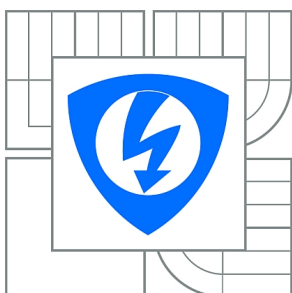




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

ADAPTIVNÍ OPTIMÁLNÍ REGULÁTORY S PRINCIPY UMĚLÉ INTELIGENCE V PROSTŘEDÍ MATLAB - B&R

ADAPTIVE OPTIMAL CONTROLLERS WITH PRINCIPLES OF ARTIFICIAL INTELLIGENCE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. VLADIMÍR BURLAK

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. PETR PIVOŇKA, CSc.

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Vladimír Burlak

ID: 78579

Ročník: 2

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Adaptivní optimální regulátory s principy umělé inteligence v prostředí MATLAB - B&R

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s metodikou návrhu adaptivních optimálních regulátorů. Seznamte se s použitím klasických identifikačních metod a identifikačních metod na bázi neuronových sítí. Zaměřte se na možnosti implementace adaptivního optimálního regulátoru z prostředí programu MATLAB do programovatelného automatu B&R. V identifikačním algoritmu realizujte grafické vyhodnocení průběhů identifikace. Ověřte a porovnejte adaptivní optimální regulátor s pevně nastaveným PID regulátorem ve spojení program MATLAB - program. automat B&R - simulační a fyzikální modely, zejména s ohledem na změnu dynamiky modelů. Porovnejte vlastnosti klasické identifikace a identifikačních algoritmů na bázi neuronových sítí. Implementujte adaptivní optimální regulátor do programovatelného automatu. Porovnejte se standardním adaptivním regulátorem firmy B&R.

DOPORUČENÁ LITERATURA:

PIVOŇKA, P.: Optimalizace regulátorů. VUT Brno, skriptum, 2005.

PIVOŇKA, P.: Číslíková řídicí technika, VUT Brno, skriptum, 2003

BOBÁL, V. a kol.: Praktické aspekty samočinně se nastavujících regulátorů. VUTUM, Brno, 1999.

Termín zadání: 8.2.2010

Termín odevzdání: 24.5.2010

Vedoucí práce: prof. Ing. Petr Pivoňka, CSc.

prof. Ing. Pavel Jura, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato diplomová práce pojednává o adaptivních optimálních regulátorech. Ukazuje principy optimálního řízení, identifikaci metodou nejmenších čtverců a metody identifikace založené na bázi umělých neuronových sítí.

Klíčová slova

adaptivní regulátor, optimální regulátor, LQ regulátor, rekurzivní identifikace, metoda nejmenších čtverců, identifikace s využitím neuronových sítí

Abstract

This master's thesis considers adaptive optimal controllers. It shows principles of optimal controllers, recursive identification using least-mean squares method and identification based on neural network.

Key words

adaptive controller, optimal controller, LQ controller, recursive identification, least-mean squares method, identification based on neural network

BURLAK, V. *Adaptivní optimální regulátory s principy umělé inteligence v prostředí MATLAB - B&R*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 84 s. Vedoucí diplomové práce prof. Ing. Petr Pivoňka, CSc.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma „Adaptivní optimální regulátory s principy umělé inteligence v prostředí MATLAB - B&R“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: **24. května 2010**

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce prof. Ing. Petru Pivoňkovi, CSc. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **24. května 2010**

.....
podpis autora

1. OBSAH

1. OBSAH	7
2. SEZNAM OBRÁZKŮ A TABULEK	9
2.1 Seznam obrázků	9
2.2 Seznam tabulek	10
3. ÚVOD	11
4. ADAPTIVNÍ ŘÍDICÍ SYSTÉMY	12
5. IDENTIFIKACE DYNAMICKÝCH SYSTÉMŮ	14
5.1 Model ARX	14
5.2 Klasické metody identifikace	16
5.2.1 Jednorázová metoda nejmenších čtverců	16
5.2.2 Průběžná metoda nejmenších čtverců	17
5.2.3 Volba struktury modelu a počáteční nastavení	18
5.3 Metody založené na umělých neuronových sítích	19
5.3.1 Neuron	19
5.3.2 Umělá neuronová síť	20
5.3.3 Učení umělé neuronové sítě	21
5.3.4 Identifikace dynamického systému jedním neuronem	22
5.4 Simulační ověření vlastností identifikačních metod	24
6. LINEÁRNÍ OPTIMÁLNÍ ŘÍZENÍ	29
6.1 Zpětná vazba od stavu	29
6.2 Definice diskrétního optimálního řízení [1]	31
6.3 Dynamické programování	31
6.4 Optimální řízení	33
6.5 Stavová reprezentace dynamického systému založená na hodnotách vstupu a výstupu	35
6.6 Stavový rekonstruktor	37
6.7 Modifikace LQ řízení	40
6.7.1 LQ řízení se sledováním referenční trajektorie	40
6.7.2 LQ řízení s asymptotickým sledováním referenční trajektorie	42

6.8 LQ řízení s diferenční složkou	44
6.8.1 LQ řízení doplněné o čistou diferenci výstupu	45
6.8.2 LQ řízení doplněné o diferenci výstupu s pevně nastaveným filtrem	46
6.8.3 LQG řízení doplněné o diferenci výstupu s pevně nastaveným filtrem	49
7. DEMONSTRAČNÍ PROGRAM ŘÍZENÍ	51
7.1 Demonstrační program Simulace LQ	51
7.2 Použité varianty diskretních ekvivalentů PID regulátorů	55
8. DEMONSTRAČNÍ PROGRAM ESTIMACE	58
9. TESTOVÁNÍ ŘÍZENÍ	63
9.1 Simulační výsledky řízení	63
9.2 Výsledky regulace fyzikální soustavy	69
9.3 Standardní adaptivní regulátor od firmy B&R.....	73
10. PROGRAMOVATELNÉ AUTOMATY	74
10.1 Implementace řídicích algoritmů do PLC	74
10.2 Implementace LQ řízení do PLC B&R.....	75
11. ZÁVĚR	79
12. LITERATURA.....	82
13. SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ	83

2. SEZNAM OBRÁZKŮ A TABULEK

2.1 SEZNAM OBRÁZKŮ

Obr. 4.1 Blokové schéma samočinně se nastavujícího regulátoru [1]	13
Obr. 4.2 Blokové schéma adaptivního LQ regulátoru [10].....	13
Obr. 5.1 Blokové schéma modelu ARX [1].....	15
Obr. 5.2 Umělý neuron [10].....	19
Obr. 5.3 Blokové schéma učení s učitelem [10]	21
Obr. 5.4 Uspořádání jednoho neuronu použitého pro identifikaci [10]	22
Obr. 5.5 Přechodové charakteristiky soustavy a modelů druhého řádu.....	25
Obr. 5.6 Přechodové charakteristiky soustavy a modelů třetího řádu s působícím poruchovým signálem	26
Obr. 5.7 Časové závislosti chyb predikcí výstupu	28
Obr. 6.1 Modelovací schéma klasického LQ řízení při použití vstupů a výstupů	37
Obr. 6.2 Schéma stavového rekonstruktoru	38
Obr. 6.3 Modelovací schéma LQ řízení se sledováním referenční trajektorie.....	42
Obr. 6.4 Modelovací schéma LQ řízení s asymptotickým sledováním referenční trajektorie	44
Obr. 6.5 Diferenční složka od výstupu soustavy.....	47
Obr. 6.6 Jiné zobrazení diferenční složky od výstupu soustavy	47
Obr. 6.7 Modelovací schéma LQ řízení s filtrací diferenční složky	49
Obr. 7.1 Blokové schéma činnosti simulační aplikace	51
Obr. 7.2 Hlavní formulář simulační aplikace.....	53
Obr. 7.3 Stavový portrét použitého PSD regulátoru	56
Obr. 7.4 Stavový portrét S-PD regulátoru.....	57
Obr. 8.1 Hlavní formulář demonstrační aplikace Estimace	58
Obr. 8.2 Formulář pro nastavení modelu	60
Obr. 8.3 Formulář pro nastavení identifikace	60
Obr. 8.4 Formulář pro nastavení signálů.....	61
Obr. 8.5 Formulář pro nastavení zobrazení.....	62

Obr. 9.1 Odezvy výsledného systému při změně dynamiky soustavy.....	66
Obr. 9.2 Odezvy výsledného systému při změně dynamiky soustavy.....	66
Obr. 9.3 Odezvy výsledného systému při působení poruchových signálů	68
Obr. 9.4 Odezvy výsledného systému při působení poruchových signálů	69
Obr. 9.5 Odezvy výsledného systému při změně dynamiky fyzikální soustavy.....	70
Obr. 9.6 Odezvy výsledného systému při změně dynamiky fyzikální soustavy.....	70
Obr. 9.7 Odezvy výsledného systému při působení poruchových signálů na fyzikální soustavu.....	72
Obr. 9.8 Odezvy výsledného systému při působení poruchových signálů na fyzikální soustavu.....	72

2.2 SEZNAM TABULEK

Tab. 5.1 Porovnání simulačních výsledků identifikace	27
Tab. 7.1 Přehled vytvořených modelů	52
Tab. 9.1 Hodnoty koeficientů pro simulaci LQ řízení	64
Tab. 9.2 Hodnoty parametrů pro simulaci PSD a S-PD řízení.....	64
Tab. 9.3 Souhrn výsledků dosažených při simulaci změny dynamiky soustavy	67
Tab. 9.4 Souhrn výsledků dosažených při simulaci působení poruchových signálů	68
Tab. 9.5 Hodnoty koeficientů pro LQ řízení fyzikální soustavy.....	70
Tab. 9.6 Hodnoty parametrů pro PSD a S-PD řízení fyzikální soustavy.....	70
Tab. 9.7 Souhrn výsledků dosažených při změně dynamiky fyzikální soustavy.....	71
Tab. 9.8 Souhrn výsledků dosažených při působení poruchových signálů na fyzikální soustavu.....	72
Tab. 9.9 Souhrn výsledků dosažených při působení poruchových signálů s použitím regulátoru R&R.....	73

3. ÚVOD

Optimální regulátory představují odklon od klasického přístupu k regulaci, která bývá často zastoupena regulátory typu PID nebo jejich diskrétními ekvivalenty. V případě klasického návrhu nastavuje regulační technik (nebo výpočetní algoritmus) parametry regulátoru přímo. V případě optimálních regulátorů se nenastavují jejich parametry přímo, ale jsou vždy určeny výpočtem, kterým je minimalizace kritéria. Požadované dynamické vlastnosti jsou zadány penalizačními maticemi. LQ řízení (lineární řízení, které splňuje kvadratické kritérium) převádí systém z libovolného bodu stavového prostoru do jeho počátku. Aby přístup LQ řízení vyhověl širokým požadavkům praxe, lze jeho strukturu rozšířit o vstup žádané hodnoty a člen s integračním charakterem, který eliminuje vliv konstantního poruchového signálu. Zařazení integrační složky však může způsobit zhoršení dynamických vlastností systému. Vystává tedy otázka, zda a jakým způsobem zakomponovat do řídicí struktury složku s diferenčním charakterem. Tato problematika je hlavní náplní této práce.

Pokud má být optimální regulátor schopen reagovat na změny dynamiky soustavy, je ji třeba neustále zkoumat (identifikovat) a na základě nalezených parametrů měnit parametry regulátoru. Tím se dostáváme k samočinně se nastavujícímu optimálnímu regulátoru.

Následující text je pomyslně rozdělen na tři celky. První se zabývá identifikací dynamických systémů, druhá LQ řízením a jeho ověřením a třetí přímou implementací řídicích algoritmů ze simulačního prostředí MATLAB/Simulink do průmyslového automatu firmy B&R.

4. ADAPTIVNÍ ŘÍDICÍ SYSTÉMY

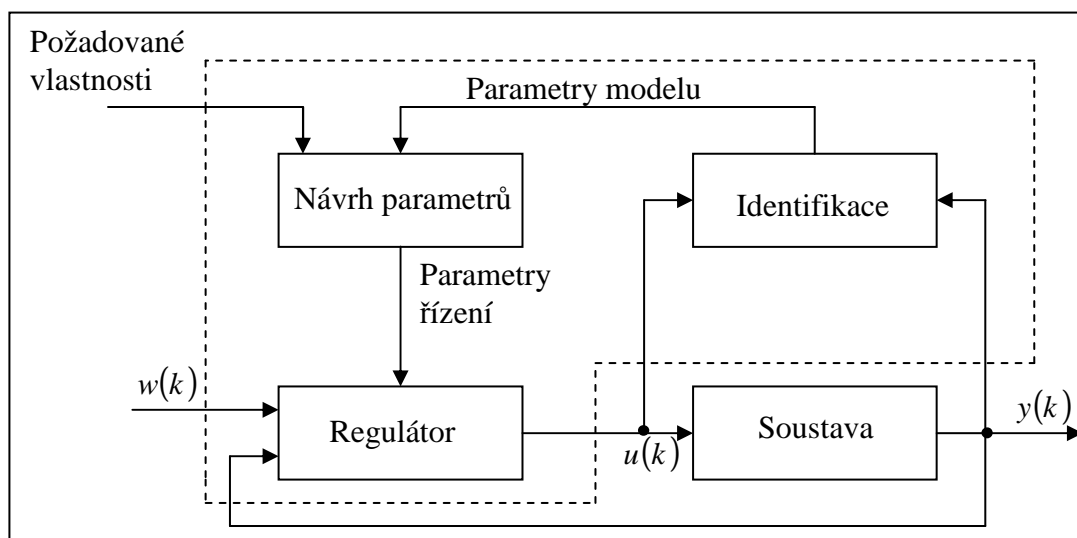
Pojem adaptace lze interpretovat jako schopnost přizpůsobit se změnám v okolí adaptujícího se systému ve snaze dosáhnout po těchto změnách opět nějakým způsobem optimálního stavu.

Pokud má řízená soustava časově stálé parametry, lze pro její řízení použít regulátor s pevně nastavenými parametry. V některých případech se parametry soustavy mohou měnit (například kvalita paliva) a použití regulátoru s pevně nastavenými parametry by vedlo ke snížení účinnosti celého systému. Pro takovéto systémy jsou vhodné adaptivní regulátory.

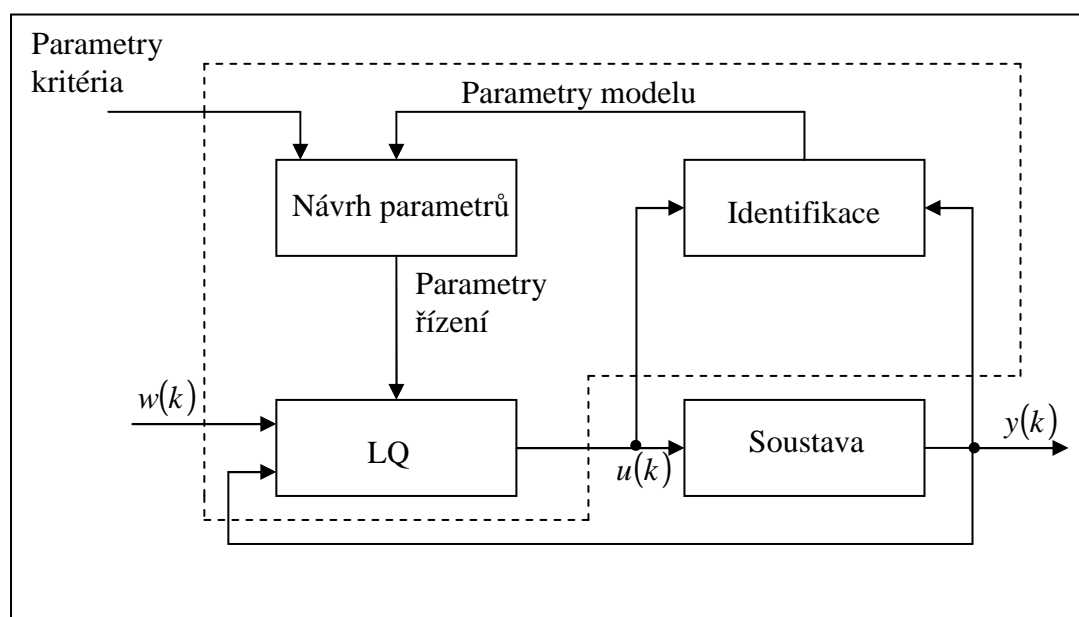
Typů adaptivních regulátorů existuje celá řada. Mezi základní přístupy adaptivních systémů patří adaptivní systémy založené na heuristickém přístupu, systémy s referenčním modelem MRAS (Model Adaptive Reference System) a samočinně se nastavující regulátory STC (Self-Tuning Controllers)[1]. Systémy založené na heuristickém přístupu využívají přímo hodnot regulační odchylky a výstupní veličiny a na základě těchto hodnot podle zvoleného kritéria (například kmitavost systému) modifikují parametry regulátoru. Metoda s referenčním modelem funguje tak, že paralelně k regulační smyčce je připojen referenční model, který nese informace o požadovaných vlastnostech systému. Adaptační mechanismus pracuje s rozdílem hodnoty výstupu řízené soustavy a hodnoty výstupu referenčního modelu. Podle tohoto rozdílu pak adaptační mechanismus volí parametry regulátoru nebo velikost přídatného signálu. Obě tyto metody nepotřebují průběžnou identifikaci řízené soustavy.

Předmětem této práce jsou samočinně se nastavující regulátory STC. Tuto skupinu lze rozdělit na systémy s implicitní identifikací (systémy, které neidentifikují přímo parametry řízené soustavy) a systémy s explicitní identifikací (systémy, které identifikují přímo parametry řízené soustavy). Dále lze skupinu STC rozdělit na systémy s jednorázovou identifikací a s průběžnou identifikací. Jednorázová identifikace umožňuje samočinné nastavení pevných parametrů regulátoru. Pokud jsou řízenou soustavou splněny podmínky použitelnosti algoritmu výpočtu parametrů regulátoru, může tento regulátor nasadit i člověk bez hlubších znalostí teorie řízení.

Tento přístup však nedokáže reagovat na změny dynamiky regulačního obvodu v čase. Pro sledování těchto změn slouží průběžná (on-line) identifikace, která provádí hledání parametrů obvodu neustále. Blokové schéma této struktury je na obr 4.1. Tato práce se dále zabývá samočinně se nastavujícími LQ regulátory (lineární regulátory splňující kvadratické kritérium) s průběžnou explicitní identifikací. Tento případ zobrazuje obr 4.2.



Obr. 4.1 Blokové schéma samočinně se nastavujícího regulátoru [1]



Obr. 4.2 Blokové schéma adaptivního LQ regulátoru [10]

5. IDENTIFIKACE DYNAMICKÝCH SYSTÉMŮ

Identifikace je proces, který slouží k nalezení modelu tak, aby tento model optimálně (podle zvoleného kritéria) aproximoval reálný systém. Cílem je nalézt takovou funkci (5.1)

$$y(t) = f(u(t), v(t), t) \quad (5.1)$$

kteřá popisuje závislost výstupu $y(t)$ na akčním zásahu $u(t)$, měřitelné poruše $v(t)$ a čase t [1]. Do identifikované soustavy také často vstupují neměřitelné poruchy $e_s(t)$. Potom vztah (5.1) přechází na tvar (5.2).

$$y(t) = f(u(t), v(t), t) + e_s(t) \quad (5.2)$$

Jednorázovou identifikaci lze provést vybuzením zkoumaného systému známým signálem (jednotkový skok, jednotkový impuls) a měřením výstupního signálu. Tím lze získat například přechodovou nebo impulsní charakteristiku, ze které lze určit parametry modelu. Pro průběžnou identifikaci je však třeba určit parametry modelu signálem, který současně představuje akční zásah během regulace soustavy. Pro identifikaci soustavy je třeba vždy nejprve zvolit strukturu modelu. Tento model musí věrně popisovat chování regulované soustavy a musí být dostatečně odolný vůči působení poruch. Nejčastěji se pro svoji jednoduchost používá model typu ARX.

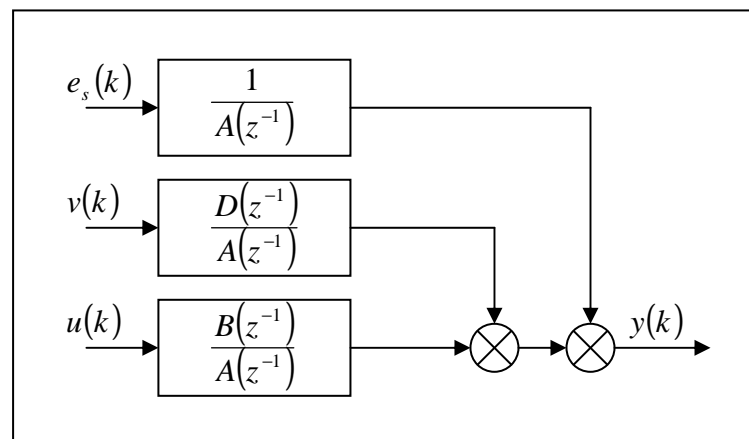
5.1 MODEL ARX

Model je dán diferenční rovnicí ve tvaru (5.3) [1]

$$y(k) = -\sum_{i=1}^{na} a_i y(k-i) + \sum_{i=1}^{nb} b_i u(k-i) + \sum_{i=1}^{nd} d_i v(k-i) + e_s(k) \quad (5.3)$$

kde $e_s(k)$ představuje stochastické vlivy působící na identifikovanou soustavu. Tuto rovnici lze přepsat na tvar (5.4). Jeho blokové schéma je zakresleno na obr. 5.1.

$$A(q^{-1})y(k) = B(q^{-1})u(k) + D(q^{-1})v(k) + e_s(k) \quad (5.4)$$



Obr. 5.1 Blokové schéma modelu ARX [1]

Vztah (5.3) lze také přepsat na vektorový tvar podle (5.5) [1]

$$y(k) = \theta^T(k)\varphi(k-1) + e_s(k) \quad (5.5)$$

kde

$$\theta^T(k) = [b_1 \quad \dots \quad b_{nb} \quad a_1 \quad \dots \quad a_{na} \quad d_1 \quad \dots \quad d_{nd}] \quad (5.6)$$

představuje vektor parametrů a

$$\varphi^T(k) = [u(k-1) \quad \dots \quad u(k-nb) \quad -y(k-1) \quad \dots \quad -y(k-na) \quad v(k-1) \quad \dots \quad v(k-nd)] \quad (5.7)$$

představuje vektor dat [1]. Aby v modelu nevznikaly algebraické smyčky, musí mít model pro řízení vyšší řád jmenovatele $A(q^{-1})$ než čitatele $B(q^{-1})$. Klíčovou roli pro hodnocení kvality modelu hraje chyba predikce definovaná podle (5.8).

$$\hat{e}(k) = y(k) - \hat{\theta}^T(k)\varphi(k) \quad (5.8)$$

Podle hodnoty výrazu (5.8) lze usuzovat na míru shody modelu a reálného procesu a také na vhodnost řádu modelu.

