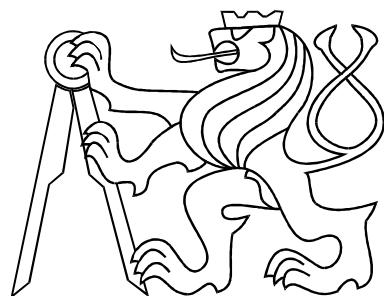


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ



BAKALÁŘSKÁ PRÁCE

**Uživatelské rozhraní
pro návrh frekvenčních filtrů**

2006

Martin Řezáč

Autor: Martin Řezáč
Vedoucí: Ing. Ondřej Holub
 Katedra řídicí techniky, Fakulta elektrotechnická,
 České vysoké učení technické v Praze
Oponent: Ing. Radek Frízel
 Katedra matematiky, Fakulta elektrotechnická,
 České vysoké učení technické v Praze
Rok: 2006

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, SW atd.) uvedené v přiloženém seznamu.

V Praze dne 30.6.2006.

podpis

Poděkování

Děkuji především svému vedoucímu bakalářské práce Ing. Ondřeji Holubovi a jeho kolegovi Ing. Petru Augustovi za ochotu a cenné připomínky při řešení této práce.

Dále pak děkuji rodičům a přítelkyni za jejich podporu během celé doby mého studia.

Abstrakt

Práce popisuje grafické uživatelské rozhraní (GUI) pro balík Matlab, sloužící k pohodlnému návrhu frekvenčních filtrů. Přínos tohoto GUI spočívá v možnosti jednoduchým způsobem měnit tvar váhových filtrů. Filtry jsou použity pro optimalizaci citlivostních funkcí zpětnovazebního systému založenou na minimalizaci jejich váhových \mathcal{H}_∞ norm. Parametry vhodné pro tento případ se liší od specifikace, běžně používané v oblasti zpracování signálu. Navržené GUI tak z uživatele, inženýra navrhujícího regulační obvod, snímá potřebu ovládat analytické vyjádření pro přenos váhových filtrů, což zrychluje a zpohodlňuje jeho práci. Použitelnost GUI je demonstrována na příkladu návrhu regulátoru rychlosti koncového bodu pružného ramene, připojeného na hřídel stejnosměrného motoru.

Abstract

This work describes a graphical user interface (GUI) for Matlab package. The GUI is used for comfortable design of frequency filters. Benefits of the GUI are the following. The GUI allows a user to easily change type and parameters of weighting filters. These filters are used for optimization of sensitivity transfer functions of a feedback system. The optimization is based on minimization of weighted \mathcal{H}_∞ norms of these transfer functions. Parameters that are suitable in this case differ from specification commonly used in the field of signal processing. The designed GUI handles analytical expressions of transfer functions. Thus the GUI makes the work of the user (control engineer designing the feedback system) much more comfortable. Usability of the GUI is demonstrated on an example of speed controller design for the tip point of a flexible link connected to a shaft of a DC motor.

Katedra řídicí techniky

Školní rok: 2005/2006

Zadání bakalářské práce

Student: Martin Řezáč

Obor: Kybernetika a měření

Název tématu: Uživatelské rozhraní pro návrh frekvenčních filtrů

Zásady pro výpracování:

1. Seznamte se s problematikou formulace úlohy návrhu řízení ve frekvenční oblasti.
2. Vytvořte grafické uživatelské rozhraní pro snadnou volbu frekvenčních filtrů, používaných jako váhovací funkce při návrhu řízení pomocí minimalizace H_∞ -normy citlivostních funkcí zpětnovazebního systému.
3. Demonstrujte použitelnost vytvořeného rozhraní na návrhu řízení laboratorního modelu, výsledky návrhu ověřte.

Seznam odborné literatury: Dodá vedoucí práce

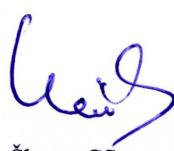
Vedoucí bakalářské práce: Ing. Ondřej Holub

Datum zadání bakalářské práce: zimní semestr 2005/06

Termín odevzdání bakalářské práce: 30. 6. 2006

Prof. Ing. Michael Šebek, DrSc.
vedoucí katedry




Prof. Ing. Zbyněk Škvor, CSc.
děkan

Obsah

1	Úvod	1
2	Metoda minimalizace \mathcal{H}_∞ normy	3
2.1	Úvod	3
2.2	Úkol řízení	4
2.3	Požadavky na citlivostní funkce a jejich omezení	6
3	Grafické uživatelské rozhraní	7
3.1	Základní popis	7
3.2	Filtry	8
3.3	Odvození parametrů filtrů	9
3.3.1	Filtr typu dolní propust	9
3.3.2	Filtr typu horní propust	10
3.3.3	Filtr typu lead	12
3.4	Programování v Matlabu	13
3.5	Příklad návrhu filtru	13
4	Příklad použití GUI na reálném modelu	15
4.1	Úvod	15
4.2	Model pružného ramene	15
4.3	Matematický model soustavy	16
4.4	Postup návrhu regulátoru	17
4.5	Měření na reálném systému	21
4.6	Výsledky	22
5	Závěr	23
	Literatura	25

A Obsah přiloženého CD	27
B Část programu	29

Seznam obrázků

2.1	Zpětnovazební smyčka	3
2.2	Zpětnovazební smyčka s váhovými filtry	5
3.1	Ukázka GUI pro návrh frekvenčních filtrů	8
3.2	Typy filtrů, které lze v GUI navrhovat	9
3.3	Tvar frekvenční charakteristiky filtru typu dolní propust	9
3.4	Tvar frekvenční charakteristiky filtru typu horní propust	10
3.5	Tvar frekvenční charakteristiky filtru typu lead	12
4.1	Obrázky modelu	16
4.2	Detail soukolí	16
4.3	Frekvenční charakteristika	17
4.4	M-soubor v Matlabu provádějící výpočet regulátoru	19
4.5	Výpis v Matlabu po zavolání příkazu „ <code>hinfopt</code> “	19
4.6	Výsledek první iterace návrhu řízení	20
4.7	Simulinkový model	21
4.8	Naměřené přechodové charakteristiky	22

Kapitola 1

Úvod

Tato práce se zabývá návrhem regulátorů pro lineární systémy pomocí takzvané metody „minimalizace \mathcal{H}_∞ normy zpětnovazebních citlivostních funkcí“. Tato metoda respektuje nejistoty modelu, problémy s potlačením poruchových veličin a omezení zásahů do řízené soustavy. Práce předvádí její použití z praktického hlediska. Vstupem této metody jsou, kromě modelu řízené soustavy, tzv. váhové filtry, reprezentující požadavky na řízení.

Samotný návrh regulátoru je iterativním procesem, neboť typicky je potřeba přesně zformulovat vágní zadání problému řízení a najít vhodný kompromis mezi jednotlivými často si odpovídajícími požadavky. Obvykle tedy návrhář vyzkouší různé typy filtrů s mnoha variantami nastavení jejich parametrů.

K tomu účelu bylo navrženo grafické uživatelské rozhraní [3] (dále jen GUI), které si klade za cíl zrychlit proces návrhu a učinit jej pohodlnějším. Snímá z uživatele potřebu rychle se orientovat v souvislostech mezi parametry filtru a tvarem jeho frekvenční charakteristiky. Pro typy filtrů, běžně používaných jako vstupní data \mathcal{H}_∞ optimalizace, nabízí snadné získání analytického zápisu přenosu filtru na základně zadaných požadavků na tvar jeho frekvenční charakteristiky. GUI používá pro analytické vyjádření váhových filtrů klasické parametry [6]. Nový je ovšem výpočet přesných hodnot těchto parametrů, splňujících zadání na tvar váhového filtru. Výpočet parametrů filtrů byl zahrnut do nástrojů GUI.

Kap. 2 seznamuje čtenáře se základy návrhu regulátoru ve frekvenční oblasti. Jsou zde vysvětleny pojmy vstupní a výstupní citlivost, představena je také zmínovaná metoda minimalizace \mathcal{H}_∞ normy zpětnovazebních citlivostních funkcí se svými možnostmi a omezeními.

