

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

ŘÍZENÍ REÁLNÉHO PROCESU STAVOVÝM REGULÁTOREM LQ

Bc. Ondřej Zouhar

Diplomová práce
2017

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2016/2017

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Ondřej Zouhar**
Osobní číslo: **I15192**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Řízení procesů**
Název tématu: **Řízení reálného procesu stavovým regulátorem LQ**
Zadávající katedra: **Katedra řízení procesů**

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

S využitím vhodné vývojové desky s jednočipovým počítačem bude vytvořen diskrétní regulátor typu LQ s estimátorem stavu pro řízení reálného procesu. Proces je modelován přenosem vyššího řádu. Parametry přenosové funkce budou určeny identifikací z měřených dat. Vlastnosti řídicího systému budou experimentálně ověřeny.

Teoretická část bude obsahovat uvedení do problematiky a popis využitých metod. Praktická část bude obsahovat naměřené odezvy v podobě grafů a výpočetní skripty.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

KUBÍK, S.; KOTEK, Z.; STREJC, V.; ŠTECHA, J. Teorie automatického řízení I. Praha: SNTL, 1982.

STENGEL, R.F. Stochastic Optimal Control: Theory and Application. New York: Wiley, 1986.

Vedoucí diplomové práce:

doc. Ing. Jan Cvejn, Ph.D.

Katedra řízení procesů

Datum zadání diplomové práce:

20. října 2016

Termín odevzdání diplomové práce:

17. května 2017



Ing. Zdeněk Němec, Ph.D.
děkan



L.S.



Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2016

Prohlášení

Prohlašuji:

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

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

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 1. 5. 2017

Bc. Ondřej Zouhar

Poděkování

Děkuji vedoucímu mé diplomové práce panu Doc. Ing. Janu Cvejnovi, Ph.D. za poskytnutí cenných odborných rad a čas, který mi věnoval během psaní mé diplomové práce. Můj dík patří rovněž rodině i přátelům, kteří mě podporovali při studiu.

V Pardubicích dne 1. 5. 2017

Bc. Ondřej Zouhar

ANOTACE

Tato diplomová práce se zabývá řízením reálného procesu pomocí lineárně-kvadratického (LQ) optimálního regulátoru, který je doplněn estimátorem stavu. Regulátor byl realizován s využitím jednočipového mikropočítače Arduino. Náplní práce bylo rovněž vyhodnocení regulačních pochodů v závislosti na nastavení pólů estimátoru stavu i volbě váhových matic LQ regulátoru. Pro odladění byly veškeré algoritmy nejprve testovány v prostředí MATLAB.

KLÍČOVÁ SLOVA

LQ regulátor, optimalizace, řízení procesů, automatizace, estimátor úplného řádu.

TITLE

CONTROL OF A REAL PROCESS BY THE LQ STATE CONTROLLER.

ANNOTATION

The diploma thesis deals with process control by means of the linear-quadratic (LQ) optimal controller with additional state observer. The LQ controller and the state observer have been implemented by using simple Arduino microcomputer. The evaluation of quality of control, depending on the state observer and LQ controller settings, was one of goals of this thesis as well. As the first step, all algorithms were tested in MATLAB environment.

KEYWORDS

LQ controller, Optimization, Process control, Automation, State observer.

OBSAH

Seznam zkratk a značek	9
Seznam symbolů proměnných veličin a funkcí	10
Seznam ilustrací	14
Seznam tabulek	16
ÚVOD	17
1 TEORETICKÁ ČÁST	18
1.1 Vybrané pojmy ze stavové teorie řízení.....	18
1.1.1 Nelineární a lineární spojitý stavový popis.....	19
1.1.2 Přenosová matice lineární soustavy ze spojitého stavového popisu	22
1.1.3 Nelineární a lineární diskrétní stavový popis.....	24
1.1.4 Matice přenosu lineární soustavy z diskrétního stavového popisu	26
1.1.5 Stavová trajektorie	27
1.1.6 Sestavení matic popisujících diskrétní stavový popis.....	28
1.2 Vlastnosti stavové realizace	33
1.2.1 Dosažitelnost, říditelnost a stabilizovatelnost.....	33
1.2.2 Pozorovatelnost, rekonstruovatelnost a detekovatelnost	35
1.2.3 Identifikovatelnost	39
1.3 Identifikace soustavy.....	41
1.3.1 Rozdělení modelů	41
1.3.2 Matematicko-fyzikální modelování	43
1.3.3 Experimentální identifikace	44
1.4 Estimace stavu.....	45
1.4.1 Diskrétní estimátor úplného řádu.....	45
1.5 LQ regulátor.....	47
2 PRAKTICKÉ ŘEŠENÍ	51
2.1 Využití HW a SW prostředky	51
2.1.1 Arduino USB v2.2.....	52
2.1.2 Programování Arduino.....	53
2.2 Propojení reálného procesu s regulátorem	54
2.3 Identifikace neznámé soustavy	56
2.4 Výpočet parametrů estimátoru	59
2.5 Výpočet matice LQ regulátoru.....	63

2.6	LQ regulátor s rozšířením stavu o sumační složku	66
2.7	Otestování estimátoru na reálné soustavě	69
2.8	Ověření funkce regulátorů na reálné soustavě	72
3	ZHODNOCENÍ	77
4	ZÁVĚR	79
	POUŽITÁ LITERATURA	80
	PŘÍLOHY	81

SEZNAM ZKRATEK A ZNAČEK

EEPROM	Electrically Erasable Programmable Read-Only Memory
GUI	Graphical User Interface
HW	Hardware
IDE	Integrated Development Environment
IIDI	Ivrea Interaction Design Institute
LCD	Liquid Crystal Display
LED	Light-Emitting diode
LQ	Linear Quadratic
MATLAB	Matrix Laboratory
PC	Personal Computer
PWM	Pulse Wide Modulation
RC	Resistance Capacitance
RXD	Receive Data
SW	Software
SRAM	Static Random Access Memory
TXD	Transmit Data
USB	Universal Serial Bus

SEZNAM SYMBOLŮ PROMĚNNÝCH VELIČIN A FUNKCÍ

a_0, \dots, a_{f-1}	koeficienty výstupního polynomu soustavy
$A(t)$	matice soustavy
b_0, \dots, b_{f-1}	koeficienty vstupního polynomu soustavy
$B(t)$	matice buzení
C	výstupní matice diskrétní soustavy
C_R	výstupní matice vypočtená normální formou rekonstruovatelnosti
$C_{\tilde{R}}$	výstupní matice vypočtená normální formou říditelnosti
$C(t)$	výstupní matice spojité soustavy
d	počet vstupních veličin
D	matice vazeb vstupu na výstup diskrétní soustavy
$D(t)$	matice převodu
e	počet výstupních veličin
$e(k)$	vektor chyb odhadu
$E(s)$	Laplaceův obraz vektoru chyb odhadu
f	počet stavových proměnných
f_p	obecně nelineární funkce pro výpočet stavové rovnice spojité soustavy
g_r	obecně nelineární funkce pro výpočet výstupní rovnice spojité soustavy
$G(s)$	přenosová matice soustavy
$G(z)$	matice přechodu diskrétní soustavy
h	pomocná proměnná pro výpočet sumace
ch	nejvyšší hodnota v a w při výpočtu parciální derivace funkce f_v
i	index Laplaceovy transformace výstupní veličiny $y(t)$
I	jednotková matice
j	index Laplaceovy transformace vstupní veličiny $u(t)$
J	kritérium LQ regulátoru
k_1, k_2, k_3	složky matice estimátoru třetího řádu
k	diskrétní vzorky času
K	matice obecného stavového regulátoru

K_e	matice estimátoru
K_R	matice regulátoru
l	pomocná proměnná pro výpočet sumace
M	matice diskretní soustavy
M_J	Jordanova matice
M_R	matice diskretní soustavy vypočtená normální formou rekonstruovatelnosti
$M_{\check{R}}$	matice diskretní soustavy vypočtená normální formou říditelnosti
N	délka intervalu řízení,
N	matice řízení diskretní soustavy
N_R	matice řízení vypočtená normální formou rekonstruovatelnosti
$N_{\check{R}}$	matice řízení vypočtená normální formou říditelnosti
p	ukazatel pro výpočet stavové rovnice spojité soustavy
P_N	matice penalizace konvergence výstupní veličiny k nule
P_P	matice výstupu
q	nejvyšší hodnota w při výpočtu parciální derivace funkce f_v
Q	konstantní matice penalizace stavových veličin
Q_i	složka matice Q , ležící na i -tém řádku i sloupci
Q_k	v čase proměnná matice penalizace stavových veličin
R	konstantní matice penalizace akčních zásahů
R_i	složka matice R , ležící na i -tém řádku i sloupci
R_k	v čase proměnná matice penalizace akčních zásahů
s	Laplaceovův operátor
S_k	aktuální hodnota Riccatiho rovnice
$S(k)$	výsledek Riccatiho iteračního výpočtu v daném kroku
S_N	hodnota Riccatiho rovnice v nekonečnu
$S(N)$	hodnota Riccatiho iteračního výpočtu v nekonečnu
t	okamžik času náležící do množiny časových okamžiků T
t_0	počáteční čas
T	matice nul

u_1, \dots, u_d	jednotlivé hodnoty vektoru vstupních hodnot
u_i^{\max}	maximální povolená hodnota akčního zásahu i-té složky vstupu soustavy
$\mathbf{u}(k)$	vektor vstupních hodnot v aktuálním kroku
$U(s)$	Laplaceovův obraz vstupní veličiny $\mathbf{u}(t)$
$\mathbf{u}(t)$	vektor spojitých vstupních proměnných
v	ukazatel pro výpočet parciální derivace
\mathbf{V}_D^D	matice dosažitelnosti diskrétní soustavy
\mathbf{V}_D^S	matice dosažitelnosti spojitě soustavy
$\mathbf{v}(l)$	vektor poruchové veličiny
\mathbf{V}_P^D	matice pozorovatelnosti
\mathbf{V}_R^D	matice rekonstruovatelnosti
$\mathbf{V}_{\tilde{R}}^D$	matice říditelnosti diskrétní soustavy
w	ukazatel pro výpočet parciální derivace
x_1, \dots, x_f	všechny složky stavového vektoru
x	stav náležející do množiny stavů \mathbf{x}
$X(s)$	Laplaceovův obraz vstupní veličiny $\mathbf{u}(t)$
$\mathbf{x}(t)$	vektor stavů spojitě soustavy v čase t
$\mathbf{x}(t_0)$	vektor stavů spojitě soustavy v čase t_0
$\dot{\mathbf{x}}_p(t)$	výsledek stavové rovnice spojitě soustavy
$\hat{\mathbf{x}}(k)$	odhad vektoru stavů
$\mathbf{x}(k+1)$	hodnota stavu v kroku $k+1$
$\mathbf{x}_J(k)$	stavový vektor získaný Jordanovým tvarem
$\mathbf{x}_P(k)$	stavový vektor získaný normální formou pozorovatelnosti
$\mathbf{x}_R(k)$	stavový vektor získaný normální formou rekonstruovatelnosti
$\mathbf{x}_{\tilde{R}}(k)$	stavový vektor získaný normální formou říditelnosti
y_i^{\max}	maximální povolená hodnota odchylky i-té složky výstupu soustavy
$\mathbf{y}(k)$	vektor výstupních hodnot v aktuálním kroku
$Y(s)$	Laplaceovův obraz výstupní veličiny $\mathbf{y}(t)$

$\mathbf{y}(t)$	vektor spojitých výstupních proměnných
$\mathbf{y}_J(k)$	vektor výstupu získaný Jordanovým tvarem
$\mathbf{y}_P(k)$	vektor výstupu získaný normální formou pozorovatelnosti
$\mathbf{y}_r(t)$	výsledek výstupní rovnice spojité soustavy
$\mathbf{y}_R(k)$	vektor výstupu získaný normální formou rekonstruovatelnosti
$\mathbf{y}_{\tilde{R}}(k)$	vektor výstupu získaný normální formou říditelnosti
z	komplexní proměnná
α	proměnná pro volbu pólu estimátoru
ε	kritérium ustálení výsledku Riccatiho rovnice
$\eta()$	výstupní zobrazení
τ	počáteční časový okamžik
φ	funkce změny stavu
ω	vstupní segment náležící do množiny přijatelných vstupních funkcí Ω
Ω	množina přijatelných vstupních funkcí

SEZNAM ILUSTRACÍ

Obr. 1.1 – Vyobrazení obecné soustavy	19
Obr. 1.2 – Blokové schéma spojitého stavového popisu	22
Obr. 1.3 – Blokové schéma diskrétního stavového popisu.....	25
Obr. 1.4 – Nákres stavové trajektorie v trojrozměrném prostoru	27
Obr. 1.5 – Blokové schéma stavových rovnic v normální formě říditelnosti.....	30
Obr. 1.6 – Blokové schéma stavových rovnic v normální formě rekonstruovatelnosti.....	31
Obr. 1.7 – Blokové schéma stavových rovnic v normální formě pozorovatelnosti.....	31
Obr. 1.8 – Blokové schéma příkladu Jordanova základního tvaru	32
Obr. 1.9 – Nákres dosažitelnosti a říditelnosti.....	35
Obr. 1.10 – Náčrt pozorovatelnosti a rekonstruovatelnosti	39
Obr. 1.11 – Rozdělení modelů	41
Obr. 1.12 – Blokové schéma diskrétního estimátoru úplného řádu.....	47
Obr. 2.1 – Prostředí Labctrl	51
Obr. 2.2 – Arduino logo.....	51
Obr. 2.3 – Arduino USB v2.2	52
Obr. 2.4 – Vývojové prostředí	53
Obr. 2.5 – Blokové schéma zapojení regulačního obvodu	54
Obr. 2.6 – Skutečný regulační obvod	55
Obr. 2.7 – Data použitá pro identifikaci soustavy	56
Obr. 2.8 – Vyšetření výsledku identifikace	58
Obr. 2.9 – Estimátor sestavený v prostředí Simulink	62
Obr. 2.10 – Průběh odhadu výstupu soustavy pro $\alpha = -0,5$	62
Obr. 2.11 – Grafické znázornění Riccatiho iteračního výpočtu	63
Obr. 2.12 – Regulační pochod s LQ regulátorem v prostředí Simulink.....	64
Obr. 2.13 – Průběhy příslušných stavů soustavy	65
Obr. 2.14 – Realizace LQ regulátoru v prostředí Simulink	65
Obr. 2.15 – Regulační obvod v prostředí Simulink	66
Obr. 2.16 – Úprava estimátoru pro zadávání žádané hodnoty.....	67
Obr. 2.17 – Regulační pochod v prostředí Simulink	67
Obr. 2.18 – Grafické znázornění stavového vektoru	68
Obr. 2.19 – Grafické znázornění sumační složky vektoru stavu	68
Obr. 2.20 – Regulační pochod s LQ regulátorem bez estimátoru.....	69

Obr. 2.21 – Vliv pólů estimátoru stavu na výsledné řízení.....	70
Obr. 2.22 – Srovnání měřeného a odhadovaného výstupu soustavy	71
Obr. 2.23 – Regulační pochod s LQ regulátorem s estimátorem.....	72
Obr. 2.24 – Regulační pochod skutečné soustavy, kmitavý	73
Obr. 2.25 – Regulační pochod skutečné soustavy, aperiodický	74
Obr. 2.26 – Reakce regulátoru na působení poruchy.....	75
Obr. 2.27 – Grafické znázornění vektoru stavů soustavy	76
Obr. 2.28 – Grafické znázornění sumační složky stavu	76

SEZNAM TABULEK

Tab. 2.1 – Přehled součástí	55
Tab. 3.1 – Přehled řízení soustavy jednotlivými regulátory	77

ÚVOD

Stavová regulace využívá tzv. vnitřního popisu soustavy, a tudíž umožňuje řídit i velmi složité soustavy navíc s možností specifikace dalších požadavků na takto vzniklý regulační obvod. S výhodou může být stavová regulace použita pro soustavy s více vstupy a výstupy, které mohou být navzájem složitě propojené. V praxi nyní převládají klasické PID regulátory, které v porovnání se stavovými regulátory vynikají svou jednoduchostí při nastavování jejich parametrů. Rovněž poskytují ucházející regulační pochody a to i v případě nepřesného nastavení parametrů. Základem konvenční teorie regulace je přechodová funkce soustavy či popis soustavy s použitím jedné diferenciální rovnice f -tého řádu. Kde f udává řád soustavy. Naopak moderní teorie řízení je založena na popisu soustavy pomocí f diferenciálních rovnic prvního řádu, které mohou být transformovány na příslušné maticové diferenciální rovnice. Ty mnohonásobně zjednoduší matematické vyjádření dané soustavy a jsou vhodné pro algoritmizaci výpočtů. Stavových regulátorů je však vhodné použít nejenom např. při regulaci složitých chemických procesů, ale i k řízení jakýchkoliv jiných soustav, přičemž je tak zajištěno optimální řešení. Optimální řešení stavové regulace procesu je první fází k možnosti úspory energie vynaložené během řízení procesu a současně může prodloužit dobu životnosti akčních členů.

Struktura práce byla rozdělena na teoretickou a praktickou část. První oddíl a jemu příslušné pododdíly byly věnovány úvodu do stavového popisu soustavy a způsobům sestavení matic reprezentujících diskrétní soustavu ve stavovém prostoru. Následoval druhý oddíl pojednávající o vlastnostech soustav zapsaných ve stavovém popisu. Dále pak byly ve třetím oddílu popsány metody identifikace, které umožňují získat matematický popis neznámé soustavy. Estimace stavového vektoru společně s podrobnou charakteristikou diskrétního estimátoru úplného řádu je uvedena v oddílu čtvrtém. Následně v návaznosti k estimaci stavu pojednává oddíl pátý o LQ regulátoru.

V praktické části práce byly v prvním oddíle uvedeny používané programy i skutečné zařízení. Druhý oddíl popisuje zapojení regulačního obvodu sloužícího k řízení reálného procesu. Poté, co došlo ve třetím oddíle k identifikaci konkrétních parametrů řízené soustavy, mohl být ve čtvrtém oddíle odvozen estimátor stavu. K odvození LQ regulátoru pro regulaci reálného procesu došlo v oddíle pátém. Tento LQ regulátor byl následně v šestém oddíle upraven tak, že pracoval se stavem soustavy rozšířeným o sumační složku. V oddíle sedmém potom došlo k otestování funkčnosti několika variant estimátoru stavu, zatímco vlastnosti variací LQ regulátorů vyšetřuje oddíl osmý.

1 TEORETICKÁ ČÁST

1.1 VYBRANÉ POJMY ZE STAVOVÉ TEORIE ŘÍZENÍ

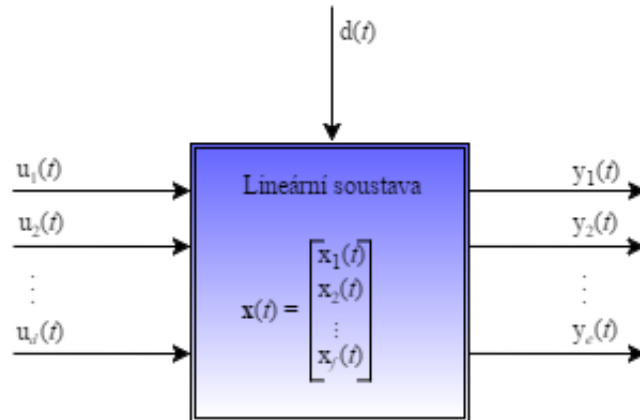
Dynamika dané soustavy může být popsána matematickým modelem soustavy. Matematický model definuje matematické vztahy mezi vstupními, výstupními a stavovými veličinami. Vstup soustavy je vyjádřen množinou časových posloupností hodnot, které ovlivňují danou soustavu. Výstup soustavy je pak vyjádřen obdobně jako vstup, avšak reprezentuje popis přímo pozorovatelného chování soustavy. Mezi základní vlastnosti kteréhokoliv dynamického systému patří skutečnost, že jeho chování v libovolném časovém okamžiku nezávisí pouze na hodnotě vstupní veličiny působící právě v tomto okamžiku, ale i na hodnotách vstupních veličin, které na soustavu působily v minulosti. Obecně tedy platí, že v soustavě je obsažena paměť.

Okamžité stavy této paměti, jsou vyjádřeny jako množina hodnot tzv. stavových veličin a odpovídají stavu soustavy. Pokud je znám stav soustavy v libovolném okamžiku t_0 a vstup soustavy v jakémkoliv následujícím čase t , je možné určit výstup a stav soustavy v libovolném okamžiku $t > t_0$. Soustava vykazující tyto vlastnosti se nazývá deterministická.

Naopak soustava je stochastická, jestliže je možné její výstup a stav v libovolném okamžiku určit pouze s určitou pravděpodobností nebo díky statistickým metodám za předpokladu znalosti jejího stavu v nějakém okamžiku $t > t_0$ a vstupu na polo uzavřeném intervalu $\langle t_0, t \rangle$ (Strejc, 1978).

1.1.1 Nelineární a lineární spojité stavový popis

Jelikož se stavový popis s výhodou často využívá pro soustavy s více vstupy a výstupy, bude uvažována lineární soustava s d vstupními veličinami, e výstupními veličinami a f stavovými proměnnými. Právě taková soustava je vyobrazena na obr. 1.1.



Obr. 1.1 – Vyobrazení obecné soustavy

Existuje předpoklad, že veličiny jsou funkcemi času a lze je zapsat do sloupcových vektorů, jak popisuje rovnice (1.1).

$$\mathbf{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_d(t) \end{bmatrix}, \mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_e(t) \end{bmatrix}, \mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_f(t) \end{bmatrix}. \quad (1.1)$$

Spojité stavový popis soustavy je založen na výpočtu odezvy soustavy $\mathbf{y}(t)$ a stavu soustavy $\mathbf{x}(t)$. Stav spojitě soustavy $\mathbf{x}(t)$ v jakémkoliv zvoleném časovém okamžiku t , lze jednoznačně určit pomocí stavu soustavy v čase t_0 , tj. hodnoty $\mathbf{x}(t_0)$ a hodnot vstupního vektoru $\mathbf{u}(t)$, který působil na soustavu od okamžiku t_0 do t . Zapsáno rovnicí

$$\mathbf{x}(t) = \mathbf{X}[\mathbf{x}(t_0), \mathbf{u}], \quad (1.2)$$

kde $\mathbf{x}(t)$ – vypočítaný stav soustavy v čase t ,

$\mathbf{x}(t_0)$ – předchozí stav soustavy v čase t_0 ,

$\mathbf{u}(t)$ – zobrazení \mathbf{u} na intervalu $\langle t_0, t \rangle$ v prostoru \mathbf{R}^d .

Odezvu soustavy $y(t)$ je možné vypočítat na základě znalosti působení vstupního vektoru $u(t)$, a to od okamžiku t_0 do t , a rovněž díky znalosti vnitřního stavu $x(t)$. A tudíž je vektor výstupních hodnot soustavy dán rovnicí

$$y(t) = Y[x(t), u], \quad (1.3)$$

kde $y(t)$ – vypočítaná odezva soustavy v čase t ,

$x(t)$ – vnitřní stav soustavy v čase t ,

$u()$ – zobrazení u na intervalu $\langle t_0, t \rangle$ v prostoru \mathbf{R}^d .

Dynamické vlastnosti soustavy znázorněné na obr. 1.1, lze popsat následující soustavou diferenciálních rovnic prvního řádu

$$\dot{x}_p(t) = f_p(x_1, x_2, \dots, x_f, u_1, u_2, \dots, u_d, t), \quad (1.4)$$

kde $p = 1, 2, \dots, f$,

x_1, x_2, \dots, x_f – jednotlivé složky vektoru stavových proměnných soustavy,

u_1, u_2, \dots, u_d – jednotlivé hodnoty vektoru vstupních hodnot,

t – čas.

$$y_r(t) = g_r(x_1, x_2, \dots, x_f, u_1, u_2, \dots, u_d, t), \quad (1.5)$$

kde $r = 1, 2, \dots, e$,

x_1, x_2, \dots, x_f – jednotlivé složky vektoru stavových proměnných soustavy,

u_1, u_2, \dots, u_d – jednotlivé hodnoty vektoru vstupních hodnot,

t – čas.

Výše uvedená rovnice (1.4) je stavovou rovnicí, potom vztah (1.5) popisuje rovnici výstupní. Obě rovnice lze zapsat v maticovém tvaru

$$\begin{aligned} \dot{x}(t) &= f[x(t), u(t), t], \\ y(t) &= g[x(t), u(t), t], \end{aligned} \quad (1.6)$$

kde f, g – obecně nelineární funkce,

$x(t)$ – vektor stavových proměnných soustavy,

$u(t)$ – vektor vstupních hodnot,

t – čas.

Je zaveden předpoklad o funkcích f a g v soustavě rovnic (1.6), že se jedná o funkce spojitě diferencovatelné a zároveň existují jejich parciální derivace podle jednotlivých prvků vektorů $\mathbf{u}(t)$ a $\mathbf{y}(t)$ tj. $\frac{\partial f_v}{\partial x_w}$, pro které platí, že $v, w = 1, \dots, ch$ a $\frac{\partial f_v}{\partial u_w}$, kde $v = 1, \dots, ch$ a $w = 1, \dots, q$. V případě, že výstup soustavy $\mathbf{y}(t)$ závisí pouze na aktuálních hodnotách stavových veličin $\mathbf{x}(t)$ a vstupní veličiny $\mathbf{u}(t)$ nejsou v argumentu funkce g vůbec zastoupeny, soustava pak splňuje silnou podmínku fyzikální realizovatelnosti. Pokud ale argument funkce g obsahuje i aktuální hodnoty $\mathbf{u}(t)$, soustava splňuje slabou podmínku fyzikální realizovatelnosti. Obecně platí, že většina reálných soustav splňuje silnou podmínku fyzikální realizovatelnosti (Dostál, 2010, Balátě, 2004, Blaha, 2010).

Ve vztahu (1.6) jsou funkce f, g uvažovány jako nelineární, ale pokud se však jedná o soustavu lineární, je možné rovnice (1.4), (1.5), přepsat do tvaru

$$\dot{\mathbf{x}}_p(t) = a_{p1}(t) \cdot x_1(t) + \dots + a_{pw}(t) \cdot x_w(t) + b_{p1}(t) \cdot u_1(t) + \dots + b_{pq}(t) \cdot u_q(t), \quad (1.7)$$

$$\mathbf{y}_r(t) = c_{r1}(t) \cdot x_1(t) + \dots + c_{rw}(t) \cdot x_w(t) + d_{r1}(t) \cdot u_1(t) + \dots + d_{rq}(t) \cdot u_q(t), \quad (1.8)$$

a odtud vznikne následující vektorově maticový zápis

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}(t) \cdot \mathbf{x}(t) + \mathbf{B}(t) \cdot \mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}(t) \cdot \mathbf{x}(t) + \mathbf{D}(t) \cdot \mathbf{u}(t), \end{aligned} \quad (1.9)$$

kde $\mathbf{A}(t)$ – matice soustavy o rozměru $(f \times f)$,

$\mathbf{B}(t)$ – matice buzení o rozměru $(f \times d)$,

$\mathbf{C}(t)$ – matice výstupní o rozměru $(e \times f)$,

$\mathbf{D}(t)$ – matice převodu o rozměru $(e \times d)$,

$\mathbf{x}(t)$ – hodnota stavové veličiny v aktuálním čase,

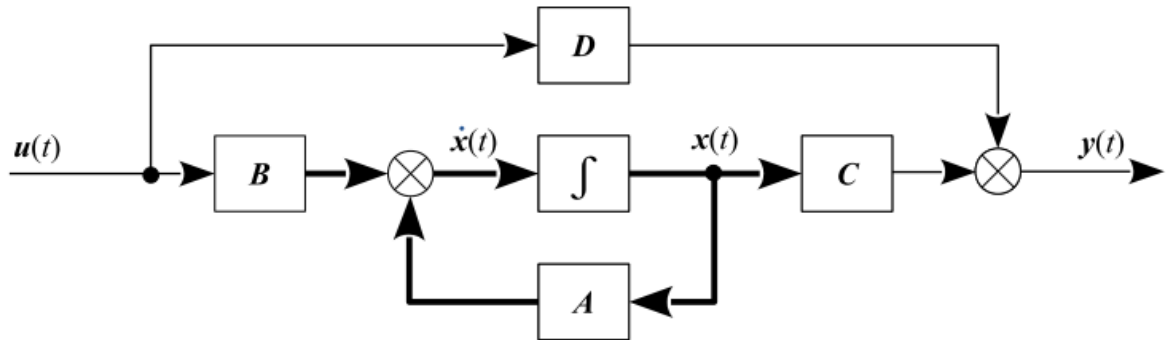
$\mathbf{u}(t)$ – aktuální hodnota akčního zásahu,

$\mathbf{y}(t)$ – hodnota na výstupu soustavy v aktuálním kroku.

Jelikož bude uvažována stacionární regulovaná soustava, tedy soustava, jejíž parametry jsou v čase neměnné lze rovnici (1.9) upravit na

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A} \cdot \mathbf{x}(t) + \mathbf{B} \cdot \mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C} \cdot \mathbf{x}(t) + \mathbf{D} \cdot \mathbf{u}(t). \end{aligned} \quad (1.10)$$

Právě taková soustava je vyobrazena formou blokového schéma na obr. 1.2.



Obr. 1.2 – Blokové schéma spojitého stavového popisu

1.1.2 Přenosová matice lineární soustavy ze spojitého stavového popisu

Přenosová matice $G(s)$ soustavy udává vztah mezi vektorem Laplaceových obrazů výstupu a vektorem Laplaceových obrazů vstupu soustavy. Každý její prvek je tvořen odpovídajícím přenosem. Je definována vztahem

$$Y(s) = G(s) \cdot U(s), \quad (1.11)$$

kde $Y(s)$ – Laplaceovův obraz výstupní veličiny $y(t)$ o rozměru $(e \times 1)$,

$G(s)$ – přenosová matice o rozměru $(e \times d)$,

$U(s)$ – Laplaceovův obraz vstupní veličiny $u(t)$ o rozměru $(d \times 1)$.