5.2 KLASICKÉ METODY IDENTIFIKACE

5.2.1 Jednorázová metoda nejmenších čtverců

Metoda nejmenších čtverců je založena na minimalizaci druhých mocnin chyb predikcí podle vztahu (5.9)

$$J = \frac{1}{2} \sum_{i=1}^n (y_i - \hat{\theta}^T \varphi_i)^2 \quad (5.9)$$

kde y_i je výstup reálného systému v kroku i , φ_i je vektor dat podle (5.11) v kroku i a $\hat{\theta}$ představuje hledaný vektor parametrů (5.10). Výsledný vztah parametrů modelu je dán vztahem (5.12).

$$\theta^T = (b_1 \quad \dots \quad b_m \quad -a_1 \quad \dots \quad -a_n) \quad (5.10)$$

$$\varphi^T(k) = (u(k-1) \quad \dots \quad u(k-m) \quad -y(k-1) \quad \dots \quad -y(k-n)) \quad (5.11)$$

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T y \quad (5.12)$$

Matice Φ vznikne z vektorů dat podle (5.13) a vektor y je tvořen jednotlivými hodnotami výstupu podle vztahu (5.14).

$$\Phi = \begin{pmatrix} \varphi_1^T \\ \varphi_2^T \\ \vdots \\ \varphi_n^T \end{pmatrix} \quad (5.13)$$

$$y^T = (y_1 \quad y_2 \quad \cdots \quad y_n) \quad (5.14)$$

Ve vztahu (5.12) lze použít kovarianční matici $P(i)$, která je definována podle vztahu (5.15).

$$P(i) = (\Phi^T \Phi)^{-1} \quad (5.15)$$

5.2.2 Průběžná metoda nejmenších čtverců

Pokud by byl při průběžné identifikaci použit předchozí algoritmus beze změny, docházelo by v každém kroku ke zvětšování rozměru matice Φ a vektoru y . Tento problém lze vyřešit použitím lemmy o inverzi matice, která zajišťuje zakomponování nově získaných hodnot do výpočtu bez nutnosti rozšiřovat velikosti matic a vektorů. Výsledné rekurzivní vztahy jsou uvedeny ve vztahu (5.16) [1].

$$\begin{aligned} K(i+1) &= P(i)\varphi(i+1)(1 + \varphi^T(i+1)P(i)\varphi(i+1)) \\ \hat{\theta}(i+1) &= \hat{\theta}(i) + K(i+1)(y(i+1) - \varphi^T(i+1)\hat{\theta}(i)) \\ P(i+1) &= P(i) - K(i+1)\varphi^T(i+1)P(i) \end{aligned} \quad (5.16)$$

Problém při použití této metody spočívá v menší numerické stabilitě, neboť při delším výpočtu začnou hodnoty kovarianční matice rychle narůstat. Tento problém lze vyřešit například exponenciálním zapomínáním. Pro použití exponenciálního zapomínání je třeba minimalizovat kritérium dané vztahem (5.17).

$$J = \sum_{i=k_0}^k \lambda^{2(k-i)} (y_i - \hat{\theta}^T \varphi_i)^2 \quad (5.17)$$

Rekurentní vztahy pro určení parametrů modelu přechází na tvar (5.18)[1].

$$\begin{aligned} K(i+1) &= P(i)\varphi(i+1)(\lambda(i+1)I + \varphi^T(i+1)P(i)\varphi(i+1)) \\ \hat{\theta}(i+1) &= \hat{\theta}(i) + K(i+1)(y(i+1) - \varphi^T(i+1)\hat{\theta}(i)) \\ P(i+1) &= (P(i) - K(i+1)\varphi^T(i+1)P(i))\lambda^{-1}(i+1) \end{aligned} \quad (5.18)$$

5.2.3 Volba struktury modelu a počáteční nastavení

Před samotným hledáním parametrů je třeba zvolit strukturu (řád) modelu. Nejčastěji se používá model druhého řádu ve tvaru (5.19) a model třetího řádu ve tvaru (5.20). Tyto modely jsou výhodné z toho důvodu, že u nich není problém nalézt spojitý ekvivalent. Tento ekvivalent je stejného řádu jako diskrétní tvar a neobsahuje nuly [1].

$$F_2(z^{-1}) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (5.19)$$

$$F_3(z^{-1}) = \frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} \quad (5.20)$$

Počáteční nastavení parametrů pro výpočet rekurzivních vztahů (5.18) se nejčastěji provádí hodnotami uvedenými v (5.21), kde I představuje jednotkovou matici.

$$P_0 = 10^9 I$$

$$\hat{\theta}_0 = (1 \ 0 \ \dots \ 0)^T \quad (5.21)$$

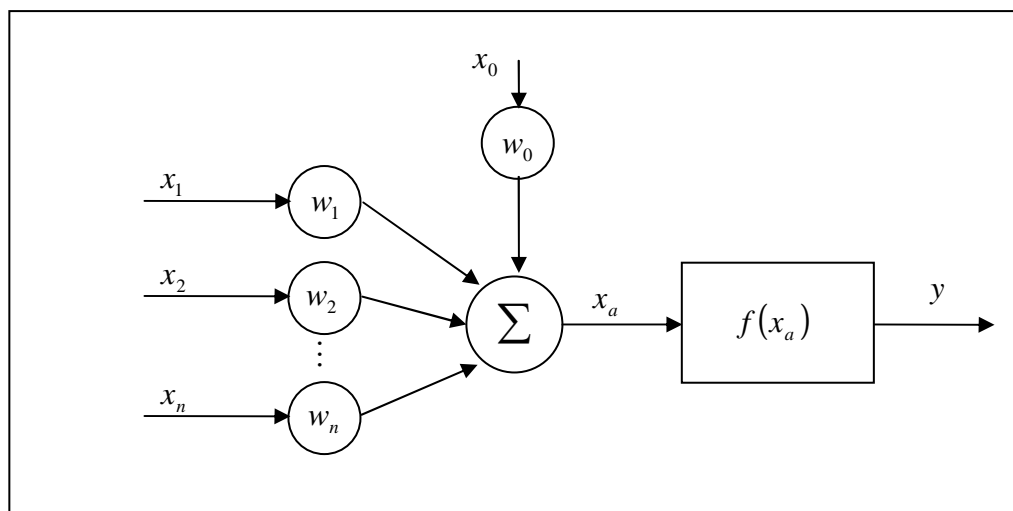
$$\lambda_0 \in \langle 0,995; 0,999 \rangle$$

5.3 METODY ZALOŽENÉ NA UMĚLÝCH NEURONOVÝCH SÍTÍCH

Problémy při použití metody nejmenších čtverců (například velký vliv periody vzorkování, numerická stabilita) vedly k hledání alternativního řešení, které bude robustnější. Jako vhodné řešení se ukázaly umělé neuronové sítě.

5.3.1 Neuron

Neuron tvoří elementární prvek umělých neuronových sítí. Tento prvek má n vstupů x_1, \dots, x_n , které jsou ohodnoceny váhami w_1, \dots, w_n . Má vždy jeden výstup y , který lze dále rozvětvit podle potřeby. Umělý neuron dále může obsahovat práh θ . Schéma umělého neuronu je zobrazeno na obr. 5.2.



Obr. 5.2 Umělý neuron [10]

Na umělý neuron lze nahlížet jako na nelineární prvek, který mapuje n vstupních hodnot na jednu výstupní. Tento proces je zde rozdělen na dvě části. První část tvoří agregační funkce. Ta má za úkol sloučit vstupní hodnoty na hodnotu skalární x_a . Tento postup je dán vztahem (5.22).

$$x_a = \sum_{i=1}^n w_i x_i - \theta = \sum_{i=1}^n w_i x_i + w_0 x_0 = \sum_{i=0}^n w_i x_i \quad (5.22)$$

Po vypočtení hodnoty x_a lze provést její mapování na výstupní hodnotu y , jež lze zapsat vztahem (5.23)

$$y = f(x_a) \quad (5.23)$$

kde funkce f je označována jako funkce aktivační. Tato funkce může být lineární nebo nelineární. Častým představitelem aktivační funkce je sigmoida. Celou činnost umělého neuronu lze tedy souhrnně zapsat ve tvaru (5.24).

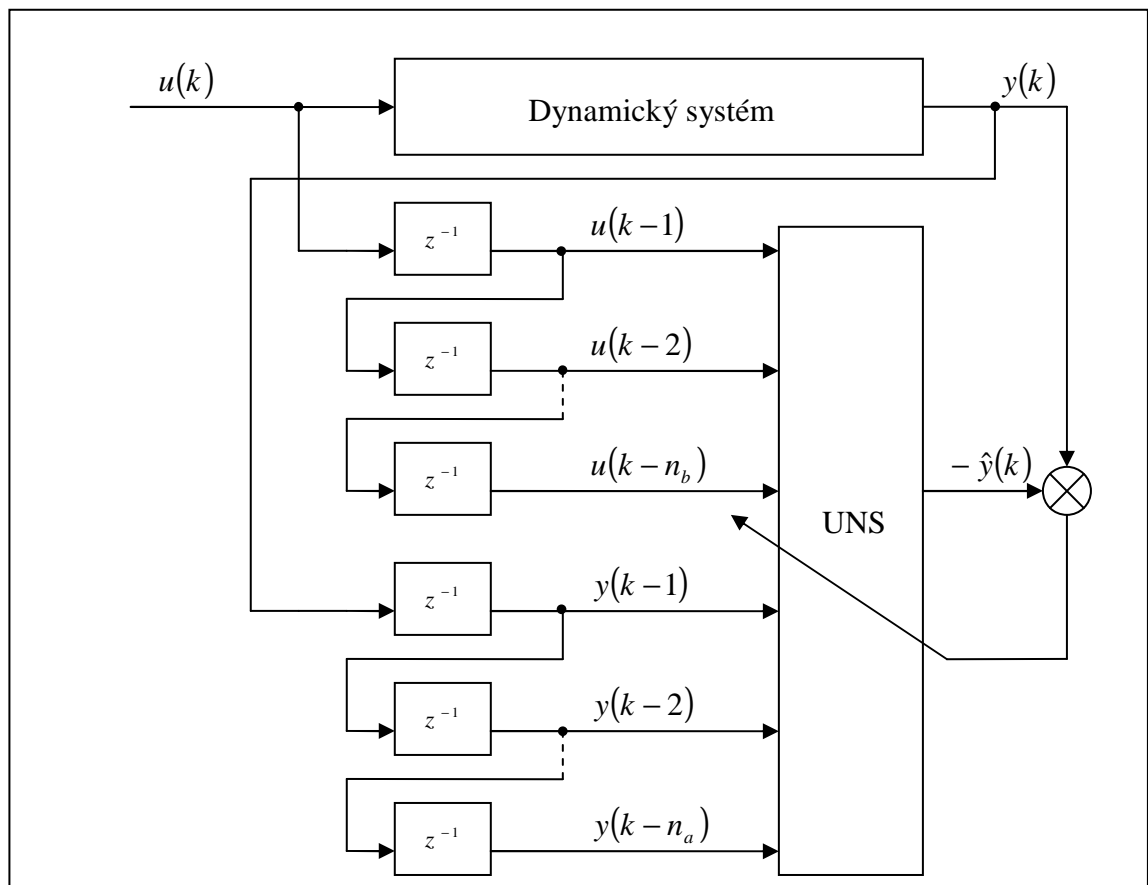
$$y = f\left(\sum_{i=1}^n w_i x_i - \theta\right) \quad (5.24)$$

5.3.2 Umělá neuronová síť

Umělé neurony je možno spojit do sítě. Existuje celá řada struktur, které mají své klady i zápory. Velmi oblíbenou sítí je dopředná neuronová síť. Tato síť je tvořena vstupní vrstvou, skrytými vrstvami a vrstvou výstupní. Dále je specifická tím, že neurony v sousedních vrstvách jsou propojeny každý s každým a neobsahuje žádné zpětné vazby. Problém tohoto uspořádání spočívá v nalezení optimálního počtu neuronů a optimálního počtu vrstev.

5.3.3 Učení umělé neuronové sítě

Procesem učení je myšleno uložení znalosti do UNS. Učení se dělí na dvě základní skupiny a to učení s učitelem a učení bez učitele. Pro identifikaci dynamických systémů bývá použito učení s učitelem, které pracuje tak, že na vstupy UNS přivedeme vektor dat (5.11) a určíme chybu predikce výstupní hodnoty. Podle této chyby jsou modifikovány parametry UNS tak, aby byla chyba minimalizována. Nejčastěji jsou modifikovány hodnoty vstupních vah. Základní blokové schéma učení s učitelem pro účely identifikace je zobrazeno na obr 5.3. Učení může být dále rozděleno na on-line učení (UNS je modifikována za chodu systému) a off-line učení (učení probíhá před použitím UNS).

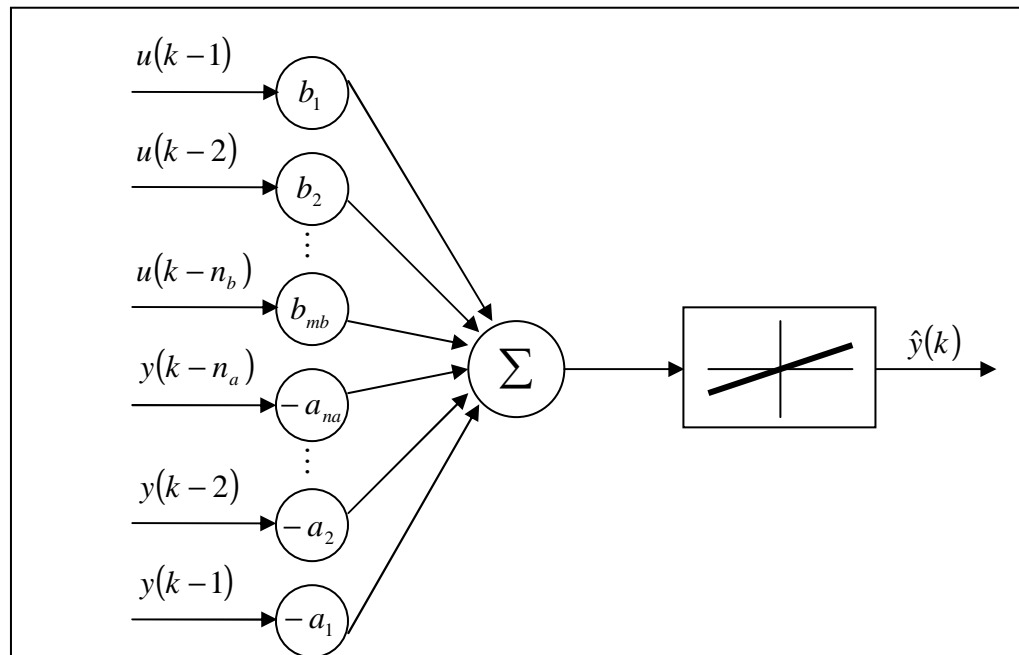


Obr. 5.3 Blokové schéma učení s učitelem [10]

5.3.4 Identifikace dynamického systému jedním neuronem

Pro identifikaci lze použít jeden neuron, který je zobrazen na obr 5.4 s lineární aktivační funkcí ve tvaru (5.25).

$$f(x_a) = x_a \quad (5.25)$$



Obr. 5.4 Uspořádání jednoho neuronu použitého pro identifikaci [10]

Pro soustavu popsanou přenosem (5.26) lze vypočítat predikci výstupní hodnoty vztahem (5.27), který lze upravit použitím vztahů (5.10) a (5.11).

$$F(z^{-1}) = \frac{b_1 z^{-1} + b_2 z^{-2} + \dots + b_{n_b} z^{-n_b}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{n_a} z^{-n_a}} \quad (5.26)$$

$$\hat{y}(k) = -\sum_{i=1}^{n_a} a_i y(k-i) + \sum_{i=1}^{n_b} b_i u(k-i) = \theta^T \varphi(k) \quad (5.27)$$

5.3.4.1 Metoda Back-propagation

Metoda Back-propagation je gradientní metoda, která minimalizuje kvadrát chyb predikcí ze vztahu (5.8). Metoda minimalizuje kritérium (5.28).

$$J = \frac{1}{2} e^T e \quad (5.28)$$

Aktualizaci vah lze nyní napsat ve tvaru (5.29)

$$w_{NN}(k+1) = w_{NN}(k) + \eta(w_{NN}(k) - w_{NN}(k-1)) + \mu e \varphi \quad (5.29)$$

kde η představuje momentum, μ představuje konstantu učení a vektor vah w je roven vektoru parametrů $\hat{\theta}$.

5.3.4.2 Metoda Levenberg-Marquardt

Výhodu této metody představuje dávkové předkládání trénovacích vzorů (vzory se předkládají najednou), což vede k urychlení učení. Matice vstupních vzorů X je dána vztahem (5.30)

$$X = \Phi^T \quad (5.30)$$

kde význam matice Φ je patrný ze vztahu (5.13). Vektor t představuje vektor výstupních vzorů (5.31)

$$t = y^T \quad (5.31)$$

kde význam vektoru y je patrný ze vztahu (5.14). Chybu predikce výstupní hodnoty lze ohodnotit vektorem e (5.32)

$$e = (t - \hat{y}) \quad (5.32)$$

kde \hat{y} označuje transponovaný vektor složený z hodnot predikcí výstupní hodnoty.

Aktualizace vah neuronu je formálně podobná metodě tečen, která slouží pro řešení nelineárních rovnic. Aktualizace probíhá podle vzorce (5.33)

$$\hat{\theta}(k+1) = \hat{\theta}(k) - (J^T J + \alpha I)^{-1} J^T e(k) \quad (5.33)$$

kde α značí konvergenční konstantu a J jakobián systému, pro který platí (5.34)

$$J = -X^T \quad (5.34)$$

Celkovou chybu lze určit ze vztahu (5.35).

$$e_c = \frac{1}{2} e^T e \quad (5.35)$$

5.4 SIMULAČNÍ OVĚŘENÍ VLASTNOSTÍ IDENTIFIKAČNÍCH METOD

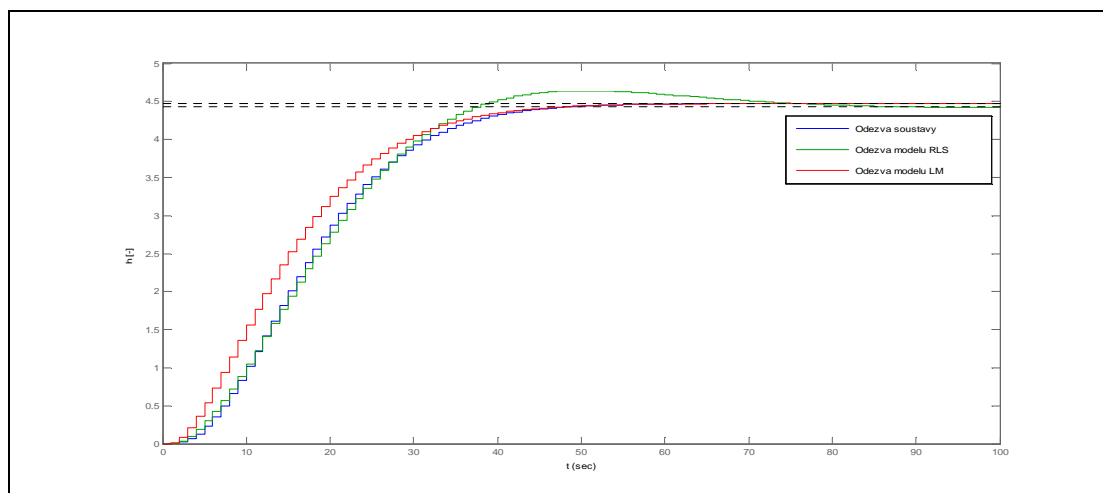
Tato kapitola má za cíl shrnout a porovnat dosažené výsledky rekurzivní metodou nejmenších čtverců (RLS) a umělou neuronovou sítí s učícím algoritmem Levenberg-Marquardt (LM). Učící algoritmus back-propagation nebyl pro testování použit, protože pro svou poměrně pomalou konvergenci se příliš nehodí pro on-line identifikaci adaptivního řízení. Data pro porovnání byla získána z programu Estimace, který je popsán v kapitole 8. Pro identifikaci byla zvolena soustava třetího řádu s operátorovým přenosem (5.36), která byla diskretizována se vzorkovací periodou jedna sekunda. Vztah (5.37) představuje operátorový přenos diskretizované soustavy, který byl zadán do programu Estimace.

$$F(s) = \frac{5}{(10s+1)(5s+1)^2} \quad (5.36)$$

$$F(z) = \frac{0,002944z^{-1} + 0,0104z^{-2} + 0,002293z^{-3}}{1 - 2,542z^{-1} + 2,152z^{-2} - 0,6065z^{-3}} \quad (5.37)$$

Pro všechny následující testy byl nastaven koeficient zapomínání rekurzivní metody nejmenších čtverců na hodnotu 0,999. Koeficient učení pro učící algoritmus Levenberg-Marquardt umělé neuronové sítě měl hodnotu 0,01 a hloubka paměti pro trénovací vzory byla zvolena 100.

Pro první porovnání byl zvolen matematický model aproximující soustavu druhého řádu. Tím lze ověřit, jak se jednotlivé algoritmy vypořádají s rozdílem struktury soustavy a modelu. Vstupní signál měl tvar jednotkového skoku a na výstupu soustavy nepůsobila porucha. Celá simulace trvala 100 sekund. Na obr. 5.5 jsou přechodové charakteristiky identifikované soustavy a matematických modelů získaných oběma identifikačními metodami na konci simulace (po 100 sekundách). Zhodnocení kvality predikce výstupního signálu je na konci této kapitoly. Závislost chyby predikce na čase je zobrazena na obr. 5.7.

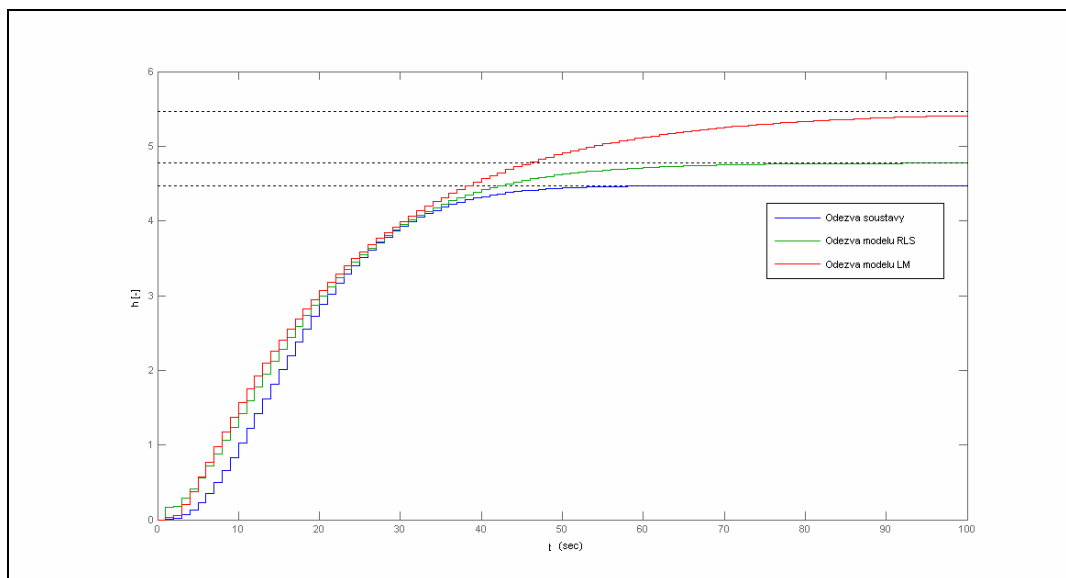


Obr. 5.5 Přechodové charakteristiky soustavy a modelů druhého řádu

Pro druhý experiment byl použit matematický model třetího řádu. Vstupním signálem byl opět jednotkový skok a opět na výstupu soustavy nepůsobil poruchový

signál. Jedná se tedy o případ, ve kterém lze předpokládat nejlepší výsledky, protože model neobsahuje rozdílnou strukturu ani na soustavu nepůsobí poruchové signály, a tedy identifikační algoritmy mohou nalézt parametry modelu přesně. Přechodové charakteristiky výsledných modelů zde nejsou uvedeny, protože jsou k nerozeznání od přechodové charakteristiky soustavy. Zhodnocení kvality predikce výstupního signálu lze nalézt opět na konci této kapitoly. Závislost chyby predikce na čase je zobrazena na obr. 5.7.

