

Využití specializovaných toolboxů v robustním řízení

Application of Specialized Toolboxes in Robust Control

Bc. Josef Slovák

Diplomová práce
2012



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2011/2012

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Josef SLOVÁK**
Osobní číslo: **A10449**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Automatické řízení a informatika**

Téma práce: **Využití specializovaných toolboxů v robustním řízení**

Zásady pro vypracování:

1. Definujte základní pojmy a myšlenky, související s problematikou robustního řízení.
2. Seznamte se s toolboxy pro Matlab, které jsou využitelné v robustním řízení (Robust Control Toolbox, Control System Toolbox, Polynomial Toolbox, YALMIP, SeDuMi, atd.).
3. Popište základy práce s těmito programovými balíky.
4. U vybraných toolboxů demonstруйте na vhodně zvolených příkladech některé jejich funkce.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **BARMISH, Ross B.** New Tools for Robustness of Linear Systems. New York: Macmillan Publishers, 1994. ISBN 978002306055.
2. **BOYD, S., L. EL GHAOU, E. FERON a V. BALAKRISHNAN.** Linear Matrix Inequalities in System and Control Theory. Philadelphia: Society for Industrial and Applied Mathematics, 1994. SIAM studies in applied mathematics, 15. ISBN 0-89871-334-X.
3. **DOSTÁL, P. a R. MATUŠŮ.** Stavová a algebraická teorie řízení. Zlín: Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2010. ISBN 978-80-7318-991-4.
4. **LÖFBERG, J. YALMIP Wiki.** [online]. Dostupné z: <http://users.isy.liu.se/johanl/yalmip/>
5. **Robust Control Toolbox – Documentation.** THE MATHWORKS, Inc. Mathworks [online]. Dostupné z: <http://www.mathworks.com/help/toolbox/robust/>
6. **ŠEBEK, M., M. HROMČÍK a J. JEŽEK.** Polynomial toolbox 2.5 and systems with parametric uncertainties. In: Proceedings of 3rd IFAC Symposium on Robust Control Design. Prague, 2000.

Vedoucí diplomové práce:

Ing. Radek Matušů, Ph.D.

Ústav automatizace a řídicí techniky


Datum zadání diplomové práce:

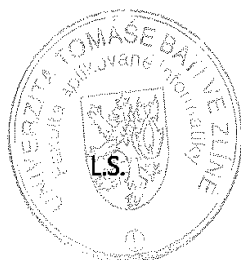
24. února 2012

Termín odevzdání diplomové práce:

6. června 2012

Ve Zlíně dne 24. února 2012


prof. Ing. Vladimír Vašek, CSc.
děkan




prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Diplomová práce se věnuje základním pojmům a myšlenkám robustního řízení, klasifikaci jednotlivých typů neurčitostí a popisu analytických a grafických nástrojů pro analýzu robustní stability systémů. Praktická část je koncipována jako přehled nejpoužívanějších toolboxů pro robustní řízení pro programové prostředí Matlab. Jsou v ní probírány ty nejpoužívanější a nejsložitější toolboxy a jejich možnosti na vhodně zvolených příkladech na robustní řízení.

Klíčová slova: robustní řízení, toolbox, parametrická neurčitost, neparametrická neurčitost, analýza robustní stability, Polynomial Toolbox, Robust Control Toolbox

ABSTRACT

Diploma thesis gives attention to the elementary notions and thoughts of robust control, classification of each individual types of uncertainties and description of analytic and graphic tools for a solution in the analysis of a robust stability of systems. The practical part is conceived as a summary of the most common toolboxes used for robust control in programming environment Matlab. It discusses the most widely used and sophisticated toolboxes and their abilities on appropriately selected examples in robust control.

Keywords: robust control, toolbox, parametric uncertainty, non-parametric uncertainty, robust stability analysis, Polynomial Toolbox, Robust Control Toolbox

Děkuji panu Ing. Radkovi Matušů, Ph.D., že byl ochotný vést mojí diplomovou práci, za jeho pomoc při řešení problémů, které se naskytly během práce, a za poskytnutí materiálů.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 ZÁKLADNÍ POJMY	11
1.1 NEURČITOST.....	11
1.2 ROBUSTNOST.....	12
2 PŘEHLED ZÁKLADNÍCH PROBLÉMŮ A METOD V ROBUSTNÍM ŘÍZENÍ	14
2.1 NORMY SIGNÁLŮ A SYSTÉMŮ	14
2.1.1 Normy pro signály.....	14
2.1.2 Normy pro přenosy	15
2.2 CITLIVOSTNÍ FUNKCE	15
2.3 AMPLITUDOVÁ A FÁZOVÁ BEZPEČNOST.....	16
2.4 KVALITA REGULACE (NOMINÁLNÍ CHOVÁNÍ)	17
2.5 NEPARAMETRICKÉ (NESTRUKTUROVANÉ) NEURČITOSTI	18
2.5.1 Robustní stabilita.....	19
2.5.2 Robustní kvalita regulace (robustní chování)	20
2.6 PARAMETRICKÉ NEURČITOSTI	21
2.6.1 Jednparametrická neurčitost.....	21
2.6.2 Intervalová neurčitost (nezávislá)	22
2.6.3 Afinity lineární neurčitost	22
2.6.4 Multilineární neurčitost	23
2.6.5 Polynomiální neurčitost	23
2.6.6 Obecná neurčitost.....	23
2.7 METODY ANALÝZY PARAMETRICKÝCH NEURČITOSTÍ.....	24
2.7.1 Bialasova věta (Bialas eigenvalue criterion).....	24
2.7.2 Množina hodnot (The Value Set).....	24
2.7.3 Věta o vyloučení nuly (Zero Exclusion Condition)	25
2.7.4 Kharitonovova věta	26
2.7.5 Konvexní množina a konvexní obal.....	26
2.7.6 Polytopy	28
2.7.7 Polytopy polynomů a jejich množina hodnot.....	28
2.7.8 Věta o hranách.....	28
2.7.9 Překrytí složitější neurčitosti intervalovou neurčitostí.....	28
2.8 NÁVRH ŘÍZENÍ PRO PARAMETRICKÉ NEURČITOSTI.....	29
2.9 LINEÁRNÍ Maticová NEROVNOST (LMI).....	30
2.9.1 Analýza a syntéza pomocí LMI	31
II PRAKTICKÁ ČÁST	33
3 PROGRAMOVÉ PROSTŘEDÍ MATLAB	34

3.1	VÝPOČETNÍ JÁDRO	34
3.2	GRAFICKÝ SUBSYSTÉM.....	35
3.3	OTEVŘENÁ ARCHITEKTURA.....	35
3.4	PRACOVNÍ NÁSTROJE.....	36
3.5	TOOLBOXY	36
4	CHARAKTERISTIKA VYBRANÝCH TOOLBOXŮ.....	38
4.1	CONTROL SYSTEM TOOLBOX	38
4.2	ROBUST CONTROL TOOLBOX	38
4.2.1	Robustnost Servo regulátoru pro stejnosměrný motor.....	42
4.3	POLYNOMIAL TOOLBOX	51
4.3.1	Příklady využití Polynomial Toolboxu	52
4.4	YALMIP.....	61
4.5	OSTATNÍ TOOLBOXY PRO ROBUSTNÍ ŘÍZENÍ A SIMULINK.....	63
	ZÁVĚR	65
	ZÁVĚR V ANGLIČTINĚ.....	66
	SEZNAM POUŽITÉ LITERATURY.....	67
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	69
	SEZNAM OBRÁZKŮ	70

ÚVOD

Diplomová práce je koncipována jako přehledová práce o specializovaných toolboxech využívaných pro robustní řízení, proto se i teoretická část snaží obsáhnout co největší záběr důležitých pojmů na poli robustního řízení.

V teoretické části jsou vysvětleny základní pojmy robustnosti a neurčitosti systému. Je definována robustní stabilita a metody její analýzy pro parametrické a neparametrické neurčitosti. Více je kladen důraz na parametrickou neurčitost z toho důvodu, že v praktické části jsou jednotlivé toolboxy demonstrovány na příkladech s parametrickou neurčitostí. Jsou definovány základní myšlenky u jednotlivých typů parametrických neurčitostí (např. jednoparametrová, afinní lineární) a metody jejich analýzy (např. věta o vyloučení nuly). Samozřejmě teoretická část neopomíjí ani neparametrickou neurčitost. U tohoto typu neurčitosti jsou definovány základní modely pro popis této neurčitosti a nastíněny normy pro signály a systémy, které se s oblibou využívají u těchto neurčitostí.

Praktická část se snaží vytvořit přehled o existujících specializovaných toolboxech pro robustní řízení. Podrobně jsou představeny komerční toolboxy Robust Control Toolbox od firmy Mathworks a Polynomial Toolbox od firmy Polyx. Možnosti těchto toolboxů jsou prezentovány na vhodně zvolených příkladech. Avšak není zapomenuto ani na volně šiřitelné a méně vyspělé toolboxy jakým je např. YALMIP.

V posledních několika letech se pojem robustního řízení velmi rozmáhá v oblasti komplexní proměnné. Velkým vývojem právě prochází například analýza robustní stability systémů s neurčitostmi, jejichž hlavní nástroje jsou popisovány mnohými autory [1], [2]. Jedním z obrovských úspěchů v analýze robustní stability je objev Kharitonovovy věty [1], [2], [11], která vyřešila elegantním způsobem testování robustní stability systémů s intervalovou neurčitostí, kdy bez ohledu na počet neurčitých parametrů se testují jen čtyři specifické polynomy. Pole robustního řízení se zabývá čím dál více složitými neurčitostmi, kdy analýza robustní stability je řešena už jen pouze díky velmi univerzálním nástrojům jako vykreslení množiny hodnot a aplikaci věty o vyloučení nuly [2], [8].

I. TEORETICKÁ ČÁST

1 ZÁKLADNÍ POJMY

1.1 Neurčitost

Pro návrh řízení reálných procesů v praxi se využívá myšlenka náhradního ideálního matematického modelu. Na této myšlence stojí v podstatě celá klasická teorie řízení včetně mnoha moderních způsobů návrhu regulátorů. Při převodu na náhradní matematický model avšak nastává problém, že uvažovaný ideální model se prakticky nikdy neshoduje s chováním reálného systému. Tento závažný problém může zapříčínovat mnoho faktorů. Jedním z důvodů může být, že fyzikální parametry řízeného procesu nejsou známy nebo se mění v čase. Dále se může vyskytnout nesoulad při lineární aproximaci nelineárního systému. Anebo pokud potřebujeme pro jednodušší výpočty zjednodušit model (např. zanedbáním rychlých dynamických jevů z důvodu snížení řádu systému), a tím se nám vyskytne odlišnost od původního komplexního systému.

Pro zohlednění těchto odlišností od původního systému musíme do matematického modelu zavést tzv. neurčitost neboli neurčitý model. Jinak řečeno, budeme definovat celou třídu modelů, které budou v jeho okolí, místo jednoho nominálního modelu. Velikost tohoto okolí pak lze popsat dvěma základními způsoby, pomocí parametrické a neparametrické neurčitosti. Vyskytnout se může i kombinace obou hlavních metod, která vede k tzv. neurčitosti smíšené. [5]

Reálná parametrická neurčitost je konkrétní fyzikální parametr v modelu systému (např.: hmotnost nákladu osobního automobilu) reprezentovaná intervalem, který je určen svými krajními hodnotami minimem a maximem, mezi kterými se skutečná hodnota parametru může měnit.

Pro označení a popis systémů s parametrickými neurčitostmi se často využívá *vektor neurčitých parametrů* q , který patří do množiny reálných čísel. Zjednodušeně se tento vektor nazývá neurčitostí. Pokud bude vektor k -rozměrný, potom můžeme q zapsat jako k -tici:

$$q = (q_1, q_2, \dots, q_k) \in Q \subset R^k \quad (1)$$

Pro zdůraznění závislosti proměnné q , vkládáme ji jako argument funkce. Např. pro vyjádření přenosové funkce s neurčitými parametry napíšeme $P(s, q)$ místo $P(s)$. Poté neurčitý systém bude popsán jako:

$$P(s, q) = \frac{n(s, q)}{d(s, q)} \quad (2)$$

kde $n(s, q)$ a $d(s, q)$ jsou polynomy, jejichž koeficienty závisejí na q . [1]

Neparametrické neurčitosti oproti parametrickým neurčitostem do modelu systému přináší kompletní neurčitou dynamiku, která je charakterizována svojí amplitudovou-frekvenční charakteristikou, např. frekvenční závislost akčních členů a jejich saturace.

1.2 Robustnost

Robustnost znamená, že určitou vlastnost regulačního obvodu, kterou regulátor zajišťuje pro jednu nominální soustavu, zajistí zároveň pro celou třídu soustav danou jejím okolím (neurčitým modelem).

Jinými slovy, regulátor sice může být navržen pro nominální soustavu, ale protože skutečně řízený objekt není přesně znám, regulátor musí zaručovat, aby požadavky kladené na regulační obvod byly splněny nejen pro tuto nominální, ale také pro všechny soustavy z okolí. „Nějakou“ robustnost budou poskytovat i „klasicky“ navržené regulátory, avšak robustní řízení jako takové explicitně a systematicky bere v úvahu neurčitost obsaženou v matematickém modelu již během návrhu řízení. [5]

Kromě vymezení přípustného okolí nominální soustavy je podle definice třeba vztáhnout robustnost na jednu konkrétní vlastnost regulačního obvodu. Jde o vlastnost spíše kvalitativní než kvantitativní. Těžko si lze představit, že jeden regulátor zajistí minimální hodnotu kritéria optimality nebo dané rozmístění pólů pro celou třídu soustav. Ale může zachovat stabilitu, zajistit stálou nulovou regulační odchylku anebo udržet hodnotu kritéria optimality pod danou mezí.

Robustnost je neodmyslitelně spojena se dvěma pojmy: modelem neurčitosti regulované soustavy a vlastností regulačního obvodu, která má být robustní. To bývá nejčastěji stabilita. Robustní stabilita znamená více než současně stabilizovat určitý počet soustav v okolí navržené soustavy, znamená stabilizovat celé okolí, tedy nekonečně mnoho soustav.

Dalším důležitým pojmem na poli robustní stability je tzv. rodina polynomů. Tento pojem se vyskytuje v souvislosti s charakteristickým polynomem uzavřeného regulačního obvodu. Tato rodina polynomů se obecně skládá z neurčitého polynomu a z množiny omezujících parametry:

$$P = \{p(s, q) : q \in Q\} \quad (3)$$

Rodina polynomů je pak robustně stabilní právě tehdy, když $p(s, q)$ je stabilní pro každé $q \in Q$. [8]

2 PŘEHLED ZÁKLADNÍCH PROBLÉMŮ A METOD V ROBUSTNÍM ŘÍZENÍ

V oblasti robustního řízení existuje mnoho způsobů a variací pro dělení systémů a metod analýzy a syntézy. V následujících kapitolách si popíšeme velmi stručně malý zlomek těchto pojmů a metod. Nejvíce se budeme zabývat systémy s parametrickou neurčitostí a metodami jejich analýzy z důvodu, že většina příkladů v praktické části je právě směřována na tento typ neurčitosti.

2.1 Normy signálů a systémů

Již víme, že pokud není možné přiřadit neurčitost k jednotlivým parametrům systému nebo nám to z nějakého důvodu ani nevyhovuje, pak můžeme využít k popisu tzv. *neparametrické neurčitosti*. Pro kvantifikaci neparametrické neurčitosti se s výhodou používají tzv. *normy*. Obecně nás při hodnocení chování systému pomocí norem zajímají otázky typu „Jak velký bude výstup, když vstup bude...“ [20]. Existují normy jak pro signály (chování systému lze ohodnotit pomocí norem signálů, které nás u tohoto systému zajímají), tak také pro přenosy systémů.

2.1.1 Normy pro signály

Chování systémů můžeme popsat pomocí „velikostí“ signálů, které nás v těchto systémech „zajímají“. Signály mohou být skutečné i umělé. K vyjádření „velikostí“ signálů používáme tyto normy. Druh normy záleží na řešeném problému. Signály uvažujeme reálné, po částech spojitě a mohou začínat v čase 0.

Nebudeme si zde definovat všechny tyto normy, ale jen si řekněme, že existují a musejí mít nějaké specificky definované vlastnosti. [5]

Mezi nejpoužívanější normy patří *norma* L_1 (*absolutní norma*), *norma* L_2 (*kvadratická norma*), *norma* L_∞ . Jen pro ilustraci, např. *kvadratická norma* má význam energie. Je-li konečná, signál má konečnou energii. Pokud je konečná norma L_∞ , znamená to, že signál má omezenou amplitudu.

Určení normy signálu nám může sloužit k vyhodnocení kvality regulace.