Uváží-li se, že obecný tvar přenosové matice je ve tvaru

$$G(s) = \begin{bmatrix} G_{11} & \cdots & G_{1d} \\ \vdots & \ddots & \vdots \\ G_{e1} & \cdots & G_{ed} \end{bmatrix}, \quad (1.12)$$

kde jednotlivé přenosy jsou pak dány vztahem

$$G_{ij} = \frac{Y_i(s)}{U_j(s)}, \text{ kde } i = 1, 2, \dots, e; j = 1, 2, \dots, d. \quad (1.13)$$

Pak je tedy vždy vypočítán příslušný přenos mezi i -tou výstupní a j -tou vstupní veličinou. Operátorový přenos je definován jako podíl Laplaceových obrazů výstupu a vstupu. Jelikož jsou však U a Y matice, nelze tento podíl provést přímo, protože není v případě vektorů a matic definován. Je nezbytné provést úpravu stavového modelu z vnitřního popisu na vnější. Tím dojde k vyloučení vektoru stavových veličin $X(s)$. Je-li soustava plně pozorovatelná i

řiditelná, vizte pododdíly 1.2.1 a 1.2.2, lze vyjít ze stavové rovnice a rovnice výstupu (1.10).

Po provedení Laplaceovy transformace těchto rovnic bylo získáno

$$s \cdot X(s) - \mathbf{x}(0) = \mathbf{A} \cdot X(s) + \mathbf{B} \cdot U(s), \quad (1.14)$$

$$Y(s) = \mathbf{C} \cdot X(s) + \mathbf{D} \cdot U(s), \quad (1.15)$$

a při zajištění nulových počátečních podmínek $\mathbf{x}(0) = 0$, které jsou předpokladem definice přenosu, lze přepsat stavovou rovnici (1.14) jako

$$(s \cdot \mathbf{I} - \mathbf{A}) \cdot X(s) = \mathbf{B} \cdot U(s), \quad (1.16)$$

následně pro vektor stavových proměnných tedy platí

$$X(s) = (s \cdot \mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{B} \cdot U(s). \quad (1.17)$$

Takto získaný vztah pro vektor stavových veličin je možné dosadit do výstupní rovnice (1.15) a vznikne

$$Y(s) = \left\{ \mathbf{C} \cdot \left((s \cdot \mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{B} + \mathbf{D} \right) \right\} \cdot U(s). \quad (1.18)$$

Při vzájemném porovnání rovnic (1.11) a (1.18) lze analogicky odvodit vztah pro výpočet přenosové matice $\mathbf{G}(s)$, který se nachází ve složených závorkách rovnice (1.18). Je nezbytné provést výpočet inverze matice $(s \cdot \mathbf{I} - \mathbf{A})^{-1}$, a to pomocí podílu adjungované matice a determinantu původní matice

$$\mathbf{G}(s) = \mathbf{C} \cdot \left(\frac{\text{adj}(s \cdot \mathbf{I} - \mathbf{A})}{\det(s \cdot \mathbf{I} - \mathbf{A})} \cdot \mathbf{B} + \mathbf{D} \right). \quad (1.19)$$

Protože se výraz $\det(s \cdot \mathbf{I} - \mathbf{A})$ nachází v rovnici (1.19) ve jmenovateli přenosové matice $\mathbf{G}(s)$, je možné ho připodobnit k charakteristickému polynomu soustavy. Pokud se tento determinant položí roven nule,

$$\det(s \cdot \mathbf{I} - \mathbf{A}) = 0, \quad (1.20)$$

pak vypočtené kořeny jsou póly soustavy, jelikož póly soustavy jsou kořeny jmenovatele přenosové matice. Z toho vyplývá, že póly přenosové matice $\mathbf{G}(s)$, odpovídají vlastním číslům matice \mathbf{A} (Balátě, 2004).

1.1.3 Nelineární a lineární diskrétní stavový popis

Diskrétní dynamické soustavy jsou takové, ve kterých figurují posloupnosti událostí, jež nastávají v příslušných časových okamžicích. Konkrétně soustava vydává posloupnost výstupních veličin $y(k)$, jako odezvu na posloupnost vstupních veličin $u(k)$. Časové okamžiky jsou zpravidla dány velikostí tzv. vzorkovací periody T , díky které je možné definovat okamžik času $t = k \cdot T$. Hodnoty jednotlivých veličin mimo tyto časové okamžiky jsou irelevantní. Hodnota stavu v následujícím časovém kroku je popsána funkcí změny stavu

$$\mathbf{x}(k+1) = \varphi(k+1, k, \mathbf{x}(k), \mathbf{u}(k)), \quad (1.21)$$

kde $\mathbf{x}(k+1)$ – hodnota stavu v kroku $k+1$,
 φ – funkce změny stavu,
 $\mathbf{u}(k)$ – vektor vstupních hodnot.

Po následném přeznačení rovnice (1.21) vznikne

$$\varphi(k+1, k, \mathbf{x}, \mathbf{u}) = f(\mathbf{x}(k), \mathbf{u}(k), k). \quad (1.22)$$

Poté mohly být sepsány rovnice diskrétního stavového popisu ve tvaru

$$\begin{aligned} \mathbf{x}(k+1) &= f(\mathbf{x}(k), \mathbf{u}(k), k), \\ \mathbf{y}(k) &= g(\mathbf{x}(k), \mathbf{u}(k), k), \end{aligned} \quad (1.23)$$

kde f, g – obecně nelineární funkce,
 $\mathbf{x}(k)$ – vektor stavových proměnných,
 $\mathbf{u}(k)$ – vektor vstupních hodnot.

Jestliže se jedná o diskrétní lineární soustavu, pak jsou funkce f, g rovněž lineární a to jak vůči stavu, tak i vůči řízení.

Poté lze zapsat rovnice popisující lineární diskrétní soustavu

$$\mathbf{x}(k+1) = \mathbf{M} \cdot \mathbf{x}(k) + \mathbf{N} \cdot \mathbf{u}(k), \quad (1.24)$$

$$\mathbf{y}(k) = \mathbf{C} \cdot \mathbf{x}(k) + \mathbf{D} \cdot \mathbf{u}(k), \quad (1.25)$$

kde \mathbf{M} – matice diskrétní soustavy o rozměru $(f \times f)$,

\mathbf{N} – matice řízení o rozměru $(f \times d)$,

\mathbf{C} – výstupní matice o rozměru $(e \times f)$,

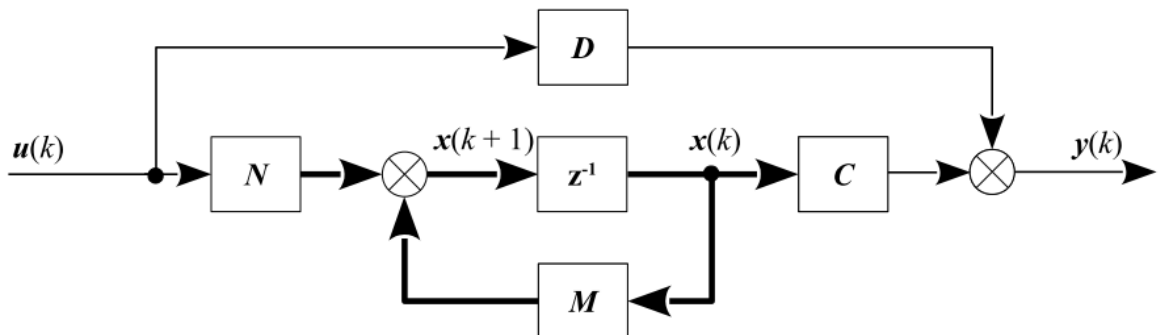
\mathbf{D} – matice vazeb vstupu na výstup o rozměru $(e \times d)$,

$\mathbf{x}(k)$ – vektor stavových proměnných,

$\mathbf{u}(k)$ – vektor vstupních hodnot,

$\mathbf{y}(k)$ – vektor výstupních hodnot,

jejichž analogie se spojitým stavovým popisem je zřejmá. Diskrétní stavový popis soustavy je vyobrazen na obr. 1.3.



Obr. 1.3 – Blokové schéma diskrétního stavového popisu

1.1.4 Matice přenosu lineární soustavy z diskrétního stavového popisu

Stejně jako v případě spojité soustavy, kdy byla určována přenosová matice $G(s)$, lze analogicky pro diskrétní soustavu vypočítat matici přenosu $G(z)$. Aplikací Z-transformace na soustavu rovnic (1.24) a (1.25) je následně získáno

$$z \cdot X(z) = \mathbf{M} \cdot X(z) + \mathbf{N} \cdot U(z), \quad (1.26)$$

$$Y(z) = \mathbf{C} \cdot X(z) + \mathbf{D} \cdot U(z), \quad (1.27)$$

kde \mathbf{M} – matice diskrétní soustavy o rozměru $(f \times f)$,

\mathbf{N} – matice řízení o rozměru $(f \times d)$,

\mathbf{C} – výstupní matice o rozměru $(e \times f)$,

\mathbf{D} – matice vazeb vstupu na výstup o rozměru $(e \times d)$,

$X(z)$ – Laplaceovův obraz stavu,

$U(z)$ – Laplaceovův obraz akční veličiny,

$Y(z)$ – Laplaceovův obraz výstupních veličiny.

Potom po vytknutí výrazu $X(z)$ z rovnice (1.26) vznikla rovnice

$$(z \cdot \mathbf{I} - \mathbf{M}) \cdot X(z) = \mathbf{N} \cdot U(z), \quad (1.28)$$

v němž bylo provedeno osamostatnění $X(z)$ díky vynásobení celé rovnice výrazem

$(z \cdot \mathbf{I} - \mathbf{M})^{-1}$ zleva, pak tedy bylo možné zapsat

$$X(z) = (z \cdot \mathbf{I} - \mathbf{M})^{-1} \cdot \mathbf{N} \cdot U(z). \quad (1.29)$$

Tak jako tomu bylo u spojité soustavy, bylo pro výpočet inverze použito determinantu a algebraického doplňku výrazu $(z \cdot \mathbf{I} - \mathbf{M})$ a tudíž po dosazení vztahu (1.29) do rovnice (1.27) byl odvozen výpočet výstupní rovnice diskrétní soustavy

$$Y(z) = \left\{ \mathbf{C} \cdot \left((z \cdot \mathbf{I} - \mathbf{M})^{-1} \cdot \mathbf{N} + \mathbf{D} \right) \right\} \cdot U(z). \quad (1.30)$$

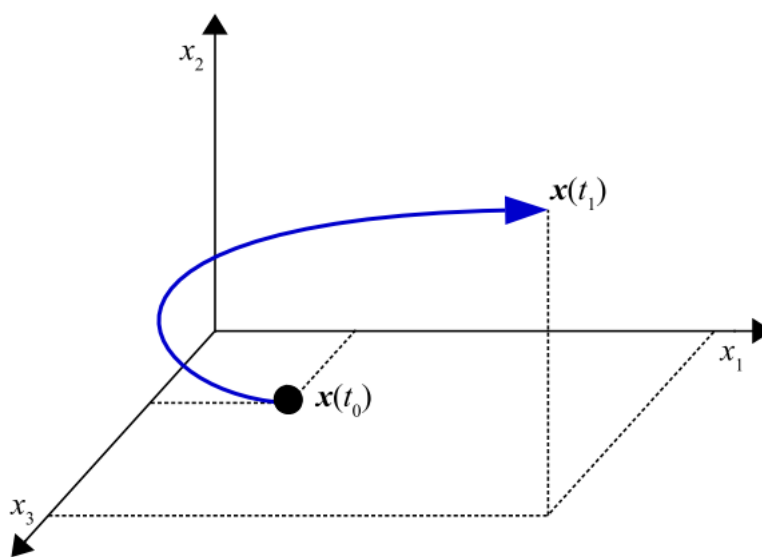
Výraz pro výpočet matice přechodu diskrétní soustavy je opět obsažen ve složených závorkách rovnice (1.30)

$$G(z) = C \cdot \left(\frac{\text{adj}(z \cdot I - M)}{\det(z \cdot I - M)} \cdot N + D \right) \quad (1.31)$$

1.1.5 Stavová trajektorie

Jak už bylo zmíněno v oddíle 1.1.1, při použití vnitřního stavového popisu je v každé soustavě definována abstraktní stavová veličina. Vývoj dané soustavy v čase může být graficky popsán ve f -rozměrném euklidovském prostoru E_f , který je rovněž nazýván stavovým prostorem. Stavový prostor je určen za pomoci souřadnic x_1, x_2, \dots, x_f . Každý bod náležící do stavového prostoru popisuje příslušný stav soustavy.

Stavová trajektorie je pak křivka ve stavovém prostoru, která protíná všechny stavy soustavy. Ty jsou definovány souřadnicemi x_1, x_2, \dots, x_f kterých soustava při své činnosti postupně dosáhla. Stavová trajektorie je ohraničena dvěma body představující počáteční a koncový stav soustavy. Při sestavení stavové trajektorie je čas nezávisle proměnnou. Ukázku stavové trajektorie ve trojrozměrném euklidovském prostoru zachycuje obr. 1.4.



Obr. 1.4 – Nákres stavové trajektorie v trojrozměrném prostoru

(Modrlák, 2004, Švarc, 2003).

1.1.6 Sestavení matic popisujících diskretní stavový popis

V předchozích pododdílech byl podrobně popsán diskretní stavový popis soustavy odpovídající rovnicím (1.24) a (1.25). Současný pododdíl pojednává o výpočtu stavových matic soustavy M, N, C a D z obecné diferenční rovnice f -tého řádu. Jinými slovy o přepočtu vnějšího popisu soustavy na stavové rovnice a tedy popis vnitřní. Pro dosažení těchto matic jsou rozlišovány jednotlivé metody, tzv. normální formy. Určitá forma může být zvolena a následně s výhodou použita pro konkrétní soustavu, přičemž dojde k usnadnění analýzy vlastností dané soustavy nebo ke zjednodušení její struktury díky použití menšího počtu parametrů.

Nespornou výhodou normálních forem je bezprostřední souvislost prvků matic a vektorů stavového modelu soustavy s koeficienty diferenční rovnice. Ke zjednodušení dochází buď vůči vstupní, výstupní, nebo stavové veličině soustavy. Byla uvažována soustava s jedním vstupem a rovněž s jedním výstupem. Je předpoklad na absenci nezkrácených kořenových činitelů, nul a pólů, v čitateli a ve jmenovateli přenosu. Přítomnost nezkrácených nul a pólů, by nemusela být zcela zřejmá a výsledná realizace soustavy by byla vyššího řádu, než je řád přenosu soustavy. Pak by se však nejednalo o realizaci minimální. Stavové rovnice budou následně získány pomocí normální formy říditelnosti, normální formy rekonstruovatelnosti, normální formy pozorovatelnosti a Jordanova základního tvaru.

Normální forma říditelnosti neboli tzv. Frobeniův základní tvar, vychází z vnějšího popisu a tedy z diferenční rovnice

$$\begin{aligned} a_f \cdot \mathbf{y}(k+f) + a_{f-1} \cdot \mathbf{y}(k+f-1) + \dots + a_0 \cdot \mathbf{y}(k) &= b_{f-1} \cdot \mathbf{u}(k+f-1) + \\ b_{f-2} \cdot \mathbf{u}(k+f-2) + \dots + b_0 \cdot \mathbf{u}(k), \end{aligned} \quad (1.32)$$

ze které, lze sestavit operátorový přenos za předpokladu, že koeficient a_f je roven jedné

$$G(z) = \frac{b_0 + b_1 \cdot z + \dots + b_{n-1} \cdot z^{n-1}}{a_0 + a_1 \cdot z + \dots + a_{n-1} \cdot z^{n-1} + z^n}, \quad (1.33)$$

kde b_0, b_1, \dots, b_{f-1} – koeficienty vstupního polynomu soustavy,

a_0, a_1, \dots, a_{f-1} – koeficienty výstupního polynomu soustavy.

Nejprve byly zvoleny stavové veličiny za pomoci difference výstupu soustavy $\mathbf{y}(k)$

$$\begin{aligned} \mathbf{x}_1(k) &= \mathbf{y}(k), \\ \mathbf{x}_2(k) &= \mathbf{y}(k+1), \\ \mathbf{x}_3(k) &= \mathbf{y}(k+2), \\ &\vdots \\ \mathbf{x}_f(k) &= \mathbf{y}(k+f-1), \end{aligned} \quad (1.34)$$

v této souvislosti lze stavové veličiny přepsat jako

$$\begin{aligned} \mathbf{x}_1(k+1) &= \mathbf{x}_2(k), \\ \mathbf{x}_2(k+1) &= \mathbf{x}_3(k), \\ \mathbf{x}_3(k+1) &= \mathbf{x}_4(k), \\ &\vdots \\ \mathbf{x}_f(k+1) &= -a_{f-1} \cdot \mathbf{x}_f - \dots - a_1 \cdot \mathbf{x}_2 - a_0 \cdot \mathbf{x}_1. \end{aligned} \quad (1.35)$$

Matice stavového popisu získané v normální formě říditelnosti jsou

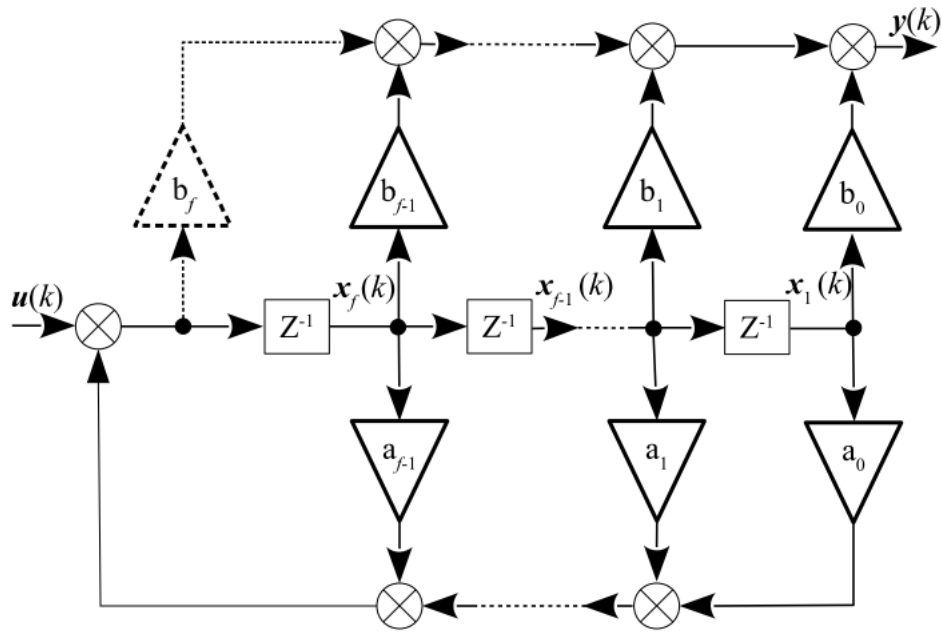
$$\mathbf{M}_{\tilde{R}} = \begin{bmatrix} 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & \dots & -a_{f-1} \end{bmatrix}, \mathbf{N}_{\tilde{R}} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \mathbf{C}_{\tilde{R}} = [b_0 \quad b_1 \quad \dots \quad b_{f-1}]. \quad (1.36)$$

Poté může stavový popis soustavy zapsaný v normální formě říditelnosti definován jako

$$\mathbf{x}_{\check{R}}(k+1) = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & \cdots & -a_{f-1} \end{bmatrix} \cdot \mathbf{x}_{\check{R}}(k) + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \cdot \mathbf{u}(k), \quad (1.37)$$

$$\mathbf{y}_{\check{R}}(k) = [b_0 \ b_1 \ \cdots \ b_{n-1}] \cdot \mathbf{x}_{\check{R}}(k). \quad (1.38)$$

Ve stavové rovnici (1.37) je zřetelná jednoduchost matice řízení. Realizace stavových rovnic (1.37) a (1.38) v normální formě říditelnosti je vyobrazena na obr. 1.5.



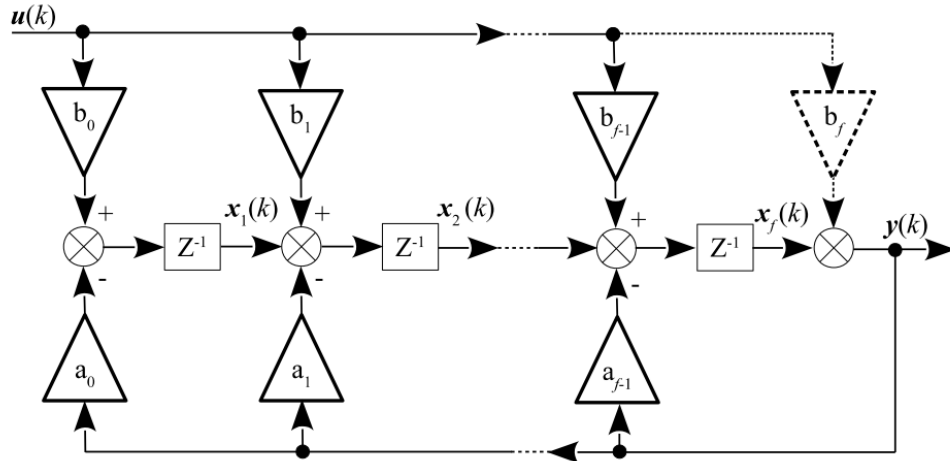
Obr. 1.5 – Blokové schéma stavových rovnic v normální formě říditelnosti

Normální forma rekonstruovatelnosti je tzv. duální forma k normální formě říditelnosti. Zatímco matice získané normální formou říditelnosti jsou označeny indexem \check{R} , tak matice vypočtené v normální formě rekonstruovatelnosti nesou index R . A tudíž tedy pro matici soustavy platí, že $\mathbf{M}_R = \mathbf{M}_{\check{R}}^T$, pro matici řízení $\mathbf{N}_R = \mathbf{C}_{\check{R}}^T$ a pro výstupní matici $\mathbf{C}_R = \mathbf{N}_{\check{R}}^T$. Příslušný stavový popis odvozený v normální formě rekonstruovatelnosti je

$$\mathbf{x}_R(k+1) = \begin{bmatrix} 0 & \cdots & 0 & -a_0 \\ 1 & 0 & 0 & -a_1 \\ 0 & \ddots & \cdots & \vdots \\ 0 & \cdots & 1 & -a_{f-1} \end{bmatrix} \cdot \mathbf{x}_R(k) + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{f-1} \end{bmatrix} \cdot \mathbf{u}(k), \quad (1.39)$$

$$\mathbf{y}_R(k) = [0 \ 0 \ \cdots \ 1] \cdot \mathbf{x}_R(k). \quad (1.40)$$

Normální forma rekonstruovatelnosti je kanonická vzhledem k výstupu, což dokazuje výstupní rovnice soustavy (1.40). Stavové rovnice (1.39) a (1.40) stavového popisu v normální formě rekonstruovatelnosti jsou graficky znázorněny na obr. 1.6.



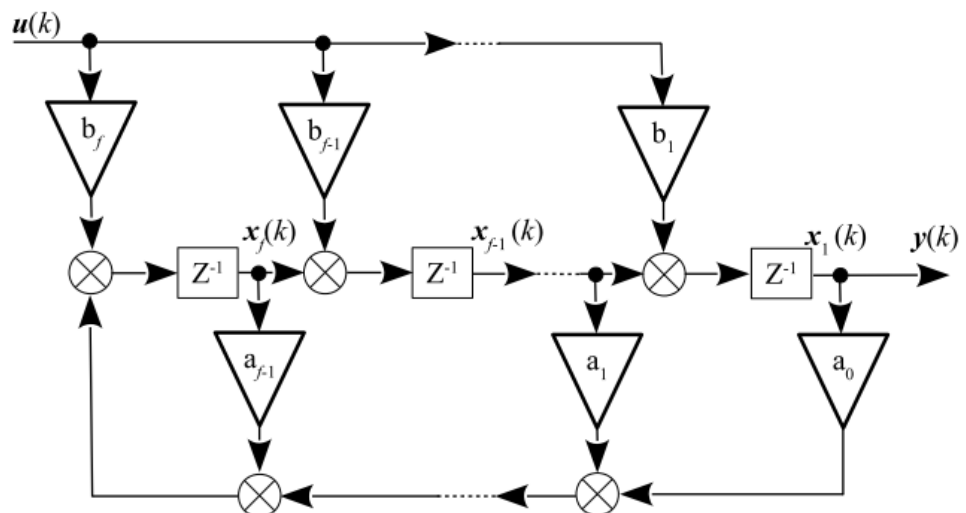
Obr. 1.6 – Blokové schéma stavových rovnic v normální formě rekonstruovatelnosti

Stavový popis odvozený normální formou pozorovatelnosti je ve tvaru

$$\mathbf{x}_p(k+1) = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & \cdots & -a_{f-1} \end{bmatrix} \cdot \mathbf{x}_p(k) + \begin{bmatrix} b_0 \\ \vdots \\ b_{f-2} \\ b_{f-1} \end{bmatrix} \cdot u(k), \quad (1.41)$$

$$y_p(k) = [1 \ 0 \ \cdots \ 0] \cdot \mathbf{x}_p(k). \quad (1.42)$$

Obr. 1.7 vykresluje realizaci stavových rovnic (1.41) a (1.42) v normální formě pozorovatelnosti.



Obr. 1.7 – Blokové schéma stavových rovnic v normální formě pozorovatelnosti

Jordanův základní tvar utváří stavový popis soustavy tak, aby bylo dosaženo jednoduchosti matice \mathbf{M}_J . Byla opět uvažována rovnice stavová (1.24) a výstupní (1.25). Jordanova matice \mathbf{M}_J je maticí diagonální, kdy prvky umístěné na hlavní diagonále jsou kořeny charakteristického polynomu příslušné soustavy. Je zde předpoklad, že vlastní čísla matice \mathbf{M}_J jsou vzájemně různá a nenásobná. A tedy obecný stavový popis vypočtený na základě Jordanova tvaru je

$$\mathbf{x}_J(k+1) = \begin{bmatrix} z_1 & 0 & \cdots & 0 \\ 1 & z_2 & \cdots & 0 \\ 0 & \cdots & \ddots & \vdots \\ 0 & \cdots & 0 & z_f \end{bmatrix} \cdot \mathbf{x}_J(k) + \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \cdot \mathbf{u}(k), \quad (1.43)$$

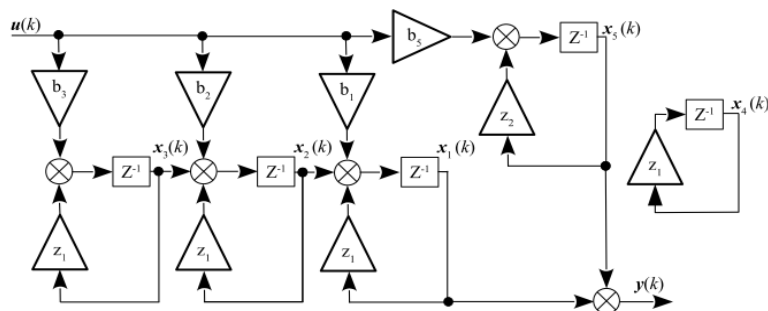
$$\mathbf{y}_J(k) = [b_0 \quad b_1 \quad \cdots \quad b_{f-1}] \cdot \mathbf{x}_J(k). \quad (1.44)$$

Pokud je např. Jordanova forma stavových rovnic ve tvaru

$$\mathbf{x}_J(k+1) = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix} \cdot \mathbf{x}_J(k) + \begin{bmatrix} 4 \\ 5 \\ 12 \\ 0 \\ 6 \end{bmatrix} \cdot \mathbf{u}(k), \quad (1.45)$$

$$\mathbf{y}_J(k) = [1 \quad 0 \quad 0 \quad 0 \quad 1] \cdot \mathbf{x}_J(k). \quad (1.46)$$

Pak se tato soustava skládá jak z dosažitelné a pozorovatelné složky představované char. polynomem $(z+1)^3 \cdot (z-3)$, tak i z nedosažitelné a nepozorovatelné části s charakteristickým polynomem $(z+1)$. Na obr. 1.8 lze vidět, že nedosažitelná a nepozorovatelná část soustavy není nijak spojena se vstupem anebo výstupem soustavy (Strejc, 1978, Štecha, 2005).



Obr. 1.8 – Blokové schéma příkladu Jordanova základního tvaru

1.2 VLASTNOSTI STAVOVÉ REALIZACE

Pro návrh regulátoru jsou podstatné některé ze základních vlastností soustav. Patří mezi ně dosažitelnost, říditelnost, stabilizovatelnost, dále pozorovatelnost, rekonstruovatelnost, detekovatelnost a indentifikovatelnost. Ty zde budou postupně definovány a budou popsány způsoby jejich určování. Odvození pravidel pro určování jednotlivých vlastností bude založeno na diskretním stavovém popisu soustavy

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{M} \cdot \mathbf{x}(k) + \mathbf{N} \cdot \mathbf{u}(k), \\ \mathbf{y}(k) &= \mathbf{C} \cdot \mathbf{x}(k) + \mathbf{D} \cdot \mathbf{u}(k). \end{aligned} \quad (1.47)$$

1.2.1 Dosažitelnost, říditelnost a stabilizovatelnost

Stav $\mathbf{x}(t_1)$ lineární soustavy je dosažitelný, pokud existuje časový okamžik $t_1 > t_0$, kde $(t_1 - t_0)$ je konečným intervalem a vstup $\mathbf{u}(t)$, kterým je možné soustavu převést z počátečního stavu $\mathbf{x}(t_0) = 0$ do požadovaného stavu $\mathbf{x}(t_1)$.

Lineární soustava je úplně dosažitelná v čase t , to znamená že, každý stav $\mathbf{x}(t) \in X$ je dosažitelný. Pokud je na dosažitelnost vyšetřována spojitá stacionární soustava a tato vlastnost je potvrzena, každý dosažitelný stav této soustavy je rovněž říditelný. Dosažitelnost stavu u stacionárních diskretních soustav nezávisí na časovém okamžiku t_1 . Vyjde-li se z obecného zápisu stavové rovnice diskretní soustavy

$$\begin{aligned} \mathbf{x}(k) &= \mathbf{M}^k \cdot \mathbf{x}(0) + \sum_{h=0}^{k-1} \mathbf{M}^j \mathbf{N} \cdot \mathbf{u}(k-h-1) + \sum_{h=0}^{k-1} \mathbf{M}^j \cdot \mathbf{v}(k-h-1) = \\ & \mathbf{M}^k \cdot \mathbf{x}(0) + \sum_{l=0}^{k-1} \mathbf{M}^{k-l-1} [\mathbf{N} \cdot \mathbf{u}(l) + \mathbf{v}(l)], \end{aligned} \quad (1.48)$$

kde $\mathbf{x}(0)$ – počáteční stavový vektor,

$\mathbf{v}(l)$ – vektor poruchové veličiny,

$\mathbf{u}(l)$ – vektor akční veličiny.

