

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Experimentální identifikace diskrétního stavového modelu

Josef Jelínek

Diplomová práce
2023

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2022/2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Josef Jelínek**
Osobní číslo: **I21306**
Studijní program: **N0714A150005 Automatické řízení**
Téma práce: **Experimentální identifikace diskrétního stavového modelu**
Zadávací katedra: **Katedra řízení procesů**

Zásady pro vypracování

Cíl: Vytvořte program v prostředí MATLAB, který realizuje automatické měření stavů elektrického systému připojeného pomocí Arduino Due. S využitím získaných průběhů napětí určete matice diskrétního stavového modelu, navrhnete LQ regulátor a ověřte jeho chování při regulaci systému.

Teoretická část:

- a) návrh LQ regulátoru se sledováním žádané bez trvalé regulační odchylky
- b) určení matic stavového modelu ze známého průběhu vstupů, stavů a výstupů
- c) MATLAB: možnosti komunikace přes sériovou linku

Praktická část:

- a) řešení sériové komunikace s Arduino Due v MATLABu
- b) vytvoření programu pro změření potřebných dat
- c) simulační ověření návrhu LQ regulátoru
- d) vytvoření programu v MATLABu pro regulaci systému na zadaný průběh žádané

Rozsah pracovní zprávy: **50 – 60 stran**
Rozsah grafických prací:
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

BALÁTEĚ, Jaroslav, 2004. *Automatické řízení. 2.*, přeprac. vyd. Praha: BEN –technická literatura. ISBN 80-730-0148-9.

OGATA, Katsuhiko. *Discrete-time control systems*. 2nd ed. Englewood Cliffs, N.J.: Prentice Hall, c1995, xi, 745 p. ISBN 01-303-4281-5.

Obsah dokumentace MATLAB. MathWorks. *MATLAB –mathematical computing software* [online]. 2022. Dostupné z: MATLAB Documentation (mathworks.com)

Vedoucí diplomové práce: **doc. Ing. František Dušek, CSc.**
Katedra řízení procesů

Datum zadání diplomové práce: **8. listopadu 2022**

Termín odevzdání diplomové práce: **19. května 2023**

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

L.S.

Ing. Daniel Honc, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 15. listopadu 2022

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

Bc. Josef Jelínek

Poděkování

Děkuji svému vedoucímu práce doc. Ing. Františku Duškovi, CSc. za odborné vedení této práce, cenné rady a připomínky, které přispěly k vytvoření této práce. Dále chci poděkovat své rodině a všem, kteří mě podporovali během studia.

V Pardubicích dne

Bc. Josef Jelínek

ANOTACE

Práce se věnuje experimentální identifikaci diskrétního stavového modelu. Na základě stavového modelu je navržen LQ regulátor se sledováním žádané bez trvalé regulační odchylky. Návrh a simulace jsou vytvořeny v MATLABu.

KLÍČOVÁ SLOVA

stavový model, identifikace parametrů, LQ regulátor, Arduino Due

TITLE

EXPERIMENTAL IDENTIFICATION OF A DISCRETE STATE MODEL

ANNOTATION

The thesis deals to experimental identification of a discrete state model. Based on state model is created LQ regulator with tracking problem without permanent regulatory deviation. Design and simulation are created in MATLAB.

KEYWORDS

state model, identification parametrs, LQ regulator, Arduino Due

OBSAH

SEZNAM ZKRATEK A ZNAČEK	9
SEZNAM SYMBOLŮ VELIČIN A FUNKCÍ.....	10
SEZNAM OBRÁZKŮ.....	12
SEZNAM TABULEK	13
ÚVOD.....	14
1 LQ regulátor.....	16
1.1 Návrh LQ regulátoru.....	16
1.1.1 Dosažitelnost a říditelnost	16
1.1.2 Formulace úlohy ve spojité oblasti.....	17
1.1.3 Modifikace pro nekonečný horizont řízení ve spojité oblasti	19
1.1.4 Formulace úlohy v diskrétní oblasti	20
1.1.5 Modifikace pro nekonečný horizont řízení v diskrétní oblasti.....	21
1.2 Sledování žádané hodnoty	22
1.2.1 Asymptotické sledování žádané	22
1.2.2 Rozšíření stavu o sumu regulačních odchylek	24
1.2.2.1 Beznárazové připnutí regulátoru	26
2 Identifikace soustavy	28
2.1 Matematicko-fyzikální analýza.....	28
2.2 Experimentální identifikace	28
2.2.1 Určení matic stavového modelu z měřených stavů, vstupů a výstupů.....	29
3 Regulovaný systém a simulace	31
3.1 Popis použitého systému a ideální model	31
3.1.1 Ideálního model.....	31
3.2 Měřicí jednotka a komunikace	37
3.2.1 Arduino Due	38
3.2.2 Komunikace.....	39

3.2.3	MATLAB	39
3.3	Simulace identifikace diskrétního modelu	39
3.3.1	Porovnání modelu s redukováným stavem s původním modelem	40
3.4	Simulace regulace	42
3.4.1	Vliv váhových matic na regulační pochod	44
4	Implementace identifikace a regulace v prostředí MATLAB	47
4.1	Komunikace s měřicí jednotkou	47
4.2	Měření dat pro identifikaci a určení parametru modelu	47
4.3	Regulace	50
5	ZÁVĚR	54
	POUŽITÁ LITERATURA	55
	PŘÍLOHY	56

SEZNAM ZKRATEK A ZNAČEK

ASCII	kódová tabulka pro definici znaků
A/D	analogově/digitální
COM	sériový port
D/A	digitálně/analogový
LQ	lineární-kvadratický
LTI	lineárně časově invariantní
MIMO	system s více vstupy a výstupy
PC	osobní počítač
PWM	pulzní šířková modulace
RISC	procesory s redukovanou instrukční sadou
UART	sběrnice pro asynchronní sériový přenos dat
URO	uzavřený regulační obvod
USB	univerzální sériová sběrnice

SEZNAM SYMBOLŮ VELIČIN A FUNKCÍ

A	matice systému
A_a	redukovaná matice systému
A_r	rozšířená matice systému
A_u	ekvivalentní matice systému
B	matice řízení
B_a	redukovaná matice řízení
B_r	rozšířená matice řízení
b_u	vektor řízení
C	matice výstupu
C_a	redukovaná matice výstupu
C_r	rozšířená matice výstupu
c_u	výstupní vektor
C₁₋₄	kapacita, F
D	matice převodu
D_a	redukovaná matice převodu
E	vlastní čísla URO
e(k)	regulační odchylka
f	vektor kořenů URO
F	váhová matice
i_{A-Z}	smyčkové proudy, A
i₁₋₄	proudy jednotlivých větví obvodu, A
J	kvadratická účelová funkce
k	diskrétní čas
L	vektor zesílení zpětné vazby
L_r	vektor zesílení zpětné vazby rozšířených stavů
L_w	vektor zesílení sumy regulačních odchylek
l_w	konstanta zesílení
L_x	vektor zesílení stavů x
N	horizont řízení v diskrétní oblasti
P	řešení Riccatiho rovnice
Q	váhová matice

R	váhová matice
R_{1-4}, R_Z	odpor, Ω
S	ustálené řešení Riccatiho rovnice
s_{e0}	počáteční suma regulačních odchylek
$s_e(k)$	suma regulačních odchylek
t	čas
t_f	horizont řízení ve spojité oblasti
T_s	perioda vzorkování
t_0	počáteční čas
U	elektrické napětí, V
$\mathbf{u}(t), \mathbf{u}(k)$	akční zásah (regulační zákon)
\mathbf{u}_0	počáteční akční zásah
\mathbf{w}_0	počáteční žádaná hodnota
$\mathbf{w}(k)$	žádaná hodnota
$\mathbf{w}(\infty)$	žádaná hodnota v ustáleném stavu
\mathbf{X}_N	matice měřených stavů od 2. do N hodnoty
\mathbf{X}_{uN}	matice obsahující měřené stavy a vstupy od 1. do $N-1$ hodnoty
$\mathbf{x}_r(k)$	rozšířený vektor stavových veličin
$\mathbf{x}_0(t_0)$	počáteční stav
$\mathbf{x}(t), \mathbf{x}(k)$	vektor stavových veličin
$\mathbf{x}(k+1)$	vektor stavových veličin v následujícím kroku
$\mathbf{x}(\infty)$	vektor stavových veličin v ustáleném stavu
\mathbf{Y}_N	vektor měřených výstupních stavů od 1. do $N-1$ hodnoty
$\mathbf{y}(t), \mathbf{y}(k)$	výstupní veličina
$\mathbf{y}(\infty)$	výstupní veličina v ustáleném stavu
\mathbf{Z}_{xu}	pomocná matice zesílení mezi \mathbf{x} a \mathbf{u}
\mathbf{Z}_{yu}	pomocná matice zesílení mezi \mathbf{y} a \mathbf{u}
Θ_{AB}	pomocná matice obsahující matici A a B
Θ_{CD}	pomocná matice obsahující matici C a D

SEZNAM OBRÁZKŮ

Obr. 2.1 Blokové schéma stavového popisu ve spojité oblasti.....	17
Obr. 2.2 Blokové schéma URO s LQ regulátorem	19
Obr. 2.3 Blokové schéma stavového popisu v diskrétní oblasti	20
Obr. 2.4 Blokové schéma LQ regulátoru s asymptotickým sledováním žádané	24
Obr. 2.5 Blokové schéma LQ regulátor se sumou regulačních odchylek.....	26
Obr. 4.1 Schéma regulovaného systému	31
Obr. 4.2 Porovnání průběhů výstupů	36
Obr. 4.3 Detail průběhů výstupů.....	37
Obr. 4.4 Arduino Due (Arduino Due, 2023).....	38
Obr. 4.6 Porovnání modelu s redukováným stavem s původním	41
Obr. 4.7 Rozdíl původních a redukováných stavů	41
Obr. 4.8 Simulace regulačního pochodu.....	44
Obr. 4.9 Vývojový diagram simulace regulace.....	44
Obr. 4.10 Průběh výstupní veličiny v závislosti na změně matice R.....	45
Obr. 4.11 Průběh akčního zásahu v závislosti na změně matice R.....	45
Obr. 4.12 Průběh výstupní veličiny v závislosti na změně matice Q.....	46
Obr. 4.13 Průběh akčního zásahu v závislosti na změně matice Q.....	46
Obr. 5.1 Vývojový diagram pro měření dat a identifikaci parametrů.....	49
Obr. 5.2 Rozdíl vypočítaných stavů a stavů naměřených.....	50
Obr. 5.3 Simulace regulačního pochodu.....	52
Obr. 5.4 Regulační pochod	52
Obr. 5.5 Vývojový diagram regulace.....	53

SEZNAM TABULEK

Tab. 1 Parametry použitých součástek.....	31
---	----

ÚVOD

Stavové regulátory využívají vnitřní popis soustavy, díky tomu se uplatní i při řízení složitých soustav. V praxi se využívají především PID regulátory, které jsou oproti stavovým jednodušší, mají výhodu v poměrně snadném nastavení parametrů, které nemusí být nastaveny úplně přesně a přesto mají uspokojující regulační pochody. LQ regulátor patří mezi optimální regulátory, které se snaží dosáhnout cíle regulace, ale ne za každou cenu. Tzn. ne za cenu libovolně velkých akčních zakazů (snaží se minimalizovat velikost akčních zásahů). Jeho cílem je převést systém z počátečního nenulového stavu do stavu koncového nulového. LQ regulátor je stavový regulátor, který využívá vnitřní popis soustavy. Nevýhodou použití je nutná znalost vnitřního popisu systému. Výhodou stavových regulátorů je použití k řízení složitějších soustav např. soustava s více vstupy a výstupy.

Metod jak získat vnitřní popis soustavy je několik. Problém může nastat, když matematický lineární model nepopisuje soustavu s dostatečnou shodou s realitou. Další problém může nastat, pokud stavy nejsou přímo měřitelné – neznalost aktuálních hodnot stavových veličin. Získání matematického modelu se nazývá identifikace, kterou se zabývá tato práce.

1 Cíl práce a její členění

Cílem DP je identifikace parametrů stavového modelu s následným návrhem LQ regulátoru a ověřením jeho funkčnosti. Jako sledovaná soustava bude použit pasivní elektrický systém IV. řádu se známým matematickým popisem ideálního chování. Soustava bude připojena k vývojové desce Arduino Due s programem, který zajistí nastavení a měření napětí podle požadavků přenášených sériovou komunikací. V prostředí MATLAB bude vytvořen program komunikující s vývojovou deskou, který zajistí experimentální data pro identifikaci stavového modelu. Dále bude vytvořen v prostředí MATLAB program pro provedení regulačního experimentu s LQ regulátorem využívajícím identifikovaný model.

Práce je rozdělena na čtyři hlavní kapitoly. V první kapitole je popsán návrh LQ regulátoru. Základním předpokladem pro návrh LQ regulátoru je znalost lineárního stavového modelu. Úloha návrhu LQ regulátoru je nejprve formulována obecně jak ve spojitě, tak v diskrétní oblasti pro konečný horizont řízení a následně modifikována pro nekonečný horizont řízení. Pro praktické použití je regulátor rozšířen o sledování žádané hodnoty a je popsáno beznárazové připnutí regulátoru.

Kvalita regulace je zásadním způsobem závislá na věrohodnosti modelu použitého při návrhu, tj. jak moc se liší chování modelu od chování řízené soustavy. Získáním stavového modelu konkrétní soustavy, tj. identifikací se zabývá druhá kapitola. Je zde vysvětleno získání modelu pomocí matematicko-fyzikální analýzy a experimentální identifikace. Protože na sledované soustavě jsou měřitelné jenom tři stavy ze čtyř, bude z experimentálních dat určen stavový model pouze III. Pro výpočet je použita metoda nejmenších čtverců. To obsahuje druhá kapitola.

Ve třetí kapitole bude popsán použitý systém, tj. soustava čtyř RC článků se zatěžovacím odporem, dle zapojení a hodnot součástek byl sestaven matematický popis ideálního chování. Poté se kapitola bude věnovat vybrané měřící jednotce, komunikaci Arduina přes sériovou linku s počítačem a vývojovému prostředí MATLAB.

V experimentální části bude provedena identifikace korekce parametrů spojitého ideálního modelu, identifikace diskrétního stavového modelu a vyhodnoceny odchylky modelů od změřených průběhů. Pro identifikovaný model se vytvoří implementace LQ regulátoru s integrací regulační odchylky a beznárazovým připnutím a implementace se ověří na simulaci regulace. Regulaci ovlivňují váhové matice a jejich vliv zde bude ukázán na simulacích.

Posledním krokem této práce je ověření funkčnosti navrženého LQ regulátoru při řízení sledované soustavy při sledování průběhu žádané hodnoty.

2 LQ regulátor

LQ regulátor (L-linear system, Q-quadratic cost) je stavový regulátor (zpětná vazba od stavu) vycházející z lineárního stavového popisu řízeného systému a kvadratického kritéria. Úloha návrhu LQ regulátoru je formulována jako optimalizační— minimalizace kritéria při dodržení podmínky popsané stavovým modelem.

Cílem optimální regulace je určení akčního zásahu $\mathbf{u}(t)$ LQ regulátoru tak, aby byla hodnota kritéria co nejmenší (Pivoňka, 2005). Výslednou regulaci lze ovlivnit pomocí volitelných váhových matic, které jsou součástí kritéria. LQ regulátor na nekonečném horizontu převádí systém z počátečního nenulového stavu do stavu koncového nulového. Jedná se tedy o řízení na nulovou žádanou hodnotu. Pro sledování žádané hodnoty se musí regulátor modifikovat.

2.1 Návrh LQ regulátoru

V této kapitole je postupně rozebrán návrh LQ regulátoru, nejdříve ve spojité oblasti a poté v diskrétní oblasti. Dále jsou zde zmíněny podmínky dosažitelnosti a říditelnosti systému.

2.1.1 Dosažitelnost a říditelnost

Dosažitelnost a říditelnost jsou podstatné základní vlastnosti systému. Díky nim je možné docílit koncového stavu systému.

Systém je dosažitelný, pokud jsou všechny jeho stavy dosažitelné. Stav $\mathbf{x}(t)$ je dosažitelný, pokud existuje akční zásah $\mathbf{u}(t)$, kterým je možné systém převést z počátečního stavu $\mathbf{x}_0(t_0)=0$ do požadovaného koncového stavu $\mathbf{x}(t) \neq 0$ na konečném časovém intervalu.

