  položí rovna nule

$$\left. \frac{\partial J}{\partial \Theta} \right|_{\Theta = \hat{\Theta}} = 0. \quad (6.7)$$

Základní maticový tvar pro odhad parametrů metodou nejmenších čtverců se získá řešením rovnice (6.7) a to ve tvaru

$$\hat{\Theta} = (F^T F)^{-1} F^T y. \quad (6.8)$$

Vztah (6.8) slouží pro jednorázové určení parametrů modelu procesu z  $N$  naměřených dat.

## 6.2 Rekurzivní metoda nejmenších čtverců

Pro předcházející metodu nejmenších čtverců je typické tzv. dávkové zpracování dat, které vede na jednorázovou identifikaci. Po naměření dat v předem daném časovém úseku se metoda nejmenších čtverců použije pro výpočet odhadu neznámých parametrů modelu. Pokud však dojde k novému experimentu a přibudou tak nové data, je třeba provést celý výpočet znovu. Z tohoto faktu vyplývá, že je tato metoda velmi náročná na velikost paměti, kde je třeba uchovávat veškerá data, a zvyšováním počtu naměřených dat také dochází k nárůstu výpočetní náročnosti. Díky těmto problémům všeobecně u všech jednorázově identifikujících algoritmů nabyly velkého významu algoritmy rekurzivní. Tyto rekurzivní metody jsou obecně vhodné pro průběžnou identifikaci, což je odhad parametrů modelu v reálném čase. Nově naměřená dat se používají pouze k opravě předchozích parametrů, což vede k poklesu výpočetní náročnosti a k poklesu nároků na výpočetní techniku. [9]

Zde je uvedeno rekurzivní řešení pro metodu nejmenších čtverců uvedené vztahem (6.8). Je-li použito pouze  $k-1$  pozorování, lze tuto rovnici modifikovat na tvar

$$\hat{\Theta}(k-1) = (F_{k-1}^T F_{k-1})^{-1} F_{k-1}^T y(k-1). \quad (6.9)$$

kde:



-  $\mathbf{y}(k-1)$  - je vektor výstupních dat do kroku  $k-1$  -  $\mathbf{y}^T(k-1) = [y(1) \ y(2) \ \dots \ y(k-1)]$ ,

-  $\hat{\boldsymbol{\theta}}(k-1)$  - je vektor optimálních odhadů parametrů na základě  $k-1$  pozorování -

$$\hat{\boldsymbol{\theta}}^T(k-1) = [\hat{a}_1(k-1) \ \hat{a}_2(k-1) \ \dots \ \hat{a}_n(k-1) \ \hat{b}_1(k-1) \ \hat{b}_2(k-1) \ \dots \ \hat{b}_n(k-1)],$$

-  $\mathbf{F}(k-1)$  - je modifikovaná matice  $\mathbf{F}$  pro  $k-1$  pozorování -

$$\mathbf{F}_{k-1} = \begin{bmatrix} -y(n) & -y(n-1) & \dots & -y(1) & u(n) & u(n-1) & \dots & u(1) \\ -y(n+1) & -y(n) & \dots & -y(2) & u(n+1) & u(n) & \dots & u(2) \\ \vdots & & & & & & & \vdots \\ -y(k-1) & -y(k-2) & \dots & -y(k-n) & u(k-1) & u(k-2) & \dots & u(k-n) \end{bmatrix}.$$

Je uskutečněno další  $k$ -té pozorování  $\mathbf{y}(k)$

$$\mathbf{y}(k) = \begin{bmatrix} \mathbf{y}(k-1) \\ y(k) \end{bmatrix}. \quad (6.10)$$

Potom lze matici  $\mathbf{F}_k$  přepsat do tvaru

$$\mathbf{F}_k = \begin{bmatrix} -y(n) & -y(n-1) & \dots & -y(1) & u(n) & u(n-1) & \dots & u(1) \\ \vdots & & & & & & & \vdots \\ -y(k-2) & -y(k-3) & \dots & -y(k-1-n) & u(k-2) & u(k-3) & \dots & u(k-1-n) \\ -y(k-1) & -y(k-2) & \dots & -y(k-n) & u(k-1) & u(k-2) & \dots & u(k-n) \end{bmatrix}, \quad (6.11)$$

což lze napsat jako

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{F}_{k-1} \\ \boldsymbol{\varphi}^T(k) \end{bmatrix}, \quad (6.12)$$

kde poslední řádek matice  $\mathbf{F}_k$  je označen jako vektor

$$\boldsymbol{\varphi}^T(k) = [-y(k-1) \ -y(k-2) \ \dots \ -y(k-n) \ u(k-1) \ u(k-2) \ \dots \ u(k-n)]. \quad (6.13)$$

Pro  $k$ -tou pozorovatelnou hodnotu  $y(k)$  lze vektorově napsat

$$y(k) = \boldsymbol{\theta}^T \boldsymbol{\varphi}(k) + e(k). \quad (6.14)$$

kde:

$$\boldsymbol{\theta}^T = [a_1(k) \ a_2(k) \ \dots \ a_n(k) \ b_1(k) \ b_2(k) \ \dots \ b_n(k)].$$

Dále je potřeba definovat matici  $\mathbf{C}(k-1)$

$$\mathbf{C}(k-1) = (\mathbf{F}_{k-1}^T \mathbf{F}_{k-1})^{-1}. \quad (6.15)$$

Sérií úprav lze získat rekurzivní algoritmus v obecném tvaru

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \mathbf{L}\hat{e}(k), \quad (6.16)$$

kde:

$$\mathbf{L} = \frac{\mathbf{C}(k-1)\boldsymbol{\varphi}(k)}{1 + \boldsymbol{\varphi}^T(k)\mathbf{C}(k-1)\boldsymbol{\varphi}(k)}, \quad (6.17)$$

je vektor zesílení a

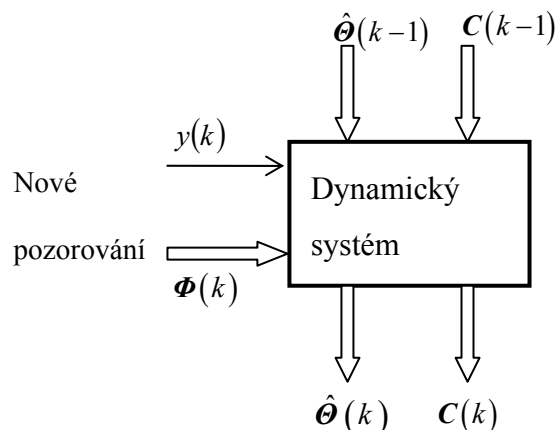
$$\hat{e}(k) = y(k) - \boldsymbol{\varphi}^T(k)\hat{\boldsymbol{\theta}}(k-1). \quad (6.18)$$

je chyba predikce.

Pro čtvercovou matici  $\mathbf{C}(k)$  o rozměru  $(2n, 2n)$  platí

$$\mathbf{C}(k) = \mathbf{C}(k-1) - \mathbf{L}\boldsymbol{\varphi}^T(k)\mathbf{C}(k-1), \quad (6.19)$$

Postup výpočtu znázorňuje obrázek 6.1. Výpočet je zahájen z počátečních hodnot  $\hat{\boldsymbol{\theta}}(0)$  a  $\mathbf{C}(0)$ , které se mohou určit například pomocí apriorní informace.



Obrázek 6.1 Postup při rekurzivní metodě  
nejmenších čtverců

Pro určení kvality identifikace rekurzivní metodou nejmenších čtverců slouží následující vztah

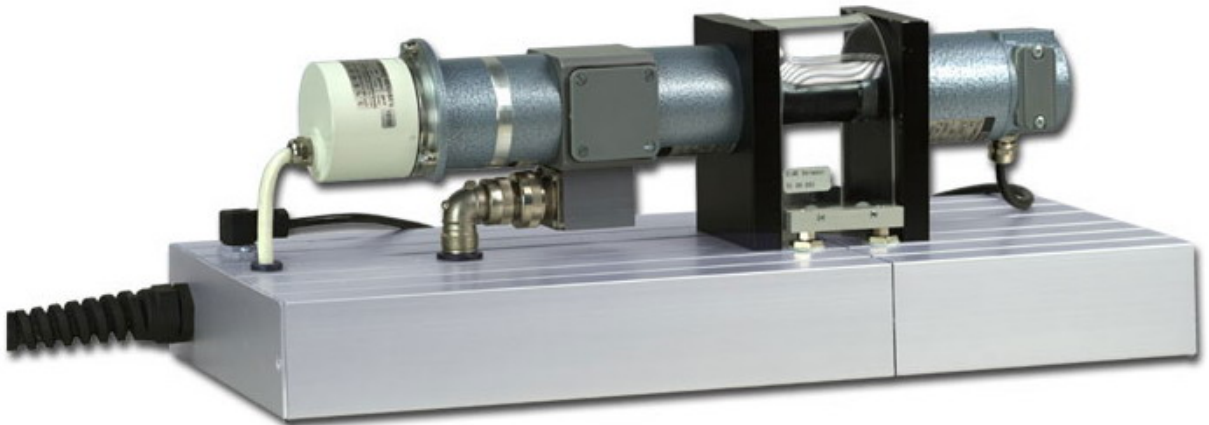
$$S_{ID} = \frac{1}{b-a+1} \sum_{k=a}^b [\hat{e}(k)]^2, \quad (6.20)$$

kde:

- $a$  - je počáteční krok identifikace,
- $b$  - je konečný krok identifikace.

## II. PRAKTICKÁ ČÁST

## 7 POPIS LABORATORNÍHO MODELU AMIRA DR300

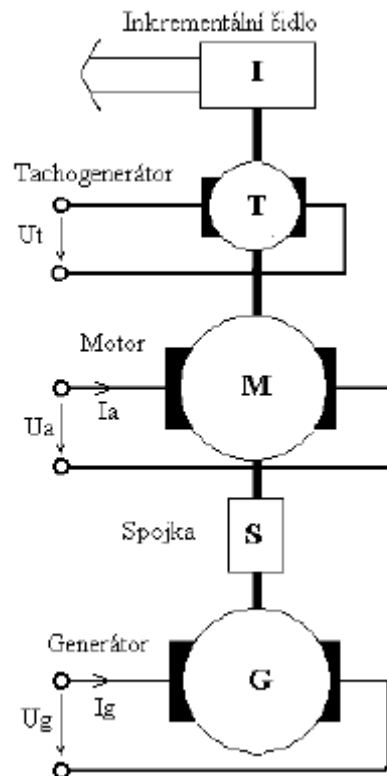


Obrázek 7.1 Laboratorní model AMIRA DR300

Laboratorní model AMIRA DR300 představuje technickou realizaci nelineárního jednorozměrného systému, který slouží k ověřování funkčnosti teoretických algoritmů pro řízení a regulaci soustav v praxi. Tento model lze také využít jako zdroj dat, která jsou nutná pro různé simulační a identifikační procesy, dále jej lze také využít pro řešení jiných problémů týkajících se oblasti automatizace a řízení, popřípadě týkajících se i jiných podobných oblastí. Model je zakoupen od firmy Amira, která má své sídlo v Německu.

Laboratorní model se skládá ze dvou základních částí. První částí je samotný mechanismus a druhou je převodní skříň.

Mechanismus je tvořen permanentně buzeným stejnosměrným motorem, jehož výstupem je proud kotvy, který je poskytován proudovým regulačním obvodem. Senzory pro výstupní signál, kterým je rychlost otáčení motoru, je tachogenerátor a přírůstkový snímač polohy. Volný konec hřídele prvního motoru je pomocí pevné spojky pevně spojen s hřídelí druhého motoru, který je identický s prvním. Tento druhý motor se využívá jako generátor chyb a jeho vstupní signál lze libovolně nastavovat.



Obrázek 7.2 Schéma modelu

Převodní skříň obsahuje napájecí zdroje, převodníky, zesilovače pro převod signálu z V/V karty na výkonové veličiny pro motory a zesilovače pro převod signálů ze senzorů na unifikované signály pro vstupy V/V karty. Z čelní strany převodní skříně se zasouvají zásuvné moduly. Současná konfigurace obsahuje jen základní moduly. Mezi tyto základní moduly patří modul motoru, generátoru, napájení, řízení a modul pro externí řízení a snímání. Dále lze například přidat modul pro PI regulátor a modul pro vytváření signálových chyb. Přítomnost napájecích napětí a způsob řízení je indikován příslušnými LED diodami.

Na modelu lze nastavovat napětí na motoru a generátoru. Snímat lze napětí na tachogenerátoru a jednotlivé proudy procházející motorem a generátorem.

Rovnice popisující chování stejnosměrného motoru s cizím buzením můžeme zapsat ve tvaru

$$L \frac{di(t)}{dt} = -Ri(t) - k_e \omega(t) + u_m(t)$$

$$J \frac{d\omega(t)}{dt} = k_m i(t) - b\omega(t) - m_z(t)$$

$$\frac{d\varphi(t)}{dt} = \omega(t)$$

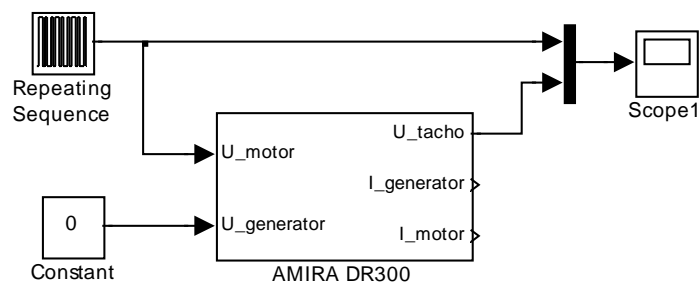
kde  $i$  [A] je proud motoru,  $\omega$  [ $s^{-1}$ ] jsou otáčky motoru,  $\varphi$  [rad] je úhel natočení hřídele motoru,  $u$  [V] je vstupní napětí motoru,  $m_z$  [Nm] je vnější zatěžovací moment,  $R$  [ $\Omega$ ] je elektrický odpor motoru,  $L$  [H] je indukčnost motoru,  $J$  [ $kgm^2s^{-1}$ ] je moment setrvačnosti motoru,  $b$  je konstanta tření motoru,  $k_e$  [ $sV^{-1}$ ] je elektrická konstanta motoru a  $k_m$  [ $kgm^2s^{-1}$ ] je mechanická konstanta motoru.

## 8 URČENÍ HORIZONTŮ PREDIKTIVNÍHO ŘÍZENÍ

Nejprve bylo třeba naměřit statickou charakteristiku laboratorního modelu AMIRA DR300, aby bylo možné vybrat oblast pro měření přechodové charakteristiky, která je důležitá pro určení horizontů prediktivního řízení tohoto modelu.