Kap. 3 pak navazuje s popisem grafického rozhraní. Jsou zde popsány parametry váhových filtrů a také odvozeny některé základní vztahy pro jejich výpočet. Kapitola je

zakončena praktickým příkladem postupu při práci s GUI.

V poslední kapitole je předvedena aplikace použití \mathcal{H}_∞ optimalizace pro návrh regulátoru rychlosti koncového bodu pružného ramene připojeného na hřídel stejnosměrného motoru. Kapitola popisuje postup návrhu krok po kroku od definice požadavků až po závěrečnou realizaci řízení modelu z prostředí Simulinku.

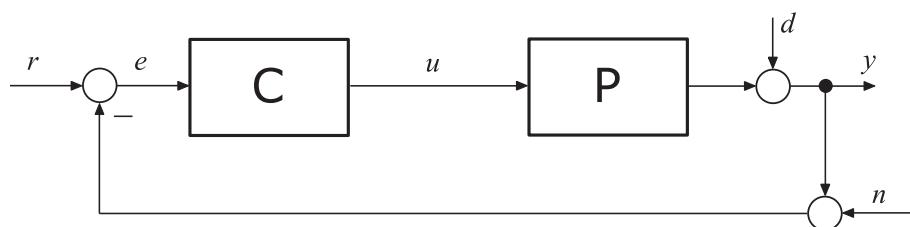
Kapitola 2

Metoda minimalizace \mathcal{H}_∞ normy zpětnovazebních citlivostních funkcí

V této kapitole budou popsány základy návrhu regulátoru ve frekvenční oblasti. Budou vysvětleny pojmy jako jsou vstupní a výstupní citlivost. Druhá část kapitoly pak popisuje metodu minimalizace \mathcal{H}_∞ normy zpětnovazebních citlivostních funkcí a seznamuje s některými omezeními při návrhu.

2.1 Úvod

Uvažujme soustavu s přenosem P . Úkolem řízení je splnit požadavky kladené na soustavu pomocí regulátoru C (vizte obr. 2.1). Regulátor ovlivňuje vstup soustavy podle velikosti odchylky mezi žádanou hodnotou r a měřenou hodnotou výstupu soustavy y .



Obrázek 2.1. Zpětnovazební smyčka

Měření výstupu y obsahuje šum senzoru a do uzavřené smyčky systému vstupují také vnější poruchy. Model P nepopisuje obvykle soustavu dostatečně přesně. Tato odchylka

(označovaná jako nejistota modelu) je způsobena jednak zjednodušeními během modelování řízeného systému, jednak změnami parametrů soustavy během činnosti.

Pro návrh regulátoru C , který zahrnuje všechny kladené požadavky na systém a je odolný vůči nejistotě modelu soustavy, je možné použít **metodu minimalizace \mathcal{H}_∞ normy zpětnovazebních citlivostních funkcí**. Tato metoda se snaží najít regulátor, který stabilizuje uzavřenou smyčku a minimalizuje \mathcal{H}_∞ normu maticové funkce, která se skládá z vážených zpětnovazebních přenosových funkcí.

2.2 Úkol řízení

Typickým úkolem návrhu regulátoru je zajistit vnitřní stabilitu uzavřené smyčky zpětnovazebního systému i přes nejistotu v modelu, urychlit přechodové děje a zabránit příliš velkým překmitům těchto dějů, minimalizovat vliv poruch a přitom nedosáhnout saturace na výstupu regulátoru. Všechny tyto požadavky lze zformulovat pomocí jistých přenosových funkcí [5]. Zavedeme tedy pojmy citlivost S , doplňkovou citlivost T , a vstupní citlivost U . Citlivost je přenosová funkce z reference r na vstup regulátoru e . Doplňková citlivost je přenos z reference r na výstup y . Poslední vstupní citlivost U je přenos z reference r na výstup regulátoru u . Podle obr. 2.1 budou vypadat vztahy S , T a U takto

$$\begin{aligned} S &= \frac{1}{1+PC} \\ T &= \frac{PC}{1+PC} \\ U &= \frac{C}{1+PC}. \end{aligned} \tag{2.1}$$

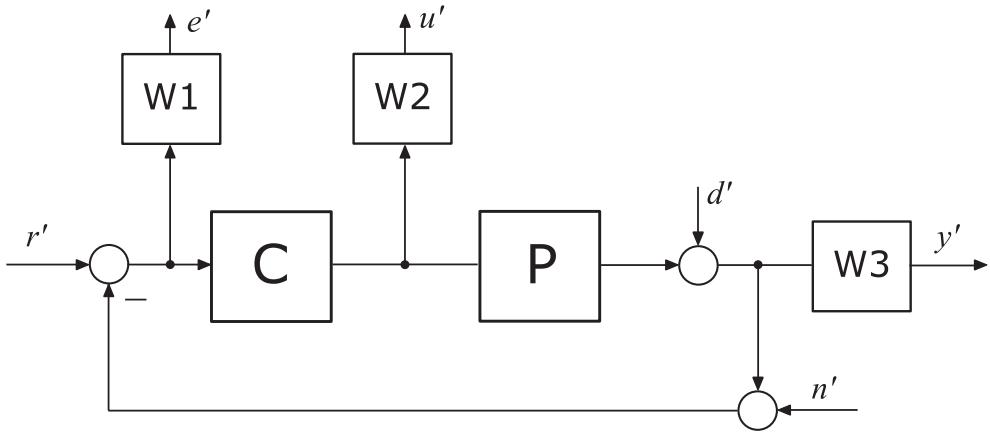
Pro odvození požadavků na přenosové funkce je třeba zavést \mathcal{H}_∞ normu. \mathcal{H}_∞ norma přenosové funkce G SISO systémů (z anglického „single input single output“ značí systémy s jedním vstupem a jedním výstupem) je definovaná podle [4] jako

$$\|G(j\omega)\|_\infty = \sup_{\omega \in \mathbb{R}} |G(j\omega)|.$$

V SISO systémech je tedy $\|G(j\omega)\|_\infty$ rovna hodnotě vrcholu amplitudové bodeho charakteristiky. Pomocí \mathcal{H}_∞ normy nyní mohou být vyjádřeny požadavky na jednotlivé přenosové funkce.

1. Pro odolnost vůči poruchám by měla být $\|S(j\omega)\|_\infty$ co nejmenší.
2. Pro potlačení šumu by měla být $\|T(j\omega)\|_\infty$ co nejmenší.
3. Pro dobré sledování reference by měla být $\|T(j\omega)\|_\infty \approx 1$.
4. Pro předcházení saturaci akčního zásahu by měla být $\|U(j\omega)\|_\infty$ co nejmenší.

Poznamenejme jen, že tyto požadavky nemusí nutně vést na požadavek minimalizace zpětnovazebních funkcí v celém frekvenčním rozsahu. Právě naopak. Každá zpětnovazební funkce musí být minimalizována v určitém frekvenčním rozsahu a s určitou rozdílnou vahou. Proto je třeba zavést frekvenčně závislé váhy – váhové filtry.