Systém je úplně říditelný v čase, když jsou všechny stavy říditelné. Stav $\mathbf{x}(t_1) \neq 0$ je říditelný, pokud existuje akční zásah $\mathbf{u}(t)$, kterým se systém převede ze stavu $\mathbf{x}(t_1)$ do stavu $\mathbf{x}(t_2)=0$ na konečném časovém intervalu. Postačující podmínka pro splnění říditelnosti je rovnost řádu modelu a hodnoty matice říditelnosti (Balátě, 2004).

Dosažitelnost je silnější vlastnost než říditelnost, protože systém může přejít do nulového stavu i bez řízení, a přitom nemusí být dosažitelný.

2.1.2 Formulace úlohy ve spojité oblasti

1) Stavový model

Pro návrh LQ regulátoru je nutností znát lineární stavový model systému. Ve spojitě verzi vypadá následovně (stavový MIMO model)

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad (1)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \quad (2)$$

kde \mathbf{A} – matice systému,

\mathbf{B} – matice řízení,

\mathbf{C} – matice výstupu,

\mathbf{D} – matice převodu

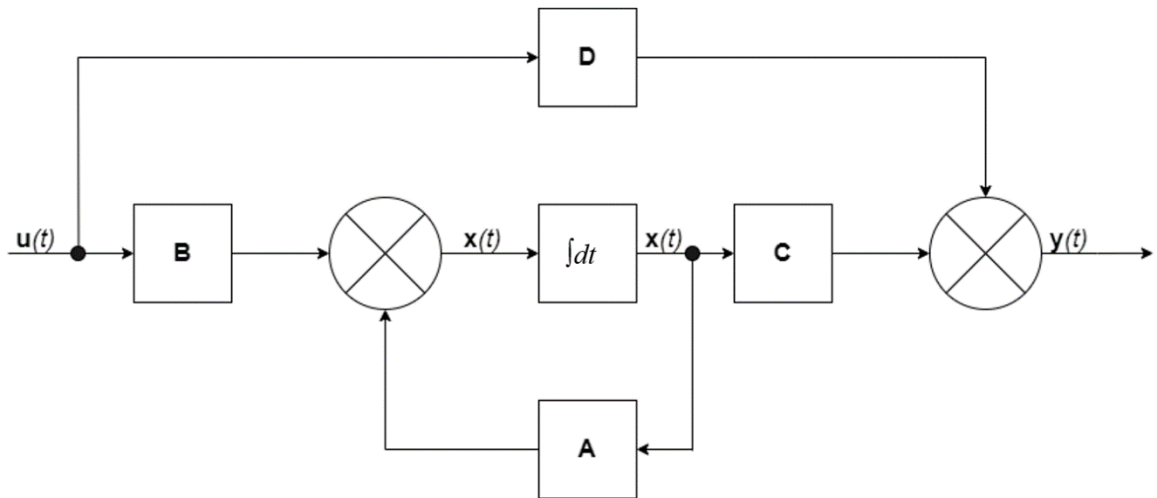
$\mathbf{x}(t)$ – vektor stavových veličin,

$\mathbf{u}(t)$ – vektor vstupních veličin,

$\mathbf{y}(t)$ – vektor výstupních veličin,

t – čas.

Stavový model lze nakreslit jako blokové schéma:



Obr. 2.1 Blokové schéma stavového popisu ve spojitě oblasti

2) Kvadratická účelová funkce na konečném horizontu řízení

Formulace kvadratické účelové funkce je dalším krokem při návrhu LQ regulátoru. Cíl řízení je dán kvadratickou účelovou funkcí (3). V kritériu jsou volitelné váhové matice. Matice \mathbf{F} a \mathbf{Q} se volí jako pozitivně semidefinitní a matice \mathbf{R} musí být vždy pozitivně definitní (Balátě, 2004).

Člen obsahující matici \mathbf{R} ovlivňuje dobu regulace. Nižší hodnoty matice \mathbf{R} dovolují větší akční zásahy. Změnou hodnoty matice \mathbf{Q} se mění rychlost ustálení regulované veličiny.

Tvar kvadratické účelové funkce na konečném horizontu řízení, tj. na konečném časovém intervalu $t_0 \leq t \leq t_f$ (Strejc, 1978)

$$J(u) = \mathbf{x}^T(t_f)\mathbf{F}\mathbf{x}(t_f) + \int_{t_0}^{t_f} [\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)dt], \quad (3)$$

kde \mathbf{F} – pozitivně semidefinitní váhová matice,

\mathbf{Q} – pozitivně semidefinitní váhová matice,

\mathbf{R} – pozitivně definitní váhová matice,

t_f – horizont řízení.

3) Riccatiho rovnice

Řešením minimalizace účelové funkce je tzv. diferenciální Riccatiho rovnice (diferenciální maticová rovnice). Pro konečný horizont řízení ve tvaru (Strejc, 1978)

$$-\frac{d\mathbf{P}}{dt} = \mathbf{A}^T\mathbf{P}(t) + \mathbf{P}(t)\mathbf{A} - \mathbf{P}(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}(t) + \mathbf{Q}. \quad (4)$$

Podmínka

$$\mathbf{P}(t_f) = \mathbf{F}. \quad (5)$$

4) Regulační zákon s proměnným zesílením

Pro regulační zákon platí

$$\mathbf{u}(t) = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}(t)\mathbf{x}(t), \quad (6)$$

zavede se vektor zesílení $\mathbf{L}(t)$ pro zjednodušení

$$\mathbf{L}(t) = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}(t), \quad (7)$$

regulační zákon s proměnným zesílením (akční zásah) bude ve tvaru

$$\mathbf{u}(t) = -\mathbf{L}(t)\mathbf{x}(t), \quad (8)$$

kde \mathbf{L} – zesílení zpětné vazby.

2.1.3 Modifikace pro nekonečný horizont řízení ve spojitě oblasti

LQ regulátor ve spojitě oblasti na nekonečném intervalu řízení je prakticky dobře použitelný, oproti regulátoru na konečném intervalu řízení, který je prakticky nepoužitelný. Matice \mathbf{P} je v tomto případě konstantní, tedy i vektor zesílení regulátoru \mathbf{L} je konstantní (Štecha, 1999).

1) Řešení

Pro nekonečný horizont řízení má kvadratická účelová funkce tvar

$$J(u) = \int_{t_0}^{\infty} [\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}] dt. \quad (9)$$

2) Riccatiho rovnice

Pro nekonečný horizont řízení dojde ke změně Riccatiho rovnice na algebraickou

$$0 = \mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} + \mathbf{Q}. \quad (10)$$

A matice \mathbf{P} (a tedy i vektor zesílení LQ regulátoru) je konstantní. Pro výpočet zesílení LQ regulátoru v MATLABu lze použít funkci `lqr`, která vrací vektor zesílení \mathbf{L} , řešení Riccatiho rovnice \mathbf{P} a vlastní čísla uzavřeného regulačního obvodu (URO) \mathbf{E}

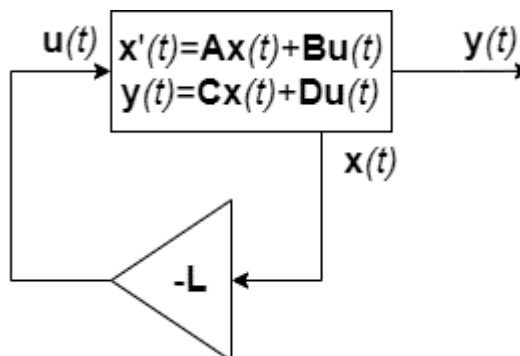
$$[\mathbf{L}, \mathbf{P}, \mathbf{E}] = \text{lqr}(\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R}, \mathbf{N}).$$

3) Regulační zákon s konstantním zesílením

Výsledný vztah pro regulační zákon je

$$\mathbf{u}(t) = -\mathbf{L} \mathbf{x}(t). \quad (11)$$

Struktura URO s LQ regulátorem se dá vyjádřit pomocí schématu na Obr. 2.2.



Obr. 2.2 Blokové schéma URO s LQ regulátorem

2.1.4 Formulace úlohy v diskrétní oblasti

1) Stavový model

Tvar stavového modelu se při přechodu ze spojité oblasti do diskrétní změní na

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \quad (12)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k), \quad (13)$$

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

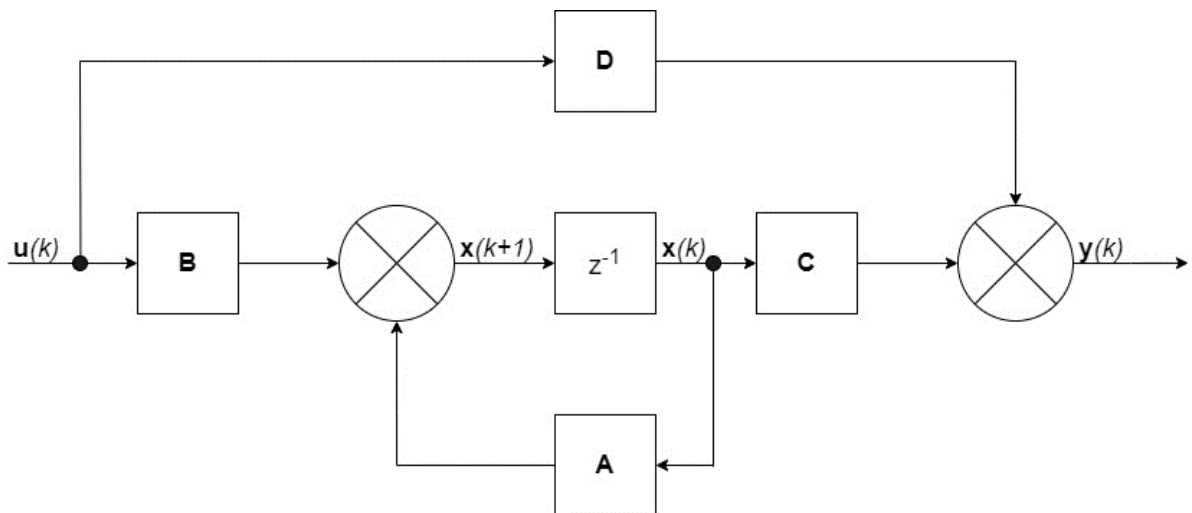
$\mathbf{u}(k)$ – vektor vstupních veličin,

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

$\mathbf{x}(k+1)$ – vektor stavových veličin v následujícím kroku,

k – diskrétní čas.

Stavový model v diskrétní oblasti lze nakreslit jako na Obr. 2.3.



Obr. 2.3 Blokové schéma stavového popisu v diskrétní oblasti

2) Kvadratická účelová funkce na konečném horizontu řízení

Základní tvar účelové funkce na konečném horizontu řízení

$$J = \mathbf{x}^T(N)\mathbf{F}\mathbf{x}(N) + \sum_{k=0}^{N-1} \mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}\mathbf{u}(k), \quad (14)$$

kde N – horizont řízení.

3) Riccatiho rovnice

Řešením minimalizace účelové funkce je diferenční Riccatiho rovnice (Ogatta,1995)

$$\mathbf{P}(k) = \mathbf{Q} + \mathbf{A}^T \mathbf{P}(k+1) \mathbf{A} - \mathbf{A}^T \mathbf{P}(k+1) \mathbf{B} (\mathbf{R} + \mathbf{B}^T \mathbf{P}(k+1) \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P}(k+1) \mathbf{A}. \quad (15)$$

4) Regulační zákon s proměnným zesílením

Pro výpočet regulačního zákona platí vztah (Ogatta,1995)

$$\mathbf{u}(k) = -(\mathbf{R} + \mathbf{B}^T \mathbf{P}(k+1) \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P}(k+1) \mathbf{A} \mathbf{x}(k), \quad (16)$$

vyjádří se vektor zesílení zpětné vazby $\mathbf{L}(k)$

$$\mathbf{L}(k) = (\mathbf{R} + \mathbf{B}^T \mathbf{P}(k+1) \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P}(k+1) \mathbf{A}. \quad (17)$$

Výsledný vztah pro regulační zákon je

$$\mathbf{u}(k) = -\mathbf{L}(k) \mathbf{x}(k). \quad (18)$$

2.1.5 Modifikace pro nekonečný horizont řízení v diskrétní oblasti

1) Kvadratická účelová funkce na nekonečném horizontu řízení

Pro nekonečný horizont řízení platí $N \rightarrow \infty$. Tím se zjednoduší rovnice (14), protože člen obsahující $\mathbf{x}(\infty)$ je roven nule. Výsledný tvar kritéria je

$$J = \sum_k^{\infty} \mathbf{x}^T(k) \mathbf{Q} \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{R} \mathbf{u}(k). \quad (19)$$

2) Riccatiho rovnice

Člen P je pro nekonečný horizont řízení neměnný, Riccatiho diferenční rovnice se změní na algebraickou. Řešení je

$$\mathbf{P} = \mathbf{Q} + \mathbf{A}^T \mathbf{P} \mathbf{A} - \mathbf{A}^T \mathbf{P} \mathbf{B} (\mathbf{R} + \mathbf{B}^T \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A}. \quad (20)$$

3) Regulační zákon s konstantním zesílením

Pro výpočet regulačního zákona platí

$$\mathbf{u}(k) = -(\mathbf{R} + \mathbf{B}^T \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A} \mathbf{x}(k), \quad (21)$$

zavede se vektor zesílení zpětné vazby \mathbf{L}

$$\mathbf{L} = (\mathbf{R} + \mathbf{B}^T \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A}. \quad (22)$$

Regulační zákon s konstantním zesílením je dán vztahem

$$\mathbf{u}(k) = -\mathbf{L}\mathbf{x}(k). \quad (23)$$

2.2 Sledování žádané hodnoty

LQ regulátor převádí systém z počátečního nenulového stavu do stavu nulového koncového. To znamená, že hodnota regulované veličiny je nulová. Pro praktické použití (regulace na žádanou hodnotu) se musí regulátor modifikovat. Modifikací LQ regulátoru je vytvořeno několik, v následující kapitole je popsáno asymptotické sledování žádané hodnoty. To je jedna z nejjednodušších modifikací.

2.2.1 Asymptotické sledování žádané

Asymptotické sledování žádané funguje na principu přičtení nějaké hodnoty k akčnímu zásahu tak, aby se v ustáleném stavu regulovaná veličina s žádanou hodnotou rovnaly. Tzn. přidaná hodnota je závislá na žádané hodnotě (člen \mathbf{L}_w je funkcí žádané hodnoty). Upravený regulační zákon pro asymptotické sledování žádané (Pivoňka, 2005)

$$\mathbf{u}(k) = -\mathbf{L}\mathbf{x}(k) + \mathbf{L}_w(k), \quad (24)$$

kde \mathbf{L}_w – člen pro dosažení žádané hodnoty.

Pro URO při dosažení žádané hodnoty v ustáleném stavu platí $y(\infty) = w$. Dojde k úpravě stavových rovnic

$$\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (25)$$

$$\mathbf{w} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}, \quad (26)$$

kde \mathbf{w} – žádaná hodnota.

Do obou stavových rovnic se dosadí za akční zásah \mathbf{u}

$$\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}(-\mathbf{L}\mathbf{x} + \mathbf{L}_w), \quad (27)$$

$$\mathbf{w} = \mathbf{C}\mathbf{x} + \mathbf{D}(-\mathbf{L}\mathbf{x} + \mathbf{L}_w). \quad (28)$$

Z rovnice (27) se vyjádří člen \mathbf{x}

$$\mathbf{x} = \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{L}\mathbf{x} + \mathbf{B}\mathbf{L}_w, \quad (29)$$

$$\mathbf{x} - (\mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{L}\mathbf{x}) = \mathbf{B}\mathbf{L}\mathbf{w}, \quad (30)$$

$$\mathbf{x}(\mathbf{I} - \mathbf{A} - \mathbf{B}\mathbf{L}) = \mathbf{B}\mathbf{L}\mathbf{w}, \quad (31)$$

$$\mathbf{x} = (\mathbf{I} - \mathbf{A} - \mathbf{B}\mathbf{L})^{-1}\mathbf{B}\mathbf{L}\mathbf{w}. \quad (32)$$

Druhá stavová rovnice (28) se upraví tak, aby se dalo dosadit za vyjádřený člen \mathbf{x}

$$\mathbf{w} = (\mathbf{C} - \mathbf{D}\mathbf{L})\mathbf{x} + \mathbf{D}\mathbf{L}\mathbf{w}, \quad (33)$$

$$\mathbf{w} = (\mathbf{C} - \mathbf{D}\mathbf{L})(\mathbf{I} - \mathbf{A} - \mathbf{B}\mathbf{L})^{-1}\mathbf{B}\mathbf{L}\mathbf{w} + \mathbf{D}\mathbf{L}\mathbf{w}. \quad (34)$$

Vytkne se $\mathbf{L}\mathbf{w}$

$$\mathbf{w} = [(\mathbf{C} - \mathbf{D}\mathbf{L})(\mathbf{I} - \mathbf{A} - \mathbf{B}\mathbf{L})^{-1}(\mathbf{B} + \mathbf{D})]\mathbf{L}\mathbf{w}, \quad (35)$$

pro zjednodušení se zavede pomocná konstanta l_w a vyjádří se $\mathbf{L}\mathbf{w}$

$$\mathbf{L}\mathbf{w} = [(\mathbf{C} - \mathbf{D}\mathbf{L})(\mathbf{I} - \mathbf{A} - \mathbf{B}\mathbf{L})^{-1}(\mathbf{B} + \mathbf{D})]^{-1}\mathbf{w}, \quad (36)$$

$$l_w = [(\mathbf{C} - \mathbf{D}\mathbf{L})(\mathbf{I} - \mathbf{A} - \mathbf{B}\mathbf{L})^{-1}(\mathbf{B} + \mathbf{D})], \quad (37)$$

kde l_w – konstanta zesílení.