Poslední experiment se zabýval vlivem působící poruchy na výstupu soustavy na nalezený model a kvalitu predikce. Pro tuto simulaci byl zvolen matematický model třetího řádu. Vstupní signál měl obdélníkový průběh s periodou třicet sekund, perioda obdélníkového poruchového signálu byla nastavena na čtyřicet sekund. Délka simulace byla zvolena sto dvacet sekund. Obr. 5.6 zobrazuje přechodové charakteristiky soustavy a nalezených modelů na konci simulace.



Obr. 5.6 Přechodové charakteristiky soustavy a modelů třetího řádu s působícím poruchovým signálem

Integrální kritéria představují nástroj pro objektivní ohodnocení kvality predikce. V každém kroku je možno spočítat chybu predikce jako rozdíl výstupu soustavy a výstupu modelu. Ty jsou vloženy jako nezávisle proměnné do vhodné

funkce, jejíž výstupy jsou následně sečteny přes všechny kroky. V případě kvadratického kritéria jsou jednotlivé chyby predikcí umocněny na druhou. Za nejlepší výsledek pak lze považovat ten, jehož hodnota kritéria je nejmenší. Právě kvadratické kritérium bylo použito pro objektivní vyhodnocení průběhu predikce výstupu. Výše popsané kvadratické kritérium lze spočítat podle vztahu (5.38)

$$J = \sum_{i=0}^{n-1} (y(i) - y_M(i))^2 \quad (5.38)$$

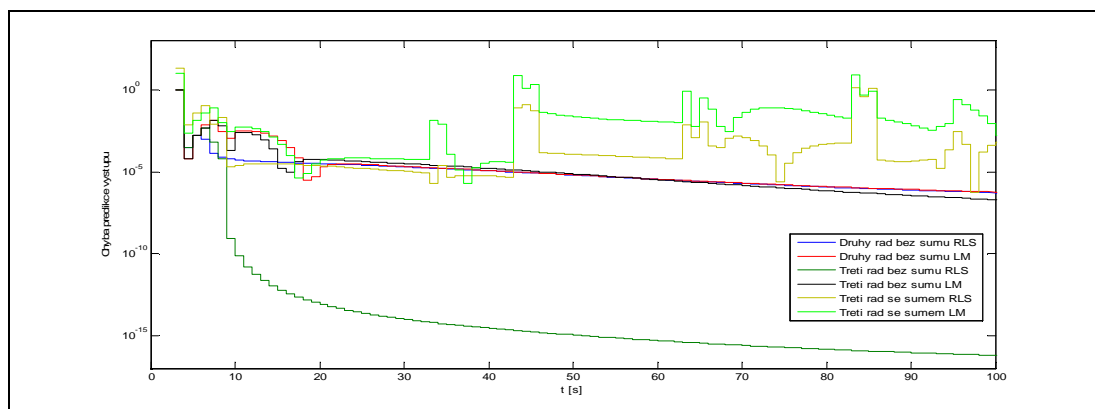
kde J představuje hodnotu kritéria, $y(i)$ je hodnota výstupu soustavy v kroku i , $y_M(i)$ představuje hodnotu výstupu matematického modelu v kroku i a n je počet kroků simulace. Tab 5.1 obsahuje vypočtené hodnoty kritérií pro jednotlivé simulace a jednotlivé modely. Výpočet byl proveden pro prvních sto kroků simulace.

Model	Vstupní signál	Poruchový signál	Metoda	Hodnota kritéria
2. řád	Jednotkový skok	-	RLS	0,9985
			LM	1,0364
3. řád	Jednotkový skok	-	RLS	1,0016
			LM	1,0302
	Obdélníkový průběh s periodou 30 s	Obdélníkový průběh s periodou 40 s	RLS	36,8574
			LM	42,4378

Tab. 5.1 Porovnání simulačních výsledků identifikace

Z tab. 5.1 je patrné, že ve všech případech dává rekurzivní metoda nejmenších čtverců (RLS) lepší výsledky než neuronová síť (LM) ve smyslu kvadratického kritéria. V případech, kdy na výstupu soustavy nepůsobil poruchový signál, jsou hodnoty kritérií vyrovnané. Při porovnání modelů druhého a třetího řádu si lze všimnout, že hodnota kritéria pro metodu nejmenších čtverců je pro druhý řád menší než pro řád třetí. To je způsobeno tím, že pro model třetího řádu je třeba nalézt o dva parametry více a je tedy potřeba mít k dispozici více dat ze simulace pro jejich stanovení. Při porovnání hodnot kritérií pro model třetího řádu s působící poruchou vychází znatelně lepší rekurzivní metoda nejmenších čtverců. Tento výsledek je zřejmě důsledkem rozdílného přístupu k zapomínání hodnot vstupů a výstupů

soustavy. Pro neuronovou síť totiž nejsou trénovací data váhována, ale každý vzor je brán se stejnou váhou jako vzory ostatní a pak po překročení hloubky paměti pro trénovací vzory zapomenut (přepsán jiným vzorem). Obr 5.7 zobrazuje časovou závislost chyb predikcí jednotlivých simulací. Pro větší přehlednost bylo zvoleno měřítko osy chyb predikcí logaritmické. Z tohoto důvodu nejsou v grafu vynášeny chyby predikcí ale jejich absolutní hodnoty.



Obr. 5.7 Časové závislosti chyb predikcí výstupu

Z obr. 5.7 je zřejmé, že chyba predikce výstupu nejrychleji klesá pro model třetího řádu s identifikačním algoritmem RLS. V prvních deseti sekundách je však vidět, že chyba predikce na tomto časovém úseku je větší, než pro soustavu druhého řádu se stejným identifikačním algoritmem. To ve výsledku způsobilo větší hodnotu kvadratického integrálního kritéria. Pro soustavu třetího řádu s identifikací neuronovou sítí klesá chyba predikce po modelu třetího řádu s metodou RLS nejrychleji. Rychlost klesání chyby predikce je však srovnatelná s modely druhého řádu. Pro modely druhého řádu vykazují na časovém úseku od dvaceti sekund obě metody identifikace (RLS i LM) stejně rychlý pokles chyby predikce. V časovém intervalu od nuly do dvaceti sekund je však patrné, že chyba pro metodu RLS je menší, než pro LM. Pro modely třetího řádu při působení poruchového signálu na výstupu soustavy je patrné, že při použití metody RLS jsou hodnoty chyb predikcí výstupu téměř na celém časovém intervalu menší než u modelu, který používal umělou neuronovou síť. To zapříčinilo takový rozdíl v hodnotách kvadratického integrálního kritéria.

6. LINEÁRNÍ OPTIMÁLNÍ ŘÍZENÍ

Pojem optimální je nejen v technické praxi vždy spojován s extrémem (maximem nebo minimem) kritériální funkce, která je řešitelem problému zvolena tak, aby systém měl vlastnosti odpovídající zadané úloze. Během dlouhého vývoje vznikla řada matematických nástrojů, které slouží k nalezení extrému zvoleného kritéria. Optimalizační problémy lze rozdělit na optimalizace statické a dynamické. U statických optimalizací je řešený problém zadán soustavou lineárních, resp. nelineárních rovnic a mohou být řešeny například lineárním, resp. nelineárním programováním. Tyto metody jsou použitelné u soustav, které mají zanedbatelnou dynamiku a nejsou tedy předmětem této práce. Naproti tomu dynamické systémy jsou popsány diferenciálními, resp. diferenčními rovnicemi. Pro řešení tohoto optimalizačního problému lze použít například variační počet, Pontrjaginův princip maxima nebo dynamické programování. Všechny tyto metody jsou popsány v [9], [3].

Jedním ze základů moderní teorie řízení je právě nalezení minima kritéria. Parametry regulátoru se pak nenastavují přímo, ale jsou určeny výpočtem ze zvoleného kritéria.

6.1 ZPĚTNÁ VAZBA OD STAVU

Zpětná vazba od stavového vektoru umožňuje, proti zpětné vazbě od výstupu, mnohem větší možnosti úpravy dynamiky dynamického systému. Jestliže máme například spojitý dynamický systém, pro který platí $n > m$, popsáný operátorovým přenosem (6.1)

$$F(p) = \frac{b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p + b_0}{p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0} \quad (6.1)$$

můžeme tento systém převést na Frobeniův kanonický tvar ve tvaru (6.2).

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_n(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} u(t)$$

$$y(t) = (b_0 \quad b_1 \quad \cdots \quad b_m \quad 0 \quad \cdots \quad 0) \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix}$$

(6.2)

Připojíme-li k tomuto systému stavový regulátor, jehož cílem je převést systém z libovolného počátečního stavu do počátku stavového prostoru, dostaneme stavové rovnice ve tvaru (6.3), které lze přepsat na maticový tvar (6.4).

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_n(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ -a_0 + k_0 & -a_1 + k_1 & \cdots & -a_{n-1} + k_{n-1} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix} \quad (6.3)$$

$$y(t) = (b_0 \quad b_1 \quad \cdots \quad b_m \quad 0 \quad \cdots \quad 0) \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix}$$

$$\dot{x}(t) = (A - BK)x(t) \quad (6.4)$$

$$y(t) = Cx(t)$$

Stavovým regulátorem, jak je patrné ze vztahu (6.3), lze ovlivnit všechny koeficienty polynomu jmenovatele operátorového přenosu řízené soustavy. Tímto způsobem lze výrazným způsobem snížit dynamiku systému nebo systém stabilizovat. Při volbě koeficientů $k_{n-1} \dots k_0$ musí však být respektována taková

omezení, aby bylo řízení přípustné. Stavový regulátor pro svoji činnost potřebuje hodnoty všech stavových proměnných. Pokud jsou některé neměřitelné, je třeba použít pro jejich rekonstrukci stavový rekonstruktor, který je popsán například v [5].

6.2 DEFINICE DISKRÉTNÍHO OPTIMÁLNÍHO ŘÍZENÍ [1]

„Je dán plně říditelný systém (6.5)

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), t) \\ y(t) &= g(x(t), u(t), t) \end{aligned} \quad (6.5)$$

s definovaným počátečním a koncovým stavem a kritériem optimálního řízení (6.6)

$$J = \frac{1}{2} (x^T(N) S x(N)) + \frac{1}{2} \left(\sum_{k=0}^N x^T(k) Q x(k) + u^T(k) R u(k) \right) \quad (6.6)$$

a úkolem optimálního řízení je najít takové řízení $u^*(t)$, které

1. převádí systém (6.5) z počátečního do koncového stavu
2. patří do třídy přípustných řízení
3. minimalizuje dané kritérium (6.6)“

Pozn.: Matice Q a S musí být pozitivně semidefinitní a matice R musí být pozitivně definitní. Tím je zaručeno, že nízká hodnota kritéria není způsobena nepříznivým odečtením jednotlivých složek od sebe.

6.3 DYNAMICKE PROGRAMOVÁNÍ

V této části bylo čerpáno z [7].

Dynamické programování je jedním z velmi účinných a oblíbených nástrojů pro numerické řešení optimalizačních problémů. Dynamické programování používá pro řešení optimalizačního problému rozdělení dané úlohy na větší počet dílčích

problémů. Toto rozdělení je u úloh diskrétního řízení přítomno již z principu (vstup řízeného systému je pro dosažení žádaného stavu buzen posloupností hodnot vstupního signálu).

Tato metoda používá dva základní principy - princip optimality a princip invariantního vnoření [9].

Pro řešení optimalizace potřebujeme nejdříve zavést funkci, která nám v každém kroku ohodnotí kvalitu rozhodnutí $f(x(k), u(k))$. Následně zavedeme funkci (6.7), jejíž hodnota je tvořena součtem hodnot funkce $f(x(k), u(k))$ v aktuálním a ve všech následujících krocích

$$J_k^N = \sum_{i=k}^N f(x(i), u(i)) \quad (6.7)$$

kde $x(k)$ je vektor stavových proměnných systému, vektor $u(k)$ představuje vektor vstupních signálů a N je počet kroků (časový horizont řízení).

Je zřejmé, že cílem je nalézt takovou posloupnost rozhodování (řízení), aby v každém kroku byla hodnota výrazu (6.7) extrémní, tedy maximální nebo minimální. Protože pouhou záměnou znamének hodnot vypočítaných podle (6.7) lze problém hledání maxima převést na problém hledání minima, bude zde dále uvažováno pouze hledání minima.

Hledání posloupnosti hodnot vektoru vstupních signálů provádíme odzadu (od kroku N), neboť tento stav již nelze ovlivnit prostřednictvím vstupu. První krok výpočtu je naznačen ve vztahu (6.8).

$$I_N^N = f(x(N), 0) \quad (6.8)$$

K -tý krok lze vypočítat podle (6.9).

$$I_k^N = \min_{u(N-k)} (f(x(N-k), u(N-k)) + I_N^N(x(N-k+1))) \quad (6.9)$$

Takto řešíme jednotlivé kroky až ke kroku prvnímu (z počátečního stavu)
(6.10).

$$I_{N-1}^N = \min_{u(N-1)} (f(x(0), u(0)) + I_1^N(x(1))) \quad (6.10)$$

Obecně lze napsat řešení podle vztahu (6.11) [7].

$$\begin{aligned} I_k^N &= \min_{u(k)} \left[f(x(k), u(k)) + \min_{u(k+1) \dots u(N)} \sum_{i=k+1}^N f(x(i), u(i)) \right] = \\ &= \min_{u(k)} [f(x(k), u(k)) + I_{k+1}^N] \end{aligned} \quad (6.11)$$

6.4 OPTIMÁLNÍ ŘÍZENÍ

V této části bylo čerpáno z [1].

Mějme diskrétní t-invariantní dynamický systém popsany stavovými rovnicemi ve tvaru (6.12).

$$x(k+1) = Ax(k) + Bu(k) \quad (6.12)$$

$$y(k) = Cx(k)$$

a kritérium optimality ve tvaru (6.13)

$$J = \frac{1}{2} x_N^T S_N x_N + \frac{1}{2} \sum_{k=1}^{N-1} (x_k^T Q x_k + u_k^T R u_k) \quad (6.13)$$

kde N představuje časový horizont řízení. Použitím dynamického programování lze odvodit (například v [1]), že pro hodnoty vektoru vstupních signálů v kroku l bude platit vztah (6.14).

$$u_l = -(B^T S_{l+1} B + R)^{-1} B^T S_{l+1} A x_l = -K_l x_{l-1} \quad (6.14)$$

Matici S_N lze určit ze vztahu (6.15) a pro hodnotu kritéria platí vztah (6.16).

$$S_l = Q + A^T S_{l+1} A - A^T S_{l+1} B (B^T S_{l+1} B + R)^{-1} B^T S_{l+1} A \quad (6.15)$$

$$J_l^{opt} = \frac{1}{2} x_l^T S_l x_l \quad (6.16)$$

Touto cestou získáme časově proměnný stavový regulátor, ve kterém je akční zásah dán zesílením stavu. Tato varianta však skrývá řadu problémů při praktické realizaci. Tento problém je možno odstranit převedením úlohy na úlohu s nekonečným horizontem ($N \rightarrow \infty$), čímž získáme kritérium ve tvaru (6.17). V tomto případě nám úloha přechází na řešení Riccatiho algebraické rovnice, ze které nás zajímá pouze pozitivně definitní řešení matice P , pro kterou platí $S \rightarrow P \rightarrow P_k = P_{k+1}$. Tím získáme konstantní zpětnovazební matici K .

$$J = \frac{1}{2} \sum_{k=1}^{\infty} (x_k^T Q x_k + u_k^T R u_k) \quad (6.17)$$

Mějme autonomní systém, který je popsán stavovými rovnicemi ve tvaru (6.18)

$$\begin{aligned} \bar{x}_{k+1} &= (A - BK)x_k \\ \bar{y}_k &= Cx_k \end{aligned} \quad (6.18)$$

kde K představuje matici zesílení ve zpětných vazbách od stavu. Nyní můžeme přepsat kritérium z tvaru (6.17) na tvar (6.19).

$$J = \frac{1}{2} \sum_{k=0}^N x_k^T (Q + K^T R K) x_k \quad (6.19)$$

Zpětnovazební matici K získáme nalezením pozitivně definitního řešení maticové rovnice (6.20) s proměnnou P .

$$P = (A - BK)^T P (A - BK) + Q + K^T R K \quad (6.20)$$

Pro řešení rovnice (6.20) lze použít Kleinmanův iterační algoritmus [1].

1. zvolíme matici $K^{(0)}$ aby matice $A - BK^{(0)}$ byla stabilní
2. vypočteme $P^{(i+1)} = (A - BK^{(i)})^T P^{(i)} (A - BK^{(i)}) + Q + K^{(i)T} R K^{(i)}$
3. vypočteme zesílení $K^{(i+1)} = (R + B^T P^{(i)} B)^{-1} B^T P^{(i)} A$
4. opakujeme body 2. a 3. dokud nedosáhneme předem zvoleného počtu opakování, nebo dokud se sobě odpovídající hodnoty dvou po sobě následujících matic K neliší o méně, než je předem stanovená přesnost

U adaptivního optimálního regulátoru tento algoritmus musí být opakován při každé periodě vzorkování. Takovýto regulátor využívá stavových proměnných a převádí systém z libovolného počátečního stavu do počátku stavového prostoru.

6.5 STAVOVÁ REPREZENTACE DYNAMICKÉHO SYSTÉMU ZALOŽENÁ NA HODNOTÁCH VSTUPU A VÝSTUPU

Předchozí článek pojednával o metodě pro výpočet optimálních hodnot zpětnovazební matice K při použití stavových rovnic. V praxi však může nastat situace, že stavové proměnné nebudou měřitelné. V takovém případě je možno řešit tento problém stavovým rekonstruktorem nebo stavovou reprezentací systému založenou na hodnotách vstupu a výstupu.

Pokud je použit stavový rekonstruktor pro adaptivní self-tuning systém, je třeba pro získání správného modelu systému nalézt dvě sady parametrů. První sadu

Lze získat on-line identifikací. Jedná se o číselné a jmenovatelové koeficienty operátorového přenosu, kterým je dynamický systém (model řízené soustavy) popsán. Následně je třeba docílit shodnosti stavu reálného systému a modelu systému. Teprve po rovnosti obou stavů lze použít hodnoty modelu pro výpočet akčních zásahů. Pokud by totiž systém použil pro výpočet akčního zásahu špatné hodnoty stavových proměnných, následky by mohly být katastrofální. Z tohoto postupu je patrné, že při realizaci budou vznikat problémy s ověřováním nejen zda již systém má správné hodnoty koeficientů operátorového přenosu, ale také zda již systém našel správný stav regulované soustavy.

Základní myšlenka stavové reprezentace, založené na vstupu a výstupu soustavy, spočívá v zavedení nového stavového vektoru, který je tvořen předchozími hodnotami vstupního a výstupního vektoru řízené soustavy. Tyto hodnoty jsou získány s maximální přesností, které je řídicí systém schopen dosáhnout a odpadá tedy potřeba testování, zda již má řídicí systém k dispozici správné hodnoty stavových proměnných. Rovnice (6.22) představují příklad nově utvořených stavových rovnic pro diskrétní SISO systém třetího řádu (6.21), které jsou formálně shodné s běžnými stavovými rovnicemi.

$$F(z) = \frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} \quad (6.21)$$

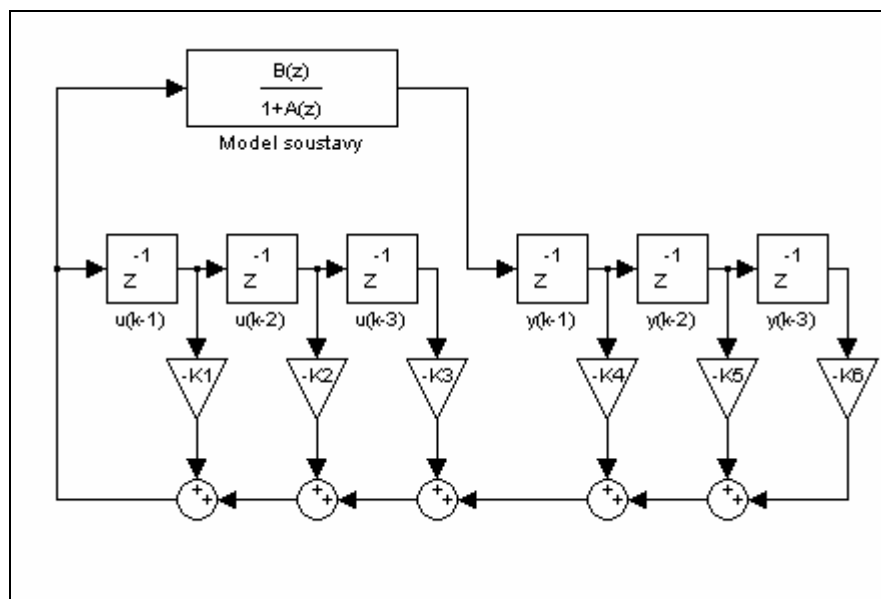
$$\begin{pmatrix} u(k) \\ u(k-1) \\ u(k-2) \\ y(k) \\ y(k-1) \\ y(k-2) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ b_1 & b_2 & b_3 & -a_1 & -a_2 & -a_3 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} u(k-1) \\ u(k-2) \\ u(k-3) \\ y(k-1) \\ y(k-2) \\ y(k-3) \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} u(k) \quad (6.22)$$

$$y(k) = (b_1 \quad b_2 \quad b_3 \quad -a_1 \quad -a_2 \quad -a_3) \begin{pmatrix} u(k-1) \\ u(k-2) \\ u(k-3) \\ y(k-1) \\ y(k-2) \\ y(k-3) \end{pmatrix}$$

Rovnice (6.22) můžeme přepsat na tvar (6.23)

$$\begin{aligned} z(k+1) &= A'z(k) + B'u(k) \\ y(k) &= C'z(k) \end{aligned} \quad (6.23)$$

Modelovací schéma této struktury je zobrazeno na obr. 6.1.