2.1.2 Normy pro přenosy

Pro oblast teorie řízení jsou velmi důležité a používané normy H_2 (kvadratická norma) a H_∞ . Nebudeme si zde definovat jejich vzorce pro jejich výpočet, více informací poskytuje publikace [21], [20]. Normy však lze vypočítat jak z vnějšího, tak také na základě stavového popisu. Pro praktické vyčíslování norem je samozřejmě vhodné využít funkcí Matlabu a jeho toolboxů (např. příkazy *norm*, *h2norm*, *hinfnorm*). Výpočet normy podle definice, zejména pro složitější příklady, není příliš vhodný. Využívají se alternativní způsoby výpočtu (např. pomocí residuí pro H_2 či graficky z bodeho diagramu pro H_∞). Jen si zde uvedme jejich základní vlastnosti:

Pro normu H_2 je racionální funkce konečná na tehdy a jen tehdy, když $G(s)$ je striktně ryzí a nemá žádný pól na imaginární ose (tedy $G(j\omega)$ je omezená).

Pro normu H_∞ je racionální funkce konečná tehdy a jen tehdy, když $G(s)$ je ryzí a nemá žádný pól na imaginární ose. Tato norma je submultiplikativní.

2.2 Citlivostní funkce

V této kapitole si, pro pozdější potřeby metod syntézy, objasníme pár důležitých pojmů.

Těmito pojmy jsou *citlivostní funkce* (*citlivost*, *sensitivity*) S a *komplementární citlivostní funkce* (*komplementární citlivost*, *complementary sensitivity*) T .

Citlivostní funkce S vyjadřuje redukci citlivosti, kterou dosáhneme pomocí zpětné vazby.

Komplementární citlivostní funkce T je shodná s přenosem řízení G_{wy} , čili nejdůležitějším přenosem celého uzavřeného regulačního obvodu.

To znamená, že S je citlivost relativní změny přenosu řízení (resp. T) na relativní změnu modelu řízeného systému. Bohužel, lze snadno ověřit, že platí:

$$S(s) + T(s) = 1 \quad (4)$$

což je jedno ze základních omezení při návrhu zpětnovazebního regulačního obvodu. Vzhledem k tomu, jakým vnitřním přenosům ve zpětnovazebním systému odpovídají funkce citlivosti S a komplementární citlivosti T je zřejmé, že například nelze navrhnout regulátor, který by současně zajistil obvod necitlivý vůči poruše v a vůči referenčnímu signálu w . V praxi je tedy nutné volit priority pro jednotlivé frekvenční rozsahy. Tato úloha se nazývá *tvarování frekvenční charakteristiky* neboli *loop shaping*.

Při klasickém *loop shapingu* tvarujeme velikost frekvenční charakteristiky otevřené smyčky $L(j\omega) = Q(j\omega)G(j\omega)$. Do tohoto procesu zpravidla není zahrnuta žádná optimalizace, ale pouze jistá základní pravidla a naše zkušenosti, přičemž cílem je získat $L(j\omega)$ tak, aby měla požadovanou šířku přenášeného pásma, nepřekročila rezonanční převýšení, atd.

Návrh je transparentní a přímočarý, protože je zřejmé, jak změna regulátoru $Q(s)$ ovlivňuje $L(s)$ a naopak. Na druhou stranu ale tento postup nebere přímo v úvahu přenosy uzavřené smyčky. Proto se využívá také možnosti tvarování přímo funkcí $S(s)$, $T(s)$, případně $Q(s)S(s)$ [20].

2.3 Amplitudová a fázová bezpečnost

S tvarováním frekvenční charakteristiky také velmi úzce souvisejí pojmy *amplitudová bezpečnost* a *fázová bezpečnost stability*, které vycházejí z *Nyquistova kritéria stability*. Připomeňme, že toto kritérium říká „Nechť přenos otevřeného regulační smyčky $L(s)$ má n nestabilních pólů. Uzavřený regulační obvod je pak stabilní právě když amplitudově-fázová frekvenční charakteristika (Nyquistova křivka) otevřeného regulačního obvodu $L(j\omega)$, vykreslená pro $\omega \in (0, \infty)$, obíhá v komplexní rovině bod $[-1, 0j]$ v kladném smyslu (proti směru hodinových ručiček) $n/2$ krát (jeden oběh se rozumí 2π)“. Nyní můžeme definovat dva zmíněné pojmy.

Amplitudová bezpečnost G_m (*gain margin*) je hodnota, kterou lze násobit zesílení otevřeného systému, aby se uzavřený systém ocitl na hranici stability.

Fázová bezpečnost P_m (*phase margin*) je úhel, o který lze natočit frekvenční charakteristiku otevřeného systému, aby se uzavřený systém ocitl na hranici stability (jinými slovy, kolik fázového zpoždění můžeme přidat). Jedná se např. o ochranu před neurčitostí dopravního zpoždění.

Slabinou klasické amplitudové a fázové bezpečnosti je jejich vzájemná oddělenost, tj. že tyto pojmy stanovují míru bezpečnosti zvlášť pouze v amplitudě či pouze ve fázi. Často však může nastat případ neurčitosti v zesílení i fázi systému současně. Z tohoto důvodu se jako vhodnější jeví myšlenka použít „pouze“ vzdálenost frekvenční charakteristiky od kritického bodu $[-1, 0j]$, jak uvidíme později.

2.4 Kvalita regulace (nominální chování)

Naše požadavky na chování a robustnost regulačního obvodu můžeme formulovat také pomocí maximálních amplitud (peaků) $M_S = \|S\|_\infty$ a $M_T = \|T\|_\infty$. Typickými požadavky praxe jsou $M_S < 2$ (tj. $6dB$) a $M_T < 1.25$ (tj. $2dB$). Vysoké hodnoty (cca 4 a více) nám naznačují špatné chování a rovněž špatnou robustnost. Z rovnosti (4) lze dokázat, že M_S a M_T se od sebe mohou lišit maximálně o 1, čili pokud je „velké“ M_S , bude „velké“ i M_T a opačně. Ve [20] je ukázáno, že požadavek $M_S < 2$ (tj. $6dB$) vlastně znamená totéž jako $G_m > 6 \sim 9dB$ a $P_m > 30 \sim 45^\circ$. Jak již bylo nastíněno, citlivostní funkce S je velmi dobrým ukazatelem chování uzavřené regulační smyčky. V ideálním případě požadujeme, aby S bylo malé a proto nám stačí zabývat se pouze jeho amplitudou $|S|$, přičemž fáze nás nemusí trápit. Obecně existuje několik typických požadavků na S , jako například minimální šířka pásma, maximální regulační odchylka na vybraných frekvencích a v ustáleném stavu, tvar S na vybraných frekvenčních rozsazích či již zmíněná maximální amplituda (peak). Typicky je $|S|$ malé na nízkých frekvencích, ale z důvodu striktní ryzosti reálných systémů se na vysokých frekvencích dostává k 1. Na středních frekvencích je často prakticky nemožné zabránit vrcholu většímu než 1, což znamená, že v této oblasti se zhoršuje chování regulačního obvodu. Tedy, omezení vrcholu, představující míru nejvyšší degradace, zabraňuje zesilování šumu na středních a vyšších frekvencích a také nám zavádí míru robustnosti.

Matematicky můžeme tyto požadavky formulovat pomocí horní hranice $1/|W_p(s)|$, kterou omezíme S . Váha $W_p(s)$ je volena uživatelem (index „ P “ je zde odvozen od *performance*, neboť S je používána zejména jako indikátor chování neboli kvality regulačního pochodu). Podmínku kvality regulace (podmínku nominálního chování) lze tedy odvodit pomocí normy H_∞ :

$$\|W_p(s)S(s)\| < 1 \quad (5)$$

Podmínka kvality regulace (5) nám tedy vlastně říká, že vzdálenost frekvenční charakteristiky (v komplexní rovině) otevřeného regulačního obvodu od bodu $[-1, 0j]$ musí být větší než absolutní hodnota váhové funkce $W_p(j\omega)$ pro všechny frekvence ω . Dále je

také vhodné poznamenat, že při praktickém návrhu se spíše setkáme s frekvenčně závislou váhovou funkcí $W_p(j\omega)$, kterou můžeme zvolit např. podle doporučení v [20], [21].

2.5 Neparаметrické (nestrukturované) neurčitosti

Základní definicí této neurčitosti je, že soustava je neurčitá, ale „nevíme přesně kde a proč“, tj. nedokážeme nebo nechceme neurčitost přiřadit jednotlivým parametrům.

Důvodu proč to nechceme nebo nedokážeme, může být hned několik:

- zanedbání rozložených parametrů,
- zanedbání nelinearity (linearizme v pracovním bodě),
- zanedbání dynamiky vyššího řádu (např. parazitní kapacity a indukčnosti u elektrických systémů),
- přítomnost vnější poruchy, o které toho moc nevíme.

Pro neparаметrickou neurčitost existuje několik dílčích typů neurčitých modelů. Nejčastěji rozlišujeme neurčitosti *multiplikativní* a *aditivní*, méně často pak také jejich inverzní (zpětnovazební) verze, které umožňují popsat i nestabilní dynamiku.

Uvedeme si zde jen pro ilustraci rovnici *multiplikativního modelu*:

$$G(s) = [1 + W_M(s)\Delta_M(s)]G_0(s) \quad (6)$$

Pro MIMO systémy můžeme u multiplikativní neurčitosti dále formálně rozlišovat, zda se nachází na vstupu či na výstupu nominálního systému. U SISO systémů na tom však nezáleží, protože dojde pouze k přehození činitelů při násobení.

Přenos $G_0(s)$ v předcházejících rovnicích a obrázcích reprezentuje nominální soustavu a $G(s)$ skutečnou (perturovanou) soustavu. $W(s)$ je váhová funkce (stabilní) představující dynamiku neurčitosti, tj. její rozložení podle frekvence. Jinými slovy jde o frekvenčně závislý profil maximální amplitudy neurčitosti. Neurčitost $\Delta(s)$ je libovolná stabilní funkce, která splňuje nerovnost:

$$\|\Delta(s)\|_\infty \leq 1 \Rightarrow |\Delta(j\omega)| \leq 1 \quad \forall \omega \quad (7)$$

a jedná se o neznámou informaci o skutečné hodnotě a fázi perturbace.

Velmi důležitým předpokladem je, aby všechny soustavy obsažené v $G(s)$ měly stejný počet nestabilních pólů, neboli aby $G(s)$ a $G_0(s)$ měly stejný počet nestabilních pólů pro všechny $\Delta(s)$. Nesmí tedy docházet k jejich krácení. Při nesplnění tohoto požadavku by bohužel neplatila většina teoretických nástrojů pro analýzu a syntézu.

Neurčitý model soustavy, např. (6), společně s konkrétním nominálním modelem $G(s)$ a váhovou funkcí $W(s)$ tvoří tzv. *rodinu (family)*. Tento pojem bude později podobně definován také pro parametrickou neurčitost.

2.5.1 Robustní stabilita

Jak již bylo nastíněno, nazveme-li regulační obvod *robustně stabilním*, znamená to, že je stabilní nejen pro nominální soustavu $G_0(s)$, ale také pro všechny soustavy z jejího okolí dané neurčitým modelem, např. (6). Stejná myšlenka je často formulována také z pohledu regulátoru, tj. regulátor, který zajistí stabilizaci nejen soustavy $G_0(s)$, ale i všech soustav z množiny $G(s)$ je *robustně stabilizující*.

Nástroje pro analýzu robustní stability systémů s neparametrickou neurčitostí vycházejí z tzv. *věty o malém zesílení (small gain theorem)*, která říká, že za předpokladu stabilního přenosu otevřené smyčky $L(s)$ je uzavřený obvod stabilní když:

$$\|L(j\omega)\| < 1 \quad \forall \omega \quad (8)$$

kde $\|L\|$ představuje libovolnou normu splňující submultiplikativnost, tj. $\|AB\| \leq \|A\| \cdot \|B\|$. Tento teorém existuje více variantách lišících se dle aplikace (více bloků ve smyčce, nelineární systémy,...), ale hlavní myšlenka je stále stejná [20].

Pro multiplikativní model neurčitosti (6) platí, že zpětnovazební systém je robustně stabilní, právě když:

$$\|W_M(s)T_0(s)\|_\infty < 1 \quad (9)$$

Neurčitost soustavy nesmí změnit počet oběhů frekvenční charakteristiky (v komplexní rovině) otevřeného regulačního obvodu $L_0(s)$ kolem bodu $[-1, 0j]$. Z grafického pohledu to znamená, že obálka Nyquistových diagramů nesmí obsahovat kritický bod $[-1, 0j]$.

Z dosavadních poznatků je evidentní, že návrh robustně stabilizujícího regulátoru bude výrazně ovlivňovat volba váhové funkce a také volba nominálního modelu. Případní zájemci o bližší seznámení s touto problematikou mohou nahlédnout např. do [20], [21].

Hodnotu $\frac{1}{\|W_M(s)T_0(s)\|_\infty}$ můžeme interpretovat jako modernější pojetí bezpečnosti stability (viz předchozí klasická amplitudová a fázová bezpečnost). Toto číslo říká, jak velká multiplikativní neurčitost může být akceptována než pro alespoň jeden člen z rodiny (6) bude regulační obvod nestabilní.

Analogicky k podmínce robustní stability (9), která platí pro multiplikativní model neurčitosti, existují i podmínky robustní stability pro jiné typy.

2.5.2 Robustní kvalita regulace (robustní chování)

Již dříve jsme si ukázali, že pro zajištění malé regulační odchylky pro nominální soustavu se snažíme zachovat malou váženou citlivost a že tedy při splnění (5) bude mít systém požadovanou kvalitu regulace, čili bude dodržovat předepsané nominální chování (zde přesněji nominální sledování referenčního signálu).

Dále víme, že robustní stabilitu zajistíme (pro multiplikativní neurčitost) pomocí podmínky (8). Stabilita je samozřejmě zásadním kritériem, ale často požadujeme víc. Pokud chceme, aby stanovená kvalita regulace byla splněna pro celou rodinu soustav, tedy i pro ten nejhorší možný případ, hovoříme o *robustní kvalitě regulace (robustním chování)*. Tento požadavek můžeme formulovat zkombinováním předchozích podmínek, to znamená:

$$\| |W_P(s)S_0(s)| + |W_M(s)T_0(s)| \|_\infty < 1 \quad (10)$$

Pro účely návrhu regulátoru bývá podmínka (10) často s poměrně malou chybou aproximována tzv. *smíšenou citlivostí (mixed sensitivity)*:

$$\left\| \begin{array}{c} W_P(s)S_0(s) \\ W_M(s)T_0(s) \end{array} \right\|_\infty < 1 \quad (11)$$

Tyto podmínky umožňují návrh regulátorů zajišťující robustní chování pomocí H_∞ se smíšenou citlivostí tak, že je formulujeme jako optimalizační úlohu:

$$\min_{\text{stabilizující } Q(s)} \left\| \begin{array}{c} W_P(s)S_0(s) \\ W_M(s)T_0(s) \end{array} \right\|_\infty \quad (12)$$

Jako speciální případ syntézy bychom podobně např. nejlepšího nominálního sledování (kvality regulace) dosáhli pomocí regulátoru, který je řešením úlohy:

$$\min_{\text{stabilizující } Q(s)} \|W_P(s)S_0(s)\|_\infty \quad (13)$$

2.6 Parametrické neurčitosti

V této kapitole se budeme velmi okrajově zabývat typy parametrických neurčitostí, které mohou nastat a nejpoužívanějšími způsoby analýzy robustní stability. Nebudeme se nijak hluboce zabývat jednotlivými druhy neurčitostí ani metodami syntézy. Jen si nastíníme myšlenku pro obecný přehled o používaných metodách používaných v toolboxech pro robustní řízení.

Existuje několik typů parametrické neurčitosti:

- intervalová,
- afinní lineární,
- multilineární,
- polynomiální,
- obecná,
- neurčitost s jedním parametrem (speciální případ).

2.6.1 Jednopermetrická neurčitost

Nejjednodušším případem parametrické neurčitosti je neurčitost s jedním parametrem. Je to speciální případ neurčitosti. Tento typ neurčitosti má svůj velký význam v tom, že řada složitějších případů robustní stability může být zredukována právě na tento případ s jedním parametrem.

Analýza robustní stability v případě jediného neurčitého parametru je speciálním a nejjednodušším případem, kterým má význam se zabývat. Jeden z hlavních důvodů je existence řady důležitých složitějších problémů robustní stability, které mohou být zredukovány na případ s jedním parametrem. V době návrhu je o parametru známo pouze, že leží v daném intervalu.

Uvažujme zde polynom ve tvaru:

$$p(s, q) = p_0(s) + qp_1(s) \quad (14)$$

Kde:

p_0 je nominálně stabilní polynom

p_1 je libovolný polynom

q je reálný neurčitý parametr ležící v intervalu $[q_{\min}, q_{\max}]$

Nejčastějším obecným principem pro analýzu robustní stability je, že se hlídá mez stability. Tzn., že vycházíme z nějakého členu v stabilní oblasti, postupnou změnou parametrů se přibližujeme mezi stability a hlídáme přechod přes ni.