Z toho vyplývá

$$\mathbf{L}\mathbf{w} = l_w\mathbf{w}, \quad (38)$$

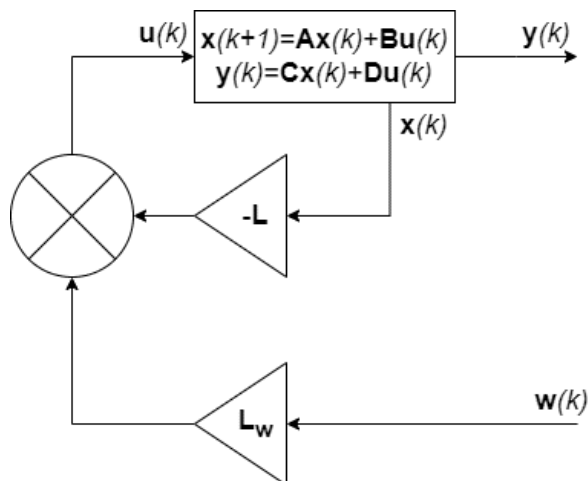
pro libovolné k platí

$$\mathbf{L}\mathbf{w}(k) = l_w\mathbf{w}(k). \quad (39)$$

Regulační zákon modifikovaný o asymptotické sledování má výsledný tvar

$$\mathbf{u}(k) = -\mathbf{L}\mathbf{x}(k) + l_w\mathbf{w}(k), \quad (40)$$

URO s modifikovaný LQ regulátor lze popsat blokovým schématem viz Obr. 2.4.



Obr. 2.4 Blokové schéma LQ regulátoru s asymptotickým sledováním žádané

2.2.2 Rozšíření stavu o sumu regulačních odchylek

LQ regulátor se sledováním žádané s rozšířením o sumu regulačních odchylek je jedna z dalších modifikací k dosažení žádané hodnoty. Pokud chování skutečného systému neodpovídá modelu, vzniká trvalá regulační odchylka. Regulační odchylku lze odstranit zavedením nové stavové proměnné – integrálu regulační odchylky, tzn. přidáním integračního charakteru LQ regulátoru (Machek, 2019). Pro výpočet regulační odchylky platí

$$\mathbf{e}(k) = \mathbf{w}(k) - \mathbf{y}(k), \quad (41)$$

kde $\mathbf{e}(k)$ – regulační odchylka.

Suma (integrace) regulační odchylky je popsána vztahem

$$\mathbf{s}_e(k+1) = \mathbf{s}_e(k) + \mathbf{e}(k), \quad (42)$$

$$\mathbf{s}_e(k+1) = \mathbf{s}_e(k) + \mathbf{w}(k) - \mathbf{y}(k), \quad (43)$$

kde $\mathbf{s}_e(k)$ – suma regulačních odchylek.

Dosažením za $\mathbf{y}(k)$ rovnice (43) dojde k úpravě

$$\mathbf{s}_e(k+1) = \mathbf{s}_e(k) + \mathbf{w}(k) - \mathbf{C}\mathbf{x}(k). \quad (44)$$

Doplněním vztahu pro integraci regulační odchylky k výchozího modelu vznikne rozšířený model. Soustava s rozšířeným stavem

$$\underbrace{\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{s}_e(k+1) \end{bmatrix}}_{\mathbf{x}_r(k+1)} = \underbrace{\begin{bmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & 1 \end{bmatrix}}_{\mathbf{A}_r} \underbrace{\begin{bmatrix} \mathbf{x}(k) \\ \mathbf{s}_e(k) \end{bmatrix}}_{\mathbf{x}_r(k)} + \underbrace{\begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}}_{\mathbf{B}_r} \mathbf{u}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{w}(k). \quad (45)$$

To je možné přepsat do tvaru

$$\mathbf{x}_r(k+1) = \mathbf{A}_r \mathbf{x}_r(k) + \mathbf{B}_r \mathbf{u}(k) + \mathbf{B}_w \mathbf{w}(k), \quad (46)$$

$$\mathbf{y}(k) = \mathbf{C}_r \mathbf{x}_r(k), \quad (47)$$

kde \mathbf{A}_r – rozšířená matice systému,

\mathbf{B}_r – rozšířená matice řízení,

\mathbf{C}_r – rozšířená matice výstupu,

$\mathbf{x}_r(k)$ – vektor stavových veličin.

Dojde k úpravě kvadratické účelové funkce

$$J = \sum_k^{\infty} \mathbf{x}_r^T(k) \mathbf{Q}_r \mathbf{x}_r(k) + \mathbf{u}^T(k) \mathbf{R} \mathbf{u}(k). \quad (48)$$

Řešením je získání vektoru zesílení \mathbf{L}_r takového, aby došlo k minimalizaci účelové funkce (48).

Výpočet zesílení \mathbf{L}_r je proveden v prostředí MATLAB pomocí funkce `dlqr`

$$[\mathbf{L}_r, \mathbf{S}, \mathbf{f}] = \text{dlqr}(\mathbf{A}_r, \mathbf{B}_r, \mathbf{Q}_r, \mathbf{R}),$$

kde \mathbf{L}_r – vektor zpětnovazebního zesílení,

\mathbf{S} – ustálené řešení Riccatiho rovnice,

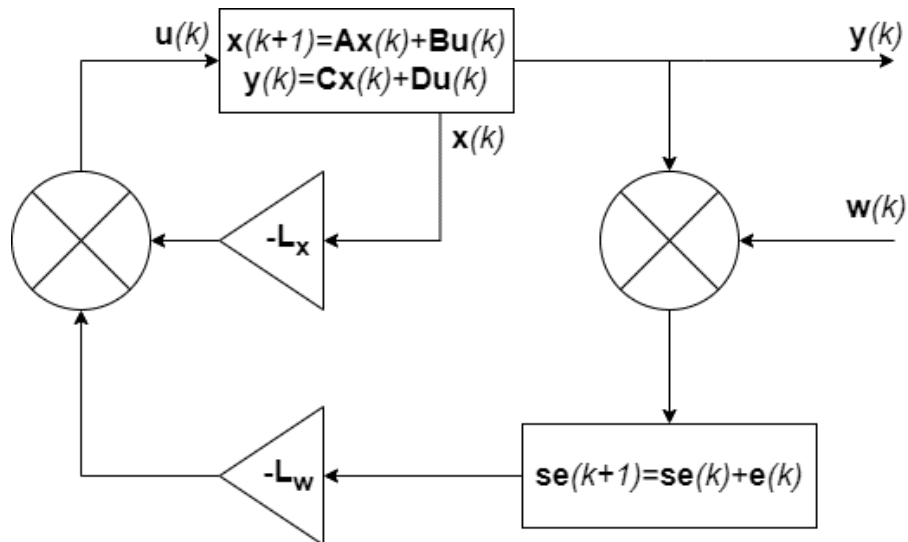
\mathbf{f} – vektor kořenů URO.

Doplněním rozšířeného stavu do regulačního zákona vznikne

$$\mathbf{u}(k) = -\mathbf{L}_r \mathbf{x}_r(k) = -[\mathbf{L}_x \quad \mathbf{L}_w] \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{s}_e(k) \end{bmatrix}, \quad (49)$$

$$\mathbf{u}(k) = -\mathbf{L}_x \mathbf{x}(k) - \mathbf{L}_w \mathbf{s}_e(k). \quad (50)$$

Pomocí zesílení \mathbf{L}_w lze ovlivnit vliv sumy regulačních odchylek na hodnotu řízení u . Vektory \mathbf{L}_x a \mathbf{L}_w se chovají jako váhové matice. URO s LQ regulátor se sledováním žádané s integrací regulační odchylky se dá zobrazit jako blokové schéma viz Obr. 2.5.



Obr. 2.5 Blokové schéma LQ regulátor se sumou regulačních odchylek

2.2.2.1 Beznárazové připnutí regulátoru

Beznárazové připnutí regulátoru znamená, že po připojení regulátoru nedojde ke změnám na vstupu soustavy, tj. hodnota akční je stejná jako byl vstup soustavy v okamžiku připojení regulátoru. V případě regulátorů s integrační složkou to znamená určit její počáteční hodnotu dle stavu v okamžiku připnutí regulátoru.

Pokud je URO v ustáleném stavu, je možné počáteční hodnotu stavu s_{e0} dopočítat z modelu. Musí se spočítat takový počáteční akční zásah \mathbf{u}_0 , který počítá s původní regulační odchylkou s_{e0} pro ustálený stav URO se zesílením $\mathbf{L}_r = [\mathbf{L}_x \quad \mathbf{L}_w]$ k žádané hodnotě $\mathbf{y}_0 = \mathbf{w}_0$. V neustáleném stavu musím znát konkrétní hodnoty stavu. Tvar výchozího modelu

$$\mathbf{x}_0 = \mathbf{A}\mathbf{x}_0 + \mathbf{B}\mathbf{u}_0, \quad (51)$$

$$\mathbf{w}_0 = \mathbf{C}\mathbf{x}_0. \quad (52)$$

Z rovnice (51) se vytkne x_0

$$\mathbf{x}_0 = \underbrace{(\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}}_{Z_{xu}} \mathbf{u}_0, \quad (53)$$

dosadí se do druhé rovnice (52) a zavede se pomocná konstanta Z_u

$$\mathbf{w}_0 = \underbrace{[\mathbf{C}Z_{xu}]}_{Z_{yu}} \mathbf{u}_0. \quad (54)$$

Vyjádří se akční zásah u_0 a upraví se vztah pro \mathbf{x}_0

$$\mathbf{u}_0 = Z_{yu}^{-1} \mathbf{w}_0 = -\mathbf{L}_x \mathbf{x}_0 - \mathbf{L}_w s_{e0}, \quad (55)$$

$$\mathbf{x}_0 = Z_{xu}Z_{yu}^{-1}\mathbf{w}_0, \quad (56)$$

dosazením za \mathbf{x}_0 do rovnice pro akční zásah dostaneme

$$Z_{yu}^{-1}\mathbf{w}_0 = -\mathbf{L}_x Z_{xu} Z_{yu}^{-1}\mathbf{w}_0 - \mathbf{L}_w \mathbf{s}_{e0}. \quad (57)$$

Výsledný vztah pro sumu regulačních odchylek je získán osamostatněním s_{e0}

$$\mathbf{s}_{e0} = -\mathbf{L}_w^{-1}[\mathbf{I} + \mathbf{L}_x Z_{xu}]Z_{yu}^{-1}\mathbf{w}_0. \quad (58)$$

Pokud není URO v ustáleném stavu je potřeba vyjít ze známého (měřeného) aktuálního stavu řízeného systému \mathbf{x}_0 a aktuální hodnoty vstupu u_0 . Počáteční hodnota sumy regulačních odchylek s_e se pak vypočte z regulačního zákona

$$u_0 = -\mathbf{L}_x \mathbf{x}_0 - L_w s_{e0}, \quad (59)$$

$$s_{e0} = -\frac{u_0 + \mathbf{L}_x \mathbf{x}_0}{L_w}. \quad (60)$$

3 Identifikace soustavy

Pro práci s neznámou soustavou je potřebné určit model soustavy, pokud nejsou stavové veličiny měřené. Proměnné vytvořeného modelu by měly svými vlastnostmi odpovídat veličinám reálné soustavy, nicméně při tvorbě dochází ke zjednodušení a model není úplně totožný s reálnou soustavou. Po vytvoření modelu je možné zkoumat chování soustavy na něm, místo experimentování s reálnou soustavou. Proces získání modelu se nazývá identifikace soustavy (Balátě, 2004). Model je možné získat analyticky (matematicko-fyzikální analýza) nebo experimentálně. Tyto metody jsou více popsány v dalších podkapitolách.

3.1 Matematicko-fyzikální analýza

Analytické získání matematického modelu vychází z elementárních jevů, které v reálné soustavě probíhají. Zároveň jsou zahrnuty konstrukční údaje soustavy. Analýzou se získá soustava diferenciálních a algebraických rovnic vycházejících z bilančních rovnic na základě zákonů o zachování hmoty a energie. Tyto rovnice obsahují vstupní, výstupní, ale i stavové veličiny. Pokud je analytický popis úplný, lze jednotlivé rovnice upravit do tvaru stavových rovnic (Strejc, 1978).

Analyticky získaný matematický model lze nazývat jako vnitřní popis systému. Získaný model může být velmi podobný reálné soustavě, ale zároveň jeho popis zbytečně složitý. V porovnání s experimentální identifikací je odvození matematicko-fyzikálního modelu náročnější (Vrožina, 2012).

3.2 Experimentální identifikace

Předpokladem pro experimentální identifikaci je, že vstupy a výstupy jsou měřitelné na zvoleném časovém úseku a vstupní signál je dostatečně vybudující. Matematický model získaný tímto způsobem popisuje nejlepší shodu odezvy reálné soustavy a odezvy modelu na konkrétní vstupní signál. Při tvorbě matematického modelu se musí určit nejdřív jeho struktura. Pro spojitý dynamický model je model popsán diferenciální rovnicí, pro diskrétní je to diferenční rovnice. Poté se určují koeficienty v dané rovnici. Cílem je najít takové parametry modelu, které minimalizují hodnotu účelové funkce. Získaný model lze nazvat jako vnější popis chování zkoumané soustavy. Tato metoda je také známá jako metoda černé skříňky (Balátě, 2004).

Experimentální identifikace se dá rozdělit na mnoho metod a další nové metody jsou stále publikovány. S rozvojem výpočetní techniky se rozšiřují metody, které mají komplikovanější výpočet parametrů. Metody identifikace se můžou lišit volbou vstupního signálu, typem matematického modelu, způsobem zpracování naměřených dat a také odhadem parametrů modelu (Vrožina, 2012).

3.2.1 Určení matic stavového modelu z měřených stavů, vstupů a výstupů

1) Formulace úlohy

Výchozí model

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k), \quad (61)$$

$$y(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}u(k), \quad (62)$$

pro další použití se model zapíše maticově

$$\mathbf{x}(k+1) = \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ u(k) \end{bmatrix}, \quad (63)$$

$$y(k) = \begin{bmatrix} \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ u(k) \end{bmatrix}. \quad (64)$$

Zápis pro N měřených hodnot $\{u(k) \quad \mathbf{x}(k) \quad y(k)\} \quad k = 1 \dots N$

$$\underbrace{\begin{bmatrix} \mathbf{x}(2) & \dots & \mathbf{x}(k) & \dots & \mathbf{x}(N) \end{bmatrix}}_{\mathbf{X}_N} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix}}_{\Theta_{AB}} \underbrace{\begin{bmatrix} \mathbf{x}(1) & \dots & \mathbf{x}(k) & \dots & \mathbf{x}(N-1) \\ u(1) & \dots & u(k) & \dots & u(N-1) \end{bmatrix}}_{\mathbf{X}_{uN}}, \quad (65)$$

$$\underbrace{\begin{bmatrix} y(1) & \dots & y(k) & \dots & y(N-1) \end{bmatrix}}_{\mathbf{Y}_N} = \underbrace{\begin{bmatrix} \mathbf{C} & \mathbf{D} \end{bmatrix}}_{\Theta_{CD}} \underbrace{\begin{bmatrix} \mathbf{x}(1) & \dots & \mathbf{x}(k) & \dots & \mathbf{x}(N-1) \\ u(1) & \dots & u(k) & \dots & u(N-1) \end{bmatrix}}_{\mathbf{X}_{uN}}, \quad (66)$$

zavedením pomocných matic se zápis pro N hodnot zjednoduší na

$$\mathbf{X}_N = \Theta_{AB}\mathbf{X}_{uN}, \quad (67)$$

$$\mathbf{Y}_N = \Theta_{CD}\mathbf{X}_{uN}, \quad (68)$$

kde \mathbf{X}_N – matice měřených stavů od 2. do N hodnoty,

\mathbf{Y}_N – vektor měřených výstupních stavů od 1. do $N-1$ hodnoty,

Θ_{AB} – pomocná matice obsahující matici \mathbf{A} a \mathbf{B} ,

Θ_{CD} – pomocná matice obsahující matici \mathbf{C} a \mathbf{D} ,

\mathbf{X}_{uN} – matice obsahující měřené stavy a vstupy od 1. do $N-1$ hodnoty.