Obrázek 2.2. Zpětnovazební smyčka s váhovými filtry

Vezmeme-li v úvahu zpětnovazební model z obr. 2.2, přidáním váhových filtrů W_1, W_2 a W_3 do zpětnovazební smyčky z obr. 2.1 budou vypadat funkce S, U a T takto:

$$S' = W_1 S = \frac{W_1}{1 + PC}$$

$$T' = W_3 T = \frac{W_3 PC}{1 + PC}$$

$$U' = W_2 U = \frac{W_2 C}{1 + PC}.$$

Použitím váhových filtrů W_1, W_2 a W_3 je možno vyjádřit požadavky na zpětnovazební funkce S, U a T jednoduše jako současnou minimalizaci norm $\|W_1 S\|_\infty$, $\|W_2 U\|_\infty$ a

$\|W_3 T\|_\infty$. Maticově zapsáno

$$\min_C \mathcal{H}_\infty = \min_C \left\| \begin{pmatrix} W_1 S \\ W_2 U \\ W_3 T \end{pmatrix} \right\|_\infty. \quad (2.2)$$

2.3 Požadavky na citlivostní funkce a jejich omezení

Nyní rozvedeme požadavky kladené na přenosové funkce v předešlém odstavci. Pro dobré sledování reference je nutné, aby $T \approx 1$. Naopak pro dobré potlačení šumu je nutné, aby T bylo co nejmenší. Z těchto dvou požadavků plyne, že pro nízké frekvence ($\omega < \omega_{BW}$, kde ω_{BW} je šířka pásma) bude $T = 1$. Naopak pro frekvence $\omega > \omega_{BW}$ se musí T co nejvíce zmenšovat, aby došlo k potlačení šumu.

Při návrhu tvaru citlivostních funkcí T a S je nutné respektovat určitá pravidla. Z rovnice (2.1) plyne zajímavý fakt, že

$$S + T = 1.$$

Takže požadavek sledování reference vede na požadavek $S \approx 0$ v oblasti nízkých frekvencí. Dokonce je možné ukázat (vizte např. [4]), že asymptotické sledování reference k -tého řádu vyžaduje, aby citlivostní funkce S měla $k + 1$ nul v nule. Odolnost vůči poruchám vyžaduje S co nejmenší. Avšak z důvodu vazby mezi S a T je ve skutečnosti tato odolnost umožněna pouze v pásmu nízkých frekvencí.

Potlačení šumu je omezeno doplňkovou citlivostí T . Není překvapením, že šum může být potlačen jen v pásmu vysokých frekvencí, a tedy že nízkofrekvenční šum není možné potlačit použitím zpětné vazby.

Konečně pro předcházení vzniku saturace řídicího signálu, by měla být vstupní citlivost U malá. Tohle se většinou řeší omezením U konstantou.

Dalším omezením při návrhu tvaru filtrů je takzvaný bodeho integrál citlivostní funkce

$$\int_0^\infty \ln S \, d\omega = 0.$$

Z tohoto integrálu plyne, že při požadavku $|S| < 1$ na nízkých frekvencích musí S na vysokých frekvencích překmitnout. Tento jev se někdy také nazývá „the waterbed effect“, čili „efekt vodní postele“.

Kapitola 3

Grafické uživatelské rozhraní

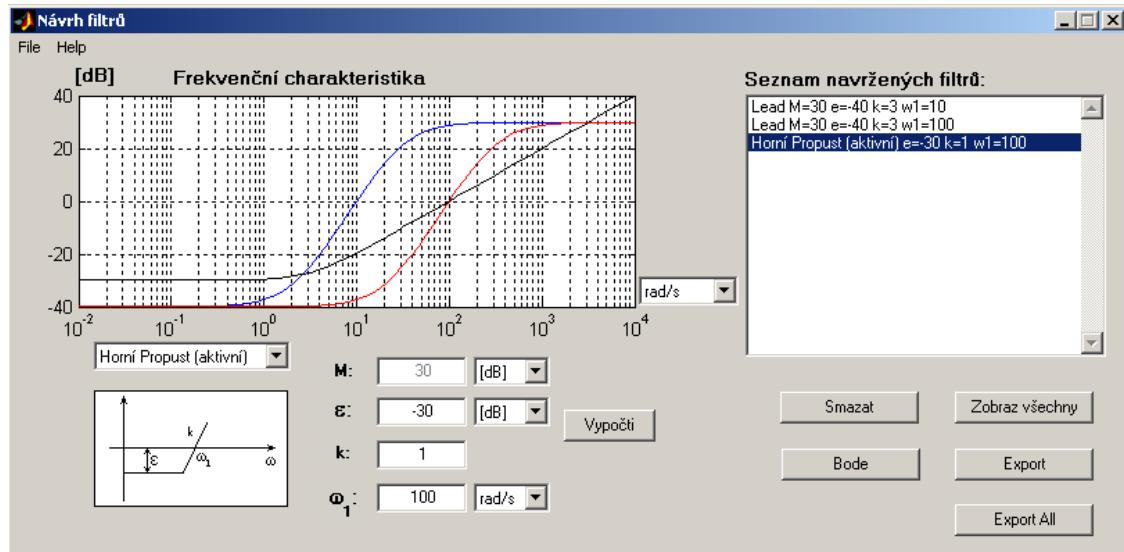
Tato kapitola představuje v úvodu zmíněné grafické uživatelské rozhraní sloužící k návrhu váhových filtrů. Jsou zde popsány parametry váhových filtrů a také odvozeny některé základní vztahy pro jejich výpočet. V závěru kapitoly je uveden praktický příklad postupu při návrhu filtru pomocí GUI.

3.1 Základní popis

Samotné grafické rozhraní bylo vytvořeno v prostředí Matlab [1]. GUI se skládá z několika částí. Obsahuje okno náhledu, editační boxy a seznam filtrů (vizte obr. 3.1). Okno náhledu zobrazuje amplitudovou frekvenční charakteristiku navržených filtrů. Editační boxy slouží ke specifikaci parametrů navrhovaných filtrů. Seznam navržených filtrů přehledně zobrazuje aktuální filtry včetně jejich klíčových parametrů [6]. Jsou to řád filtru (v GUI označen jako k), limitní hodnoty bodeho charakteristiky (označeny jako ϵ a M) a kmitočet pro který má daný filtr jednotkové zesílení (označen jako ω_1). Parametry ϵ a M jsou označeny tak, že M je vždy větší než 0 dB a ϵ naopak vždy menší než 0 dB. Význam jednotlivých parametrů právě navrhovaného filtru je jasně zřejmý z okna náhledu filtru. Uživatel tedy musí nejprve specifikovat své požadavky výběrem příslušného filtru a nastavením patřičných parametrů. Poté po výpočtu filtru jsou všechny potřebné informace o filtru uloženy v seznamu filtrů.

S filtrem je možné také dále pracovat. Filtr (filtry) je možné exportovat do aktuálního workspace Matlabu. Přitom je uživatel vyzván k zadání jména proměnné pod kterou bude filtr uložen ve formátu ZPK (zero-pole-gain formát). GUI také umožňuje zobrazit

v náhledu všechny filtry dohromady. To je výhodné zejména pro porovnání charakteristik jednotlivých filtrů.



Obrázek 3.1. Ukázka GUI pro návrh frekvenčních filtrů

3.2 Filtry

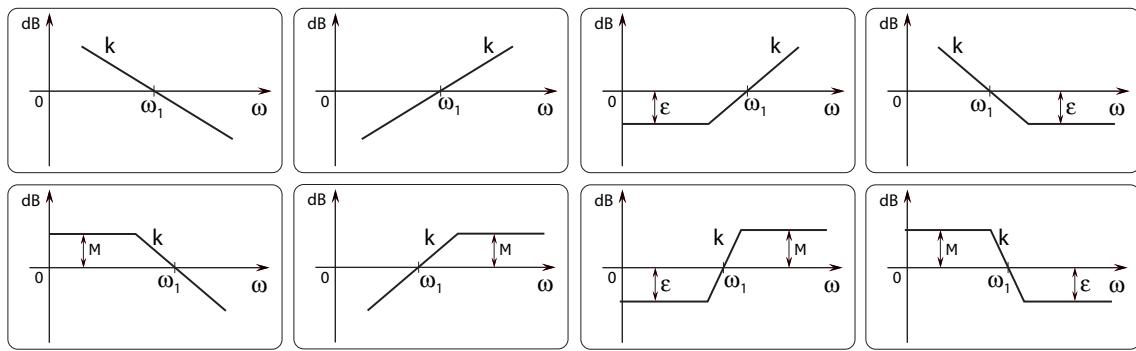
Pomocí grafického rozhraní je možno navrhnut celkem 8 typů filtrů (vizte obr. 3.2). Jejich přenosy jsou však ve skutečnosti postaveny na čtyřech různých přenosech. Zbylé 4 přenosy jsou vždy převrácenou hodnotou některého z těchto čtyř přenosů.