2.6.2 Intervalová neurčitost (nezávislá)

Základní myšlenkou tohoto typu neurčitosti je, aby neurčitost měla *nezávislou strukturu*. Pokud budeme uvažovat neurčitý polynom ve tvaru

$$p(s, q) = \sum_{i=0}^n \rho_i(q) s^i \quad (15)$$

bude mít nezávislou strukturu, když neurčitost q_i bude vstupovat pouze do jediného koeficientu polynomu $p(s, q)$. Jinak řečeno, změny jednoho parametru neovlivní změny ostatních parametrů polynomu.

Zápis neurčitých koeficientů u intervalové neurčitosti je formulován jako interval. Koeficienty neurčitosti náležejí do intervalu $q_i \in [q_i^-, q_i^+]$. Z toho poté vyplývá tvar pro obecnou intervalovou neurčitost jako:

$$p(s, q) = \sum_{i=0}^n [q_i^-, q_i^+] s^i \quad (16)$$

2.6.3 Afinní lineární neurčitost

Další z větších kategorií v dělení parametrických neurčitých systémech je reprezentována systémy s afinní lineární neurčitostí. Pro definici této neurčitosti si vezměme neurčitý polynom:

$$p(s, q) = \sum_{i=0}^n \rho_i(q) s^i \quad (17)$$

Tento polynom má afinní lineární strukturu, pokud každý jeho koeficient $\rho_i(q)$ je afinní lineární funkce. Tzn., že existuje takový sloupcový vektor α_i a takový skalár β_i , že platí:

$$\rho_i(q) = \alpha_i^T q + \beta_i \quad (18)$$

Kde α_i^T je transponován k α_i .

Pro účely vyšetřování robustní stability má afinní lineární neurčitá struktura poměrně důležitou vlastnost a tou je její zachování v uzavřené smyčce. Další informace o této struktuře můžete nalézt v knihách [1], [2].

2.6.4 Multilineární neurčitost

Znovu pro definici multilineární neurčitosti vyjdeme z neurčitého polynomu

$$p(s, q) = \sum_{i=0}^n a_i(q) s^i \quad (19)$$

, který bude mít multilineární strukturu neurčitosti, pokud budou všechny jeho parametry $a_i(q)$ multilineární funkce, tj. fixujeme-li všechny složky q až na jednu (např. q_i), pak je to afinní lineární funkce té zbývající složky q_i .

2.6.5 Polynomiální neurčitost

Tuto neurčitost můžeme definovat pomocí neurčitého polynomu (19), který obsahuje polynomiální neurčitost, pokud jsou všechny parametry $a_i(q)$ polynomu polynomiální funkce. Pokud polynom obsahuje parametry s druhou a vyšší mocninou, tak již říkáme, že polynom obsahuje polynomiální neurčitost.

Polynomiální strukturu můžeme převést na multilineární substitucí parametrů za každou mocninu. Tím se nám zjednoduší struktura, ale nemá to velký význam, protože nám naroste počet parametrů, takže se nám neurychlí vykreslení množiny hodnot.

Tyto druhy neurčitosti jsou velmi složité a více se jimi nebudeme zabývat. Podrobněji ji popisuje ve své literatuře Šebek [21].

2.6.6 Obecná neurčitost

Tento druh teorie není snad nijak popsán teoreticky. Je to jeden z nejsložitějších druhů neurčitosti a většinou obsahuje složité funkce ve svých parametrech. Prakticky to vypadá tak, že se zde vyskytují většinou goniometrické funkce či absolutní hodnoty aj. Prakticky jediným způsobem testování stability u toho druhu neurčitosti je věta o vyloučení nuly. Důležitou podmínkou však je, že parametry musí být spojité funkce v námi testovaném intervalu.

2.7 Metody analýzy parametrických neurčitostí

2.7.1 Bialasova věta (Bialas eigenvalue criterion)

Prvním krokem u analýzy jednoparametrické neurčitosti je, zda je $p(s, q)$ stabilní pro $q = 0$. Poté je potřeba najít nejmenší levostranné q_{\min} a největší pravostranné q_{\max} , pro které je polynom stabilní. Samozřejmě musíme najít i výsledky stability pro všechna ostatní $q \in [q_{\min}, q_{\max}]$, pokud toto platí, tak můžeme říci, že jsme našli maximální interval stability $Q_{\max} \in [q_{\min}^-, q_{\max}^+]$.

K analýze stability a výpočtu mezí využívá Bialasova věta princip hlídání singularity Hurwitzovy matice. Tato matice se sestavuje stejnými pravidly, jako když zjišťujeme stabilitu pomocí Hurwitzova kritéria. Způsob sestavování matice si zde vysvětlovat nebudeme. Touto problematikou sestavování matice a hledání mezí se podrobněji zabývají ve svých knížkách Barmish [1] nebo Bialas [3].

2.7.2 Množina hodnot (The Value Set)

K porovnání mnoha výsledků různých přístupů analýz robustní stability se často využívá tzv. množiny hodnot. Pochopení množiny hodnot umožňuje ohodnotit vývoj robustních analýz z jedné perspektivy. Tato metoda se v literaturách často využívá k ověření nových metod pro robustní stabilitu.

Hlavní myšlenkou množiny hodnot je, že můžeme přeformulovat mnoho robustních systémů do dvourozměrné množiny $V(\delta)$, která leží v komplexní rovině. Všimněme si, že je závislá na reálném skaláru δ , který nazýváme obecná frekvenční proměnná. Pokud bude δ růst, hodnoty $V(\delta)$ se obvykle pohybují po komplexní rovině. Pro mnoho případů robustní stability se ukázalo, že věta o vyloučení nuly $0 \notin V(\delta)$ je pro všechna δ nutná a také dostačující podmínka pro splnění specifikace pro robustní stabilitu.

Pro naše potřeby u toolboxů pro robustní řízení však v základě jen potřebujeme vědět, že pro intervalový polynom

$$p(s, q) = \sum_{i=0}^n [q_i^-, q_i^+] s^i \quad (20)$$

a pro jednu konkrétní pevně danou frekvenci ω_0 je množina hodnot

$$p(j\omega_0, Q) = \{p(j\omega_0, q) : q \in Q\} \quad (21)$$

vždy ve tvaru obdélníku, který má strany rovnoběžné s osami. V případě, že je jedna z částí na parametrech nezávislá, množinou hodnot je úsečka (jako zvláštní druh obdélníku). Těmto obdélníkům se říká Kharitonovovy obdélníky pro frekvenci ω_0 . Při změně ω se obdélník pohybuje a mění rozměry.

Podrobněji tuto problematiku popisuje ve své knize Barmish [1] nebo Šebek [21] ve svých přednáškách.

Pokud chceme testovat stabilitu těchto obdélníků, můžeme na ně využít např. podmínku o vyloučení nuly nebo Kharitonovovu větu, kterou si popíšeme níže.

2.7.3 Věta o vyloučení nuly (Zero Exclusion Condition)

Uvažujme spojitý intervalový polynom

$$P = \{p(., q) : q \in Q\} \quad (22)$$

invariantního stupně a předpokládejme, že existuje alespoň jeden stabilní člen $p(s, q^0)$. Polynom $p(s, q)$ je robustně stabilní právě tehdy, když je z množiny hodnot vyloučen počátek souřadnicového systému, tj. pro všechna $\omega \geq 0$ platí, že

$$0 \notin p(j\omega, Q) \quad (23)$$

Nemusíme procházet všechny polynomy z intervalového polynomu a testovat jejich stabilitu. Jestli existuje jeden stabilní polynom z rodiny polynomů, můžeme zjistit nestabilitu testováním všech polynomů podél imaginární osy. Pokud nastane případ, že pro některé koeficienty nebude platit podmínka (23), pak jsou kořeny na imaginární ose a polynom není robustně stabilní. Pokud jsou všechny výpočty nenulové, pak je polynom robustně stabilní. Výsledkem této metody pro intervalové polynomy jsou tzv. Kharitonovovy obdélníky.

Analýzu pomocí této metody lze použít i pro složité typy neurčitosti a také je možno ji jednoduše ověřit pomocí grafického testu. Tato obecná metoda v kombinaci s grafickým zobrazením je velice výhodná a přehledná pro analýzu robustnosti u složitých systému, u kterých je velmi náročné použít jiné metody. Tuto metodu podrobněji popisují ve svých knihách Kawamura [11] nebo Munro [15]. Avšak u intervalových polynomů velkého

využití nemá z důvodu existence Kharitonovových polynomů, o kterých si povíme později, a jsou mnohem výhodnější pro analýzu robustní stability u intervalových polynomů.

2.7.4 Kharitonovova věta

Tuto větu poprvé publikoval V. L. Kharitonov [12] v ruské technické literatuře již v roce 1978. Avšak zůstala nepovšimnuta skoro dvě desetiletí. Důvodem z velké části byl komplikovaný původní důkaz autora, který byl velmi složitý k pochopení. Průlom této věty přišel v roce 1994, kdy byl tento teorém využit v literatuře [1].

Definici této věty najdeme v mnoha publikacích [1], [11], [2]. Tato věta nám říká, že intervalový polynom s invariantním stupněm je robustně stabilní právě tehdy, když jsou stabilní všechny čtyři *Kharitonovovy polynomy*. Bez ohledu na počet neurčitých parametrů se testují právě jen tyto čtyři polynomy.

Je zřejmé, že pokud se vyskytuje v polynomu i koeficientů s intervalovou neurčitostí, tak celkový počet vrcholů konvexního polyedru, který ohraničuje všechny možné polynomy, je 2^i . Kharitonovův teorém stanovuje, že k vyšetření stability stačí určit stabilitu pouze ve čtyřech vrcholech, bez ohledu na velikost čísla 2^i . Tyto tzv. Kharitonovovy polynomy jsou definovány následujícím způsobem. Nechť je dán intervalový polynom neměnného stupně i (tzn., předpokládá se, že $q_i \neq 0$)

$$p(s, q_i) = q_0 + q_1 s + \dots + q_{i-1} s^{i-1} + q_i s^i \quad (24)$$

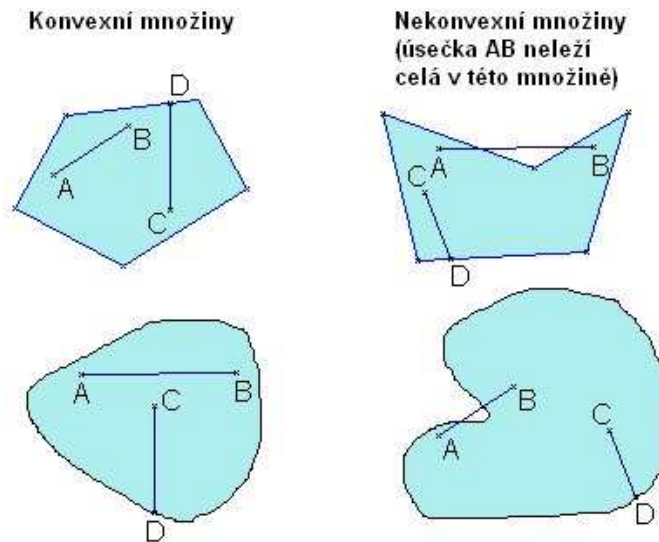
s intervalovou neurčitostí ve všech q_i . Pak nutná a postačující podmínka pro stabilitu polynomu (24) v celém rozsahu všech intervalů je ekvivalentní stabilitě těchto čtyř (Kharitonovových) polynomů [14]:

$$\begin{aligned} K_1(s) &= q_0^- + q_1^- s + q_2^+ s^2 + q_3^+ s^3 + q_4^- s^4 + q_5^- s^5 + q_6^+ s^6 + \dots \\ K_2(s) &= q_0^+ + q_1^+ s + q_2^- s^2 + q_3^- s^3 + q_4^+ s^4 + q_5^+ s^5 + q_6^- s^6 + \dots \\ K_3(s) &= q_0^+ + q_1^- s + q_2^- s^2 + q_3^+ s^3 + q_4^+ s^4 + q_5^- s^5 + q_6^- s^6 + \dots \\ K_4(s) &= q_0^- + q_1^+ s + q_2^+ s^2 + q_3^- s^3 + q_4^- s^4 + q_5^+ s^5 + q_6^+ s^6 + \dots \end{aligned} \quad (25)$$

2.7.5 Konvexní množina a konvexní obal

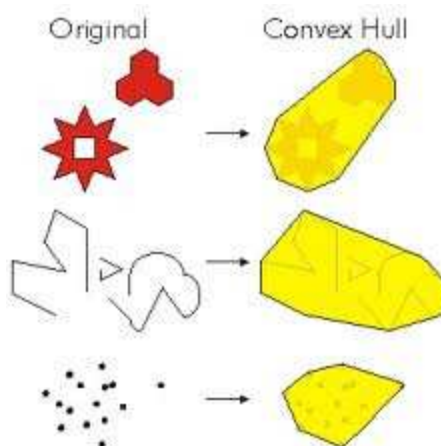
Abychom mohli provádět analýzu afinních lineárních struktur, je potřeba znát některé základní pojmy z konvexní analýzy. Tyto pojmy budou objasněny jen velmi stručně a jejich podrobnější definice popisuje ve svých kurzech [8].

Množina $C \subseteq R^k$ je *konvexní*, pokud spojnice jakýchkoliv dvou bodů z množiny C zůstane celá obsažena v C . Jednoduchou ilustraci této myšlenky můžeme vidět na obrázku (Obr. 1).



Obr. 1 Příklad konvexní a nekonzexní množiny

Konvexní obal (Convex hull) množiny $C \subset R^k$ (pro konvexní i nekonzexní množinu) definujeme jako „nejmenší“ konvexní množinu obsahující C . Přesněji, pokud konvexní množina obsahuje C se označuje jako C^+ a pole všech C^+ jako C^{++} . Konvexní obal množiny C označujeme $conv\{C\}$. Je zřejmé, že obal konvexní množiny zahrnuje předpokládanou množinu a je-li množina konvexní, její konvexní obal je totožný s touto množinou. Příklad konvexního obalu je uveden níže (Obr. 2).



Obr. 2 Příklady konvexních obalů

2.7.6 Polytopy

Definice polytopu [1] je následující. Polytop P v R^k je konvexním obalem konečné množiny bodů $\{p^1, p^2, \dots, p^m\} \in R^k$. Tudíž ho zapisujeme jako $\text{conv}\{p^i\}$.

Skupinu bodů $\{p^1, p^2, \dots, p^m\}$ nazýváme *množinou generátorů* polytopu P . Konkrétně řečeno polytopy v R^2 jsou konvexní polygony (např. obdélník), ale nejsou nekonvexní polygony (např. hvězda).

2.7.7 Polytopy polynomů a jejich množina hodnot

Před samotným popisem konkrétní metody analýzy vycházející z polytopů nám chybí ještě vysvětlit pojem *polytop polynomů*. Vyjdeme z obecné rodiny polynomů:

$$P = \{p(\cdot, q) : q \in Q\} \quad (26)$$

Tato rodina polynomů je polytopem polynomů, právě tehdy když $p(\cdot, q)$ má afinní lineární neurčitou strukturu a zároveň musí platit, že $Q = \text{conv}\{q_i\}$ (tzn., že je polytop). Z toho plyne, že $p(s, q^i)$ jsou generátory polynomů P .

Grafickým vykreslením množiny hodnot polytopu polynomů zjistíme, že výsledkem je obecný mnohoúhelník (polygon), nemusí to nutně být Kharitonovův obdélník. Tyto myšlenky jsou rozvedeny v publikacích [21], [6].

2.7.8 Věta o hranách

Z vlastností množiny hodnot a z věty o vyloučení nuly plyne, že vnitřek množiny hodnot není zas až tak důležitý. Dříve, než se nula dostane dovnitř množiny hodnot, objeví se nejdříve na hraně. Jinými slovy, pokud se nula neobjeví na žádné hraně, nemůže dojít k přechodu přes mez stability. Proto se můžeme omezit jen na zkoumání hran. U systémů s afinní lineární strukturou neurčitosti se vyšetřuje robustní stabilita *větou o hranách (The Edge Theorem)*, která je založena na této myšlence [1], [21].

Věta říká, že polynom $p(s, q)$ je robustně stabilní, jestliže $p(s, q)$ je stabilní na hranách polytopu Q .

2.7.9 Překrytí složitější neurčitosti intervalovou neurčitostí

Pravdou zůstává, že intervalová struktura neurčitosti je značně idealizovaný stav a že v praktických aplikacích se mnohdy setkáváme spíše se složitějšími strukturami (dokonce

intervalová soustava ve zpětnovazebním zapojení s regulátorem vede, vyjma speciálních případů, na charakteristický polynom uzavřeného regulačního obvodu s afinní lineární neurčitostí). Přesto však mohou být nástroje analýzy robustní stability intervalových polynomů využity i pro tyto obecnější případy. Složitější strukturu neurčitosti lze totiž „přehrančit“ intervalovou, a tak získanou rodinu pak testovat. Tato metoda však vnáší do analýzy jistou míru konzervatismu. Robustní stabilitu tedy vyšetřujeme pouze s postačující (silnější), a nikoli nutnou a postačující podmínkou. Při intervalovém „přehrančení“ totiž ignorujeme vzájemné vazby mezi koeficienty neurčitého polynomu.