Následně z (1.48) vplyne rovnice

$$\begin{aligned} \mathbf{x}(k) &= \\ & \mathbf{M}^k \cdot \mathbf{x}(0) + \mathbf{M}^{k-1} \mathbf{N} \cdot \mathbf{u}(0) + \mathbf{M}^{k-2} \mathbf{N} \cdot \mathbf{u}(1) + \dots + \mathbf{M} \mathbf{N} \cdot \mathbf{u}(k-2) + \mathbf{N} \cdot \mathbf{u}(k-1), \end{aligned} \quad (1.49)$$

$$\mathbf{x}(k) = \mathbf{M}^k \cdot \mathbf{x}(0) + [\mathbf{N}, \mathbf{M} \mathbf{N}, \dots, \mathbf{M}^{k-1} \mathbf{N}] \times [\mathbf{u}(k-1), \mathbf{u}(k-2), \dots, \mathbf{u}(0)]^T. \quad (1.50)$$

Během zkoumání dosažitelnosti stavu $\mathbf{x}(k)$ byla za $\mathbf{x}(0)$ dosazena nula, a tudíž dojde k vyrušení prvního členu pravé strany rovnice (1.50). Každý stav $\mathbf{x}(f)$ bude pak dosažitelný v f krocích při působení posloupnosti akční veličiny $\mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(f-1)$, který náleží do prostoru nad vektory $\mathbf{N}, \mathbf{MN}, \dots, \mathbf{M}^{f-1}\mathbf{N}$. Přičemž se musí jednat o vektory lineárně nezávislé, aby bylo možné převést diskrétní stacionární soustavu do požadovaného stavu $\mathbf{x}(f)$. Proměnná f zastupuje rozměr stavového prostoru. Stav stacionární soustavy, je dosažitelný právě tehdy, když hodnota matice

$$\begin{aligned} \mathbf{V}_D^S &= [\mathbf{A}, \mathbf{AB}, \dots, \mathbf{A}^{f-1}\mathbf{B}], \\ \mathbf{V}_D^D &= [\mathbf{N}, \mathbf{MN}, \dots, \mathbf{M}^{f-1}\mathbf{N}], \end{aligned} \quad (1.51)$$

je rovna hodnotě rozměru f stavového prostoru. Potom platí že, žádný stav $\mathbf{x}(k)$ nemůže být dosažen při $k < f$. Je dosažitelný s $k = f$, ale obecně nemusí platit, že je dosažitelný při $k > f$, jako k tomu dochází v například u nelineárních soustav s omezeným vstupem. Soustavu lze prohlásit za dosažitelnou, pokud každý její stav je dosažitelný.

Stav $\mathbf{x}(t_1)$ lineární soustavy je říditelný, pokud existuje časový okamžik $t_2 > t_1$ a takový vstup soustavy $\mathbf{u}(t)$, díky jehož působení se soustava převede ze stavu $\mathbf{x}(t_1)$ do stavu $\mathbf{x}(t_2) = 0$, na konečném intervalu $(t_2 - t_1)$.

Lineární soustava je úplně říditelná v čase t , to znamená že, každý stav $\mathbf{x}(t) \in X$ je říditelný. Jak bylo uvedeno výše, v případě spojitých stacionárních soustav se vyšetřování omezuje pouze na ověření říditelnosti, protože není nutno dosažitelnost a říditelnost rozlišovat. Říditelnost stavu u stacionárních diskrétních soustav nezávisí na časovém okamžiku t_1 . Podmínky říditelnosti budou následně odvozeny z rovnice (1.49). Podle výše uvedené definice říditelnosti platí, že $\mathbf{x}(k) = \mathbf{x}(f) = 0, \mathbf{x}(0) \neq 0$ a jsou tedy získány rovnice

$$\mathbf{M}^f \cdot \mathbf{x}(0) = -[\mathbf{N}, \mathbf{MN}, \dots, \mathbf{M}^{f-1}\mathbf{N}] \times [\mathbf{u}(f-1), \dots, \mathbf{u}(0)]^T, \quad (1.52)$$

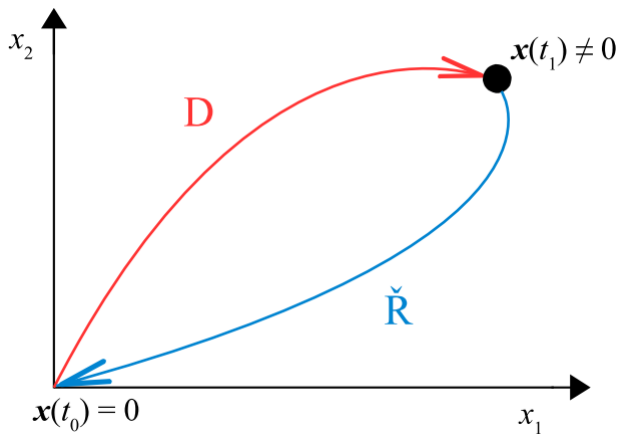
$$\mathbf{x}(0) = -[\mathbf{M}^{-f}\mathbf{N}, \mathbf{M}^{-f+1}\mathbf{N}, \dots, \mathbf{M}^{-1}\mathbf{N}] \times [\mathbf{u}(f-1), \dots, \mathbf{u}(0)]^T = -\mathbf{V}_R^D \cdot \mathbf{u}_f(k) \quad (1.53)$$

Aby platilo, že je soustava říditelná, jinými slovy že je převeditelná ze stavu $\mathbf{x}(0) \neq 0$ v f krocích za působení vstupního vektoru $\mathbf{u}(0), \dots, \mathbf{u}(f-1)$, do stavu $\mathbf{x}(f) = 0$, pak se zřejmě stav $\mathbf{x}(0)$ nachází v prostoru nad vektory $\mathbf{M}^{-f}\mathbf{N}, \mathbf{M}^{-f+1}\mathbf{N}, \dots, \mathbf{M}^{-1}\mathbf{N}$. Zároveň je zachována skutečnost, že uvedené vektory jsou navzájem lineárně nezávislé, v opačném případě by nebylo

možné přejít do stavu $\mathbf{x}(f) = 0$. A tedy stav $\mathbf{x}(0) \neq 0$ je říditelný právě tehdy, pokud hodnota matice

$$\mathbf{V}_R^D = [\mathbf{M}^{-1}\mathbf{N}, \mathbf{M}^{-2}\mathbf{N}, \dots, \mathbf{M}^{-f}\mathbf{N}], \quad (1.54)$$

je rovna rozměru f stavového prostoru. Jinými slovy říditelná soustava je taková soustava, jejíž každý stav je říditelný. Obr. 1.9 zachycuje princip dosažitelnosti a říditelnosti.



Obr. 1.9 – Nákres dosažitelnosti a říditelnosti

Soustavy, které nespĺňují uvedené podmínky dosažitelnosti a říditelnosti mohou i tak v uzavřeném regulačním obvodu s regulátorem poskytovat stabilní regulační pochody. Takové soustavy jsou nazývány stabilizovatelné. Soustava popsaná maticemi \mathbf{M}, \mathbf{N} je stabilizovatelná, pokud existuje reálná matice \mathbf{K} , že výsledná matice $\mathbf{M} - \mathbf{N} \cdot \mathbf{K}$ je stabilní. Všechna vlastní čísla této

matice leží na jednotkové kružnici. Stabilizovatelnost je v podstatě definována podmínkami pro uzavřený regulační obvod s nestabilní soustavou. Ke stabilizaci tedy postačí regulátor, jehož výstup je lineární transformací stavového vektoru

$$\mathbf{u}(k) = -\mathbf{K} \cdot \mathbf{x}(k). \quad (1.55)$$

1.2.2 Pozorovatelnost, rekonstruovatelnost a detekovatelnost

Jelikož mnohdy není stav diskretní soustavy měřitelný a není tedy možné realizovat řízení podle vztahu (1.55). Vystává problém jak z měřitelného výstupu či výstupů vícerozměrné soustavy určit stavový vektor soustavy. V souvislosti s touto problematikou se rozlišuje případ pozorovatelného stavu a rekonstruovatelného stavu.

Stav $\mathbf{x}(t_0)$ lineární soustavy je pozorovatelný, pokud může být určen pomocí budoucích hodnot výstupní veličiny $y(t)$, pro $t > t_0$ a $(t - t_0)$ je konečným intervalem.

Podmínku pozorovatelnosti lze odvodit díky stavové rovnici a rovnici výstupu (1.47). Byly vypočteny hodnoty výstupní veličiny pro $k, k + 1, k + f - 1$ a vzniklo

$$\begin{aligned}
y(k) &= \mathbf{C} \cdot x(k) + \mathbf{D} \cdot u(k), \\
y(k+1) &= \mathbf{C} \times \mathbf{M} \cdot x(k) + \mathbf{C} \times \mathbf{N} \cdot u(k) + \mathbf{D} \cdot u(k+1), \\
y(k+2) &= \mathbf{C} \times \mathbf{M}^2 \cdot x(k) + \mathbf{C} \times \mathbf{M} \times \mathbf{N} \cdot u(k) + \mathbf{C} \times \mathbf{N} \cdot u(k+1) + \mathbf{D} \cdot u(k+2), \\
&\vdots \\
y(k+f-1) &= \mathbf{C} \times \mathbf{M}^{f-1} \cdot x(k) + \mathbf{C} \times \mathbf{M}^{f-2} \times \mathbf{N} \cdot u(k) + \dots + \\
&\mathbf{C} \times \mathbf{M} \times \mathbf{N} \cdot u(k+f-3) + \mathbf{C} \times \mathbf{N} \cdot u(k+f-2) + \mathbf{D} \cdot u(k+f-1),
\end{aligned} \tag{1.56}$$

neboli zapsáno ve vektorově maticovém zápisu

$$\begin{bmatrix} y(k) \\ y(k+1) \\ y(k+2) \\ \vdots \\ y(k+f-1) \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CM} \\ \mathbf{CM}^2 \\ \vdots \\ \mathbf{CM}^{f-1} \end{bmatrix} \cdot \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_f(k) \end{bmatrix} + \begin{bmatrix} \mathbf{D} & 0 & 0 & 0 \\ \mathbf{CN} & \mathbf{D} & 0 & 0 \\ \mathbf{CMN} & \mathbf{CN} & \mathbf{D} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{CM}^{f-2}\mathbf{N} & \mathbf{CM}^{f-3}\mathbf{N} & \dots & \mathbf{D} \end{bmatrix} \cdot \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+f-1) \end{bmatrix}, \tag{1.57}$$

který může být zkráceně přepsán jako

$$y_f(k) = \mathbf{V}_p^D \cdot x(k) + \mathbf{P}_p \cdot u(k), \tag{1.58}$$

kde \mathbf{V}_p^D – matice pozorovatelnosti,

\mathbf{P}_p – matice výstupu.

Avšak u lineárních soustav, můžou být podmínky pozorovatelnosti definovány pouze na základě výstupů $y_f(k)$. Výstupní zobrazení lineárních soustav bylo zapsáno jako

$$\eta(t, \varphi(t; \tau, \mathbf{x}, \omega)) = \eta(t, \varphi(t; \tau, \mathbf{x}, 0)) + \eta(t, \varphi(t; \tau, 0, \omega)), \tag{1.59}$$

kde $\eta()$ – výstupní zobrazení,

t – okamžik času náležící do množiny časových okamžiků T ,

φ – funkce změny stavu,

τ – počáteční časový okamžik,

\mathbf{x} – stav náležící do množiny stavů X ,

ω – vstupní segment náležící do množiny přijatelných vstupních funkcí Ω .

Přičemž první člen na pravé straně rovnice (1.59) zastupuje složku výstupu vyplývajícího z počátečního stavu, kdežto druhý člen vyjadřuje složku vyplývající ze vstupního signálu soustavy. A tudíž po provedení výpočtu

$$\eta(t, \varphi(t; \tau, x_1, \omega)) - \eta(t, \varphi(t; \tau, x_2, \omega)) = \eta(t, \varphi(t; \tau, x_1 - x_2, 0)), \quad (1.60)$$

je zřejmé, že složky odpovídající vstupu ω , se navzájem vyruší a výsledný tvar výstupního zobrazení závisí pouze na rozdílu stavů $x_1 - x_2$.

Pokud bude x_1 brán jako stav počáteční a x_2 jako stav konečný, který bude roven nule, lze počáteční stav vyjádřit pro lineární soustavu jako lineární transformaci budoucích výstupů. Z tohoto důvodu lze zanedbat druhý člen pravé strany rovnice (1.58). Pokud je matice V_P^D regulární, pak jednotlivé její řádky jsou vzájemně lineárně nezávislé a existuje její inverze $(V_P^D)^{-1}$ pak lze napsat

$$\mathbf{x}(k) = (V_P^D)^{-1} \cdot \mathbf{y}_f(k). \quad (1.61)$$

Potom lineární soustava definovaná vztahy (1.47) je pozorovatelná tehdy a pouze tehdy, pokud se hodnota matice

$$V_P^D = \begin{bmatrix} C \\ CM \\ CM^2 \\ \vdots \\ CM^{f-1} \end{bmatrix}, \quad (1.62)$$

rovná rozměru f stavového prostoru.

Stav $\mathbf{x}(t_0)$ lineární soustavy je rekonstruovatelný, pokud je možné ho stanovit pomocí předchozích hodnot výstupní veličiny $y(t)$, pro $t < t_0$ a $(t - t_0)$ je konečným intervalem.

Odvození podmínek rekonstruovatelnosti je založeno na rovnici (1.58) a zároveň nechť je

$$\mathbf{x}(k+f) = \mathbf{M}^f \cdot \mathbf{x}(k). \quad (1.63)$$

Po dosazení rovnice (1.63) získává rovnice (1.58) tvar

$$\mathbf{y}_f(k) = \mathbf{V}_P^D \cdot \mathbf{M}^{-f} \cdot \mathbf{x}(k+f) = \mathbf{V}_R^D \cdot \mathbf{x}(k+f), \quad (1.64)$$

kde \mathbf{V}_P^D – matice pozorovatelnosti,

\mathbf{M} – matice dynamiky soustavy,

\mathbf{V}_R^D – matice rekonstruovatelnosti,

$\mathbf{y}_f(k)$ – vektor výstupů,

$\mathbf{x}(k+f)$ – vektor stavů.

Jelikož matice \mathbf{V}_P^D je regulární maticí, pak po jejím součinu s maticí dynamiky soustavy

\mathbf{M} , je získána opět regulární matice. Existuje tedy její inverze $(\mathbf{V}_R^D)^{-1}$ a je možné napsat

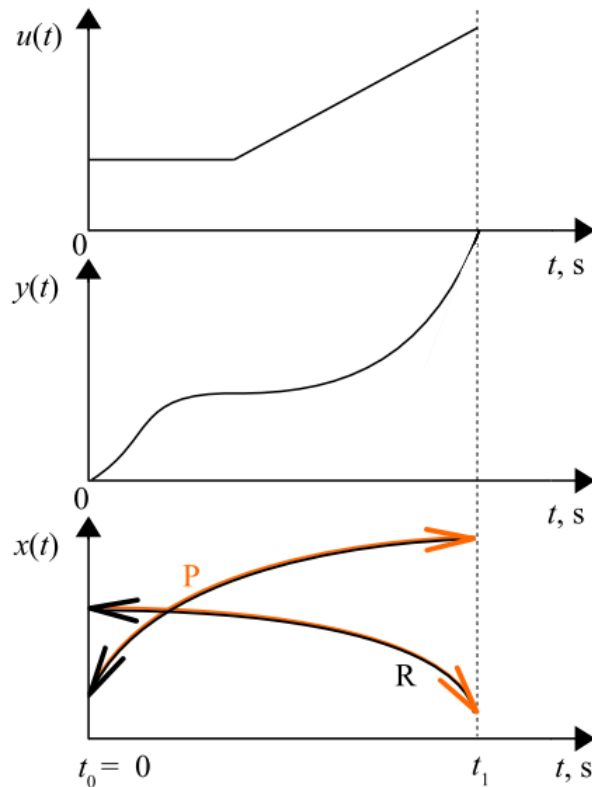
$$\mathbf{x}(k+f) = (\mathbf{V}_R^D)^{-1} \cdot \mathbf{y}_f(k). \quad (1.65)$$

A tedy lineární soustava definovaná rovnicemi (1.47) je rekonstruovatelná tehdy a pouze tehdy, pokud je determinant matice \mathbf{M} nenulový a hodnost matice

$$\mathbf{V}_R^D = \begin{bmatrix} \mathbf{C}\mathbf{M}^{-f} \\ \mathbf{C}\mathbf{M}^{-(f-1)} \\ \mathbf{C}\mathbf{M}^{-(f-2)} \\ \vdots \\ \mathbf{C}\mathbf{M}^{-1} \end{bmatrix}, \quad (1.66)$$

je rovna rozměru f stavového prostoru. Obr. 1.10 zachycuje základní thesi pozorovatelnosti a rekonstruovatelnosti. Jak je zde vidět, pozorovatelné stavy $x(t)$ jsou určeny v čase $t_0 = 0$ z budoucích hodnot výstupu $y(t)$, kdežto rekonstruovatelné stavy $x(t)$ jsou v čase t_1 získány zpětně.

Pokud množina nestabilních stavů soustavy spadá do podprostoru pozorovatelných stavů a některé složky výstupu soustavy jsou nepozorovatelné avšak stabilní, jedná se o soustavu detekovatelnou.



Obr. 1.10 – Náčrt pozorovatelnosti a rekonstruovatelnosti

Dvojice (C, M) soustavy je detekovatelná, pokud existuje reálná matice K , že výsledná matice $M - K \cdot C$ je stabilní, jinými slovy všechna vlastní čísla výsledné matice $M - K \cdot C$ leží na jednotkové kružnici a tedy $|\lambda_i| < 1$.

1.2.3 Identifikovatelnost

Teorie poskytuje nespočet metod sloužících k identifikaci neznámé soustavy. Ať už se jedná o jednorázové naměření vstupních a výstupních veličin soustavy, kde je pak výsledek použit k výpočtu regulátoru, nebo je možné identifikaci provádět opakovaně v čase pro tzv. adaptivní řízení. Každopádně však musí být soustava detekovatelná a stabilizovatelná. Dále aby bylo možné určit matematický model soustavy, musí být soustava rovněž identifikovatelná. Pokud soustava nevykazuje výše uvedené vlastnosti, nelze ji pak identifikovat, a tudíž nalézt optimální řešení jejího řízení. Odvození identifikovatelnosti vychází z tzv. volné soustavy, na kterou nepůsobí žádné vnější vlivy, ta je popsána rovnicí

$$x(k+1) = M \cdot x(k). \quad (1.67)$$

Existuje takový předpoklad, že počáteční stav soustavy $x(k)$ je znám a soustava je pozorovatelná. Problém identifikovatelnosti by se dal definovat jako určování matice \mathbf{M} . Pro soustavu popsanou rovnicí (1.67) bude platit, že

$$\begin{aligned} x(k+1) &= \mathbf{M} \cdot x(k), \\ x(k+2) &= \mathbf{M} \cdot x(k+1) = \mathbf{M}^2 \cdot x(k), \\ &\vdots \\ x(k+f) &= \mathbf{M} \cdot x(k+f-1) = \mathbf{M}^f \cdot x(k). \end{aligned} \tag{1.68}$$

Jelikož jsou podle předpokladu všechny stavové veličiny pozorovatelné, lze po f krocích výpočtů sepsat

$$\begin{bmatrix} x(k+1) \\ x(k+2) \\ \vdots \\ x(k+f) \end{bmatrix} = \mathbf{M} \cdot \begin{bmatrix} x(k) \\ x(k+1) \\ \vdots \\ x(k+f-1) \end{bmatrix}, \quad \begin{bmatrix} x(k) \\ x(k+1) \\ \vdots \\ x(k+f-1) \end{bmatrix} = \begin{bmatrix} x(k) \\ \mathbf{M} \cdot x(k) \\ \vdots \\ \mathbf{M}^{f-1} \cdot x(k) \end{bmatrix}, \tag{1.69}$$

a je tak získána rovnice umožňující definici podmínek identifikovatelnosti.

Matice \mathbf{M} soustavy (1.67) je identifikovatelná tehdy a pouze tehdy, pokud determinant matice identifikovatelnosti

$$\mathbf{V}_1^D = \begin{bmatrix} x(k) \\ \mathbf{M} \cdot x(k) \\ \vdots \\ \mathbf{M}^{f-1} \cdot x(k) \end{bmatrix}, \tag{1.70}$$

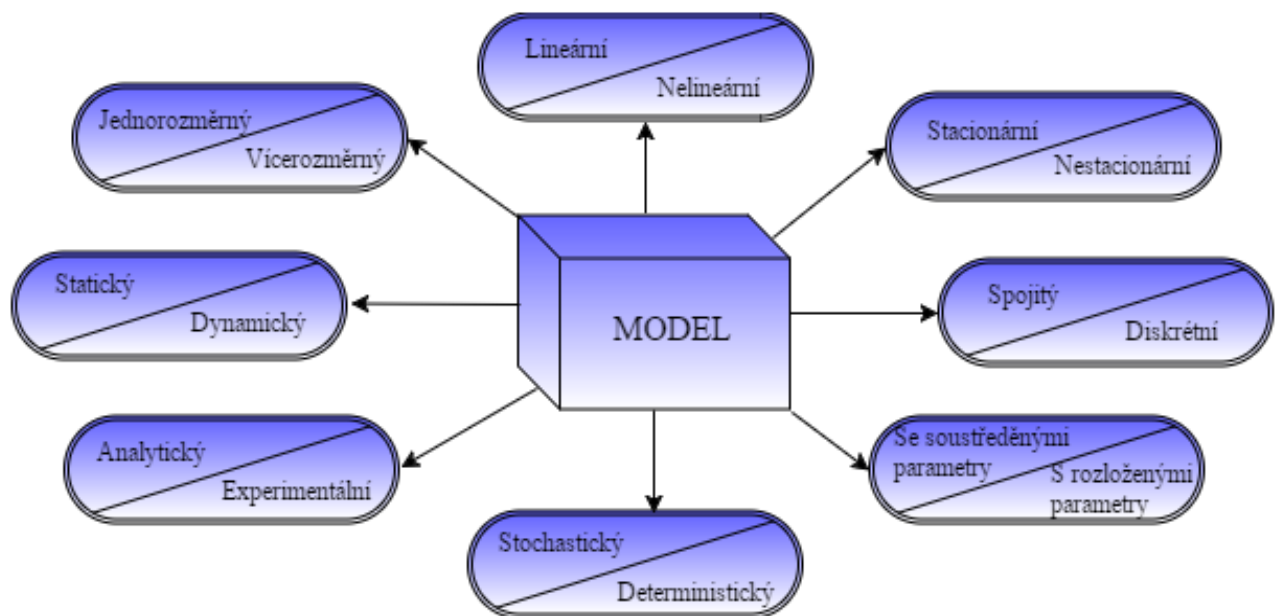
je různý od nuly a jedná se tedy o matici regulární (Strejc, 1978, Balátě, 2004).

1.3 IDENTIFIKACE SOUSTAVY

Aby bylo možné s neznámou soustavou pracovat matematickými prostředky a aby mohla být následně regulována, je nezbytné určit model takovéto soustavy. Tento proces se nazývá identifikace. Takto vzniklý model je možné zkoumat namísto experimentování se skutečnou soustavou, přičemž model by měl svými vlastnostmi odpovídat původní soustavě. Avšak platí, že model obsahuje zjednodušení, a tudíž není naprosto přesný v porovnání s původní soustavou (Balátě, 2004).

1.3.1 Rozdělení modelů

Na základě apriorních informací o soustavě, tj. předem daných a obecně známých poznatků, a rovněž podle záměrů řízení je vyšetřováno, zda je vhodné zkoumanou soustavu definovat jako model: lineární nebo nelineární, statický nebo dynamický, spojitý nebo diskrétní, deterministický nebo stochastický, jednorozměrný nebo vícerozměrný, se soustředěnými nebo s rozloženými parametry, analytický nebo experimentální, nebo jako stacionární či nestacionární (Vrožina, 2012).



Obr. 1.11 – Rozdělení modelů

Přehled typů modelů je vyobrazen na obr 1.11. Konkrétnímu modelu pak vždy náleží právě jedna vlastnost z příslušné větve. Při dodržení tohoto pravidla je počet zvolených větví irelevantní. Následně byl uveden slovní popis jednotlivých vlastností modelu.

Podle charakteru matematického popisu jsou děleny modely na lineární a nelineární. Lineární model je takový, pro který platí, že všechny vzájemné vazby mezi jeho veličinami jsou lineární. Zároveň platí princip superpozice a tudíž výsledná odezva na součet dvou obecných signálů na vstupu soustavy, je totožná se součtem odezev na jednotlivé vstupní signály působící na soustavu každý zvlášť. Model je nelineární, pokud je alespoň jedna rovnice matematického popisu nelineární.

Dále je rozlišováno, zda model popisuje statické či dynamické vlastnosti soustavy. Statický model definuje vztah mezi vstupními a výstupními veličinami soustavy v ustáleném stavu. Díky algebraickým rovnicím, které popisují vazbu mezi vstupem a výstupem soustavy, lze při známém vstupu vypočítat hodnotu výstupu. Jelikož však v těchto rovnicích nefiguruje čas jako nezávisle proměnná, nepodává statický model žádnou informaci o čase, ve kterém bude vypočítaná hodnota výstupu dosažena, a proto není vhodný pro řízení. Na druhé straně dynamický model popisuje nejen statické, ale rovněž i dynamické vlastnosti soustavy. Poskytuje informaci o hodnotě výstupní veličiny v čase při konkrétní hodnotě vstupní a stavové veličiny. Vztah mezi vstupem a výstupem soustavy je popsán diferenciální, popř. diferenční rovnicí.

Dalším dělením je podle způsobu zpracování informace na spojité a nespojité. Model je spojité, pokud se jeho vstupy i výstupy mění plynule, neboli spojitě. U diskrétního modelu dochází v diskrétních okamžicích ke změnám vstupu a výstupu skokově.

Podle charakteru vztahu mezi vstupem a výstupem soustavy, jsou modely členěny na deterministické a stochastické. Pro deterministický model platí, že existují jednoznačné a přesné vztahy popisující vazbu mezi vstupem a výstupem soustavy. Aktuální výstup soustavy je zpravidla ovlivněn hodnotou na vstupu a předchozím stavem. Pokud je znám aktuální vstup, a minulý stav, je možné odhadnout jakýkoliv budoucí stav a výstup soustavy. Stochastický model má náhodný charakter a tudíž vztahy mezi vstupem a výstupem nejsou definovány určitě. Stochastický model může tedy věrněji popisovat reálné děje, kterým je přítomnost nahodilých složek vlastní. Přestože je známa struktura modelu, lze jeho aktuální stav určit pouze s určitou pravděpodobností pomocí statistických metod.

Podle počtu vstupů a výstupů, jsou soustavy děleny na jednorozměrné a vícerozměrné. Jednorozměrné soustavy mají právě jeden vstup a jeden výstup. Co se týče vícerozměrných soustav, ty mají dva a více vstupů a jeden a více výstupů.

Dalším rozdělením je podle rozložení sledované veličiny ve zkoumané soustavě. V modelech se soustředěnými parametry jsou zaváděny předpoklady a odstraňovány závislosti sledovaných veličin na poloze v prostoru. Matematický popis je reprezentován soustavou

obyčejných diferenciálních rovnic, ve kterých nefigurují souřadnice jako nezávisle proměnné. Model s rozloženými parametry je naopak takový model, v jehož matematickém popisu vystupuje alespoň jedna souřadnice jako nezávisle proměnná. Sledované veličiny jsou závislé nejen na čase, ale i na poloze v prostoru. Matematický popis je složen ze soustavy parciálních diferenciálních rovnic.

Podle způsobu identifikace neznámé soustavy, jsou modely děleny na analytické a experimentální. Analytické modely jsou získány na základě analytických metod modelování, které pracují s bilancemi hmoty či energie. Experimentální modely jsou naopak získávány pomocí experimentálních metod identifikace, spočívajících v měření dat na skutečných soustavách.

Podle závislosti parametrů dynamických modelů na čase, jsou rozlišovány modely stacionární a nestacionární. Stacionární modely jsou časově invariantní, čili hodnoty parametrů modelů se nemění v závislosti na čase a zůstávají konstantní. Pokud je soustava popsána diferenciálními rovnicemi s konstantním koeficienty, pak se jedná o soustavu stacionární. Naopak v případě nestacionárních modelů dochází ke změně jejich parametrů v závislosti na čase (Vrožina, 2012).

V oblasti automatizace se jeví jako nejvhodnější použít matematický model soustavy. Jedná se o zobrazení důležitých vlastností reálné soustavy matematickým popisem. Ten může být získán pomocí identifikace. Do jaké míry platí skutečnost, že získaný matematický model odpovídá dané soustavě, je ověřováno simulacemi. A tudíž vzniká požadavek, že matematický model musí umožňovat analýzu svých statických i dynamických vlastností. Identifikace soustavy se dělí na dvě části. Na analytickou část, tj. matematicko-fyzikální analýzu a empirickou část, čili experimentální identifikaci (Balátě, 2004).

1.3.2 Matematicko-fyzikální modelování

Matematicko-fyzikální analýza je založena na matematickém popisu elementárních jevů, ke kterým v soustavě dochází a rovněž na konstrukčních a provozních údajích o soustavě. Během analýzy vzniká soustava algebraických a diferenciálních rovnic prvního řádu vycházející z bilance hmoty, energie nebo z rovnic kontinuity. V těchto rovnicích jsou obsaženy nejenom veličiny vstupní, výstupní a stavové, ale i další nadbytečné vnitřní veličiny soustavy, které lze zanedbat. V případě, že je matematicko-fyzikální popis úplný, lze jednotlivé rovnice tohoto popisu přepsat do stavových rovnic. Tehdy mají složky stavového vektoru konkrétní a jednoduchý fyzikální význam (Strejc, 1978).

Matematicko-fyzikální modelování udává matematický model zkoumané soustavy vyjadřující její vnitřní popis. Jedná se o tzv. bílou skříňku. Takto získaný model může být velice přesný, ale může být zbytečně moc složitý. Jeho odvození je v porovnání s experimentální identifikací náročnější, avšak takovýto model má mnohem širší pole platnosti. Mimoto výhodou je i souvislost mezi parametry modelu soustavy a technologickými parametry vlastní soustavy (Vrožina, 2012).