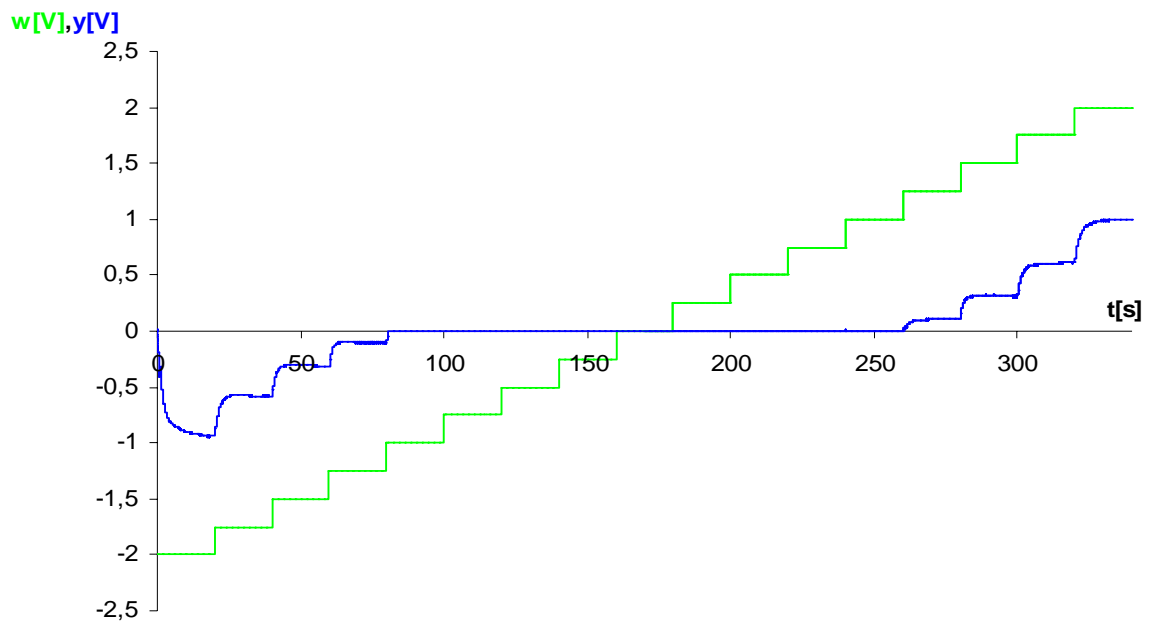
### 8.1 Statická charakteristika

Pro měření statické charakteristiky laboratorního modelu AMIRA DR300 bylo použito simulační schéma na obrázku 8.1 vytvořené pomocí programu MATLAB/SIMULINK:



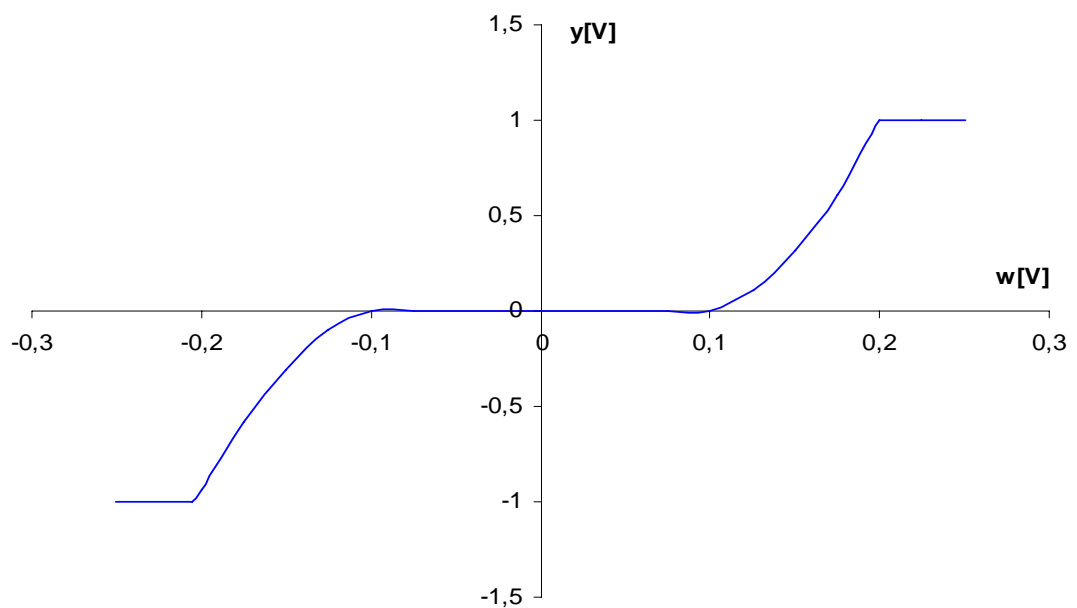
Obrázek 8.1 Simulinkové schéma pro měření statické charakteristiky

Ze simulinkového schématu lze vidět, že žádaná hodnota je přiváděna do soustavy jako napětí motoru a jako výstup se snímá napětí na tachogenerátoru. Pro posloupnost žádané veličiny byl naměřen následující časový průběh odezvy napětí tachogenerátoru na posloupnost hodnot vstupního napětí motoru. Tento průběh je znázorněn na následujícím obrázku 8.2:



Obrázek 8.2 Časový průběh měření statické charakteristiky

Dále pro získání výsledné statické charakteristiky, byly z hodnot odezvy pro jednotlivé skoky od času, kdy byly zhruba konstantní, vytvořeny průměry, a tak byly získány jednotlivé body statické charakteristiky. Tato charakteristika je uvedena na obrázku 8.3:



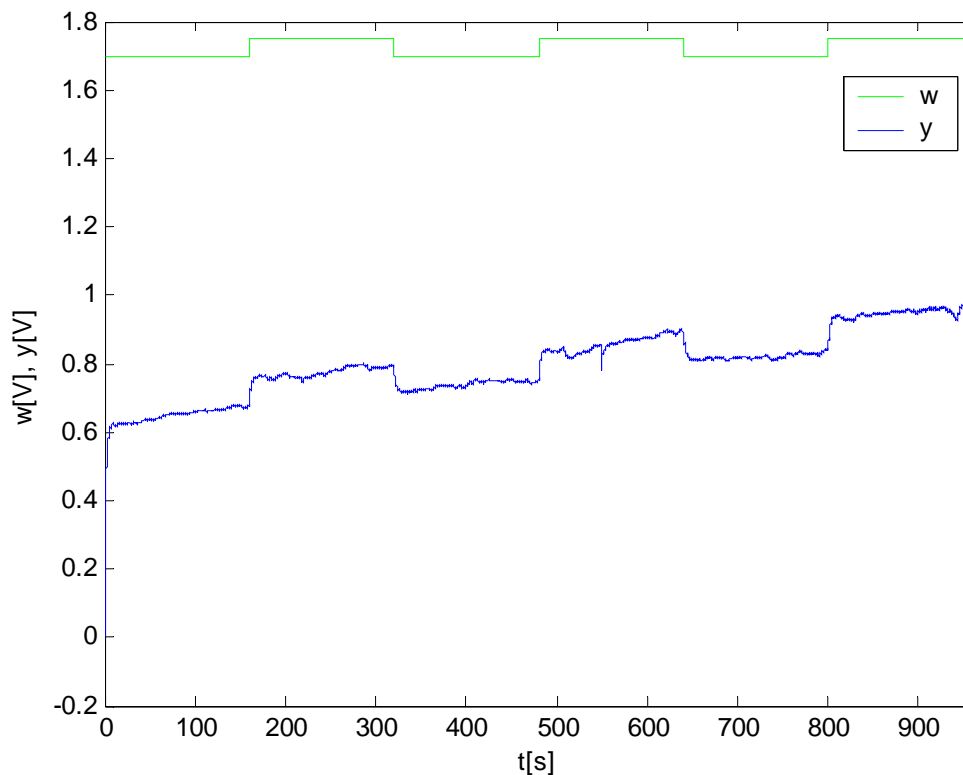
Obrázek 8.3 Výsledná statická charakteristika



Z výše uvedené statické charakteristiky je zřejmé, že pásmo necitlivosti je v oblasti vstupního napětí od -1V až po 1V, kdy se servomechanismus netočí. Dále je zřejmé, že saturace nastává, pokud vstupní napětí přeroste hodnotu zhruba 2V a to pro oba směry otáčení (tedy pokud poklesne pod -2V).

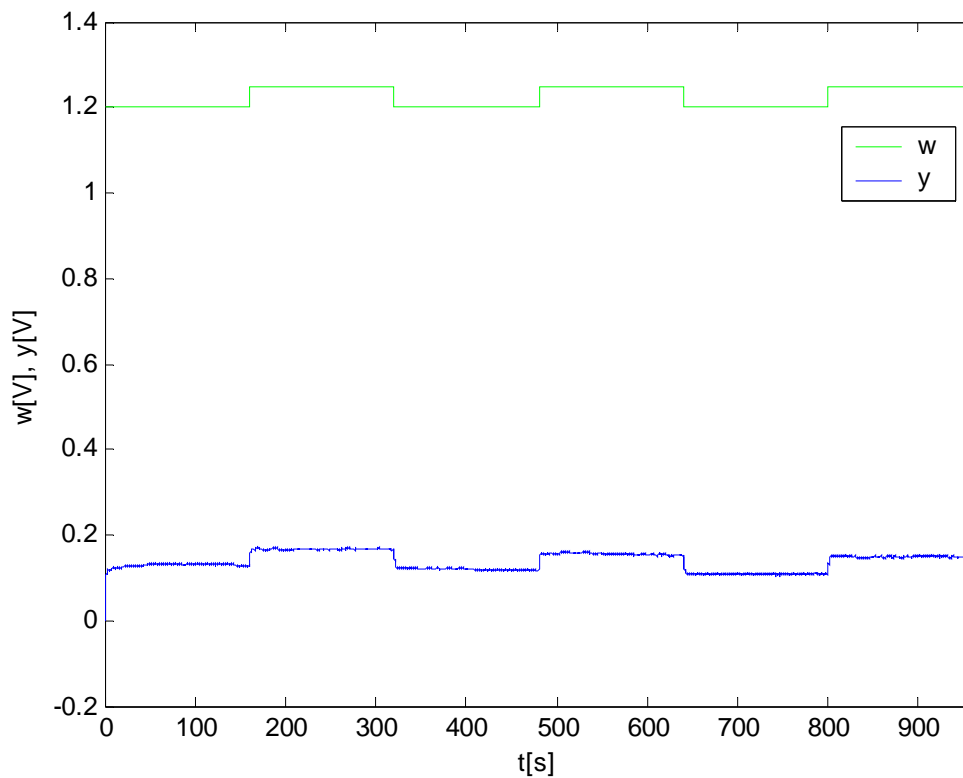
## 8.2 Přechodová charakteristika

Měření přechodové charakteristiky bylo provedeno v oblasti, kde byl nárůst odezvy modelu na vzrůstající žádanou veličinu nejvíce lineární. Toto měření je zobrazeno na obrázku 8.4:



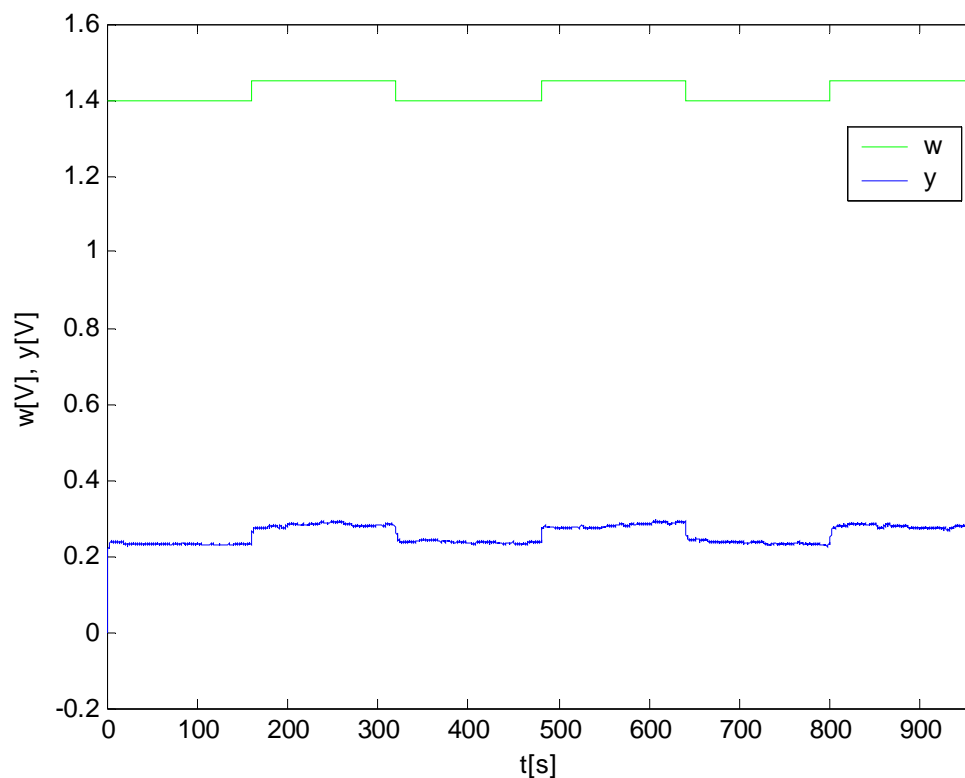
Obrázek 8.4 Přechodové charakteristiky pro skoky žádané veličiny z 1.7V na 1.75V

Z grafu lze vidět, že pro dané skoky žádané veličiny má celý průběh zvláštní tendenci neustále růst. Vzhledem k tomuto mnou nevysvětlitelnému jevu bylo dalším krokem změření přechodových charakteristik z oblasti nižšího vstupního napětí. Tento časový průběh je uveden na obrázku 8.5:



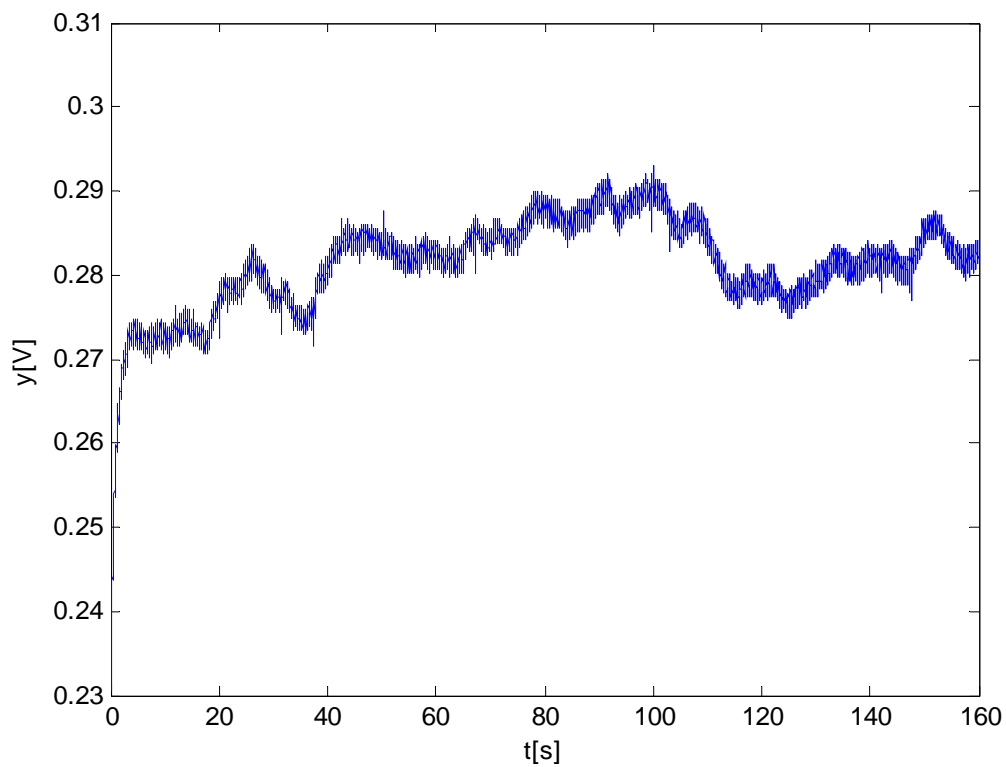
Obrázek 8.5 Přejchodové charakteristiky pro skoky žádané veličiny z 1.2V na 1.25V

Z grafu lze vidět, že pro dané skoky žádané veličiny má celý průběh opět zvláštní trend, ale tentokrát klesající. Dalším krokem tedy bylo metodou pokus omyl najít takové vhodné skoky žádané veličiny tak, aby se tyto trendy vyrušily. Jediné co se dalo předpokládat bylo to, že tyto skoky se budou nacházet někde mezi hodnotami skoků, pro které byly naměřené předchozí charakteristiky. Na obrázku 8.6 je zobrazen průběh právě pro experimentálně získané skoky žádané veličiny, kdy odezva na tyto skoky nemá přibližně tendenci ani klesat ani růst.