Přepočítání nových mezí podrobněji popisuje ve své knize Barmish [1]. Poté aplikací Kharitonovova teoremu či grafickým testem snadno dojdeme k závěru, zda je nová vyšetřovaná rodina polynomu robustně stabilní. Pokud bude robustně stabilní, musí být stabilní i původní rodina. Jak již bylo naznačeno, pokud by nově vzniklá rodina robustně stabilní nebyla, nelze nic usuzovat o rodině originální, tj. přestože Kharitonovovy obdélníky mohou zasahovat do počátku komplexní roviny, originální množina hodnot nutně nemusí [14].

2.8 Návrh řízení pro parametrické neurčitosti

Zatím jsme si v předešlých kapitolách představili jen metody analýzy parametrických systémů. Ve výjimečných případech se dají některé tyto metody využít také pro syntézu, ale většinou se postupuje heuristicky. Návrh regulátorů pro tyto systémy je ještě složitější než jejich analýza, protože řešení často neexistuje (jsou na to často i důkazy) a i když třeba existuje, je těžké ho najít. V případech kdy takový regulátor najdeme, často má příliš vysoký řád.

Hlavním problémem při syntéze neurčitých soustav je, že oblast stability v prostoru neurčitých koeficientů je obecně nekonvexní. K řešení toho problému jsou možné dva přístupy:

- Aproximovat nekonvexní oblast stability konvexní oblastí (polytopem, LMI).
- Použít přímo metody nekonvexní optimalizace (lokální, globální).

Pomocí nástrojů analýzy robustní stability můžeme vytvořit okolo nominálně stabilního polynomu:

- stabilní segmenty (Bialas),

- stabilní kvádry (Kharitonov),
- stabilní polytopy (Věta o hranách).

Jakmile máme konvexní aproximaci oblasti stability, můžeme navrhnout robustní regulátor, a to buď:

- lineárním programováním (polytopy),
- kvadratickým programováním (sféry, elipsoidy),
- semidefinitním programováním (LMI).

Lineární programování má nejjednodušší algoritmus, avšak konservativnost tohoto návrhu je největší. Naopak tomu je u semidefinitního programování, kde je použit nejsložitější algoritmus, ale konservativnost tohoto návrhu je podstatně nižší. My se těmito metodami více zabývat nebude, ale bylo potřeba je zmínit, jelikož se k syntéze také stále využívají. Více o nich popisuje ve své literatuře [21], [6].

Existují ještě lepší vnitřní aproximace nekonvexní oblasti konvexní oblastí. Jsou založené na LMI a o těch si povíme v následující kapitole.

2.9 Lineární maticová nerovnost (LMI)

Pojem LMI je zkratkou pro anglická slova Linear Matrix Inequality. Základem této myšlenky je lineární diferenciální rovnice

$$\frac{d}{dt}x(t) = Ax(t) \quad (27)$$

, která je stabilní, právě když existuje řešení maticové nerovnosti

$$A^T P + PA < 0, \quad P = P^T > 0 \quad (28)$$

, která je lineární v neznámé matici P .

Pro naše účely v robustním řízení se používá kanonická forma

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i > 0 \quad (29)$$

Kde:

x je vektor m rozhodovacích proměnných

F_i je daná symetrická konstantní matice

Znak $>$ znamená pozitivně definitní. Také jsou neostré LMI: $F(x) \geq 0$, kde znak \geq znamená pozitivně semi-definitní. Jinak řečeno, LMI je konvexní omezení na x .

Avšak v řízení se málokdy setkáme s LMI v kanonickém tvaru, častěji mají nerovnosti maticové proměnné. Software bohužel obvykle pracuje s kanonickým tvarem, a proto je obvykle nutné náročné předzpracování (často časově náročné).

Omezení v LMI je lineární a tedy i konvexní. Z toho plynou zajímavé vlastnosti:

- množina řešení je konvexní,
- obecně neexistuje analytické řešení, ale existují numerické algoritmy, které najdou řešení, pokud existuje.

2.9.1 Analýza a syntéza pomocí LMI

Pro popis způsobu testování stability pomocí LMI je potřeba si nejprve objasnit několik pojmů.

Prvním pojmem je *pozitivní polynom*. Tyto polynomy tvoří na reálné ose konvexní množinu, kterou můžeme vyjádřit pomocí LMI.

Dále pak definujeme racionální matici $G(s)$, která je *striktně pozitivně reálná* (SPR), když platí:

$$G(s) > 0, \forall s \in \partial D \quad (30)$$

Kde ∂D značí hranici oblasti stability v komplexní rovině.

Pomocí kombinace SPRnosti racionální matice a positivity polynomu můžeme zkoumat robustní stabilitu polynomiálních a stavových matic.

Hlavní myšlenkou testování stability pomocí LMI je, že polynomiální matice $C(s)$ je stabilní, právě když existuje stabilní polynomiální matice $D(s)$ a matice $P = P^*$ splňující LMI:

$$D^* C + C^* D - S(P) > 0 \quad (31)$$

Všimněte si, že v LMI jsou matice systému $C(s)$ a neznámá matice P odděleny. Jakmile stabilní polynomiální matice $D(s)$ je dána, problém je lineární a tedy konvexní. Pro vysvětlení robustního návrhu pomocí LMI si uvedeme názorný obecný příklad.

Uvažujme rodinu soustav $A^{-1}(s, q)B(s, q)$ s multilineární strukturou neurčitosti a polytopem Q . Spojenou zpětnovazebním regulátorem $Y(s)X^{-1}(s)$, tak že dynamiky uzavřené regulační smyčky je popsána rodinou matic $C(s, q) = A(s, q)X(s) + B(s, q)Y(s)$. Abychom zajistili SPRnost racionální matice $D^{-1}(s)C(s, q)$, volíme stabilní polynomiální matice $D(s)$ blízko nominální matici uzavřené smyčky, např. přímo $D(s) = C(s, q_0)$, tedy rovnou matici jmenovatele nominální uzavřené smyčky, kterou získáme standardním návrhem (např. metodou umístění pólů). Úlohy řešené pomocí SPRnosti vedou na LMI s Toeplitzovou strukturou matic, ale při výpočtu toho nevyužíváme (neboť využíváme obecné řešitele). Zatím chybí systematický způsob volby centrálního polynomu. Vylepšení by mohlo přinést i použití jiných bází pro polynomy.

Samozřejmě toto je jen hrubý nástin celého problému týkajícího se LMI. Přesnější a více konkrétní definice popisuje ve svých kurzech Henrion [8].

II. PRAKTICKÁ ČÁST

3 PROGRAMOVÉ PROSTŘEDÍ MATLAB

MATLAB je integrované prostředí pro vědeckotechnické výpočty, modelování, návrhy algoritmů, simulace, analýzu a prezentaci dat, paralelní výpočty, měření a zpracování signálu, návrhy řídicích a komunikačních systémů.

Výpočetní systém MATLAB se během uplynulých let stal celosvětovým standardem v oblasti technických výpočtů a simulací ve sféře vědy, výzkumu, průmyslu i v oblasti vzdělávání.

MATLAB poskytuje svým uživatelům nejen mocné grafické a výpočetní nástroje, ale i rozsáhlé specializované knihovny funkcí spolu s výkonným programovacím jazykem čtvrté generace. Knihovny jsou svým rozsahem využitelné prakticky ve všech oblastech lidské činnosti.

Díky své architektuře je MATLAB určen zejména těm, kteří potřebují řešit početně náročné úlohy a přitom nechťejí nebo nemají čas zkoumat matematickou podstatu problémů. Více než milion uživatelů po celém světě využívá možnosti jazyka MATLABu, který je mnohem jednodušší než například Fortran nebo C a který skýtá obrovský potenciál produktivity a tvořivosti. Za nejsilnější stránku MATLABu je považováno mimořádně rychlé výpočetní jádro s optimálními algoritmy, které jsou prověřeny léty provozu na špičkových pracovištích po celém světě. MATLAB byl implementován na všech významných platformách (Windows, Linux, Solaris, Mac).

Název MATLAB vznikl zkrácením slov MATrix LABoratory (volně přeloženo „laboratoř s maticemi“), což odpovídá skutečnosti, že klíčovou datovou strukturou při výpočtech v MATLABu jsou matice. Vlastní programovací jazyk vychází z jazyka Fortran.

3.1 Výpočetní jádro

Nejpodstatnější součástí numerického jádra MATLABu jsou algoritmy pro operace s maticemi reálných a komplexních čísel. MATLAB umožňuje provádět všechny běžné operace jako násobení, inverze, determinant atd. a v nejjednodušší podobě je možno jej použít jako maticový kalkulátor, protože všechny tyto operace se zapisují téměř tak, jako bychom je psali na papíře. Kromě datových typů jednodušších než tradiční matice podporuje MATLAB také typy složitější, jako jsou např. vícerozměrná pole reálných nebo komplexních čísel. Dalším datovým typem jsou tzv. pole buněk, tedy struktury podobné maticím, ve kterých ovšem každý prvek může být jiného typu. Podobně lze tvořit datové

struktury, kde jsou prvky rozlišeny ne souřadnicemi, ale jménem, takže připomínají struktury známé z běžných programovacích jazyků. Skládáním těchto datových typů je pak možné vytvořit libovolně složité datové struktury.

3.2 Grafický subsystém

Grafika v MATLABu umožňuje snadné zobrazení a prezentaci výsledků získaných výpočtem. Je možné vykreslit různé druhy grafů: dvourozměrné pro funkce jedné proměnné, třírozměrné pro funkce dvou proměnných, histogramy, koláčové grafy a další. Všem grafickým objektům je možné téměř libovolně měnit vzhled, a to jak při jejich vytváření, tak po jejich nakreslení. Tak je možné stínovat třírozměrné grafy s určením zdroje dopadajícího světla, animovat grafy včetně třírozměrných, zobrazovat kontury a transparentní objekty, používat pseudobarevné zobrazení, a mnoho dalšího. Obrázky v grafických oknech MATLABu navíc nejsou statické - každý již nakreslený objekt má přiřazen identifikátor, jehož prostřednictvím je možné měnit vlastnosti objektu a tím i jeho vzhled. Vzhled grafických objektů je také možno měnit interaktivně, pomocí lišty nástrojů umístěné pod záhlavím obrázku. Grafický systém MATLABu, nazvaný Handle Graphics, dovoluje vkládat do obrazů ovládací prvky (tlačítka, apod.) a vytvořit tak aktivní graficky ovládané uživatelské rozhraní.

3.3 Otevřená architektura

Vlastností, která patrně nejvíce přispěla k rozšíření MATLABu, je jeho otevřená architektura. MATLAB je úplný programovací jazyk, to znamená, že uživatelé v něm mohou vytvářet funkce "šité na míru" pro jejich aplikace. Tyto funkce se způsobem volání nijak neliší od vestavěných funkcí a jsou uloženy v souborech v čitelné formě. Dokonce většina funkcí s MATLABem dodávaných je takto vytvořena a opravdu vestavěné jsou jen funkce základní. To má dvě velké výhody: jazyk MATLABu je téměř neomezeně rozšiřitelný a kromě toho se uživatel může při psaní vlastních funkcí poučít z dodaných algoritmů. Všechny moduly systému doprovází rozsáhlá tištěná i hypertextová on-line dokumentace, která uživatelům usnadňuje orientaci ve funkcích MATLABu. Otevřená architektura MATLABu inspirovala mnoho nezávislých firem k vývoji a distribuci vlastních produktů, které buď rozšiřují výpočetní prostředí MATLAB o další knihovny a nástroje nebo zajišťují propojení MATLABu s jinými specializovanými programy.

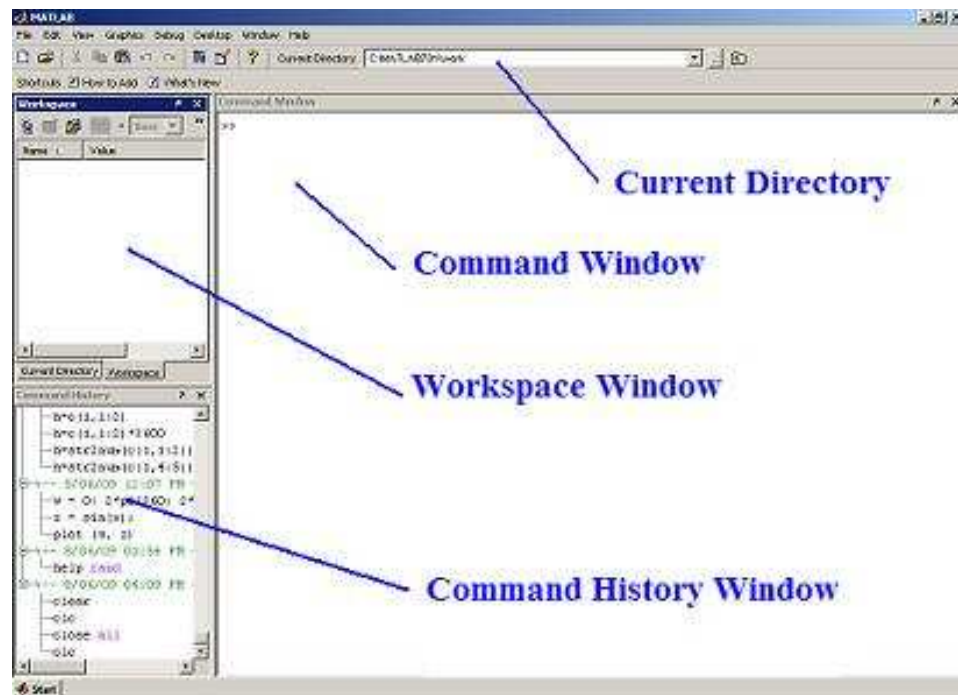
3.4 Pracovní nástroje

MATLAB je koncipován tak, aby kromě pohodlné interaktivní práce umožňoval i programování aplikací. Programovací jazyk obsahuje všechny nezbytné příkazy pro psaní programů, jako jsou podmíněné příkazy, větvicí příkazy, cykly a podobně. Díky jednoduchosti a snadné zvladatelnosti je jazyk MATLABu úplným programovacím jazykem čtvrté generace, ve kterém je možné vyvíjet i velice složité aplikace. Základním nástrojem výpočetního systému je uživatelské rozhraní MATLAB Desktop. Pracovní nástroje jako prohlížeč adresářů a souborů, prohlížeč pracovního prostoru, okno historie příkazů, interaktivní spouštěč aplikací, editor, debugger, profiler, hypertextová nápověda a příkazové okno jsou do prostředí plně integrovány. Uživatelské rozhraní je konfigurovatelné, takže si uživatel může přizpůsobit rozměry a počet zobrazených nástrojů tak, aby to maximálně vyhovovalo jeho potřebám. Důležitým pomocníkem je interaktivní nástroj pro import dat, který výrazně usnadní načítání dat prakticky z jakéhokoli zdroje (text, tabulky, databáze, binární data, obrázky, animace,...). MATLAB je koncipován tak, aby kromě pohodlné interaktivní práce umožňoval i programování aplikací. Programovací jazyk obsahuje všechny nezbytné příkazy pro psaní programů, jako jsou podmíněné příkazy, větvicí příkazy, cykly a podobně.

3.5 Toolboxy

Otevřená architektura MATLABu vedla ke vzniku knihoven funkcí, nazývaných toolboxy, které rozšiřují použití programu v příslušných vědních a technických oborech. Tyto knihovny, navržené a v jazyce MATLABu napsané nejvýznačnějšími světovými odborníky, nabízejí předzpracované specializované funkce, které je možno rozšiřovat, modifikovat, anebo jen čerpat informace z přehledně dokumentovaných algoritmů.

Mnoho Toolboxů je nyní již integrováno přímo v základním balíčku MATLABu, avšak pokud chceme přidat ke stávajícím další, musíme jej zkopírovat do určité složky s nainstalovaným MATLABem. V základním adresáři MATLABu se nachází složka s názvem „Toolbox“, do které se nakopíruje námi požadovaný nový toolbox. Dalším krokem v instalaci je, že musíme spustit programové prostředí MATLAB a přiřadit tuto nově nakopírovanou složku k programu, aby věděl, že ji má prohledávat při zadání příkazu, který využívá tento toolbox. Složku jednoduše přidáme kliknutím myši na *File -> Set Path -> Add Folder*. Poté jen vybereme cestu k našemu toolboxu a potvrdíme tlačítkem *Save*.