$$W_1 = \left(\frac{a}{s} \right)^k \quad (3.1)$$

$$W_2 = \left(\frac{b}{s+a} \right)^k \quad (3.2)$$

$$W_3 = \left(\frac{b \cdot s}{s+a} \right)^k \quad (3.3)$$

$$W_4 = \left(\frac{as+b}{s+c} \right)^k \quad (3.4)$$



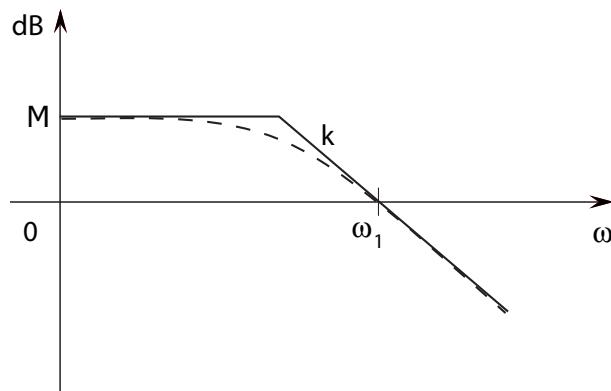
Obrázek 3.2. Typy filtrů, které lze v GUI navrhovat

3.3 Odvození parametrů filtrů

V této části kapitoly bude naznačen postup odvození jednotlivých parametrů filtrů. Pomocí těchto rovnic GUI vytváří přenosy jednotlivých filtrů.

3.3.1 Filtr typu dolní propust

Jedná se o filtr s přenosem (3.2). Na obr. 3.3 je schéma tvaru frekvenční charakteristiky s klíčovými parametry ω_1, ϵ a k . Význam jednotlivých parametrů byl popsán v úvodu kap. 3.1.



Obrázek 3.3. Tvar frekvenční charakteristiky filtru typu dolní propust

Přenos (3.2) upravíme do tvaru vhodného pro odvození.

$$W_2 = K_0 \cdot \frac{\omega_0^k}{(j\omega + \omega_0)^k}$$

Hodnotu parametru ϵ určíme z limity pro ω resp. s jdoucí k nule.

$$\epsilon = \lim_{j\omega \rightarrow 0} W_2 = \lim_{j\omega \rightarrow 0} K_0 \cdot \frac{\omega_0^k}{(j\omega + \omega_0)^k} = K_0$$

Pro další zjednodušení výpočtu zavedeme substituci

$$K_A = \sqrt[k]{K_0}.$$

Po aplikaci substituce dospějeme ke vztahu

$$W_2 = K_A^k \cdot \frac{\omega_0^k}{(j\omega + \omega_0)^k}.$$

Parametr ω_1 udává kmitočet, pro který má filtr jednotkové zesílení. Tedy úpravou základního vztahu substitucí $\omega = \omega_1$ a odstraněním mocniny dospějeme k rovnici

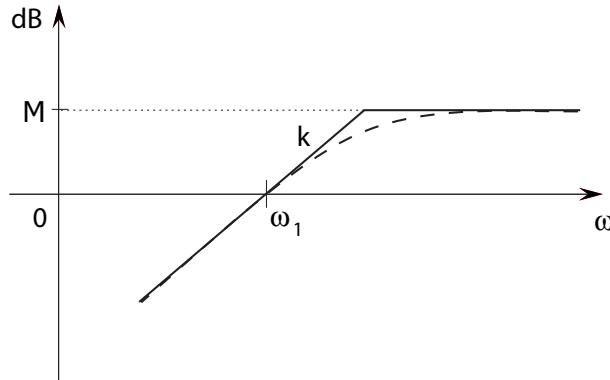
$$\frac{K_A \cdot \omega_0}{\sqrt{\omega_0^2 + \omega_1^2}} = 1.$$

Postupným řešením rovnice s neznámou ω_0 dospějeme k závěrečnému vztahu

$$\omega_0 = \frac{\omega_1}{\sqrt[k]{\sqrt{K_A^2} - 1}}.$$

3.3.2 Filtr typu horní propust

Jedná se o filtr s přenosem (3.3). Na obr. 3.4 je schéma tvaru frekvenční charakteristiky s klíčovými parametry ω_1, M a k . Při odvození jednotlivých parametrů budeme postupovat stejným postupem jako při filtru typu dolní propust.



Obrázek 3.4. Tvar frekvenční charakteristiky filtru typu horní propust

Přenos (3.3) upravíme do tvaru vhodného pro odvození

$$W_3 = K_0 \cdot \frac{(j\omega)^k \cdot \omega_0^k}{(j\omega + \omega_0)^k}.$$

Hodnotu parametru M určíme z limity pro ω jdoucí k nekonečnu.

$$M = \lim_{j\omega \rightarrow \infty} W_3 = \lim_{j\omega \rightarrow \infty} K_0 \cdot \frac{j\omega \cdot \omega_0^k}{(j\omega + \omega_0)^k} = K_0 \cdot \omega_0^k \quad (3.5)$$

Pro další zjednodušení výpočtu zavedu substituci

$$K_A = \sqrt[k]{K_0}.$$

Po aplikaci substituce dospějeme ke vztahu

$$W_3 = \frac{(K_A \cdot j\omega \cdot \omega_0)^k}{(j\omega + \omega_0)^k}.$$

Nyní přejdeme k výpočtu ω_0 z frekvence ω_1 .

$$|W_3(j\omega_1)| = 1$$

Po odstranění absolutní hodnoty dostáváme rovnici

$$\frac{K_A \cdot \omega_1 \cdot \omega_0}{\sqrt{\omega_0^2 + \omega_1^2}} = 1. \quad (3.6)$$

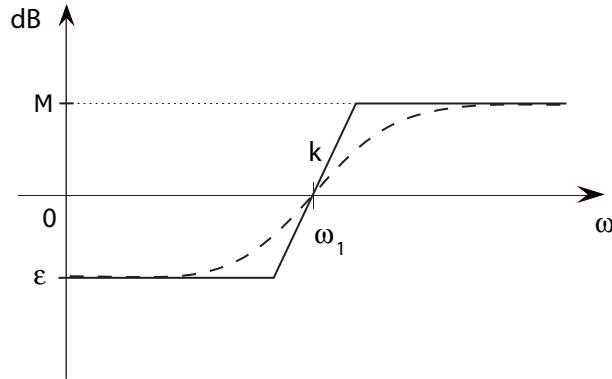
Postupným řešením soustavy dvou rovnic (3.5) a (3.6) o dvou neznámých dospějeme k závěrečným vztahům pro ω_0 a K_0 .

$$\omega_0 = \omega_1 \cdot \sqrt{(\sqrt[k]{M^2} - 1)}$$

$$K_0 = \frac{M}{\omega_0^k}$$

3.3.3 Filtr typu lead

Jedná se o filtr s přenosem (3.4). Na obr. 3.5 je schéma tvaru frekvenční charakteristiky s klíčovými parametry ω_1, M, ϵ a k .