1.3.3 Experimentální identifikace

Určení modelu proběhne na základě znalosti experimentálně získaných dat. Je zde předpoklad, že na zvoleném časovém úseku lze měřit hodnoty vstupu i výstupu soustavy. Tento způsob je znám jako metoda černé skříňky. Následně získaný model by mohl být klasifikován jako vnější popis chování zkoumané soustavy. Výsledný model má většinou jednoduchý tvar, ale je použitelný pouze v omezeném rozsahu. Pouze pro konkrétní soustavu a okolí, které na soustavu působí. Nalezení parametrů modelu je jednoduché, avšak ty nemají konkrétní fyzikální význam (Balátě, 2004).

Metody experimentální identifikace lze rozdělit na metody deterministické a stochastické. Co se týče deterministických metod, zde se počítá se znalostí počátečního stavu. Dále pak vstupním signálem soustavy je jeden ze základních signálů, jakým je např. jednotkový skok, nebo sinusový průběh. Naopak u stochastických metod, počáteční stav soustavy je libovolný a není předem znám. Na vstup soustavy pak může být přiveden jakýkoliv vstupní signál, ke kterému se však přidává i parazitní šum. Rovněž platí, že statistické vlastnosti takového šumu nemusí být vždy předem známé. Obecným rozdílem mezi metodami experimentální analýzy je druh zvoleného modelu a kritéria jakosti odhadu modelu. Dále se však liší podle toho, zda jsou určeny pro lineární či nelineární soustavy, jestli je přítomen šum či nikoliv a zda ho lze nebo nelze měřit. Dalším faktorem je znalost či neznalost struktury a řádu modelu soustavy.

Ať už byl zvolen jakýkoliv z uvedených přístupů, jak získat model soustavy, je odvození tohoto modelu nezbytné pro zajištění správné regulace soustavy, které model přísluší. Díky tomu je pak možné odvodit konkrétní estimátor i stavový regulátor.

1.4 ESTIMACE STAVU

Jelikož stavový regulátor potřebuje ke své činnosti znalost vektoru stavových proměnných dané soustavy, z nichž mohou být měřitelné pouze některé anebo žádná, je tedy nezbytné nedostupné stavové proměnné estimovat. Zařízení, nebo program sloužící k tomuto účelu se nazývá estimátor stavu. Pokud estimátor stavu odhaduje veškeré stavové proměnné bez ohledu na to, zdali jsou některé z nich měřitelné, jedná se o estimátor úplného řádu. Použití tohoto estimátoru nemusí být v mnoha případech nezbytné. Pokud jsou některé stavové proměnné měřitelné a naměřené hodnoty lze považovat za správné, dojde k estimaci pouze zbývajících neměřitelných stavových proměnných. Estimátor, který odhaduje méně než f stavových proměnných, kde f udává rozměr stavového vektoru je nazýván jako estimátor redukovaného řádu. Jestliže je použit nejnížší možný řád tohoto estimátoru, pak se tedy jedná o tzv. estimátor minimálního řádu. Pro účely této práce byl použit estimátor úplného řádu a v následujícím pododdíle byl uveden jeho podrobný popis (Ogata, 2010).

1.4.1 Diskrétní estimátor úplného řádu

Tato práce se zabývá řízením reálného procesu, který je zastoupen diskrétní soustavou 3. řádu. Právě stavy této soustavy je třeba odhadovat. Byly tedy odvozeny výpočetní vztahy pro diskrétní estimátor 3. řádu. Základem jsou rovnice diskrétního stavového popisu soustavy:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{M} \cdot \mathbf{x}(k) + \mathbf{N} \cdot \mathbf{u}(k), \\ \mathbf{y}(k) &= \mathbf{C} \cdot \mathbf{x}(k) + \mathbf{D} \cdot \mathbf{u}(k). \end{aligned} \tag{1.71}$$

kde \mathbf{M} – matice dynamiky,

\mathbf{N} – vektor vstupu,

\mathbf{C} – matice stavového vektoru,

\mathbf{D} – koeficient vstupu,

$\mathbf{x}(k)$ – hodnota stavové veličiny v aktuálním kroku,

$\mathbf{u}(k)$ – aktuální hodnota akčního zásahu,

$\mathbf{y}(k)$ – hodnota na výstupu soustavy v aktuálním kroku.

Poté co byl zvolen vztah pro odhad stavu v následujícím kroku:

$$\hat{\mathbf{x}}(k+1) = \mathbf{M} \cdot \hat{\mathbf{x}}(k) + \mathbf{N} \cdot \mathbf{u}(k) + \mathbf{K}_e \cdot (\mathbf{y}(k) - \mathbf{C} \cdot \hat{\mathbf{x}}(k)), \tag{1.72}$$

bylo možné napsat

$$\mathbf{x}(k+1) = \mathbf{M} \cdot \mathbf{x}(k) + \mathbf{N} \cdot \mathbf{u}(k) - \mathbf{K}_e \cdot \mathbf{C} \cdot \mathbf{x}(k) + \mathbf{K}_e \cdot \mathbf{y}(k) \quad (1.73)$$

a následně byla tato rovnice upravena na

$$\mathbf{x}(k+1) = (\mathbf{M} - \mathbf{K}_e \cdot \mathbf{C}) \cdot \mathbf{x}(k) + \mathbf{N} \cdot \mathbf{u}(k) + \mathbf{K}_e \cdot \mathbf{C} \cdot \mathbf{x}(k). \quad (1.74)$$

Pokud se vypočítá rozdíl rovnic (1.72) a (1.74) vznikne

$$\hat{\mathbf{x}}(k+1) - \mathbf{x}(k+1) = [(\mathbf{M} - \mathbf{K}_e \cdot \mathbf{C}) \cdot \hat{\mathbf{x}}(k) + \mathbf{N} \cdot \mathbf{u}(k) + \mathbf{K}_e \cdot \mathbf{y}(k)] - [(\mathbf{M} - \mathbf{K}_e \cdot \mathbf{C}) \cdot \mathbf{x}(k) + \mathbf{N} \cdot \mathbf{u}(k) + \mathbf{K}_e \cdot \mathbf{C} \cdot \mathbf{x}(k)], \quad (1.75)$$

a aplikací dalších úprav:

$$\begin{aligned} \hat{\mathbf{x}}(k+1) - \mathbf{x}(k+1) &= (\mathbf{M} - \mathbf{K}_e \mathbf{C}) \cdot (\hat{\mathbf{x}}(k) - \mathbf{x}(k)) + \mathbf{K}_e \mathbf{C} \cdot \mathbf{x}(k) - \mathbf{K}_e \mathbf{C} \cdot \mathbf{x}(k) = \\ &= (\mathbf{M} - \mathbf{K}_e \mathbf{C}) \cdot (\hat{\mathbf{x}}(k) - \mathbf{x}(k)) + \mathbf{K}_e \mathbf{C} \cdot \mathbf{x}(k) - \mathbf{K}_e \mathbf{C} \cdot \mathbf{x}(k). \end{aligned} \quad (1.76)$$

Finální úpravou je získána rovnice chyby estimátoru

$$\hat{\mathbf{x}}(k+1) - \mathbf{x}(k+1) = (\mathbf{M} - \mathbf{K}_e \cdot \mathbf{C}) \cdot (\hat{\mathbf{x}}(k) - \mathbf{x}(k)). \quad (1.77)$$

kde $\hat{\mathbf{x}}(k+1)$ – odhad stavu v následujícím kroku,

$\mathbf{x}(k+1)$ – hodnota stavu v následujícím kroku,

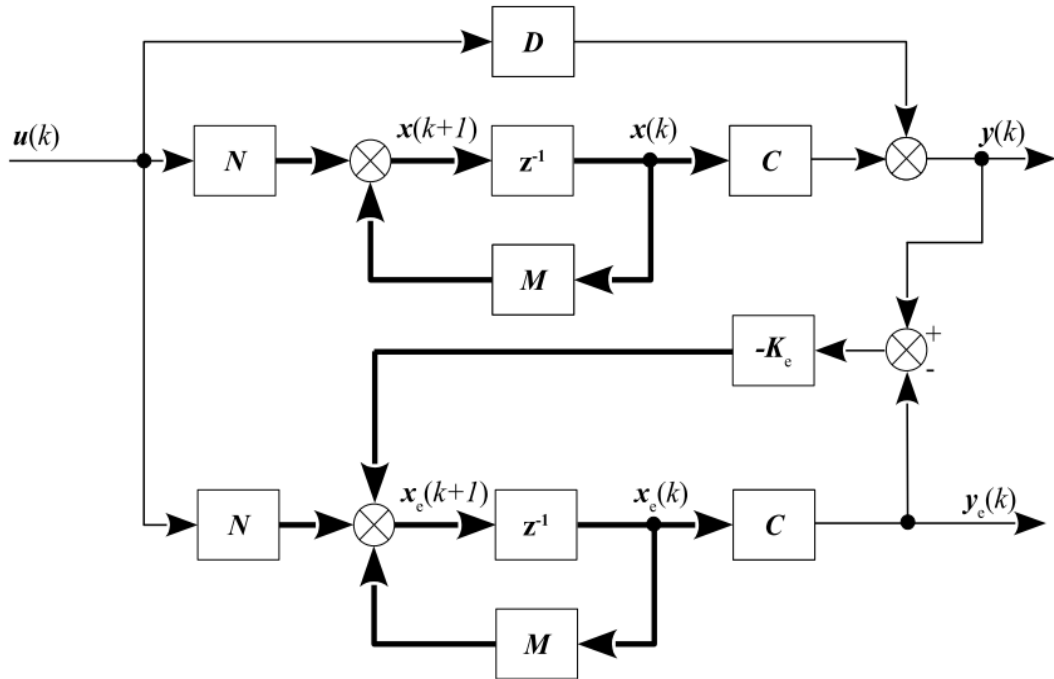
\mathbf{M} – matice dynamiky,

\mathbf{K}_e – matice estimátoru,

\mathbf{C} – matice výstupu.

Dále bude nutné vhodně zvolit matici \mathbf{K}_e , potřebnou k řešení rovnice (1.77). Volba musí být provedena tak, aby výsledná matice $(\mathbf{M} - \mathbf{K}_e \cdot \mathbf{C})$, jejíž determinant je ekvivalentem jmenovatele matice přechodu (1.31), byla stabilní a měla vhodná vlastní čísla. Je přihlédnuto i ke skutečnosti, že časová konstanta výsledného estimátoru, by měla být výrazně nižší než je časová konstanta řízeného procesu. Vhodná volba matice estimátoru \mathbf{K}_e zajišťuje minimalizaci rozdílu hodnoty odhadovaného a skutečného výstupu soustavy.

Díky znalosti konkrétní matice K_e je nyní možné vypočítat odhad stavů diskrétní soustavy v následujícím kroku dle rovnice (1.72) a sestavit tak diskrétní estimátor úplného řádu dle obr. 1.12 (Strejc, 1978).



Obr. 1.12 – Blokové schéma diskrétního estimátoru úplného řádu

1.5 LQ REGULÁTOR

Výpočet LQ regulátoru vychází z požadavku minimalizace obecného kvadratického kritéria

$$J = \frac{1}{2} \mathbf{x}_N^T \cdot \mathbf{P}_N \cdot \mathbf{x}_N + \frac{1}{2} \cdot \sum_{k=0}^{N-1} \begin{bmatrix} \mathbf{x}_k^T & \mathbf{u}_k^T \end{bmatrix} \cdot \begin{bmatrix} \mathbf{Q}_k & \mathbf{T}_k \\ \mathbf{T}_k^T & \mathbf{R}_k \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}, \quad (1.78)$$

kde \mathbf{P}_N – matice penalizace konvergence výstupní veličiny k nule,

\mathbf{Q}_k – pozitivně semidefinitní matice penalizace stavových veličin,

\mathbf{R}_k – pozitivně definitní matice penalizace akčních zásahů,

\mathbf{x}_k – vektor stavových proměnných,

\mathbf{u}_k – vektor akčních zásahů.

\mathbf{T}_k – obvykle se jedná o nulovou matici.

Rovnice kvadratického kritéria byla přepsána jako

$$J = \frac{1}{2} \mathbf{x}_N^T \cdot \mathbf{P}_N \cdot \mathbf{x}_N + \frac{1}{2} \cdot \sum_{k=0}^{N-1} \left[\mathbf{x}^T \mathbf{Q} \mathbf{x} + 2 \mathbf{x}^T \mathbf{T} \mathbf{u} + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \right], \quad (1.79)$$

jelikož je však matice \mathbf{T} nulová, výsledný tvar kritéria je

$$J = \frac{1}{2} \mathbf{x}_N^T \cdot \mathbf{P}_N \cdot \mathbf{x}_N + \frac{1}{2} \cdot \sum_{k=0}^{N-1} \left[\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \right]. \quad (1.80)$$

První člen na pravé straně rovnice představuje poslední element funkcionálu, protože v okamžiku $k = N$ se v kritériu uplatňují pouze kvadráty stavových veličin. Druhý člen rovnice popisuje ztrátovou funkci na intervalu od nuly do $N - 1$, která hodnotí kvadráty jak stavových tak i akčních veličin. Jelikož může být toto kritérium uplatněno na konečném i nekonečném intervalu, liší se tak od klasických metod syntézy podle kvadratických kritérií, kde je vyžadován nekonečný interval.

Matice \mathbf{P} a \mathbf{Q} jsou voleny symetrické a pozitivně semidefinitní. Matice \mathbf{R} je pak pozitivně definitní. Pro usnadnění výpočtů byly matice uvažovány jako časově invariantní. Nejrozumnější volbou je zapsání matic \mathbf{Q} a \mathbf{R} jako diagonální a nezáporné. Matice \mathbf{Q} je tedy maticí diagonální s kladnými prvky a její rozměry odpovídají počtu stavů regulované soustavy. Pokud matice \mathbf{Q} není pozitivně semidefinitní, pak se mohou v uzavřené smyčce vyskytnout nestabilní módy, které ale nemusí výslednou funkci regulátoru nijak ovlivnit. Pozitivní definitnost matice \mathbf{R} je vyžadována vždy. Pokud není pozitivní definitnost zachována, není zabráněno vzniku bezmezně velkých akčních zásahů a nelze vypočítat inverzi \mathbf{R}^{-1} . Konkrétní volba složek váhových matic \mathbf{Q} a \mathbf{R} , by se dala považovat za kompromis mezi výkonem regulace a malými akčními zásahy. Čím menší je hodnota matice \mathbf{Q} , tím vyšší je rychlost ustálení regulované veličiny. Malé hodnoty matice \mathbf{R} odpovídají povolení k aplikaci velkých akčních zásahů. Současné zvyšování či snižování hodnot obou matic pozbývá významu a je tedy třeba jednu matici určit a měnit hodnoty matice druhé.

Zapsáno matematicky pro matici \mathbf{Q}

$$\sqrt{Q_i} = \frac{1}{|y_i^{\max}|}, \quad i = 1, 2, \dots, e, \quad (1.81)$$

kde Q_i – i -tá diagonální složka matice \mathbf{Q} ,

$|y_i^{\max}|$ – maximální povolená hodnota odchylky i -té složky výstupu soustavy.

I pro matici \mathbf{R}

$$\sqrt{R_i} = \frac{1}{|u_i^{\max}|}, \quad i = 1, 2, \dots, d, \quad (1.82)$$

kde R_i – složka matice \mathbf{R} , ležící na i -tém řádku i sloupci,

$|u_i^{\max}|$ – maximální povolená hodnota akčního zásahu i -té složky vstupu soustavy.

Další použitelný přístup volby hodnot složek matice \mathbf{Q} je založen na korekci vah jednotlivých složek stavového vektoru. Pokud veškeré složky stavového vektoru mají stejnou váhu, může být správnou volbou matice \mathbf{Q} jedna nebo více složek následně upřednostněny, aby se v kritériu více projevíly. V případě, že amplitudy jednotlivých složek stavového vektoru jsou vzájemně rozdílné, může vhodná volba složek matice \mathbf{Q} zajistit jejich rovnost vzhledem ke kritériu, pokud je tak požadováno.

Díky vyřešení minimalizace kritéria (1.78) je dána rovnice akčního zásahu regulátoru (1.86), kde \mathbf{S}_k je řešením Riccatiho rovnice (1.83). Existují dva možné přístupy výpočtu matice \mathbf{S}_k . První z nich uvažuje konečný interval řízení. V tomto případě se složky matice \mathbf{S}_k mění v čase a tudíž i akční zásah regulátoru je v čase proměnný. Rovnice je ošetřena koncovou podmínkou ve tvaru $\mathbf{S}_{k+1} = \mathbf{P}_N$.

$$\begin{aligned} \mathbf{S}_k = & \mathbf{Q}_k + \mathbf{M}^T \cdot \mathbf{S}_{k+1} \cdot \mathbf{M} - \\ & \left(\mathbf{T}_k + \mathbf{M}^T \cdot \mathbf{S}_{k+1} \cdot \mathbf{N} \right) \cdot \left(\mathbf{R}_k + \mathbf{N}^T \cdot \mathbf{S}_{k+1} \cdot \mathbf{N} \right)^{-1} \cdot \left(\mathbf{T}_k^T + \mathbf{N}^T \cdot \mathbf{S}_{k+1} \cdot \mathbf{M} \right) \end{aligned} \quad (1.83)$$

kde \mathbf{M} – matice soustavy,

\mathbf{N} – matice řízení,

\mathbf{S}_{k+1} – hodnota matice v předchozím diskrétním okamžiku,

\mathbf{Q}_k – pozitivně semidefinitní matice penalizace stavových veličin,

\mathbf{R}_k – pozitivně definitní matice penalizace akčních zásahů.

Poté co byla získána výsledná hodnota \mathbf{S}_{k+1} , je možné vypočítat matici stavového regulátoru

$$\mathbf{K}_R = - \left(\mathbf{R}_k + \mathbf{N}^T \cdot \mathbf{S}_{k+1} \cdot \mathbf{N} \right)^{-1} \cdot \left(\mathbf{T}_k^T + \mathbf{N}^T \cdot \mathbf{S}_{k+1} \cdot \mathbf{M} \right), \quad (1.84)$$

a následný akční zásah takového regulátoru, by se dal popsat rovnicí

$$\mathbf{u}_k = \mathbf{K}_R \cdot \mathbf{x}_k. \quad (1.85)$$

Po dosazení \mathbf{K}_R z rovnice (1.84) vzniká rovnice pro regulační zákon

$$\mathbf{u}_k = -\left(\mathbf{R}_k + \mathbf{N}^T \cdot \mathbf{S}_{k+1} \cdot \mathbf{N}\right)^{-1} \cdot \left(\mathbf{T}^T + \mathbf{N}^T \cdot \mathbf{S}_{k+1} \cdot \mathbf{M}\right) \cdot \mathbf{x}_k, \quad (1.86)$$

Druhým přístupem je iterační výpočet Riccatiho rovnice, kde je uvažován nekonečný interval řízení. Maticový iterační výpočet Riccatiho rovnice vychází z předpokladu, že $N \rightarrow \infty$ a také, že se hodnota stavů v nekonečnu blíží k nule. A proto je hodnota $\mathbf{S}(N)$ zvolena jako rovna nule. V prvním kroku je proměnná $\mathbf{S}(N)$ dosazena do rovnice (1.87) za $\mathbf{S}(k+1)$. V každém dalším kroku je však již za $\mathbf{S}(k+1)$ dosazená hodnota $\mathbf{S}(k)$, která byla vypočtená v předchozím kroku. Při provádění kalkulaci se tedy postupuje zpětně. K upřesňování výpočtu dochází do doby, dokud nedojde k ustálení výsledku. A sice že $\|\mathbf{S}(k) - \mathbf{S}(k+1)\| \leq \varepsilon$. Jako hodnota ε je zpravidla zvoleno číslo v řádech tisícín. Zápis maticového iteračního výpočtu Riccatiho rovnice je

$$\mathbf{S}(k) = \mathbf{Q} + \mathbf{M}^T \cdot \mathbf{S}(k+1) \cdot \mathbf{M} - \left(\mathbf{T} + \mathbf{M}^T \cdot \mathbf{S}(k+1) \cdot \mathbf{N}\right) \cdot \left(\mathbf{R} + \mathbf{N}^T \cdot \mathbf{S}(k+1) \cdot \mathbf{N}\right)^{-1} \cdot \left(\mathbf{T}^T + \mathbf{N}^T \cdot \mathbf{S}(k+1) \cdot \mathbf{M}\right) \quad (1.87)$$

kde \mathbf{M} – matice soustavy,

\mathbf{N} – matice řízení,

$\mathbf{S}(k+1)$ – matice vypočtená v následujícím kroku,

\mathbf{Q} – konstantní pozitivně semidefinitní matice penalizace stavových veličin,

\mathbf{R} – konstantní pozitivně definitní matice penalizace akčních zásahů.

Jakmile byla získána konečná a konstantní matice \mathbf{S} , je možné vypočítat matici stavového regulátoru \mathbf{K}_R , která je rovněž nazývána jako matice Kalmanova zesílení

$$\mathbf{K}_R = -\left(\mathbf{R} + \mathbf{N}^T \cdot \mathbf{S} \cdot \mathbf{N}\right)^{-1} \cdot \left(\mathbf{T}^T + \mathbf{N}^T \cdot \mathbf{S} \cdot \mathbf{M}\right), \quad (1.88)$$

odpovídající akční zásah regulátoru, je možné popsat rovnicí

$$\mathbf{u}(k) = -\left(\mathbf{R} + \mathbf{N}^T \cdot \mathbf{S} \cdot \mathbf{N}\right)^{-1} \cdot \left(\mathbf{T}^T + \mathbf{N}^T \cdot \mathbf{S} \cdot \mathbf{M}\right) \cdot \mathbf{x}(k), \quad (1.89)$$

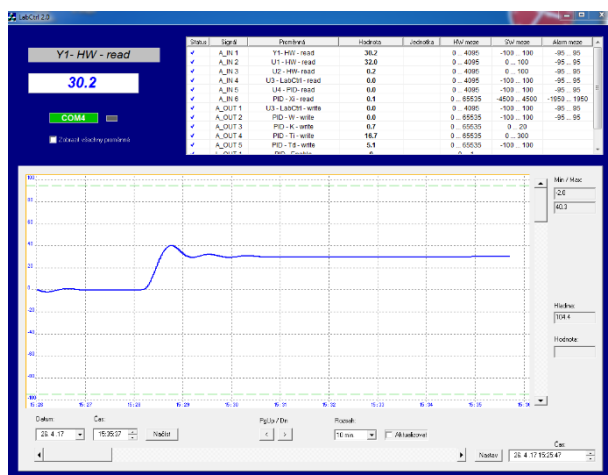
(Stengel, 1986, Strejc, 1978, Bosgra, 2007).

2 PRAKTICKÉ ŘEŠENÍ

Tato kapitola pojednává o plnění cíle diplomové práce, kterým bylo realizovat stavový LQ regulátor v jednoduchém mikropočítači. Regulátor tak jako k němu příslušný estimátor byl navržen pro konkrétní reálný proces, který reagoval měnicími se hodnotami výstupní veličiny v závislosti na akční veličině.

Dynamika soustavy však byla z počátku neznámá, tudíž její identifikace byla prvním krokem k odvození diskrétního estimátoru a LQ regulátoru. Možnost simulovat chování konkrétního nastavení estimátoru či LQ regulátoru poskytoval program Matlab-Simulink, zatímco algoritmy pro mikropočítač Arduino byly tvořeny v prostředí IDE. Způsoby využití Arduino byly popsány v následujících oddílech.

Náplní této diplomové práce bylo odvození vhodného estimátoru a stavového regulátoru pro identifikovanou soustavu. Následovalo jejich otestování v prostředí Simulink. Po úspěšném



Obr. 2.1 – Prostředí Labctrl

prozkoušení daného zapojení, byl naprogramován jednoduchý mikropočítač Arduino USB v2.2 takovým způsobem, aby zastával funkci diskrétního LQ regulátoru s příslušným estimátorem. Jak z prostředí Simulink, tak i z prostředí Labctrl při řízení reálného procesu byly zaznamenány grafy znázorňující chování regulačního obvodu i některých jeho samostatných součástí. Ukázka grafického uživatelského rozhraní Labctrl je vyobrazena na obr. 2.1.

2.1 VYUŽITÉ HW A SW PROSTŘEDKY



Obr. 2.2 – Arduino logo

Arduino původně vzniklo v italském institutu IIDI jako pomůcka pro studenty. Z počátku se tedy jednalo o velice jednoduchý mikropočítač, vyžadující minimální nároky na znalost elektroniky a programování. Postupem času, kdy se Arduino rozšířilo veřejnosti, vzniklo mnoho verzí s různými vylepšeními. Novější verze nabízejí podstatně vyšší výpočetní výkon, větší interní paměť či více vstupů a výstupů. Mezi

nejpoužívanější rozšíření stojí za zmínku Ethernet, GSM, Bluetooth, Wi-fi, rozšíření pro LCD, nebo rozšíření pro ovládání krokových motorů.

Platforma Arduino patří mezi otevřené a tudíž je možné upravovat nejen zdrojové kódy, které mohou být volně staženy, pozměňovány a následně poskytovány dalším uživatelům, ale můžou být modifikovány i vlastní obvody a zapojení mikropočítače podle potřeby každého uživatele. LQ regulátor bude realizován na desce Arduino USB v2.2, která byla podrobněji popsána dále.

2.1.1 Arduino USB v2.2



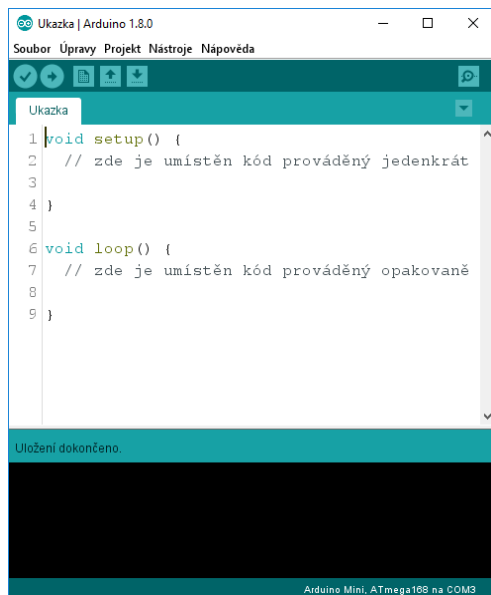
Obr. 2.3 – Arduino USB v2.2

Již výše uvedená deska Arduino USB v2.2 je vybavena mikroprocesorem Atmel ATMEGA168-20P a je vyobrazena na obr. 2.3. Mikroprocesor je taktován krystalem o frekvenci 16 MHz. K dispozici je 14 digitálních vstupů/ výstupů a 6 analogových vstupů.

Z toho digitální svorky číslo nula a jedna můžou sloužit pro sériovou komunikaci, kde svorka číslo nula je přijímací (RXD) a svorka číslo jedna je vysílací (TXD). Jak RXD, tak i TXD má svou indikační červenou LED. Dále zelená LED signalizuje připojené napájení. Přítomna je i jedna pracovní žlutá LED, která je permanentně připojena ke vstupně-výstupní svorce číslo 13. Na desce je umístěno reset tlačítko, které se použije v případě potřeby spuštění programu znovu od začátku. Deska může být napájena z externího zdroje, anebo z PC, a to propojením pomocí kabelu USB typu B. Skrze tento kabel dochází zejména k vlastnímu programování desky. Korektní spojení PC a desky je zajištěno díky USB - USART převodníku, který je připájený na desce v podobě integrovaného obvodu FT232BL.

Pracovní napětí desky je 5 V, přičemž napájecí napětí by se mělo pohybovat v rozsahu od 7 V do 12 V. Potom proud na každé jednotlivé vstupně-výstupní svorce je roven 20 mA. A zároveň hodnota proudu na vstupu nesmí překročit 40 mA, aby nedošlo k poškození mikroprocesoru. Na desce je možné najít svorku, ze které lze odebírat napětí o velikosti 3,3 V. Hodnota proudu na této svorce je pak 50 mA. Deska Arduino USB v2.2 je osazena několika typy pamětí. Velikosti jednotlivých pamětí jsou: flash paměť 16 kB, kde 0,5 kB je využito pro zavaděč, SRAM 1 kB a EEPROM 0,512 kB.

2.1.2 Programování Arduino



Obr. 2.4 – Vývojové prostředí

Jakoukoliv desku Arduino lze programovat díky vývojovému prostředí IDE, které je vyobrazeno na obr. 2.4. Po spuštění vývojového prostředí je třeba vybrat příslušnou verzi Arduino desky v nabídce nástroje. V případě, že by tak nebylo učiněno, napsaný program by nebyl do zařízení nahrán správně anebo vůbec. Nejvíce používané jsou pak ikony umístěné ve druhém řádku menu. V pořadí zleva doprava to jsou: kontrola programu na chyby v syntaxi, kontrola programu na chyby v syntaxi a při jejich absenci nahrání programu do zařízení. Dále založení nového projektu, otevření již existujícího programu a uložení stávajícího programu. Úplně vpravo je pak umístěna

ikona pro spuštění sériového monitoru. Ten lze spouštět i v nabídce nástroje, stejně jako sériový plotter. Zatímco sériový monitor bude využit pro výpis dat, sériový plotter slouží k jejich přímému vykreslení v grafu. Pokud dojde k nalezení jakékoli chyby, je o tom zobrazeno hlášení v dolní části okna.

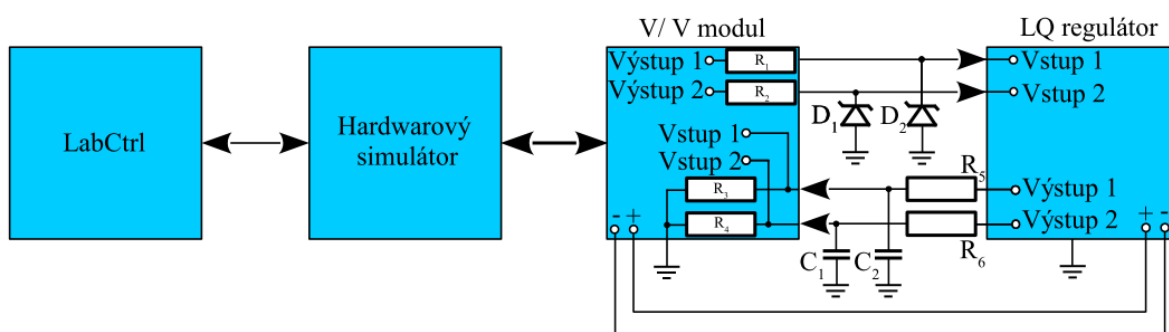
Programování desky Arduino je založeno na jazyce C++, přesněji řečeno na knihovně Wiring, algoritmy však lze programovat i standardním jazykem C++. Programování v knihovně Wiring patří mezi nejrozšířenější a rovněž takto vytvořené programy fungují ve všech podporovaných mikropočítačích Arduino. Programovací prostředí Arduino IDE může být rozšířeno o řadu knihoven, které výrazně uživateli usnadňují programování. Popřípadě zkušenější uživatel si může vytvořit takovou knihovnu, která přesně vyhoví jeho potřebám.