Účelová funkce (suma kvadrátů rozdílů mezi měřenými a z modelu vypočtenými hodnotami stavů a výstupů – metoda nejmenších čtverců) pro tento model je ve tvaru

$$J(\Theta_{AB}) = (\mathbf{X}_N - \Theta_{AB}\mathbf{X}_{uN})(\mathbf{X}_N - \Theta_{AB}\mathbf{X}_{uN})^T, \quad (69)$$

$$J(\Theta_{CD}) = (\mathbf{Y}_N - \Theta_{CD}\mathbf{X}_{uN})(\mathbf{Y}_N - \Theta_{CD}\mathbf{X}_{uN})^T. \quad (70)$$

2) Řešení

Řešení minimalizace účelové funkce $J(\Theta_{AB})$ a $J(\Theta_{CD})$ je možné např. doplněním na úplný čtverec. Pro obecné řešení $J(\Theta)$ platí

$$\begin{aligned} J(\Theta) &= (\mathbf{G} - \Theta\mathbf{H})(\mathbf{G} - \Theta\mathbf{H})^T = (\mathbf{G} - \Theta\mathbf{H})(\mathbf{G}^T - \Theta^T\mathbf{H}^T) = \\ &= \underbrace{\mathbf{G}\mathbf{G}^T}_{\mathbf{c}} - \underbrace{\Theta\mathbf{H}\mathbf{G}^T}_{\mathbf{m}^T} - \underbrace{\mathbf{G}\mathbf{H}^T\Theta^T}_{\mathbf{m}} + \underbrace{\Theta\mathbf{H}\mathbf{H}^T\Theta^T}_{\mathbf{M}} = \Theta\mathbf{M}\Theta^T - \mathbf{m}^T\Theta - \mathbf{m}\Theta^T + \mathbf{c}, \end{aligned} \quad (71)$$

$$J(\Theta) = (\Theta - \mathbf{m}\mathbf{M}^{-1})\mathbf{M}(\Theta - \mathbf{m}\mathbf{M}^{-1})^T + \mathbf{c} - \mathbf{m}(\mathbf{M}^{-1})^T\mathbf{m}^T. \quad (72)$$

Po dosazení za pomocné konstanty se vyjádří výsledné rovnice Θ_{AB} a Θ_{CD} . Pro tyto výsledné parametry nabývá účelová funkce minimální hodnoty. Podmínkou je, že matice \mathbf{M} musí být pozitivně definitní. Výsledné vztahy

$$\mathbf{m} = \mathbf{G}\mathbf{H}^T = \mathbf{X}_N\mathbf{X}_{uN}^T \quad \text{nebo} \quad \mathbf{Y}_N\mathbf{X}_{uN}^T, \quad (73)$$

$$\mathbf{M} = \mathbf{H}\mathbf{H}^T = \mathbf{X}_{uN}\mathbf{X}_{uN}^T, \quad (74)$$

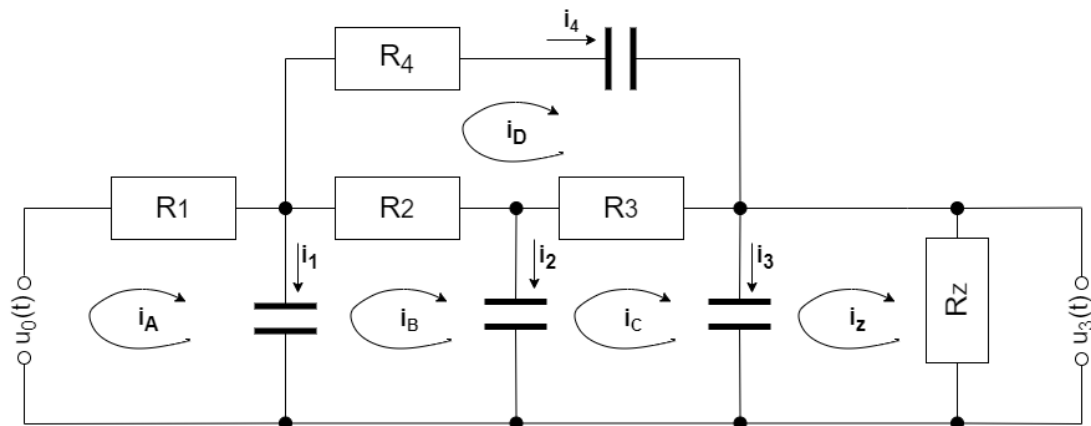
$$\Theta_{AB}^T = [\mathbf{X}_{uN}\mathbf{X}_{uN}^T]^{-1}\mathbf{X}_{uN}\mathbf{X}_N^T, \quad (75)$$

$$\Theta_{CD}^T = [\mathbf{X}_{uN}\mathbf{X}_{uN}^T]^{-1}\mathbf{X}_{uN}\mathbf{Y}_N^T. \quad (76)$$

4 Regulovaný systém a simulace

4.1 Popis použitého systému a ideální model

Regulovaný systém je tvořen zapojením čtyř RC článků na nepájivém poli se zatěžovacím odporem R_z . Schéma zapojení RC článků je na Obr. 4.1. Generování vstupního napětí u_0 zajišťuje vývojová deska Arduino Due, která slouží i k měření napětí. Arduino Due je připojené pomocí USB rozhraní k počítači, v kterém je řídicí program pro odesílání a přijímání požadavků z Arduina.



Obr. 4.1 Schéma regulovaného systému

Tab. 1 Parametry použitých součástek

R1 [kΩ]	R2 [kΩ]	R3 [kΩ]	R4 [Ω]	R _z [kΩ]	C ₁ [μF]	C ₂ [μF]	C ₃ [μF]	C ₄ [μF]
1,6	1,6	1,6	51	51	47	820	47	47

4.1.1 Ideální model

Pro potřeby simulace byl sestaven matematický model řízené soustavy umožňující nahradit měřená data výpočtem z modelu. K odvození matematického modelu byla využita matematicko-fyzikální analýza využívající metodu smyčkových proudů, které jsou znázorněny ve schématu Obr. 4.1. Použitím obecných vztahů (77) a metody smyčkových proudů dostaneme soustavu diferenciálních rovnic (78)

$$u = \frac{1}{C} \int_0^t i dt, \quad i = C \frac{du}{dt}, \quad (77)$$

$$u_0 = R_1 i_A + \frac{1}{C_1} \int_0^t (i_A - i_B) dt, \quad (78)$$

$$0 = R_2(i_B - i_D) + \frac{1}{C_2} \int_0^t (i_B - i_C) dt + \frac{1}{C_1} \int_0^t (i_B - i_A) dt,$$

$$0 = R_3(i_C - i_D) + \frac{1}{C_3} \int_0^t (i_C - i_Z) dt + \frac{1}{C_2} \int_0^t (i_C - i_B) dt,$$

$$0 = R_4 i_D + \frac{1}{C_4} \int_0^t i_D dt + R_2(i_D - i_B) + R_3(i_D - i_C),$$

$$0 = R_Z i_Z + \frac{1}{C_3} \int_0^t (i_Z - i_C) dt,$$

kde u_0 – napájecí napětí, V,

R_{1-4}, R_Z – odpor, Ω ,

C_{1-4} – kapacita, F,

i_{A-Z} – smyčkové proudy, A.

Následně se vyjádří smyčkové proudy i_{A-Z}

$$i_1 = i_A - i_B \rightarrow i_A = i_1 + i_2 + i_3 + i_Z,$$

$$i_2 = i_B - i_C \rightarrow i_B = i_2 + i_3 + i_Z,$$

$$i_3 = i_C - i_Z \rightarrow i_C = i_3 + i_Z,$$

$$i_4 = i_D \rightarrow i_D = i_4,$$

$$i_Z = \frac{1}{R_Z} u_3,$$

kde u_3 – výstupní napětí, V,

i_{1-4} – smyčkové proudy, A.

Rovnice (78) se dají zjednodušit dosazením za smyčkové proudy a formulací napětí pomocí obecných vztahů

$$u_0 = R_1(i_1 + i_2 + i_3 + i_Z) + u_1,$$

$$0 = R_2(i_2 + i_3 + i_Z - i_4) + u_2 - u_1,$$

$$0 = R_3(i_3 + i_Z - i_4) + u_3 - u_2,$$

$$0 = R_4 i_4 + u_4 + R_2(i_4 - i_2 - i_3 - i_Z) + R_3(i_4 - i_3 - i_Z).$$

(79)

(80)

Následným dosazením proudu i_z a převedením napětí na jednu stranu a proudů na druhou dostaneme

$$\begin{aligned}
 u_0 - u_1 - \frac{R_1}{R_Z} u_3 &= R_1 i_1 + R_1 i_2 + R_1 i_3, \\
 u_1 - u_2 - \frac{R_2}{R_Z} u_3 &= R_2 i_2 + R_2 i_3 - R_2 i_4, \\
 u_2 - u_3 - \frac{R_3}{R_Z} u_3 &= R_3 i_3 - R_3 i_4, \\
 u_4 - \frac{R_2 + R_3}{R_Z} u_3 &= R_4 i_4 + R_2 i_4 - R_2 i_2 - R_2 i_3 + R_3 i_4 - R_3 i_3.
 \end{aligned} \tag{81}$$

Zápisem rovnic (79)-(81) v maticovém tvaru a vyloučením smyčkových proudů i_{A-D} je možné rovnice upravit na maticové vyjádření (82) popisující časovou závislost proudů i_{1-4} , i_z a napětí u_{1-4} na průběhu vstupního napětí $u_0(t)$. Matice je možné použít v MATLABu a zjednodušují další úpravy. Výsledné maticové výrazy

$$\mathbf{Ii} = \mathbf{Uu} + \mathbf{m}u_0, \quad \mathbf{K} \frac{d\mathbf{u}}{dt} = \mathbf{i}, \tag{82}$$

kde

$$\begin{aligned}
 \mathbf{i} &= \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{bmatrix}, & \mathbf{u} &= \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}, & \mathbf{m} &= \begin{bmatrix} 1 \\ R_1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \\
 \mathbf{K} &= \begin{bmatrix} C_1 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 \\ 0 & 0 & C_3 & 0 \\ 0 & 0 & 0 & C_4 \end{bmatrix}, & \mathbf{I} &= \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & R_2 & R_2 + R_3 & -R_2 - R_3 - R_4 \end{bmatrix}, \\
 \mathbf{U} &= \begin{bmatrix} -\frac{1}{R_1} & 0 & -\frac{1}{R_Z} & 0 \\ \frac{1}{R_2} & -\frac{1}{R_2} & -\frac{1}{R_Z} & 0 \\ 0 & \frac{1}{R_3} & -\frac{1}{R_3} - \frac{1}{R_Z} & 0 \\ 0 & 0 & -\frac{R_2 + R_3}{R_Z} & 1 \end{bmatrix},
 \end{aligned} \tag{83}$$

kde \mathbf{I} – matice odporů,

\mathbf{i} – vektor proudů i_{1-4} ,

\mathbf{U} – pomocná matice,
 \mathbf{u} – vektor napětí na kondenzátorech,
 \mathbf{m} – pomocný vektor,
 u_0 – napájecí napětí,
 \mathbf{K} – matice kapacit.

Z první rovnice (82) můžeme vyjádřit vektor proudů

$$\mathbf{i} = \mathbf{I}^{-1}(\mathbf{U}\mathbf{u} + \mathbf{m}u_0), \quad (84)$$

dosazením do druhé rovnice (82) dostaneme vztah

$$\mathbf{K} \frac{d\mathbf{u}}{dt} = \mathbf{I}^{-1}(\mathbf{U}\mathbf{u} + \mathbf{m}u_0). \quad (85)$$

Další úpravou je osamostatnění derivace vektoru napětí na levé straně rovnice. Rovnice přejde do tvaru

$$\frac{d\mathbf{u}}{dt} = \mathbf{K}^{-1}\mathbf{I}^{-1}(\mathbf{U}\mathbf{u} + \mathbf{m}u_0). \quad (86)$$

Roznásobením závorek získáme standartní stavový model (88) popisující závislost výstupního napětí u_3 na vstupním napětí u_0 prostřednictvím napětí na kondenzátorech u jako stavových proměnných

$$\frac{d\mathbf{u}}{dt} = \underbrace{\mathbf{K}^{-1}\mathbf{I}^{-1}\mathbf{U}}_{\mathbf{A}_u} \mathbf{u} + \underbrace{\mathbf{K}^{-1}\mathbf{I}^{-1}\mathbf{U}\mathbf{m}}_{\mathbf{b}_u} u_0, \quad (87)$$

$$\mathbf{u}_3 = \underbrace{[0 \quad 0 \quad 1 \quad 0]}_{\mathbf{c}_u} \mathbf{u},$$

výsledný tvar stavového modelu

$$\frac{d\mathbf{u}}{dt} = \mathbf{A}_u \mathbf{u} + \mathbf{b}_u u_0, \quad (88)$$

$$\mathbf{u}_3 = \mathbf{c}_u \mathbf{u},$$

kde \mathbf{A}_u – matice systému,

\mathbf{b}_u – vektor řízení,

\mathbf{c}_u – výstupní vektor.

Matice modelu se budou rovnat

$$\mathbf{A}_u = \mathbf{K}^{-1}\mathbf{I}^{-1}\mathbf{U}$$

$$\mathbf{A}_u = \begin{bmatrix} -\frac{1}{C_1}\left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_4}\right) & \frac{1}{C_1 R_2} & \frac{1}{C_1 R_4} & \frac{1}{C_1 R_4} \\ \frac{1}{C_2 R_2} & -\frac{1}{C_2}\left(\frac{1}{R_2} + \frac{1}{R_3}\right) & \frac{1}{C_2 R_3} & 0 \\ \frac{1}{C_3 R_4} & \frac{1}{C_3 R_3} & -\frac{1}{C_3}\left(\frac{1}{R_3} + \frac{1}{R_4} + \frac{1}{R_z}\right) & -\frac{1}{C_3 R_4} \\ \frac{1}{C_4 R_4} & 0 & -\frac{1}{C_4 R_4} & -\frac{1}{C_4 R_4} \end{bmatrix}, \quad (89)$$

$$\mathbf{b}_u = \mathbf{K}^{-1}\mathbf{I}^{-1}\mathbf{m} = \begin{bmatrix} \frac{1}{C_1 R_1} \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\mathbf{c}_u = [0 \quad 0 \quad 1 \quad 0], \quad \mathbf{D} = 0.$$

4.1.2 Korekce parametrů modelu

Pro zpřesnění matematického modelu je potřeba provést korekci jeho parametrů. Tzn. že se musí upravit hodnoty součástek, protože nominální hodnoty součástek se aspoň nepatrně liší od skutečných hodnot. K identifikaci parametrů musíme znát výstup ideálního modelu s nominálními hodnoty součástek a měřený výstup na reálné soustavě.

Nejdříve je potřeba změřit výstup na reálné soustavě. Soustava musí při měření vycházet z ustáleného stavu. Následně se z původních matic (použití nominálních hodnot) dopočítá pomocí příkazu `lsim` výstup ideálního modelu. Příkaz `lsim` obsahuje počáteční podmínky

$$\mathbf{x}_0 = \mathbf{A}^{-1}\mathbf{B}u_0, \quad (90)$$

kde \mathbf{A} – matice systému,

\mathbf{B} – matice řízení,

\mathbf{x}_0 – vektor počátečních podmínek,

u_0 – akční zásah.