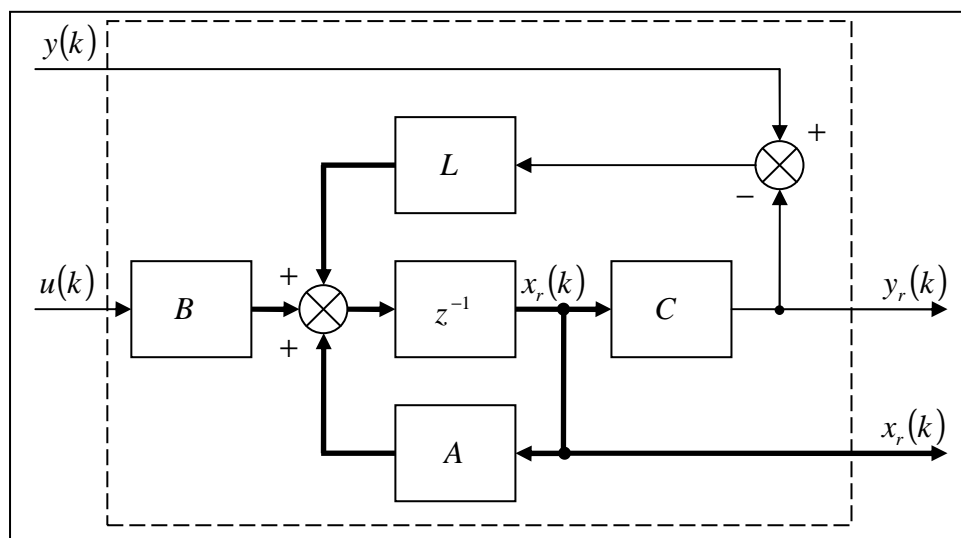


Obr. 6.1 Modelovací schéma klasického LQ řízení při použití vstupů a výstupů

6.6 STAVOVÝ REKONSTRUKTOR

Jak již bylo zmíněno v předchozí kapitole, většina regulovaných soustav nemá měřitelné stavové proměnné. Pokud by z jakéhokoli důvodu nebylo vhodné použít pro řízení stavovou reprezentaci založenou na předchozích hodnotách vstupu a výstupu soustavy, je třeba hodnoty stavových proměnných zrekonstruovat. K tomu slouží stavový rekonstruktor. Jedná se v podstatě o model regulované soustavy, jehož vstupy jsou připojeny na vstup a výstup soustavy. Prostřednictvím svých vstupů upravuje svůj stav tak, aby byla nalezena maximální možná shoda výstupů soustavy a modelu. Protože dynamika modelu a soustavy je velice podobná, mají oba systémy společný vstup. Samotné řešení se společným vstupem ovšem nestačí, neboť neřeší

počáteční stav regulované soustavy a nepřesnost modelu. K řešení tohoto problému se využívá výstupu soustavy, respektive rozdílu mezi výstupem soustavy a modelu. Tento rozdíl je dále vynásoben maticí L a výsledek sečten se stavem modelu. Tím je vytvořena vazba přímo do stavu modelu, jejíž možnosti jsou popsány v kapitole 6.1. Aby bylo možno dosáhnout shody obou stavů rychle, musí mít výsledný model mnohem menší časové konstanty než regulovaná soustava. Toho je zpravidla dosaženo velkými hodnotami v matici L . Pro účely této práce byl použit stavový rekonstruktor, který rekonstruuje všechny hodnoty stavových proměnných a protože má být použit pro adaptivní řízení, ve kterém modely on-line identifikace neuvažují proporcionalní složku soustavy, není ani zde matice přímých vazeb mezi vstupem a výstupem uvažována. Schéma stavového rekonstruktora je zobrazeno na obr. 6.2.



Obr. 6.2 Schéma stavového rekonstruktora

Rovnice (6.24) představují stavové rovnice rekonstruktora

$$\begin{aligned} x_r(k+1) &= Ax_r(k) + Bu(k) + L(y(k) - y_r(k)) \\ y_r(k) &= Cx_r(k) \end{aligned} \quad (6.24)$$

kde $x_r(k)$ představuje stav modelu v kroku k , $y_r(k)$ je výstup modelu, $y(k)$ představuje výstup regulované soustavy, $u(k)$ je vstupní signál regulované soustavy, matice L představuje matici pro nalezení shody stavů prostřednictvím rozdílů výstupů a význam matic A , B a C je patrný z rovnice (6.12).

Pro použití stavového rekonstruktoru je třeba navrhnout matici L . Nejprve je třeba vypočítat obdobu inverzní matice říditelnosti. Tento výpočet je formálně shodný s výpočtem matice říditelnosti, jen je proveden z transponovaných matic A a C . Samotný výpočet matice říditelnosti lze nalézt například v [12]. Její poslední řádek představuje proměnnou p_1 . Dále je potřeba zvolit vlastní čísla nově navrhovaného systému λ_i a podle rovnice (6.25) vypočítat polynom $R(z)$, kde n představuje řád systému. Výsledný výpočet matice L popisuje rovnice (6.26).

$$R(z) = \prod_{i=0}^{n-1} (z - \lambda_i) = z^n + r_{n-1}z^{n-1} + \dots + r_1z + r_0 \quad (6.25)$$

$$L^T = p_1(A^T)^n + p_1 \sum_{i=0}^{n-1} r_i(A^T)^i \quad (6.26)$$

Pro použití stavového rekonstruktoru v adaptivním řízení je nezbytné přizpůsobovat model aktuálním parametrům soustavy. Tyto parametry lze získat z on-line identifikace. Zde použité metody identifikace mají na svém výstupu vektor parametrů operátorového přenosu se zvolenou strukturou. Protože přechod od operátorového přenosu ke stavovým rovnicím není jednoznačný, bylo třeba zvolit tvar stavových rovnic. Pro všechny následující kapitoly byl zvolen tvar ve Frobeniově kanonickém tvaru podle (6.2), který odpovídá zápisu podle (6.12). Změny parametrů regulované soustavy se v tomto případě projeví v maticích A a C a je potřeba při každé změně v těchto maticích provést nový výpočet matice L .

V této kapitole bylo čerpáno z [5].

6.7 MODIFIKACE LQ ŘÍZENÍ

Výše odvozené řízení převádí řízenou soustavu z libovolného počátečního stavu do počátku stavového prostoru. Častou úlohou řízení však bývá převedení systému z libovolného počátečního stavu do takového stavu, aby hodnoty výstupního vektoru konvergovaly k hodnotám vektoru žádaných hodnot. Tato úprava bývá označována jako LQ řízení se sledováním referenční trajektorie. Kromě vybuzení výstupu systému na požadovanou hodnotu je často třeba eliminovat vliv poruch, působících na řízený systém. Aby systém dokázal vyregulovat konstantní poruchu působící na vstupu regulované soustavy, musí být před místem jejího působení ve struktuře řídicího systému zařazen člen s integračním charakterem. Tato úprava bývá označována jako LQ řízení s asymptotickým sledováním referenční trajektorie.

Dále se v textu hovoří o SISO systémech, neboť pro MIMO systémy lze použít stejný postup. Jediným rozdílem je zvětšení rozměru jednotlivých skalárních proměnných na vektory či matice. V této kapitole byla použita stavová reprezentace, založená na předchozích hodnotách vstupu a výstupu.

6.7.1 LQ řízení se sledováním referenční trajektorie

Pro potřeby sledování referenční trajektorie lze uvažovat kritérium s nekonečným časovým horizontem ve tvaru (6.24)

$$J = \sum_{k=0}^{\infty} (q_e (w(k) - y(k))^2 + q_u u(k)^2) \quad (6.24)$$

kde $y(k)$ představuje výstup v diskrétním čase k , $w(k)$ představuje výstup generátoru referenční trajektorie a $u(k)$ představuje vstup řízené soustavy. Kritérium podle (6.24) vyžaduje rozšíření stavového prostoru o generátor žádané hodnoty. Pro konstantní žádanou hodnotu platí vztahy (6.25).

$$\begin{aligned} w(k+1) &= w(k) \\ y(k) &= w(k) \end{aligned} \quad (6.25)$$

Nyní můžeme rozšířit vektor $z(k)$ ze vztahu (6.23) o stav generátoru žádané hodnoty. Stavové rovnice lze přepsat do tvaru (6.26).

$$\begin{pmatrix} z(k+1) \\ w(k+1) \end{pmatrix} = \begin{pmatrix} A' & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z(k) \\ w(k) \end{pmatrix} + \begin{pmatrix} B' \\ 0 \end{pmatrix} u(k) \quad (6.26)$$

$$y(k) = \begin{pmatrix} C' & 0 \end{pmatrix} \begin{pmatrix} z(k) \\ w(k) \end{pmatrix}$$

Použitím vztahů (6.14) a (6.26) lze přepsat kritérium (6.24) na tvar (6.27), které je formálně shodné se vztahem (6.19).

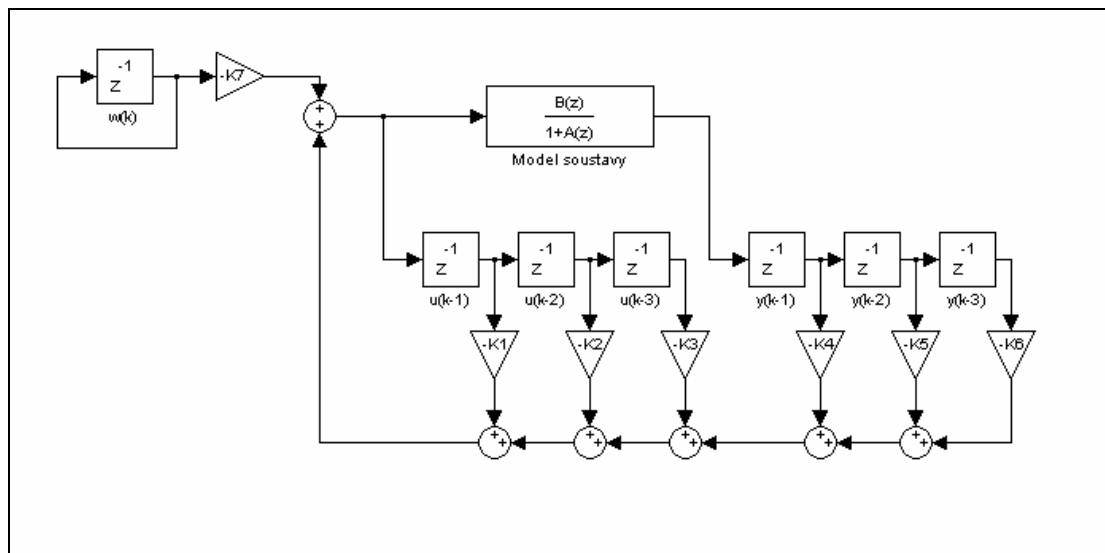
$$J = \sum_{k=0}^{\infty} \left(\begin{pmatrix} z(k) \\ w(k) \end{pmatrix}^T (M^T Q M) \begin{pmatrix} z(k) \\ w(k) \end{pmatrix} \right) \quad (6.27)$$

Matice M umožňuje výpočet hodnot vstupu a výstupu v čase k a matice Q představuje matici penalizačních koeficientů umístěných tak, aby bylo dosaženo shody kritérií (6.24) a (6.27). Tato dekompozice zajišťuje průhlednost významu jednotlivých prvků matice Q . Pro soustavu třetího řádu je matice M ve tvaru (6.28) a matice Q ve tvaru (6.29). Následný výpočet je již shodný s postupem popsáním v kapitole 6.4.

$$M = \begin{pmatrix} -K_1 & -K_2 & -K_3 & -K_4 & -K_5 & -K_6 & -K_7 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ b_1 & b_2 & b_3 & -a_1 & -a_2 & -a_3 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (6.28)$$

$$Q = \begin{pmatrix} q_u & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_e & 0 & 0 & -q_e \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -q_e & 0 & 0 & q_e \end{pmatrix} \quad (6.29)$$

Modelovací schéma této struktury je zobrazeno na obr. 6.2.



Obr. 6.3 Modelovací schéma LQ řízení se sledováním referenční trajektorie

6.7.2 LQ řízení s asymptotickým sledováním referenční trajektorie

V tomto případě je třeba přidat další stavovou proměnnou, pro kterou platí vztahy (6.30).

$$\begin{aligned} x_i(k+1) &= x_i(k) + w(k) - y(k) \\ y(k) &= x_i(k) \end{aligned} \quad (6.30)$$

Stavové rovnice jsou nyní ve tvaru (6.31)

$$\begin{pmatrix} z(k+1) \\ w(k+1) \\ x_i(k+1) \end{pmatrix} = \begin{pmatrix} A' & 0 & 0 \\ 0 & 1 & 0 \\ -C' & 1 & 1 \end{pmatrix} \begin{pmatrix} z(k) \\ w(k) \\ x_i(k) \end{pmatrix} + \begin{pmatrix} B' \\ 0 \\ 0 \end{pmatrix} u(k) \quad (6.31)$$

$$y(k) = (C' \quad 0 \quad 0) \begin{pmatrix} z(k) \\ w(k) \\ x_i(k) \end{pmatrix}$$

Po tomto rozšíření lze použít kritérium ve tvaru (6.32), které lze přepsat na tvar (6.33). Matice Q a M jsou dány vztahy (6.34) a (6.35).

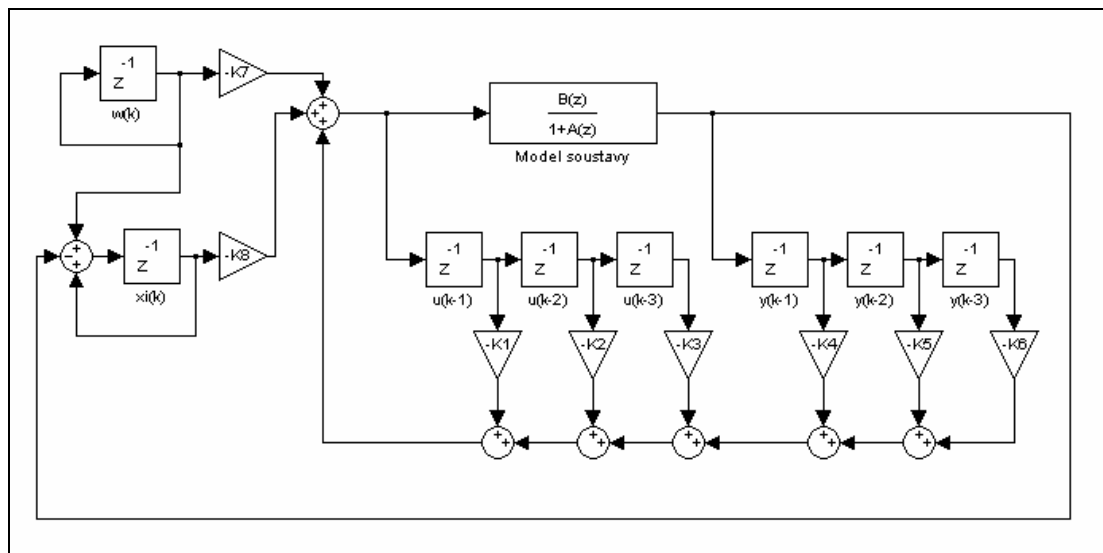
$$J = \sum_{k=0}^{\infty} (q_e (w_k - y_k)^2 + q_u u_k^2 + q_i x_i^2) \quad (6.32)$$

$$J = \sum_{k=0}^{\infty} \left(\begin{pmatrix} z(k) \\ w(k) \\ x_i(k) \end{pmatrix}^T (M^T Q M) \begin{pmatrix} z(k) \\ w(k) \\ x_i(k) \end{pmatrix} \right) \quad (6.33)$$

$$M = \begin{pmatrix} -K_1 & -K_2 & -K_3 & -K_4 & -K_5 & -K_6 & -K_7 & -K_8 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ b_1 & b_2 & b_3 & -a_1 & -a_2 & -a_3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -b_1 & -b_2 & -b_3 & a_1 & a_2 & a_3 & 1 & 1 \end{pmatrix} \quad (6.34)$$

$$Q = \begin{pmatrix} q_u & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_e & 0 & 0 & -q_e & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -q_e & 0 & 0 & q_e & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_i \end{pmatrix} \quad (6.35)$$

Modelovací schéma této struktury je zobrazeno na obr. 6.3.



Obr. 6.4 Modelovací schéma LQ řízení s asymptotickým sledováním referenční trajektorie

6.8 LQ ŘÍZENÍ S DIFERENČNÍ SLOŽKOU

Výše popsané modifikace klasického LQ řízení jsou nezbytné pro dobrý průběh regulace a rozmanitost použití LQ řízení. LQ řízení s asymptotickým sledováním referenční trajektorie si také lze představit jako diskrétní variantu PI regulátoru, který svým výstupem působí na regulovanou soustavu, jejíž všechny póly jsou posunuty prostřednictvím zpětnovazebních zesílení. Použití diskrétní PI

struktury však často přináší zpomalení přechodného děje a větší kmitavost. Tento problém je podrobně popsán například v [2]. U klasického přístupu k regulaci se tento nepříznivý jev eliminuje rozšířením struktury PI na PID. Tato kapitola se ubírá stejnou cestou a popisuje rozšíření struktury LQ řízení s asymptotickým sledováním referenční trajektorie o diferenční složku.

Nejjednodušší úpravou je použití čistě diferenční složky od výstupu, se zesílením K_d , jejíž výstup $y_d(k)$ je popsán vztahem (6.36).

$$y_d(k) = K_d(y(k) - y(k-1)) \quad (6.36)$$

Problém čisté difference spočívá v zesilování šumu v okolí řídicího systému a regulované soustavy. Tento neduh lze eliminovat přidáním filtru, který shora omezí frekvenční rozsah diferenční složky a sníží množství šumu přiváděného do regulované soustavy. Filtraci diferenční složky jsou věnovány pasáže například v [2], [4]. Problém použití filtrace derivační složky v LQ řízení představuje potřeba nastavit vlastnosti filtru dopředu.

V této kapitole jsou popsány dvě varianty diferenční složky a to varianta s čistou diferencí od výstupu regulované soustavy a varianta s pevně nastavenou časovou konstantou filtru.

6.8.1 LQ řízení doplněné o čistou diferencí výstupu

LQ řízení doplněné o čistou diferencí výstupu představuje nejjednodušší variantu rozšíření. Tato varianta, kterou lze použít pouze se stavovou reprezentací, založenou na předchozích hodnotách vstupu a výstupu, vychází z běžné difference odchylky a uvažuje neměnnou žádanou hodnotu. Toto tvrzení popisuje vztah (6.37)

$$\begin{aligned} y_d &= K_d[(w(k) - y(k)) - (w(k-1) - y(k-1))] = \\ &= K_d[y(k-1) - y(k)] \end{aligned} \quad (6.37)$$

kde K_d představuje zesílení diferenční složky, w je žádaná hodnota, y představuje výstup soustavy a k vyjadřuje diskrétní čas.

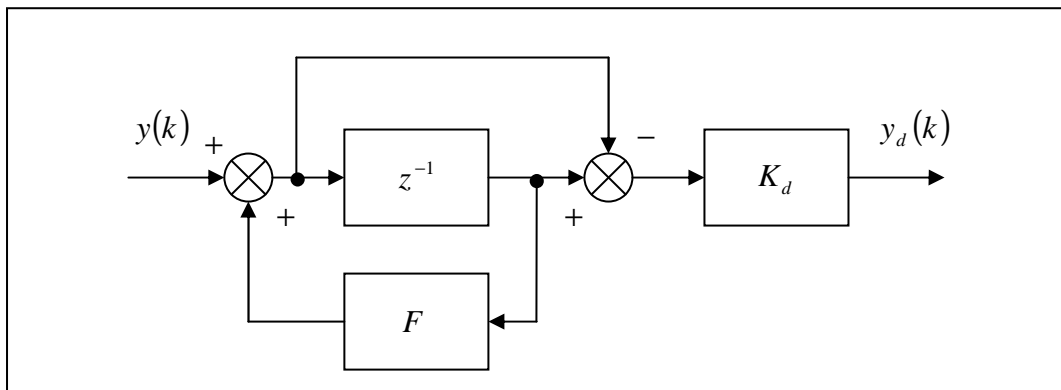
Tato modifikace nevyžaduje rozšíření dimenze stavového prostoru oproti variantě popsané v kapitole 6.6.2. Stavové rovnice jsou tedy popsány vztahem (6.31), kritérium lze zapsat ve tvaru (6.38), které lze přepsat na tvar (6.33) a matice M má tvar (6.34). Jedinou změnu představuje matice Q , která respektuje vztah (6.36) a má tvar (6.39).

$$J = \sum_{k=0}^{\infty} (q_e (w_k - y_k)^2 + q_u u_k^2 + q_i x_i^2 + q_d (y_{k-1} - y_k)^2) \quad (6.38)$$

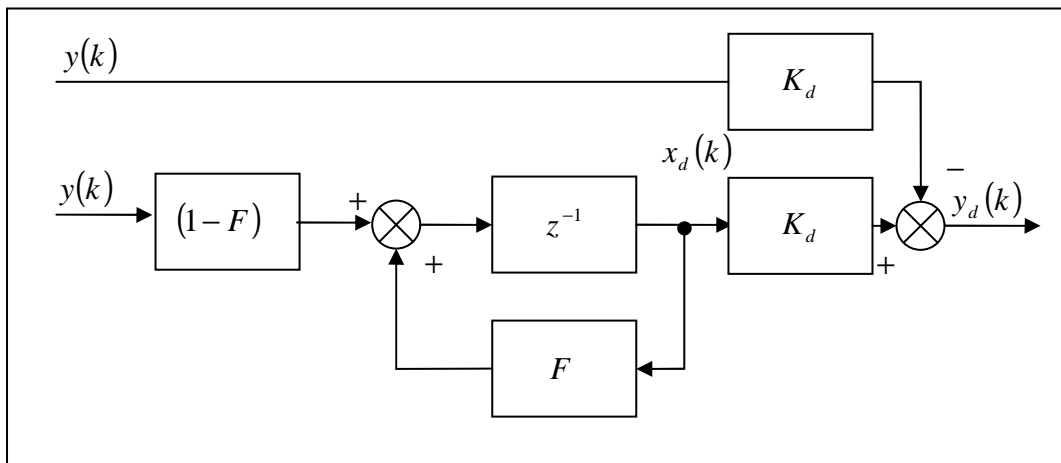
$$Q = \begin{pmatrix} q_u & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_e + q_d & -q_d & 0 & -q_e & 0 \\ 0 & 0 & 0 & -q_d & q_d & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -q_e & 0 & 0 & q_e & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_i \end{pmatrix} \quad (6.39)$$

6.8.2 LQ řízení doplněné o diferenci výstupu s pevně nastaveným filtrem

Čistá diferenční složka má velkou nevýhodu v tom, že zesiluje šum z okolí řídicího systému a z řízené soustavy. Z tohoto důvodu je u PSD regulátorů realizována filtrace diferenční složky. Tato sekce popisuje stejný postup pro použití v LQ řízení. Stavový portrét diferenční složky zobrazuje obr 6.4. Tento portrét lze překreslit na obr 6.5.



Obr. 6.5 Diferenční složka od výstupu soustavy



Obr. 6.6 Jiné zobrazení diferenční složky od výstupu soustavy

Protože horní větev z obr 6.5 je součástí zesílení z předchozích variant, stačí provést rozšíření stavového vektoru o výstup z filtru, jehož koeficient F má uživatelem nastavenou hodnotu z intervalu $F \in (0;1)$.