Obrázek 3.5. Tvar frekvenční charakteristiky filtru typu lead

Přenos filtru bude funkce ve tvaru

$$W_4 = K_0 \cdot \frac{(s + \omega_2)^k}{(s + \omega_1)^k \cdot (\frac{\omega_2}{\omega_1})^k} = \left(\frac{as + b}{s + c} \right)^k,$$

kde

$$\frac{b}{a} = \omega_2, \quad c = \omega_1, \quad a = \frac{\omega_1}{\omega_2} \cdot \sqrt[k]{K_0}.$$

Parametr a:

$$M = \lim_{j\omega \rightarrow \infty} W_4|_{s=j\omega} = a^k$$

$$a = \sqrt[k]{M} \tag{3.7}$$

Parametr b:

$$\epsilon = \lim_{j\omega \rightarrow 0} W_4|_{s=j\omega} = \left(\frac{b}{c} \right)^k$$

$$b = c \cdot \sqrt[k]{\epsilon} \tag{3.8}$$

Parametr c:

$$\begin{aligned}|W_4|_{s=j\omega_1} &= 1 \\ \left| \left(\frac{j\omega_1 a + b}{j\omega_1 + c} \right)^k \right| &= 1 \\ \left| \left(\frac{j\omega_1 a + b}{j\omega_1 + c} \right) \right| &= 1 \\ \sqrt{\frac{a^2\omega_1^2 + b^2}{\omega_1^2 + c^2}} &= 1\end{aligned}$$

Řešením rovnice dospějeme ke vztahu

$$\begin{aligned}c &= \omega_1 \cdot \sqrt{\frac{a^2 - 1}{1 - \frac{b^2}{c^2}}} \\ c &= \omega_1 \cdot \sqrt{\frac{\sqrt[k]{M^2} - 1}{1 - \sqrt[k]{\epsilon^2}}}. \quad (3.9)\end{aligned}$$

Pomocí vztahů (3.7),(3.8) a (3.9) je již možné jednoduchým výpočtem určit přenos lead filtru s parametry ω_1, M, ϵ a k .

3.4 Programování v Matlabu

GUI bylo navrženo ve verzi Matlabu r13. Bylo však testováno a bude správně fungovat také i v novější verzi Matlabu r14. Pro samotné programování bylo použito prostředí GUIDE (GUI Design Environment) [1]. Program se spouští příkazem `filtry` z workspace Matlabu. Celý program je rozdělen do tří m-souborů. Hlavní část je v souboru `filtry.m`, dále 2 dialogová okna pro výpis upozornění a pro zadání proměnné. Ke svému běhu potřebuje program také obrázky náhledu jednotlivých funkcí v podadresáři `./obr`.

3.5 Příklad návrhu filtru

Při návrhu regulátorů pomocí \mathcal{H}_∞ metody jsou jako vstupní parametry požadovány váhové funkce reprezentující uživatelské požadavky na výstupní regulátor. GUI velmi jednoduchým způsobem umožňuje návrh těchto funkcí, zejména funkcí typu lead a lag,

kdy je velmi obtížné určit tvar přenosu tak, aby odpovídal zadaným požadavkům. Výhoda GUI spočívá také v tom, že GUI umí při návrhu velmi jednoduše přepínat mezi frekvencí zadanou v Hz nebo $\text{rad}\cdot\text{s}^{-1}$. Parametry pro výpočet přenosu si samo převádí do potřebného tvaru.

Jako příklad uvedu návrh filtru typu lead, což je typický filtr používaný pro stanovení míry neurčitosti modelu při návrhu regulátoru pomocí \mathcal{H}_∞ metody. V okně pro výběr filtru vybereme tedy příslušný filtr typu lead. Po zadání všech příslušných parametrů do GUI (ω_1 , ϵ , M , k), provedeme výpočet přenosu filtru stlačením tlačítka „Vypočti“. GUI zobrazí náhled bodeho charakteristiky a vpravo se objeví filtr v seznamu. Pokud bude filtr odpovídat námi zadaným požadavkům, můžeme filtr exportovat do prostředí Matlabu a dále jej používat pro návrh regulátoru.

Kapitola 4

Příklad použití GUI na reálném modelu

4.1 Úvod

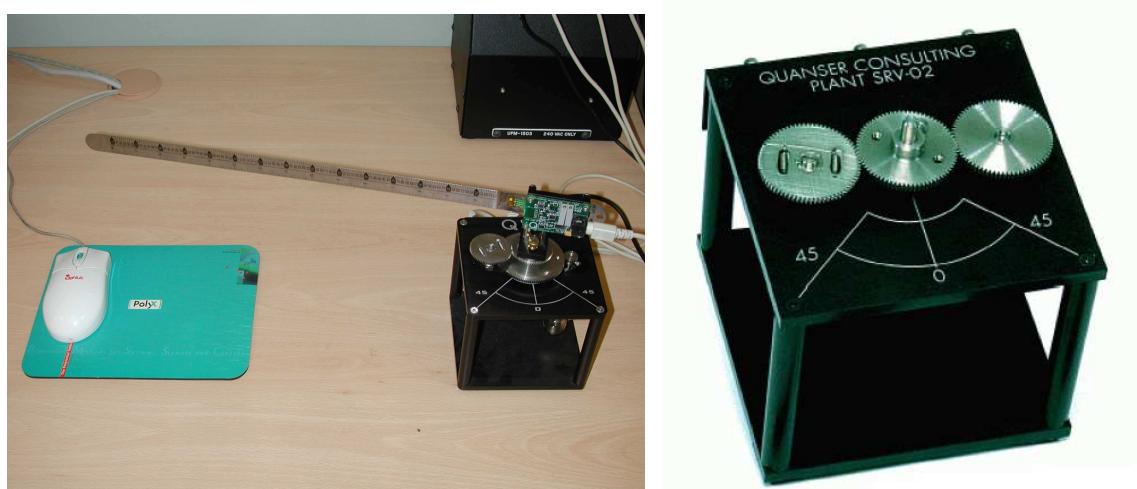
Použitelnost popsaného uživatelského rozhraní bude v této části demonstrována na praktickém příkladu návrhu regulátoru rychlosti pružného ramene, kterým otáčí servo ovládané prostřednictvím Simulinku z Matlabu. Cílem návrhu je vytvořit zpětnovazební systém, který zajistí (asymptotické) sledování skokových změn požadované rychlosti ramene a co největší odolnost vůči poruchám, které mohou na systém působit zvenčí.

Technicky je problém řízení simulinkem řešen pomocí programového vybavení od firmy QUANSER [2], která dodává všechny potřebné simulinkové knihovny pro práci s modelem. Simulinkový model je vždy automaticky před spuštěním přeložen přes externí překladač C++ kódu. Teprve tento kód je nahráván do řídicí karty, která již přímo pracuje s modelem.

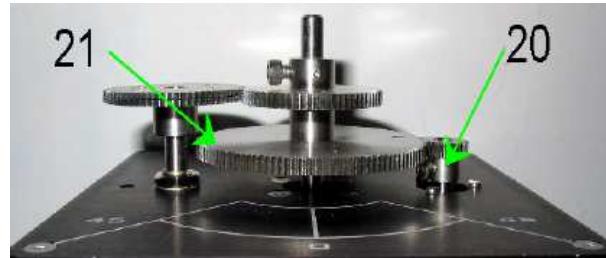
4.2 Model pružného ramene

Jak ukazuje obr. 4.1, model se skládá ze dvou částí. Pružného ramene a základny se servomotorem. Základna se skládá z DC motoru v pevném hliníkovém rámu. Motor je vybaven převodovkou. Výstup převodovky je napojen na vnější ozubená kola. Základna je vybavena potenciometrem pro měření výstupní úhlové polohy a tachometrem měřícím rychlosť otáček motoru. Soukolí na základně může být sestaveno ve dvěma způsoby. První

sestavení je patrné z obr. 4.1. Používá se pro řízení polohy a rychlosti bez připojeného dalšího modulu na výstup. V našem případě bylo použito sestavení druhé, jak naznačuje obr. 4.2. Na výstupní ozubené kolo (v obrázku označeno číslem 21) je připevněné samotné pružné rameno.