Program může být rozdělen do několika částí. První částí je oblast `setup()`, která slouží k navázání sériové komunikace s PC, k zavedení proměnných, určení vstupů a výstupů, nebo k definování následně používaných knihoven. Tato oblast je spuštěna vždy pouze jedenkrát a to v případě prvního spuštění programu, nebo po stisknutí reset tlačítka. Po této části následuje sekce `loop()`. Ta po provedení posledního příkazu v ní umístěného, začne automaticky znovu od svého začátku. Dále jsou ještě rozlišovány rozhodovací cykly a to buď s podmínkou na začátku, nebo na konci cyklu. Tyto cykly jsou umístěny právě do sekce `loop()`. Uživatel má během programování k dispozici všechny potřebné aritmetické a logické operace a také širokou škálu datových typů proměnných (Arduino, 2017).

2.2 PROPOJENÍ REÁLNÉHO PROCESU S REGULÁTOREM

Vedoucím diplomové práce bylo pro testování regulátoru poskytnuto technické vybavení. Jak znázorňuje obr. 2.5, základ byl tvořen mikropočítačem Arduino Due. Ten plnil funkci hardwarového simulátoru soustavy v reálném čase. Na počítači, ke kterému byl simulátor připojen pomocí USB kabelu, bylo spuštěno grafické uživatelské rozhraní Labctrl. Toto GUI sloužilo k vykreslování průběhů vstupů a výstupů soustavy, simulaci poruch a sběr měřených dat. Vzorkovací perioda simulace dosahovala 30 ms.

Dynamika soustavy nebyla předem známa, ale bylo možné naměřit reakci soustavy na skokovou změnu na vstupu z ustáleného stavu a tudíž provést identifikaci soustavy. Hardwarový simulátor byl vybaven dvěma analogovými výstupy, v rozsahu 0,55 V až 2,75 V a dvanácti analogovými vstupy se vstupním rozsahem 0 V až 3,3 V. K regulátoru však nebyl připojen přímo, ale přes vstupně-výstupní modul.



Obr. 2.5 – Blokové schéma zapojení regulačního obvodu

Spojení hardwarového simulátoru a vstupně-výstupního modulu zajišťovala příslušná kabeláž. Napětí analogových výstupů modulu se pohybovalo v rozpětí 0 V až 7,26 V, kdežto napětí na vstupech se mohlo pohybovat od 0 V do 10 V. Na výstupech modulu byly umístěny rezistory o hodnotě 300 Ω . Vstupy modulu byly spojeny se zemí přes rezistory o hodnotě 1 M Ω , které zaručily nulovou hodnotu napětí na vstupu v případě, že akční zásah regulátoru na daném vstupu soustavy není větší než nula. Veškeré pasivní členy vyobrazené na obr. 2.5 jsou přehledně sepsány do tab. 2.1. Vstupně-výstupní modul byl napájen z externího zdroje napětí. Jelikož bylo nutné navrhnout regulační obvod tak, aby nebylo třeba pracovat se záporným napětím, výsledná hodnota napětí je proto součtem dvou stejně orientovaných napětí. Vstup číslo jedna je považován za kladný a vstup číslo dva za záporný. Pro nastavení příslušné kladné hodnoty vstupu je na vstup číslo jedna přivedeno napětí ve formě PWM pulsů, zatímco vstup dva je nulový. V případě záporné hodnoty vstupu je tomu právě naopak. Takový způsob

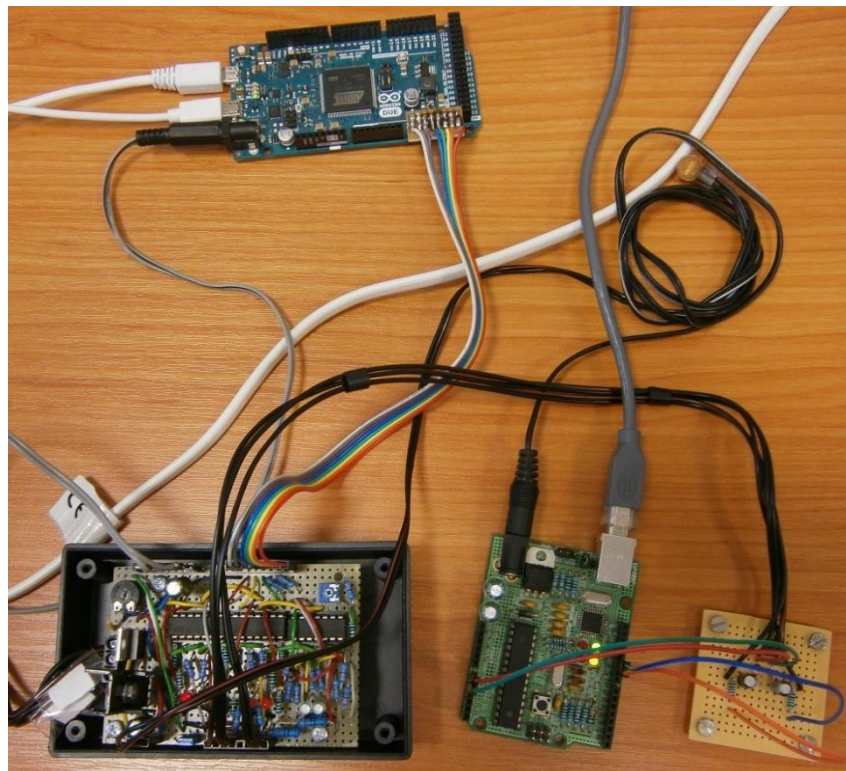
regulace soustavy vychází z principu řízení soustavy rozděleným akčním zásahem. Na vstupně-výstupním modulu byly využity dva vstupy, dva výstupy, kladný a záporný pól napájení pro Arduino USB v2.2 a společná zem. Protože vstupní rozsah mikropočítače, který fungoval jako LQ regulátor, je od 0 V do 5 V, bylo nutné jeho vstupy ošetřit.

Tab. 2.1 – Přehled součástek

-	R, k Ω	C, μ F
R1	0,3	-
R2	0,3	-
R3	1000	-
R4	1000	-
R5	10	-
R6	10	-
C1	-	10
C2	-	10

Tento problém byl vyřešen pomocí Zenerových diod zapojených v závěrném směru, přičemž průrazné napětí činilo 4,7 V. Pro správnou funkci regulátoru, tento nesoulad řešil program. Jak už bylo uvedeno, akční zásah regulátoru byl představován signálem PWM, jelikož deska Arduino USB v2.2 nebyla vybavena analogovými výstupy. Působení takových pulsů na řízený proces se z důvodu nespojitosti ukázalo jako nevhodné. Dodáním vhodných RC článků na výstup regulátoru došlo k jeho filtrování a výrazně vyšší spojitosti akčního zásahu.

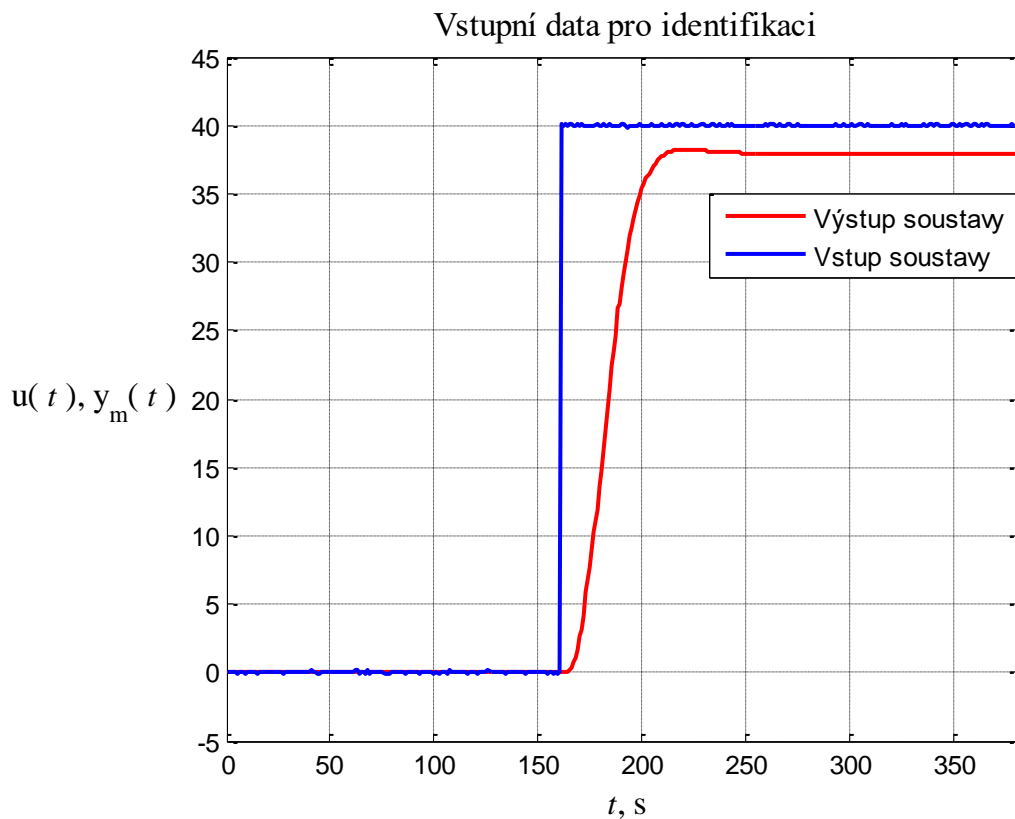
RC články se skládaly z rezistorů o hodnotě 10 k Ω a kondenzátorů o kapacitě 10 μ F, čili výsledná časová konstanta RC článku nabývala 0,1 s. Vlivem vstupního odporu vstupně-výstupního modulu vlastně vznikl odporový dělič, přičemž na výstupu tohoto děliče bylo 99 % ze vstupního napětí. Takto vzniklá nepřesnost byla ošetřena v programu regulátoru. Skutečnou podobu reálného procesu zapojeného do regulačního obvodu ukazuje obr. 2.6.



Obr. 2.6 – Skutečný regulační obvod

2.3 IDENTIFIKACE NEZNÁMÉ SOUSTAVY

Na vstup neznámé soustavy byl přiveden zvolený vstupní signál, na který soustava reagovala příslušnými změnami svého výstupu. Hodnoty vstupní i výstupní veličiny byly zaznamenány v závislosti na čase, přičemž perioda vzorkování soustavy byla jedna sekunda. Pro výpočetní algoritmus byla použita data představující skokovou změnu na vstupu soustavy i data odpovídající výstupní veličině soustavy. Data použitá pro identifikaci byla vyobrazena na obr. 2.7.



Obr. 2.7 – Data použitá pro identifikaci soustavy

Jako vhodný matematický model byla zvolena diferenční rovnice

$$y(k+3) + a_2 \cdot y(k+2) + a_1 \cdot y(k+1) + a_0 \cdot y(k) = b_0 \cdot u(k), \quad (2.1)$$

jejíž koeficienty a_2 , a_1 , a_0 a b_0 byly identifikovány off-line metodou nejmenších čtverců. Jedná se o minimalizaci celkové kvadratické chyby mezi odezvou navrženého modelu a skutečného procesu na totožný vstupní signál.

Poté, co byly koeficienty a_2, a_1, a_0 a b_0 známé, mohl být sestaven stavový popis identifikované soustavy

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{M} \cdot \mathbf{x}(k) + \mathbf{N} \cdot \mathbf{u}(k), \\ \mathbf{y}(k) &= \mathbf{C} \cdot \mathbf{x}(k) + \mathbf{D} \cdot \mathbf{u}(k). \end{aligned} \quad (2.2)$$

Sestaven byl v normální formě pozorovatelnosti

$$\begin{aligned} \mathbf{x}_1(k) &= \mathbf{y}(k), \\ \mathbf{x}_2(k) &= \mathbf{y}(k+1), \\ \mathbf{x}_3(k) &= \mathbf{y}(k+2), \end{aligned} \quad (2.3)$$

$$\begin{aligned} \mathbf{x}_1(k+1) &= \mathbf{x}_2(k), \\ \mathbf{x}_2(k+1) &= \mathbf{x}_3(k), \\ \mathbf{x}_3(k+1) &= -(a_2 \cdot \mathbf{y}(k+2) + a_1 \cdot \mathbf{y}(k+1) + a_0 \cdot \mathbf{y}(k) - b_0 \cdot \mathbf{u}(k)). \end{aligned} \quad (2.4)$$

Dosažením vztahů z (2.4) do rovnice (2.2) byly získány stavové matice

$$\mathbf{M} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix}, \mathbf{N} = \begin{bmatrix} 0 \\ 0 \\ b_0 \end{bmatrix}, \mathbf{C} = [1 \ 0 \ 0], \mathbf{D} = 0, \quad (2.5)$$

a tudíž lze rovnici (2.2) přepsat jako

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_1(k) \\ \mathbf{x}_2(k) \\ \mathbf{x}_3(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ b_0 \end{bmatrix} \cdot \mathbf{u}(k), \\ \mathbf{y}(k) &= [1 \ 0 \ 0] \cdot \begin{bmatrix} \mathbf{x}_1(k) \\ \mathbf{x}_2(k) \\ \mathbf{x}_3(k) \end{bmatrix}. \end{aligned} \quad (2.6)$$

Diferenční rovnice popisující zkoumaný proces byla identifikována ve tvaru

$$\mathbf{y}(k+3) - 1,3615 \cdot \mathbf{y}(k+2) - 0,0849 \cdot \mathbf{y}(k+1) + 0,4541 \cdot \mathbf{y}(k) = 0,0073 \cdot \mathbf{u}(k), \quad (2.7)$$

a tudíž diskrétní přenos soustavy třetího řádu odpovídal

$$F(s) = \frac{0,0073}{z^3 - 1,3615 \cdot z^2 - 0,0849 \cdot z + 0,4541}. \quad (2.8)$$

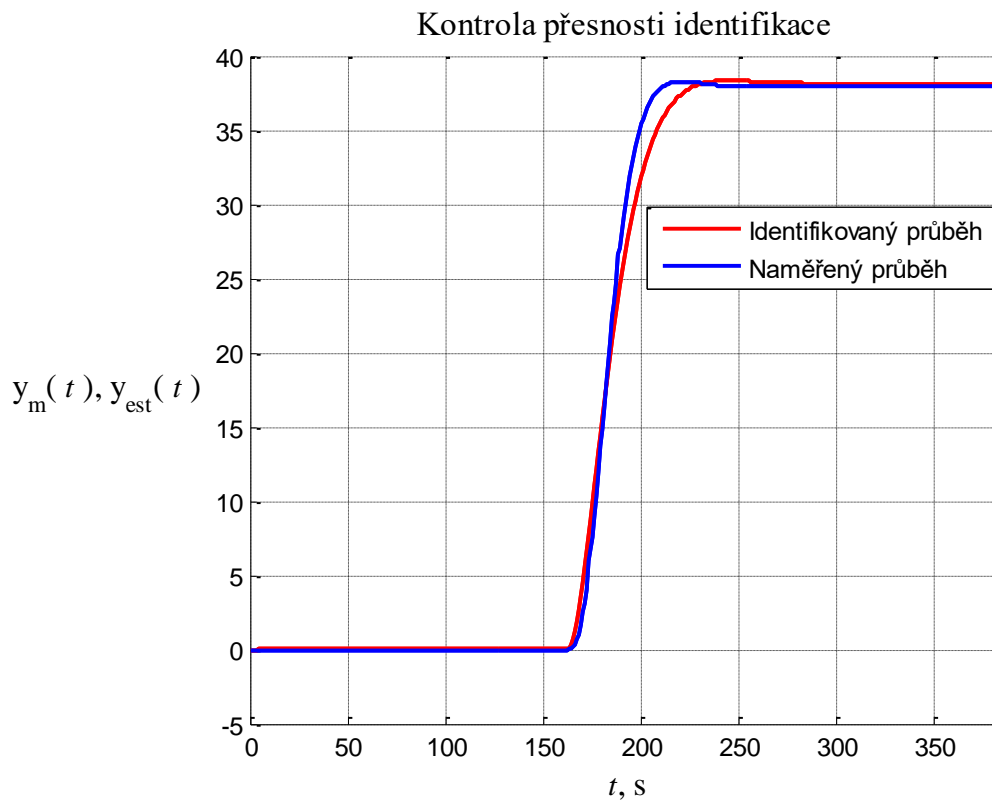
Vypočítané zesílení takto identifikované soustavy bylo ve výši 0,9528.

Poté mohl být stanoven stavový popis soustavy

$$\mathbf{x}(k+1) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0,4541 & 0,0849 & 1,3615 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_1(k) \\ \mathbf{x}_2(k) \\ \mathbf{x}_3(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0,0073 \end{bmatrix} \cdot \mathbf{u}(k),$$

$$\mathbf{y}(k) = [1 \ 0 \ 0] \cdot \begin{bmatrix} \mathbf{x}_1(k) \\ \mathbf{x}_2(k) \\ \mathbf{x}_3(k) \end{bmatrix}.$$
(2.9)

Správnost identifikovaného modelu byla ověřena v prostředí Matlab. Průběh výstupu identifikovaného modelu byl porovnán s výstupem skutečné soustavy, přičemž v obou případech byl vstup totožný. Toto porovnání vykresluje obr. 2.8.



Obr. 2.8 – Vyšetření výsledku identifikace

Jak lze na grafu výše zřetelně vidět, zesílení identifikovaného průběhu je totožné se zesílením skutečné soustavy. V tomto ohledu je model naprosto přesný. Dynamika modelu však úplně identická není. Projevilo se zavedené zjednodušení modelu, kdy byla soustava s dopravním zpožděním identifikována jako soustava bez dopravního zpoždění, pouze s navýšenou hodnotou řádu o jeden.

2.4 VÝPOČET PARAMETRŮ ESTIMÁTORU

K odhadu vektoru stavů řízené soustavy bylo zapotřebí stavového estimátoru. Výpočet odhadu stavů soustavy je obecně dán rovnicí (1.72), kde matice \mathbf{K}_e je vhodně zvolena, jak je popsáno níže. Vektor chyb odhadu byl stanoven jako rozdíl skutečného a odhadovaného stavového vektoru soustavy, tedy jako

$$\mathbf{e}(k) = \hat{\mathbf{x}}(k) - \mathbf{x}(k). \quad (2.10)$$

Ten byl následně dosazen do rovnice (1.77) a vzniklo

$$\mathbf{e}(k+1) = (\mathbf{M} - \mathbf{K}_e \cdot \mathbf{C}) \cdot \mathbf{e}(k). \quad (2.11)$$

Aplikací Z-transformace byla získáno

$$E(z) \cdot z = (\mathbf{M} - \mathbf{K}_e \cdot \mathbf{C}) \cdot E(z), \quad (2.12)$$

kde $E(z)$ – obraz vektoru chyb odhadu,

z – komplexní proměnná,

\mathbf{M} – matice dynamiky,

\mathbf{K}_e – matice estimátoru,

\mathbf{C} – výstupní matice.

Vytknutím $E(z)$ byla získána rovnice

$$(z \cdot \mathbf{I} - \mathbf{M} + \mathbf{K}_e \cdot \mathbf{C}) \cdot E(z) = 0. \quad (2.13)$$

Ze vztahu v závorce na levé straně této rovnice byl v dalším kroku spočítán determinant, který byl položen roven nule.

$$\det(z \cdot \mathbf{I} - \mathbf{M} + \mathbf{K}_e \cdot \mathbf{C}) = 0. \quad (2.14)$$

Matice \mathbf{K}_e je volena tak, aby vlastní čísla výrazu $(\mathbf{M} - \mathbf{K}_e \cdot \mathbf{C})$ byla totožná a ležela uvnitř jednotkové kružnice. Byla-li zadána obecná diskrétní soustava 3. řádu popsána stavovými maticemi

$$\mathbf{M} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix}, \mathbf{N} = \begin{bmatrix} 0 \\ 0 \\ b_0 \end{bmatrix}, \mathbf{C} = [1 \ 0 \ 0], \mathbf{D} = 0, \quad (2.15)$$

potom je úkolem nalézt matici estimátoru

$$\mathbf{K}_e = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix}, \quad (2.16)$$

která zajistí jeho správnou funkci. Pro výpočet je vhodné vypočítat součin matic \mathbf{K}_e a \mathbf{C} .

$$\mathbf{K}_e \cdot \mathbf{C} = \begin{bmatrix} k_1 & 0 & 0 \\ k_2 & 0 & 0 \\ k_3 & 0 & 0 \end{bmatrix}. \quad (2.17)$$

Cílem je zvolit diskrétní reálný trojnásobný pól v oblasti kladné poloroviny jednotkové kružnice, který zajistí dostatečně rychlou odezvu bez oscilací. Rovněž je kladen důraz na zachování stability. Z rovnice pro výpočet vlastních čísel soustavy (1.31), je získán vztah

$$\det(z \cdot \mathbf{I} - \mathbf{M} + \mathbf{K}_e \cdot \mathbf{C}) = 0. \quad (2.18)$$

Poté, co byly do této rovnice dosazeny jednotlivé matice, mohlo dojít k vyjádření charakteristického polynomu soustavy obsahující prozatím neznámou matici estimátoru \mathbf{K}_e ,

$$\begin{aligned} \det(z \cdot \mathbf{I} - \mathbf{M} + \mathbf{K}_e \cdot \mathbf{C}) &= \det \begin{bmatrix} z + k_1 & -1 & 0 \\ k_2 & z & -1 \\ a_0 + k_3 & a_1 & z + a_2 \end{bmatrix} = \\ &= (z + k_1) \cdot z \cdot (z + a_2) + a_0 + k_3 - [-a_1 \cdot (z + k_1) - k_2 \cdot (z + a_2)] = \\ &= z^3 + z^2 \cdot a_2 + z^2 \cdot k_1 + z \cdot k_1 \cdot a_2 + a_0 + k_3 + a_1 \cdot z + a_1 \cdot k_1 + k_2 \cdot z + a_2 \cdot k_2, \end{aligned} \quad (2.19)$$

a výsledný charakteristický polynom je pak ve tvaru

$$\det(z \cdot \mathbf{I} - \mathbf{M} + \mathbf{K}_e \cdot \mathbf{C}) = z^3 + (a_2 + k_1) \cdot z^2 + (a_1 + a_2 \cdot k_1 + k_2) \cdot z + a_0 + a_1 \cdot k_1 + a_2 \cdot k_2 + k_3. \quad (2.20)$$

Jak už bylo výše uvedeno, byl zvolen příslušný trojnásobný pól $-\alpha$, ten odpovídá

$$(z + \alpha)^3 = z^3 + 3z^2\alpha + 3z\alpha^2 + \alpha^3. \quad (2.21)$$

Následné porovnání koeficientů u totožných mocnin rovnic (2.20) a (2.21), umožnilo získat soustavu tří rovnic o třech neznámých, které tvoří matici estimátoru \mathbf{K}_e .

A tedy

$$\begin{aligned}(a_2 + k_1) &= 3\alpha, \\ (a_1 + a_2 \cdot k_1 + k_2) &= 3\alpha^2, \\ a_0 + a_1 \cdot k_1 + a_2 \cdot k_2 + k_3 &= \alpha^3.\end{aligned}\tag{2.22}$$

Úpravou této soustavy rovnic byly získány vztahy pro výpočet jednotlivých členů matice estimátoru \mathbf{K}_e , v závislosti na zvoleném pólu $-\alpha$.

$$\begin{aligned}k_1 &= 3\alpha - a_2, \\ k_2 &= 3\alpha^2 - a_1 - a_2 \cdot k_1, \\ k_3 &= \alpha^3 - a_0 - a_1 \cdot k_1 - a_2 \cdot k_2.\end{aligned}\tag{2.23}$$

Po dosazení konkrétních potřebných koeficientů z diferenční rovnice (2.7) a pólu estimátoru $-\alpha$, byla sestavena matice estimátoru \mathbf{K}_e . Sestavení matice probíhalo za pomoci vytvořeného skriptu v prostředí Matlab.

Tyto rovnice tvoří matici estimátoru \mathbf{K}_e . A tedy

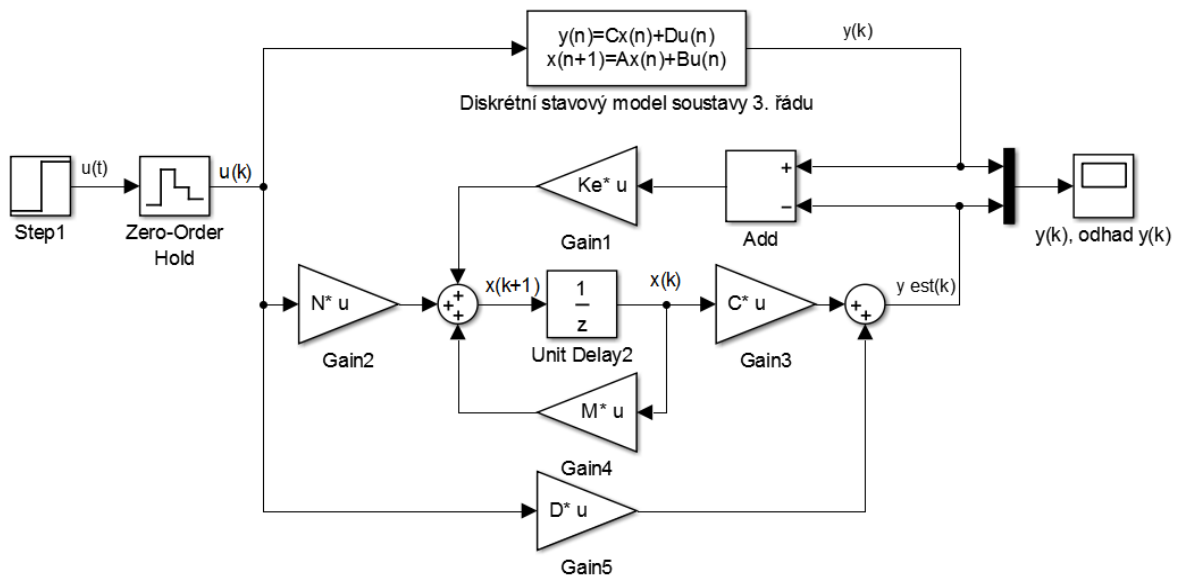
$$\begin{aligned}(-1,3615 + k_1) &= -\frac{3}{2}, \\ (-0,0849 + -1,3615 \cdot k_1 + k_2) &= \frac{3}{4}, \\ 0,4541 - 0,0849 \cdot k_1 - 1,3615 \cdot k_2 + k_3 &= -\frac{1}{8}.\end{aligned}\tag{2.24}$$

Výpočet členů matice estimátoru \mathbf{K}_e je pak možné zapsat jako

$$\begin{aligned}k_1 &= -\frac{3}{2} + 1,3615, \\ k_2 &= \frac{3}{4} + 0,0849 + 1,3615 \cdot k_1, \\ k_3 &= -\frac{1}{8} - 0,4541 + 0,0849 \cdot k_1 + 1,3615 \cdot k_2.\end{aligned}\tag{2.25}$$

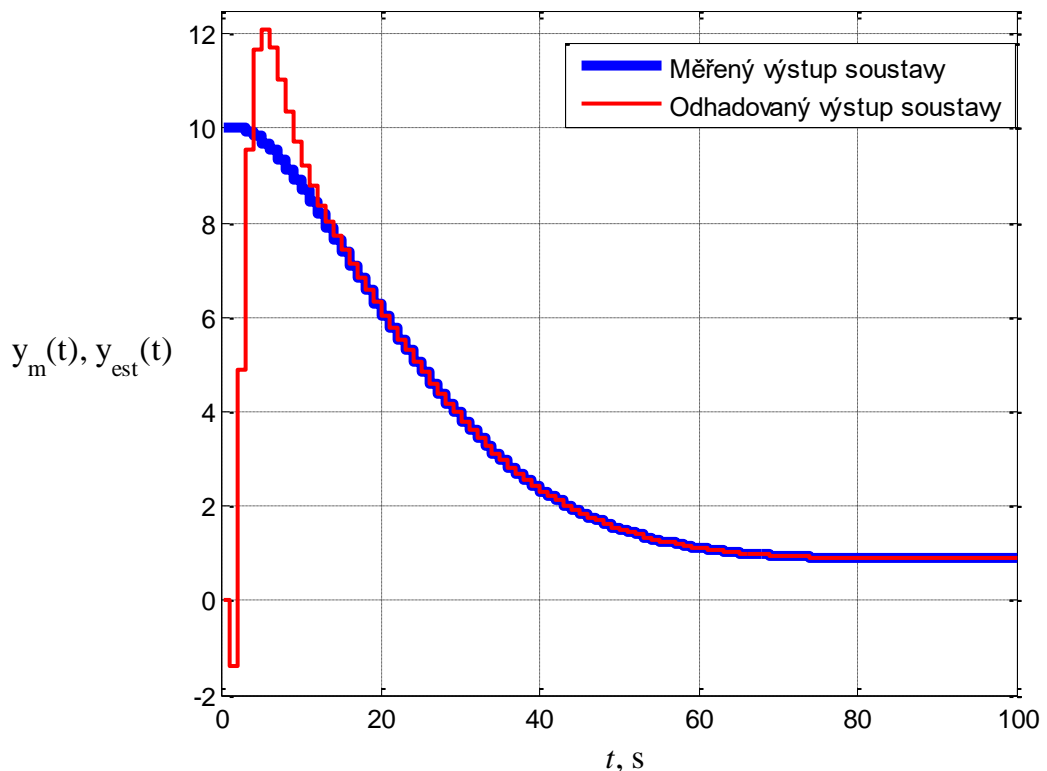
Takto vypočítaná matice estimátoru \mathbf{K}_e společně se známými maticemi stavového popisu diskretní soustavy umožnila realizaci estimátoru v prostředí Simulink, která je zachycena na obr. 2.9. Stavů diskretní soustavy, která byla řízena, jsou představovány zpožděnými výstupy soustavy. Přestože by bylo možné tento stavový vektor měřit, byl použit estimátor úplného řádu, který také plnil funkci filtru.