Obr. 3 Hlavní okno programu MATLAB

4 CHARAKTERISTIKA VYBRANÝCH TOOLBOXŮ

V této kapitole si představíme ty nejznámější a nejvyužívanější toolboxy pro robustní řízení pro programové prostředí MATLAB. Samozřejmě takových toolboxů existuje nepřeberné množství jak komerčních, tak volně šiřitelných. Zde si zmíníme jen ty nejvíce sofistikované a vybavené nástroje pro řešení co největší škály problémů na poli robustního řízení.

4.1 Control System Toolbox

Control System Toolbox poskytuje řadu standardních algoritmů a nástrojů pro systematické analýzy, návrhy a ladění lineárních řídicích systémů. Můžete zadat systém jako přenosovou funkci, póly-nuly-zesílení systému, nebo frekvenční odezvy modelu. Interaktivní nástroje a funkce příkazového řádku, jako je vykreslení přechodové charakteristiky a Bodeho grafu, umožní člověku si představit chování systému v časové a frekvenční oblasti. Můžete nastavit parametry kompenzátorů pomocí automatického ladění PID regulátoru, tvarování Bodeho smyčky a jiné interaktivní a automatizované techniky. Můžete ověřit svůj návrh tím, že ověříte dobu náběhu, překročení, čas ustálení, zesílení a další požadavky.

Lineární metody řízení jsou základem analýzy a návrhu řídicích systémů. Control System Toolbox umožňuje vytvářet a manipulovat s lineárními modely řídicího systému. Pomocí interaktivních vizualizačních nástrojů můžete analyzovat tyto modely a získat představu např. o chování nebo výkonu systému. Můžete také systematicky ladit parametry řídicích systémů pomocí Single-Input/Single-Output (SISO) a Multi-Input/Multi-Output (MIMO) návrhových technik.

Lineární modely z Control System Toolbox mohou být použity v jiných produktech pro návrh řízení, jako je Robust Control Toolbox a Model Predictive Control Toolbox. Můžete použít Simulink Control Design spolu s Control System Toolbox pro návrh řídicího systému a analýzy v Simulinku . [23]

4.2 Robust Control Toolbox

Robust Control Toolbox poskytuje nástroje pro analýzu a automatické ladění řídicích systémů pro výkon a robustnost. Můžeme díky němu vytvářet neurčité modely kombinací nominální dynamiky s neurčitými prvky, jako jsou neurčité parametry nebo nemodelované

dynamiky. Můžeme analyzovat dopad neurčitosti modelu systému na výkon řídicího systému a zjistit nejhorší možné případy kombinace neurčitých prvků. Pokud použijeme H_{∞} nebo μ -syntézu, můžeme navrhnout regulátory, které maximalizují robustní stabilitu a výkon. Toolbox dokáže automaticky ladit jak SISO tak MIMO robustní regulátory, včetně decentralizované architektury řízení navržené v Simulinku. Můžeme ověřit svůj návrh tím, že např. amplitudovou a fázovou bezpečnost či nejhorší citlivost na rušení.

Hlavní rysy toolboxu:

- modelování systémů s neurčitými parametry,
- analýza nejhorší možné stability a výkonu neurčitých systémů,
- automatické ladění centralizovaných a decentralizovaných řídicích systémů,
- analýza robustnosti a ladění regulátoru v Simulinku,
- algoritmy H_{∞} a μ -syntéza,
- univerzální LMI řešitelé proveditelnosti, minimalizace lineárních cílů a minimalizace všeobecných vlastních čísel,
- algoritmy na snížení stupně modelu na základě Hankelových singulárních hodnot.

Díky Robust Control Toolboxu můžeme zachytit nejen typické nebo nominální chování našeho modelu, ale také množství neurčitosti a variabilitu. Neurčitost modelů systému může vyplývat z:

- parametrů modelu s přibližně známými nebo měnící se hodnotami,
- opomíjené nebo nedostatečně známé dynamiky, jako je například vysokofrekvenční dynamika,
- změn v provozních podmínkách,
- lineární aproximace nelineárního chování,
- odhadu chyby modelu identifikovaného z naměřených dat.

Toolbox umožňuje vytvářet detailní neurčité modely tím, že kombinuje nominální dynamiku s neurčitými prvky, jako jsou neurčité parametry nebo zanedbaná dynamika. Kvantifikací míry neurčitosti v každém prvku můžeme zachytit celkovou věrnost a variabilitu našeho modelu systému. Poté můžeme analyzovat, jak každý neurčitý prvek

ovlivňuje výkon a identifikovat nejhorší možný případ kombinace neurčitých hodnot prvků.

Pomocí Robust Control Toolboxu lze analyzovat vliv neurčitosti modelu systému na stabilitu uzavřené smyčky a výkon řídicího systému. Zejména můžeme zjistit, zda řídicí systém bude řádně regulovat po celou šířku jeho rozsahu, a to jaký zdroj neurčitosti může s největší pravděpodobností ohrozit výkon systému.

Je možno náhodně rozdělit neurčitost modelu k provedení Monte Carlo analýzy. Případně můžeme použít více přímé nástroje založené na μ -analýze a optimalizaci lineární maticové nerovnosti (LMI), tyto nástroje identifikují scénáře nejhoršího možného vývoje bez vyčerpávajících simulací.

Robust Control Toolboxu poskytuje funkce pro posouzení nejhorších možných případů hodnot pro:

- celý rozsah zesílení a fází,
- rozpětí stability, které berou v úvahu interakce smyčky,
- zesílení mezi libovolnými dvěma body v systému s uzavřenou smyčkou,
- citlivost na vnější rušení.

Tyto funkce také poskytují informace o citlivosti, které pomáhají identifikovat neurčité prvky, které nejvíce přispívají ke snížení výkonu. S touto informací můžeme určit, zda navrhnout přesnější model, přísnější výrobní tolerance nebo přesnější senzor, který by nejvíce posílil robustnost systému.

Robust Control Toolbox dokáže automaticky ladit centralizované a decentralizované MIMO řídicí systémy. Algoritmy pro syntézu regulátoru jsou založeny na technikách H_∞ nebo μ -syntéza v kombinaci s LMI optimalizací. Tyto algoritmy jsou použitelné pro SISO i MIMO řídicí systémy. Syntéza MIMO regulátoru nevyžaduje uzavření sekvenční smyčky, a je proto vhodný i pro vícesmyčkové systémy řízení se vzájemnou vazbou a interakcí smyček.

Většina vestavěných řídicích systémů mají pevnou, decentralizovanou architekturu s jednoduchými laditelnými prvky, jako jsou zesílení, PID regulátory nebo nízkopásmové filtry. Tyto architektury se lépe chápou, realizují a přeladují než komplexní centralizované

regulátory. Robust Control Toolbox poskytuje nástroje pro modelování a ladění těchto decentralizovaných řídicích architektur. Můžete:

- Specifikovat laditelné prvky, jako jsou zesílení, PID regulátory, pevně stanovené přenosové funkce a stavové modely.
- Kombinovat laditelné prvky s obyčejnými lineárními časově invariantními (LTI) modely pro vytvoření laditelného modelu řídicí architektury.
- Specifikovat požadavky na šířku pásma, tvaru smyčky, kvalitu řízení a kompenzaci poruchy.
- Automaticky ladit parametry regulátoru, aby splňovaly požadavky.
- Ověřit výkon regulátoru v časové a frekvenční oblasti.

Robust Control Toolbox nabízí několik algoritmů pro syntézu robustních MIMO regulátorů přímo z frekvenčních specifikací uzavřené smyčky. Například můžete omezit maximální zesílení citlivostní funkce pro zlepšení stability a snížení překmitu, nebo omezit zesílení vstupního rušení na měřený výstup pro zlepšení odstupu od rušení. Pomocí algoritmů μ -syntézy, můžete optimalizovat výkon regulátoru neurčitého modelu kvůli zajištění efektivního výkonu pro všechny reálně možné situace. Techniky H_∞ a μ -syntéza poskytují jedinečný pohled do limitů výkonu řídicí architektury a umožňují rychle vyvinout první návrhy kompenzátorů.

Podrobné první návrhy modelu systému nebo modely systému s konečným počtem prvků mají často velký počet stavů. Podobně H_∞ a μ -syntéza algoritmy mají tendenci navrhovat regulátory s příliš vysokým řádem. Robust Control Toolbox poskytuje algoritmy, které umožňují snížit řád (počet stavů) systému nebo regulátoru modelu při zachování jejich základních dynamik. Při snižování řádů modelů, které jsou lépe fyzikálně realizovatelné, lze kontrolovat chybu aproximace.

Algoritmy ke snížení řádu modelu jsou založeny na hodnotách Hankelových singulárních čísel systému, který měří energii stavů. Tím, že ponechá vysoce energetické stavy a ignoruje nízkoenergetické stavy, zredukovaný model zachovává základní rysy původního modelu. Lze použít absolutní nebo relativní chybu aproximace pro výběr řádu a používat frekvenčně závislé na parametry k zaměření algoritmu na snížení řádu modelu v konkrétních frekvenčních pásmech. [24]

4.2.1 Robustnost Servo regulátoru pro stejnosměrný motor

V této části je demonstrováno, jak používat neurčitých objektů v Robust Control Toolbox pro modelování neurčitých systémů. Nástroje pro analýzu robustnosti, které automatizují robustní výpočty, jsou rovněž demonstrována.

Datové struktury pro modelování neurčitosti

Robust Control Toolbox umožňuje vytvořit neurčité prvky, jako jsou fyzikální parametry, jejichž hodnoty nejsou přesně známy, a tyto prvky zkombinovat do neurčitých modelů. Poté můžeme snadno analyzovat dopad neurčitosti na výkon řídicího systému.

Pro názorný příklad budeme uvažovat model systému:

$$G(s) = \frac{\gamma}{\tau s + 1} \quad (32)$$

kde gama se může pohybovat v intervalu [3,5] a τ má průměrnou hodnotu 0,5 s 30% rozsahem. Pomocí jednoduchých příkazů v MATLABu můžeme vytvořit neurčitý model $G(s)$ jakož tomu je i v tomto příkladu:

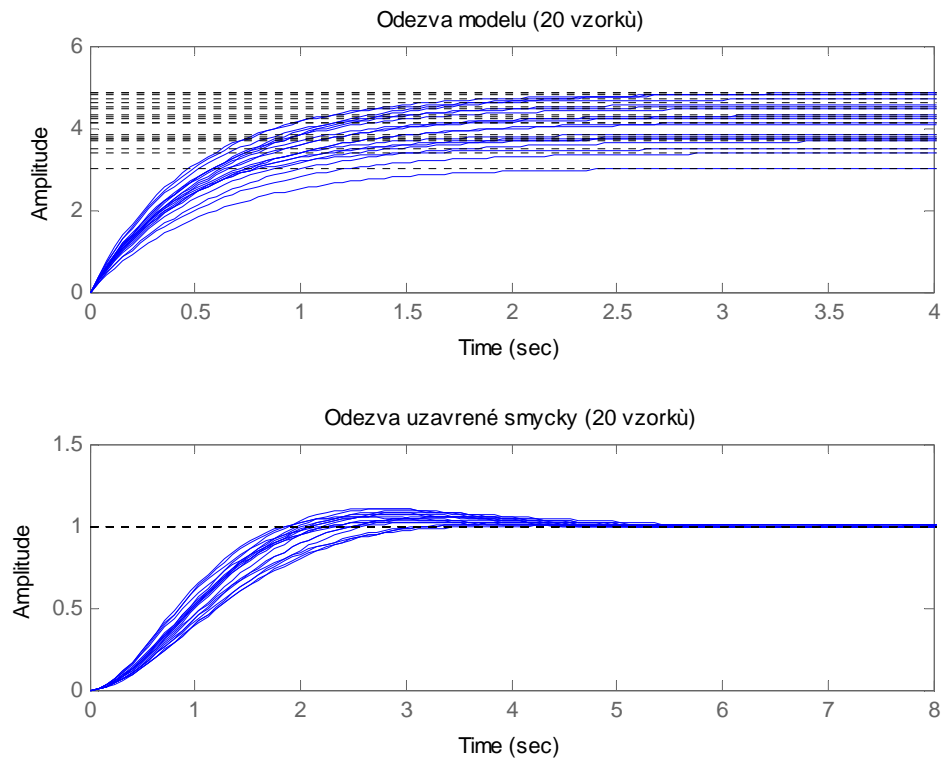
```
gamma = ureal('gamma',4,'range',[3 5]);
tau = ureal('tau',.5,'Percentage',30);
G = tf(gamma,[tau 1])
```

Předpokládejme, že jsme navrhli integrační regulátor C pro nominální hodnota modelu (gama = 4 a tau = 0,5). Chceme-li zjistit, jak variace gama a tau ovlivní model a výkon uzavřené smyčky, utvořili jsme uzavřenou smyčku CLG z C a G pomocí příkazů:

```
KI = 1/(2*tau.Nominal*gamma.Nominal);
C = tf(KI,[1 0]);
CLG = feedback(G*C,1)
```

Nyní můžeme generovat 20 náhodných vzorků neurčitých parametrů gama a tau a vykreslit odpovídající přechodové charakteristiky modelu a uzavřené smyčky modelu pomocí příkazů:

```
subplot(2,1,1); step(usample(G,20)), title('Odezva modelu (20 vzorků)')
subplot(2,1,2); step(usample(CLG,20)), title('Odezva uzavřené smyčky (20 vzorků)')
```



Obr. 4 Přechodové charakteristiky modelu systému a uzavřené smyčky

Spodní graf – Odezva uzavřené smyčky (Obr. 4) ukazuje, že systém uzavřené smyčky je poměrně robustní i přes značné výkyvy v zesílení DC modelu. Toto je žádoucí charakteristika správně navrženého systému se zpětnou vazbou.

Stejnoseměrný motor s neurčitými parametry a nemodelovanou dynamikou

Nyní přidáme neurčitý parametr a nemodelovanou dynamiku a vyšetříme robustnost serva při takové neurčitosti.

Nominální model stejnosměrného motoru je definován odporem R , indukčností L , elektromotorickou konstantou K_B , konstantou armatur k_m , lineární aproximací viskózního tření K_F a setrvačnou zátěží J . Každá z těchto složek se pohybuje v určitém rozmezí hodnot. Konstanty odporu a indukčnosti se pohybují v rozmezí $+ / - 40\%$ jejich jmenovitých hodnot. Používáme funkci *ureal* pro vytvoření těchto nejistých parametrů:

```
R = ureal('R', 2, 'Percentage', 40);
L = ureal('L', 0.5, 'Percentage', 40);
```

Z fyzikálních důvodů, hodnoty K_f a K_b jsou stejné, i když jsou neurčité. V tomto příkladu jmenovitá hodnota je 0,015 v rozmezí od 0,012 do 0,019. Viskózní tření K_f má jmenovitou hodnotu 0,2 s 50% kolísáním jeho hodnoty.

```
K = ureal('K',0.015,'Range',[0.012 0.019]);
Km = K;
Kb = K;
Kf = ureal('Kf',0.2,'Percentage',50);
```

Elektrické a mechanické rovnice

Proud v elektrickém obvodu a krouticí moment působící na rotor lze vyjádřit z hlediska napětí a úhlové rychlosti. Vytvoříme přenosovou funkci H týkající se těchto proměnných a vytvoříme *AngularSpeed* výstup z H pro pozdější použití:

```
H = [1;0;Km] * tf(1,[L R]) * [1 -Kb] + [0 0;0 1;0 -Kf];
H.InputName = {'AppliedVoltage';'AngularSpeed'};
H.OutputName = {'Current';'AngularSpeed';'RotorTorque'};
```

Motor zpravidla řídí setrvačnost, jejíž dynamické vlastnosti vychází z aplikovaného krouticího momentu na rychlost změny úhlové rychlosti. Pro nehybná tělesa, to je konstanta. Realističtější, ale neurčitý model může obsahovat neznámé tlumené rezonance. Použijte se na to *ultidyn* objekt, který slouží pro modelování neurčitých lineárních časově invariantních dynamik. Nominální hodnota setrvačnosti tělesa je nastavena na 0,02 a přidáme 15% dynamické neurčitosti v multiplikativním tvaru:

```
J = 0.02*(1 + ultidyn('Jlti',[1 1],'Type','GainBounded','Bound',0.15,...
'SampleStateDim',4));
```

Neurčitý model SS motoru

Jednoduše můžeme vztáhnout vstup *AngularSpeed* vstup do výstupu *RotorTorque* přes neurčitou setrvačnost J pomocí příkazu *lft*. Vstup *AngularSpeed* se rovná *RotorTorque/(J*s)*, tedy „pozitivní“ zpětnou vazbou ze 3. výstupu do 2. vstupu H se používá k propojení. To má za následek systém s 1 vstupem (*AppliedVoltage*) a 2 výstupy, (*Current* a *AngularSpeed*).

```
Pal1 = lft(H, tf(1,[1 0])/J);
```