Obrázek 4.1. Vlevo soustava pružného ramene, vpravo základna modelu



Obrázek 4.2. Detail soukolí

4.3 Matematický model soustavy

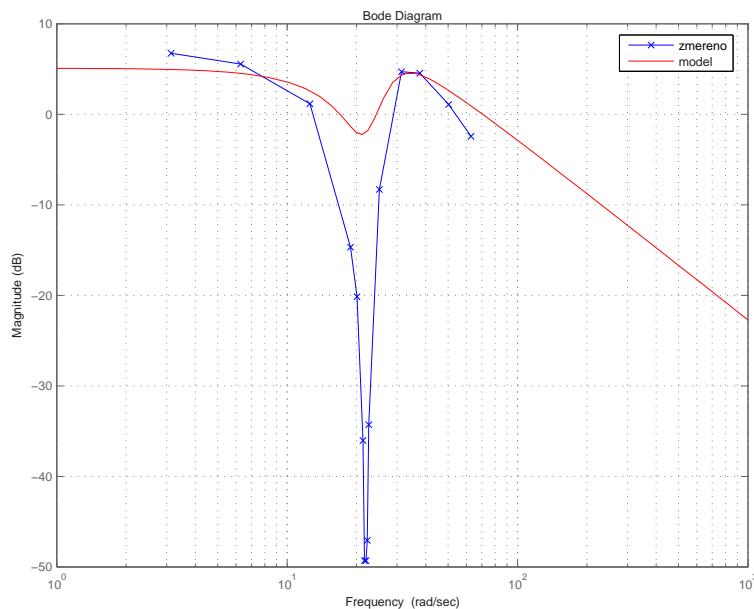
Model pružného ramene vychází z práce [5]. Aby nedošlo k poškození systému (kartáčů serva), šířka pásma regulačního obvodu nesmí přesáhnout 50 Hz. Proto i při práci v Simulinku je vhodné mít na výstupu dolnofrekvenční propust. Na frekvencích do 50 Hz lze odezvu rychlosti pružného ramena na budící napětí motoru zjednodušeně namodelovat

přenosem

$$G = \frac{-72,6(s^2 + 8s + 480)}{(s + 26)(s^2 + 18s + 750)}. \quad (4.1)$$

Na obr. 4.3 je zobrazena frekvenční charakteristika soustavy. Z obrázku je vidět porovnání charakteristiky modelu s reálnou soustavou.

Přibližně na frekvenci 3,5 Hz vykazuje řízený systém první rezonanční mód. Na vyšších kmitočtech již charakteristika není dostatečně přesná, a použitý jednoduchý model již nepopisuje soustavu dostatečně věrně.



Obrázek 4.3. Frekvenční charakteristika

4.4 Postup návrhu regulátoru

Matematicky lze problém návrhu (takového) řízení zformulovat ve frekvenční oblasti jako optimalizační úlohu hledání stabilizujícího regulátoru, který minimalizuje \mathcal{H}_∞ normu přenosové matice (2.2). V našem případě jednoduchého SISO modelu se matice (2.2) redukuje na hledání takového stabilizujícího regulátoru, který zajistí nejnižší maximální hodnotu součinů W_1S , W_2U a W_3T .

Jak již bylo uvedeno, na vyšších kmitočtech již jednoduchý model dostatečně přesně nepopisuje reálné chování modelu. Neurčitost modelu tak snižuje dosažitelnou šířku

pásma regulačního obvodu. Toto omezení je možné vyjádřit prostřednictvím váhového filtru W_3 s následujícími vlastnostmi. Na nízkých frekvencích bude mít zesílení $\epsilon = -4$ dB. To dá optimalizačnímu algoritmu prostor k návrhu zpětnovazebního systému s mírným překmitem doplňkové citlivosti T . Na frekvenci $\omega_1 = 1,5$ Hz protne 0 dB, tím se definuje neurčitost modelu a přibližně stanoví šířka pásma zpětnovazebního systému. Bude mít řád $k = 2$, který zaručí dostatečně rychlý pokles zesílení regulátoru na frekvencích blížících se 50 Hz. Na těchto kmitočtech je však tvar bodeho charakteristiky již nezajímavý. Proto s výhodou užijeme filtru typu lead. GUI pro tyto parametry vrátí váhový filtr

$$W_3 = 25000 \frac{(s + 5)^2}{(s + 1000)^2}.$$

Požadavek na asymptotické sledování konstantní reference odpovídá požadavku, aby citlivostní funkce S měla na nízkých frekvencích nulovou amplitudu. To je možné vyjádřit pomocí váhového filtru W_1 ve tvaru

$$W_1 = \frac{k}{s},$$

kde k je volitelná konstanta. Vzhledem k tomu, že nemáme další specifikaci na citlivostní funkci, zvolíme např. $k = 1$.

Velikost akčního zásahu, čili vstupního napětí motoru, nesmí přesáhnout 10 V. Omezíme-li se při regulaci na úhlové rychlosti do $180 \text{ deg}\cdot\text{s}^{-1}$ a vezmeme-li v úvahu převodní poměry vzorkovací karty, pak lze zaručit nezbytné omezení akčního zásahu pomocí váhového filtru W_2 s jednoduchým tvarem

$$W_2 = \frac{1}{3}.$$

Nyní již máme definované všechny potřebné parametry pro spuštění optimalizační úlohy. Pro řešení využijeme funkcí RCT Toolboxu v Matlabu, zejména funkce „`hinfopt`“. Obr. 4.4 ukazuje m-soubor provádějící řešení úlohy; obr. 4.5 potom výsledek zobrazený Matlabem. Funkce „`hinfopt`“ vrací parametr `gamma`, což je převrácená hodnota vážené \mathcal{H}_∞ normy, kterou funkce minimalizuje. Z tabulky na obr. 4.5 je možné vysledovat postup, jakým funkce „`hinfopt`“ provádí hledání regulátoru. Funkce nastaví počáteční hodnotu parametru `gamma` a snaží se jej metodou půlení intervalu co nejvíce zvýšit, přitom však regulátor musí zůstat stabilní (na obr. 4.5 sloupec C.L.).

```

G =-1.1*tf([1 8 480],[1 18 750])*tf([66.5941],[1 26.1]);
W1=tf([3],[1 0]);
W2=tf(1/3);
W3=zpk([-5 -5],[-1e3-1e3],1e3*10^(28/20));
TSS=augtf(G,W1,W2,W3);
[gamopt,C,sscl]=hinfopt(TSS);
L=series(C,G);
S=feedback(1,L);
T=feedback(L,1);
U=minreal(series(S,C));

```

Obrázek 4.4. M-soubor v Matlabu provádějící výpočet regulátoru

<< H-Infinity Optimal Control Synthesis >>

No	Gamma	D11<=1	P-Exist	P>=0	S-Exist	S>=0	lam(PS)<1	C.L.
<hr/>								
1	1.0000e+000	OK	OK	FAIL	OK	OK	OK	UNST
2	5.0000e-001	OK	OK	OK	OK	OK	OK	STAB
3	7.5000e-001	OK	OK	OK	OK	OK	OK	STAB
4	8.7500e-001	OK	OK	OK	OK	OK	OK	STAB
5	9.3750e-001	OK	OK	OK	OK	OK	OK	STAB
6	9.6875e-001	OK	OK	FAIL	OK	OK	OK	UNST
7	9.5313e-001	OK	OK	FAIL	OK	OK	OK	UNST
8	9.4531e-001	OK	OK	FAIL	OK	OK	OK	UNST

Iteration no. 5 is your best answer under the tolerance: 0.0100 .