Optimalizace má spočítat korekci devíti parametrů (hodnoty součástek $R_1, C_1, R_2, C_2, R_3, C_3, R_4, C_4, R_z$), které ovlivňují výsledné matice popisující soustavu. Z důvodu, že hodnoty odporů a kondenzátorů jsou řádově hodně odlišné, zavedl se korekční vektor. Korekční vektor

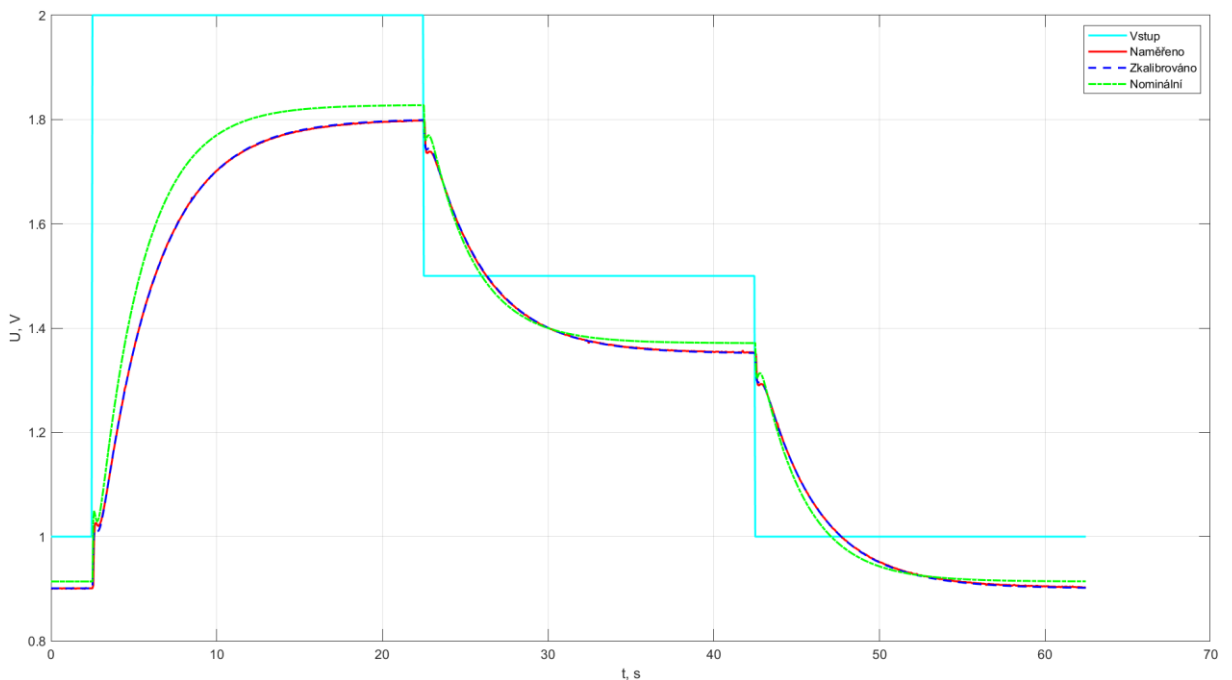
obsahuje devět prvků, kterými se násobí hodnoty součástek. Potom je možné provést optimalizaci, protože sjednotí řád parametrů pro korekci.

Optimalizace je provedena pomocí funkce `fminsearch`. Jako počáteční podmínka je zvolen korekční vektor plný jedniček (počítá s nominálními hodnoty součástek). V kritériu se počítá chyba mezi měřeným výstupem a výstupem ideálního modelu, pro výpočet je použita metoda nejmenších čtverců. Výsledné kritérium je formulováno jako

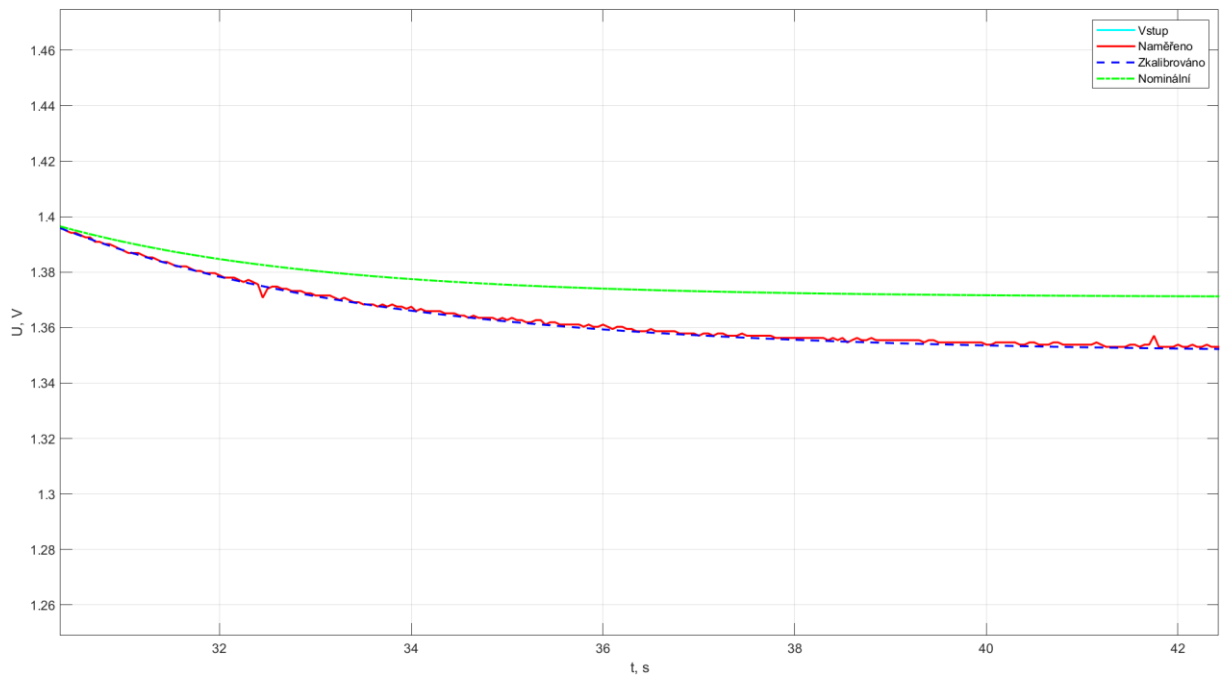
$$J = (\mathbf{y}_m - \mathbf{y})^T (\mathbf{y}_m - \mathbf{y}), \quad (91)$$

kde \mathbf{y}_m – měřený vektor výstupu,
 \mathbf{y} – výstup ideálního modelu.

Po skončení optimalizace je korekční vektor naplněn novými hodnotami. Tím dojde k úpravě hodnot součástek a výsledných matic soustavy. Poté stačí už dopočítat pouze výstup soustavy s korigovanými parametry. Porovnání průběhů měřeného výstupu, výstupu s nominálními hodnotami a výstupu po optimalizaci je na Obr. 4.2, na Obr. 4.3 je detail průběhů výstupů, kde je vidět velké zlepšení díky optimalizaci.



Obr. 4.2 Porovnání průběhů výstupů



Obr. 4.3 Detail průběhů výstupů

4.2 Měřící jednotka a komunikace

V této kapitole bude popsáno Arduino, které slouží jako měřící jednotka a její komunikace. Arduino je název pro jednodeskový počítač, které se můžou lišit procesorem, velikostí desky, velkou variací pinů a jejich funkcí, dalšími vlastnostmi. Bylo původně vyvinuto jako elektronická platforma pro studenty, která umožní jednoduché používání a rychlý vývoj projektů. Z toho důvodu je volně dostupná (licence open-source) jak hardwarová část, tak i vývojové prostředí Arduino IDE s programovacím jazykem Wiring. U hardwaru je volně dostupná technická dokumentace, vlastnosti DPS i použité součástky. U softwaru jsou k dispozici např. zdrojové kódy.

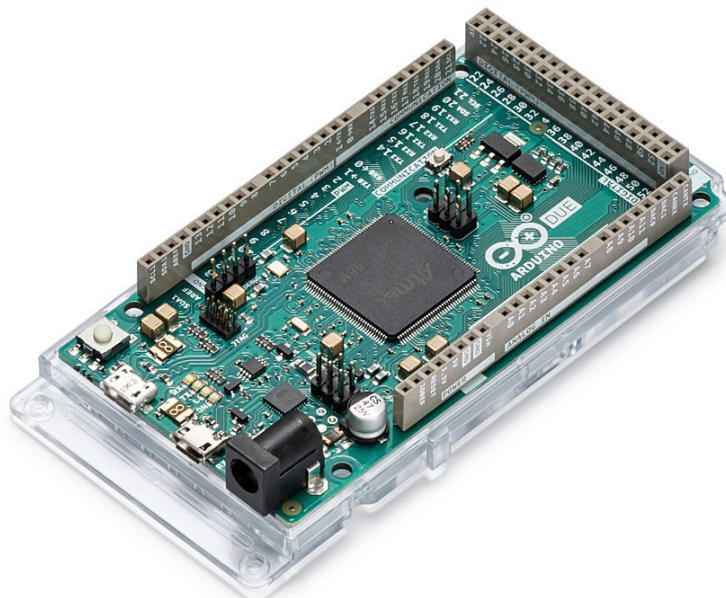
Výhodou Arduina je možnost rozšíření pomocí rozšiřujících modulů tzv. Arduino shields. Shildy se připojují přímo na piny desky. Často používané rozšíření jsou např. Wifi Shield, pomocí kterého je možné Arduino připojit k internetu pomocí bezdrátové sítě Wi-Fi nebo GPS Shield. Rozšířením o GPS modul lze zjistit aktuální polohu, rychlost nebo nadmořskou výšku.

4.2.1 Arduino Due

Arduino Due patří k jedné z výkonnějších verzí Arduina. Je to první deska založená na 32bitové architektuře, oproti předchozím verzím (8bitovým) dokáže pracovat s 4 B daty v rámci jednoho taktu. Je osazeno RISC procesorem ATMEL SAM3X8E ARM Cortex-3M s taktovací frekvencí 84 MHz. Procesor má flash paměť o velikost 512 kB pro nahrání zdrojového kódu a 96 kB SRAM paměť pro data (Arduino Due, 2023).

Arduino Due má 54 digitálních vstupních / výstupních pinů, 12 z nich použitelných jako PWM výstupy, dále má 12 analogových vstupů a 4 UART porty. Oproti ostatním deskám Arduino běží Arduino Due na 3,3 V (ostatní 5 V). 3,3 V je maximální napětí, které může na pinech být, jinak by mohlo dojít k poškození. Napájení může být externí 7–12 V (doporučené hodnoty) nebo přes USB port 5 V. USB porty jsou na desce dva, jeden slouží pro komunikaci s Arduinem a nahrávání programu, druhý USB port slouží pro připojení dalších zařízení (Arduino Due, 2023).

Pomocí A/D převodníků je možné na 12 pinech měřit spojitý signál a díky D/A převodníkům na 2 pinech generovat spojitý signál. D/A převodník je omezený hardwarem a může generovat napětí přibližně jenom 0,6–2,6 V. Převodníky jsou 12bitové, s výchozím rozlišením 10 bitů pro měření a 8 bitů pro generování napětí. Na Obr. 4.4 je deska Arduino Due (Arduino, 2023).



Obr. 4.4 Arduino Due (Arduino Due, 2023)

4.2.2 Komunikace

Komunikace Arduina a PC je v tomto případě typu master – slave. PC s Arduinem jsou propojeny přes USB sběrnici (softwarově je simulována komunikace přes sériový port). Řídící aplikaci obsahuje PC, který se chová jako master a odesílá požadavky přes sériové rozhraní do Arduina. Řídící aplikace je napsána v MATLABu. Má za úkol odesílat požadavky do Arduina na generování žádaného napěťového signálu, který slouží jako napájecí napětí RC článků. Následně má Arduino měřit napětí na kondenzátorech a hodnoty posílat zpět přes USB do PC. Komunikace probíhá podle daného ASCII protokolu. Zpráva je tvořena posloupností znaků.

4.2.3 MATLAB

MATLAB je interaktivní softwarové prostředí a vysoce výkonný skriptovací jazyk pro technické výpočty. Je vyvíjen společností MathWork za účelem integrovat výpočty, vizualizaci a programování do jednoduchého prostředí. Základním datovým typem je komplexní matice. MATLAB obsahuje spoustu zabudovaných funkcí, pomocí kterých se dají problémy řešit rychleji než v ostatních jazycích, pokud je jejich formulace vektorová nebo maticová (Dušek, 2000).

Nadstavbou MATLABu je SIMULINK, který slouží pro tvorbu dynamických soustav ve formě blokových schémat. S daty ze SIMULINKu lze pracovat zároveň i v MATLABu pro další analýzu.

Součástí MATLABu jsou knihovny funkcí neboli toolboxy. Jeden z více používaných v této práci je Control System Toolbox, který rozšiřuje MATLAB o funkce a vizualizační nástroje z teorie systémů a řídicí techniky. Z důvodu jednoduchosti bude MATLAB použitý pro identifikaci diskrétního modelu, návrh a simulaci LQ regulátoru s redukovaným stavem a beznárazovým připnutím. Následně pro komunikaci s Arduinem a implementaci identifikace a regulace na reálnou soustavu.

4.3 Identifikace diskrétního modelu

V této kapitole je popsán kód, který byl vytvořen za účelem identifikace diskrétního modelu v MATLABu. Odvozený matematický model sledované soustavy je 4. řádu, tj. obsahuje čtyři stavy odpovídající napětím na kondenzátorech. Pomocí Arduina lze měřit jenom hodnoty napětí proti zemi. Z toho důvodu se dají změřit jenom 3 napětí viz Obr. 4.1.

Napětí na kondenzátoru C_4 nelze změřit, a proto se musí provést identifikace stavového modelu 3. řádu na základě průběhu vstupního napětí a 3 měřených stavů.

Z naměřených průběhů vstupního napětí a napětí na kondenzátorech C_1 až C_3 se určí korekce parametrů ideálního modelu (81) a matice diskrétního stavového modelu 3. řádu v obecném tvaru (12)

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

$$y(k) = \mathbf{C}\mathbf{x}(k).$$

V MATLABu se vytvoří matice \mathbf{A} , \mathbf{B} , \mathbf{C} a vyplní se hodnotami podle výpočtů z (89). Pomocí funkce `ss` se z matic \mathbf{A} , \mathbf{B} , \mathbf{C} udělá spojitý stavový model. Následně se funkcí `c2d` převede ze spojitého na diskrétní stavový model. Pro výpočet stavů $\mathbf{x}(k)$ a hodnot $y(k)$ se použije funkce `lsim`, která má jako vstupy vypočítaný diskrétní stavový model, zvolený vektor vstupů, vektor času a počáteční podmínky.