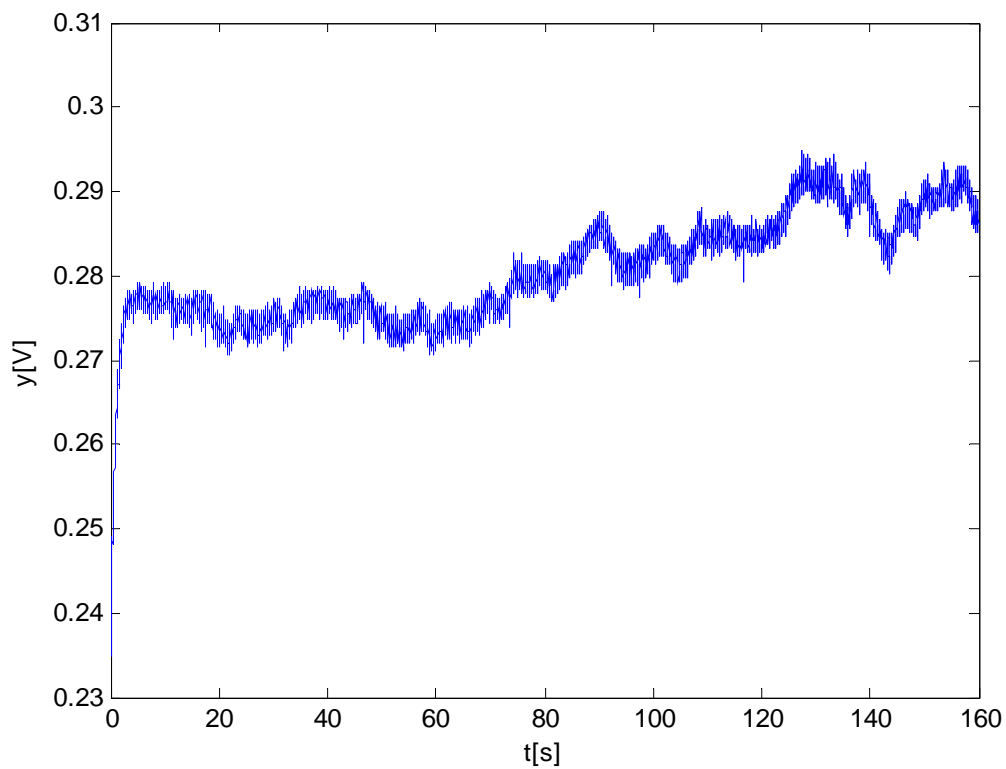


Obrázek 8.6 Přechodové charakteristiky pro skoky žádané veličiny z 1.4V na 1.45V

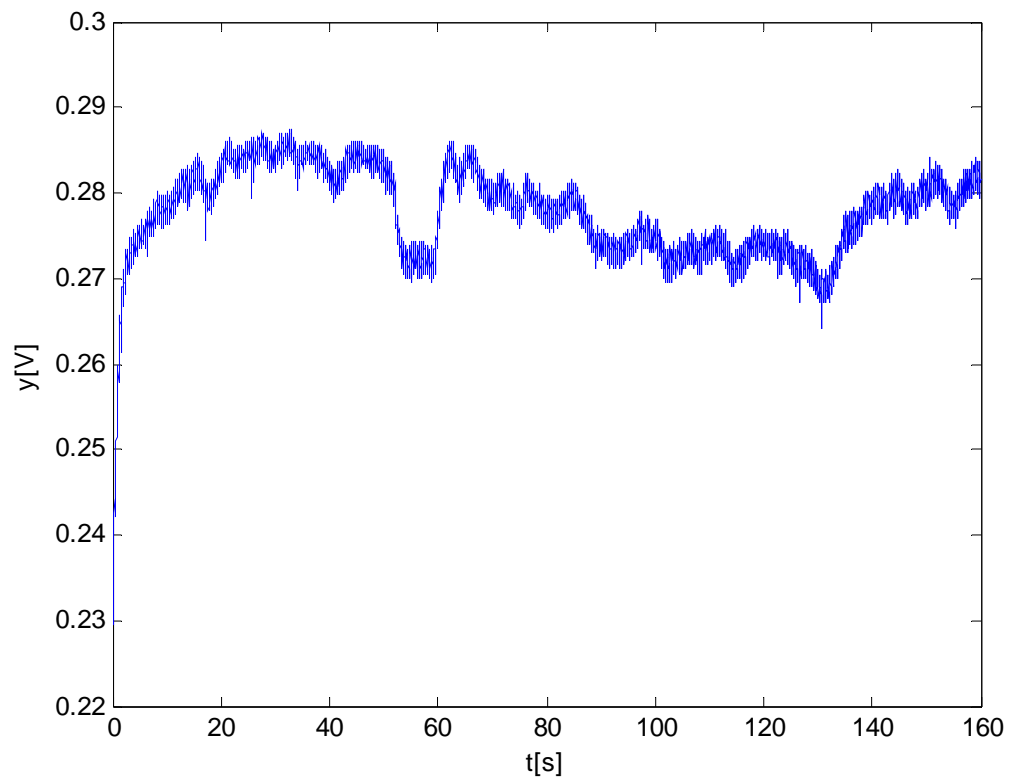
Pro určení výsledných horizontů byl proveden průměr těchto přechodových charakteristik pro skoky žádané veličiny z 1.4 V na 1.45 V, aby došlo ke statistickému potlačení případných chyb. Před tím, než bylo možné udělat jejich průměr, bylo nutné celý průběh rozdělit na tři části a všechny tři takto vzniklé přechodové charakteristiky posunout do počátku časové osy. Vznikly tak následující tři přechodové charakteristiky:



Obrázek 8.7 Přejchodová charakteristika s počátkem v čase 160 s

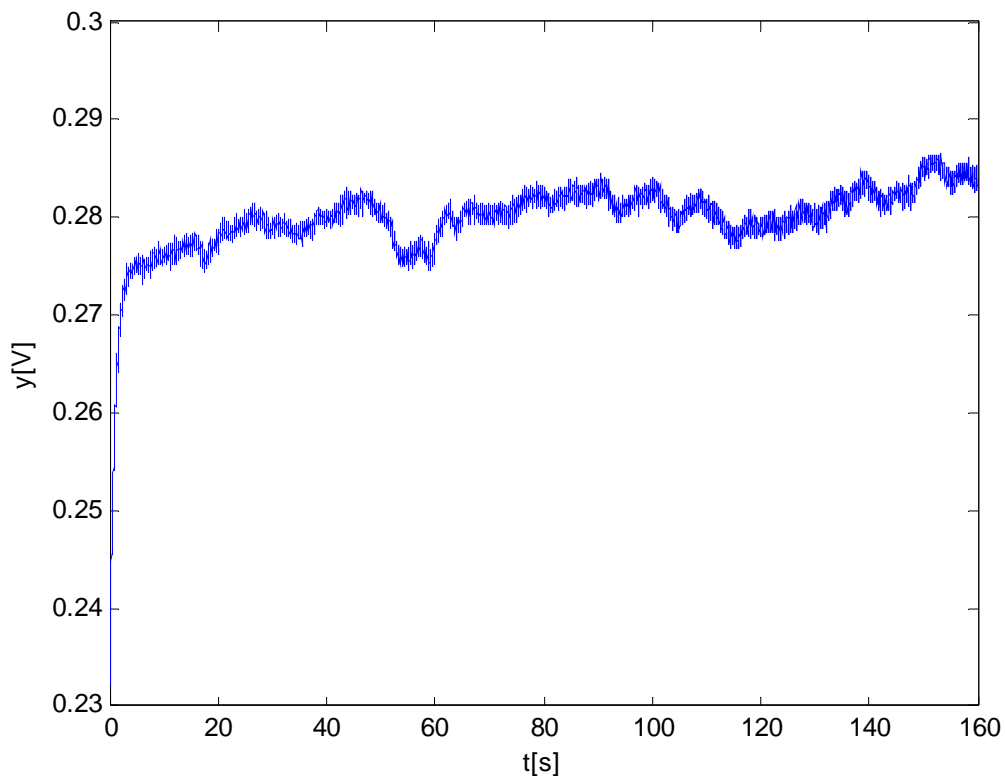


Obrázek 8.8 Přejchodová charakteristika s počátkem v čase 480 s



Obrázek 8.9 Přechodová charakteristika s počátkem v čase 800s

Výsledná přechodová charakteristika vznikla jako průměr těchto tří charakteristik a je znázorněna na obrázku 8.10:



Obrázek 8.10 Výsledná přechodová charakteristika

Pro určení jednotlivých horizontů byly použity vztahy z kapitoly 4.2. Pro určení horizontu  $N_1$  platí:  $N_1 = T_d + 1$ . Z obrázku 7.10 lze vidět, že dopravní zpoždění  $T_d$  je nulové a tudíž  $N_1 = 1$ . Pro horizont  $N_2$  platí:  $N_2 = T_{90} = t((0.9(y(\infty) - y(0))) + y(0)) - t((0.1(y(\infty) - y(0))) + y(0))$ , kde  $y(\infty)$  je hodnota na které se přechodový proces ustálí. Z obrázku 8.10 lze vidět, že výsledná přechodová charakteristika je ještě poměrně hodně zašuměná, proto byla ustálená hodnota získána jako průměr výstupních hodnot od času 40 s. Takto byla získána ustálená výstupní hodnota rovna  $0.281 \text{ s}^{-1}$ . Tím se získají hodnoty: 90% ustálené hodnoty se rovná  $(0.9(0.281 - 0.23)) + 0.23 = 0.2759 \text{ s}^{-1}$  a 10% ustálené hodnoty se rovná  $(0.1(0.281 - 0.23)) + 0.23 = 0.2351 \text{ s}^{-1}$ . Pro přesnější odečtení hodnot časů odpovídajícím těmto hodnotám odezvy byla použita data, která se použila pro vykreslení této charakteristik. Z těchto dat, ale i z grafu vyplývá, že přechodová charakteristika dosahuje 10% ustálené hodnoty prakticky okamžitě, lze tedy uvažovat s časem 0 s, a 90% dosahuje v čase zhruba 5 s. Díky těmto faktům byl horizont  $N_2$  zvolen 5. Pro určení horizontu  $N_U$  není dán žádný výpočetní vztah, byl tedy náhodně zvolen  $N_U = 2$ .

## 9 IDENTIFIKACE LABORATORNÍHO MODELU AMIRA DR300

Identifikace laboratorního modelu AMIRA DR300 byla provedena jak metodou nejmenších čtverců pro první i druhý řád modelu, tak i rekurzivní metodou nejmenších čtverců taktéž pro první i druhý řád. Jako vstupní signál sloužící pro identifikaci tohoto modelu byl zvolen pseudonáhodný binární signál (PNBS). Pro tento signál byly zvoleny následující parametry:

-  $\Delta t$  - interval hodinových impulsů -  $\Delta t = T = 0.05s$ ,

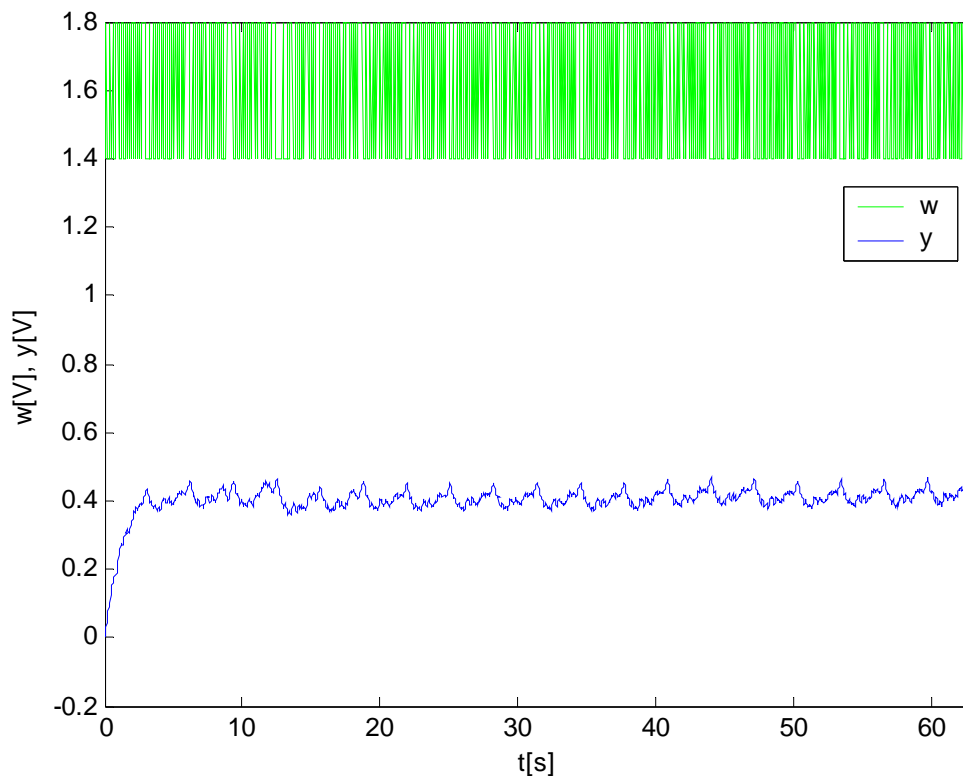
kde:

-  $T$  je perioda vzorkování laboratorního modelu AMIRA DR300,

-  $n$  - počet stupňů posuvného registru -  $n = 6$ .

Takto vytvořený pseudonáhodný binární signál se skládá pouze z jedniček a nul. Bylo tedy potřebné ho upravit tak, aby pokrýval co největší oblast lineární části statické charakteristiky laboratorního modelu AMIRA DR300. Takže výsledný signál obsahuje hodnoty 1.4V a 1.8V, takže pokrývá lineární oblast. Tento signál je generován pomocí m-filu PNBS.m, jehož zdrojový kód je zobrazen v příloze P II.

Tento signál byl vložen na vstup laboratorního modelu AMIRA DR300 a naměřená odezva byla pak použita pro identifikaci. Na obrázku 9.1 je znázorněno měření odezvy na tento pseudonáhodný binární signál:



Obrázek 9.1 Odezva na pseudonáhodný binární signál

Naměřená výstupní data byla dále použita pro všechny čtyři výše uvedené způsoby identifikace, které jsou provedeny pomocí m-filu identifikace.m, jehož zdrojový kód je zobrazen v příloze P III. Dále jsou zde uvedeny jednotlivé výsledné parametry modelu získané jednotlivými způsoby identifikace a výsledné přenosy identifikované soustavy.