Obrázek 4.5. Výpis v Matlabu po zavolání příkazu „hinfopt“

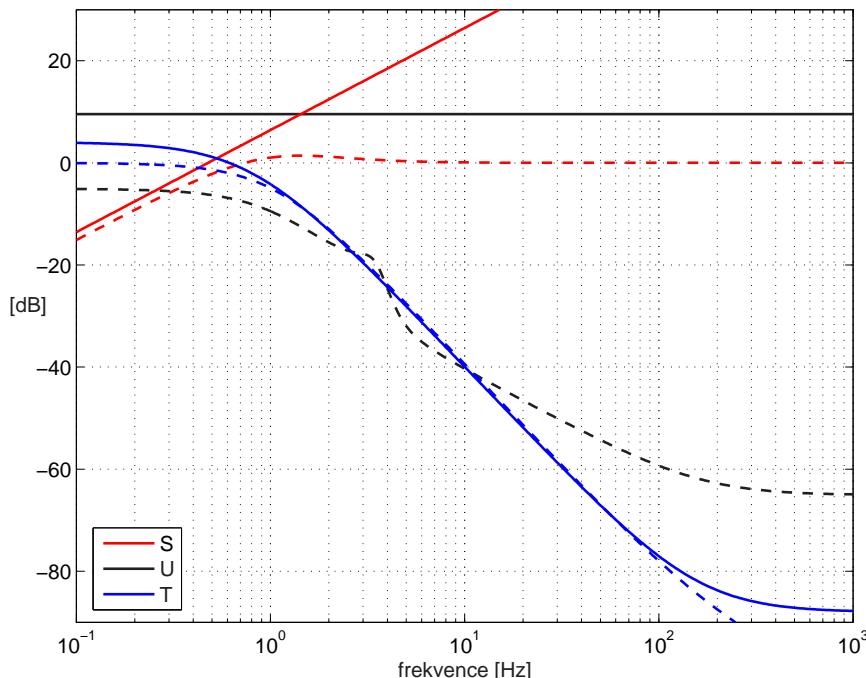
Výsledkem iterační úlohy je regulátor C s následujícím přenosem

$$C = -3079 \cdot \frac{(s + 1097)(s + 919)(s + 26,1)(s^2 + 18s + 750)}{s(s + 5,52 \cdot 10^6)(s + 969,8)(s + 11,89)(s^2 + 8,027s + 480,8)}. \quad (4.2)$$

Je zřejmé, že přenos (4.2) obsahuje některé nuly a póly, které jsou mimo potřebný frekvenční rozsah, a že tedy nebudou mít vliv na výsledek regulace. Tyto módy je možné z přenosu regulátoru odebrat bez újmy na kvalitě regulátoru. Tedy přenos regulátoru po úpravě bude

$$C = -0,622 \cdot \frac{(s + 26,1)(s^2 + 18s + 750)}{s(s + 11,92)(s^2 + 8s + 480)}.$$

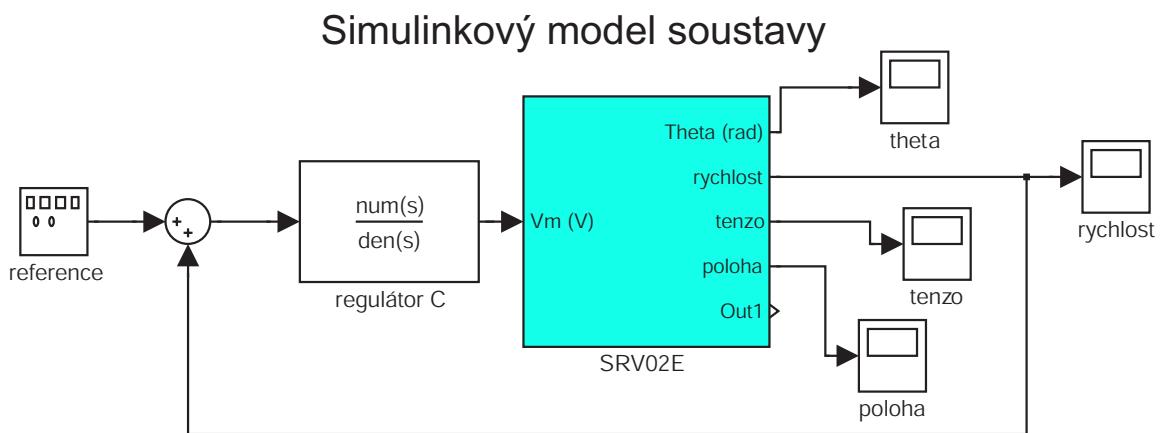
Obr. 4.6 přehledně zobrazuje výsledné citlivostní funkce.



Obrázek 4.6. Výsledek první iterace návrhu řízení – plnou čarou omezení definovaná váhovými filtry, čárkováně dosažené citlivostní funkce

4.5 Měření na reálném systému

Jak již bylo uvedeno výše, pro měření na reálném modelu bylo použito simulinkových modelů od firmy QUANSER [2]. Na obr. 4.7 je schéma vytvořeného modelu v prostředí Simulinku. Jedná se o klasickou zpětnovazební smyčku s regulátorem C .

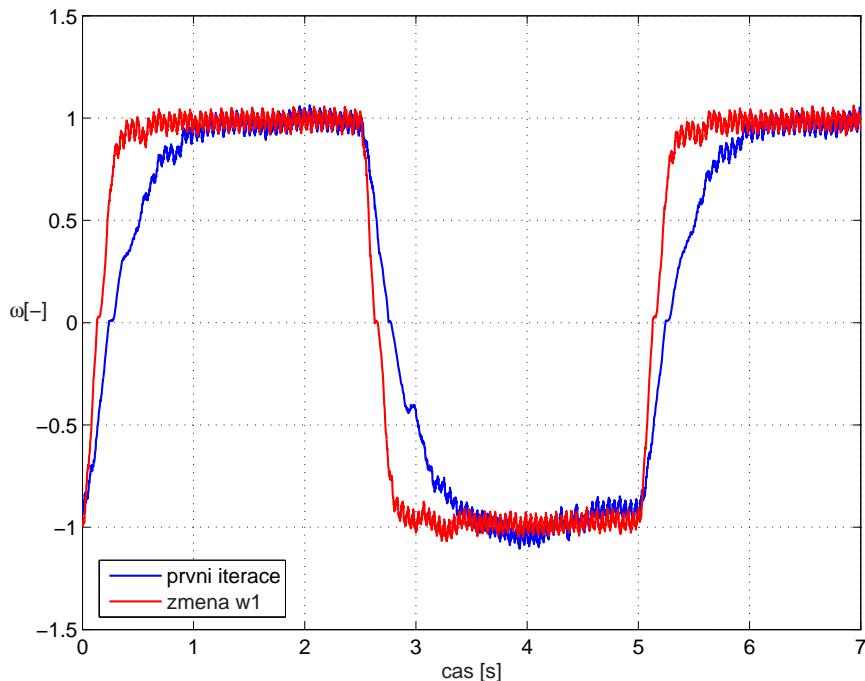


Obrázek 4.7. Simulinkový model

Měření na reálném systému (vizte obr. 4.8) ukázalo, že omezení šířky pásma vahou W_3 je poněkud konzervativní. Neurčitost modelu zřejmě v oblasti okolo první rezonanční frekvence systému nedosahuje velikosti dané zesílením filtru W_3 . Je proto možné zvětšit šířku pásma řízeného systému posunutím zlomové frekvence ω_1 váhy W_3 z 0,6 Hz až na 1,5 Hz. GUI vrátí pro změněnou hodnotu ω_1 přenos

$$W_3 = 17800 \cdot \frac{(s + 11,8)^2}{(s + 1980)^2}.$$

Jak je patrné z obr. 4.8 tato změna rozšířila pásmo regulačního obvodu, a tím zrychlila přechodový děj. Přitom nedošlo ke vzniku oscilací. To ukazuje, že ve zvoleném frekvenčním pásmu použitý model dostatečně přesně popisuje chování řízeného systému.



Obrázek 4.8. Naměřené přechodové charakteristiky

4.6 Výsledky

V této kapitole byl popsán samotný praktický postup při návrhu regulátoru pomocí výše popsané metody s využitím mnoha vyvinutého grafického rozhraní. Podařilo se navrhnut regulátor, který co nejvíce zrychlil přechodový děj, přitom však nezpůsobil překmit přechodové charakteristiky. Při měření přechodových charakteristik bylo zvláště důležité dávat pozor na přívodní kabely k modelu, které v případně špatného umístění ovlivňovaly naměřené průběhy.

Kapitola 5

Závěr

Cílem této bakalářské práce bylo vytvořit grafické uživatelské rozhraní, které by umožnilo snadný a pohodlný návrh frekvenčních filtrů, které jsou použity jako váhové funkce při navrhování řízení pomocí metody minimalizace \mathcal{H}_∞ normy zpětnovazebních citlivostních funkcí. Základní vlastnosti této metody byly popsány v kap. 2.