Jak bylo již poznamenáno výše, vlastnosti estimátoru jsou ovlivněny volbou jeho trojnásobného pólu. Obr. 2.10 vykresluje průběh odhadu stavů soustavy. Na základě simulací



Obr. 2.9 – Estimátor sestavený v prostředí Simulink

Kontrola funkce estimátoru

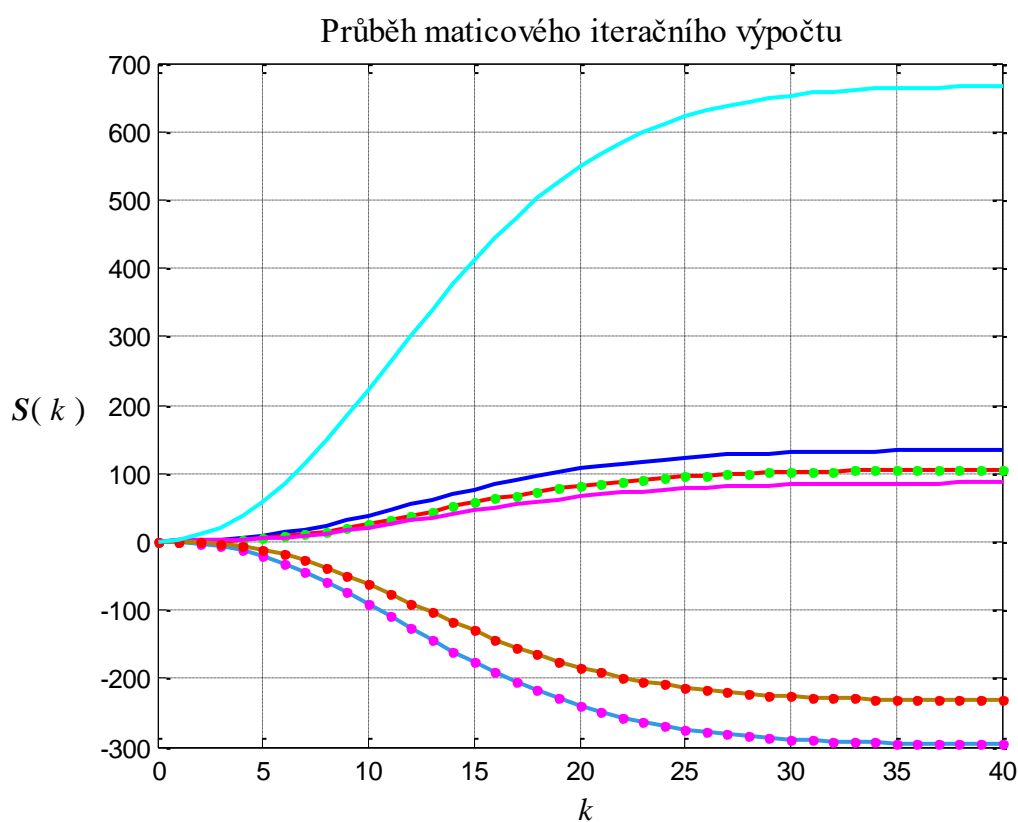


Obr. 2.10 – Průběh odhadu výstupu soustavy pro $\alpha = -0,5$

bylo zvoleno optimální nastavení $\alpha = -0,5$. Toto nastavení poskytovalo aperiodický, stabilní, nekmitavý a ne přes příliš rychlý průběh přiblížení odhadovaného stavu ke skutečnému stavu.

2.5 VÝPOČET MATICE LQ REGULÁTORU

LQ regulátor, který představuje matice \mathbf{K}_R , byl opět získán pomocí výpočetního algoritmu v prostředí Matlab. Jeho funkce byla založena na maticovém iteračním výpočtu Riccatiho rovnice (1.83). Počáteční hodnota matice $\mathbf{S}(k+1)$ byla nula. V každém dalším kroku je však tato hodnota přepsána hodnotou matice $\mathbf{S}(k)$, která byla vypočítána v aktuálním kroku. Výpočet se opakuje tak dlouho, dokud neplatí $\|\mathbf{S}(k) - \mathbf{S}(k+1)\| \leq \varepsilon$, kde bylo stanoveno $\varepsilon = 0,001$. Průběžné hodnoty $\mathbf{S}(k+1)$ zobrazuje obr. 2.11.



Obr. 2.11 – Grafické znázornění Riccatiho iteračního výpočtu

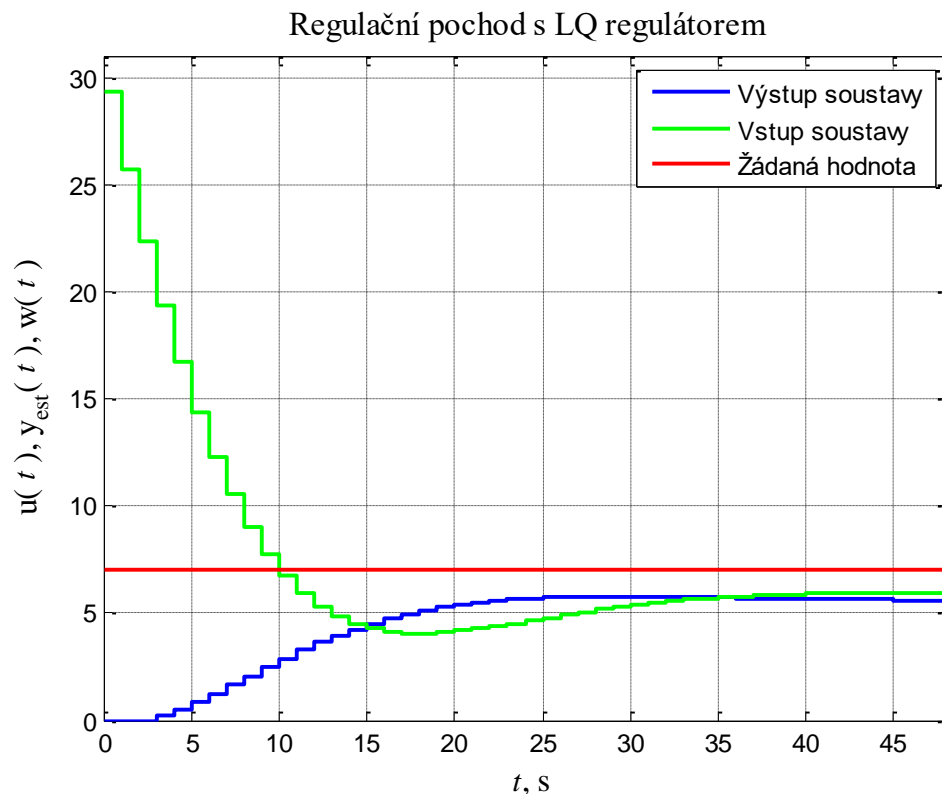
Jakmile je výsledná hodnota matice $\mathbf{S}(k+1)$ známa, je možné uskutečnit výpočet matice regulátoru \mathbf{K}_R dle rovnice (1.88). Díky dosazení matice regulátoru \mathbf{K}_R vznikl v rovnici (1.89) výpočet regulačního zákona LQ regulátoru. Ustálená hodnota matice \mathbf{S} která představuje výsledek Riccatiho iteračního výpočtu (1.83) byla zjištěna jako

$$\mathbf{S} = \begin{bmatrix} 133,7454 & 104,9918 & -296,0483 \\ 104,9918 & 86,2910 & -232,3732 \\ -296,0483 & -232,3732 & 666,5233 \end{bmatrix}. \quad (2.26)$$

Jak je zřetelně vidět, matice S je maticí symetrickou. Po dosazení její hodnoty do rovnice pro výpočet akčního zásahu LQ regulátoru, byla vykalkulována matice regulátoru K_R ve tvaru

$$K_R = [2,1409 \quad 1,6933 \quad -4,7746]. \quad (2.27)$$

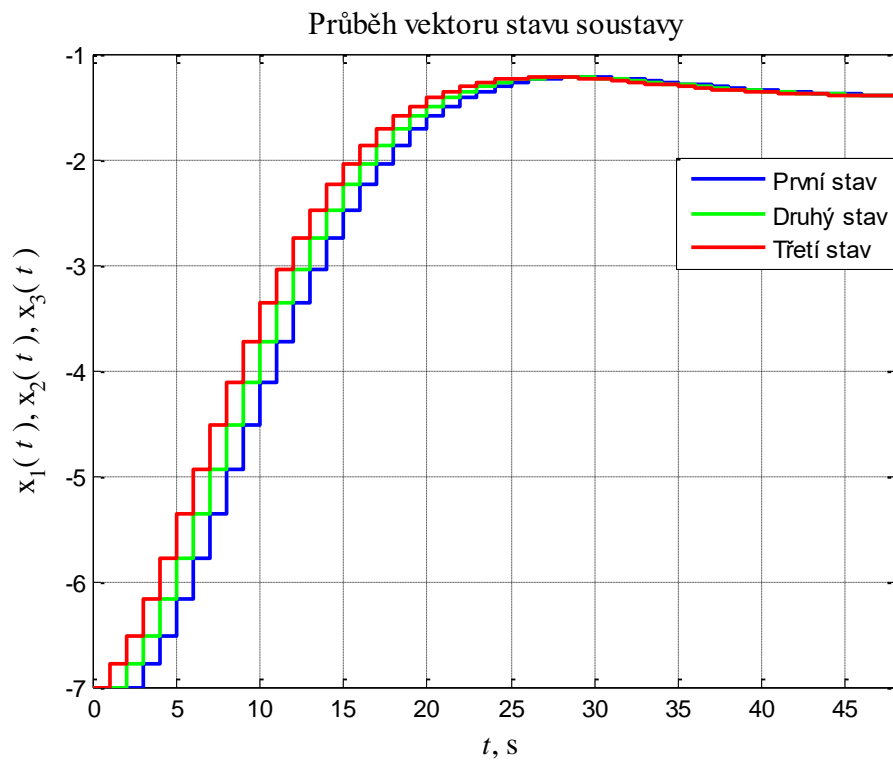
Průběhy vstupní a výstupní veličiny soustavy společně s žádanou hodnotou vykresluje obr. 2.12, zřetelný je výskyt trvalé regulační odchylky. Tento LQ regulátor by se tedy dal připodobnit k proporcionálnímu regulátoru.



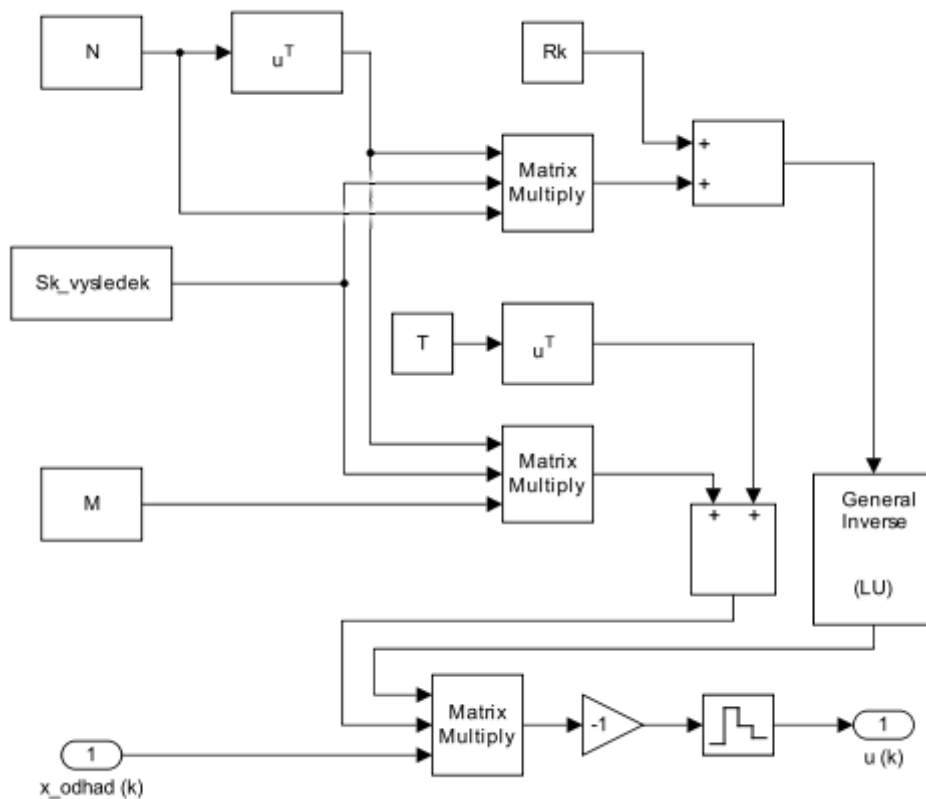
Obr. 2.12 – Regulační pochod s LQ regulátorem v prostředí Simulink

Zatímco obr. 2.12 zobrazuje základní veličiny soustavy, na obr. 2.13 lze vidět průběhy jednotlivých stavů soustavy. Na obr. 2.14 je možné spatřit blokové schéma zapojení LQ regulátoru v prostředí Simulink. Jakmile byl příslušným algoritmem získán výsledek Riccatiho maticové iterační rovnice i matice stavového popisu soustavy, mohlo dojít k výpočtu matice

regulátoru. Jak lze vidět na obr. 2.14, aktuální akční zásah LQ regulátoru odpovídá maticovému násobení matice regulátoru a aktuálního odhadu stavu soustavy.



Obr. 2.13 – Průběhy příslušných stavů soustavy

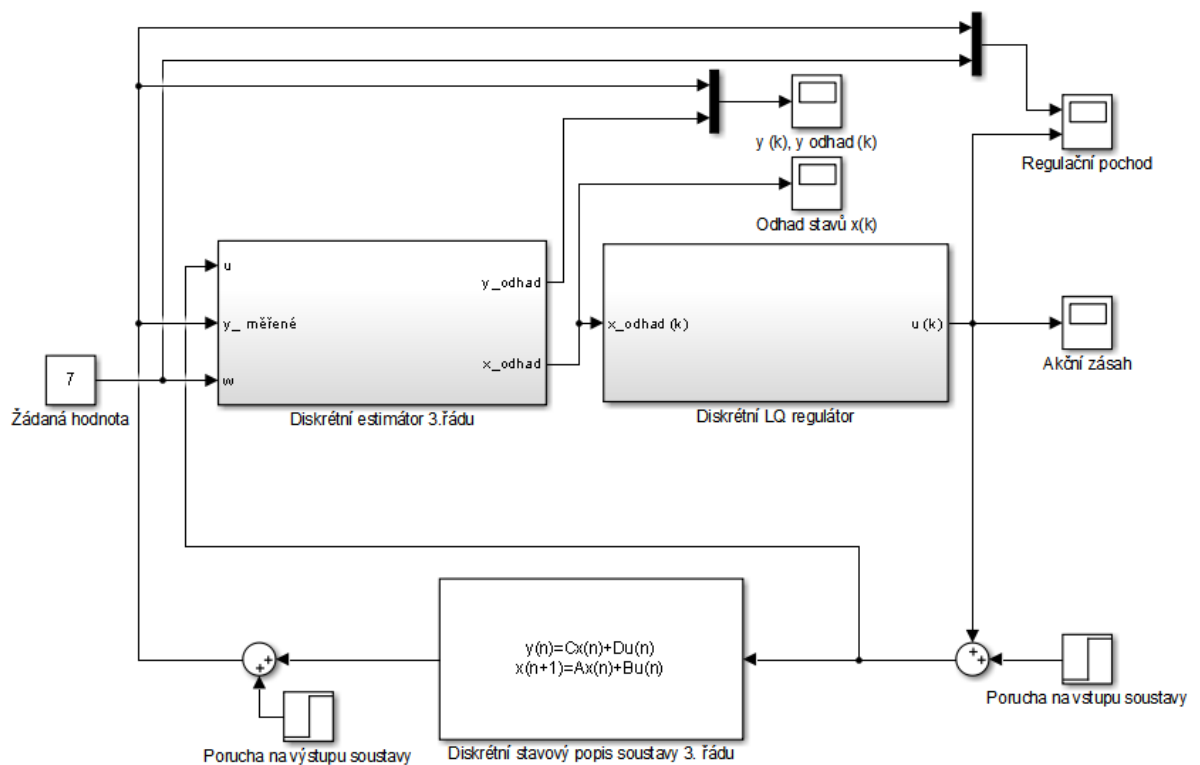


Obr. 2.14 – Realizace LQ regulátoru v prostředí Simulink

2.6 LQ REGULÁTOR S ROZŠÍŘENÍM STAVU O SUMAČNÍ SLOŽKU

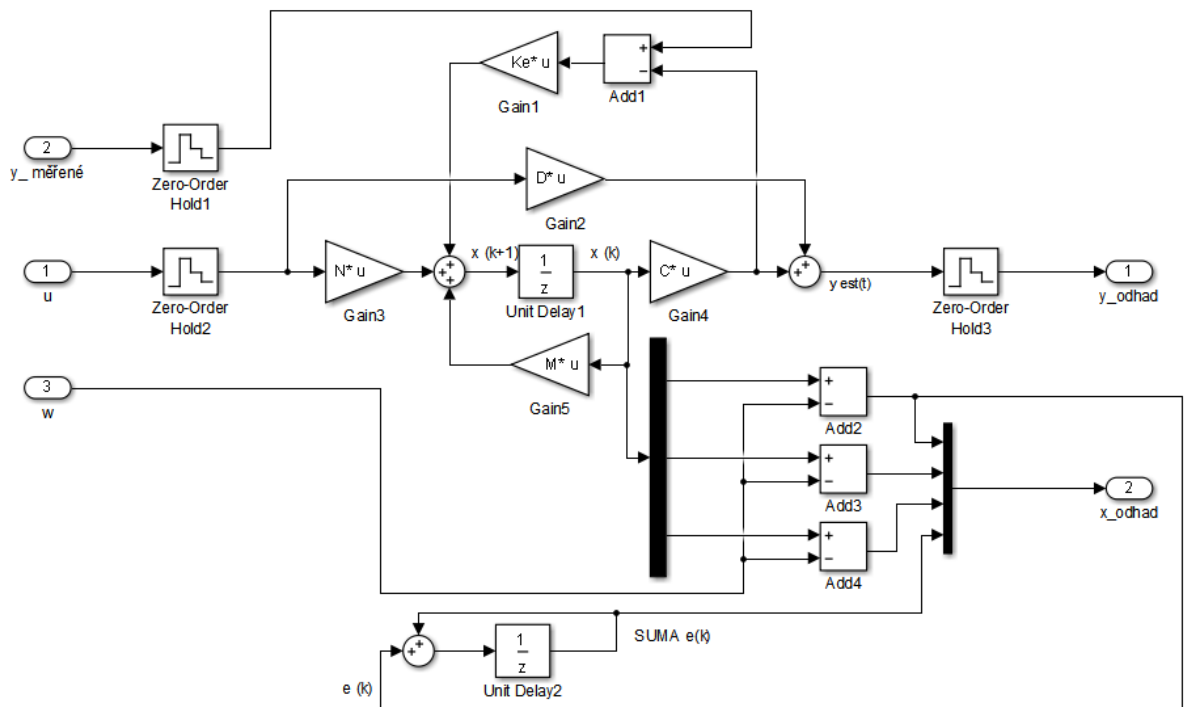
Z důvodu trvalé regulační odchylky, kterou vykazoval regulační obvod s LQ regulátorem, byl stavový vektor rozšířen o složku ve formě sumy odchylek výstupu soustavy od žádané hodnoty. Jelikož došlo ke změně velikosti stavového vektoru, bylo nutné rozšířit matice stavového popisu soustavy tak, aby byla získána matice regulátoru o správném rozměru.

V prostředí Simulink byl právě takový regulátor zapojen do regulačního obvodu, který je zobrazen na obr. 2.15.

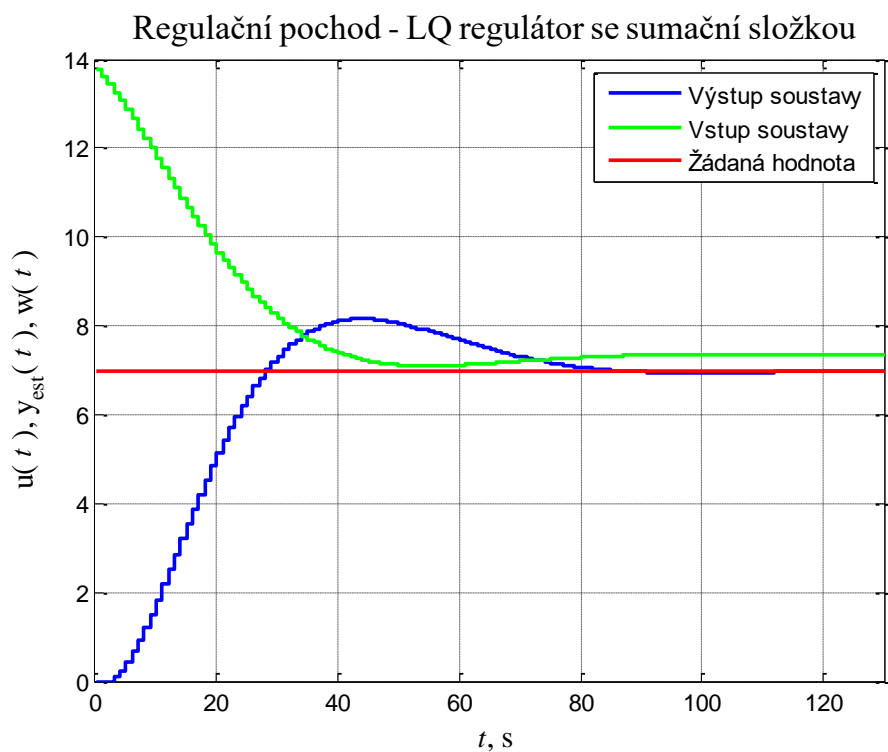


Obr. 2.15 – Regulační obvod v prostředí Simulink

Aby bylo možné zadávat žádanou hodnotu o velikosti jiné než je nula, bylo třeba provést úpravu estimátoru. Došlo tedy k tzv. transformaci stavového vektoru. Od každé složky stavového vektoru byla odečtena žádaná hodnota. Provedená úprava v prostředí Simulink je graficky znázorněna na obr. 2.16.



Obr. 2.16 – Úprava estimátoru pro zadávání žádané hodnoty

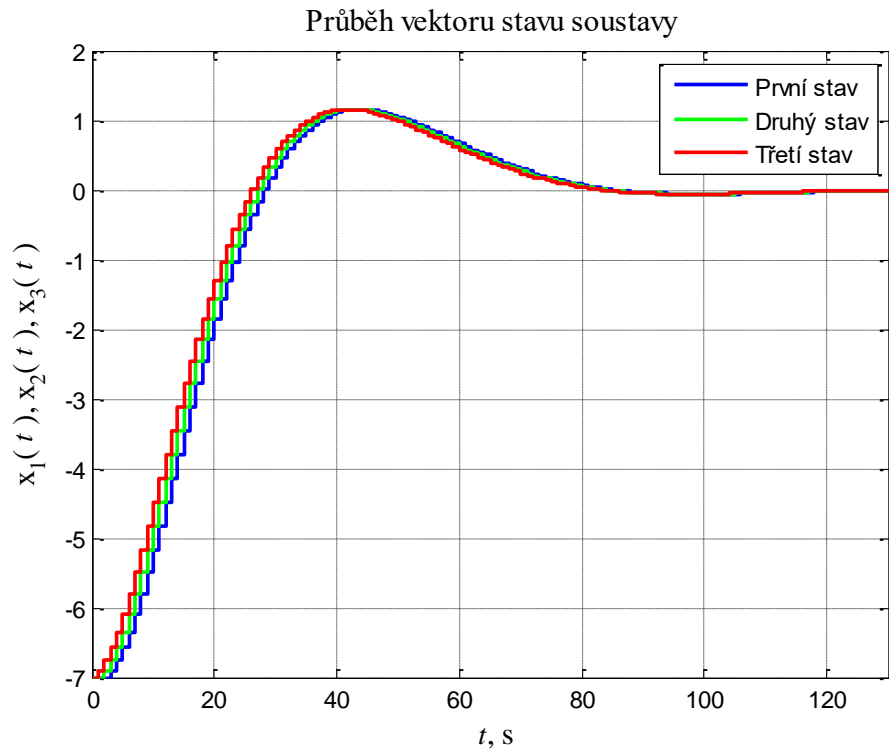


Obr. 2.17 – Regulační pochod v prostředí Simulink

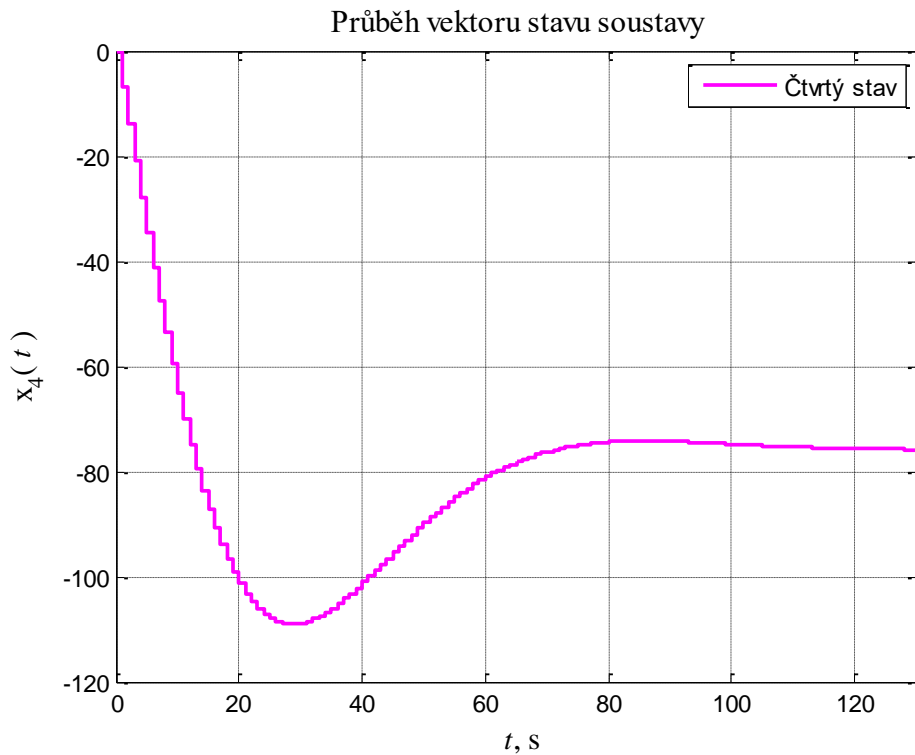
V prostředí Simulink byly provedeny simulace řízení soustavy, jak ukazuje schéma na obr. 2.15. Byl získán regulační pochod, jehož průběh je zachycen na obr. 2.17. Jak je vidět v grafu výše, průběh výstupu soustavy konverguje k nulové regulační odchylce a nekmitá.

V obr. 2.18 byly vykresleny průběhy stavů soustavy, kde první, druhá i třetí složka stavového vektoru odpovídá zpožděné hodnotě výstupní veličiny soustavy.

Naopak čtvrtá složka stavového vektoru byla zobrazena do samostatného grafu v obr. 2.19. Tato složka představuje sumační člen.



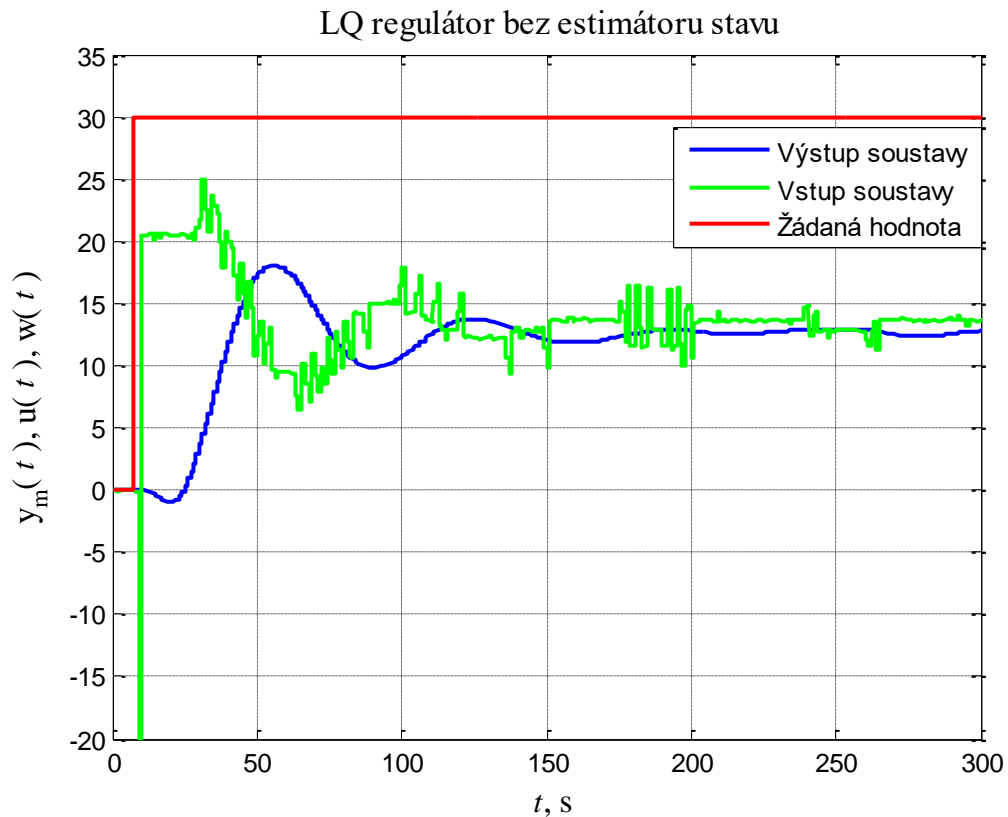
Obr. 2.18 – Grafické znázornění stavového vektoru



Obr. 2.19 – Grafické znázornění sumační složky vektoru stavu

2.7 OTESTOVÁNÍ ESTIMÁTORU NA REÁLNÉ SOUSTAVĚ

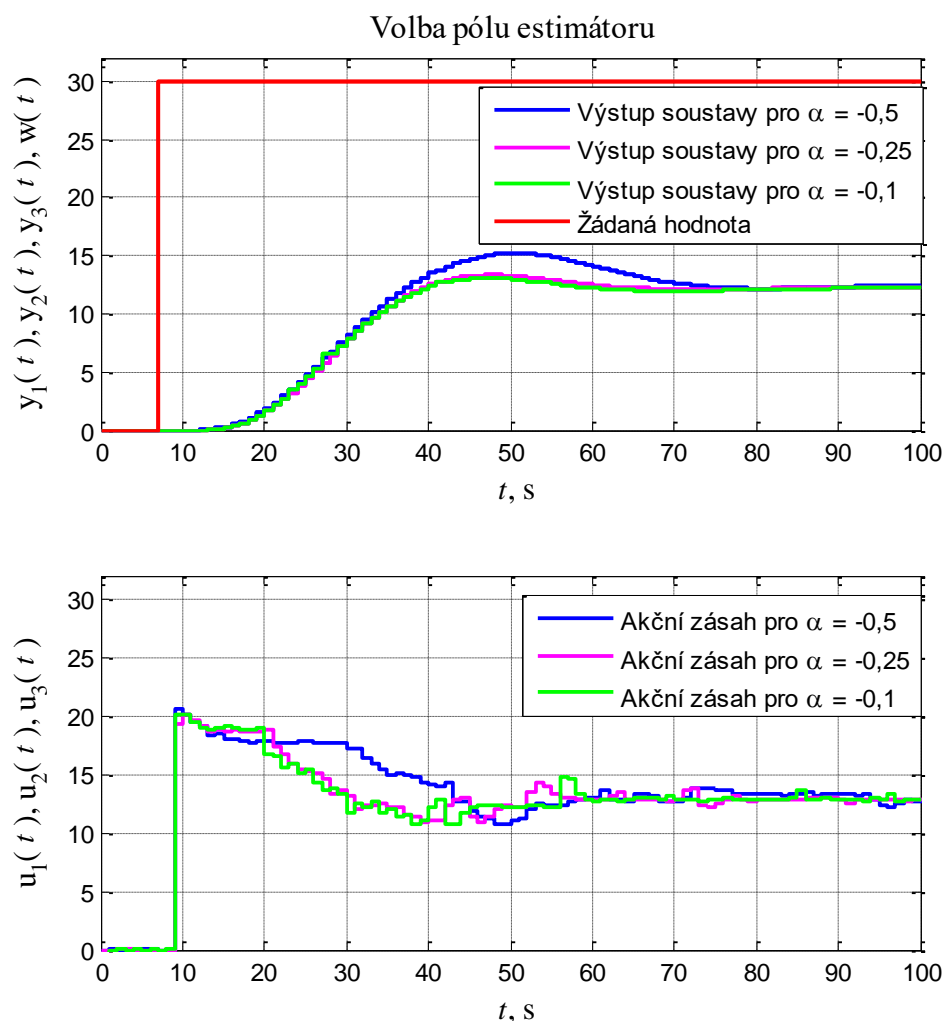
Pro úplnost byla otestována i základní verze LQ regulátoru, kde nebyl použit estimátor, nýbrž jako stavy byly brány zpožděné hodnoty měřeného výstupu soustavy. Regulační pochod zachycuje obr. 2.20.