Metoda nejmenších čtverců pro první řád:

$$\begin{aligned}
 - a_0 &= 1 & - b_0 &= 0 \\
 - a_1 &= -0.9789 & - b_1 &= 0.0545 \\
 - G_s(z^{-1}) &= \frac{0.0545z^{-1}}{1 - 0.9789z^{-1}}
 \end{aligned}$$

Metoda nejmenších čtverců pro druhý řád:

$$\begin{aligned}
 - a_0 &= 1 & - b_0 &= 0 \\
 - a_1 &= -1.0945 & - b_1 &= -0.0722 \\
 - a_2 &= 0.1717 & - b_2 &= 0.2691
 \end{aligned}$$



$$- G_s(z^{-1}) = \frac{-0.0722z^{-1} + 0.2691z^{-2}}{1 - 1.0945z^{-1} + 0.1717z^{-2}}$$

Rekurzivní metoda nejmenších čtverců pro první řád:

$$\begin{aligned} - a_0 &= 1 & - b_0 &= 0 \\ - a_1 &= -0.9789 & - b_1 &= 0.0545 \end{aligned}$$

$$- G_s(z^{-1}) = \frac{0.0545z^{-1}}{1 - 0.9789z^{-1}}$$

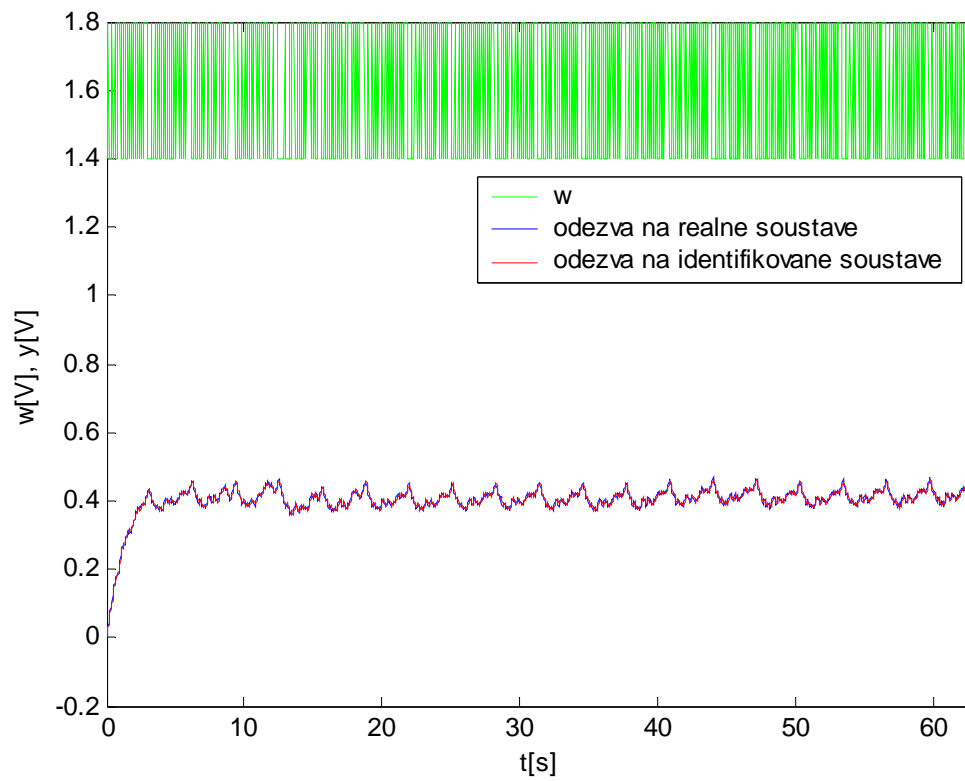
Rekurzivní metoda nejmenších čtverců pro druhý řád:

$$\begin{aligned} - a_0 &= 1 & - b_0 &= 0 \\ - a_1 &= -1.0945 & - b_1 &= -0.0722 \\ - a_2 &= 0.1717 & - b_2 &= 0.2691 \end{aligned}$$

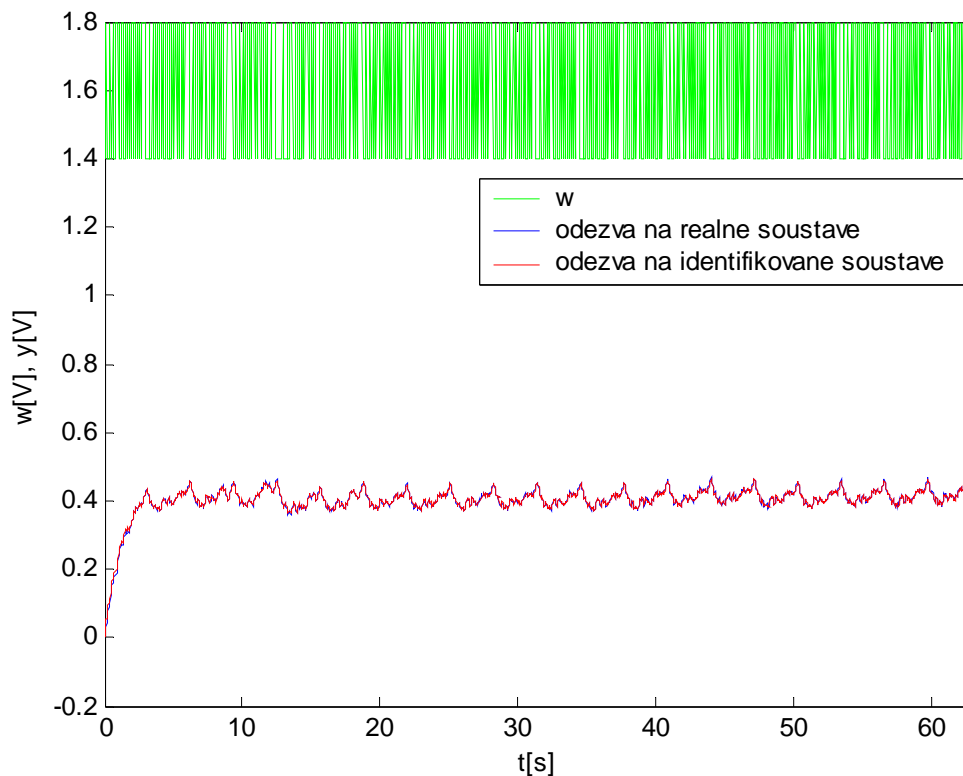
$$- G_s(z^{-1}) = \frac{-0.0722z^{-1} + 0.2691z^{-2}}{1 - 1.0945z^{-1} + 0.1717z^{-2}}$$

Z výsledků lze vidět, že parametry modelu získaných pomocí metod nejmenších čtverců jak pro první tak i druhý řád se shodují s parametry modelu získanými metodami rekurzivními, z čehož lze usuzovat správnost identifikace, protože by tyto parametry měly být stejné.

Pro ověření shody dynamického chování matematického modelu a reálného laboratorního zařízení je porovnána odezva na pseudonáhodný binární signál na reálné soustavě a na simulovaném matematickém modelu a to jak pro první řád tak i pro druhý řád.



Obrázek 9.2 Porovnání odezev reálné soustavy a identifikovaného modelu prvního řádu



Obrázek 9.3 Porovnání odezev reálné soustavy a identifikovaného modelu druhého řádu

Jednotlivé odezvy na identifikovaných modelech byly porovnány pomocí kritéria kvadrátu odchylky odezvy na reálné soustavě od odezvy na identifikované soustavě a to pomocí vztahu  $S_y = \sum_{i=1}^N (y_r(i) - y_i(i))^2$ , kde  $y_r$  je odezva na reálné soustavě,  $y_i$  je odezva na zidentifikovaném modelu a  $N$  je počet naměřených dat. Dále byla kvalita obou rekurzivních metod zhodnocena pomocí vztahu (6.20). Hodnoty těchto kritérií pro první a druhý řád jsou uvedeny v následující tabulce:

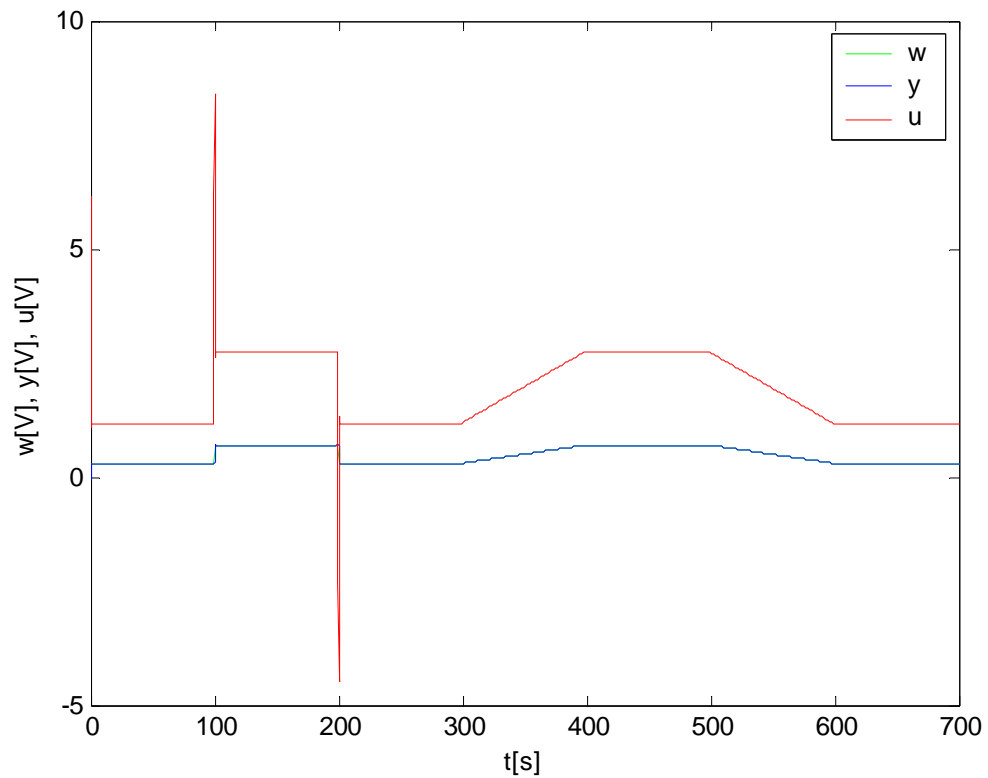
Tabulka 9.1 Hodnoty kritéria kvadrátu odchylky  $S_y$  a kvality identifikace  $S_{ID}$

Řád identifikované soustavy	$S_y$	$S_{ID}$
První řád	0.0663	$5.3735 \cdot 10^{-5}$
Druhý řád	0.0119	$1.4134 \cdot 10^{-5}$

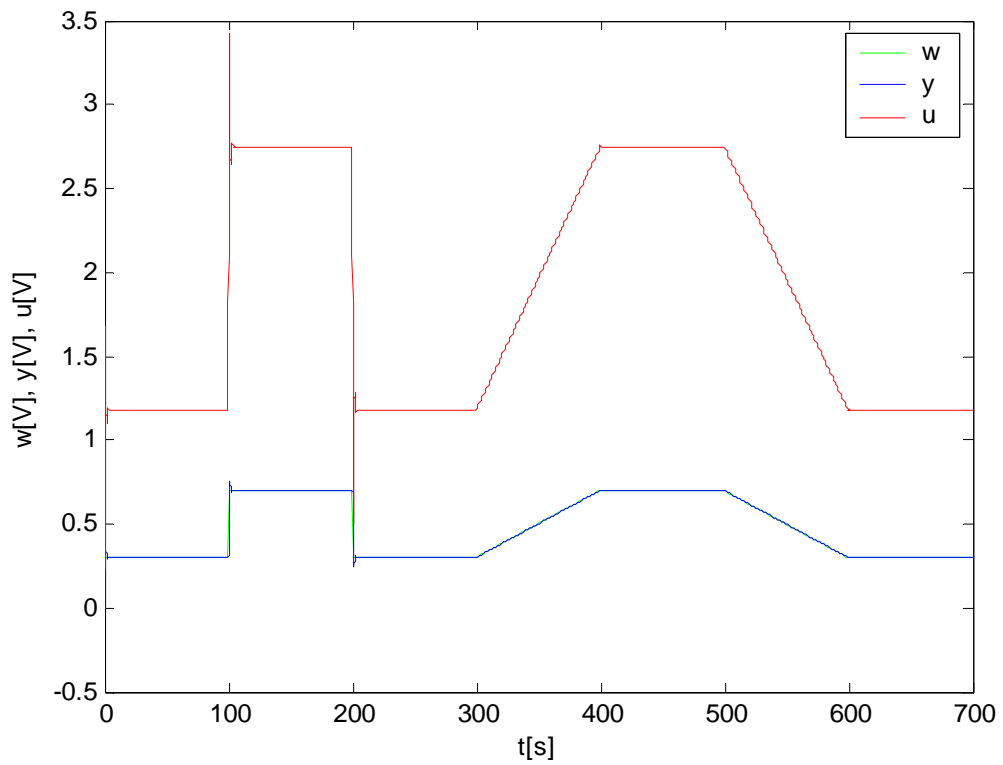
Z jednotlivých obrázků porovnání jednotlivých odezev a z tabulky 9.1 vyplývá, že lepší výsledky identifikace byly dosaženy identifikací druhým řádem, proto pro simulaci prediktivního řízení byly použity jako parametry modelu parametry získané rekurzivní metodou nejmenších čtverců pro druhý řád.



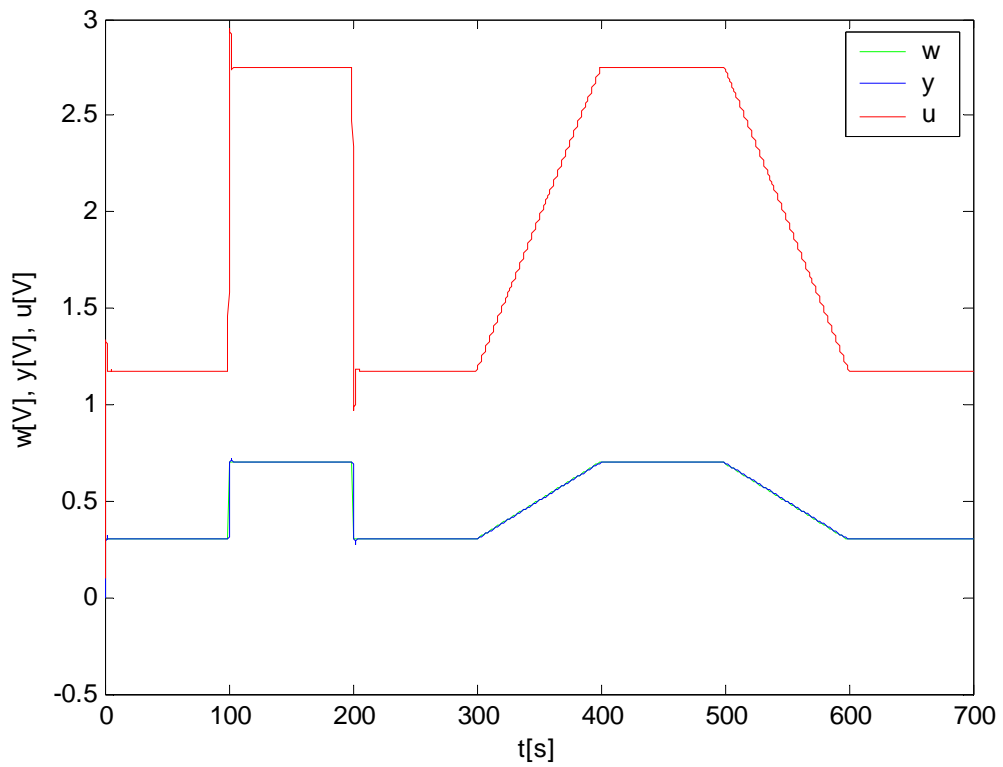
Na následujících obrázcích jsou uvedeny jednotlivé simulace prediktivního řízení bez omezení pro jednotlivé  $\lambda$ :



Obrázek 10.2 Simulace prediktivního řízení bez omezení pro  $\lambda=0.05$



Obrázek 10.3 Simulace prediktivního řízení bez omezení pro  $\lambda=20$



Obrázek 10.4 Simulace prediktivního řízení bez omezení pro  $\lambda=50$

Jednotlivé průběhy byly dále srovnány pomocí kritéria kvadrátu odchylky výstupní veličiny  $y$  od žádané veličiny  $w$  pomocí vztahu  $S_y = \sum_{i=1}^N (w(i) - y(i))^2$ , kde  $N$  je počet

naměřených dat, a pomocí kritéria sumy přírůstků akční veličiny  $\Delta u$  podle vztahu

$S_u = \sum_{i=1}^N \Delta u^2(i) = \sum_{i=1}^N (u(i) - u(i-1))^2$ , kde  $N$  je počet naměřených dat. V následující tabulce

10.1 jsou uvedeny hodnoty tohoto kritéria pro jednotlivé průběhy s různými  $\lambda$ :

Tabulka 10.1 Hodnoty kritéria kvadrátu odchylky  $S_y$  a kritéria sumy přírůstků akční veličiny  $S_u$  pro simulaci prediktivního řízení bez omezení