Chování filtru lze popsat stavovými rovnicemi (6.40)

$$\begin{aligned} x_d(k+1) &= (1-F)y(k) + Fx_d(k) \\ y_f(k) &= K_d x_d(k) \end{aligned} \quad (6.40)$$

kde x_d je stavová proměnná filtru, y je výstup soustavy, y_f představuje výstup filtru a K_d označuje zesílení diferenční složky.

Stavové rovnice celého systému přechází na tvar (6.41)

$$\begin{pmatrix} z(k+1) \\ w(k+1) \\ x_i(k+1) \\ x_d(k+1) \end{pmatrix} = \begin{pmatrix} A' & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -C' & 1 & 1 & 0 \\ (1-F)C' & 0 & 0 & F \end{pmatrix} \begin{pmatrix} z(k) \\ w(k) \\ x_i(k) \\ x_d(k) \end{pmatrix} + \begin{pmatrix} B' \\ 0 \\ 0 \\ 0 \end{pmatrix} u(k) \quad (6.41)$$

$$y(k) = (C' \ 0 \ 0 \ 0) \begin{pmatrix} z(k) \\ w(k) \\ x_i(k) \\ x_d(k) \end{pmatrix}$$

Kritérium lze definovat vztahem (6.42), který lze přepsat na tvar (6.43).

Matice M má tvar (6.44) a matice Q je ve tvaru (6.45).

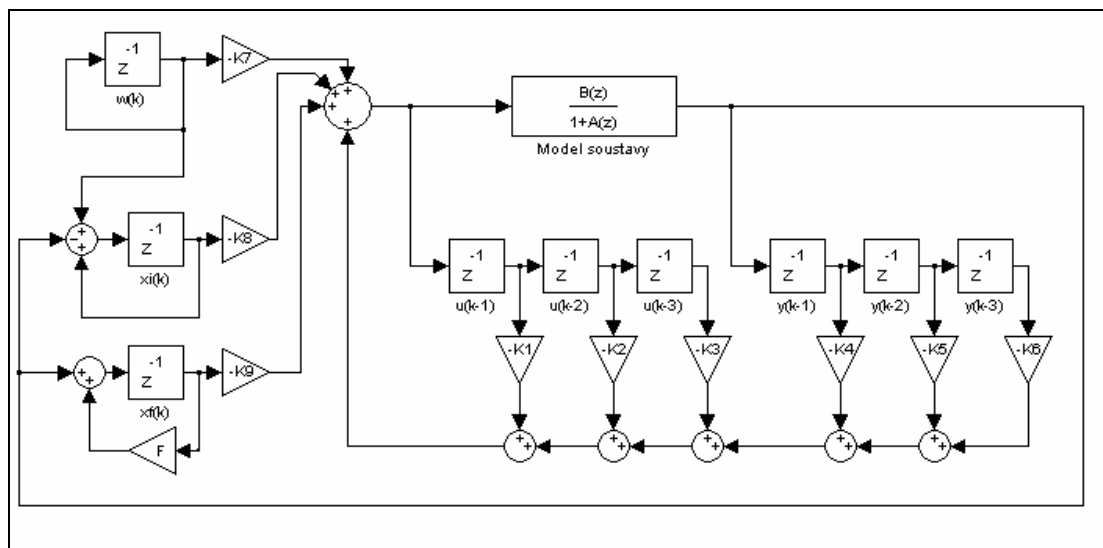
$$J = \sum_{k=0}^{\infty} (q_e (w_k - y_k)^2 + q_u u_k^2 + q_i x_i^2 + q_d (x_{dk} - y_k)^2) \quad (6.42)$$

$$J = \sum_{k=0}^{\infty} \begin{pmatrix} z(k) \\ w(k) \\ x_i(k) \\ x_d(k) \end{pmatrix}^T (M^T Q M) \begin{pmatrix} z(k) \\ w(k) \\ x_i(k) \\ x_d(k) \end{pmatrix} \quad (6.43)$$

$$M = \begin{pmatrix} -K_1 & -K_2 & -K_3 & -K_4 & -K_5 & -K_6 & -K_7 & -K_8 & -K_9 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ b_1 & b_2 & b_3 & -a_1 & -a_2 & -a_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -b_1 & -b_2 & -b_3 & a_1 & a_2 & a_3 & 1 & 1 & 0 \\ (1-F)b_1 & (1-F)b_2 & (1-F)b_3 & -(1-F)a_1 & -(1-F)a_2 & -(1-F)a_3 & 0 & 0 & F \end{pmatrix} \quad (6.44)$$

$$Q = \begin{pmatrix} q_u & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_e + q_d & 0 & 0 & -q_e & 0 & -q_d \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -q_e & 0 & 0 & q_e & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_i & 0 \\ 0 & 0 & 0 & -q_d & 0 & 0 & 0 & 0 & q_d \end{pmatrix} \quad (6.45)$$

Modelovací schéma této varianty je zobrazeno na obr 6.7.



Obr. 6.7 Modelovací schéma LQ řízení s filtrací diferenční složky

6.8.3 LQG řízení doplněné o diferenci výstupu s pevně nastaveným filtrem

Variantu s filtrací diferenční složky lze použít i ve spojení se stavovým rekonstruktorem. Rovnice pro tvar filtru diferenční složky, který je třeba přidat do stavových rovnic výsledného systému, jsou shodné s rovnicemi (6.40). Stavové rovnice mají tvar (6.46). Kritérium je totožné s rovnicí (6.42) a lze ho přepsat na tvar (6.43). Rozdíl oproti předchozí variantě, kromě tvaru stavových rovnic, spočívá

v maticích Q (6.47) a M (6.48). Matice M podle (6.49) je platná pro soustavu třetího řádu při použití Frobeniova kanonického tvaru. Čtvercová matice Q má totožnou hodnotu s počtem proměnných v kritériu jako jsou například žádaná hodnota, výstup soustavy, atd., a matice M má takový tvar, aby po jejím vynásobení se stavovým vektorem byl výsledkem vektor, který výše uvedené proměnné kritéria obsahuje.

$$\begin{pmatrix} x(k+1) \\ w(k+1) \\ x_i(k+1) \\ x_d(k+1) \end{pmatrix} = \begin{pmatrix} A & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -C & 1 & 1 & 0 \\ (1-F)C & 0 & 0 & F \end{pmatrix} \begin{pmatrix} x(k) \\ w(k) \\ x_i(k) \\ x_d(k) \end{pmatrix} + \begin{pmatrix} B \\ 0 \\ 0 \\ 0 \end{pmatrix} u(k) \quad (6.46)$$

$$y(k) = \begin{pmatrix} C & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x(k) \\ w(k) \\ x_i(k) \\ x_d(k) \end{pmatrix}$$

$$Q = \begin{pmatrix} q_u & 0 & 0 & 0 & 0 \\ 0 & q_w + q_d & -q_w & 0 & -q_d \\ 0 & -q_w & q_w & 0 & 0 \\ 0 & 0 & 0 & q_i & 0 \\ 0 & -q_d & 0 & 0 & q_d \end{pmatrix} \quad (6.47)$$

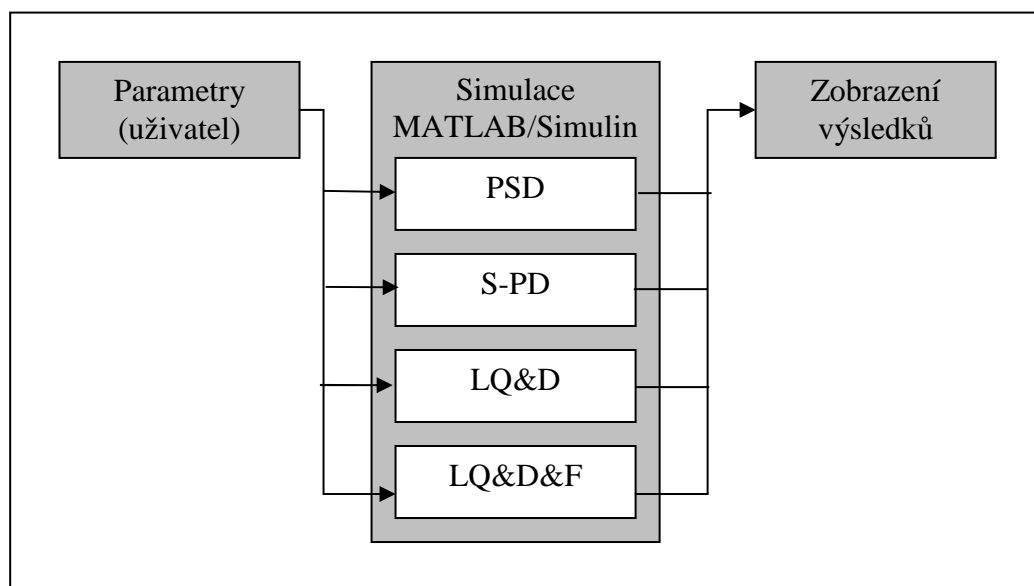
$$M^T = \begin{pmatrix} C & \vdots & \vdots & \vdots \\ -K & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (6.48)$$

$$M^T = \begin{pmatrix} -K_1 & b_1 & 0 & 0 & 0 \\ -K_2 & b_2 & 0 & 0 & 0 \\ -K_3 & b_3 & 0 & 0 & 0 \\ -K_4 & 0 & 1 & 0 & 0 \\ -K_5 & 0 & 0 & 1 & 0 \\ -K_6 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (6.49)$$

7. DEMONSTRAČNÍ PROGRAM ŘÍZENÍ

7.1 DEMONSTRAČNÍ PROGRAM SIMULACE LQ

Pro pohodlné ověření funkce výše popsaných řídicích algoritmů byla vytvořena aplikace SimulaceLQ v prostředí MATLAB 2009b. Ta umožňuje nastavení všech potřebných parametrů pohodlně z jednoho místa. Aplikace se skládá ze dvou základních částí. Jedna z nich je tvořena modely, které jsou vytvořeny v prostředí MATLAB/Simulink, druhá představuje uživatelské rozhraní. Ideové schéma činnosti je zobrazeno na obr. 7.1.



Obr. 7.1 Blokové schéma činnosti simulační aplikace

Nejprve je třeba, aby uživatel zvolil požadované druhy řízení a identifikace a provedl nastavení potřebných parametrů. Následně jsou tyto parametry předány modelům v Simulinku, ve kterém proběhne samotná simulace a následně předání požadovaných parametrů zpět do uživatelského rozhraní, ze kterého jsou výsledky zobrazeny. Pro každý druh řízení byl vytvořen zvláštní model. Aby bylo možno ověřit možnosti a kvalitu identifikace i při buzení vstupu soustavy z PSD a S-PD regulátorů, byly identifikační bloky přidány i do modelů s pevně nastavenými regulátory. Simulaci lze provést ve dvou režimech, a to s matematickým modelem

soustavy, nebo s fyzikální soustavou připojenou k řídicímu systému firmy B&R. Z něho je možno prostřednictvím mk_pvi klienta spojit proměnné řídicího systému a Simulinku. Pro správnou funkci komunikace mezi prostředím MATLAB/Simulink je třeba mít mk_pvi klienta správně nastaveného. Toto nastavení je třeba provést mimo popisovanou aplikaci. Pro možnost volby mezi fyzikální soustavou a matematickým modelem soustavy byly vytvořeny dvě sady modelů, které se liší právě blokem regulované soustavy. Tab 7.1 obsahuje přehled jednotlivých modelů. Během jedné simulace není možné kombinovat pro různá řízení matematický model a fyzikální soustavu.

Název souboru	Implementované řízení	Řízená soustava	Reprezentace soustavy
PSDMdltest.mdl	PSD	Matematický model	-
PSDMdltestF.mdl	PSD	Fyzikální soustava	-
S_PDMdltest.mdl	S-PD	Matematický model	-
S_PDMdltestF.mdl	S-PD	Fyzikální soustava	-
LQDMdltest.mdl	LQ&D	Matematický model	Zpožděné vstupy a výstupy
LQDMdltestF.mdl	LQ&D	Fyzikální soustava	Zpožděné vstupy a výstupy
LQDFMdltest.mdl	LQ&D&F	Matematický model	Zpožděné vstupy a výstupy
LQDFMdltestF.mdl	LQ&D&F	Fyzikální soustava	Zpožděné vstupy a výstupy
LQDFMdltest2.mdl	LQ&D&F	Matematický model	Stavový rekonstruktor
LQDFMdltest2F.mdl	LQ&D&F	Fyzikální soustava	Stavový rekonstruktor

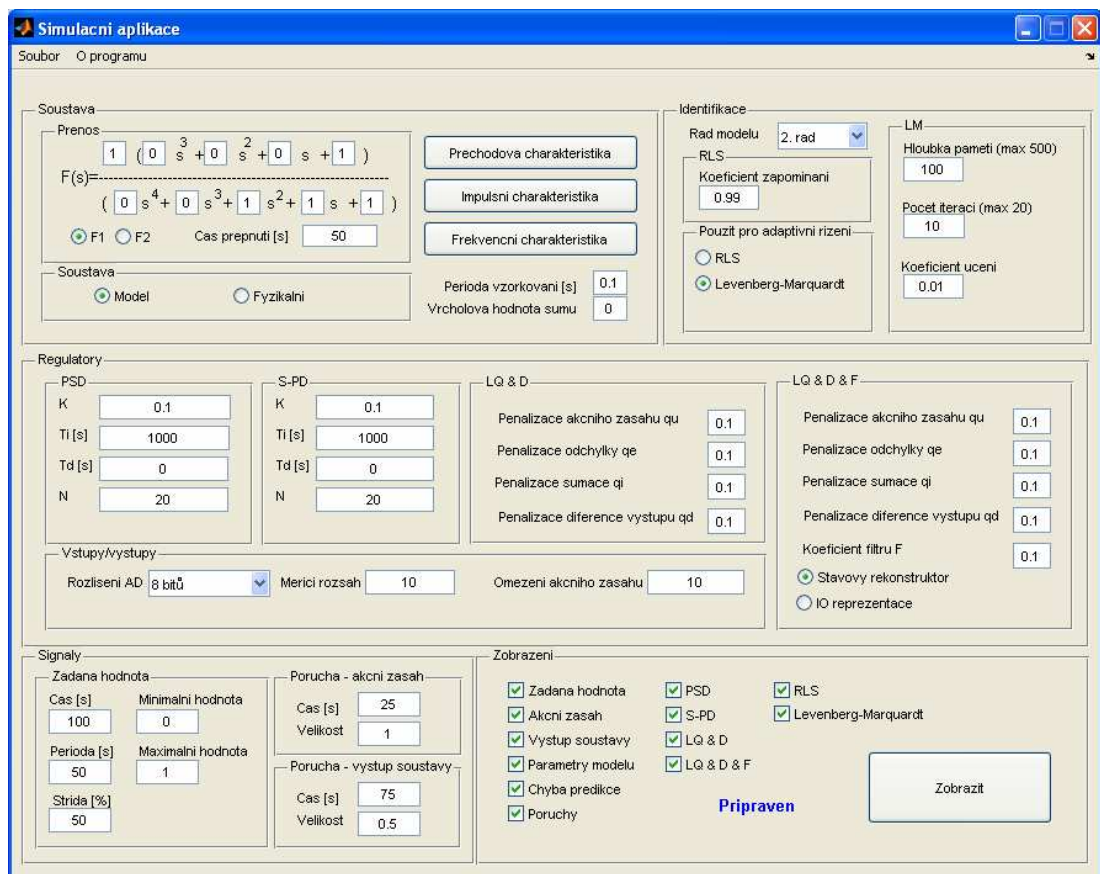
Tab. 7.1 Přehled vytvořených modelů

Stavový portrét matematického modelu řízené soustavy byl vytvořen přímým programováním. Tento model umožňuje na rozdíl od předdefinovaných bloků soustav, do kterých vstupuje operátorový přenos stejný stav systému při změně parametrů soustavy. Pokud by byly použity dvě soustavy, mezi kterými by se přepínalo, mohl by rozdílný stav obou soustav vyvolat přechodný děj, který není reálný. Tento přechodný děj by mohl způsobit falešně lepší simulační výsledky, neboť právě přechodný děj vytváří vhodná data pro identifikaci řízené soustavy.

Pro adaptivní řízení je vždy uživatelem vybrána pouze jedna varianta identifikace. Pro zobrazení průběhu identifikace se pak použijí hodnoty z poslední

simulace. Simulace, pokud jsou použity všechny algoritmy řízení, probíhají v tomto pořadí: PSD, S-PD, LQ&D, LQ&D&F. Identifikační i regulační algoritmy byly vytvořeny jako s-funkce v jazyce ANSI C z důvodu rychlosti simulace a snadného implementování algoritmů do programovatelného automatu firmy B&R. Všechny adaptivní regulátory startují svoji činnost až po čtyřiceti krocích, po kterých je již k dispozici dostatečně přesný model získaný identifikací.

Pro spuštění programu SimulaceLQ je třeba nejprve nastavit v prostředí MATLAB Current Folder na adresář, ve kterém se aplikace nachází. Následně lze do Command Windows zadat simulaceLQ. Po stisku klávesy Enter se spustí aplikace. Na obr 7.2 se nachází hlavní formulář této aplikace.



SimulaceLQ - Hlavní formulář

Soustava

Prenos: $F(s) = \frac{1 \cdot (0s^3 + 0s^2 + 0s + 1)}{(0s^4 + 0s^3 + 1s^2 + 1s + 1)}$

Prechodova charakteristika
Impulsní charakteristika
Frekvencni charakteristika

Soustava: Model Fyzikalni

Perioda vzorkovani [s]: 0.1
Vrcholova hodnota sumy: 0

Identifikace

Rad modelu: 2. rad
RLS: Koeficient zapominani: 0.99
LM: Hloubka pameti (max 500): 100
Pocet iteraci (max 20): 10
Koeficient uceni: 0.01

Pouzit pro adaptivni rizeni: RLS Levenberg-Marquardt

Regulatory

PSD: K: 0.1, Ti [s]: 1000, Td [s]: 0, N: 20
S-PD: K: 0.1, Ti [s]: 1000, Td [s]: 0, N: 20
LQ & D: Penalizace akcniho zasahu q_u: 0.1, Penalizace odchylky q_e: 0.1, Penalizace sumace q_i: 0.1, Penalizace difference vystupu q_d: 0.1
LQ & D & F: Penalizace akcniho zasahu q_u: 0.1, Penalizace odchylky q_e: 0.1, Penalizace sumace q_i: 0.1, Penalizace difference vystupu q_d: 0.1, Koeficient filtru F: 0.1

Vstupy/vystupy: Rozliseni AD: 8 bitů, Merici rozsah: 10, Omezeni akcniho zasahu: 10
 Stavovy rekonstruktor IO reprezentace

Signaly

Zadana hodnota: Cas [s]: 100, Minimalni hodnota: 0, Perioda [s]: 50, Maximalni hodnota: 1, Strida [%]: 50
Porucha - akcni zasah: Cas [s]: 25, Velikost: 1
Porucha - vystup soustavy: Cas [s]: 75, Velikost: 0.5

Zobrazeni

Zadana hodnota PSD RLS
 Akcni zasah S-PD Levenberg-Marquardt
 Vystup soustavy LQ & D
 Parametry modelu LQ & D & F
 Chyba predikce
 Poruchy

Pripraven

Obr. 7.2 Hlavní formulář simulační aplikace

Hlavní okno je rozděleno do několika částí. První panel je označen soustava. V něm je možno provést výše popsanou volbu mezi matematickým modelem

soustavy a soustavou fyzikální. Pokud se uživatel rozhodne pro matematický model, dojde k aktivaci panelu označeného přenos. Ten slouží k zadání parametrů dvou spojitých soustav. Přepínači F1 a F2 lze mezi nimi přepínat. Protože byl matematický model realizován tak, aby po přepnutí soustav byl zachován stejný počáteční stav, bylo zde přidáno omezení, že soustava, na kterou se bude přepínat, musí být stejného řádu jako soustava, ze které se přepíná. Aplikace také hlídá okamžitou realizovatelnost systému. Proto je nejlepším způsobem zadání zadat nejprve koeficienty jmenovatele původní soustavy (aktivní přepínač F1), následně pak její koeficienty čitatele, a teprve poté zadávat koeficienty cílové soustavy (aktivní přepínač F2). Obě soustavy mohou být až čtvrtého řádu. Pro zadání byl zvolen roznásobený tvar, protože ve tvaru jednotlivých kořenů by bylo obtížnější uhlídat správné zadání (komplexně sdružené páry) a hlavně by nebylo možno simulovat soustavu, která obsahuje astatismus. V panelu „Soustava“ je možno zadat také periodu vzorkování a vrcholovou hodnotu šumu, který se přičítá k výstupu soustavy. Posledním parametrem, který lze v tomto panelu nastavit, je čas přepnutí mezi matematickými modely regulovaných soustav.

Další panel se týká identifikace. Před zahájením simulace je zde možno vybrat, zda se pro adaptivní řízení (pouze pro regulátory LQ) bude používat identifikace metodou nejmenších čtverců nebo neuronovou sítí a provést nastavení všech potřebných parametrů identifikace, jako je koeficient zapomínání, koeficient učení a hloubka paměti. Dále je zde možno zadat řád modelu. Pro identifikaci neuronovou sítí byla implementována metoda Levenberg-Marquardt.