Obr. 2.20 – Regulační pochod s LQ regulátorem bez estimátoru

V porovnání s LQ regulátorem pracujícím s vektorem stavů získaného pomocí estimátoru je zřetelný rozkmit akčního zásahu regulátoru. Dále také počáteční akční zásah regulátoru působící v právě opačném smyslu, než je požadováno. Regulační pochod je poměrně kmitavý a tudíž k ustálení výstupu došlo v čase 281 s, tj. na mnohem delším intervalu než je tomu v případě využití estimátoru stavu. Trvalá regulační odchylka mírně oscillovala okolo hodnoty 17,30 neboli 57 %. Regulační pochod, kde byl použit LQ regulátor společně s estimátorem stavu, je vyobrazen na obr. 2.23.

Dále byla vyšetřována kvalita regulačního pochodu v závislosti na nastavení pólu estimátoru stavu. V regulačním obvodu byla zapojena základní verze LQ regulátoru. Ten kvůli přítomnosti trvalé regulační odchylky nebyl vhodný pro regulaci, avšak mohl být použit pro testovací účely. Penalizační matice LQ regulátoru byly zvoleny $\mathbf{Q}=1, \mathbf{R}=1$. Byly postupně nastavovány $\alpha = -0,5$, $\alpha = -0,25$, a $\alpha = -0,1$, přičemž pól estimátoru odpovídá $-\alpha$. Získané průběhy výstupní veličiny při skokové změně žádané hodnoty byly znázorněny v obr. 2.21.

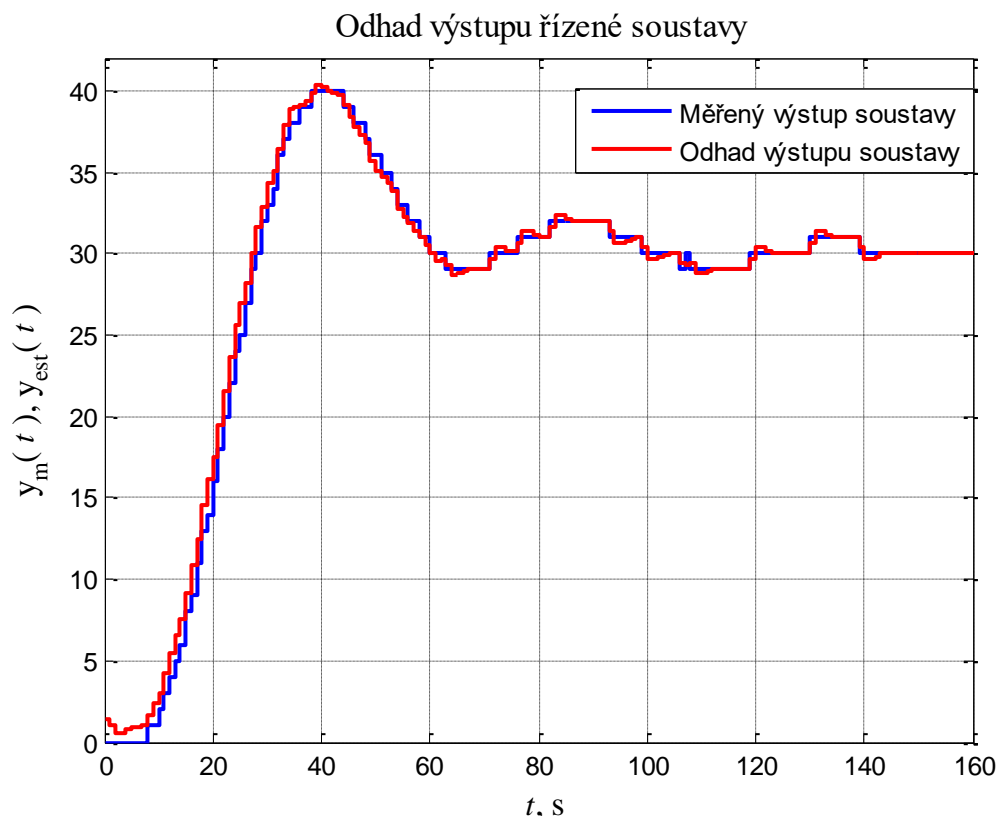


Obr. 2.21 –Vliv pólů estimátoru stavu na výsledné řízení

Původní nastavení estimátoru $\alpha = -0,5$, které v simulacích podávalo uspokojivé výsledky, se však ukázalo v praxi jako ne úplně vhodné. Empiricky bylo zjištěno, že snížení časové konstanty estimátoru zvýší citlivost odezvy estimátoru na šумы, na druhou stranu byl ale regulační pochod urychlen a vykazoval výrazně nižší hodnotu překmitu. Jako optimální hodnota nastavení estimátoru byla empiricky zvolena $\alpha = -0,25$. Takto nastavený estimátor

stavu byl použit v regulačním obvodu se zapojeným LQ regulátorem, kde došlo k rozšíření stavového vektoru soustavy o sumační složku.

Porovnání průběhu odhadovaného výstupu soustavy se skutečným výstupem vykresluje obr. 2.22. Na intervalu od nuly do 16 s je viditelná počáteční odchylka mezi odhadovaným a skutečným výstupem soustavy. Její vznik je zapříčiněn skutečností, že řízená soustava se před zahájením regulace nacházela ve stavu, který není estimátoru známý. Docházelo k postupnému upřesňování odhadu výstupu soustavy a v čase 16 s byla hodnota odhadu výstupu velice blízká

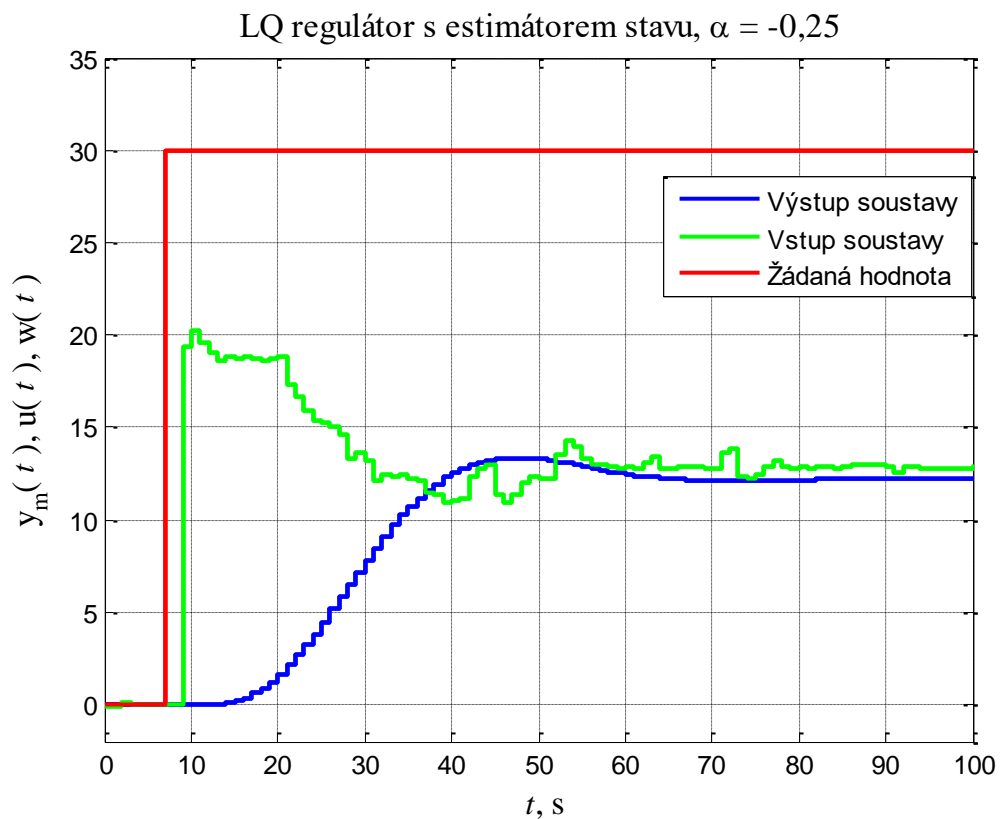


Obr. 2.22 – Srovnání měřeného a odhadovaného výstupu soustavy

hodnotě skutečného výstupu soustavy. Po dobu kdy se výstup soustavy diskrétně měnil v jednotlivých krocích, hodnota odhadu stavu se vždy pohybovala v určitém rozmezí okolo měřeného výstupu soustavy. Avšak po ustálení měřeného výstupu soustavy došlo v časovém okamžiku 151 s k dosažení naprosté shody měřeného a odhadovaného výstupu soustavy.

2.8 OVĚŘENÍ FUNKCE REGULÁTORŮ NA REÁLNÉ SOUSTAVĚ

Jak potvrzuje obr. 2.23, estimátor stavu se chová jako filtr. Rozdíl je markantní při porovnání akčních zásahů v obr. 2.20 a obr. 2.23. V případě použití estimátoru ke stanovení vektoru stavu, který byl rozhodující pro výpočet akčního zásahu LQ regulátoru, vznikl rovnoměrný, ne rozmítaný průběh akčního zásahu jako tomu bylo v případě absence estimátoru. Průběh výstupu soustavy byl výrazně aperiodický, s mnohem kratší dobou ustálení než tomu bylo v případě, kdy nebyl použit estimátor stavu. K ustálení výstupu došlo v čase 64 s. Trvalá regulační odchylka dosahovala hodnoty 17,76 neboli 59 %.



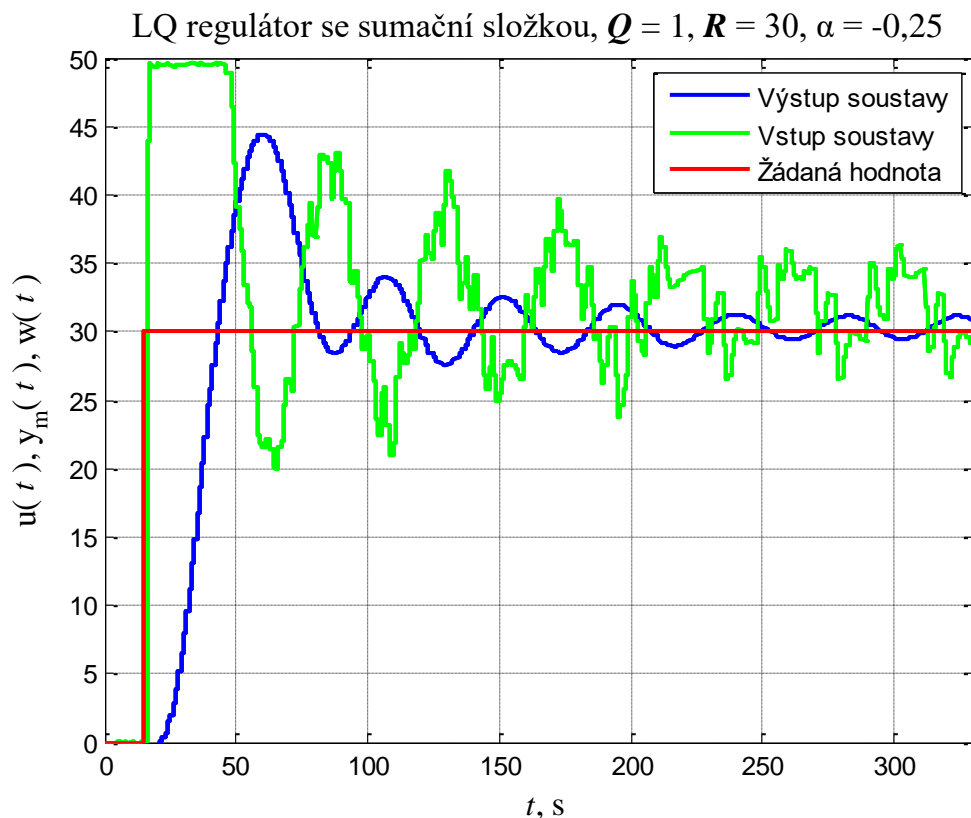
Obr. 2.23 – Regulační pochod s LQ regulátorem s estimátorem

Jak už bylo uvedeno, kvůli přítomnosti trvalé regulační odchylky, byl odvozen LQ regulátor pracující s vektorem stavu soustavy rozšířeným o sumační složku. Již ze simulací bylo zřejmé, že použitelnost takového regulátoru je mnohem vyšší, než je tomu u předchozí verze. Zbytek oddílu se tedy zabývá volbou váhových matic právě pro tento regulátor.

Průběhy regulačního pochodu skutečné soustavy řízené LQ regulátorem, který ke své funkci využívá rozšířený vektor soustavy o sumační složku, jsou zakresleny v obr. 2.24.

Nejprve byla zvolena penalizace ve formě $\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ a $\mathbf{R} = 30$. Matice LQ regulátoru

odpovídající takovéto volbě penalizačních matic byla ve tvaru $\mathbf{K}_R = [5,1962 \ 3,9470 \ -12,3567 \ -0,1745]$. Toto nastavení však vedlo na poměrně kmitavý

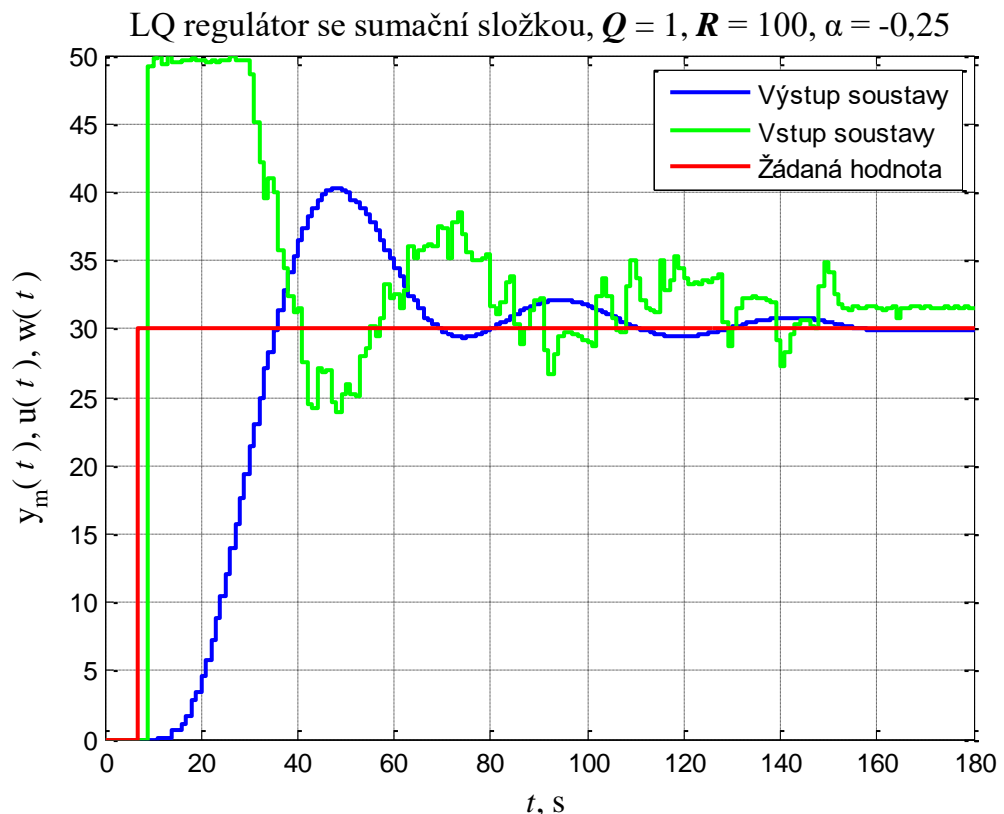


Obr. 2.24 – Regulační pochod skutečné soustavy, kmitavý průběh výstupní veličiny. V intervalu 15 až 49 s je viditelná saturace akčního zásahu, který je takto ohraničen kvůli omezenému rozsahu výstupu regulátoru. Došlo tedy v kritériu k navýšení velikosti váhy akčního zásahu.

V obr. 2.25 je možné vidět regulační pochod s upravenými hodnotami penalizací v kvadratickém kritériu. Hodnoty penalizace byly tedy upravena na

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ a } \mathbf{R} = 100. \text{ Matice LQ regulátoru vypočítaná na základě těchto váhových}$$

matic, pak byla ve tvaru $\mathbf{K}_R = [3,5552 \ 2,7602 \ -8,2829 \ -0,0970]$. Přestože byla váha akčního

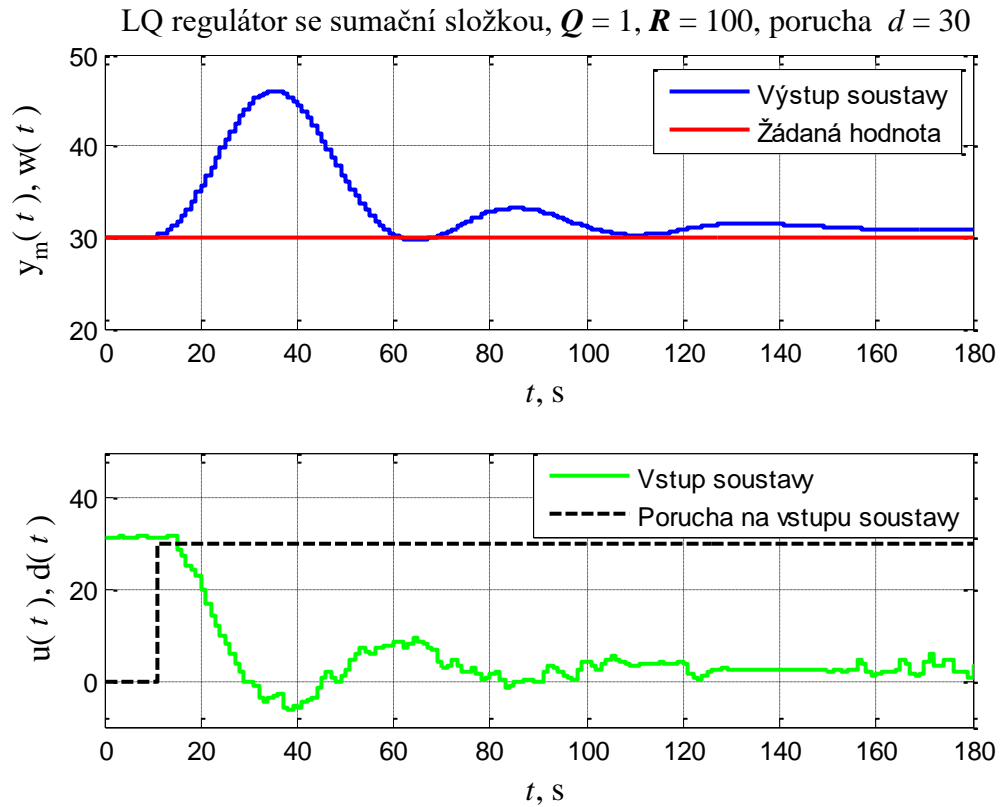


Obr. 2.25 – Regulační pochod skutečné soustavy, aperiodický

zásahu v kvadratickém kritériu navýšena, je však opět patrná saturace akčního zásahu v intervalu 8 až 26 s. Průběh výstupní veličiny soustavy se stal výrazně méně kmitavým. Výstup soustavy se oproti předchozímu nastavení váhových matic regulátoru ustálil na žádané hodnotě za 157 s.

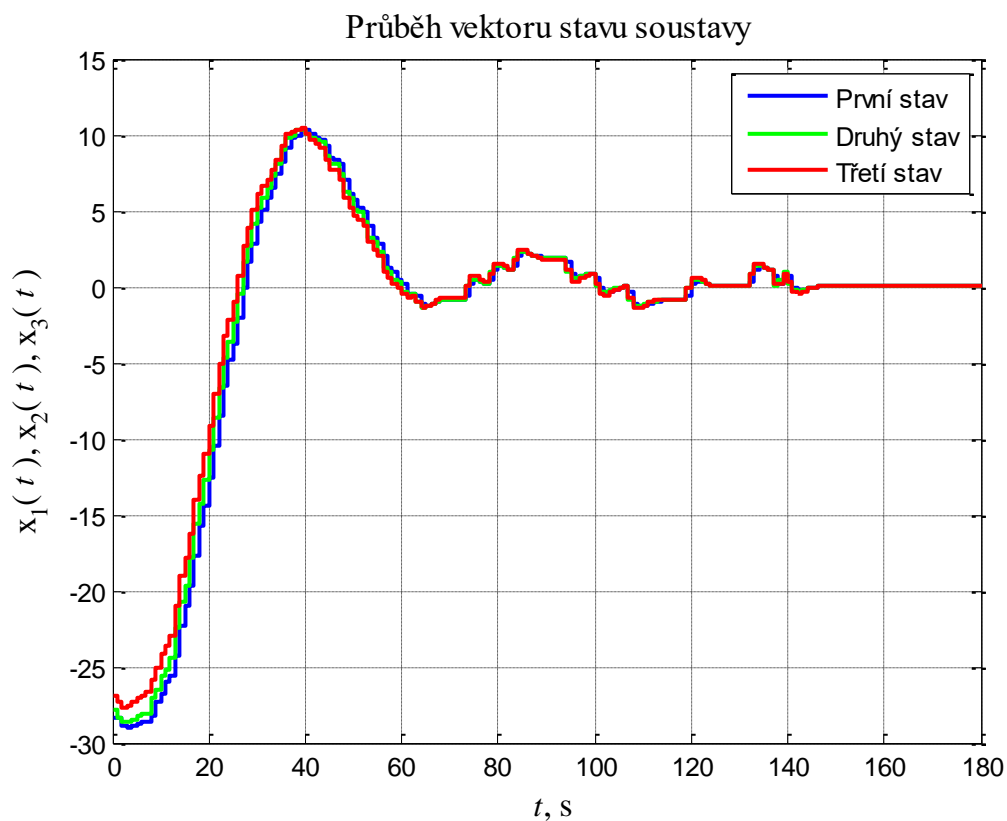
Právě takové nastavení matic \mathbf{Q} a \mathbf{R} bylo považováno za nejvhodnější z praktického hlediska. Vlastnosti tohoto nastavení byly testovány pro skokovou změnu poruchy na vstupu soustavy.

Na vstup soustavy byla pomocí grafického uživatelského rozhraní Labctrl přivedena porucha o velikosti 30. Průběh výstupu soustavy znázorňuje obr. 2.26. Proti této skokové poruše, která stále přetrvává, reaguje regulátor akčním zásahem k jejímu odstranění.

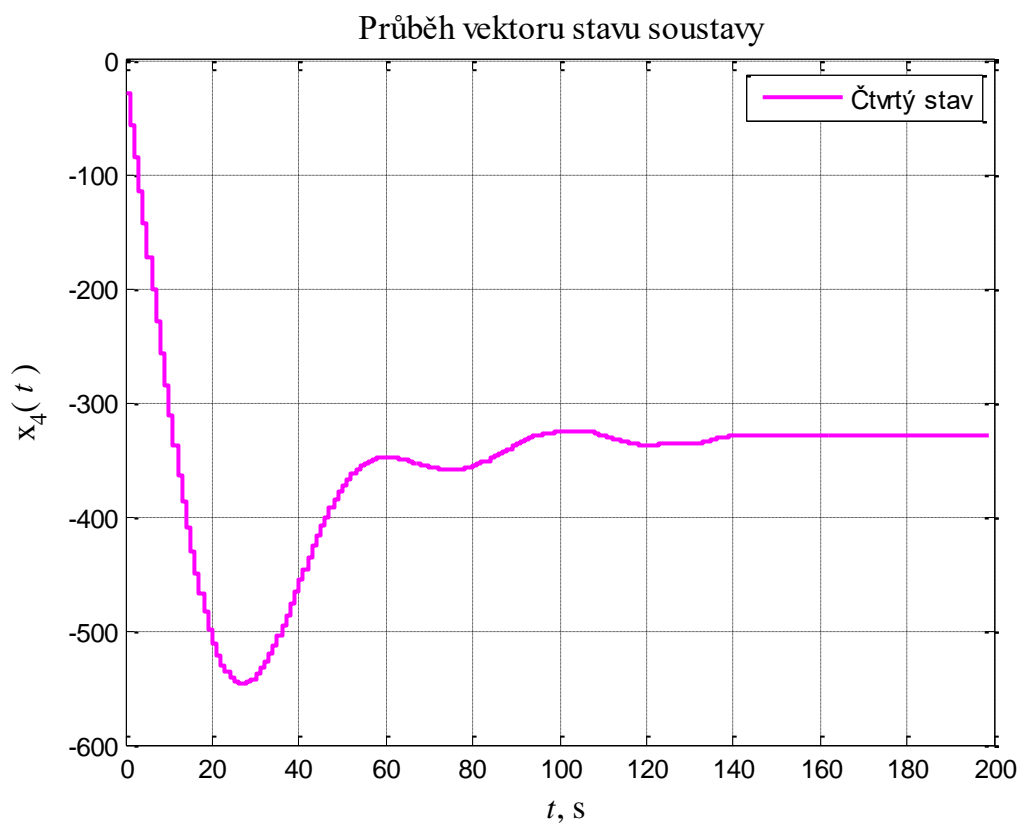


Obr. 2.26 – Reakce regulátoru na působení poruchy

Průběh stavového vektoru řízené soustavy při skokové změně žadané hodnoty znázorňují obr. 2.27 a obr. 2.28.



Obr. 2.27 – Grafické znázornění vektoru stavů soustavy



Obr. 2.28 – Grafické znázornění sumační složky stavu

3 ZHODNOCENÍ

V této práci byl využit estimátor stavu úplného řádu a byly testovány jeho varianty dle zvoleného pólu estimátoru $-\alpha$. S estimátorem stavu pracovaly dva regulátory, které byly postupně odvozeny za účelem regulace identifikované soustavy. První z nich by se dal považovat za základní tvar LQ regulátoru. Druhý z nich pak pracoval s rozšířeným vektorem stavu soustavy o sumační složku, která zajišťovala konvergenci výstupní veličiny soustavy k žádané hodnotě.

Byl sepsán konečný přehled parametrů regulace identifikované soustavy, které závisely nejen na zvoleném typu regulátoru, ale i na jeho nastavení. Rozdíl mezi jednotlivými nastaveními pólů estimátoru stavu je znatelný. Všechny podstatné parametry byly zaznamenány do tab. 3.1.

Tab. 3.1 – Přehled řízení soustavy jednotlivými regulátory

	Simulace				Reálná soustava			
	Kritérium	Doba regulace, s	Relativní překmit, %	Trvalá reg. Odchylka, %	Kritérium	Doba regulace, s	Relativní překmit, %	Trvalá reg. Odchylka, %
LQ bez estimátoru	3008,10	161,00	0	56,00	6530,00	315,00	0	57,60
LQ, Q=1, R=1, $\alpha = -0,5$	2038,70	91,00	0	52,72	5540,00	131,00	0	58,10
LQ, Q=1, R=1, $\alpha = -0,25$	2052,40	88,00	0	52,73	5650,00	102,00	0	59,30
LQ, Q=1, R=1, $\alpha = -0,1$	2062,70	84,00	0	52,73	5670,00	96,00	0	59,30
LQ se sumační složkou, Q=1, R=30, $\alpha = -0,25$	567,10	180,00	26,00	0	3110,00	394,00	46,00	0
LQ se sumační složkou, Q=1, R=100, $\alpha = -0,25$	623,70	136,00	16,00	0	1070,00	157,00	33,00	0

Z tabulky výše tedy vyplývá, že veškeré regulační pochody reálného procesu trvaly déle, než bylo původně předpokládáno z výsledků simulací. Regulace skutečné soustavy vykazuje vyšší hodnoty, jak kritéria absolutní hodnoty regulační odchylky, tak i doby regulace a relativního překmitu, než regulační pochody získané simulacemi.