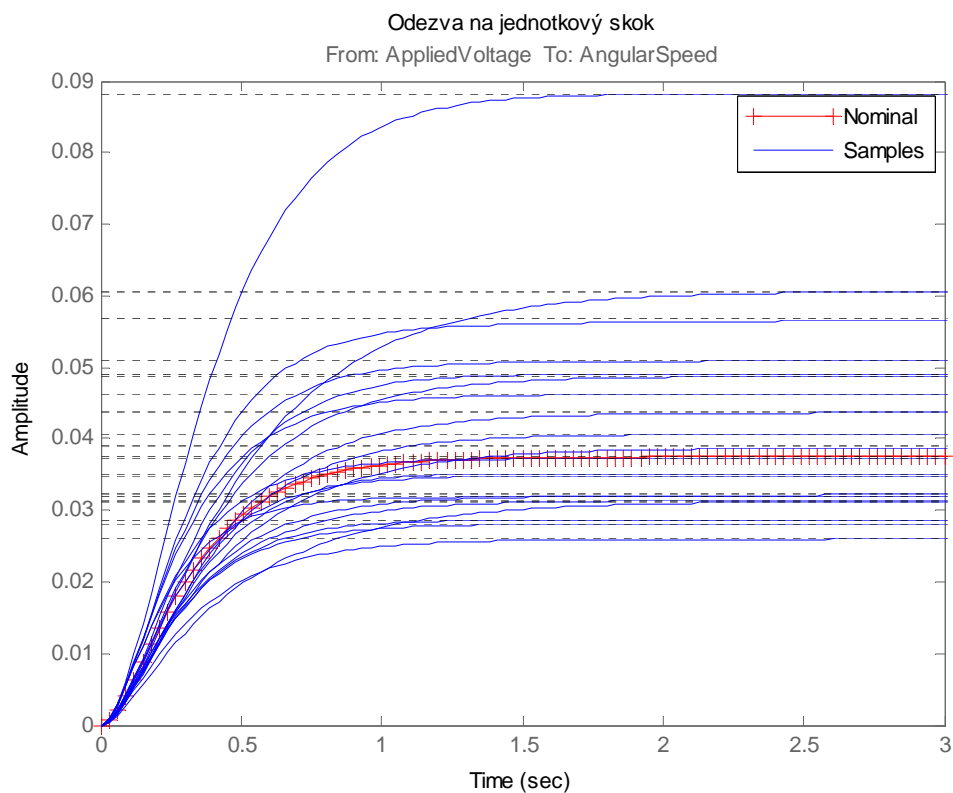
Vybere se pouze *AngularSpeed* výstup pro zbytek řídicí analýzy:

```
P = Pal1(2,:)
```

P je neurčitý model DC motoru s jedním vstupem a jedním výstupem. Pro účely analýzy používáme nominální regulátor navrhnutý pro motor DC v "Začínáme s Toolbox Control System" manuálu.

Analýza otevřené smyčky

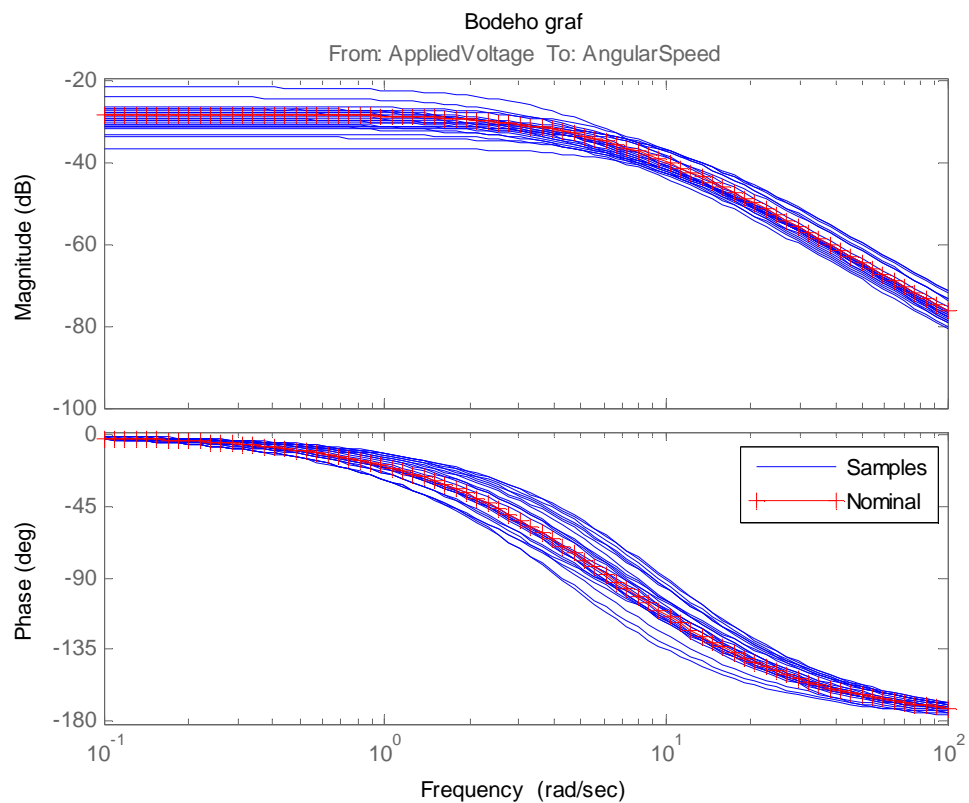
Nejprve srovnáme odezvu na jednotkový skok jmenovitého stejnosměrného motoru se 20 vzorků neurčitého modelu stejnosměrného motoru:



Obr. 5 Analýza odezvy otevřené smyčky na jednotkový skok

Stejně tak můžeme porovnat Bodeho graf otevřené smyčky s nominální hodnotou (červená) a vzorků (modrá) neurčitého modelu DC motoru:

```
om = logspace(-1,2,80);
Pg = ufrd(P,om);
bode(usample(Pg,25),'b',Pg.NominalValue,'r-+');
legend('Samples','Nominal')
```

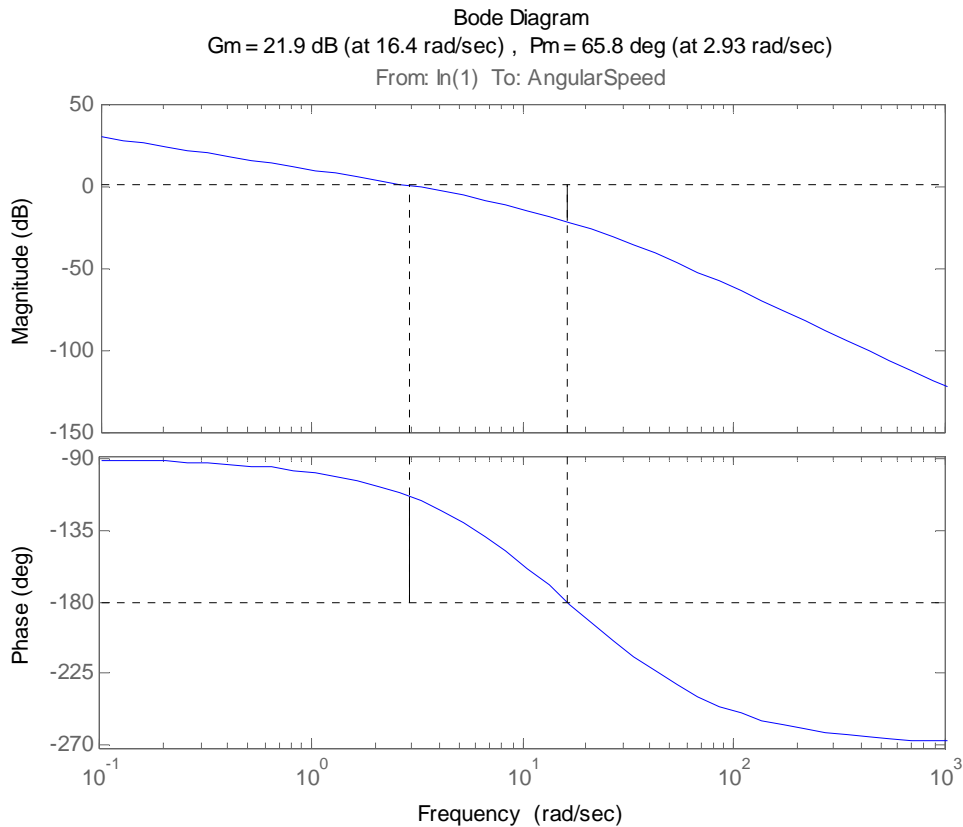


Obr. 6 Bodeho graf otevřené smyčky stejnosměrného motoru

Robustní analýza uzavřené smyčky

Nyní ověříme stabilitu a robustnost řízení uzavřené smyčky systému stejnosměrného motoru. Naše počáteční analýza nominálního systému uzavřené smyčky indikuje, že jmenovitý systém uzavřené smyčky je velmi robustní. Parametr amplitudové bezpečnosti vyšel 10,5 a parametr fázové bezpečnosti má hodnotu $54,3^\circ$.

```
margin(P.NominalValue*Cont)
```



Obr. 7 Analýza robustnosti uzavřené smyčky

Funkce *loopmargin* poskytuje komplexní analýzu stability pro více proměnných systémů se zpětnou vazbou. U řídicího systému s N smyčkami zpětné vazby, vrátí *loopmargin* funkce:

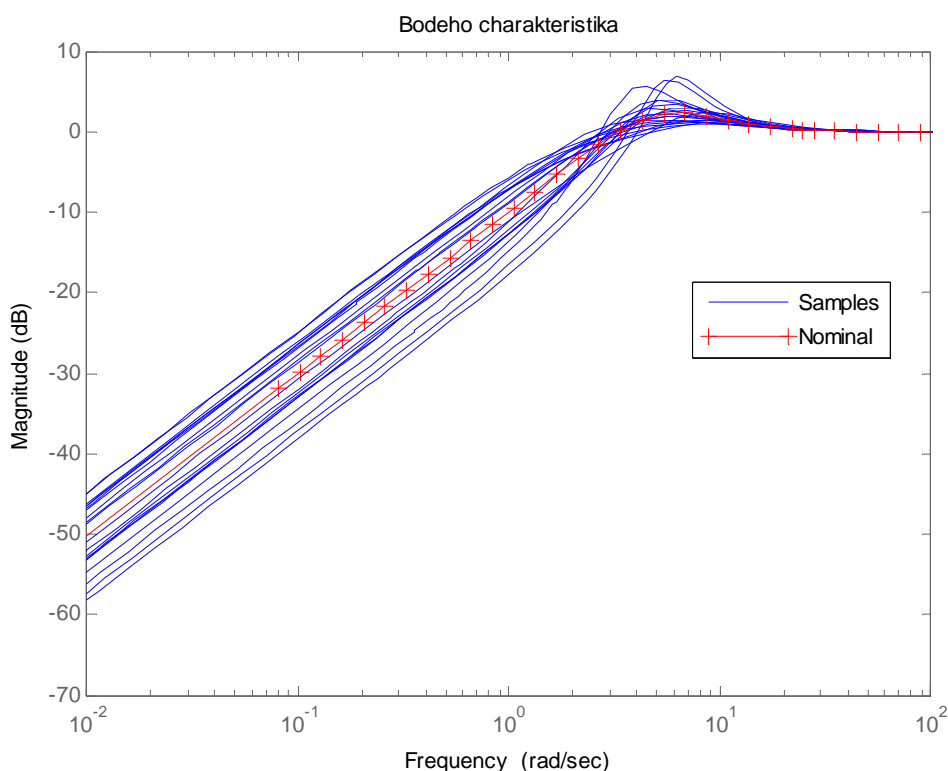
- Klasickou amplitudovou a fázovou bezpečnost pro každou jednotlivou smyčku zpětné vazby.
- Diskovou bezpečnost (Disk Gain Margin) pro každou jednotlivou smyčku zpětné vazby. Disková bezpečnost j -té smyčky zpětné vazby indikuje o kolik se přenosová funkce $L_j(s)$ může lišit, než právě tato smyčka přejde v nestabilní.
- Více-smyčkovou diskovou bezpečnost. Tato indikuje, kolik může být simultánních, nezávislých variací amplitudové a fázové bezpečnosti v každé smyčce než se celková uzavřená smyčka stane nestabilní (stejně jako disková bezpečnost pro řídicí systém s jednou smyčkou).

Připomeňme, že model systému stejnosměrného motoru je neurčitý. Kromě standardní amplitudové a fázové bezpečnosti můžeme použít funkci *wmargin* pro určení nejhoršího možného případu amplitudové/fázové bezpečnosti pro smyčku zpětné vazby systém-regulátor. Funkce vypočítá nejhorší případ diskové amplitudové a fázové bezpečnosti pro každý vstupní/výstupní kanál. Analýza nejhoršího možného případu ukazuje možnou degradaci. Z hodnot 11dB a 59° na 1,2dB a 8° v přítomnosti 5 forem neurčitosti modelovaných v P.

Robustnost charakteristik pro kompenzaci poruchy

Citlivostní funkce je standardní měřítko kvality uzavřené smyčky zpětné vazby. Spočítejme neurčitou citlivostní funkci S a porovnejme Bodeho amplitudové charakteristiky pro jmenovitou a vzorkovanou neurčitou citlivostní funkci:

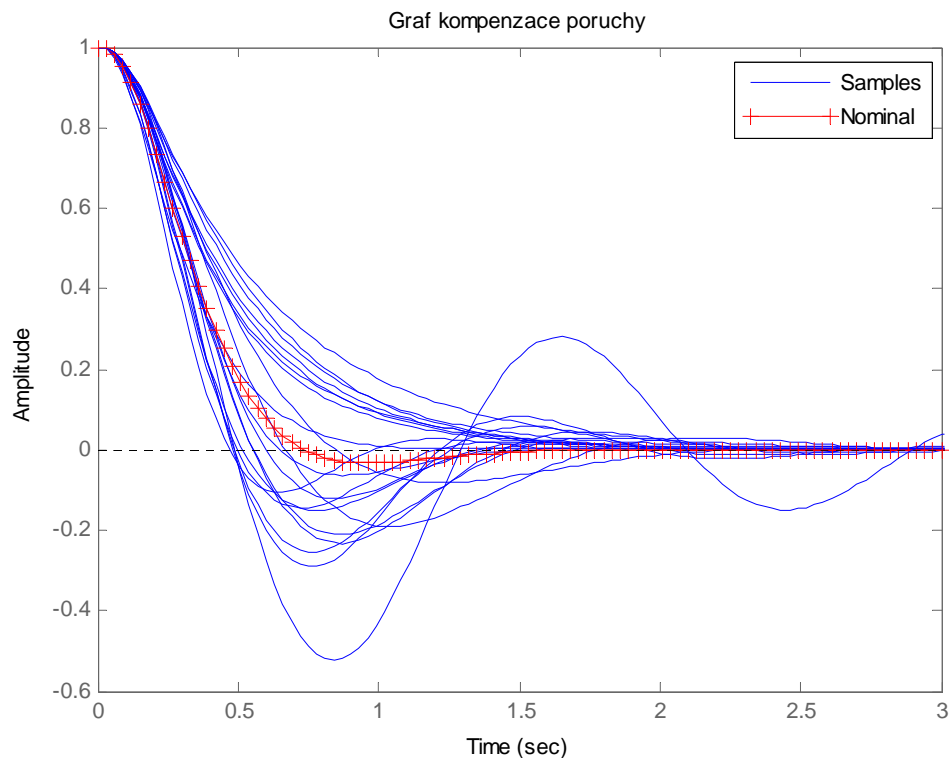
```
S = feedback(1,P*Cont);
bodemag(usample(S,20),'b',S.Nominal,'r-+');
legend('Samples','Nominal')
```



Obr. 8 Amplituda citlivostní funkce S

V časové oblasti citlivostní funkce indikuje, jak dobře může být kompenzována porucha ve tvaru jednotkového skoku. Neurčitou citlivostní funkci nyní navzorkujeme a vykreslíme graf odezvy na jednotkový skok, abychom viděli proměnlivost v charakteristikách pro kompenzaci poruchy. Jmenovité hodnoty jsou vykresleny červeně.

```
step(usample(S,20),'b',S.Nominal,'r-+',3);
title('Disturbance Rejection')
legend('Samples','Nominal')
```



Obr. 9 Kompenzace poruchy ve tvaru jednotkového skoku

Můžeme použít funkci *wcgain* pro výpočet nejhorší hodnoty zesílení neurčité citlivostní funkce (špička napříč frekvencí).