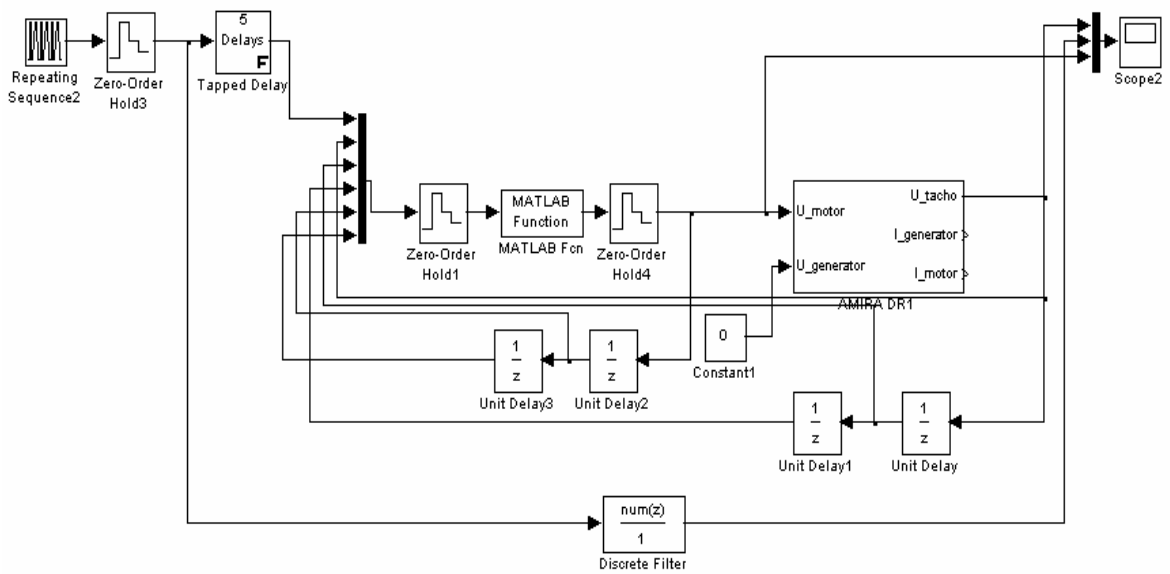
<b>Lambda</b>	<b><math>S_y</math></b>	<b><math>S_u</math></b>
0.05	0.5331	0.6984
20	2.4696	0.0120
50	4.2584	0.0039

Z jednotlivých obrázků a z tabulky 10.1 lze vidět, že čím vyšší je  $\lambda$  tím se snižují přírůstky jednotlivých akčních zásahů, ale výstupní veličina hůře sleduje žádanou hodnotu.

## 10.2 Prediktivní řízení bez omezení na reálném modelu

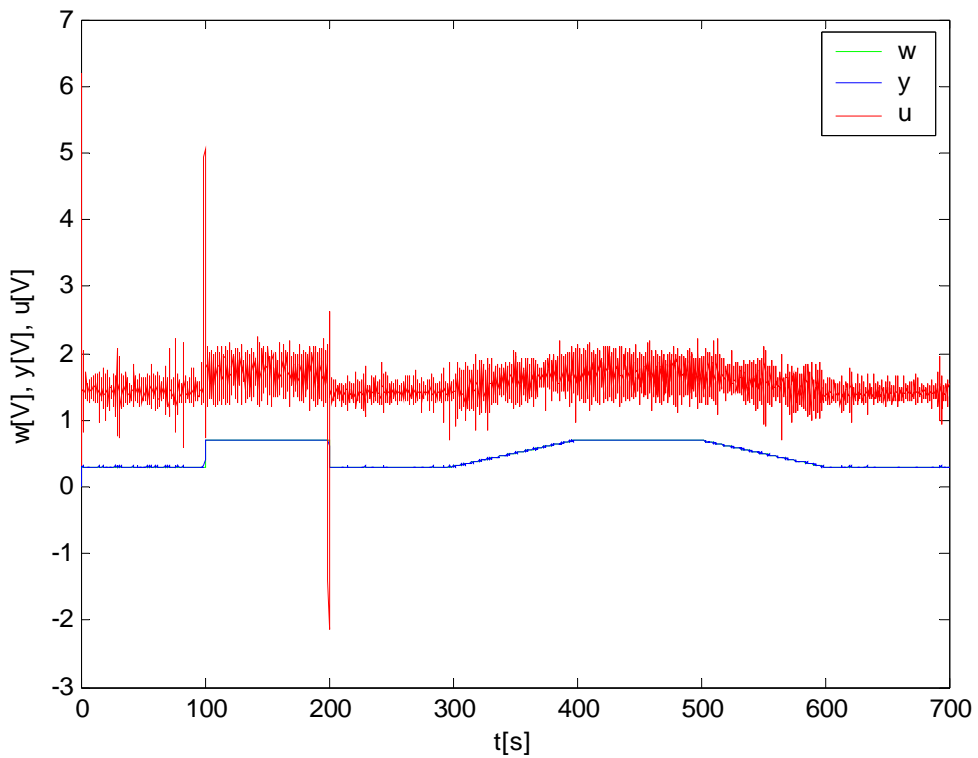
Pro reálné měření je nutné při spuštění postupovat naprosto stejně jako u simulace až do kroku otevření simulačního schématu, které je možné zavřít a je nutné zvlášť otevřít simulinkové schéma pod názvem spojka.mdl, které vypadá následovně:



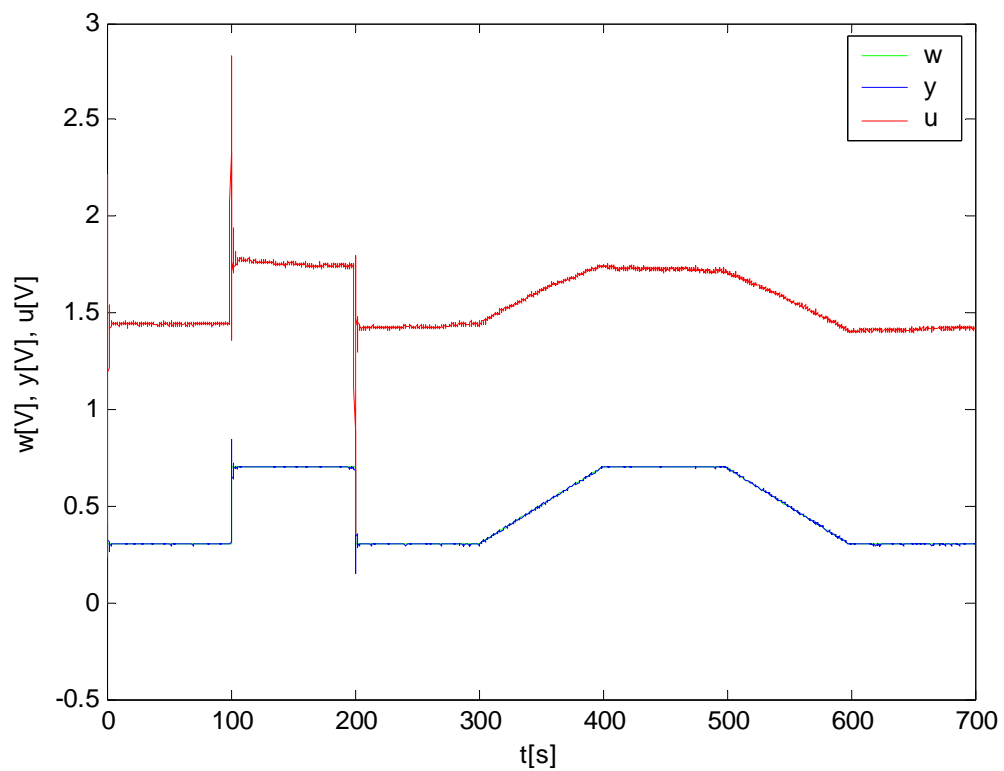
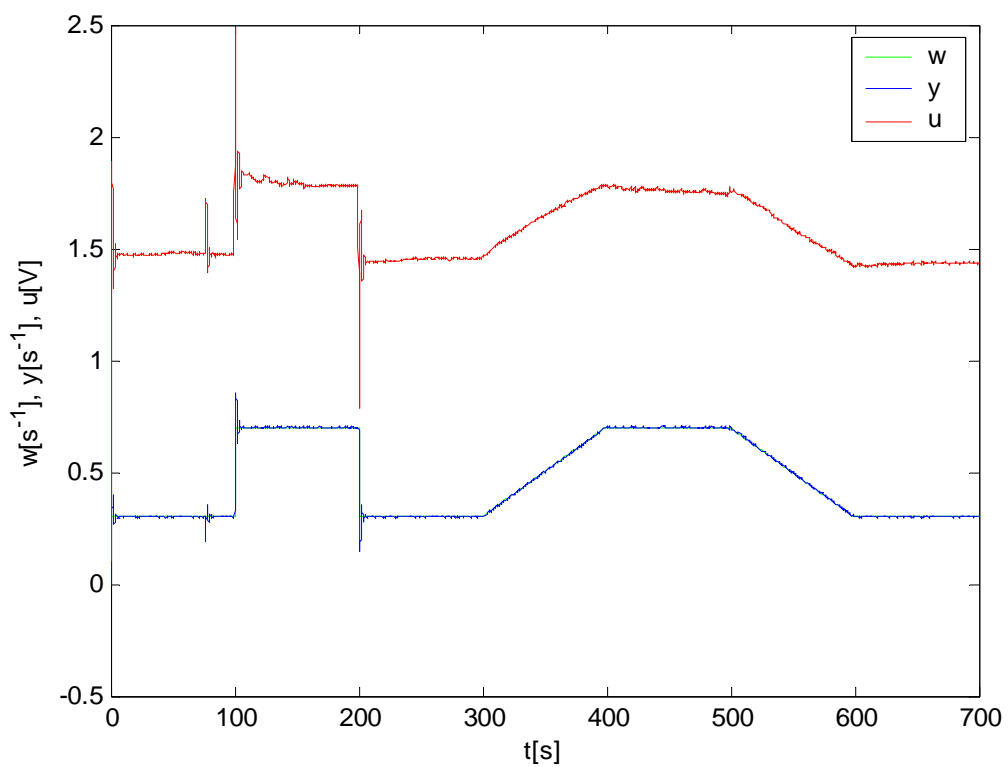


Obrázek 10.5 Simulinkové schéma pro reálné měření prediktivního řízení bez omezení

Na následujících grafech jsou znázorněny jednotlivé časové průběhy prediktivního řízení bez omezení na reálném modelu:



Obrázek 10.6 Reálné měření prediktivního řízení bez omezení pro  $\lambda=0.05$


 Obrázek 10.7 Reálné měření prediktivního řízení bez omezení pro  $\lambda=20$ 

 Obrázek 10.8 Reálné měření prediktivního řízení bez omezení pro  $\lambda=50$

Pro srovnání jednotlivých průběhů je opět použito kritérium kvadrátu odchylky výstupní veličiny od žádané hodnoty a kritérium sumy přírůstků akční veličiny. Vztahy jsou uvedeny v kapitole 10.2. V následující tabulce 10.2 jsou uvedeny hodnoty tohoto kritéria pro jednotlivé průběhy s různými  $\lambda$ :

Tabulka 10.2 Hodnoty kritéria kvadrátu odchylky  $S_y$  a kritéria sumy přírůstků akční veličiny  $S_u$  pro prediktivní řízení bez omezení na reálném modelu

<b>Lambda</b>	<b><math>S_y</math></b>	<b><math>S_u</math></b>
0.05	0.3260	9.6180
20	2.0027	0.0165
50	4.1550	0.0051

Platí zde obdobná věc jako u simulace, že s rostoucí  $\lambda$  se zmenšují přírůstky akčních zásahů, ale zhoršuje se sledování žádané veličiny veličinou výstupní. V porovnání simulace a reálného měření se dospělo k výsledku, že výstupní veličina u reálného měření dokonce lépe sleduje žádanou veličinu než u simulace.

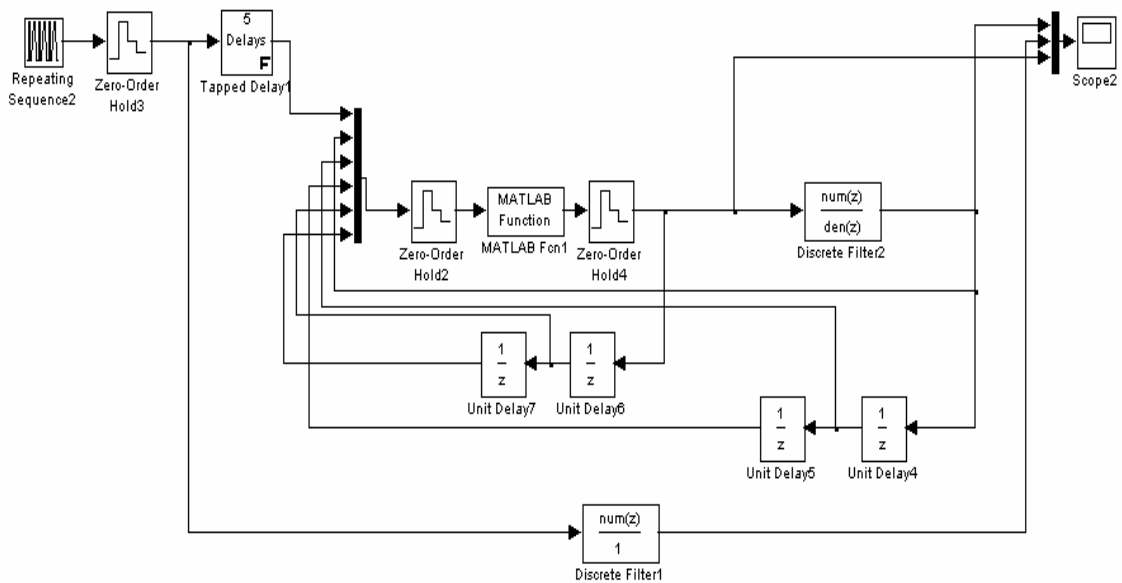
### 10.3 Simulace prediktivního řízení s omezením na modelu získaném identifikací

Jako omezení u laboratorního modelu AMIRA DR300 bylo zvoleno tvrdé omezení akčního zásahu. Spodní a horní hranice tohoto omezení byly vybrány na základě statické charakteristiky tak, aby ležely v horní polovině statické charakteristiky a mimo oblast pásma necitlivosti a pásma saturace. Byly tedy zvoleny následující parametry:  $u_{min} = 1$  V a  $u_{max} = 2$  V.