Co se týče základního tvaru LQ regulátoru, nejlepší výsledky podával v kombinaci s estimátorem stavu nastaveným pro $\alpha = -0,1$. Jednalo se o nejkratší regulační pochod, ale bohužel s existencí trvalé regulační odchylky o hodnotě 59,3 %. Snížením váhy penalizace akčního zásahu na jednu setinu aktuálního nastavení v kvadratickém kritériu, tedy $\mathbf{R} = 0,01$, byla v simulaci dosažena hodnota trvalé regulační odchylky 7 %. To by však při řízení reálného procesu znamenalo aplikaci nadměrných akčních zásahů, které není regulátor schopen vykonat.

Opakem k tomuto případu byl totožný LQ regulátor, avšak pracující bez estimátoru. Ten pracoval pouze se zpožděnými hodnotami výstupu diskrétní soustavy třetího řádu, jejíž výstup bylo možné měřit. Doba regulace dosahovala druhé nejvyšší hodnoty ze všech. Průběh výstupu soustavy byl kmitavý, zatížený šumem na výstupu soustavy.

Z testovaných regulátorů byl jistě pro praxi nejpoužitelnějším LQ regulátor doplněný sumační složkou. Při praktické realizaci byla rozhodující volba váhových matic \mathbf{Q} a \mathbf{R} . Matice \mathbf{Q} byla ve všech případech volena jako pozitivně semidefinitní. Kde složky Q_{11} a Q_{44} byly rovny jedné, ostatní prvky byly nulové. Hodnota matice \mathbf{R} však musela nabývat mnohem vyšších hodnot, než tomu bylo u matice \mathbf{Q} . Hodnoty matice \mathbf{R} nižší než třicet způsobovaly naprosto nepoužitelné regulační pochody. Neúměrné akční zásahy regulátoru ovlivňovaly výstupní veličinu soustavy do takové míry, že kmitala harmonickými kmity. Jako optimální hodnota byla empiricky nastavena $\mathbf{R} = 100$. Právě takovýto regulátor zapojený do regulačního obvodu s řízeným procesem dokázal odstranit regulační odchylku na nulu. Doba regulace nabývala hodnoty 157 s, což odpovídá třetí nejdelší době ustálení výstupu soustavy. Na druhou stranu kritérium absolutní hodnoty regulační odchylky vykazuje nejnižší hodnotu ze všech regulačních pochodů.

4 ZÁVĚR

Zadané cíle diplomové práce byly splněny. Podle logické návaznosti, byly postupně vyřešeny příslušné body zadání. Byla prokázána a odzkoušena možnost regulace diskrétní soustavy vyššího řádu za pomoci jednoduchého mikropočítače. Ten pracoval na principu řízení soustavy rozděleným akčním zásahem, což odpovídá mnoha procesům v potravinářském, či chemickém průmyslu. Jedná se o regulaci teploty u soustav s možností topit i chladit současně. Nebo také regulace koncentrace, kdy je za pomoci dvou ventilů regulován přítok jednotlivých substancí do směšovací nádoby.

Námětem dalšího výzkumu by mohlo být sestavení zesilovače akčního zásahu regulátoru. Ten by zdvojnásobil výstup regulátoru, a tudíž by byl využit celý rozsah akčního členu. Došlo by tak k odstranění případné saturace akčního zásahu v čase skokové změny žádané hodnoty nebo poruchy. Porovnání kvality regulačních pochodů uvedených v této práci s průběhy výstupu soustavy, která by byla řízena takto upraveným regulátorem, by rovněž spadalo do navazujícího výzkumu.

POUŽITÁ LITERATURA

- Arduino. *Language Reference* [online]. Italy: Arduino, 2017 [cit. 2017-04-22]. Dostupné z: <https://www.arduino.cc/>
- BALÁTĚ, Jaroslav 2003. *Automatické řízení*. [online]. Praha: BEN - technická literatura. 664 s. ISBN 80-7300-020-2. Dostupné z: http://utb.tsx.cz/Automaticke_rizeni.PDF
- BLAHA, Petr a Petr VAVŘÍN. *Řízení a regulace I* [online]. Vyd. 1. 2010 [cit. 2. 1. 2017]. Dostupné z: http://www.uamt.feec.vutbr.cz/~richter/vyuka/0809_BR_R1/texty/brr1.pdf
- BOSGRA, Okko, Huibert KWAKERNAAK a Gjerrit MEINSMA. *Design Methods for Control Systems* [online]. Delft, The Netherlands: Dutch Institute of Systems and Control, 2007 [cit. 2017-02-17]. Dostupné z: <http://wwhome.math.utwente.nl/~meinsmag/dmcs/dmcs0708.pdf>
- DOSTÁL, Petr a Radek MATUŠŮ. *Stavová a algebraická teorie řízení* [online]. Zlín: Univerzita Tomáše Bati ve Zlíně, 2010 [cit. 2017-04-15]. ISBN 978-80-7318-991-4. Dostupné z: http://www.utb.cz/file/15548_1_1/
- MODRLÁK, Osvald. *Základy analýzy a syntézy ve stavovém prostoru* [online]. 2004 [cit. 2017-02-26]. Dostupné z: http://vincenzo.servegame.com/upload/killer/tar2_zas.pdf
- OGATA, Katsuhiko. *Modern control engineering* [online]. 5th. United States of America, Boston: Prentice-Hall, 2010 [cit. 2017-02-17]. ISBN 01-361-5673-8. Dostupné z: http://portal.tpu.ru:7777/SHARED/s/SMIKE/Uchebnaya/Tab5/Modern_Control_Engineering.pdf
- STREJC, Vladimír 1978. *Stavová teorie lineárního diskrétního řízení*. Praha: Academia.
- STENGEL, Robert F. *Stochastic optimal control: theory and application*. New York: Wiley, c1986. ISBN 978-0471864622.
- ŠTECHA, Jan a Vladimír HAVLENA. *Teorie dynamických systémů*. 1. Praha: Ediční středisko ČVUT, 2005.
- ŠVARC, Ivan. *Teorie automatického řízení* [online]. Brno: Vysoké učení technické v Brně, 2003 [cit. 2017-02-26]. Dostupné z: http://www.fsiforum.cz/upload/soubory/knihy/Automatizace/Teorie.automatickeho.rizeni_Svarc_2003.pdf
- VROŽINA, Milan a JANČÍKOVÁ, Zora a DAVID, Jiří. 2012. *Identifikace systémů* [online]. Ostrava. 178 s.[cit. 27. 5. 2015]. ISBN 978-80-248-2594-6. Dostupné z: <http://www.person.vsb.cz/archivcd/FMMI/IS/Identifikace%20systemu.pdf>

PŘÍLOHY

Příloha A

Příloha B-CD-ROM

Příloha k diplomové práci

Řízení reálného procesu stavovým regulátorem LQ

Bc. Ondřej Zouhar

Zdrojové kódy

ALGORITMY Z MATLABU

```
% Výpočet základního LQ regulátoru
% Inicializace potřebných proměnných
load('hodnotyZiskanePomociIdentifikace.mat')
a0 = a(1,4);
a1 = a(1,3);
a2 = a(1,2);
M = [0 1 0; 0 0 1; -a(1,4) -a(1,3) -a(1,2)];
N = [0;0;b(1,end)];
C = [1 0 0];
D = 0;
ind = 1;
Pf = zeros(3,3); % Penalizace konvergence k nule v nekonečnu
% Riccatiho rce. bude řešena "odzadu" jako první hodnota Sk+1 se volí nula
Skf = Pf;
Qk = 1*eye(3); % Penalizace stavů
Rk = 0.01; % Penalizace akčních zásahů
T = zeros(3,1); % Matice nul
Sk = zeros(3,3,300); % Inicializace vektoru výsledků
norma = 1; % Prvotní hodnota normy >> než tolerance normy
% První krok výpočtu
Sk(:,:,1) = Qk + M'*Skf*M - (T + M'*Skf*N)*...
    inv(Rk + N'*Skf*N)*(T' + N'*Skf*M);
% Maticový iterační výpočet Riccatiho rovnice
while(norma >= 0.001)
    ind = ind + 1;
    Sk(:,:,ind) = Qk + M'*Sk(:,:,ind-1)*M - (T + M'*Sk(:,:,ind-1)*N)*...
        inv(Rk + N'*Sk(:,:,ind-1)*N)*(T' + N'*Sk(:,:,ind-1)*M);
    norma = norm(Sk(:,:,ind))-norm(Sk(:,:,ind-1));
end
disp('Výsledná hodnota Sk: ')
disp(Sk(:,:,ind))
Sk_vysledek = Sk(:,:,ind);
% Regulace diskrétní soustavy 3. řádu
x = zeros(3,1,100);
t = linspace(0,98,100);
u = zeros(1,100);
y = u;
x(:,:,1) = [9; 4; 7];
for krok = 1:1:length(x)-1
    % Výpočet akčního zásahu regulátoru
    u(krok) = -(inv(Rk + N'*Sk(:,:,ind)*N)*(T' + N'*Sk(:,:,ind)*M))*...
        x(:,:,krok);
    % Výpočet vektoru stavu soustavy
    x(:,:,krok+1) = M*x(:,:,krok) + N*u(krok);
    x1(krok) = x(1,1,krok);
    x2(krok) = x(2,1,krok);
    x3(krok) = x(3,1,krok);
    % Odhad výstupu soustavy
```

```

    y(krok) = C *x(:,krokov);
end
% Vykreslení stavů, akční a regulované veličiny
subplot(3,1,1), stairs(t(1:99),x1,'Linewidth',2),xlabel('t, k*T'), ylabel('x1, x2, x3')
title('Hodnoty stavů v intervalech vzorkování')
hold on, grid on
stairs(t(1:99),x2,'r','Linewidth',2),
stairs(t(1:99),x3,'g','Linewidth',2)
legend('x1(k)','x2(k)','x3(k)')
axis([0 60 -1 10])
subplot(3,1,2), stairs(t,u,'Linewidth',2),xlabel('t, k*T'), ylabel('u')
grid on
title('Průběh akční veličiny')
legend('u(k)')
axis([0 60 -40 20])
subplot(3,1,3), stairs(t,y,'r','Linewidth',2),xlabel('t, k*T'), ylabel('y')
grid on
title('Průběh regulované veličiny')
legend('y(k)')
axis([0 60 -2 10])

% Výpočet estimátoru pro základní LQ regulátor
% Inicializace potřebných proměnných
c = 200;
Tvz = 1;
load('hodnotyZiskanePomociIdentifikace.mat')
% Trojnásobný pól estimátoru -1/alfa
a0 = a(1,4);
a1 = a(1,3);
a2 = a(1,2);
% Volba trojnásobného polu -1/alfa
alfaZ = -0.5;
% Prvky matice K získané dosazovací metodou
k1 = (3*alfaZ)-a2;
k2 = -a1-a2*k1+3*alfaZ^2;
k3 = (alfaZ^3)-a0-a1*k1-a2*k2;
% Prvky matice K získané Gaussovou eliminační metodou x =A\b
maticePoly = [1 0 0; a2 1 0; a1 a2 1];
matVysledku = [(3*alfaZ)-a2;(3*alfaZ^2)-a1; (alfaZ^3)-a0];
Ke = maticePoly\matVysledku;
% Zápis jednotlivých matic diskrétního stavového popisu
M = [0 1 0; 0 0 1; -a(1,4) -a(1,3) -a(1,2)];
N = [0;0;b(1,end)];
C = [1 0 0];
D = 0;
StavovyD = ss(M,N,C,D,1)
% Porovnání diskrétního stavového a diskrétního V/V popisu
figure
[yv,tv] = step(Dsystem3,c);
plot(tv,yv,'r*')

```

```

hold on
[ys,ts] = step(StavovyD,c);
stairs(ts,ys,'b')
legend('Diskrétní vstupně-výstupní popis', 'Diskrétní stavový popis')
grid on
sim REG_OBVOD_S_LQ

% Výpočet LQ regulátoru rozšířeného o sumační složku
% Inicializace potřebných proměnných
load('hodnotyZiskanePomociIdentifikace.mat')
a0 = a(1,4);
a1 = a(1,3);
a2 = a(1,2);
M = [0 1 0; 0 0 1; -a(1,4) -a(1,3) -a(1,2)];
N = [0;0;b(1,end)];
C = [1 0 0];
D = 0;
% Přepočítání stavových matic na jejich rozšířený tvar
rad = size(M, 1);
Mr = [M zeros(rad, 1); C 1];
Nr = [N; D];
Cr = [C 0];
ind = 1;
Pf = zeros(4,4); % Penalizace konvergence k nule v nekonečnu
% Riccatiho rce. bude řešena "odzadu" jako první hodnota Sk+1 se volí nula
Skf = Pf;
% První verze regulátoru
% Qk = [1 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 1]; % Penal. stavů
% Rk = 30; % Penal. akčních zásahů
% Druhá verze regulátoru
Qk = [1 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 1];
Rk = 100;
T = zeros(4,1); % Nulová matice
Sk = zeros(4,4,300); % Inicializace vektoru výsledků
norma = 1; % Prvotní hodnota normy >> než tolerance normy
% První krok iteračního výpočtu
Sk(:,:,1) = Qk + Mr'*Skf*Mr - (T + Mr'*Skf*Nr)*...
    inv(Rk + Nr'*Skf*Nr)*(T' + Nr'*Skf*Mr);
% Maticový iterační výpočet Riccatiho rovnice
while(norma >= 0.001)
    ind = ind + 1;
    Sk(:,:,ind) = Qk + Mr'*Sk(:,:,ind-1)*Mr - (T + Mr'*Sk(:,:,ind-1)*Nr)*...
        inv(Rk + Nr'*Sk(:,:,ind-1)*Nr)*(T' + Nr'*Sk(:,:,ind-1)*Mr);
    norma = norm(Sk(:,:,ind))-norm(Sk(:,:,ind-1));
end
disp('Výsledná hodnota Sk: ')
disp(Sk(:,:,ind))
Sk_vysledek = Sk(:,:,ind);
% Regulace diskrétní soustavy 3. řádu
x = zeros(4,1,100);

```

```

t = linspace(0,98,100);
u = zeros(1,100);
y = u;
w = 3;
x(:,1) = [9; 4; 7; 0];
for krok = 1:length(x)-1
    % Výpočet akčního zásahu regulátoru
    u(krok) = -(inv(Rk + Nr*Sk(:,ind)*Nr)*(T' + Nr*Sk(:,ind)*Mr))*...
        x(:,krok);
    % Výpočet vektoru stavu soustavy
    x(1:3,:,krok+1) = (M*x(1:3,:,krok) + N*u(krok));
    if(krok >= 2)
        x(4,:,krok+1) = (x(4,:,krok) + x(1,:,krok)) ;
    else
        x(4,:,krok+1) = 0;
    end
    x1(krok) = x(1,1,krok);
    x2(krok) = x(2,1,krok);
    x3(krok) = x(3,1,krok);
    x4(krok) = x(4,1,krok);
    y(krok) = (Cr *x(:,krok))+w;
end
% Vykreslení stavů, akční a regulované veličiny
subplot(3,1,1)
stairs(t(1:99),x1,'Linewidth',2),xlabel('t, k*T'), ylabel('x1, x2, x3, x4')
title('Hodnoty stavů v intervalech vzorkování')
hold on, grid on
stairs(t(1:99),x2,'r','Linewidth',2)
stairs(t(1:99),x3,'g','Linewidth',2)
stairs(t(1:99),x4,'y','Linewidth',2)
legend('x1(k)','x2(k)','x3(k)','x4(k)')
axis([0 95 -20 80])
subplot(3,1,2), stairs(t,u,'Linewidth',2),xlabel('t, k*T'), ylabel('u')
grid on
title('Průběh akční veličiny')
legend('u(k)')
axis([0 95 -30 5])
subplot(3,1,3), stairs(t,y,'r','Linewidth',2),xlabel('t, k*T'), ylabel('y')
grid on
title('Průběh regulované veličiny')
legend('y(k)')
axis([0 95 0 15])

%Výpočet estimátoru pro LQ regulátor rozšířený o sumační složku
%Inicializace potřebných proměnných
c = 200;
Tvz = 1;
load('hodnotyZiskanePomociIdentifikace.mat')
% Trojnásobný pól estimátoru -1/alfa
a0 = a(1,4);

```

```

a1 = a(1,3);
a2 = a(1,2);
% Volba trojnásobného polu -1/alfa
alfaZ = -0.25;
% Prvky matice K získané dosazovací metodou
%alfaZ = exp(-alfa);
k1 = (3*alfaZ)-a2;
k2 = -a1-a2*k1+3*alfaZ^2;
k3 = (alfaZ^3)-a0-a1*k1-a2*k2;
% Prvky matice K získané Gaussovou eliminační metodou x =A\b
maticePoly = [1 0 0; a2 1 0; a1 a2 1];
matVysledku = [(3*alfaZ)-a2;(3*alfaZ^2)-a1; (alfaZ^3)-a0];
Ke = maticePoly\matVysledku;

% Zápis jednotlivých matic diskrétního stavového popisu
M = [0 1 0; 0 0 1; -a(1,4) -a(1,3) -a(1,2)];
N = [0;0;b(1,end)];
C = [1 0 0];
D = 0;
rad = size(M, 1);
Mr = [M zeros(rad, 1); C 1];
Nr = [N; D];
Cr = [C 0];
StavovyD = ss(Mr,Nr,Cr,D,1)
% Porovnání diskrétního stavového a diskrétního V/V popisu
figure
[yv,tv] = step(Dsystem3,c);
plot(tv,yv, 'ro','MarkerSize',4, 'MarkerFaceColor','r')
hold on
step(StavovyD,c);
legend('Diskrétní vstupně-výstupní popis','Diskrétní stavový popis')
grid on
% Simulace diskrétního estimátoru 3. řádu v Simulinku
% s vypočítanou maticí K
sim REG_OBVOD_S_LQ_int

```

ALGORITMY ARDUINO

```
//<<<<<<[ DISKRÉTNÍ STAVOVÝ <P> REGULÁTOR S ESTIMÁTOREM 3. ŘÁDU ]>>>>>>
//~~~~~[INCLUDE]~~~~~
//http://www.arduinolibraries.info/libraries/basic-linear-algebra
#include <BasicLinearAlgebra.h> //
//https://code.google.com/archive/p/arduino-pwm-frequency-library/downloads
#include <PWM.h>
//~~~~~[ KONSTANTY ]~~~~~
//Příkazy pro estimátor
const float a0 = 0.4541;
const float a1 = -0.0849;
const float a2 = -1.3616;
const float alfa = -0.1;
const float k1 = (3 * alfa) - a2;
const float k2 = -a1 - a2 * k1 + 3 * pow(alfa, 2);
const float k3 = pow(alfa, 3) - a0 - a1 * k1 - a2 * k2;
const float poleKE[3][1] = {{k1}, {k2}, {k3}};
const Matrix<3, 1> KE(poleKE);
const float poleM[3][3] = {{0, 1, 0}, {0, 0, 1}, {-a0, -a1, -a2}};
const float poleN[3][1] = {{0}, {0}, {0.0073}};
const float poleC[1][3] = {1, 0, 0};
const float poleD[1][1] = {0};
const Matrix<3, 3> M(poleM);
const Matrix<3, 1> N(poleN);
const Matrix<1, 3> C(poleC);
const Matrix<1, 1> D(poleD);
//Matice regulátoru
//const float poleKR[1][3] = {0.0241, 0.0195, -0.0528}; // 1 100
const float poleKR[1][3] = {1.6159, 1.2857, -3.5899}; // 1 1
//const float poleKR[1][3] = {6.0911, 4.5934, -13.9971}; // 1 0.1
const Matrix<1, 3> KR(poleKR);
```



```

//~~~~~[ VSTUPY ]~~~~~
const byte MerenyVystup = A0;
const byte MerenyVystup2 = A1;
//~~~~~[ VÝSTUPY ]~~~~~
const byte AkcniZasah = 9;
const byte AkcniZasah2 = 10;
//~~~~~[ PROMĚNNÉ ]~~~~~
const float w = 30;
float poleW1[3][1] = {{w}, {w}, {w}};
Matrix<3, 1> w1(poleW1);
Matrix<1, 1, float> Yest;
Matrix<1, 1, float> Ym;
int Ym1;
int Ym2;
Matrix<1, 1, float> u;
float poleXest1[3][1] = {{9}, {4}, {7}}; //Náhodný minulý stav soustavy
Matrix<3, 1> Xest1(poleXest1); //Náhodný minulý stav soustavy
float poleXest[3][1] = {{0}, {0}, {0}}; //Vektor pro výpočet aktuálního stavu soustavy
Matrix<3, 1> Xest(poleXest); //Vektor pro výpočet aktuálního stavu soustavy
int32_t frekvencePWM = 900; //Nastavení frekvence PWM v jednotkách Hz
//~~~~~[ SETUP ]~~~~~
void setup() {
  Serial.begin(9600);
  analogReference(DEFAULT);
  Serial << "Matice M: " << M << '\n';
  Serial << "Matice N: " << N << '\n';
  Serial << "Matice C: " << C << '\n';
  Serial << "Matice D: " << D << '\n';
  Serial << "Matice regulatoru: " << KR << '\n';
  Serial.println("u(k), Ym(k), Ym1(k), Ym2(k), Yest(k), Xest(k+1)<1,2,3>");
  Xest.Fill(0);
  Xest1.Fill(0);
  Yest.Fill(0);
  pinMode(MerenyVystup, INPUT);
}

```

```

pinMode(MeremyVystup2, INPUT);
pinMode(AkcniZasah, OUTPUT);
pinMode(AkcniZasah2, OUTPUT);
pinMode(LED_BUILTIN, OUTPUT);
// Nastavení všech časovačů kromě časovače 0, aby se neovlivnila činnost časových funkcí
InitTimersSafe();
//Nastavení požadované PWM frekvence na příslušných pinech
bool uspesne = SetPinFrequencySafe(AkcniZasah, frekvencePWM);
bool uspesne2 = SetPinFrequencySafe(AkcniZasah2, frekvencePWM);
Serial.println(uspesne);
Serial.println(uspesne2);
delay(1000);
    if(uspesne&&uspesne2) {
        digitalWrite(LED_BUILTIN, HIGH);
    }
}
//~~~~~[ HLAVNÍ SMYČKA ]~~~~~
void loop() {
    //Estimátor
    Ym1 = (((analogRead(MeremyVystup)*5)/1023.0)/7.24)*100;
    Ym2 = (((analogRead(MeremyVystup2)*5)/1023.0)/7.24)*100;
    Ym(0,0) = Ym1 - Ym2;
    Xest1 = ((M * Xest) + (N * u(0,0)) + (KE * (Ym - Yest)));
    Xest = Xest1;
    Yest = (C * Xest + D * u(0,0));
    //Regulátor
    u = (KR * (Xest1-w1));
    //Akční zásah
    float pomVstupu = (255*u(0, 0)/50)*(100.0/99.0);
    pomVstupu = constrain(pomVstupu, -255, 255);
    if (pomVstupu > 0){
        pwmWrite(AkcniZasah, pomVstupu);
        pwmWrite(AkcniZasah2, 0);
        Serial.print("kladny");
    }
}

```

```

    Serial.print(pomVstupu);
}
else{
    pwmWrite(AkcniZasah, 0);
    pwmWrite(AkcniZasah2, -pomVstupu);
    Serial.print("zaporny");
    Serial.print(pomVstupu);
}
//Vykreslení estimovaných stavů, výstupu a akční veličiny
Serial.print(" ");
Serial.print(Ym(0,0));
Serial.print(" ");
Serial.print(Ym1);
Serial.print(" ");
Serial.print(Ym2);
Serial.print(" ");
int pomVystupu = (Yest(0, 0));
Serial.print(pomVystupu);
Serial.print(" ");
Serial.print(Xest1(0, 0));
Serial.print(" ");
Serial.print(Xest1(1, 0));
Serial.print(" ");
Serial.println(Xest1(2, 0));
pomVstupu = 0;
delay(1000);
}

```

```

//<<<<[ DISKRÉTNÍ STAVOVÝ <PI > REGULÁTOR S ESTIMÁTOREM 3. ŘÁDU ]>>>>
//~~~~~[INCLUDE]~~~~~
//http://www.arduinolibraries.info/libraries/basic-linear-algebra
#include <BasicLinearAlgebra.h> //
//https://code.google.com/archive/p/arduino-pwm-frequency-library/downloads
#include <PWM.h>
//~~~~~[ KONSTANTY ]~~~~~
//Inicializace
const float a0 = 0.4541;
const float a1 = -0.0849;
const float a2 = -1.3616;
const float alfa = -0.25;
const float k1 = (3 * alfa) - a2;
const float k2 = -a1 - a2 * k1 + 3 * pow(alfa, 2);
const float k3 = pow(alfa, 3) - a0 - a1 * k1 - a2 * k2;
const float poleKE[3][1] = {{k1}, {k2}, {k3}};
//Matice estimátoru
const Matrix<3, 1> KE(poleKE);
const float poleM[3][3] = {{0, 1, 0}, {0, 0, 1}, {-a0, -a1, -a2}};
const float poleN[3][1] = {{0}, {0}, {0.0073}};
const float poleC[1][3] = {1, 0, 0};
const float poleD[1][1] = {0};
const Matrix<3, 3> M(poleM);
const Matrix<3, 1> N(poleN);
const Matrix<1, 3> C(poleC);
const Matrix<1, 1> D(poleD);
const float poleCr[1][4] = {1, 0, 0, 0};
const Matrix<1, 4> Cr(poleCr);
//Matice regulátoru
//const float poleKR[1][4] = {7.0444, 5.2084, -17.1787, -0.2969}; // 1 10
//const float poleKR[1][4] = {5.1962, 3.9470, -12.3567, -0.1745}; // 1 30
const float poleKR[1][4] = {3.5552, 2.7602, -8.2829, -0.0970}; // 1 100
const Matrix<1, 4> KR(poleKR);

```

```

//~~~~~[ VSTUPY ]~~~~~
const byte MerenyVystup = A0;
const byte MerenyVystup2 = A1;
//~~~~~[ VÝSTUPY ]~~~~~
const byte AkcniZasah = 9;
const byte AkcniZasah2 = 10;
//~~~~~[ PROMĚNNÉ ]~~~~~
const float w = 30;
float poleW1[3][1] = {{w}, {w}, {w}};
Matrix<3, 1> w1(poleW1);
Matrix<1, 1, float> Yest;
Matrix<1, 1, float> Ym;
int Ym1;
int Ym2;
Matrix<1, 1, float> u;
float poleXa[3][1] = {{0}, {0}, {0}};
Matrix<3, 1> Xa(poleXa); //Vektor stavů soustavy po odečtení w
float poleXb[1][1] = {{0}};
Matrix<1,1> Xb(poleXb); //Vektor pro uložení sumy odchylek
float poleXreg[4][1] = {{0}, {0}, {0}, {0}};
Matrix<4, 1> Xreg(poleXreg); //Stavu soustavy, pro regulátor
float poleXest_a[3][1] = {{1}, {2}, {3}};
Matrix<3, 1> Xest_a(poleXest_a); //Náhodný odhadovaný stav soustavy
float poleXest1_a[3][1] = {{0}, {0}, {0}};
Matrix<3, 1> Xest1_a(poleXest1_a); //Vektor násl. stavu, vyp. ze stavové rce
int32_t frekvencePWM = 900; //Nastavení frekvence PWM v jednotkách Hz
//~~~~~[ SETUP ]~~~~~
void setup(){
  Serial.begin(9600);
  Yest.Fill(0);
  Ym.Fill(0);
  Serial.println("u(k), Ym(k), Yest(k), x1(k), x2(k), x3(k), x4(k)");
  pinMode(MerenyVystup, INPUT);
  pinMode(MerenyVystup2, INPUT);
}

```

```

pinMode(AkcniZasah, OUTPUT);
pinMode(AkcniZasah2, OUTPUT);
pinMode(LED_BUILTIN, OUTPUT);
// Nastavení všech časovačů kromě časovače 0, aby se neovlivnila činnost časových funkcí
InitTimersSafe();
//Nastavení požadované PWM frekvence na příslušných pinech
bool uspesne = SetPinFrequencySafe(AkcniZasah, frekvencePWM);
bool uspesne2 = SetPinFrequencySafe(AkcniZasah2, frekvencePWM);
Serial.println(uspesne);
Serial.println(uspesne2);
delay(1000);
    if(uspesne&&uspesne2) {
        digitalWrite(LED_BUILTIN, HIGH);
    }
}
//~~~~~[ HLAVNÍ SMYČKA ]~~~~~
void loop(){
    //Naměření výstupu soustavy
    int PomYm1 = analogRead(MeremyVystup);
    int PomYm2 = analogRead(MeremyVystup2);
    Ym1 = (((PomYm1*5)/1023.0)/7.24)*100;
    Ym2 = (((PomYm2*5)/1023.0)/7.24)*100;
    Ym(0,0) = (Ym1 - Ym2);
    //Odhad stavů estimátorem
    Xest1_a = (M * Xest_a) + (N * u(0,0)) + (KE * (Ym - (C * Xest_a)));
    Xest_a = Xest1_a;
    Yest = C * Xest1_a;
    //Transformace stavů dle žádané hodnoty
    Xa = Xest_a - w1;
    Xb = Xb + (C * Xa);
    Xreg = VertCat(Xa,Xb);
    //Výpočet regulátoru
    u = KR * Xreg;
    delay(1000);
}

```

```

//Zápis na výstupy Arduina podle znaménka akčního zásahu
float pomVstupu = (255*u(0, 0)/50)*(100.0/99.0);
pomVstupu = constrain(pomVstupu, -255, 255);
  if (pomVstupu > 0){
    pwmWrite(AkcniZasah, pomVstupu);
    pwmWrite(AkcniZasah2, 0);
    Serial.print("kladny");
    Serial.print(pomVstupu);
  }
  else{
    pwmWrite(AkcniZasah, 0);
    pwmWrite(AkcniZasah2, -pomVstupu);
    Serial.print("zaporny");
    Serial.print(-pomVstupu);
  }
//Vykreslení akční veličiny, měřeného výstupu a estimovaných stavů
Serial.print(" ");
Serial.print(Ym(0, 0));
Serial.print(" ");
Serial.print(Yest(0, 0));
Serial.print(" ");
Serial.print(Xreg(0, 0));
Serial.print(" ");
Serial.print(Xreg(1, 0));
Serial.print(" ");
Serial.print(Xreg(2, 0));
Serial.print(" ");
Serial.println(Xreg(3, 0));
}

```

Příloha k diplomové práci

Řízení reálného procesu stavovým regulátorem LQ

Bc. Ondřej Zouhar

CD – ROM

Obsah

1. Text diplomové práce
2. Využité zdrojové kódy
3. Schémata z prostředí Simulink