4.3.1 Porovnání modelu s redukováným stavem s původním modelem

Z důvodu, že nelze měřit všechny stavy soustavy, bylo využito experimentální identifikace pro získání matic diskrétního stavového modelu z měřených stavů, vstupů a výstupů viz kapitola 3.2.1. V simulaci byly místo měřených dat použita data ideálního modelu (v kódu označená jako $\mathbf{X4}$ a y) získaná pomocí funkce `lsim`. Následující část kódu z MATLABu popisuje získání redukováných matic \mathbf{Aa} , \mathbf{Ba} , \mathbf{Ca} , \mathbf{Da} .

```
Xn = X4(2:end,1:3)'; %vytvoření matice Xn
Xr = X4(1:end-1,:); Xr(:,4)=u(1:end-1,:);
Xun = Xr'; %vytvoření matice Xun
thetaAB = (Xun*Xun')\ (Xun*Xn'); %výpočet thetaAB
thetaAB = thetaAB';
Aa = thetaAB(:,1:3); %určení redukované matice A
Ba = thetaAB(:,4); %určení redukované matice B

Yn = y'; Yn = Yn(:,1:end-1);
thetaCD = (Xun*Xun')\ (Xun*Yn');
thetaCD = thetaCD'; %výpočet thetaCD
Ca = thetaCD(1,1:end-1); %určení redukované matice C
Da = thetaCD(1,end); %určení redukované matice D
```

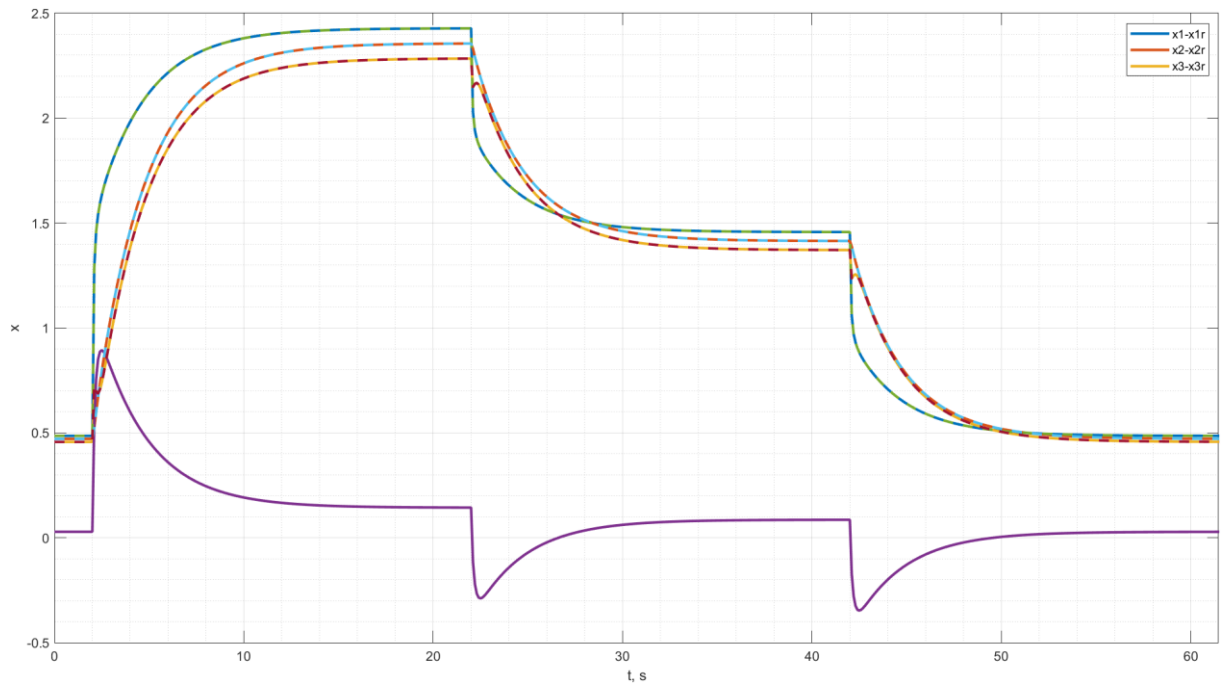
Pro ověření správnosti a možnost vykreslení byl vytvořen model a pomocí funkce `lsim` dopočítán výstup y_r a stavy $\mathbf{X3}$ stejně jako v předchozím případě. Na Obr. 4.6 je rozdíl původních a redukováných stavů. Nově vypočítané matice z experimentálních dat mají podle

porovnání stavů správné hodnoty. Jelikož osa y má hodnoty řádově 10^{-4} , jsou rozdíly stavů minimální, takže jsou zanedbatelné. Největší rozdíly jsou na vysokých frekvencích.

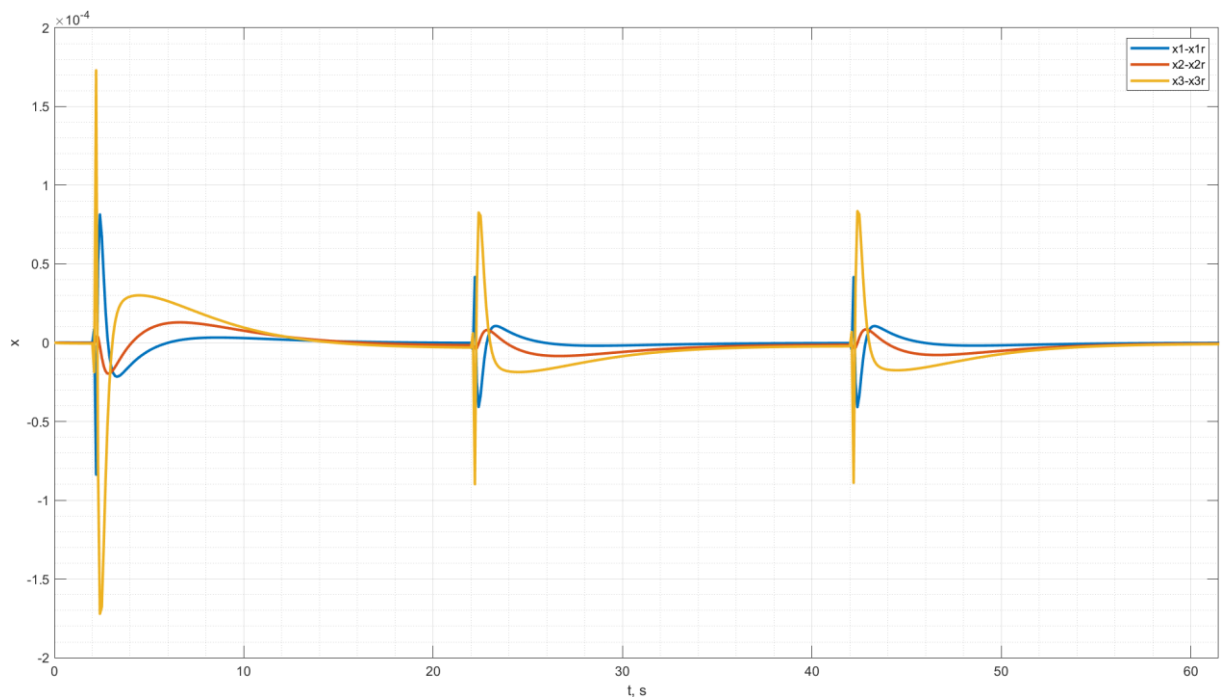
```

sss2=ss(Aa,Ba,Ca,Da,Ts);           %diskrétní aproximační model
[yr,~,X3] = lsim(sss2,u,t,x0(1:3)); %výstup modelu

```



Obr. 4.5 Porovnání modelu s redukováným stavem s původním



Obr. 4.6 Rozdíl původních a redukováných stavů

4.4 Simulace regulace

V MATLABu byl proveden návrh LQ regulátoru s integrací regulační odchylky pro simulaci regulace. Regulace je provedena pro model s redukovaným stavem. Výpočty pro simulaci vychází z rovnic (92), které byly detailněji popsány v kapitole 2.2.2

$$\begin{aligned}
 y(k) &= \mathbf{C}\mathbf{x}(k), \\
 u(k) &= -\mathbf{L}_x\mathbf{x}(k) - \mathbf{L}_w s_e(k), \\
 s_e(k+1) &= s_e(k) + w(k) - y(k), \\
 \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k).
 \end{aligned}
 \tag{92}$$

Pro výpočet akčního zásahu $u(k)$ bylo nutné zjistit hodnoty vektoru zesílení \mathbf{L} . K tomu slouží funkce `dlqr`, ale nejdřív se musely vytvořit rozšířené matice podle vztahu (45) a zvolit vhodně váhové matice \mathbf{Q} a \mathbf{R} .

```

Alq = [Aa; -Ca]; PM = [0;0;0;1];      %pomocné matice
Ar = [Alq PM];                       %rozšířená matice A
Br = [Ba; 0];                        %rozšířená matice B
[Lr,S,e] = dlqr(Ar,Br,Q,R);          %zjištění zesílení Lr

```

Výsledné váhové matice byly zvoleny

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}; \quad \mathbf{R} = [5].$$

Hodnoty vektoru zesílení pro ideální model jsou

$$\mathbf{Lr} = \left[\underbrace{-0,3101 \quad 3,8703 \quad 1,4122}_{\mathbf{L}_x} \quad \underbrace{-0,5721}_{\mathbf{L}_w} \right].
 \tag{93}$$

Pro sestavení algoritmu LQ regulátoru bylo potřeba znát hodnoty při počátečním stavu, který měl být ustálený a nenulový. Zároveň část kódu níže zajišťuje beznázarové připnutí regulátoru. Vychází se z teorie popsané v kapitole 2.2.2.1

```

Zxu = inv(eye(size(Aa))-Aa)*Ba;
Zyu = Ca*Zxu;
u0 = inv(Zyu)*w0;                       %akční zásah v čase t0

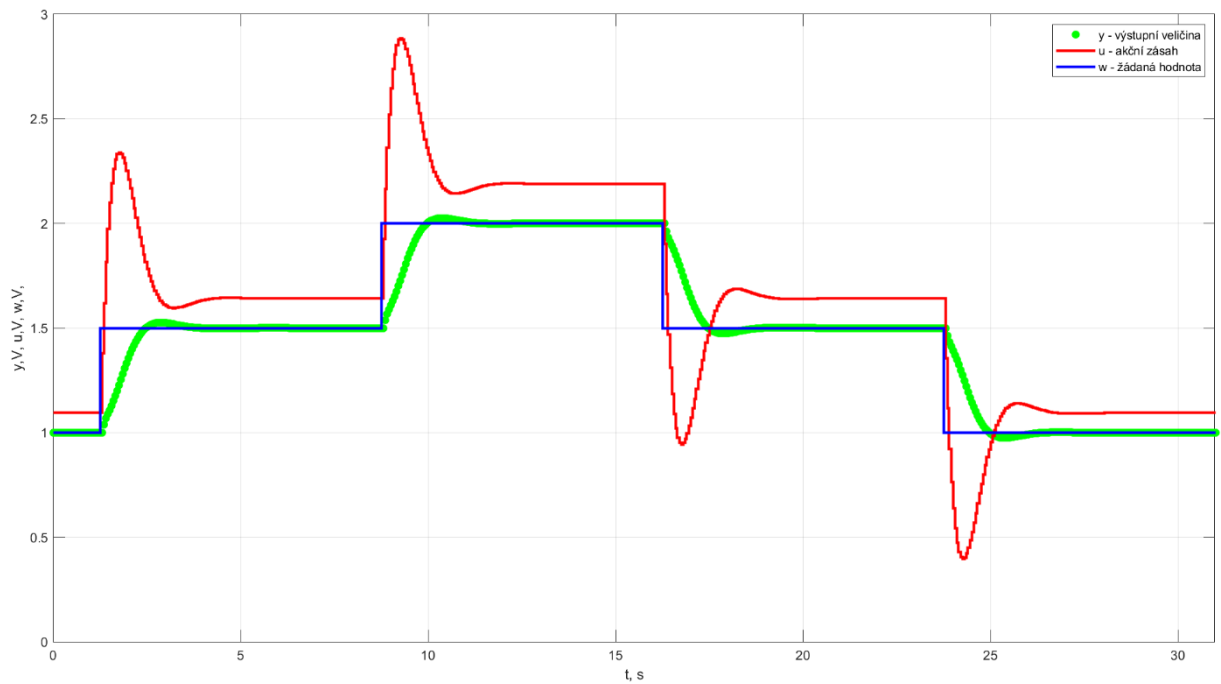
```

```
x = Zxu*inv(Zyu)*w0; %průběh stavů v čase t0
se = -inv(Lw)*(eye+Lx*Zxu)*inv(Zyu)*w0; %regulační odchylka
```

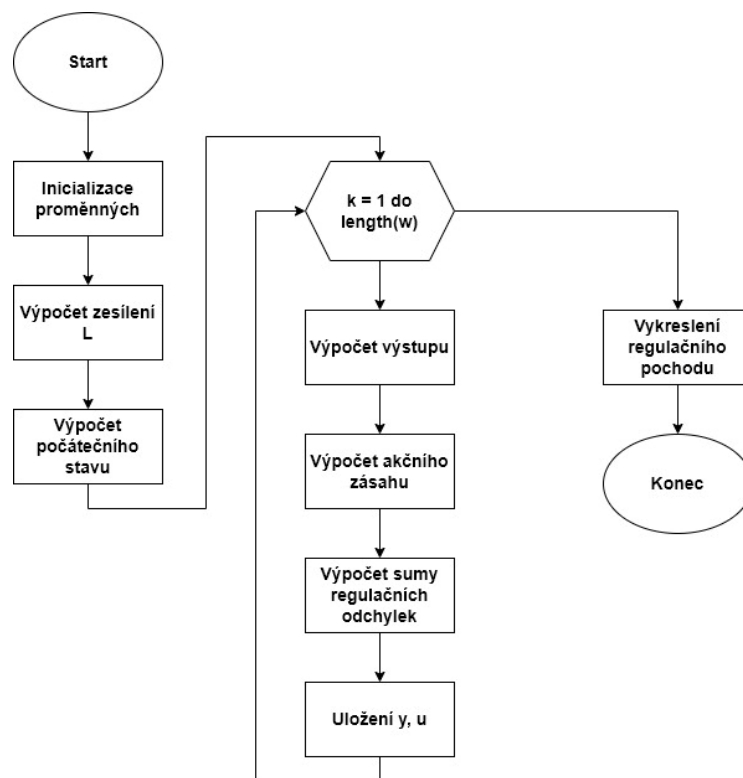
Posledním krokem bylo sestavení vlastního algoritmu pro regulaci, který je níže. Výstupem ze simulace je graf, který zobrazuje výstupní veličinu, žádanou veličinu a akční zásah. Regulace se dá ovlivnit volbou váhových matic **Q** a **R**. Z toho důvodu bylo provedeno více experimentů, zároveň se tím ověřila funkčnost algoritmu pro různé podmínky. Protože nebyly dány požadavky na řízení, váhové matice byly zvoleny tak, aby nebyl akční zásah zbytečně velký a zároveň nebyl regulační pochod pomalý. Na Obr. 4.7 je výsledný regulační pochod, tím se ověřila simulačně funkce LQ regulátoru.

```
for k = 1:1:n
    y = Ca*x; %výpočet výstupu
    u = -Lx*x-Lw*se; %výpočet akčního zásahu
    se = se+w(1,k)-y; %výpočet regulační odchylky
    x=Aa*x+Ba*u; %výpočet stavů
    Y(k)=y; %zapsání y
    U(k)=u; %zapsání u
end
```

Nejdříve se vypočítá výstupní veličina y v počátečním stavu, poté akční zásah v následujícím kroku a suma regulačních odchylek v následujícím kroku. V poslední řadě dojde k výpočtu následujících stavů x a uložení hodnot. Všechny tyto výpočty se provádí v cyklu, dokud není dosaženo poslední hodnoty n .



Obr. 4.7 Simulace regulačního pochodu

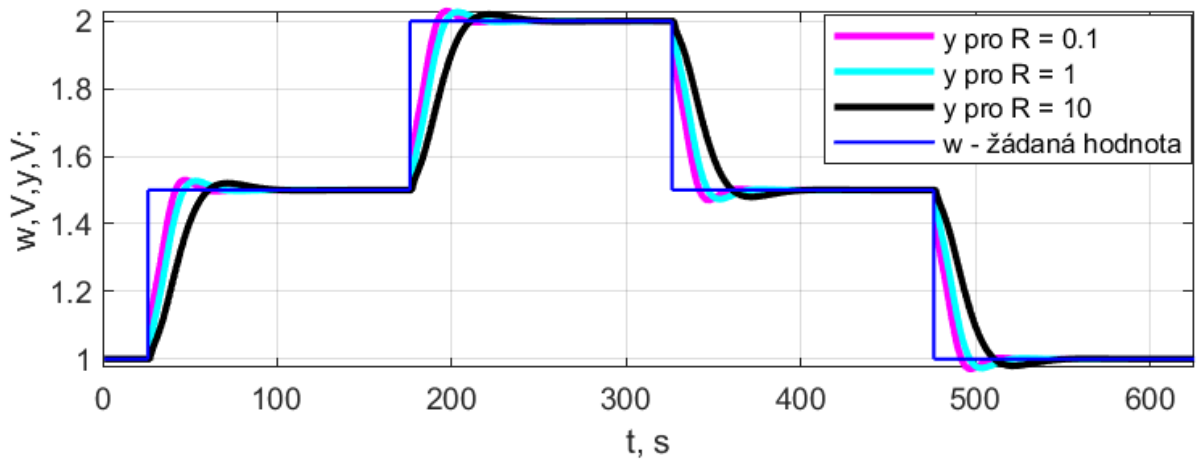


Obr. 4.8 Vývojový diagram simulace regulace

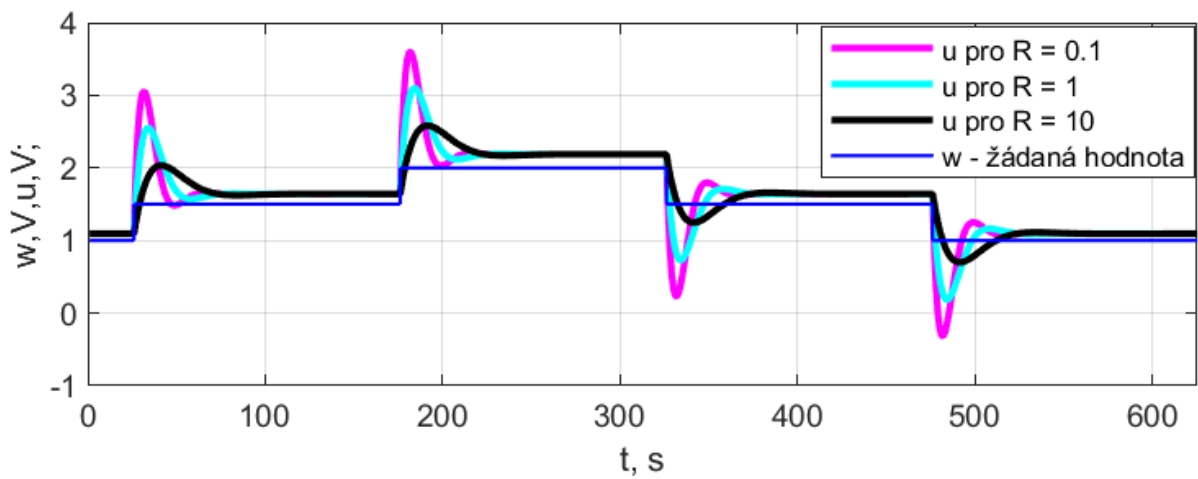
4.4.1 Vliv váhových matic na regulační pochod