Pro provedení simulačního měření prediktivního řízení s omezením na matematickém modelu reálné soustavy AMIRA DR300 je nejprve nutné spustit opět m-file identifikace.m, ze kterého se parametry modelu použijí v dalším m-filu pod názvem Prediktivni\_regulator\_s\_omezenim.m, jehož zdrojový kód je zobrazen v příloze P VI, který je potřeba otevřít po sléze. Možnost volby a hodnoty jednotlivých parametrů jsou shodné jako u kapitoly 10.1. Po spuštění tohoto m-filu dojde k otevření simulinkového

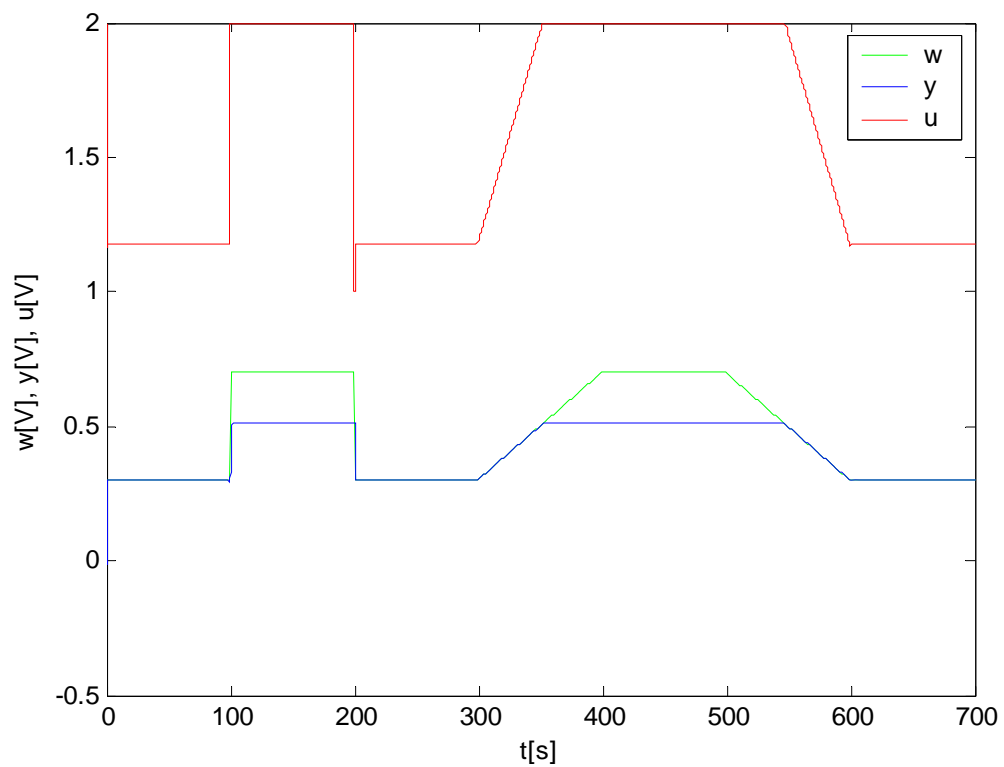
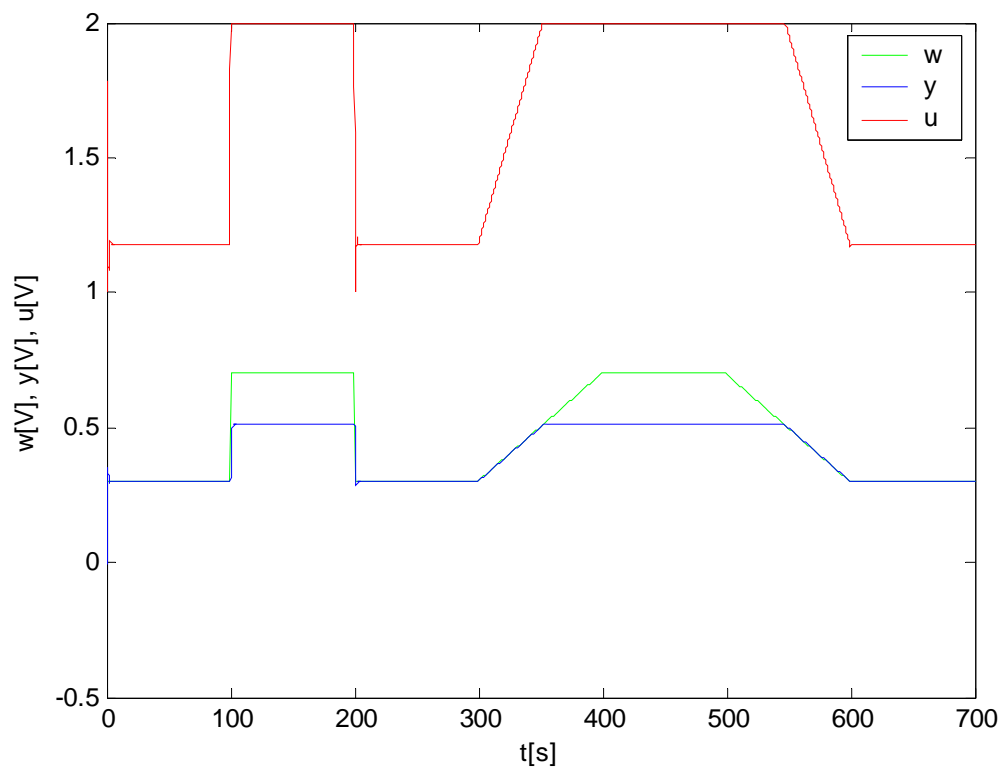
schématu pod názvem Prediktivni\_regulace\_s\_omezenim.mdl. Toto simulinkové schéma se při spuštění ještě odkazuje v každém kroku na m-file regulátor\_s\_omezenim.m, jehož zdrojový kód je zobrazen v příloze P VII, ve kterém je toto tvrdé omezení akčního zásahu řešeno pomocí kvadratického programování pomocí funkce quadprog.

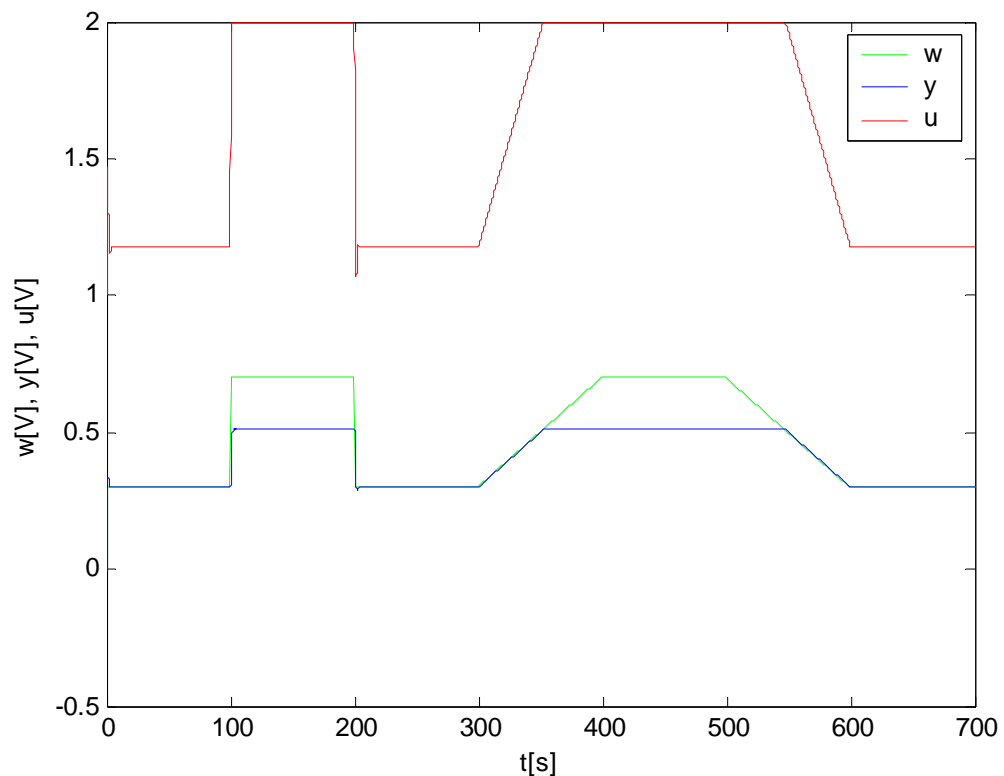
Pro simulaci prediktivního řízení s omezením bylo použito následující simulinkové schéma:



Obrázek 10.9 Simulinkové schéma pro simulaci prediktivního řízení bez omezení

Na následujících obrázcích jsou uvedeny jednotlivé simulace prediktivního řízení s omezením pro jednotlivé  $\lambda$ :

Obrázek 10.10 Simulace prediktivního řízení s omezením pro  $\lambda=0.05$ Obrázek 10.11 Simulace prediktivního řízení s omezením pro  $\lambda=20$

Obrázek 10.12 Simulace prediktivního řízení s omezením pro  $\lambda=50$ 

Pro srovnání jednotlivých průběhů je opět použito kritérium kvadrátu odchylky výstupní veličiny od žádané hodnoty a kritérium sumy přírůstků akční veličiny. Vztahy jsou uvedeny v kapitole 10.2. V následující tabulce 10.3 jsou uvedeny hodnoty tohoto kritéria pro jednotlivé průběhy s různými  $\lambda$ :

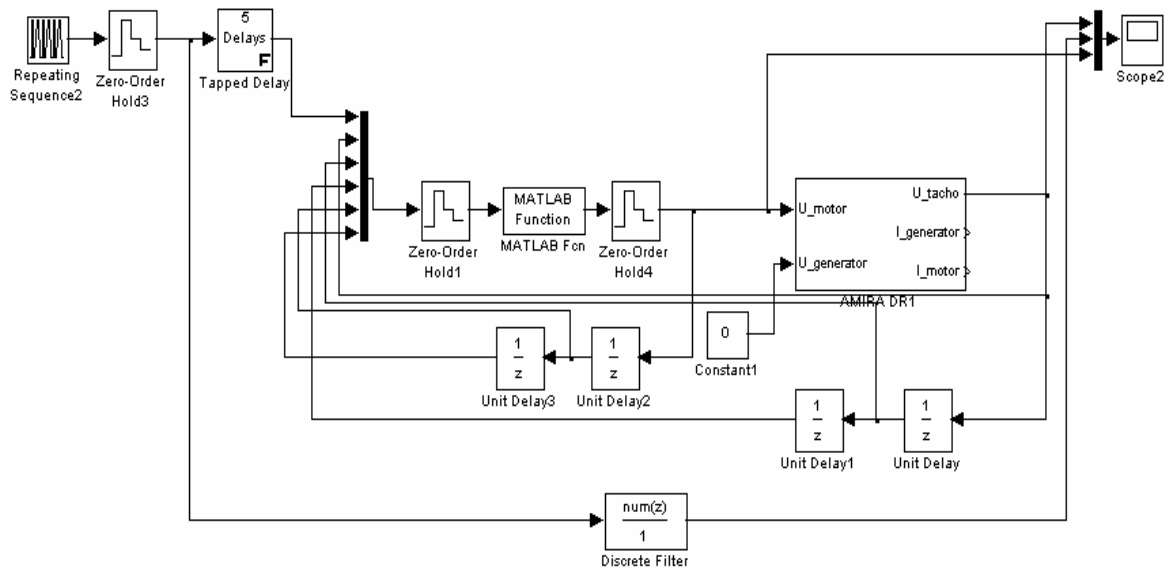
Tabulka 10.3 Hodnoty kritéria kvadrátu odchylky  $S_y$  a kritéria sumy přírůstků akční veličiny  $S_u$  pro simulaci prediktivního řízení s omezením

<b>Lambda</b>	<b><math>S_y</math></b>	<b><math>S_u</math></b>
0.05	169.4531	0.0159
20	169.9015	0.0042
50	170.4071	0.0016

Z jednotlivých obrázků a z tabulky 10.3 lze vidět, že čím vyšší je  $\lambda$  tím se snižují jednotlivé akční zásahy, ale výstupní veličina hůře sleduje žádanou hodnotu.

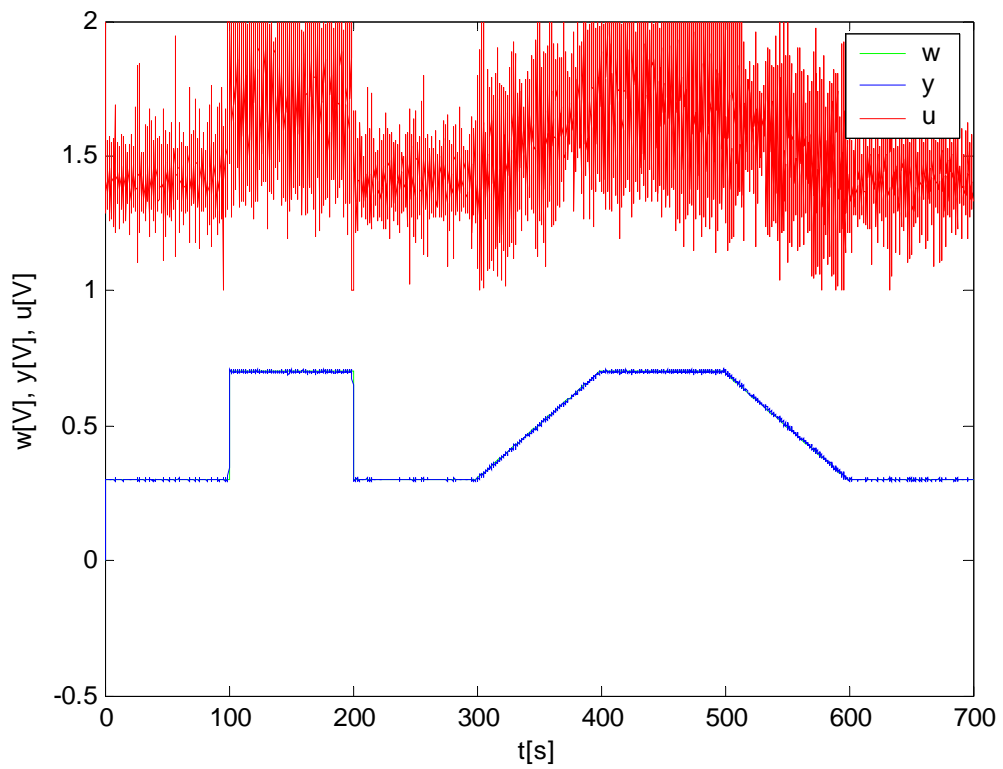
## 10.4 Prediktivní řízení s omezením na reálném modelu

Pro reálné měření je nutné při spuštění postupovat naprosto stejně jako u simulace až do kroku otevření simulačního schématu, které je možné zavřít a je nutné zvlášť otevřít simulinkové schéma pod názvem spojka\_s\_omezenim.mdl, které vypadá následovně:

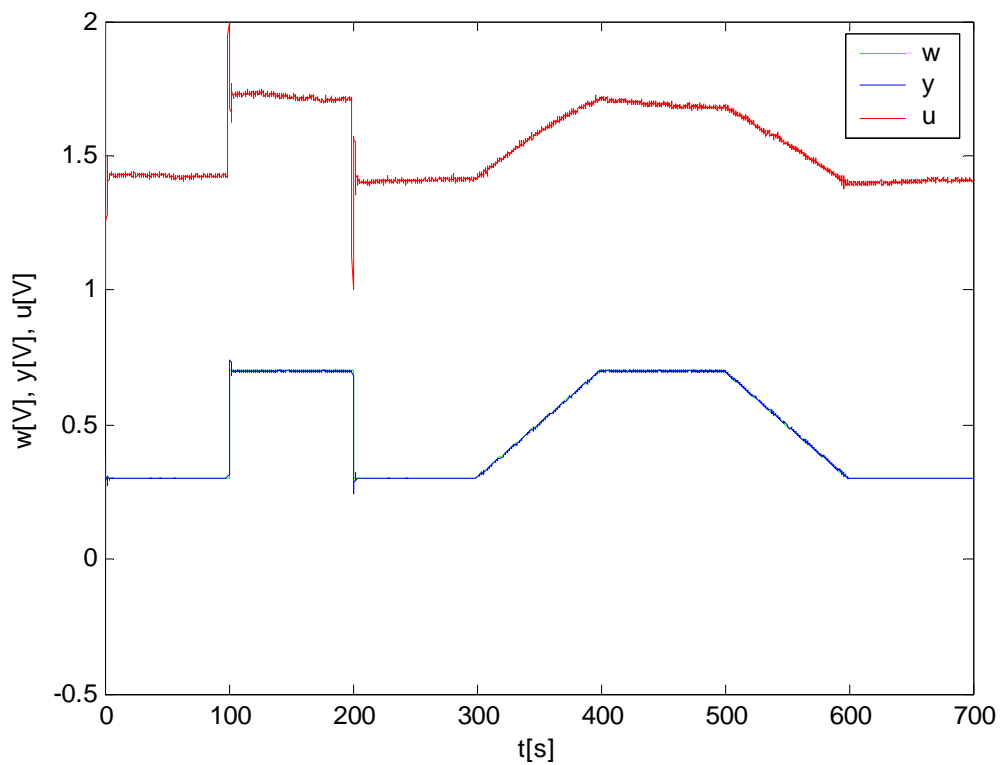


Obrázek 10.13 Simulinkové schéma pro reálné měření prediktivního řízení s omezením