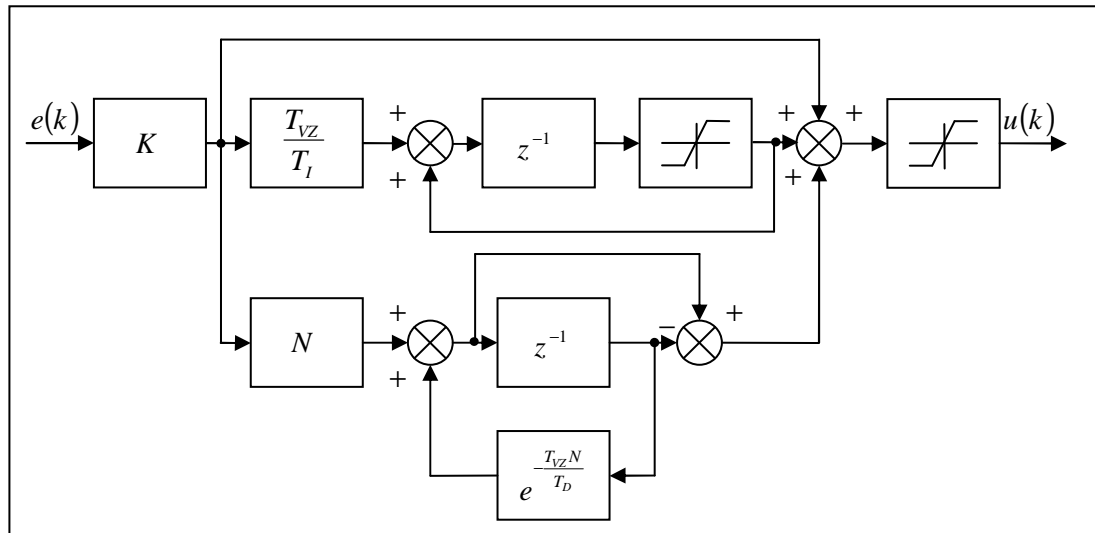
Uprostřed okna se nachází panel „Regulátory“, který je určen pro zadávání parametrů regulátorů. Prostřednictvím něho lze zadat parametry napevno nastavených regulátorů PSD a S-PD, jejichž konkrétní tvar je uveden konci této kapitoly. Prostřednictvím dalších dvou panelů lze zadat penalizační koeficienty LQ regulátorů. První je označen jako LQ&D a představuje variantu popsanou v kapitole 6.7.1 (LQ řízení s asymptotickým sledováním referenční trajektorie doplněné o čistou diferenci výstupu soustavy). Vynecháním některého koeficientu lze tuto variantu zredukovat na LQ řízení s asymptotickým sledováním referenční trajektorie nebo na variantu bez přidané integrace (LQ řízení se sledováním referenční

trajektorie). Druhý panel je označen LQ&D&F a představuje variantu s filtrací diferenční složky od výstupu popsanou v kapitole 6.7.2. Zde je třeba si uvědomit, že pro použití čisté diferenční složky není tato varianta vhodná, neboť použitím Kleinmanova iteračního algoritmu by bylo zesílení od zpoždovacího členu, který není obklíčen zpětnou vazbou, nulové. V poslední zmíněné variantě má uživatel možnost vybrat, zda bude použit adaptivní stavový rekonstruktor nebo reprezentace systému založená na předchozích hodnotách vstupu a výstupu. Pro zvýšení věrohodnosti simulačních výsledků (při použití matematického modelu regulované soustavy) má uživatel možnost zvolit parametry simulovaných AD a DA převodníků. Jedná se o parametry rozlišení převodníků, měřicí rozsah AD převodníku a rozsah DA převodníku. Oba zmíněné rozsahy jsou symetrické kolem nuly. Rozlišení převodníků lze zvolit osm, deset, dvanáct a šestnáct bitů.

Ve spodní části okna jsou umístěny další dva panely. První obsahuje prvky pro nastavení vstupních signálů. Žádaná hodnota má vždy obdélníkový průběh, u kterého je možno nastavit minimální a maximální hodnotu, periodu a střidu. Dále je zde možno nastavit poruchové signály působící na vstupu a výstupu regulované soustavy. Ty mají vždy tvar skoku, u kterého je možno nastavit čas, ve kterém skok proběhne, a cílovou hodnotu. V posledním panelu lze zvolit signály, které mají být zobrazeny, regulátory, které mají být postupně použity a identifikační algoritmy, jejichž výsledky mají být zobrazeny. Tento panel obsahuje také tlačítko Zobrazit, které slouží k předání zadaných parametrů do příslušných modelů, spuštění jejich simulace a zobrazení výsledků. Pro zobrazení výsledků jsou otevřena dvě okna s požadovanými průběhy. První se týká samotného řízení, druhé je věnováno identifikaci.

7.2 POUŽITÉ VARIANTY DISKRÉTNÍCH EKVIVALENTŮ PID REGULÁTORŮ

PID jsou, pro svoji robustnost a jednoduchost, dnes nejznámější a nejpoužívanější regulátory. Jednou z možností, jak implementovat původně spojitý regulátor do číslicového řídicího systému, bylo použití ekvivalentu v podobě PSD regulátoru. Obr. 7.3 zobrazuje stavový portrét implementovaného PSD regulátoru.

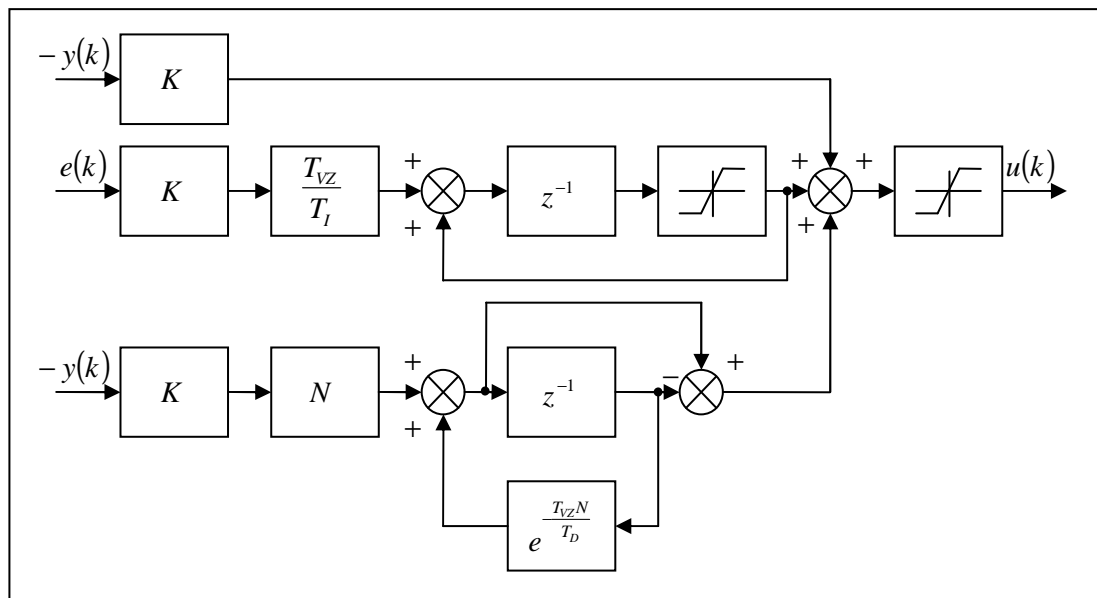


Obr. 7.3 Stavový portrét použitého PSD regulátoru

Přechod mezi PID a PSD regulátorem představuje nahrazení integrační složky složkou sumační a derivační složky složkou diferenční. Diferenční složka je na rozdíl od složky derivační realizovatelná bez problému. Použití čisté diference však není žádoucí, protože by vedlo k zesilování vysokofrekvenčního šumu [2]. Do regulátoru tedy lze přidat její filtraci, jež připomíná realizační filtr původní derivační složky. Další úprava se týká sumační složky. Ta byla rozšířena o nelinearitu typu nasycení, která realizuje omezení rozsahu hodnot, kterých může výstup sumační složky nabývat (anti-windup). Poslední úprava se týká výstupu regulátoru, který byl také doplněn o nelinearitu typu nasycení. Ta představuje omezený akční zásah. Rozsahy obou nelinearit (v demonstračním programu SimulaceLQ) je možno nastavit prostřednictvím položky saturace. Více o PSD regulátorech lze nalézt například v [2].

Druhá implementovaná varianta nese označení S-PD. Jedná se o jednu z variant, které vznikly ve snaze snížit překmit přechodného děje. Od PSD regulátoru se liší svou strukturou. Ta je upravena tak, že regulační odchylka vstupuje pouze do sumačního členu. Proporcionální a diferenční složka je připojena pouze na výstup regulované soustavy [2]. Stavový portrét je zobrazen na obr. 7.4. Parametry obou

nasycení lze v programu SimulaceLQ nastavit zcela shodným způsobem jako u regulátoru PSD. Pro obě varianty představuje proměnná K zesílení regulátoru, T_I časovou konstantu sumačního členu, T_D časovou konstantu diferenčního členu, N filtrační koeficient a T_{VZ} periodu vzorkování.

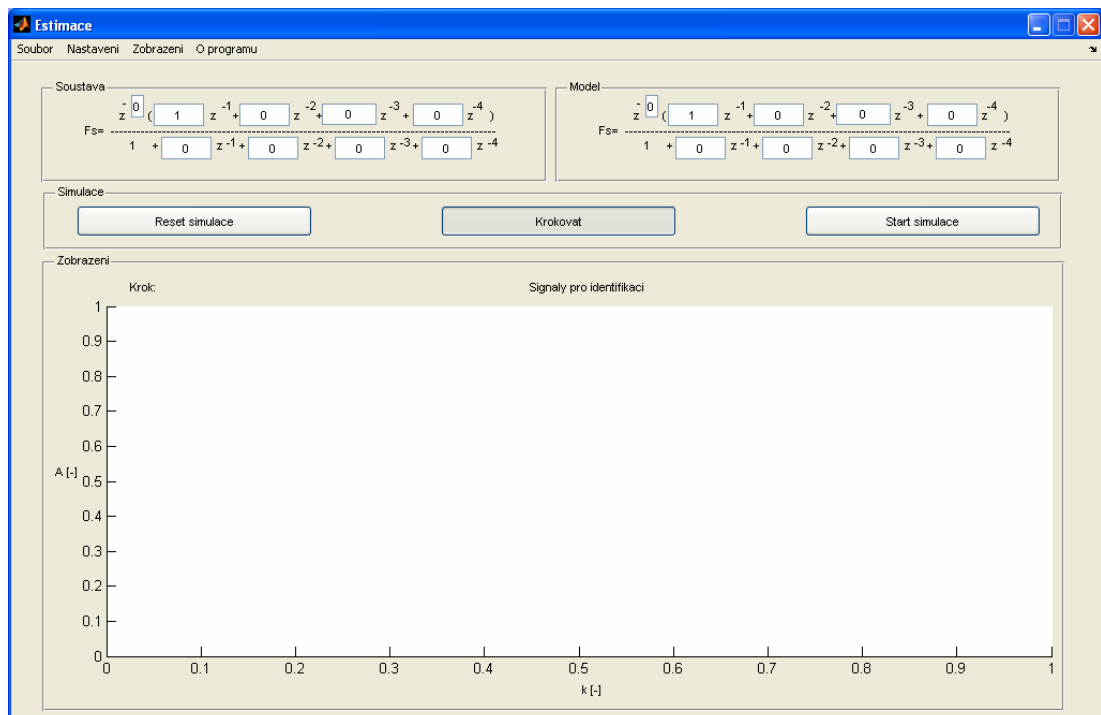


Obr. 7.4 Stavový portrét S-PD regulátoru

8. DEMONSTRAČNÍ PROGRAM ESTIMACE

Dostatečně přesný model regulované soustavy je nezbytný pro bezpečné a optimální adaptivní řízení. Je tedy třeba co nejvíce vědět o vlastnostech a možnostech identifikačních algoritmů. Z tohoto důvodu byl vytvořen demonstrační program Estimace, který umožňuje zobrazení vývoje matematického modelu a vnitřních proměnných identifikačních algoritmů v čase. Tento program byl vytvořen v prostředí MATLAB 2009b. Byla zde implementována rekurzivní metoda nejmenších čtverců a umělá neuronová síť s jedním neuronem a učícím algoritmem Levenberg-Marquardt. Simulaci je možno spustit v cyklickém režimu, nebo ji krokovat.

Pro spuštění programu Estimace je třeba nejprve nastavit v prostředí MATLAB Current Folder na adresář, ve kterém se aplikace nachází. Následně lze do Command Windows zadat Estimace. Po stisku klávesy Enter se spustí aplikace. Obr. 8.1 zobrazuje hlavní formulář aplikace po jejím spuštění.



Obr. 8.1 Hlavní formulář demonstrační aplikace Estimace

Ten je rozdělen na čtyři panely. První panel označený jako soustava umožňuje zadat operátorový přenos identifikované soustavy v z-transformaci. Lze zadat soustavu, která má až čtyři koeficienty v čitateli a jmenovateli. Dále je možno zadat libovolné dopravní zpoždění o velikosti celočíselného násobku vzorkovací periody.

Panel, který nese název model, slouží k zobrazení koeficientů současného matematického modelu.

Pod těmito panely se nachází oblast označená „simulace“. Zde se nachází tři tlačítka. První tlačítko zleva, které je označeno „Reset simulace“, slouží k nastavení všech parametrů na výchozí hodnoty. Tuto akci lze provést pouze ve stavu, kdy je zamáčknuté tlačítko „Krokovat“. Prostřední tlačítko „Krokovat“ je dvoustavové a jeho stav definuje, zda bude po stisku tlačítka „Start simulace“ proveden pouze jediný simulační cyklus, nebo zda bude simulace prováděna cyklicky. Přepínání mezi cyklickým během simulace a krokování je možno provádět bez nutnosti restartovat simulaci. Možnost krokování umožňuje pozastavení simulace a tedy i detailní prohlédnutí vývoje matematického modelu v zajímavých časech. Při zamáčknutém tlačítku „Krokovat“ lze provést nastavení celé aplikace na původní hodnoty (reset aplikace) nebo aplikaci ukončit. Tato podmínka zde byla doplněna kvůli bezproblémovému ukončení aplikace.

Poslední panel (Zobrazení) obsahuje zobrazovací plochu, ve které jsou zakresleny průběhy podle přání uživatele. Lze zde zobrazit průběhy vstupních signálů a odezvy soustavy a modelu, časový vývoj rozložení nul a pólů modelu, parametry modelu a kovarianční matici. Tato zobrazení je možno použít jak v režimu krokování, tak v režimu cyklické simulace. V režimu krokování lze také zobrazit odezvy modelu a soustavy na jednotkový skok. Zobrazení těchto odezev v cyklickém režimu postrádá smysl. Volba toho, co se bude zobrazovat, se provádí v roletovém menu hlavního formuláře pod položkou Zobrazení.

Další nezbytná nastavení lze provést přes formuláře, které je možno zobrazit přes položku Nastavení v roletovém menu hlavního formuláře.

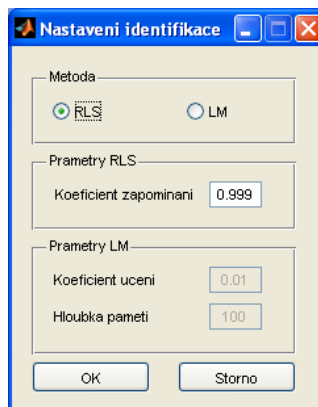
První z těchto formulářů zobrazuje obr. 8.2. Prostřednictvím něho se nastavují parametry modelu. Těmito parametry jsou počet koeficientů čitatele a jmenovatele

operátorového přenosu modelu a počet celočíselných násobků vzorkovací periody, které tvoří dopravní zpoždění. Tato nastavení lze provést pouze po spuštění aplikace nebo po stisku tlačítka Reset simulace.



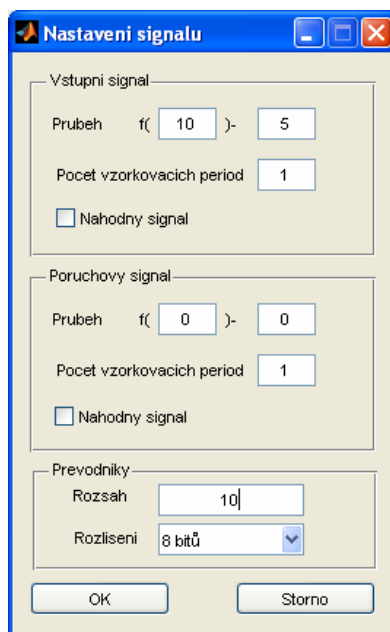
Obr. 8.2 Formulář pro nastavení modelu

Druhý formulář slouží k nastavení identifikace. Tento formulář je zobrazen na obr. 8.3. Skládá se ze třech panelů. První panel slouží k výběru, zda bude probíhat identifikace rekurzivní metodou nejmenších čtverců (RLS) nebo bude použita umělá neuronová síť (LM). Druhý panel je určen pro nastavení koeficientu zapomínání pro metodu RLS. Poslední panel je svázán s identifikací umělou neuronovou sítí. Zde je možno zadat koeficient učení a hloubku paměti. Hloubka paměti představuje jediný parametr, který lze nastavit pouze po staru aplikace nebo po stisknutí tlačítka Restart simulace. Všechny ostatní parametry mohou být změněny za chodu simulace (při stisknutém tlačítku krokovat) a tím pádem je umožněno demonstrovat jejich vliv na nalezený model soustavy.



Obr. 8.3 Formulář pro nastavení identifikace

Třetí formulář, který je zobrazen na obr. 8.4, slouží k nastavení signálů, které působí na soustavu. Je rozdělen do třech panelů pro vstupní signál (akční zásah) neměřitelný poruchový signál, který působí na výstupu soustavy a nastavení parametrů AD a DA převodníků. Oba signály mohou být náhodné nebo obdélníkové s konstantní mezi vrcholovou hodnotou. Náhodné signály jsou generovány generátorem náhodných čísel od nuly do hodnoty parametru v závorkách. U obdélníkového signálu představuje tento parametr mezi vrcholovou hodnotu signálu. Druhý parametr představuje offset, kterým lze nastavit střední hodnotu. Pokud je offset polovinou parametru v závorkách, bude střední hodnota signálu nulová. Poslední parametr představuje počet kroků, ve kterých bude příslušný signál konstantní. Všechny tyto parametry lze měnit za chodu simulace (při stisknutí tlačítka „Krokovat“). Na tomto formuláři se také nachází ovládací prvky pro nastavení simulovaných AD a DA převodníků. Těm lze nastavit rozlišení a rozsah, který je rozdělen podle jejich rozlišení.



Obr. 8.4 Formulář pro nastavení signálů

Poslední formulář pro nastavení umožňuje nastavení počtu kroků zobrazených průběhů. Toto nastavení se týká zobrazení odezev na jednotkový skok

(délka odezvy), zobrazení signálů, které působí na soustavu a odezev soustavy a modelu a zobrazení parametrů modelu (počet předchozích kroků). Při nastavení zobrazení rozložení nul a pólů jsou zobrazovány nuly a póly modelů, které jsou získány až po přepnutí zobrazení. Tím bylo dosaženo možnosti zobrazení rozložení nul a pólů od požadovaného okamžiku. Formulář pro nastavení zobrazení se nachází na obr. 8.5. Nastavení délky odezvy je možno provést i za běhu simulace (při stisknutém tlačítku krokovat).



Obr. 8.5 Formulář pro nastavení zobrazení

9. TESTOVÁNÍ ŘÍZENÍ

Cílem této kapitoly je shrnout výsledky testování všech výše uvedených řídicích algoritmů. Testování bylo rozděleno na dvě hlavní části a to regulaci matematického modelu regulované soustavy a regulaci fyzikální soustavy. V obou případech bylo provedeno ověření vlastností při působení poruchových signálů a při změně dynamických parametrů (časových konstant) regulované soustavy. Závěr kapitoly je věnován standardnímu adaptivnímu regulátoru firmy B&R.

9.1 SIMULAČNÍ VÝSLEDKY ŘÍZENÍ

Pro ověření možností zde zmíněných řídicích algoritmů byl použit demonstrační program SimulaceLQ. Nejprve byla otestována schopnost řídicích algoritmů adaptovat se na změnu dynamických vlastností. Pro testování byly použity dvě soustavy, jejichž operátorové přenosy jsou dány vztahy (9.1) a (9.2).

$$F_1(s) = \frac{1}{(4s+1)(s+1)} \quad (9.1)$$

$$F_2(s) = \frac{1}{(10s+1)(s+1)} \quad (9.2)$$

Jedná se o nekmitavé soustavy druhého řádu, které se liší velikostí dominantní časové konstanty.

Protože bylo třeba co možná nejobjektivněji porovnat vlastnosti různých variant LQ řízení s pevně nastavenými diskrétními ekvivalenty PID regulátorů, bylo pro nastavení všech koeficientů jednotlivých variant řízení použito minimalizace integrálního kritéria. Toto kritérium, které uvádí vztah (9.3), bylo aplikováno na soustavu s operátorovým přenosem ve tvaru (9.1). Pro minimalizaci kritéria (9.3) byla použita simplexová metoda, kterou lze nalézt například v [3]. Jedná se tedy o pseudooptimalizaci parametrů, protože použití simplexové metody nezaručuje nalezení globálního minima.

$$J = d \max_k (h(k)) + \sum_{i=0}^{n-1} (h(i) - 1(i))^2 \quad (9.3)$$

Kritérium (9.3) se skládá ze dvou sčítanců. Druhý sčítanec představuje klasickou diskrétní náhradu kvadratického integrálního kritéria, ve kterém jsou sčítány kvadráty hodnot regulačních odchylek. Ve vztahu (9.3) byl použit jako vstupní signál jednotkový skok $(1(k))$, výstup soustavy potom představuje přechodovou charakteristiku celého systému $h(k)$. Protože všechny navržené varianty LQ řízení měly při použití pouze druhého sčítance vztahu (9.3) jako kritéria tendenci dosáhnout v co nejkratším čase co nejmenší regulační odchylky bez ohledu na překmit, bylo kritérium (9.3) rozšířeno o člen, který penalizuje překmit. Ten je tvořen součinem koeficientu d a maximální hodnoty přechodové charakteristiky $h(k)$. Pro všechny varianty LQ řízení byl koeficient d roven sto. Pro PSD a S-PD regulátor byl tento koeficient snížen na hodnotu jedna, protože při použití hodnoty sto docházelo k velkému prodloužení přechodného děje. Výsledná nastavení všech variant LQ regulátorů obsahuje tab. 9.1. Nastavení pro diskrétní ekvivalenty PID regulátorů jsou uvedeny v tab. 9.2.