Volbou váhových matic je možné ovlivnit regulační pochod. V této kapitole je experimentálně ověřeno, jakým způsobem změny matic ovlivňují regulaci. Změny matic jsou simulovány na systému z předchozí kapitoly. Nejdříve se uvažovalo diagonální matice

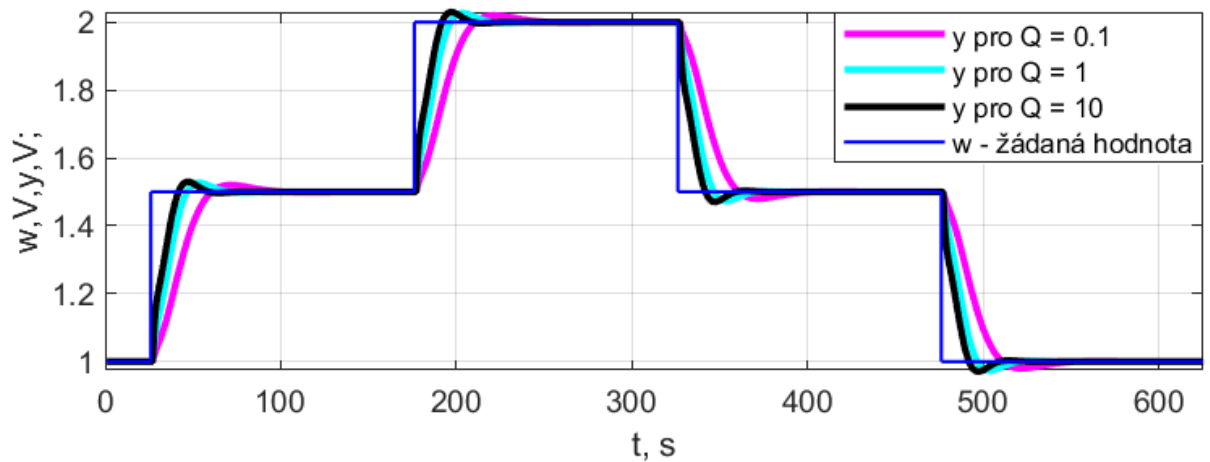
Q s jedničkami na hlavní diagonále a mění se hodnota matice R . V druhém případě se simulace prováděla s maticí R rovno jedné a měnila se matice Q .



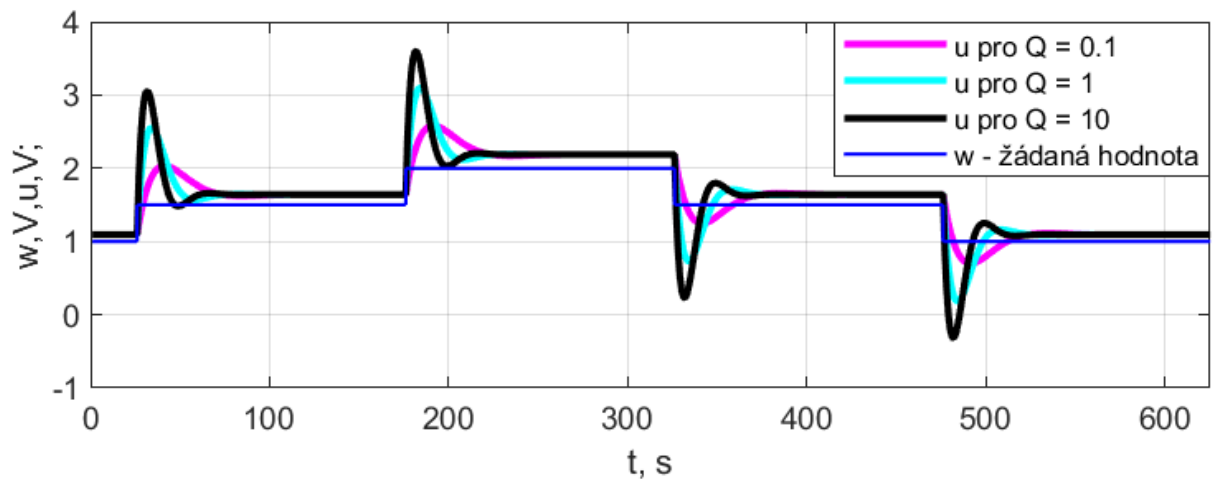
Obr. 4.9 Průběh výstupní veličiny v závislosti na změně matice R



Obr. 4.10 Průběh akčního zásahu v závislosti na změně matice R



Obr. 4.11 Průběh výstupní veličiny v závislosti na změně matice Q



Obr. 4.12 Průběh akčního zásahu v závislosti na změně matice Q

Z průběhů je vidět, že snižování hodnoty matice R povoluje větší akční zásah a tím dochází i k rychlejšímu dosažení žádané hodnoty. To samé platí při neměnné hodnotě matice R a zvyšování hodnoty matice Q . Z toho vyplývá, že čím je větší poměr váhové matice Q vůči váhové matici R , tím větší je akční zásah a ustálení na žádané hodnotě.

5 Implementace identifikace a regulace v prostředí MATLAB

Pro implementaci LQ regulátoru bylo opět využito vývojové prostředí MATLAB. Nejdříve byl vytvořen skript pro komunikaci s měřicí jednotkou (Arduinem). Následně se změřila data a uložila pro identifikaci. Skriptem pro identifikaci se určily vlastnosti soustavy, pro kterou se navrhne LQ regulátor. Jako poslední krok byla vytvořena simulace regulace s novými daty. Následně se skript upravil pro možnost regulace na řídicí jednotce.

5.1 Komunikace s měřicí jednotkou

Komunikace s měřicí jednotkou probíhá přes sériové rozhraní. V měřicí jednotce (Arduinu) je nahrán program, který má za úkol odpovídat na požadavky řídicí aplikace z PC. Hlavní funkcí Arduina má být generování vstupního napětíového signálu pro RC články, následně měření napětí na kondenzátorech a odeslání dat zpět do PC.

Řídicí aplikace využívá funkce v MATLABu, které jsem dostal k dispozici a mají za úkol nastavování parametrů Arduina. Funkce:

- **ArInIM** – vytvoření objektu sériového portu, nastavení Arduina do základního stavu a získání identifikačního řetězce programu v Arduinu. Vstupem je sériový port, přes který je Arduino připojeno do PC.
- **ArParSet** – nastavení parametrů měření v Arduinu (interval měření a počet měřených kanálů ADC).
- **ArAOSetM** – nastavení napětí na DAC převodníku, v řídicí aplikaci nastavuje hodnotu vstupu do RC článků (změna žádané hodnoty).
- **ArAIGetM** – čtení poslední sady měřených napětí a nastavení režimu. V režimu jednorázového čtení je při volání funkce **ArAIGetM** odeslán požadavek na přenos posledních měřených napětí a vyčte se odpověď. V režimu cyklického čtení vysílá Arduino periodicky s nastaveným intervalem měřená napětí a ta jsou na pozadí v MATLABu přijímána. Volání funkce **ArAIGetM** pak vrátí poslední přijatá data okamžitě pokud jsou k dispozici nebo čeká na jejich příjem, pokud již byla vyčtena.

5.2 Měření dat pro identifikaci a určení parametru modelu

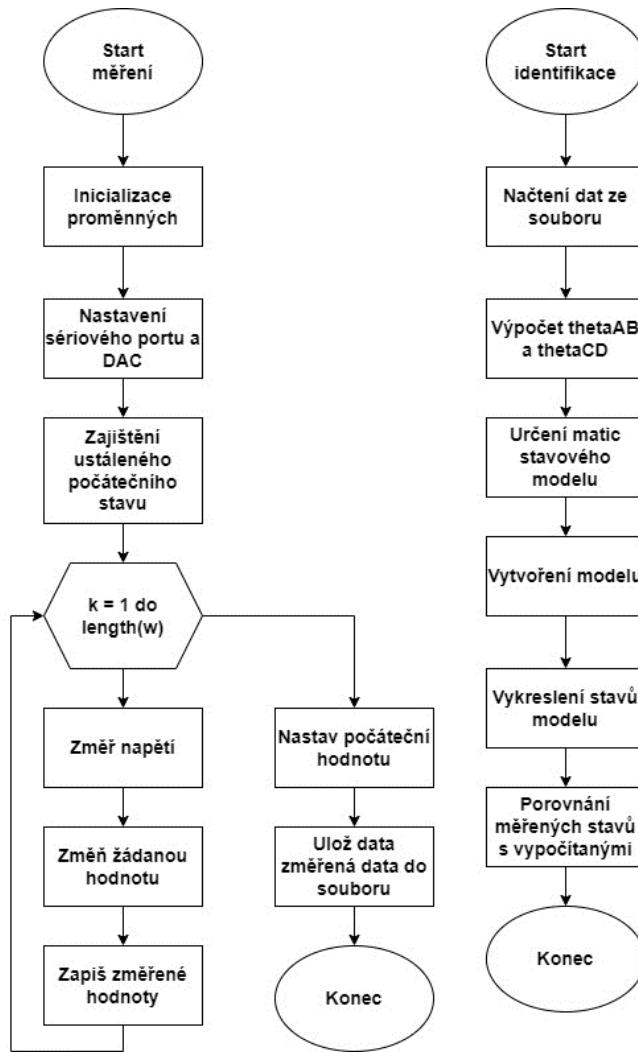
Jelikož není možné měřit všechny stavy RC článků, musí se provést měření a udělat identifikace parametrů modelu. Ve skriptu pro měření se nejdřív nastaví sériový port („COM“), nastaví se parametry Arduina a požadované vstupní napětí do soustavy definující počáteční

ustálený stav. Poté se inicializují všechny proměnné potřebné k měření (perioda vzorkování - 0,05s , počáteční stav, žádaná hodnota) viz kód níže.

```
s = ArIniM("COM3");           %vytvoření sériového portu
Ts = 0.05;                    %vzorkovací perioda
u0 = 1; u1 = 2;               %počáteční stav
ArParSetM (s,Ts*1e6,4);       %serialport, Ts, počet měřených kanálů
ArAOSetM(s,[u0,1]);          %nastavení DAC, vektor hodnot napětí
```

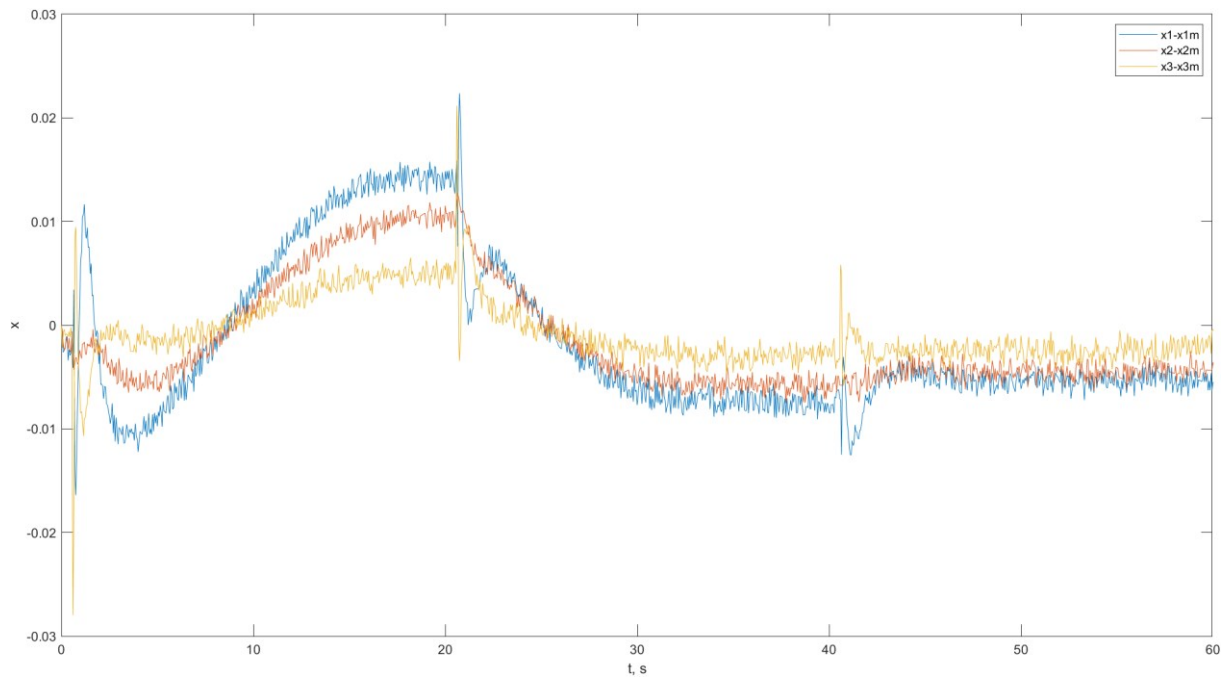
Po dosažení ustálení proběhne měření dat pro identifikaci. V cyklu se nastavují připravené hodnoty (vektor v) vstupního napětí a měřené hodnoty napětí vracené funkcí `ArAIGetM` se ukládají do matice D . Použití funkce `ArAIGetM` v cyklickém režimu zajišťuje synchronizaci provádění cyklu s přijímanými daty posílanými Arduinem s nastavenou periodou. Po změření dojde na nastavení napětí na počáteční hodnotu u_0 , aby se při dalším měření nemuselo čekat na pokles a ustálení. Aplikace skončí uložením potřebných hodnot do souboru pro možnost použití v dalším skriptu. Na Obr. 5.1 je zobrazen vývojový diagram jak pro měření, tak i pro identifikaci parametrů.

```
for k = 1:N
    AI = ArAIGetM (s,1);       %měření dat
    ArAOSetM (s,[v(k),0]);    %změna žádané
    D(:,k) = AI(1:5);         %uložení měřených stavů
end
ArAOSetM(s,[u0,1]);          %nastavení na původní hodnotu
```

Obr. 5.1 Vývojový diagram pro měření dat a identifikaci parametrů

Ve skriptu pro identifikaci se nejdříve načtou naměřená data ze souboru a upraví se pro použití do výpočtů. Určení matic stavového modelu probíhá stejně jako v kapitole 4.3.1. Poté se vytvoří diskrétní aproximační model příkazem `ss` a pomocí příkazu `lsim` získáme výstup modelu. Závěrem je vykreslení pro ověření správnosti výpočtů v MATLABu. Na Obr. 5.2 je rozdíl vypočítaných stavů x a stavů naměřených. Maximální odchylka nepřesahuje ani hodnotu 0,03 tím pádem je zanedbatelná.



Obr. 5.2 Rozdíl vypočítaných stavů a stavů naměřených

5.3 Regulace

Pro ověření funkčnosti navrženého regulátoru se nejdřív provedla simulace LQ regulátoru dosazením naměřených a identifikovaných parametrů do kódu popsaného v kapitole 4.4 Simulace byla kromě ověření funkčnosti využita pro rychlejší nastavení parametrů regulátoru. Při regulaci s reálnou soustavou, by se muselo čekat na dokončení celého experimentu a až potom by bylo možné znovu upravit parametry váhových matic.