Na následujících grafech jsou znázorněny jednotlivé časové průběhy prediktivního řízení s omezením na reálném modelu:

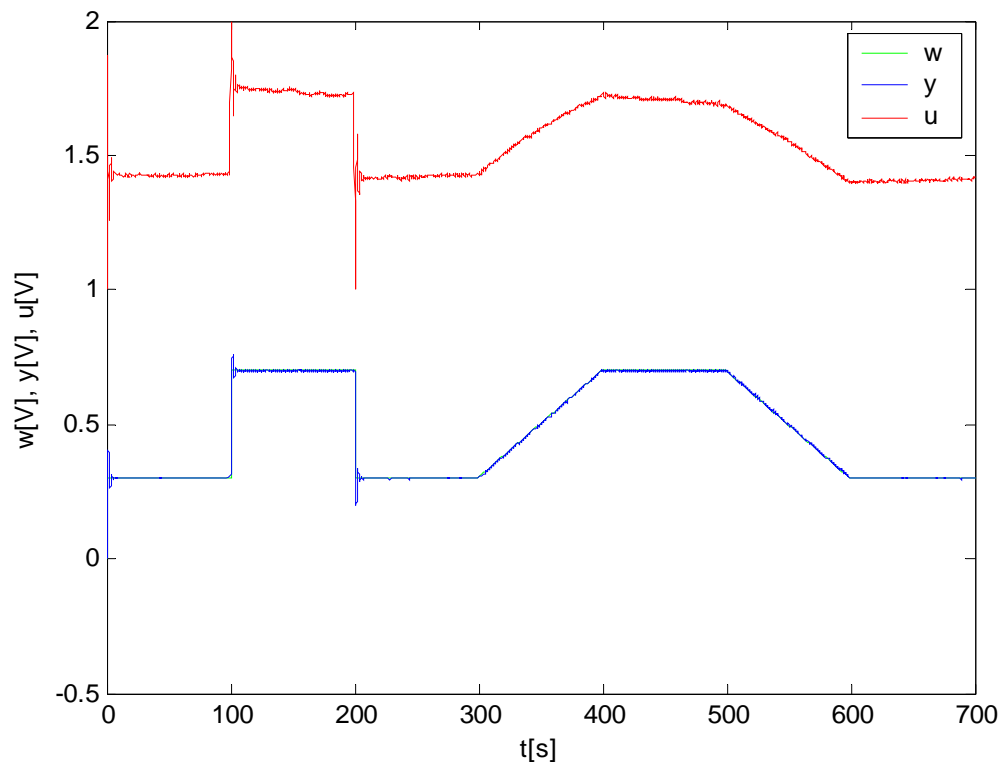


Obrázek 10.14 Reálné měření prediktivního řízení s omezením pro  $\lambda=0.05$



Obrázek 10.15 Reálné měření prediktivního řízení s omezením pro  $\lambda=20$



Obrázek 10.16 Reálné měření prediktivního řízení s omezením pro  $\lambda=50$ 

Pro srovnání jednotlivých průběhů je opět použito kritérium kvadrátu odchylky výstupní veličiny od žádané hodnoty a kritérium sumy přírůstků akční veličiny. Vztahy jsou uvedeny v kapitole 10.2. V následující tabulce 10.4 jsou uvedeny hodnoty tohoto kritéria pro jednotlivé průběhy s různými  $\lambda$ :

Tabulka 10.4 Hodnoty kritéria kvadrátu odchylky  $S_y$  a kritéria sumy přírůstků akční veličiny  $S_u$  pro prediktivní řízení s omezením na reálném modelu

<b>Lambda</b>	<b><math>S_y</math></b>	<b><math>S_u</math></b>
0.05	1.4842	8.3558
20	1.9728	0.0078
50	2.9557	0.0028

Platí zde obdobná věc jako u simulace, že s rostoucí  $\lambda$  se zmenšují akční zásahy ale zhoršuje se sledování žádané veličiny veličinou výstupní. V porovnání simulace a reálného měření se dospělo k výsledku, že výstupní veličina u reálného měření dokonce lépe sleduje žádanou veličinu než u simulace.

## ZÁVĚR

Tato práce se zabývá problematikou týkající se prediktivního řízení bez omezení i s omezením vstupních a výstupních veličin.

Jako součást této práce byly vytvořeny programy a simulační schémata pro simulaci a řízení reálného modelu pomocí prediktivního řízení bez omezení vstupních a výstupních veličin a s omezením akčního zásahu v programovém prostředí Matlab a Matlab/Simulink.

Pro otestování těchto algoritmů při reálném řízení byl vybrán laboratorní model AMIRA DR300. Z několika naměřených přechodových charakteristik byla získána výsledná přechodová charakteristika, z které byly určeny jednotlivé horizonty řízení pro prediktivní řízení tohoto modelu. Dále bylo potřeba získat diskrétní přenos reálného modelu pro použití algoritmů prediktivního řízení. Na základě identifikačních experimentů byl vybrán diskrétní model druhého řádu. Z průběhu experimentálně získané statické charakteristiky byly zvoleny horní a dolní hranice omezení akční veličiny. Získaný matematický model byl simulačně ověřen při prediktivním řízení bez i s omezením akční veličiny pro tři různé hodnoty parametru  $\lambda$ . Za stejných podmínek bylo provedeno řízení laboratorního modelu v reálném čase. Z regulačních průběhů vyplývá, že s rostoucím parametrem  $\lambda$  se zmenšují přírůstky akční veličiny, ale naopak se zhoršuje sledování žádané veličiny regulovanou veličinou. Z porovnání řízení v simulačních a reálných podmínkách vyplývají rozdíly v ustálených hodnotách akční veličiny, což vede pro zvolené omezení akční veličiny u simulace k trvalé regulační odchylce. Tento nesoulad je možný nelineárním chováním laboratorního modelu.

## ZÁVĚR V ANGLIČTINĚ

This thesis deals with problems of constrained and unconstrained predictive control. Programs and simulation diagrams for simulation and real time control of a laboratory model by algorithms of predictive control were created in the Matlab/ Simulink environment. The algorithms of both constrained and unconstrained predictive control were implemented. For real time verification of the algorithms was chosen the model AMIRA DR300. Step response of the system was measured. On the basis of the measured step response were obtained prediction and control horizons. Further it was necessary to obtain a discrete model of the real system in the form of transfer function. On the basis of identification experiments was chosen the discrete model of second order. On the basis of the experimentally measured static characteristics were chosen maximum and minimum values of the input signal. The obtained mathematical model was verified by simulation for three values of the parameter  $\lambda$ . Upon the same conditions was performed real time control of the laboratory model. From the control results it was evident, that increasing  $\lambda$  causes attenuation of increments of the manipulated variable. On the other hand asymptotic tracking of the reference signal got worsen with increasing  $\lambda$ . From comparison of the real time and simulation results ensues discrepancy between steady values of manipulated variable. This could be caused by nonlinear behaviour of the laboratory model.

**SEZNAM POUŽITÉ LITERATURY**

- [1] Maciejowski, J.M.: Predictive Control with Constraints, Prentice Hall, 2002.
- [2] Camacho, E.F. and C. Bordons: Model Predictive Control, Springer-Verlag London, 1999.
- [3] Camacho, E.F. and C. Bordons: Model Predictive Control, Second Edition, Springer-Verlag London, 2004.
- [4] Kanjilal, P. P.: Adaptive Prediction and Predictive Control, IEE Control Engineering Series, 1995.
- [5] Allgöwer, F. and Zheng, A. (2000). Nonlinear Model Predictive Control. Birkhäuser Verlag AG.
- [6] Clarke, D. W., Mohtadi, C. and Tuffs, P. S. (1987a). ‘Generalized predictive control – I. the basic algorithm’. Automatica 23, 137-148.
- [7] Clarke, D. W., Mohtadi, C. and Tuffs, P. S. (1987b). ‘Generalized predictive control – II. extensions and interpolations’. Automatica 23, 149-160.
- [8] Mikleš, J., Fikar, M.: Modelovanie, identifikácia a riadenie procesov 2. Identifikácia a optimálne riadenie. Slovenská technická univerzita v Bratislavě, 2004.
- [9] Bobál, V., Böhm, J., Prokop, R., Fessl, J. (1999). Praktické aspekty samočinně se nastavujících regulátorů: algoritmy a implementace. Nakladatelství VUTIUM, Vysoké učení technické v Brně.
- [10] Navrátil, P. *Informační systém CAAC – Počítačová podpora automatického řízení*, Disertační práce, UTB – FT ve Zlíně, 2004.
- [11] Nguyen, T.C.P. *Počítačová podpora automatického řízení CAAC*, Disertační práce, VUT – FSI v Brně, 2001.
- [12] Giesl, P.: Prediktivní řízení laboratorních modelů. Diplomová práce, UTB ve Zlíně, FAI, 2006.
- [13] Kalman, E., Bucy, RC.: New results in linear filtering and prediction theory. J. Basic Engr., **83**, 95-100, 1961.

- [14] Ziegler, J.G. and N. B. Nichols. Optimum settings for automatic controllers. Trans. ASME, **64**, 759-768, 1942.
- [15] Rositer, J. A. Model Based Predictive Control. A Practical Approach. CRC Press London, 2003.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

GPC	Generalized Predictive Control.
MPC	Model Predictive Control.
DMC	Dynamic Matrix Control.
QDMC	Quadratic Dynamic Matrix Control.
SMOC	Shell Multivariable Optimizing Controller.
$y(k)$	Výstupní veličina v kroku $k$ .
$w(k)$	Žádaná veličina v kroku $k$ .
$u(k)$	Akční veličina v kroku $k$ .
$e(k)$	Regulační odchylka v kroku $k$ .
$N$	Počet naměřených dat.
$\hat{y}$	Predikovaná hodnota výstupu.
$k$	Časový okamžik.
$\delta(t)$	Jednotkový (Diracův) impuls.
$z^{-1}$	Operátor zpětného posuvu.
$h_i$	Posloupnost impulsní odezvy.
$p$	Počet vstupů.
$\eta(t)$	Jednotkový (Heavisideův) skok.
$g_i$	Koeficienty přechodové posloupnosti.
$\Delta u(k)$	Přírůstek akční veličiny v kroku $k$ .
$e_s(k)$	Hodnota bílého šumu v kroku $k$ .
$B(z^{-1})$	Polynom čitatele přenosu systému.
$A(z^{-1})$	Polynom jmenovatele přenosu systému.
$G(z^{-1})$	Diskrétní přenos systému.

$T$	Perioda vzorkování.
$v(k)$	Porucha systému v kroku $k$ .
SISO	Single Input Single Output.
MIMO	Multi Input Multi Output.
$J$	Účelová funkce.
$N_1$	Minimální horizont.
$N_2$	Maximální horizont.
$N_U$	Řídicí horizont.
$y_0$	Volná odezva systému.
$\mathbf{G}$	Jacobián model.
$u_{min}$	Spodní hranice akčního zásahu.
$u_{max}$	Horní hranice akčního zásahu.
$T_d$	Dopravní zpoždění.
$\mathbf{g}$	Gradient.
$\mathbf{H}$	Hessova matice.
$\boldsymbol{\varphi}$	Vektor dat.
$\boldsymbol{\Theta}$	Vektor parametrů soustavy.
$\hat{\boldsymbol{\Theta}}$	Vektor odhadů parametrů soustavy.
$\Delta t$	Interval hodinových impulsů.
$n$	Počet stupňů posuvného registru.
$\lambda$	Penalizační faktor.

## SEZNAM OBRÁZKŮ

Obrázek 3.1 Základní struktura systému prediktivního řízení.....	12
Obrázek 3.2 Princip prediktivního řízení.....	12
Obrázek 4.1 Jednotkový (Diracův) impuls .....	14
Obrázek 4.2 Jednotkový (Heavisideův) skok .....	16
Obrázek 6.1 Postup při rekurzivní metodě nejmenších čtverců .....	41
Obrázek 7.1 Laboratorní model AMIRA DR300 .....	43
Obrázek 7.2 Schéma modelu .....	44
Obrázek 8.1 Simulinkové schéma pro měření statické charakteristiky .....	46
Obrázek 8.2 Časový průběh měření statické charakteristiky.....	47
Obrázek 8.3 Výsledná statická charakteristika .....	47
Obrázek 8.4 Přechodové charakteristiky pro skoky žádané veličiny z 1.7V na 1.75V .....	48
Obrázek 8.5 Přechodové charakteristiky pro skoky žádané veličiny z 1.2V na 1.25V .....	49
Obrázek 8.6 Přechodové charakteristiky pro skoky žádané veličiny z 1.4V na 1.45V .....	50
Obrázek 8.7 Přechodová charakteristika s počátkem v čase 160 s.....	51
Obrázek 8.8 Přechodová charakteristika s počátkem v čase 480 s.....	51
Obrázek 8.9 Přechodová charakteristika s počátkem v čase 800s.....	52
Obrázek 8.10 Výsledná přechodová charakteristika .....	53
Obrázek 9.1 Odezva na pseudonáhodný binární signál.....	55
Obrázek 9.2 Porovnání odezev reálné soustavy a identifikovaného modelu prvního řádu .....	57
Obrázek 9.3 Porovnání odezev reálné soustavy a identifikovaného modelu druhého řádu .....	58
Obrázek 10.1 Simulinkové schéma pro simulaci prediktivního řízení bez omezení.....	60
Obrázek 10.2 Simulace prediktivního řízení bez omezení pro $\lambda=0.05$ .....	61
Obrázek 10.3 Simulace prediktivního řízení bez omezení pro $\lambda=20$ .....	62
Obrázek 10.4 Simulace prediktivního řízení bez omezení pro $\lambda=50$ .....	62
Obrázek 10.5 Simulinkové schéma pro reálné měření prediktivního řízení bez omezení.....	64
Obrázek 10.6 Reálné měření prediktivního řízení bez omezení pro $\lambda=0.05$ .....	64
Obrázek 10.7 Reálné měření prediktivního řízení bez omezení pro $\lambda=20$ .....	65
Obrázek 10.8 Reálné měření prediktivního řízení bez omezení pro $\lambda=50$ .....	65



---

Obrázek 10.9 Simulinkové schéma pro simulaci prediktivního řízení bez omezení.....	67
Obrázek 10.10 Simulace prediktivního řízení s omezením pro $\lambda=0.05$ .....	68
Obrázek 10.11 Simulace prediktivního řízení s omezením pro $\lambda=20$ .....	68
Obrázek 10.12 Simulace prediktivního řízení s omezením pro $\lambda=50$ .....	69
Obrázek 10.13 Simulinkové schéma pro reálné měření prediktivního řízení s omezením .....	70
Obrázek 10.14 Reálné měření prediktivního řízení s omezením pro $\lambda=0.05$ .....	71
Obrázek 10.15 Reálné měření prediktivního řízení s omezením pro $\lambda=20$ .....	71
Obrázek 10.16 Reálné měření prediktivního řízení s omezením pro $\lambda=50$ .....	72

**SEZNAM TABULEK**

Tabulka 9.1 Hodnoty kritéria kvadrátu odchyly $S_y$ a kvality identifikace $S_{ID}$ .....	58
Tabulka 10.1 Hodnoty kritéria kvadrátu odchyly $S_y$ a kritéria sumy přírůstků akční veličiny $S_u$ pro simulaci prediktivního řízení bez omezení.....	63
Tabulka 10.2 Hodnoty kritéria kvadrátu odchyly $S_y$ a kritéria sumy přírůstků akční veličiny $S_u$ pro prediktivní řízení bez omezení na reálném modelu.....	66
Tabulka 10.3 Hodnoty kritéria kvadrátu odchyly $S_y$ a kritéria sumy přírůstků akční veličiny $S_u$ pro simulaci prediktivního řízení s omezením.....	69
Tabulka 10.4 Hodnoty kritéria kvadrátu odchyly $S_y$ a kritéria sumy přírůstků akční veličiny $S_u$ pro prediktivní řízení s omezením na reálném modelu.....	72

**SEZNAM PŘÍLOH**

- P I        OBSAH PŘILOŽENÉHO CD-ROMU
- P II        ZDROJOVÝ KÓD M-FILU PNBS.M
- P III       ZDROJOVÝ KÓD M-FILU IDENTIFIKACE.M
- P IV       ZDROJOVÝ KÓD M-FILU PREDIKTIVNI\_REGULATOR.M
- P V        ZDROJOVÝ KÓD M-FILU REGULATOR.M
- P VI       ZDROJOVÝ KÓD M-FILU  
             PREDIKTIVNI\_REGULATOR\_S\_OMEZENIM.M
- P VII      ZDROJOVÝ KÓD M-FILU REGULATOR\_S\_OMEZENIM.