Číslo kritéria	Řídicí algoritmus	q_u	q_w	q_i	q_d	F
1	LQ	1,42	1,19	0,27	-	-
2	LQ&D	0,023	2,05	0,001	708	-
3	LQ&D&F	0,76	1,53	0,038	910	0,1
4	LQ&D&F&R	0,062	0,094	0,027	1717	0,1

Tab. 9.1 Hodnoty koeficientů pro simulaci LQ řízení

Řídicí algoritmus	K	Ti [s]	Td [s]	N
PSD	3,74	4,64	0,1	20
S-PD	3,74	3,06	0,1	20

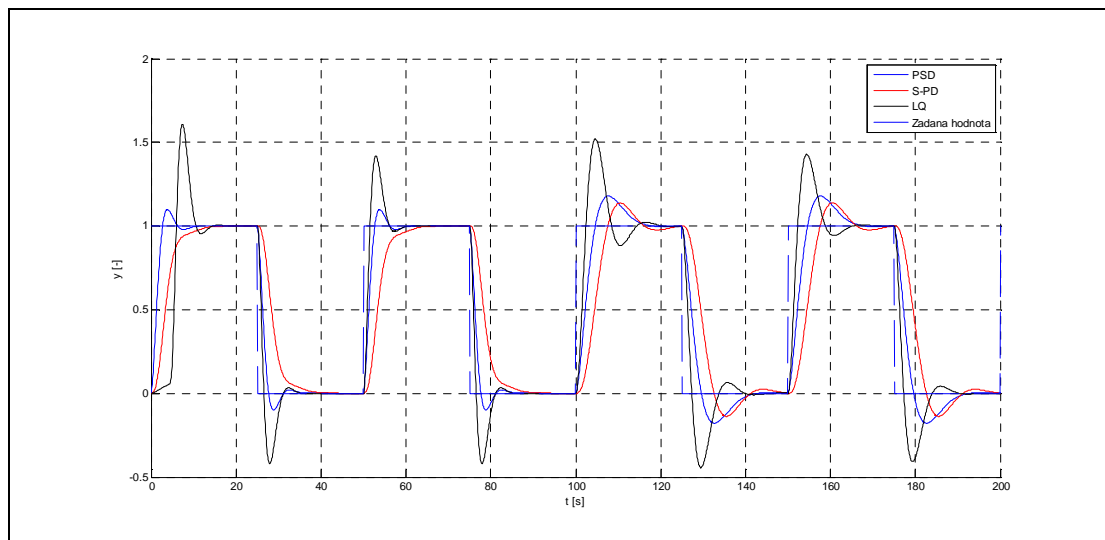
Tab. 9.2 Hodnoty parametrů pro simulaci PSD a S-PD řízení

Z tab. 9.1 je patrný nepoměr koeficientu q_d pro penalizaci difference od výstupu a koeficientů ostatních. Tento nepoměr je způsoben přidáním prvního členu

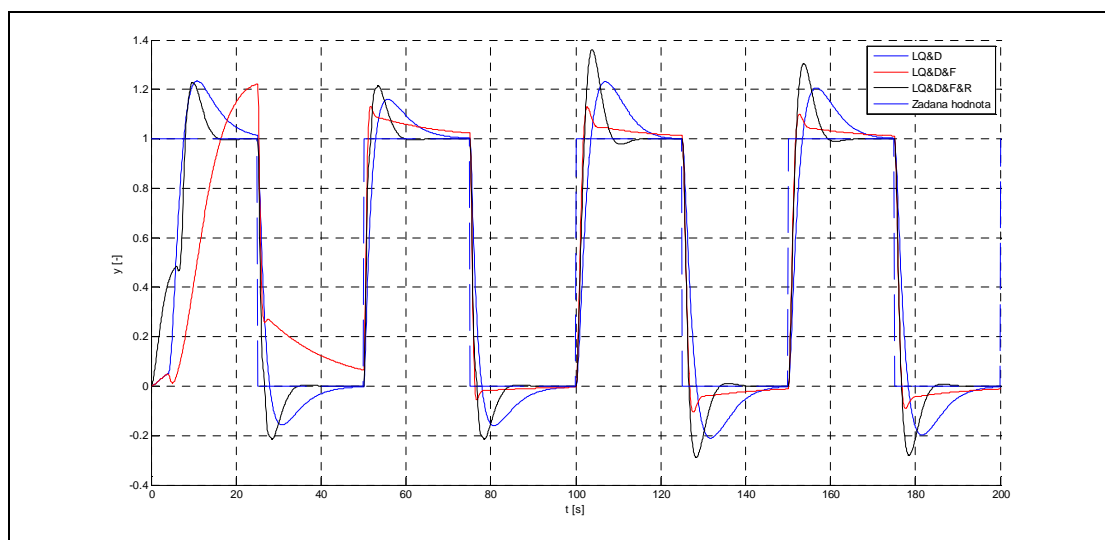
do kritéria (9.3) a značnou penalizací překmitu prostřednictvím koeficientu d . Výslední vliv difference od výstupu na řízení soustavy je také ovlivněn koeficienty q_u a q_i . Pokud nabývá koeficient q_u vyšších hodnot, dochází k velké penalizaci velikosti akčního zásahu a tedy k výraznému potlačení vlivu difference od výstupu soustavy. Vliv koeficientu q_i je zřejmý z faktu, že celá úprava LQ řízení s asymptotickým sledováním referenční trajektorie byla přidána právě kvůli vlivu členu s integračním charakterem. Pokud se tedy zvýší hodnota koeficientu q_i a koeficient q_d zůstane stejný, vzroste vliv integračního členu na úkor difference od výstupu a naopak.

Pro otestování vlastností adaptivního řízení při změně dynamiky soustavy byla zvolena perioda vzorkování 0,1 s, identifikace rekurzivní metodou nejmenších čtverců (RLS) s koeficientem zapomínání 0,999 a modelem druhého řádu bez dopravního zpoždění. Všechny adaptivní modely byly nastaveny tak, aby prvních čtyřicet kroků byl akční zásah dán prostým zesílením regulační odchylky. Tato prodleva slouží k tomu, aby identifikace určila dostatečně přesný matematický model řízené soustavy. Dále přechází adaptivní LQ regulátory do režimu, ve kterém je již využito zpětné vazby od stavu. Krátce po přepnutí dochází k přechodnému ději, protože zpětnovazební zesílení ještě nejsou spočítána a také se ještě mohou značně měnit parametry matematického modelu regulované soustavy. Dále již následuje režim, ve kterém je regulátor optimální vzhledem ke zvolenému kritériu. Obr. 9.1 a obr. 9.2 zobrazují odezvy výsledných systémů při řízení jednotlivými algoritmy. V čase dvě stě sekund došlo k přepnutí modelu regulované soustavy ze soustavy s operátorovým přenosem podle (9.1) na model s operátorovým přenosem podle (9.2). Pro ověření schopnosti regulátoru přizpůsobit se změnám regulované soustavy byly jako ukazatele zvoleny velikosti překmitu před, během a po změně parametrů regulované soustavy a čas, kdy se výstup soustavy poprvé liší od žádané hodnoty o méně jak pět procent a již tuto pětiprocentní hranici nepřekročí. Všechny tyto hodnoty jsou zaznamenány v tab. 9.3. Při pohledu na charakteristiky z obr. 9.2 je patrné, že nastavení penalizačních koeficientů podle kritéria (9.3) vedlo při LQ řízení s čistou diferencí od výstupu (LQ&D) ke značnému prodloužení přechodného děje. Podobně tomu je i u LQ řízení s filtrovanou diferencí od výstupu (LQ&D&F). Zde došlo ke značnému vlivu difference od výstupu na úkor integračního členu, což

způsobilo prodloužení přechodného děje a značné potlačení překmitu. Nejlépe dopadlo použití LQ řízení s filtrovanou diferenční složkou a adaptivním stavovým rekonstruktorem (LQ&D&F&R). Při porovnání s variantou bez difference (LQ) od výstupu na obr 9.1 je patrné, že přidání difference od výstupu umožnilo snížení překmitu výstupu soustavy během přechodného děje bez jeho výrazného prodloužení. Koeficienty LQ řízení, které je na obr. 9.1, byly získány minimalizací kritéria (9.3) a jsou obsaženy na prvním řádku v tab 9.1.



Obr. 9.1 Odezvy výsledného systému při změně dynamiky soustavy



Obr. 9.2 Odezvy výsledného systému při změně dynamiky soustavy

Regulátor	Kritérium	Překmit pro soustavu [%]			Toleranční pásmo $\pm 5\%$ [s]		
		F_1	$F_1 \rightarrow F_2$	F_2	F_1	$F_1 \rightarrow F_2$	F_2
PSD	-	10,0	17,8	17,8	5,1	13,0	13,0
S-PD	-	0,0	13,9	13,9	8,5	14,2	14,2
LQ	1	41,8	52,1	40,7	5,5	13,0	8,0
LQ	2	19,1	26,9	22,7	8,5	11,7	11,7
LQ&D	2	14,7	23,2	19,8	13,0	15,1	14,6
LQ	3	37,4	47,0	36,5	7,1	17,0	10,9
LQ&D&F	3	13,2	12,9	9,2	11,5	5,0	3,3
LQ	4	51,8	66,5	54,7	4,5	9,2	6,5
LQ&D&F&R	4	21,4	36,1	28,0	6,8	7,6	7,2

Tab. 9.3 Souhrn výsledků dosažených při simulaci změny dynamiky soustavy

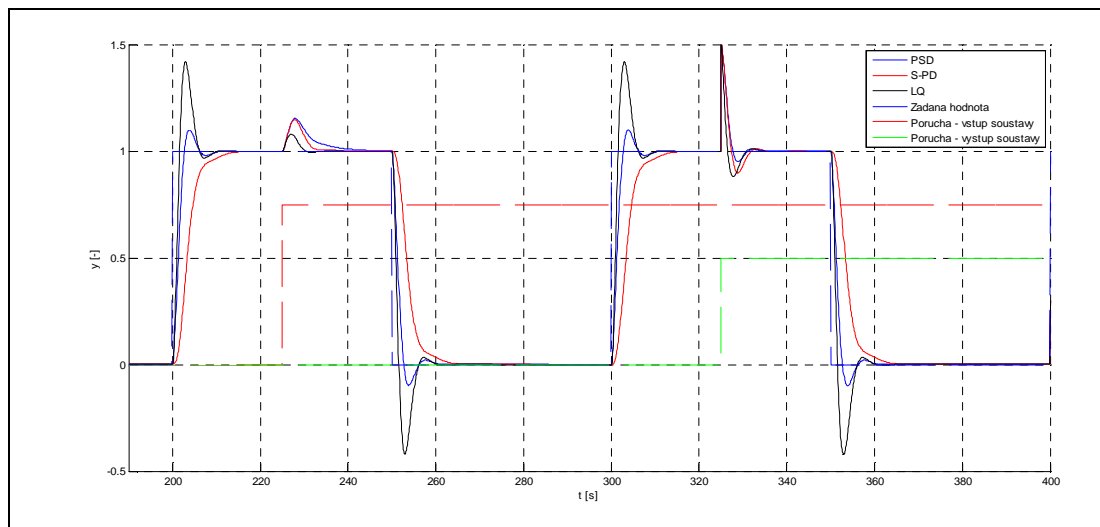
Tab. 9.3 dále obsahuje výsledky dosažené při LQ řízení bez použití diference od výstupu soustavy při volbě stejných penalizačních koeficientů jako u odpovídající varianty s diferencí od výstupu.

Druhým jevem, který si zaslouží pozornost při testování je schopnost vyregulování poruchy, která působí na vstupu a výstupu regulované soustavy. Pro simulaci působení konstantních poruchových signálů a jejich vyregulování byla zvolena stejná nastavení jako v předchozím případě. Operátorový přenos regulované soustavy byl ve tvaru (9.1). Nastavení penalizačních koeficientů je tedy obsaženo v tab. 9.1, parametry diskrétních ekvivalentů PID regulátoru jsou obsaženy v tab. 9.2. Obr. 9.3 a obr. 9.4 zobrazují výstup soustavy při simulaci. Překmit a dosažení pětiprocentního tolerančního pásma od žádané hodnoty při skoku řízení a poruchy, která působí na vstupu soustavy, představují parametry, podle kterých lze usuzovat kvalitu a schopnosti regulace. Tyto parametry jsou obsaženy v tab. 9.4. Reakce na poruchu, která působí na výstupu regulované soustavy, není v tab. 9.4 obsažena protože její překmit je vždy roven právě velikosti působící poruchy. Schopnost jejího vyregulování je tedy demonstrována pouze na obr. 9.3 a obr. 9.4. Pro výsledky simulace při působení poruchových signálů platí stejný závěr jako v případě změny parametrů dynamiky regulované soustavy. Obě varianty, které byly rozšířeny o diferenci od výstupu a využívají stavovou reprezentaci založenou na minulých hodnotách vstupu a výstupu (LQ&D a LQ&D&F) značně prodlužují přechodný děj.

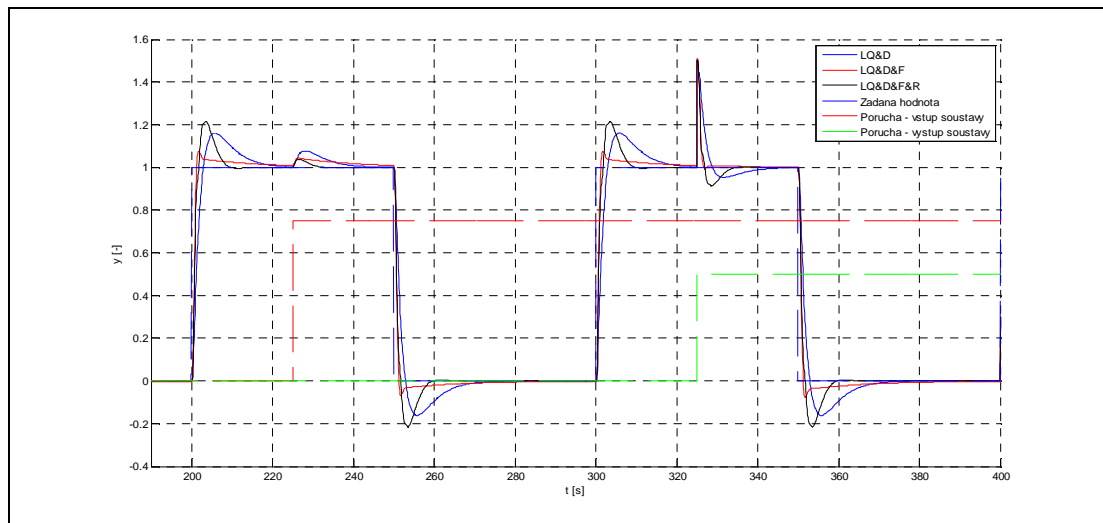
Varianta se stavovým rekonstruktorem (LQ&D&F&R) dosáhla nejlepšího výsledku při reakci na poruchu, která vstupuje na vstupu regulované soustavy a to nejen mezi LQ regulátory ale i při porovnání s regulátory PSD a S-PD.

Regulátor	Kritérium	Překmit [%]		Toleranční pásmo $\pm 5\%$ [s]	
		Skok řízení	Skok poruchy	Skok řízení	Skok poruchy
PSD	-	10,0	15,4	5,1	17,3
S-PD	-	0,0	15,4	8,5	5,5
LQ	1	41,8	7,9	5,5	3,3
LQ	2	19,2	7,6	8,5	4,5
LQ&D	2	16,1	7,7	13,0	6,8
LQ	3	37,4	10,9	7,2	4,8
LQ&D&F	3	7,7	4,4	2,4	-
LQ	4	43,4	6,4	7,0	2,5
LQ&D&F&R	4	21,2	3,7	5,0	-

Tab. 9.4 Souhrn výsledků dosažených při simulaci působení poruchových signálů



Obr. 9.3 Odezvy výsledného systému při působení poruchových signálů



Obr. 9.4 Odezvy výsledného systému při působení poruchových signálů

9.2 VÝSLEDKY REGULACE FYZIKÁLNÍ SOUSTAVY

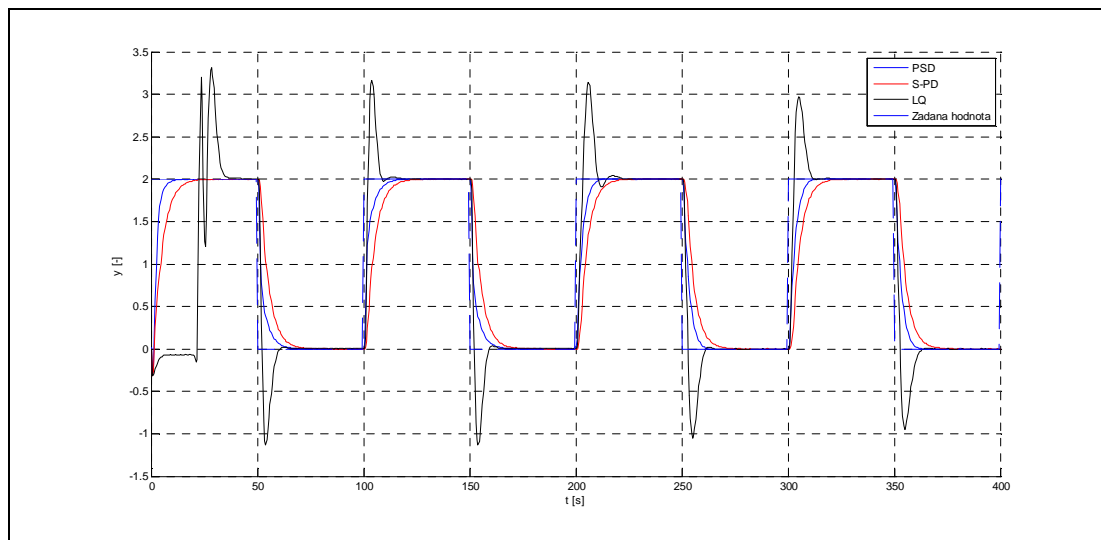
Pro regulaci fyzikální soustavy byl opět použit program SimulaceLQ. Tentokrát však v režimu, který využívá spojení modelů v prostředí MATLAB/Simulink s programovatelným automatem firmy B&R na jehož vstup a výstup je připojena fyzikální soustava. Přestože je pro LQ řízení výhodnější krátká perioda vzorkování, byla zde použita perioda 0,5 sekundy. Tato hodnota byla minimální možnou při dodržení podmínek řízení v reálném čase. Pro identifikaci byla zvolena rekurzivní metoda nejmenších čtverců, která pracovala s matematickým modelem třetího řádu bez dopravního zpoždění. Opět byly řídicí algoritmy nejprve testovány na schopnost adaptace na změnu parametrů (časových konstant) dynamiky regulované soustavy. V tomto případě byly voleny penalizační koeficienty podle tvaru odezvy systému. Konkrétní hodnoty lze nalézt v tab. 9.5. Tab. 9.6 obsahuje nastavení parametrů diskretních ekvivalentů PID regulátorů, které byly použity pro řízení fyzikální soustavy. Obr. 9.5 a obr. 9.6 zobrazují dosažené průběhy výstupu regulované soustavy. Změna časové konstanty regulované soustavy nastala v čase 200 sekund po spuštění simulace. Tab. 9.7 obsahuje obdobně jako tab. 9.3 velikost překmitu a dobu potřebnou pro dosažení tolerančního pásma pěti procent od žádané hodnoty před, v průběhu a po změně časové konstanty.

Číslo kritéria	Řídicí algoritmus	q_u	q_w	q_i	q_d	F
5	LQ	1	0,9	0,1	-	-
6	LQ&D	1	0,9	0,1	1000	-
7	LQ&D&F	1	0,9	0,5	150	0,1
8	LQ&D&F&R	10	0,9	0,1	150	0,2

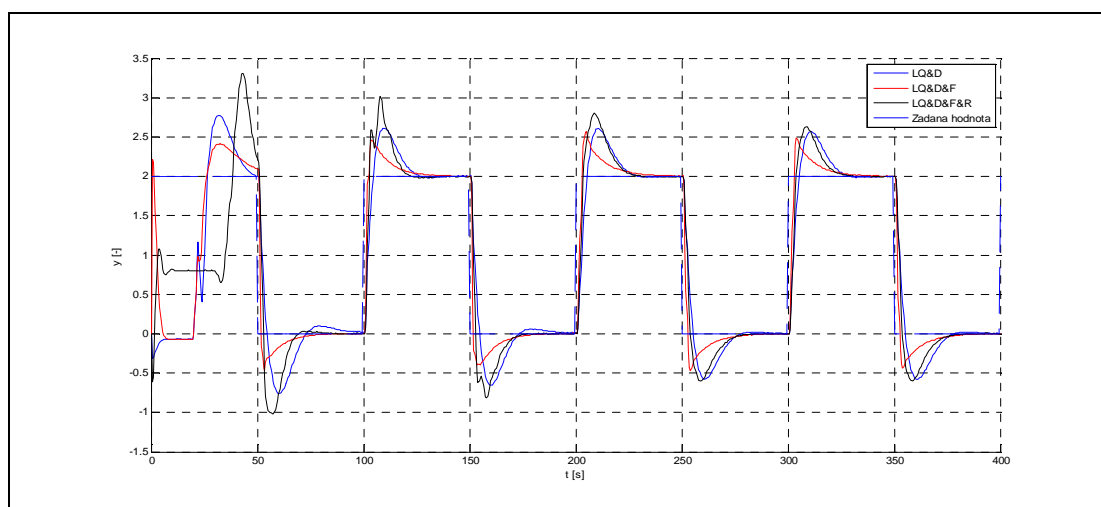
Tab. 9.5 Hodnoty koeficientů pro LQ řízení fyzikální soustavy

Řídicí algoritmus	K	Ti [s]	Td [s]	N
PSD	0,85	3	0,5	3
S-PD	2	4	0,5	3

Tab. 9.6 Hodnoty parametrů pro PSD a S-PD řízení fyzikální soustavy



Obr. 9.5 Odezvy výsledného systému při změně dynamiky fyzikální soustavy



Obr. 9.6 Odezvy výsledného systému při změně dynamiky fyzikální soustavy

Regulátor	Kritérium	Překmit pro soustavu [%]			Toleranční pásmo $\pm 5\%$ [s]		
		F_1	$F_1 \rightarrow F_2$	F_2	F_1	$F_1 \rightarrow F_2$	F_2
PSD	-	0,0	0,0	0,0	9,0	7,5	8,0
S-PD	-	0,0	0,0	0,0	13,5	13,0	13,0
LQ	5	60,0	60,0	50,0	8,0	10,0	10,0
LQ&D	6	30,0	30,3	22,5	20,0	22,5	22,0
LQ&D&F	7	23,1	27,5	24,4	16,5	18,5	16,5
LQ&D&F&R	8	40,5	40,0	30,0	18,0	21,5	20,5

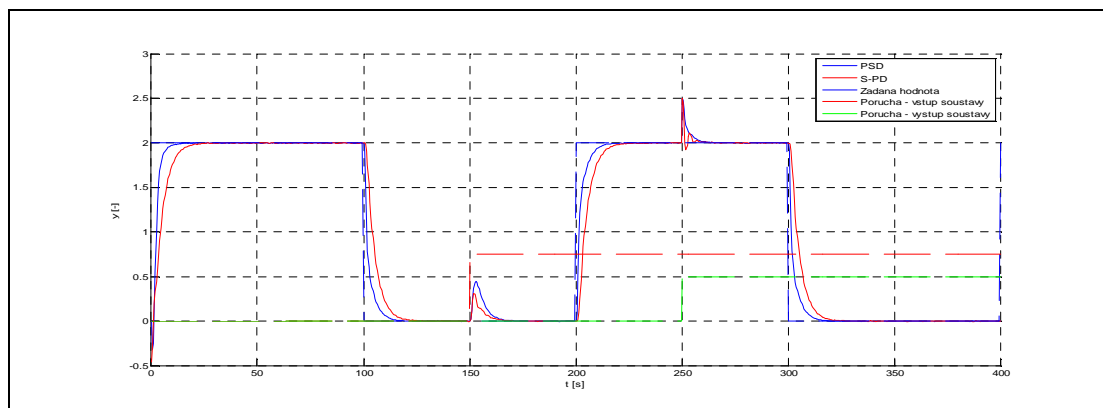
Tab. 9.7 Souhrn výsledků dosažených při změně dynamiky fyzikální soustavy

Při porovnání LQ řízení bez a s diferencí do výstupu je patrné, že přidaná diference opět způsobila snížení překmitu za cenu prodloužení doby trvání přechodného děje. Na variantě LQ&D&F, která využívá stavovou reprezentaci založenou na minulých hodnotách vstupů a výstupů a používá filtrovanou diferenci od výstupu je stejně jako u simulačních výsledků zřejmý nízký vliv integračního členu, který je způsoben působením diference od výstupu. Stejně jako je tomu u simulačních výsledků i při řízení fyzikální soustavy tato varianta dosahuje nejnižšího překmitu. Varianta s čistou diferencí od výstupu (LQ&D) je opět charakteristická značným prodloužením doby trvání přechodného děje. Varianta s filtrovanou diferencí od výstupu, která využívá adaptivní stavový rekonstruktor, nedosahovala pro fyzikální soustavu před změnou jejich parametrů dobrých výsledků, což je patrné z tvaru přechodného děje na obr. 9.6. Tento jev byl však způsoben ne zcela vhodným nastavením kritéria optimality.