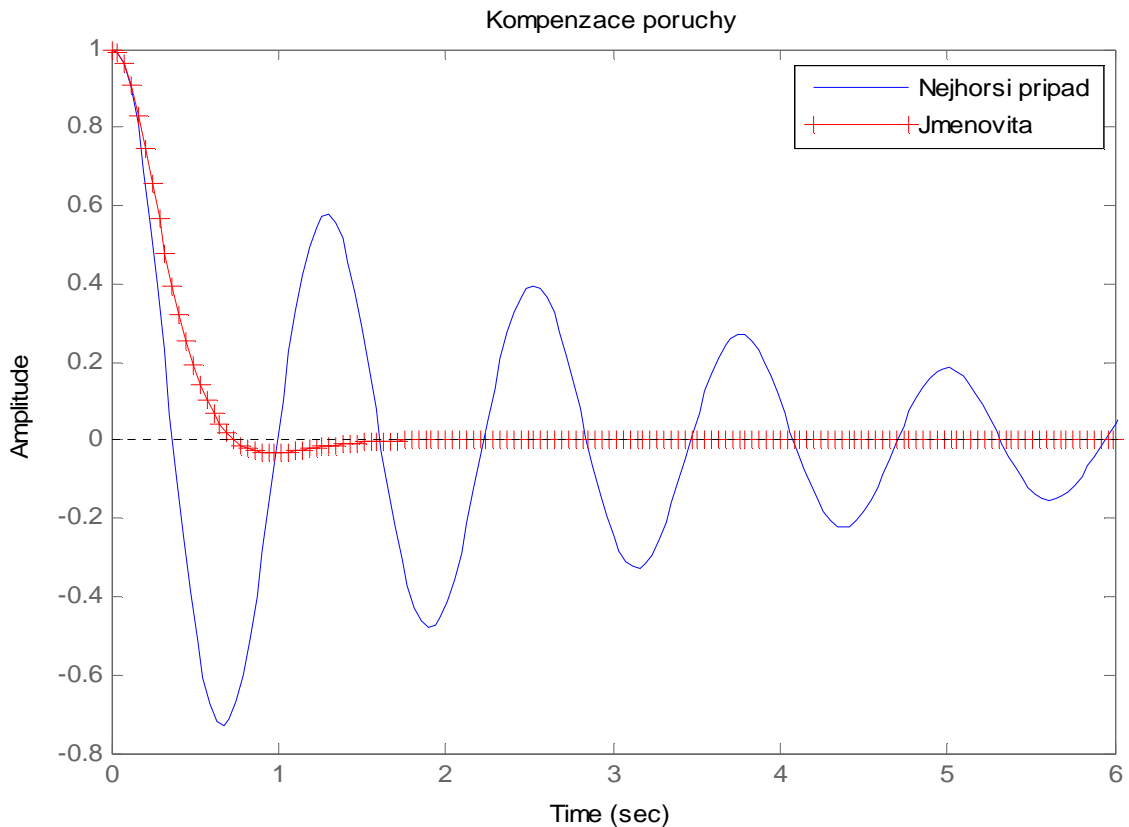
```
Sg = ufrd(S,om);
[maxgain,worstuncertainty] = wcgain(Sg);
maxgain
```

S usubs funkcí můžeme nahradit hodnoty v proměnné nejhorší neurčitosti *worstuncertainty* do neurčité citlivostní funkce *S*. Toto dává nejhorší případ citlivostní funkce *Sworst* v celém rozsahu neurčitosti. Všimněte si, že špička zesílení *Sworst* odpovídá dolní mezi vypočítané funkcí *wcgain*.

```
Sworst = usubs(S,worstuncertainty);
Sgworst = frd(Sworst,Sg.Frequency);
norm(Sgworst,inf)
maxgain.LowerBound
```

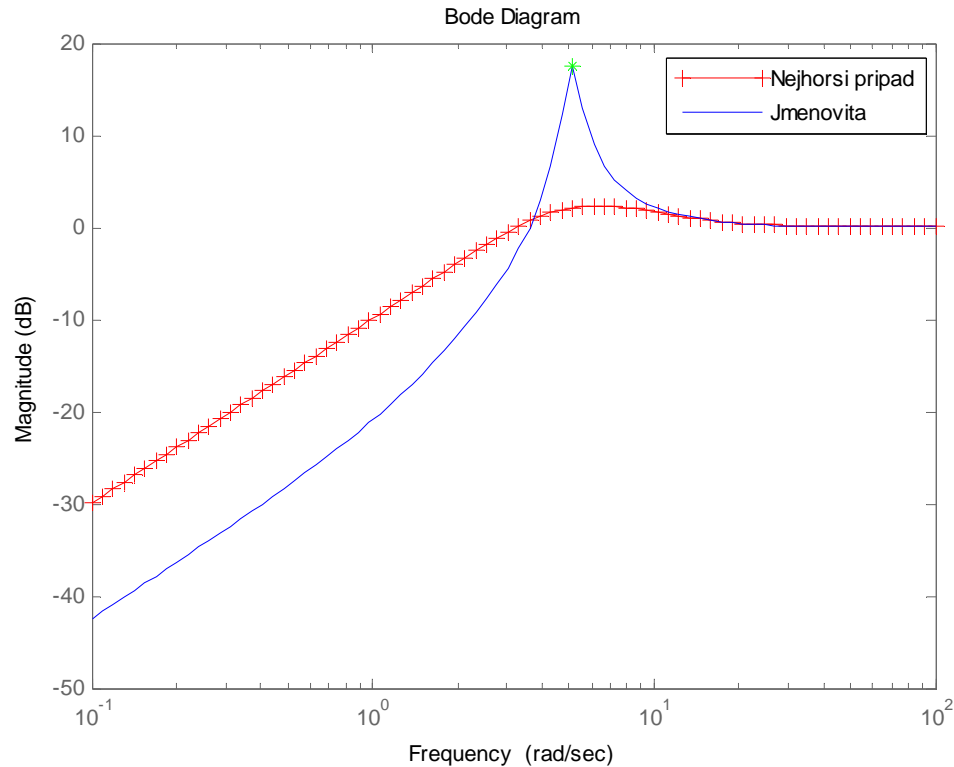
Nyní srovnáme odezvy na jednotkový skok nominální a nejhoršího případu citlivosti:

```
step(Sworst,'b',S.NominalValue,'r-+',6);
title('Kompensace poruchy')
legend('Nejhorsi pripad','Jmenovita')
```



Obr. 10 Graf pro jmenovitý a nejhorší případ kompenzace poruchy ve tvaru jednotkového skoku

Nakonec, pojďme vykreslit Bodeho amplitudové charakteristiky nominální a nejhorší hodnoty citlivostní funkce. Všimněte si, že špičková hodnota $Sworst$ nastane v kmitočtu $maxgain.CriticalFrequency$:



Obr. 11 Amplituda jmenovité a nejhorší citlivosti

4.3 Polynomial Toolbox

Polynomial toolbox je balíček pro systémy, signály, řídicí analýzy a návrhy založené na moderních polynomiálních metodách. Byl vytvořen firmou Polyx, Ltd. přímo pro MATLAB a je zpoplatněn. Sestává z více než 200 M-souborů v MATLAB kódu a je snadno ovladatelný. Tento toolbox není součástí programového prostředí MATLAB, proto se musí po stažení nainstalovat způsobem, který byl popsán v kapitole 3.5.

Toolbox má širokou škálu využití, kterou zde nebudeme vypisovat, vzpomeňme pro nás jen ty nejdůležitější:

- jednoduché vložení, manipulace, zobrazení polynomů a polynomiálních matic na základě nového polynomiálního objektu matice,
- klasická a robustní analýza LTI systémů a filtrů,
- robustní řízení s parametrickou neurčitostí.

Toolbox používá nové originální algoritmy, které jsou rychlé a spolehlivé. Četný počet konvertorů umožňuje přímou spolupráci s Control System Toolbox a

Symbolic Math Toolbox. SIMULINKová sada bloků pro LTI systémy popsané polynomiálními maticovými zlomky je také k dispozici.

Každá relace Polynomial Toolboxu začíná inicializačním příkazem *pinit*. Tato funkce vytvoří globální polynomiální proměnné a přiřadí jim jejich výchozí hodnoty.

4.3.1 Příklady využití Polynomial Toolboxu

Polynomial Toolbox nabízí několik jednoduchých nástrojů, které jsou užitečné pro robustní analýzu a návrh u systémů s parametrickou neurčitostí.

Jednparametrická neurčitost

Mnoho systémů s praktickým významem obsahuje jediný neurčitý parametr. V době návrhu je o parametru známo pouze, že leží v daném intervalu. Docela často i složitější problémy (se složitější strukturou neurčitosti) mohou být redukovány na případ s jediným parametrem. Netřeba dodávat, že nejlepší výsledky toolboxu jsou k dispozici pro tento jednoduchý případ.

I když neurčitý parametr je pouze jeden, může se objevit v několika koeficientech přenosové matice ve stejnou dobu. Zcela v duchu Polynomial Toolboxu jsou koeficienty považovány za polynomiální funkce neurčitého parametru.

Pro analýzu neurčitého polynomu s jedním parametrem zjistíme jeho interval robustní stability. Daný polynom má tvar:

$$p(s, q) = 3 + (10 + q)s + 12s^2 + (6 + q)s^3 + s^4 \quad (33)$$

Nejprve zkontrolujeme, zda $p(s, q)$ je stabilní pro $q = 0$. Poté je nutné najít jeho levostrannou a pravostrannou mez stability, tj. takové nejmenší negativní q_{\min} a největší kladné q_{\max} , při kterém $p(s, q)$ zůstane stabilní, tedy pro libovolné $q \in [q_{\min}, q_{\max}]$.

S Polynomial Toolboxem je to snadný úkol: Nejprve vyjádříme daný polynom jako $p(s, q) = p_0(s) + qp_1(s)$ a vložíme data:

```
p0 = 3 + 10*s + 12*s^2 + 6*s^3 + s^4;
p1 = s + s^3;
```

K prověření jmenovité stability (tj. stability při $q = 0$) zavoláme příkaz:

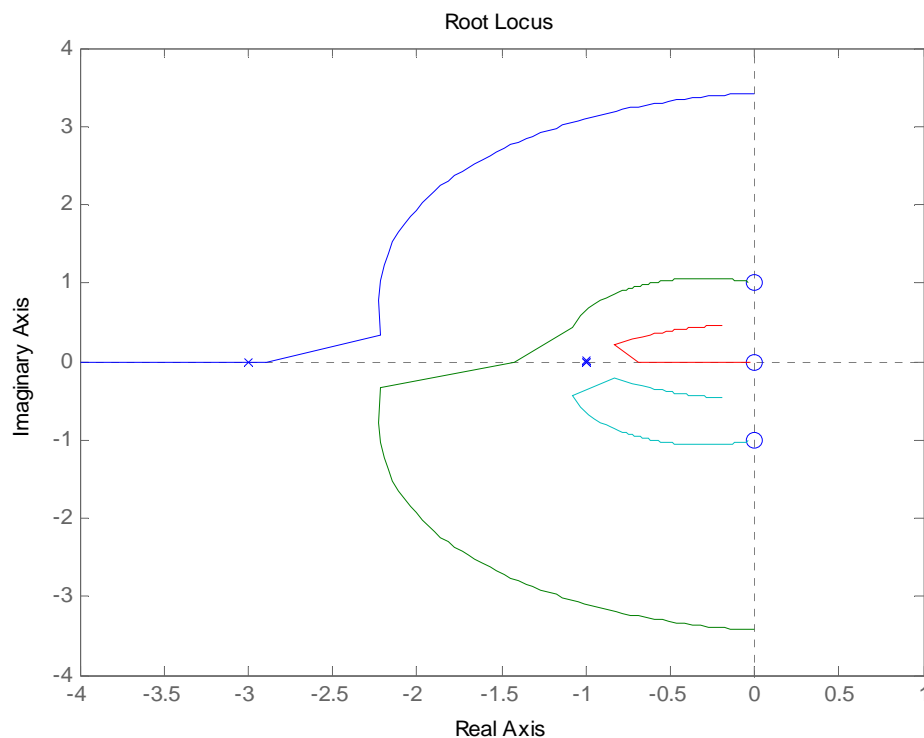
```
[qmin, qmax] = stabint(p0, p1)
```

```
qmin = -5.6277
qmax = Inf
```

Tento výsledek ukazuje, že $p(s, q)$ je nejen stabilní po $q = 0$, ale také pro všechna $q \in (-5.6277, \infty)$. Při $q = -5.6277$ je stabilita ztracena. Nic není proklamováno pro $q < -5.6277$, jen to, že stabilita není garantována. Pokud máte také k dispozici Control System Toolbox, pak můžete propojit využití s Polynomial Toolboxem a vizualizovat výsledek vynesemím grafu umístění kořenů fiktivního modelu $p_1(s)/p_0(s)$ pod fiktivním zpětnovazebním zesílením q v rozmezí $(-5.6277, \infty)$. Příkazy:

```
rlocus(ss(p1, p0), qmin:.1:100)
```

vytvoříme graf umístění kořenů (Obr. 12). To potvrzuje, že všechny kořeny $p(s, q) = p_0(s) + qp_1(s)$ zůstanou v oblasti stability pro všechna $q \in (-5.6277, 100)$. Všimněte si také role makra `ss`, které převádí polynomiální zlomek do stavového formátu pro Control System Toolbox.



Obr. 12 Graf umístění kořenů

Intervalové polynomy

Další významná třída neurčitých systémů je popsána intervalovými polynomy s nezávislými neurčitostmi v koeficientech. Intervalový polynom je definován jako

$$p(s, q) = \sum_{i=0}^n [q_i^-, q_i^+] s^i \quad (34)$$

kde $[q_i^-, q_i^+]$ označuje hranice intervalu i -tého koeficientu. Použitím Polynomial Toolboxu je vhodné popisovat intervalové polynomy jejich „dolními“ a „horními“ prvky

$$p^-(s) = \sum_{i=0}^n q_i^- s^i \quad \text{a} \quad p^+(s) = \sum_{i=0}^n q_i^+ s^i \quad (35)$$

V mnoha aplikacích se objevují intervalové polynomy. Jsou to většinou polynomy, u kterých je původní neurčitá struktura známa, ale je příliš složitá (např. vysoce nelineární), aby byla řešitelná, avšak může být „přehraňována“ pomocí jednoduchého intervalu, jakmile je předepsána nezávislá neurčitá struktura.

Vezměme si spojitý intervalový polynom:

$$p(s, q) = [0.45, 0.55] + [1.95, 2.05]s + [2.95, 3.05]s^2 + [5.95, 6.05]s^3 + [3.95, 4.05]s^4 + [3.95, 4.05]s^5 + s^6 \quad (36)$$

První krok v grafickém testu pro robustní stabilitu požaduje stanovení, že alespoň jeden polynom z rodiny polynomů je stabilní. Pomocí středního bodu každého z intervalů získáme

```
p_mid = pol([0.5 2 3 6 4 4 1], 6)
```

```
isstable(p_mid)
```

Dále zadáme daný interval polynomu jako dva mezní polynomy z obou konců intervalu

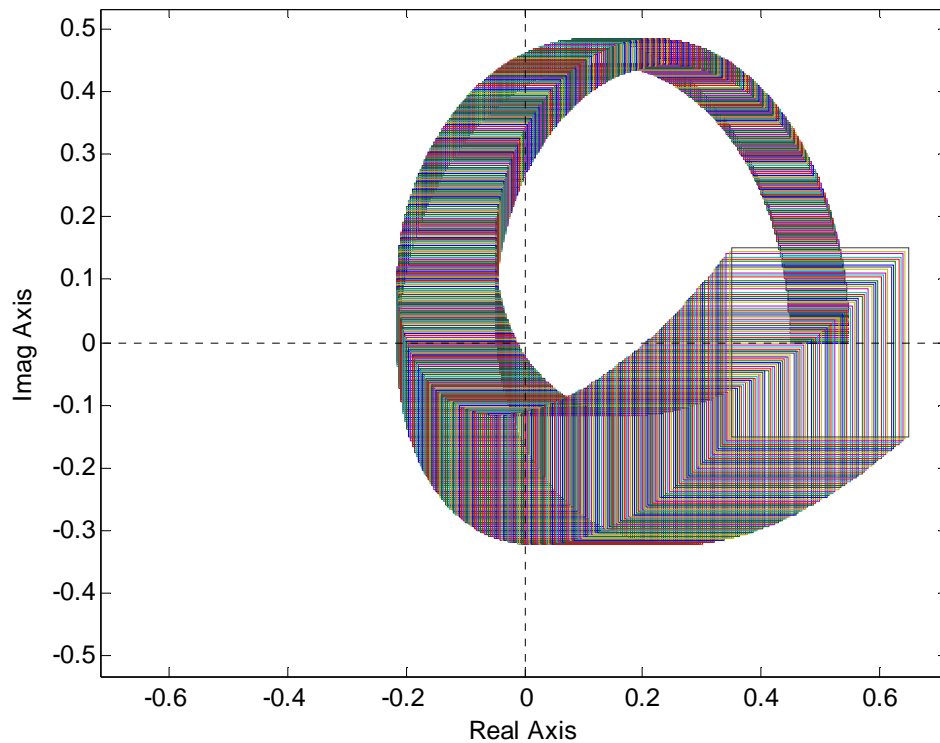
```
pminus = 0.45+1.95*s+2.95*s^2+5.95*s^3+3.95*s^4+3.95*s^5+s^6
```

```
pplus = 0.55+2.05*s+3.05*s^2+6.05*s^3+4.05*s^4+4.05*s^5+s^6
```

Použitím těchto polynomů vykreslíme graf pro $p(j\omega, q)$, skládající se z Kharitonovových obdélníků pro $0 \leq \omega \leq 1$ pomocí příkazu:

```
khplot(pminus, pplus, 0:.001:1)
```

Tímto příkazem vykreslíme níže uvedený graf (Obr. 13). Protože žádný z obdélníků se nedotýká bodu $z = 0$, je splněna podmínka o vyloučení nuly $0 \notin p(j\omega, q)$ a my jsme dospěli k závěru, že intervalový polynom je robustně stabilní. Všimněte si, že dokud všechny koeficienty polynomu jsou reálná čísla, je potřeba zkoumat pouze $\omega \geq 0$. Graf pro $\omega \leq 0$ je symetrický jako $p(-j\omega, q) = \bar{p}(j\omega, q)$.