Kód v MATLABu pro simulaci regulace se upravil. Po inicializaci proměnných a výpočtu zesílení \mathbf{L} se musel zajistit ustálený stav. Regulace měla začínat z ustáleného nenulového počátečního stavu se zajištěním beznárazového připnutí regulátoru. Beznárazové připnutí regulátoru vychází z hodnot počátečních měřených stavů \mathbf{x}_0 , z kterých se vypočítala počáteční suma regulačních odchylek se_0 . Jak je v kódu níže.

```
AI = ArAIGetM (s,θ);      %měření dat
x = AI(3:5);             %měřené x0
se = -(u0+Lx*x)/Lw;     %výpočet se0
```

Po výpočtu ustáleného počátečního stavu a beznárazového připnutí, začne probíhat cyklus `for`, který zajišťuje vlastní regulaci. Na začátku nejdříve změří data pomocí Arduina. Ze změřených dat se dosadí hodnota výstupu y a stavů \mathbf{x} . Oproti simulaci se u výpočtu akčního zásahu musí počítat s omezením, protože na Arduinu není možné měřit napětí větší než

přibližně 2,73 V a nižší než 0,53 V (omezení je dodělané i v simulaci, aby odpovídala co nejvíc reálné soustavě). Pokud dojde k překročení těchto limitů, akční zásah se nastaví na limitní hodnotu. Poté se vypočítá následující suma regulačních odchylek a odešle požadavek na nastavení akčního zásahu LQ regulátoru. Na konci cyklu se uloží změřené hodnoty stavů x , výstupní y a spočítaný akční zásah u .

Celý experiment byl prováděn s periodou vzorkování 0,05 s a délkou přibližně 45 sekund. Na Obr. 5.5 je vývojový diagram, který popisuje implementaci.

```

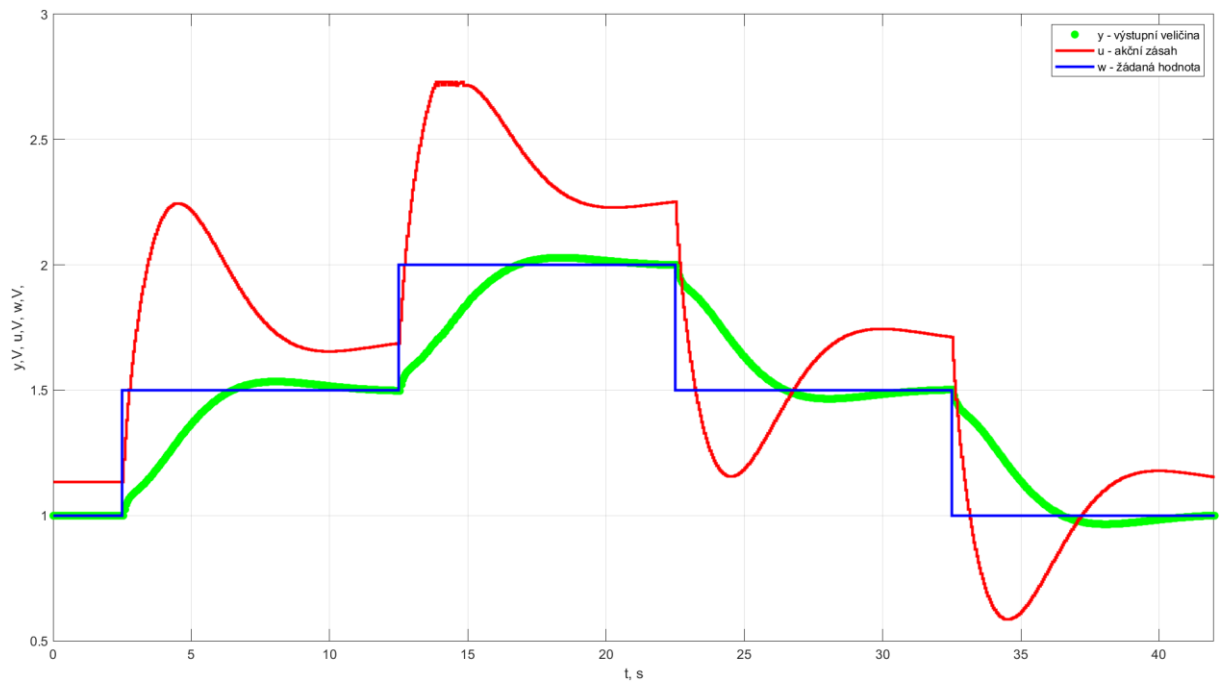
for k = 1:n          %cyklus pro regulaci
    AI = ArAIGetM (s,1); %měření dat
    y = AI(2);       %měřené yk
    x = AI(3:5);     %měřené xk
    u = -Lx*x-Lw*se; %výpočet akčního zásahu
    if u>2.7, u=2.7; se = -(u+Lx*x)/Lw;
    else se = se+w(1,k)-y;
    end %omezení překročeního max. u
    if u<0.53, u=0.53; se = -(u+Lx*x)/Lw;
    else se = se+w(1,k)-y;
    end %omezení překročeního min. u
    ArAOSetM(s,[u,1]); %nastavení u LQ regulátoru
    X(k,:)=x;         %uložení x
    Y(k)=y;          %uložení y
    U(k)=u;          %uložení u
end

```

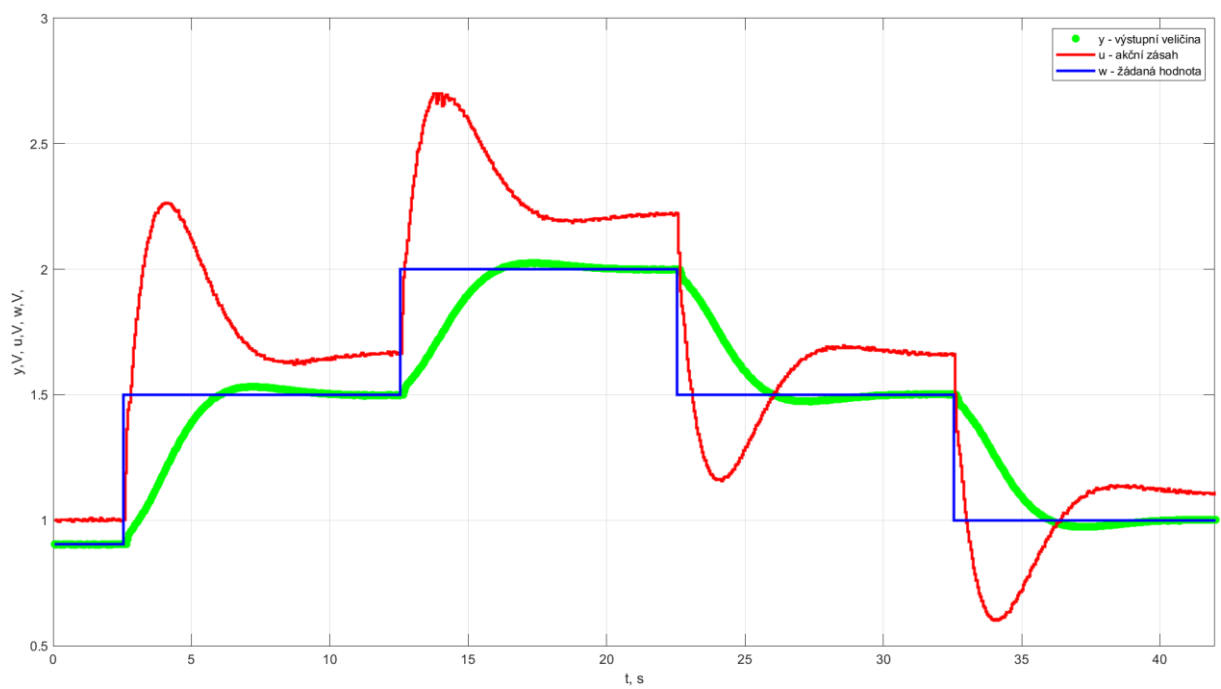
Váhové matice byly nastaveny

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0,002 \end{bmatrix}; \quad \mathbf{R} = [0,5].$$

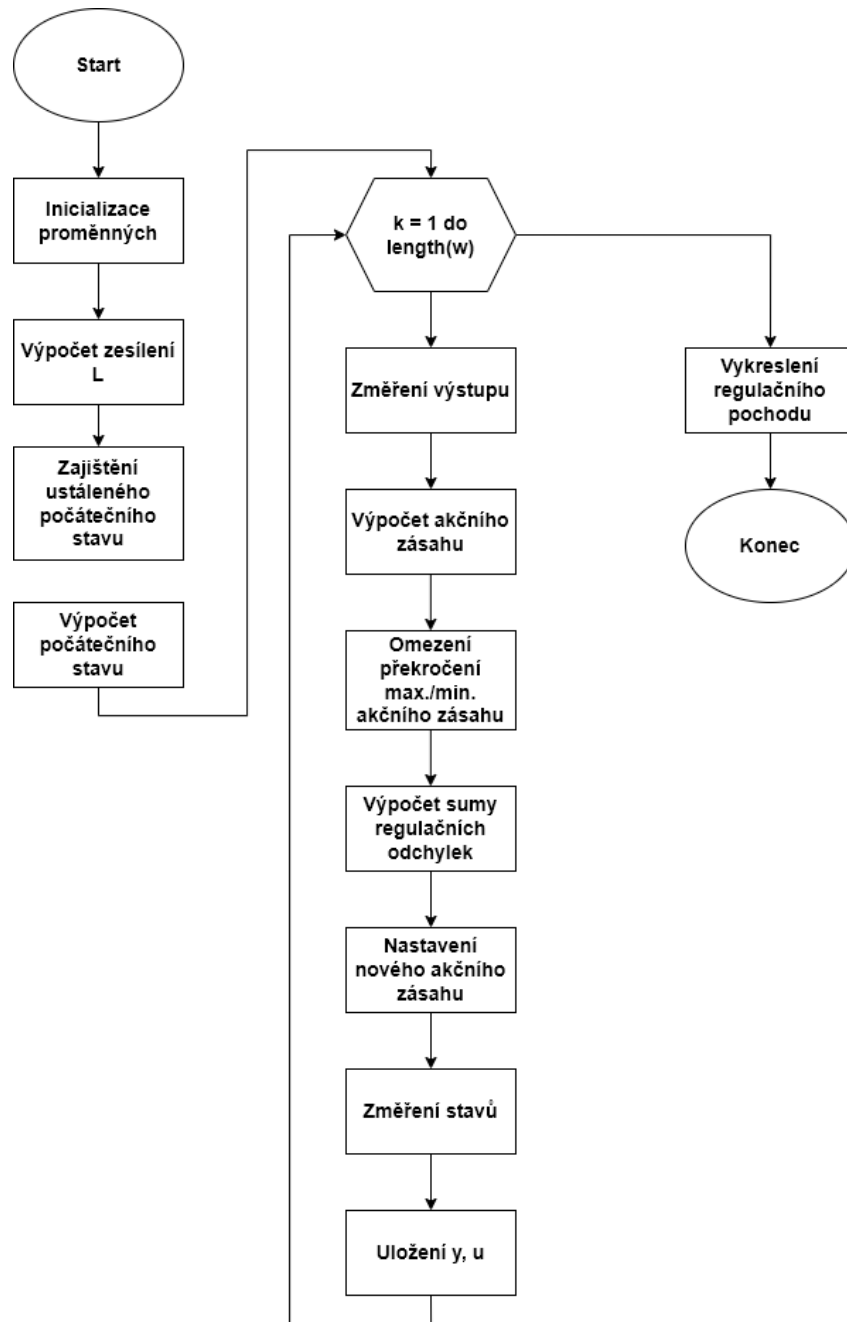
Na Obr. 5.3 je vykreslen průběhy regulace ze simulace a na Obr. 5.4 regulační pochod pro ověření regulace s reálnou soustavou. Průběhy jsou téměř totožné. V čase cca 15s je vidět, že dochází k omezení akčního zásahu, kdy akční zásah zůstane na limitní hodnotě.



Obr. 5.3 Simulace regulačního pochodu



Obr. 5.4 Regulační pochod



Obr. 5.5 Vývojový diagram regulace

6 ZÁVĚR

Cílem práce bylo identifikovat parametry stavového modelu s následným návrhem LQ regulátoru se sledováním žádané bez trvalé regulační odchylky.

V teoretické části je popsána problematika návrhu LQ regulátoru jak ve spojité oblasti, tak v diskrétní oblasti. Včetně rozšíření o asymptotické sledování žádané hodnoty. Následně je v teorii popsána identifikace soustavy, která se provádí za účelem získání stavového modelu soustavy. Experimentální identifikace popisuje získání parametrů z měřených stavů, vstupů a výstupů. Aby nemusel být v praktické části použit pozorovatel stavů, tak se počítá s aproximačním modelem s redukováným stavem.

V praktické části jsou popsány části kódů programů, které sloužily pro vytvoření modelu, identifikaci parametrů stavového modelu. Poté návrh a ověření funkce LQ regulátoru, jak simulačně, tak i s elektrickým systémem připojeným k PC přes Arduino DUE.

Při identifikaci bylo ověřeno, že rozdíl mezi aproximačním modelem a ideálním modelem byl zanedbatelný. Ale po naměření dat na soustavě byl zjištěn rozdíl mezi reálnými součástkami a hodnotami nominálních hodnot součástek použitých pro výpočet ideálního modelu. Z toho důvodu bylo potřeba provést korekci parametrů součástek. Pomocí optimalizace byl vytvořen korekční vektor, kterým se násobí nominální hodnoty součástek, aby došlo k co největší shodě s realitou.

Návrh a ověření LQ regulátoru proběhlo v MATLABu. V porovnání se simulací se výsledky řízení velmi shodují. Nejdříve byl oproti simulaci rozdíl, protože Arduino má limitní hodnoty maximální a minimální, které dokáže generovat. Proto muselo být omezení přidáno programově i do simulace. Soustava byla dobře regulovatelná. Výslednou identifikaci ovlivňovalo, pokud počáteční stav nebyl ustálený, identifikované průběhy stavů potom byly pokaždé jiné. Na tom závisela i regulace.

POUŽITÁ LITERATURA

Arduino. *Arduino* [online]. 2023 [cit. 2023-04-17]. Dostupné z:

<https://www.arduino.cc/en/about>.

Arduino Due. *Arduino* [online]. 2023 [cit. 2023-04-17]. Dostupné z:

<https://store.arduino.cc/products/arduino-due>.

BALÁTĚ, Jaroslav. *Automatické řízení*. 2. aktualizované vydání. Praha: BEN – technická literatura, 2004. ISBN 978-80-7300-148-3.

DUŠEK, František. *MATLAB a SIMULINK: úvod do používání*. Pardubice: Univerzita Pardubice, 2000. ISBN 80-7194-273-1.

MACHEK, Ondřej. *LQ REGULACE SYSTÉMU MOTOR GENERÁTOR* [online]. Pardubice, 2019, 56 s. [cit. 2023-04-17]. Dostupné z:
https://dk.upce.cz/bitstream/handle/10195/73107/MachekO_LQRegulace_FD_2019.pdf?sequence=1&isAllowed=y. Diplomová práce. Univerzita Pardubice. Vedoucí práce doc. Ing. František Dušek, CSc.

OGATA, Katsuhiko. *Discrete-time control systems*. Second edition. Englewood Cliffs: Prentice Hall, 1995. ISBN 0-13-034281-5.

PIVOŇKA, Petr. *Optimalizace regulátorů* [online]. Brno: VUT FEKT, 2005 [cit. 2023-04-17]. Dostupné z:
http://matlab.fei.tuke.sk/raui_new/subory/literatura/pivonka_optimalizace_regulatoru.pdf. Skriptum.

STREJC, Vladimír. *Stavová teorie lineárního diskrétního řízení*. Praha: Academica, 374 s, 1978.

ŠTECHA, Jan a Vladimír HAVLENA. *Teorie dynamických systémů*. 2. vydání. Praha: České vysoké učení technické, 1999. ISBN ISBN 80-01-01971-3.

VROŽINA, Milan, Zora JANČÍKOVÁ a Jiří DAVID. *Identifikace systémů* [online]. Ostrava: VŠB – Technická univerzita Ostrava, 2012 [cit. 2023-04-17]. ISBN 978-80-248-2594-6. Dostupné z: <http://www.person.vsb.cz/archivcd/FMMI/IS/Identifikace%20systemu.pdf>.

PŘÍLOHY

A – CD

Příloha k diplomové práci

Experimentální identifikace diskrétního stavového modelu

Josef Jelínek

CD

Obsah

- 1 Text diplomové práce ve formátu PDF
- 2 Úplný zdrojový kód v MATLABu pro vytvoření modelu, korekci a identifikaci parametrů, generování a měření napětí na soustavě, návrh a ověření LQ regulátoru