## **PŘÍLOHA P I: OBSAH PŘILOŽENÉHO CD-ROMU**

Na zadní straně desek je vložen 1 CD-ROM, který obsahuje dva adresáře s níže uvedeným obsahem:

- adresář „Data“
  - obsahuje všechny naměřená data na laboratorním modelu AMIRA DR300.
- adresář „Programy“
  - obsahuje soubory (m-soubory) použité pro identifikaci, simulace a reálné měření prediktivního řízení, které byly vytvořeny v programovém prostředí Matlab,
  - dále obsahuje veškeré simulinkové schémata použité pro simulaci i reálné měření prediktivního řízení, které byly vytvořeny v programovém prostředí Matlab/Simulink.

## PŘÍLOHA P II: ZDROJOVÝ KÓD M-FILU PNBS.M

```
clear all;

clc;

% generovani pseudonahodneho signalu

n=6;

R(n,1)=0;

N=(2^n)-1;

for i=1:n

    R(i)=round(rand);

end;

for k=1:20*N

    RP=xor(R(n),R(n-1));

    for i=n:-1:2

        R(i)=R(i-1);

    end

    R(1)=RP;

    u(k)=0.14+0.04*R(n);

    pnbs(k,1)=0.05*k;

    pnbs(k,2)=u(k);

end;

% vykreslení PNBS

plot(pnbs(:,1),pnbs(:,2));

m=1;

for j=1:length(pnbs)

    PNBS(m)=pnbs(j,2);

    PNBS(m+1)=pnbs(j,2);
```

```
m=m+2;
end;
t(1)=pnbs(1,1);
m=2;
for j=2:length(pnbs)
    t(m)=pnbs(j,1);
    t(m+1)=pnbs(j,1);
    m=m+2;
end;
t(m)=pnbs(length(pnbs),1)+0.01;
```

## PŘÍLOHA P III: ZDROJOVÝ KÓD M-FILU IDENTIFIKACE.M

```
clear all;

close all;

clc;

%načtení dat

data=csvread('pnbs_T_0,05.dat');

%určení pásma necitlivosti

pasma_necitlivosti=0;

%nejmenší čtverce pro první řád

    %naplnění vektoru y

    for i=2:length(data)

        y(i-1)=data(i,3);

    end;

%naplnění matice F

n=1;

for i=1:length(data)-1

    F(i,1)=-data(n,3);

    F(i,2)=data(n,2)-pasma_necitlivosti;

    n=n+1;

end;

%výpočet parametrů soustavy

theta_mnc_1=(F'*F)^(-1)*F'*y'

%nejmenší čtverce pro druhý řád

clear y F

%naplnění vektoru y
```

```

for i=3:length(data)
    y(i-2)=data(i,3);
end;

%naplnění matice F
n=2;
for i=1:length(data)-2
    F(i,1)=-data(n,3);
    F(i,2)=-data(n-1,3);
    F(i,3)=data(n,2)-pasma_necitlivosti;
    F(i,4)=data(n-1,2)-pasma_necitlivosti;
    n=n+1;
end;

%výpočet parametrů soustavy
theta_mnc_2=(F'*F)^(-1)*F'*y'
clear y
y=data(:,3);
u=data(:,2)-pasma_necitlivosti;
%pocatecni hodnoty vektoru fi
fi(:,1)=[0;0];
%pocatecni odhad parametru
theta(:,1)=[0.1;0.2];
%pocatecni hodnota matice C
C1=zeros(2,2);
for i=1:length(C1)
    C1(i,i)=10^8;
end;

```



```

%vypocet parametru
for i=2:length(y)
    fi(:,i)=[-y(i-1);u(i-1)];
    e(i)=y(i)-fi(:,i)*theta(:,i-1);
    L=(C1*fi(:,i))/(1+fi(:,i)'*C1*fi(:,i));
    theta(:,i)=theta(:,i-1)+L*e(i);
    C2=C1-L*fi(:,i)'*C1;
    C1=C2;
end;
theta_rmnc_1=theta(:,i)
for i=2:length(data)
    y_novy_1(i)=-data(i-1,3)*theta_rmnc_1(1)+...
    (data(i-1,2)-pasmo_necitlivosti)*theta_rmnc_1(2);
end;
Sy_1=0;
SID_1=0;
for i=1:length(data)
    Sy_1=Sy_1+(data(i,3)-y_novy_1(i))^2;
    SID_1=SID_1+e(i)^2;
end;
Sy_1
SID_1=SID_1/length(data)
%rekurzivní nejmenší čtverce pro druhý řád
clear theta e L C1 C2 fi
%pocatecni hodnoty vektoru fi
fi(:,1)=[0;0;0;0];

```

```

fi(:,2)=[-y(1);0;u(1);0];

%pocatecni odhad parametru

theta(:,1)=[0.1;0.2;0.3;0.4];

theta(:,2)=[0.1;0.2;0.3;0.4];

%pocatecni hodnota matice C

C1=zeros(4,4);

for i=1:length(C1)

    C1(i,i)=10^8;

end;

%vypocet parametru

for i=3:length(y)

    fi(:,i)=[-y(i-1);-y(i-2);u(i-1);u(i-2)];

    e(i)=y(i)-fi(:,i)*theta(:,i-1);

    L=(C1*fi(:,i))/(1+fi(:,i)*C1*fi(:,i));

    theta(:,i)=theta(:,i-1)+L*e(i);

    C2=C1-L*fi(:,i)*C1;

    C1=C2;

end;

theta_rmnc_2=theta(:,i)

for i=3:length(data)

    y_novy_2(i)=-data(i-1,3)*theta_rmnc_2(1)-data(i-2,3)*theta_rmnc_2(2)+...

    (data(i-1,2)-pasmu_necitlivosti)*theta_rmnc_2(3)+...

    (data(i-2,2)-pasmu_necitlivosti)*theta_rmnc_2(4);

end;

Sy_2=0;

SID_2=0;

```

```

for i=1:length(data)
    Sy_2=Sy_2+(data(i,3)-y_novy_2(i))^2;
    SID_2=SID_2+e(i)^2;
end;
Sy_2
SID_2=SID_2/length(data)
plot(data(:,1),data(:,2)*10,'g')
hold on
plot(data(:,1),data(:,3))
plot(data(:,1),y_novy_1,'r')
xlabel('t[s]')
ylabel('w[V], y[V]')
legend('w','odezva na realne soustave','odezva na identifikovane soustave')
figure
plot(data(:,1),data(:,2)*10,'g')
hold on
plot(data(:,1),data(:,3))
plot(data(:,1),y_novy_2,'r')
xlabel('t[s]')
ylabel('w[V], y[V]')
legend('w','odezva na realne soustave','odezva na identifikovane soustave')

```

## PŘÍLOHA P IV: ZDROJOVÝ KÓD M-FILU PREDIKTIVNI\_REGULATOR.M

```
%určení parametrů soustavy

a0=1;

a1=theta_rmnc_2(1);

a2=theta_rmnc_2(2);

b0=0;

b1=theta_rmnc_2(3);

b2=theta_rmnc_2(4);

%určení periody vzorkování

T=0.05;

% určení horizontů

N1=1;

N2=5;

Nu=2;

% vytvoření matice y0

y0=zeros(N2,4);

y0(1,1)=(a0-a1)/a0;

y0(1,2)=(a1-a2)/a0;

y0(1,3)=a2/a0;

y0(1,4)=b2;

y0(2,1)=y0(1,1)*y0(1,1)+y0(1,2);

y0(2,2)=y0(1,2)*y0(1,1)+y0(1,3);

y0(2,3)=y0(1,3)*y0(1,1);

y0(2,4)=y0(1,4)*y0(1,1);

y0(3,1)=y0(2,1)*y0(1,1)+y0(1,1)*y0(1,2)+y0(1,3);
```

```

y0(3,2)=y0(2,2)*y0(1,1)+y0(1,2)*y0(1,2);
y0(3,3)=y0(2,3)*y0(1,1)+y0(1,3)*y0(1,2);
y0(3,4)=y0(2,4)*y0(1,1)+y0(1,4)*y0(1,2);
for i=4:N2
    y0(i,1)=y0(i-1,1)*y0(1,1)+y0(i-2,1)*y0(1,2)+y0(i-3,1)*y0(1,3);
    y0(i,2)=y0(i-1,2)*y0(1,1)+y0(i-2,2)*y0(1,2)+y0(i-3,2)*y0(1,3);
    y0(i,3)=y0(i-1,3)*y0(1,1)+y0(i-2,3)*y0(1,2)+y0(i-3,3)*y0(1,3);
    y0(i,4)=y0(i-1,4)*y0(1,1)+y0(i-2,4)*y0(1,2)+y0(i-3,4)*y0(1,3);
end;
% vytvoření matice G
G=zeros(N2,Nu);
G(1,1)=b1;
G(1,2)=b0;
G(2,1)=G(1,1)*y0(1,1)+y0(1,4);
G(3,1)=G(2,1)*y0(1,1)+G(1,1)*y0(1,2);
for i=4:N2
    G(i,1)=G(i-1,1)*y0(1,1)+G(i-2,1)*y0(1,2)+G(i-3,1)*y0(1,3);
end;
for i=2:N2
    for j=2:Nu
        G(i,j)=G(i-1,j-1);
    end;
end;
% určení koeficientu lambda a vytvoření jednotkové matice
lambda=50;
I=zeros(Nu,Nu);

```

```
for i=1:Nu
    I(i,i)=1;
end;
% výpočet matice H
H=G'*G+lambd*I;
%výpočet matice K
K=inv(H)*G';
%výpočet proměnné pomocná
pomocna=K*y0;
open('Prediktivni_regulace.mdl');
```

## PŘÍLOHA P V: ZDROJOVÝ KÓD M-FILU REGULATOR.M

```
function u=regulator(data, N2, pomocna, K)

%data(1:N2)=w(k+1:N2)

%data(N2+1)=y(k)

%data(N2+2)=y(k-1)

%data(N2+3)=y(k-2)

%data(N2+4)=u(k-1)

%data(N2+5)=u(k-2)

du=0;

for j=1:N2

    du=du+K(1,j)*data(j);

end;

for i=1:3

    du=du-pomocna(1,i)*data(N2+i);

end;

i=4;

du=du-pomocna(1,i)*(data(N2+i)-data(N2+i+1));

u=data(N2+4)+du;
```

## PŘÍLOHA P VI: ZDROJOVÝ KÓD M-FILU PREDIKTIVNI\_REGULATOR\_S\_OMEZENIM.M

```
%určení parametrů soustavy  
  
a0=1;  
  
a1=theta_rmnc_2(1);  
  
a2=theta_rmnc_2(2);  
  
b0=0;  
  
b1=theta_rmnc_2(3);  
  
b2=theta_rmnc_2(4);  
  
%určení periody vzorkování  
  
T=0.05;  
  
% určení horizontů  
  
N1=1;  
  
N2=5;  
  
Nu=2;  
  
% vytvoření matice y0  
  
y0=zeros(N2,4);  
  
y0(1,1)=(a0-a1)/a0;  
  
y0(1,2)=(a1-a2)/a0;  
  
y0(1,3)=a2/a0;  
  
y0(1,4)=b2;  
  
y0(2,1)=y0(1,1)*y0(1,1)+y0(1,2);  
  
y0(2,2)=y0(1,2)*y0(1,1)+y0(1,3);  
  
y0(2,3)=y0(1,3)*y0(1,1);  
  
y0(2,4)=y0(1,4)*y0(1,1);  
  
y0(3,1)=y0(2,1)*y0(1,1)+y0(1,1)*y0(1,2)+y0(1,3);
```



```

y0(3,2)=y0(2,2)*y0(1,1)+y0(1,2)*y0(1,2);
y0(3,3)=y0(2,3)*y0(1,1)+y0(1,3)*y0(1,2);
y0(3,4)=y0(2,4)*y0(1,1)+y0(1,4)*y0(1,2);
for i=4:N2
    y0(i,1)=y0(i-1,1)*y0(1,1)+y0(i-2,1)*y0(1,2)+y0(i-3,1)*y0(1,3);
    y0(i,2)=y0(i-1,2)*y0(1,1)+y0(i-2,2)*y0(1,2)+y0(i-3,2)*y0(1,3);
    y0(i,3)=y0(i-1,3)*y0(1,1)+y0(i-2,3)*y0(1,2)+y0(i-3,3)*y0(1,3);
    y0(i,4)=y0(i-1,4)*y0(1,1)+y0(i-2,4)*y0(1,2)+y0(i-3,4)*y0(1,3);
end;
% vytvoření matice G
G=zeros(N2,Nu);
G(1,1)=b1;
G(1,2)=b0;
G(2,1)=G(1,1)*y0(1,1)+y0(1,4);
G(3,1)=G(2,1)*y0(1,1)+G(1,1)*y0(1,2);
for i=4:N2
    G(i,1)=G(i-1,1)*y0(1,1)+G(i-2,1)*y0(1,2)+G(i-3,1)*y0(1,3);
end;
for i=2:N2
    for j=2:Nu
        G(i,j)=G(i-1,j-1);
    end;
end;
% určení koeficientu lambda a vytvoření jednotkové matice
lambda=50;
I=zeros(Nu,Nu);

```

```
for i=1:Nu
    I(i,i)=1;
end;
% výpočet matice H
H=G'*G+lambda*I;
%výpočet matice K
K=inv(H)*G';
%výpočet proměnné pomocná
pomocna=K*y0;
open('Prediktivni_regulace_s_omezenim.mdl');
```

## PŘÍLOHA P VII: ZDROJOVÝ KÓD M-FILU REGULATOR\_S\_OMEZENIM.M

```
function u=regulator_s_omezenim(data, N2, Nu, H, G, y0)

umin=0.1;

umax=0.2;

%data(1:N2)=w(k+1:N2)

%data(N2+1)=y(k)

%data(N2+2)=y(k-1)

%data(N2+3)=y(k-2)

%data(N2+4)=u(k-1)

%data(N2+5)=u(k-2)

%nastavení pro kvadratické programování

options = optimset('LargeScale','off','Display','off');

%vytvoření matice k

k=G'*y0*[data(N2+1);data(N2+2);data(N2+3);data(N2+4)-data(N2+5)]-G'*data(1:N2);

%vytvoření matic A a b

A=[-tril(ones(Nu));tril(ones(Nu))];

b=[-ones(Nu,1)*umin+ones(Nu,1)*data(N2+4);ones(Nu,1)*umax-
    ones(Nu,1)*data(N2+4)];

%funkce kvadratického programování

deu=quadprog(H,k,A,b,[],[],[],[],[],options);

%určení výsledného akčního zásahu

du=deu(1);

u=data(N2+4)+du;
```