Ukázalo se, že při návrhu řízení se velmi často setkáváme s nutností neustále iteračně měnit tvar jednotlivých váhových funkcí. K tomuto účelu velmi dobře poslouží navržené GUI (podrobně popsané v kap. 3), které umožňuje jednoduchým způsobem měnit tvar frekvenční charakteristiky filtru, přitom však není vůbec nutné znát analytické vyjádření pro přenos filtru. To oceníme nejvíce při návrhu filtrů typu „lead“ a „lag“ s přenosy založenými na rovnici (3.4), neboť bez použití GUI není jednoduché rychlým způsobem vtělit uživatelem definované požadavky na filtr do analytického vyjádření přenosu. Při návrhu se například setkáváme s potřebou „posunovat“ charakteristiku filtru po frekvenční ose doprava či doleva. Tento posun GUI umožňuje velmi jednoduše změnou jednoho parametru.

Použití GUI bylo předvedeno na návrhu regulátoru rychlosti otáčení pružného ramene. Úspěšně se podařilo navrhnout regulátor, který co nejvíce urychlí přechodový děj.

Vytvořené grafické rozhraní bylo představeno v příspěvku s názvem „Uživatelské rozhraní pro návrh frekvenčních filtrů“ na konferenci Řízení procesů 2006 [3].

Literatura

- [1] Webová dokumentace firmy mathworks [online]. <http://www.mathworks.com>, 2005.
- [2] Webová dokumentace firmy quanser [online]. <http://www.quanser.com>, 2005.
- [3] ŘEZÁČ M., HOLUB O., AUGUSTA P., HURÁK Z. Uživatelské rozhraní pro návrh frekvenčních filtrů. Ve sborníku konference *Rízení procesů 2006. Kouty nad Desnou*. Pardubice: Fakulta chemicko-technologická, Univerzita Pardubice, 2005.
- [4] HAVLENA, V. *Moderní teorie řízení - Doplňkové skriptum*. Praha: FEL ČVUT, 2002.
<http://dce.felk.cvut.cz/mtr/cv/skripta/mtr.pdf>.
- [5] HOLUB, O. *Tool Development for VLT Telescope Controller Design*. Praha: Katedra řídicí techniky, FEL ČVUT, 2005.
- [6] ZHOU, K. *Essentials of Robust Control*. New Jersey: Prentice Hall, 1998.

Dodatek A

Obsah přiloženého CD

- zdrojové kódy ke grafickému rozhraní v Matlabu,
- tento text ve formátu PDF.

Dodatek B

Část programu

Kód přiložené funkce „Spocti_Prenos_Filtru“ provádí výpočet filtru podle zadaných parametrů. Následující funkce „Button_Vypocti_Callback“ reaguje na stlačení tlačítka Vypočti .

```
%funkce vracející přenos filtru podle zadaných parametrů
function Prenos=Spocti_Prenos_Filtru(typ,e,M,k,w1);
Prenos=tf([1],[1 1 1]);
s=tf('s');

switch typ
    case 'Dolní Propust'
        w0=w1/(sqrt(M^(2/k)-1));
        Prenos=M*w0^k/(s+w0)^k;
        return
    case 'Horní Propust'
        w0=w1*sqrt(M^(2/k)-1);
        k0=M/w0^k;
        Prenos=k0*s^k*w0^k/(s+w0)^k;
        return
    case 'Dolní Propust (aktivní)'
        w0=w1*sqrt((1-e^(2/k))/(e^(2/k)));
        k0=e*w0^k;
        Prenos=k0*(s+w0)^k/(s*w0)^k;
        return
```

```
case 'Horní Propust (aktivní)'  
w0=sqrt(e^(2/k)/(1-e^(2/k)))*w1;  
Prenos=e*(s+w0)^k/w0^k;  
return;  
case 'Lead'  
a=M^(1/k);  
c=sqrt((M^(2/k)-1)/(1-e^(2/k)))*w1;  
b=c*e^(1/k);  
Prenos=((a*s+b)/(s+c))^k;  
return  
case 'Lag'  
a=e^(1/k);  
c=sqrt((e^(2/k)-1)/(1-M^(2/k)))*w1;  
b=c*M^(1/k);  
Prenos=((a*s+b)/(s+c))^k;  
return  
case 'Integrator'  
Prenos=w1^k/s^k;  
return  
case 'Derivator'  
Prenos=s^k/w1^k;  
return  
end
```

```
% --- Executes on button press in Button_Vypocti.
function Button_Vypocti_Callback(hObject, eventdata, handles)
% hObject    handle to Button_Vypocti (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

epsilon1=-20;
if (strcmp(get((handles.edit_epsilon), 'Enable'), 'on'))
[epsilon1,ok]=str2num(get(handles.edit_epsilon, 'String'));
if (ok==0) warning_message('string', 'Chyba pri zadani "e".');
return; end
if (get(handles.jednotka_e, 'Value')==1)&&(epsilon1>=0),
warning_message('string', '"e" musi byt mensi nez 0dB.');
return; end
if (get(handles.jednotka_e, 'Value')==2)&&
((epsilon1<=0)|| (epsilon1>=1)),
warning_message('string', '"e" musi byt v intervalu (0,1).');
return; end
end

M1=20;
if (strcmp(get((handles.edit_M), 'Enable'), 'on'))
[M1,ok]=str2num(get(handles.edit_M, 'String'));
if (ok==0), warning_message('string', 'Chyba pri zadani "M".');
return; end
if (get(handles.jednotka_M, 'Value')==1)&&(M1<=0),
warning_message('string', '"M" musi byt vetsi nez 0dB.');
return; end
if (get(handles.jednotka_M, 'Value')==2)&&(M1<=1),
warning_message('string', '"M" musi byt vetsi nez 1.');
return; end
end

k=1;
```

```

if (strcmp(get((handles.edit_k), 'Enable'), 'on'))
    [k,ok]=str2num(get(handles.edit_k, 'String'));
    if (ok==0) warning_message('string', 'Chyba pri zadani "k".');
        return; end
end

w1=100;
if (strcmp(get((handles.edit_w1), 'Enable'), 'on'))
    [w1,ok]=str2num(get(handles.edit_w1, 'String'));
    if (ok==0) warning_message('string', 'Chyba pri zadani "w1".');
        return; end
    if (w1<0) warning_message('string',
        'Chyba ve frekvenci - zaporny kmitocet.');
        return; end
    if (get(handles.jednotka_w, 'Value')==2), w1=w1*2*pi; end
end

M=M1;epsilon=epsilon1;
if (get(handles.jednotka_M, 'Value')==1), M=10^(M1/20); end
if (get(handles.jednotka_e, 'Value')==1), epsilon=10^(epsilon1/20); end

style=get(handles.typ_filtru, 'Value');
typy=get(handles.typ_filtru, 'String');
typ=typy{style};

F=Spocti_Prenos_Filtru(typ,epsilon,M,k,w1);

handles.last=F;
handles.Filtr{length(handles.Filtr)+1}=F;

epstext='';Mtext='';
if (strcmp(get((handles.edit_epsilon), 'Enable'), 'on')),
    epstext=strcat(char(' e='),num2str(epsilon1));end
if (strcmp(get((handles.edit_M), 'Enable'), 'on')),
    Mtext=strcat(char(' M='),num2str(M1));end

```

```
pomoc=strcat(typ,Mtext,epstext,char(' k='),
               num2str(k),char(' w1='),num2str(w1));

list = get(handles.List_filtru,'String');
[rows cols]=size(list);

if rows>0
    puvodni=get(handles.List_filtru,'String');
    set(handles.List_filtru,'String',strvcat(char(puvodni),char(pomoc)));
    set(handles.List_filtru,'Value',rows+1);
else
    set(handles.List_filtru,'String',char(pomoc));
    set(handles.List_filtru,'Value',1);
end

[mag,phase,w]=bode(F);
axes(handles.axes1);
semilogx(w,20*log10(mag(:,:,1)));
grid on;

guidata(hObject, handles);
```