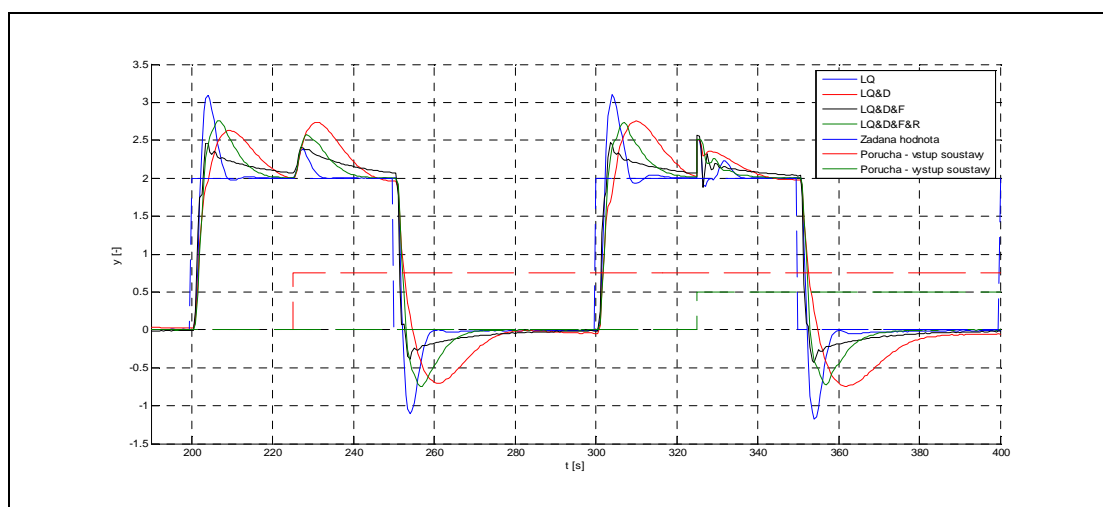
Na obr. 9.7 jsou odezvy výstupu soustavy na změny řízení a poruchových signálů, které působí na vstupu a výstupu regulované soustavy. Obr. 9.8 zobrazuje tytéž odezvy pro všechny zde používané varianty LQ řízení. Při porovnání výsledků jednotlivých variant LQ řízení je patrné, že použití varianty s adaptivním stavovým rekonstruktorem představuje rozumný kompromis mezi velikostmi překmitů při skokové změně řízení a poruchového signálu působícího na vstupu regulované soustavy a délkou doby trvání přechodných dějů. Tab. 9.8 obsahuje získané hodnoty relevantních atributů pro porovnání kvality regulace při působení poruchových signálů.

Regulátor	Kritérium	Překmit [%]		Toleranční pásmo $\pm 5\%$ [s]	
		Skok řízení	Skok poruchy	Skok řízení	Skok poruchy
PSD	-	0,0	22,0	6,5	9,5
S-PD	-	0,0	15,5	13,5	7,0
LQ	5	54,8	19,0	8,0	8,0
LQ&D	6	31,5	36,5	20,5	22,0
LQ&D&F	7	22,5	20,3	20,5	16,5
LQ&D&F&R	8	38,0	28,5	15,5	16,5

Tab. 9.8 Souhrn výsledků dosažených při působení poruchových signálů na fyzikální soustavu



Obr. 9.7 Odezvy výsledného systému při působení poruchových signálů na fyzikální soustavu



Obr. 9.8 Odezvy výsledného systému při působení poruchových signálů na fyzikální soustavu

9.3 STANDARDNÍ ADAPTIVNÍ REGULÁTOR OD FIRMY B&R

Adaptivní regulátor stejně jako velké množství dalších funkcí je součástí knihoven vývojového prostředí Automation Studio. Toto vývojové prostředí slouží jako jediný univerzální nástroj pro veškeré řídicí systémy firmy B&R. To s sebou samozřejmě přináší řadu výhod.

Součástí standardních knihoven je i knihovna LoopConR, která soustřeďuje funkce pro spojitou regulaci v uzavřené zpětnovazební smyčce. V této knihovně lze nalézt blok, který nese označení LCPID a který realizuje PID řízení. Takto realizovaný PID regulátor má široké množství nastavení, což zajišťuje jeho značnou univerzálnost. Jedním ze vstupů bloku PID je sada parametrů, která sdružuje zesílení regulátoru a zesílení integrační a derivační složky. Tento vstup je možno připojit k bloku LCPIDTune, který slouží k automatickému nastavení parametrů PID regulátoru, Blok LCPIDTune umožňuje nastavení z odezvy na jednotkový skok nebo rozkmitáváním a následným použitím metody Zieglera – Nicholse. Tato volba byla použita pro otestování. Testování bylo provedeno na fyzikální soustavě, na kterou působily poruchové signály. Výsledné hodnoty překmitů lze nalézt v tab. 9.9. Testování se změnou dynamiky fyzikální soustavy nebylo provedeno, protože adaptivní regulátor popisovaný v této kapitole není typu STC.

Překmit [%]		Toleranční pásmo $\pm 5\%$ [s]	
Skok řízení	Skok poruchy	Skok řízení	Skok poruchy
23,0	25,0	17,0	13,0

Tab. 9.9 Souhrn výsledků dosažených při působení poruchových signálů s použitím regulátoru R&R

10. PROGRAMOVATELNÉ AUTOMATY

Programovatelné automaty představují jeden za základních stavebních kamenů dnešních automatizovaných procesů. Tuto pozici získaly především díky robustnosti a širokému rozsahu použití. Algoritmy pro programovatelné automaty lze definovat několika způsoby, které jsou předmětem normy IEC1131-3. Mezi často používané metody patří Ladder Diagram (LD), Instruction List (IL), Structured Text (ST), Sequential Function Chart (SCF) a velmi oblíbený jazyk ve standardizované verzi ANSI C.

10.1 IMPLEMENTACE ŘÍDICÍCH ALGORITMŮ DO PLC

Efektivní tvorba řídicích algoritmů se skládá ze třech částí [7].

1. Počáteční etapu představuje vývoj a ladění řídicích algoritmů v simulačních programech. Oblíbeným prostředím je MATLAB/Simulink.
2. Po dosažení dobrých simulačních výsledků následuje ověření algoritmů v reálném procesu, neboť simulační výsledky mohou být z nejrůznějších důvodů zkresleny. V této fázi je třeba propojit MATLAB/Simulink s reálným procesem. Rozhraní mezi MATLABem a reálným procesem tvoří programovatelný automat. Kritickou záležitostí této etapy je zajištění výměny a zpracování dat v reálném čase. Problém představuje fakt, že operační systémy (Microsoft Windows 2000, XP), na kterých je MATLAB většinou, spuštěn nepatří do skupiny operačních systémů pevného reálného času a nezaručují dodržení vzorkovací periody. Je tedy třeba vyřešit výměnu dat mezi PLC a MATLABem v reálném čase.
3. Třetí etapu představuje implementace algoritmů přímo do programovatelného automatu. V této etapě již MATLAB může sloužit pouze jako zdroj signálů nebo jako pozorovatel.

Aby byl čas na tvorbu algoritmu co možná nejkratší a výskyt chyb minimální, musí být přenositelnost řídicího algoritmu mezi všemi etapami co možná nejvyšší.

Algoritmy lze v prostředí MATLAB psát jako skripty do souborů s příponou .m (M-File). V prostředí Simulink lze kromě použití skriptů použít kód napsaný

v jazyce C (S-File), který je následně přeložen do dynamicky linkované knihovny (DLL Dynamic Link Library). Použití s-funkcí má dvě výhody. První výhodou tvoří přenositelnost kódu do programovatelného automatu s nutností minimálních úprav a druhou výhodou je fakt, že pro vykonávání s-funkce v prostředí MATLAB je třeba znatelně kratší strojový čas.

Komunikaci mezi programovatelnými automaty firmy B&R s okolím zajišťuje rozhraní PVI (Process Visualization Interface). Sériové sběrnice, které lze využít pro komunikaci, jsou RS232, Ethernet, Profibus, CAN a modem. Na straně prostředí MATLAB byla vytvořena knihovna pro komunikaci v reálném čase. Tato knihovna zajišťuje stabilní a robustní komunikaci s PLC a usnadňuje přímou implementaci řídicího algoritmu do PLC.

10.2 IMPLEMENTACE LQ ŘÍZENÍ DO PLC B&R

Aby byly algoritmy co nejvíce přenositelné mezi jednotlivými etapami, byly všechna napsána v jazyce ANSI C, protože tento jazyk podporuje jak prostředí MATLAB/Simulink, tak i Automation Studio firmy B&R. Pro implementaci v obou prostředích by bylo nejlepší použít některý z objektově orientovaných jazyků, nicméně prostředí Automation Studio 3.0.71 žádný takový jazyk nepodporuje. Pro účely implementace byla vytvořena knihovna matice, která je obsažená na příloženém CD. Ta byla následně použita v prostředí MATLAB/Simulink. Během překlápění kódu z MATLABu do Automation Studia se však objevil problém, který byl způsoben tím, že Automation Studio nepodporuje standardizovanou funkci malloc. Tato funkce slouží k alokaci dynamických proměnných. Z tohoto důvodu vznikla knihovna matice01, která je opět součástí elektronických příloh obsažených na CD. Ta je již celá postavená na statických proměnných.

Implementované algoritmy lze nalézt na příloženém CD v projektu označeném LQrizeni. Po jeho otevření se objeví šest programů. Ty realizují jednotlivé algoritmy a umožňují modularitu výsledného řízení. Každý program obsahuje jediný soubor s koncovkou c a knihovnu matice01. Ten vždy obsahuje inicializační funkci a funkci, jejíž kód je určen k cyklickému provádění. Programy označené LM a RLS obsahují on-line identifikaci. Program RLS, který realizuje

identifikaci rekurzivní metodou nejmenších čtverců, pro svoji činnost potřebuje následující globální proměnné:

```
_GLOBAL LREAL U;  
_GLOBAL LREAL Y;  
_GLOBAL LREAL THETA[9];  
_GLOBAL LREAL COEF;  
_GLOBAL INT rad;
```

kde U představuje proměnnou akčního zásahu (vstupu do soustavy), která je pro řízení fyzikální soustavy propojena (po denormalizaci) na výstup výstupní analogové karty. Proměnná Y představuje proměnnou výstupu ze soustavy. Ta by měla být připojena (po normalizaci) na vstup vstupní karty. Proměnná THETA slouží k předávání identifikovaných parametrů matematického modelu mezi identifikací a řídicími algoritmy. Proměnná COEF má význam koeficientu zapomínání. Proměnnou rad lze nastavit strukturu matematického modelu. Její hodnota představuje počet členů čitatele a jmenovatele operátorového přenosu. Maximální možná hodnota tohoto parametru je čtyři. Toto omezení je dáno použitím pouze statických dat. Vstupem do programu jsou tedy proměnné U a Y, které jsou získány měřením v procesu a proměnné COEF a rad, které je třeba nastavit před spuštěním procesu. Proměnná THETA představuje výstup, který obsahuje parametry matematického modelu regulované soustavy. Program LM soužít k identifikaci umělou neuronovou sítí. Proměnné popsané pro program RLS jsou platné i pro program LM se stejným významem kromě proměnné COEF, která zde představuje koeficient učení. Program LM dále potřebuje následující proměnné:

```
_GLOBAL INT Max_hloubka;  
_GLOBAL INT Pocet_iteraci;
```

kde Max_hloubka má význam maximálního počtu trénovacích vzorů a Počet_iteraci představuje počet iterací učení, které se má provést při jednom volání cyklické funkce programu.

Programy LQD, LQDF a LQDFR realizují jednotlivé řídicí algoritmy. Opět pro svoji činnost potřebují následující globální proměnné:

```
_GLOBAL LREAL U;  
_GLOBAL LREAL Y;  
_GLOBAL LREAL THETA[9];  
_GLOBAL INT rad;  
_GLOBAL LREAL qu;  
_GLOBAL LREAL qi;  
_GLOBAL LREAL qd;  
_GLOBAL LREAL qw;  
_GLOBAL LREAL W;
```

kde význam proměnných U, Y, THETA a rad jsou shodné s proměnnými použitými pro identifikaci. Dále je třeba přidat proměnné penalizačních koeficientů kritéria řídicího algoritmu qu, qi, qd a W a proměnnou W, která má význam žádané hodnoty. Pro programy LQDF a LQDFR je dále třeba přidat následující proměnnou

```
_GLOBAL LREAL F;
```

kteřá má význam koeficientu filtru difference od výstupu. Program LQDFR vyžaduje pro svoji činnost stavový rekonstruktor. Ten je v programu REK. Program REK vyžaduje následující proměnné:

```
_GLOBAL LREAL U;  
_GLOBAL LREAL Y;  
_GLOBAL LREAL THETA[9];  
_GLOBAL INT rad;
```

`_GLOBAL LREAL fiR[4];`

kde první čtyři proměnné mají shodný význam jako v programech pro identifikaci a řízení, proměnná `fiR` slouží k předávání aktuálního stavu stavového rekonstruktoru programu LQDFR, pro který musí být tato proměnná také viditelná. Projekt LQřízení obsahuje soubor `Global.var`, ve kterém jsou deklarace a inicializace všech výše popsaných proměnných; pro použití v jiném projektu je možno potřebné proměnné z něho zkopírovat. Po vytvoření výše uvedených potřebných proměnných lze přistoupit k samotnému použití programů. Všechny programy pro řízení (LQ&D, LQ&D&F, LQ&D&F&R) vyžadují, aby v některé smyčce byl obsažen právě jeden program, který realizuje identifikaci (RLS nebo LM). Identifikační program a řídicí program přitom nemusí být umístěny ve stejné smyčce. Jak již bylo zmíněno program LQDFR vyžaduje přítomnost programu REK a to ve stejné smyčce jako je on sám a musí být umístěn nad programem LQ&D&F&R. V projektu byla vytvořena jediná hardwarová konfigurace a to konfigurace pro simulátor, ve kterém byly všechny programy testovány.

11. ZÁVĚR

Tato práce popisuje problematiku adaptivního LQ regulátoru. Na počátku práce je zmíněn stručný přehled adaptivních systémů, především systémů typu self-tuning. Po této úvodní kapitole následuje pasáž věnovaná identifikaci dynamických systémů, která představuje jeden ze dvou elementárních bloků samočinně se nastavujících regulátorů. Pozornost je upřena především na metody vhodné pro průběžnou identifikaci a to identifikaci metodou nejmenších čtverců a identifikaci založenou na bázi umělých neuronových sítí. V závěru této kapitoly lze nalézt porovnání obou přístupů při identifikaci dynamické soustavy třetího řádu. Pro identifikaci byly postupně použity modely druhého a třetího řádu. Ve výsledném porovnání dosahovala rekurzivní metoda nejmenších čtverců přesnějších a robustnějších výsledků a to bez i s působícím poruchovým signálem na výstupu identifikované soustavy.

Další velká kapitola je věnována optimálnímu řízení. Po krátkém představení dynamického programování následuje popis lineárního řízení, které splňuje kvadratické kritérium (LQ řízení). Tato pasáž je zaměřena na rozšíření půdního LQ řízení, které převádí systém z libovolného bodu stavového prostoru do jeho počátku o sledování referenční trajektorie a rozšíření struktury řídicího algoritmu o sumační složku. Přidání sumačního členu do struktury sice umožňuje splnění požadavků na hodnotu regulační odchylky v ustáleném stavu, ale také často způsobuje nárůst překmitu přechodného děje. Ve snaze snížit tento překmit bylo navrženo další rozšíření struktury regulátoru. Rozšíření bylo inspirováno diskretní obdobou PI-D regulátoru, konkrétně diferencí od výstupu regulované soustavy. Tuto variantu lze celkem snadno implementovat do stávajících rozšíření. Byly vytvořeny tři nové struktury, které jsou v této práci popsány. Jedná se o rozšíření čistou diferencí od výstupu, kterou lze aplikovat pouze na systém, jehož stavová reprezentace je tvořena minulými hodnotami vstupu a výstupu. Čistou diferencí lze do systému, který pracuje se stavovým prostorem založeným na předchozích hodnotách vstupu a výstupu regulované soustavy, implementovat bez nutnosti rozšíření struktury. Pro její implementaci stačí pouze upravit tvar kritéria. Druhá skupina byla rozšířena

o filtraci difference od výstupu. Zde vznikla varianta, která jako v předchozím případě využívá stavová reprezentace založené na minulých hodnotách vstupu a výstupu a varianta, která využívá adaptivní stavový rekonstruktor. Pro tuto variantu je již třeba strukturu rozšířit o další stavovou veličinu a to o výstup filtru. Protože má kritérium u variant s a bez filtrace difference od výstupu jiný tvar, nelze nastavením koeficientu filtru na nulu přejít z jedné varianty na druhou. Úpravy, které využívají difference od výstupu soustavy v LQ řízení lze považovat za originální, protože jsem je nenašel v žádné rešerši.

Následně byla popsána simulační aplikace SimulaceLQ, která slouží k ověření funkce upraveného LQ. Aplikace umožňuje použít jako regulovanou soustavu matematický model i fyzikální soustavu, která je připojena k programovatelnému automatu firmy B&R.

Po popisu aplikace SimulaceLQ následuje popis programu Estimace, který slouží jako nástroj pro demonstraci vývoje parametrů průběžné (online) identifikace. V aplikaci byla realizována průběžná metoda nejmenších čtverců a umělá neuronová síť.

Další kapitola obsahuje souhrn výsledků provedených testů při změně dynamických vlastností regulované soustavy a při působení poruchových signálů na vstupu a výstupu soustavy. Testování bylo provedeno za použití aplikace SimulaceLQ na matematickém modelu regulované soustavy a na fyzikální soustavě. Z dosažených výsledků je patrné, že všechny tři navržené úpravy jsou schopny snížit překmit přechodného děje oproti LQ regulátoru bez přidané difference od výstupu při shodném nastavení sobě odpovídajících koeficientů kritéria. V některých případech představuje snížení překmitu přechodného děje více jak deset procent. Varianty, které využívají stavovou reprezentaci založenou na minulých hodnotách vstupu a výstupu vykazovali schopnost značného potlačení překmitu za cenu značného snížení vlivu sumačního členu a tedy prodloužení přechodného děje. Za nejlepší variantu lze považovat variantu, která využívá adaptivní stavový rekonstruktor. Simulační výsledky této varianty byly srovnatelné se simulačními výsledky PSD regulátoru. Aby bylo rozšíření o difference od výstupu dostatečně účinné, musí mít penalizační koeficient poměrně vysokou hodnotu při porovnání s velikostmi

ostatních penalizačních koeficientů. Závěr této části je pak věnován standardnímu adaptivnímu regulátoru od firmy B&R.

Závěr práce je věnován problematice přístupu k vývoji a ladění řídicích algoritmů, možnostem propojení PC s programovatelným automatem firmy B&R a problémům s ním spojeným. V této závěrečné kapitole lze nalézt řešení problémů spojených s implementací jak v obecné rovině, tak i konkrétní implementaci všech zde popsaných variant LQ řízení.

12. LITERATURA

- [1] PIVOŇKA, P.: *Optimalizace regulátorů*. VUT Brno, skriptum 2005.
- [2] PIVOŇKA, P.: *Číslicová řídicí technika*. VUT Brno, skriptum 2003.
- [3] KUBÍK, S. - KOTEK, Z., RAZÍM, M. - HRUŠÁK, J., BRANŽOVSKÝ, J.:
Teorie automatického řízení 2. SNTL, Praha 1982
- [4] PIVOŇKA, P. - SCHMIDT, M.: *Comparative Analysis of Discrete Derivative Implementations in PID Controllers. Systems Theory and Applications, vol. 2*,
Řecko: WSEAS, 2007, s. 33-37. ISBN: 978-960-8457-90-4.
- [5] HAVLENA, V. - ŠTECHA, J.: *Moderní teorie řízení*. ČVUT, skriptum, 1999.
Dostupné z: <http://dce.felk.cvut.cz/stecha/mtr.pdf> [citováno 2009-04-04]
- [6] LORENC, V.: *LQ regulátor* (učební text). UAMT VUT Brno, 2007.
- [7] ŠVANCARA, K.: *Adaptive optimal controller with identification based on neural networks*. Brno, 2004. Disertační práce na Fakultě elektrotechniky a komunikačních technologií Vysokého učení technického v Brně na Ústavu automatizace. Vedoucí disertační práce Prof. Petr Pivoňka.
- [8] BOBÁL, V. - BÖHM, J. - PROKOP, R. - FESSL, J.: *Praktické aspekty samočinně se nastavujících regulátorů: algoritmy a implementace*. VUTIUM, 1999. ISBN 80-214-1299-2.
- [9] ŠTECHA, J.: *Optimální rozhodování a řízení*. ČVUT, skriptum, 1999.
Dostupné z: <http://dce.felk.cvut.cz/stecha/mtr.pdf> [citováno 2009-03-15]
- [10] MRÁZEK, M. *Adaptivní optimální regulátory s principy umělé inteligence v prostředí MATLAB - B&R*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 73 s. Vedoucí diplomové práce prof. Ing. Petr Pivoňka, CSc.
- [11] DRÁBEK, O. - TAUFER, I. - SEIDL, P.: *Umělé neuronové sítě – základy teorie a aplikace (3)*. *CHEMagazín*. 2006, roč. XVI, č. 1, str. 12-14. ISSN 1210-7409.
- [12] KUBÍK, S. – KOTEK, Z. – STREJC, V. – ŠTECHA, J.: *Teorie automatického řízení 1*. SNTL, Praha 1982

13. SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

MRAS	System s referenčním modelem
STC	Samočinně se nastavující regulátor
LQ	Lineární řízení splňující kvadratické kritérium
LQ&D	LQ řízení s čistou diferenční složkou od výstupu
LQ&D&F	LQ řízení s filtrovanou diferenční složkou od výstupu
LQ&D&F&R	LQ řízení s filtrovanou diferenční složkou od výstupu a stavovým rekonstruktorem
ARX	Autoregresní model
UNS	Umělá neuronová síť
SISO	System s jedním vstupem a jedním výstupem
MIMO	System s více vstupy a více výstupy
PSD	Proporcionálně sumačně diferenční regulátor
D	Diferenční složka
LD	Ladder Diagram
IL	Instruction List
ST	Structured Text
SCF	Sequential Function Chart
PLC	Programovatelný automat
DLL	Dynamicky linkovaná knihovna
PVI	Process Visualization Interface
$y(t)$	Výstupní signál
$v(t)$	Měřitelný poruchový signál
$u(t)$	Akční zásah, vstupní signál soustavy
$e_s(k)$	Stochastický signál
$e(k)$	Regulační odchylka
$\hat{e}(k)$	Chyba predikce

$w(k)$	Signál žádané hodnoty
w_{NN}	Vektor vah neuronu
k	Diskrétní čas
$\varphi(k)$	Vektor dat
$\theta(k)$	Vektor parametrů
$\hat{e}(k)$	Chyba predikce
J	Kritérium
Φ	Matice dat
$P(i)$	Kovarianční matice
λ	Koeficient zapomínání
R	Penalizační matice vstupu
Q	Matice koeficientů kritéria
q	Koeficient kritéria
F	Koeficient filtru
$x(k)$	Stavový vektor
$z(k)$	Stavový vektor založený na minulých hodnotách vstupů a výstupů