Obr. 13 Kharitonovovy obdélníky

Pro řešení spojitých intervalových polynomů je k dispozici ještě jednodušší metoda. Intervalový polynom invariantního stupně (s reálnými koeficienty) je stabilní právě tehdy, když jen jeho čtyři „extrémní“ polynomy (tzv. Kharitonovovy polynomy) jsou stabilní.

Pro intervalový polynom $p(s, q)$ se Kharitonovovy polynomy vypočítají příkazem:

```
[stability, K1, K2, K3, K4] = kharit(pminus, pplus)
stability = 1
```

Makro také kontroluje stabilitu Kharitonovových polynomů. Výsledná hodnota proměnné *stability* potvrzuje, že všechny čtyři polynomy jsou stabilní a můžeme dospět k závěru, že intervalový polynom je robustně stabilní.

Robustní stabilita diskrétních intervalových polynomů

U diskrétních polynomů 4 a vyššího stupně nejsou Kharitonovovy extrémální výsledky k dispozici, avšak grafickou metodu lze použít pro diskrétní polynomy stejně dobře jako u jiných oblastí stability.

Uvažujme intervalový polynom:

$$p(z, q) = [10, 20] + [20, 30]z + [128, 138]z^2 + [260, 270]z^3 + 168z^4 \quad (37)$$

Abychom mohli otestovat jeho robustní stabilitu, tak si ho rozepíšeme jako

$$p(z, q) = p_0(z) + q_1 p_1(z) + q_2 p_2(z) + q_3 p_3(z) + q_4 p_4(z) \quad (38)$$

Kde: $p_0(z) = 10 + 20z + 128z^2 + 260z^3 + 168z^4$, $p_1(z) = z$, $p_2(z) = z^2$, $p_3(z) = z^2$, $p_4(z) = z^3$

Takový výraz se nazývá polytopický. Umožňuje nám popsat každý intervalový koeficient samostatným neurčitým parametrem q_i v rozsahu $[0, 10]$.

K analyzování intervalového polynomu nejprve zadáme data

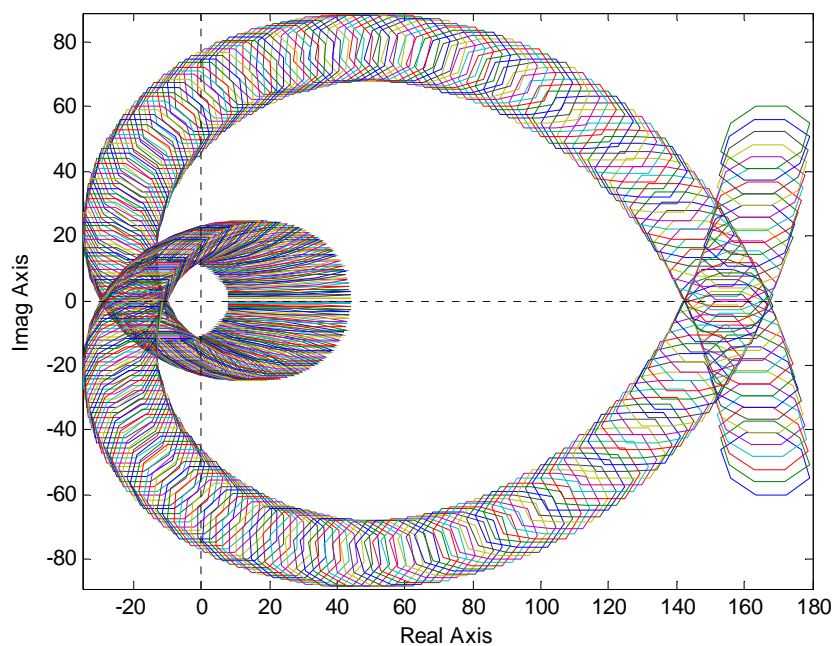
```
p0 = 10 + 20*z + 128*z^2 + 260*z^3 + 168*z^4;
p1 = 1; p2 = z; p3 = z^2; p4 = z^3;
Qbounds = [ 0 10; 0 10; 0 10; 0 10 ];
```

Dále zjistíme, zda je p_0 stabilní

```
isstable(p0)
ans = 1
```

Poté vykreslíme soubor hodnot $p(c, q)$, ale nyní s c obíhající kolem jednotkové kružnice. Všimněte si, že hodnoty již nemají obdélníkový tvar, a proto je třeba použít obecnější příkaz k obdržení následujícího grafu (Obr. 14). Pro lepší zobrazení toho co se děje v okolí bodu 0 si přiblížíme graf, pro generalizované frekvence $e^{j\omega}$ v kritickém rozmezí $\omega \in (0.6\pi, 1.4\pi)$:

```
ptopplot(p0, p1, p2, p3, p4, Qbounds, exp(j*(0:0.001:1)*2*pi))
```



Obr. 14 Přiblížená množina hodnot polynomu

Jak můžeme vidět na obrázku (Obr. 14), podmínka o vyloučení nuly je splněna. Nula nenáleží žádnému z osmiúhelníků (pro všechna c na jednotkové kružnici $0 \notin p(c, q)$) a můžeme učinit závěr, že diskretní časový interval polynom je robustně stabilní.

Polytopy polynomů

Obecnější typ systémů je popsán neurčitými polynomy, jejichž koeficienty závisí lineárně na několika parametrech, ale každý parametr se může objevit současně v několika koeficientech. Takový neurčitý polynom může vypadat jako:

$$p(s, q) = \sum_{i=1}^n a_i(q)s^i \quad (39)$$

kde každý koeficient $a_i(q)$ je afinní funkcí q . Tzn., že pro každé $i \in \{0, 1, 2, \dots, n\}$ existuje sloupcový vektor a_i a skalár b_i takový že platí $a_i(q) = a_i^T q + b_i$.

Neurčité polynomy s afinní neurčitou strukturou v prostor polynomů tvoří polytopy. Podobně jako v jednoparametrickém případě mohou tyto polynomy být vždy vyjádřeny jako: $p(s, q) = p_0(s) + q_1 p_1(s) + q_2 p_2(s) + \dots + q_n p_n(s)$

Tato forma je preferována v Polynomial toolboxu. Tedy polytop polynomů s n parametry je vždy popsán $n+1$ polynomy $p_0(s), p_1(s), \dots, p_n(s)$ a n hraničními intervaly parametrů $[q_1^-, q_1^+], [q_2^-, q_2^+], \dots, [q_n^-, q_n^+]$. Abychom zachovali invariantní stupeň v celém polytopu, tak se obvykle předpokládá, že $\deg p_0(s) \geq \deg p_1(s)$ pro všechna i . Jedním z důvodů proč afinní lineární neurčitá struktura je tak důležitá je, že je zachována v rámci zpětnovazebního propojení. Afinní lineární neurčitá struktura je také zachována při lineární transformaci s a má i mnoho dalších zajímavých funkcí.

Nyní si ukážeme menší vylepšení obdélníkových mezí. Pro polytop polynomů p popsany rovnicí

$$p(s, q) = (q_1 - 2q_2 + 2) + (q_2 + 1)s + (2q_1 - q_2 + 4)s^2 + (2q_2 + 1)s^3 + s^4 \quad (40)$$

Kde $q_1 \in [-0.5, 2]$ a $q_2 \in [-0.3, 0.3]$. Provedeme schválně dva druhy analýz robustní stability.

První robustní analýza bude provedena pomocí Kharitonovových polynomů.

Takže nejprve nahradíme $p(s, q)$ přehraničujícím intervalovým polynomem $\bar{p}(s, \bar{q})$ popsaném jako:

$$\bar{p}(s, \bar{q}) = [0.9, 4.6] + [0.7, 1.3]s + [2.7, 8.3]s^2 + [0.4, 1.6]s^3 + s^4 \quad (41)$$

Zadáním příkazu pro použití Kharitonovových polynomů:

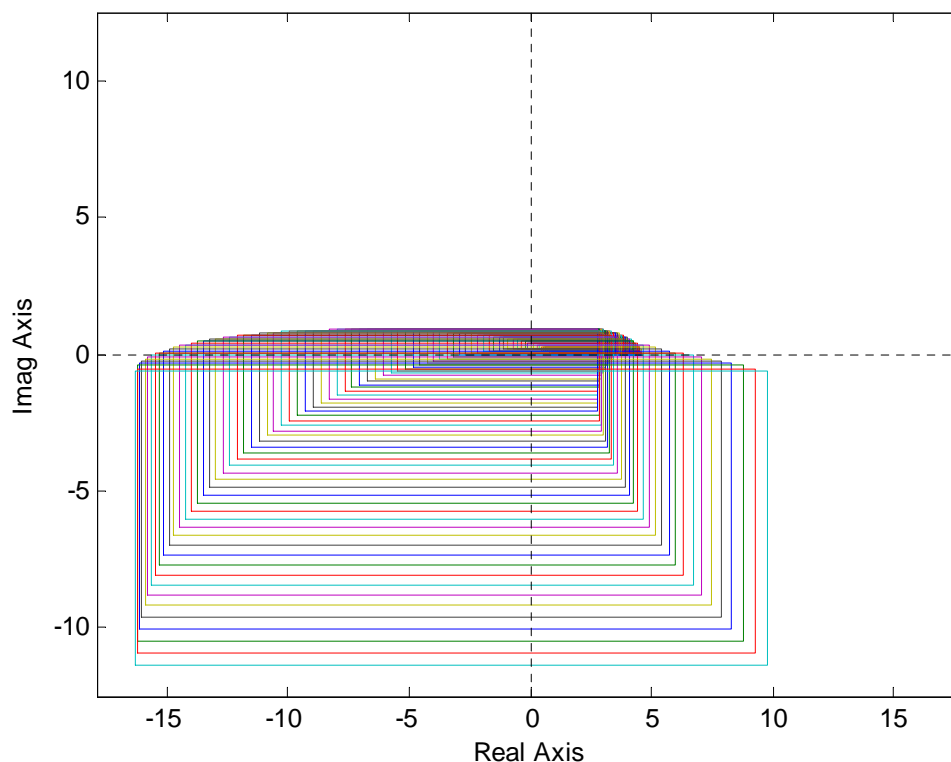
```
pminus = pol([0.9 0.7 2.7 0.4 1],4);
pplus = pol([4.6 1.3 8.3 1.6 1],4);
[stable,K1,K2,K3,K4] = kharit(pminus,pplus); stable
stable = 0
```

V proměnné *stable* vyšla nula, a tudíž zjišťujeme, že polynom $p(s, q)$ není robustně stabilní. Lehce si můžeme ověřit, že třetí Kharitonův polynom není stabilní pomocí příkazu:

```
isstable(K3)
ans = 0
```

Na závěr pro názornost vykreslíme Kharitonovovy obdelníky příkazem:

```
khplot(pminus,pplus,0:0.025:2)
```



Obr. 15 Kharitonovovy obdelníky

Druhý způsob robustní analýzy je pomocí srovnání souboru hodnot.

Nyní si vyjádříme polynom $p(s, q)$ jako $p(s, q) = p_0(s) + q_1 p_1(s) + q_2 p_2(s)$,

kde:

$$p_0(s) = 2 + s + 4s^2 + s^3 + s^4$$

$$p_1(s) = 1 + 2s^2$$

$$p_2(s) = -2 + s - s^2 + 2s^3$$

Data zadáme pomocí příkazu

```
p0 = pol([2 1 4 1 1], 4);  
p1 = pol([1 0 2], 2);  
p2 = pol([-2 1 -1 2], 3);  
Qbounds = [-0.5 2; -0.3 0.3]
```

Dále ověříme rozhodující předpoklad pro uplatnění podmínky o vyloučení nuly.

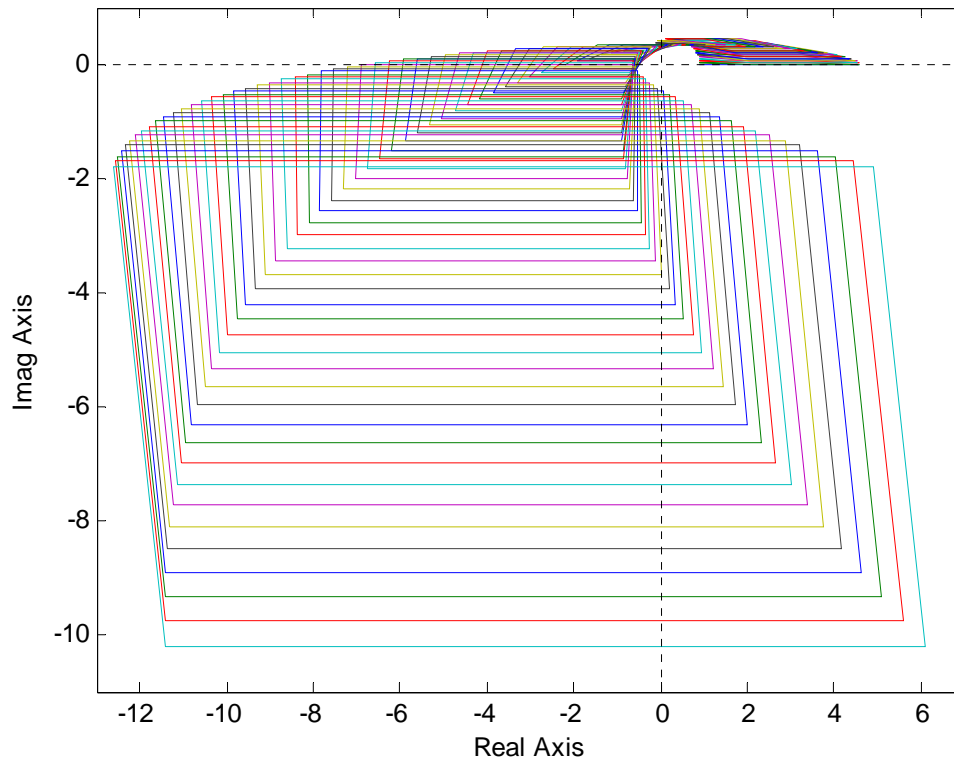
Opavdu $p_0(s)$ je stabilní člen rodiny daného intervalového polynomu jak vyplývá z

```
isstable(p0)  
ans = 1
```

Následně vygenerujeme 80 polygonálních hodnot odpovídajících frekvencím rovnoměrně rozložených mezi $\omega=0$ a $\omega=2$:

```
ptopplot(p0, p1, p2, Qbounds, j*(0:0.025:2))
```

V rámci výpočtových mezí jsme došli k závěru z grafu (Obr. 16), že $0 \notin p(j\omega, q)$ pro všechna $\omega \leq 0$. Proto podle podmínky o vyloučení nuly, můžeme dojít k závěru, že p je robustně stabilní.



Obr. 16 Extremální polygony

Avšak z grafu 15 (Obr. 15) je jasné, že podmínka o vyloučení nuly nebyla splněna pro Kharitonovy obdélníky, i když platí pro polygony na obrázku výše (Obr. 16).

Shrneme-li to, práce s „přehrančujícím“ intervalovým polynomem je neprůkazná. Zatímco při práci s polygonálními hodnotami nás vede k jednoznačnému závěru, že $p(s, q)$ je robustně stabilní.

Přesněji řečeno, výsledek analýzy robustní stability je při použití tohoto „přehrančujícího“ polynomu zbytečně konzervativní, jinými slovy podmínka je „pouze“ postačující (čili silnější), nikoliv nutná a postačující jako u použití polynomu s původní strukturou. Tedy pokud nám u „přehrančujícího“ polynomu vyjde, že je robustně stabilní, tak víme, že robustně stabilní musí být i původní rodina (s afinní lineární, resp. polytopickou strukturou neurčitosti). Když nám ale vyjde, že ten „přehrančující“ je robustně nestabilní, tak nám to o té originální rodině neřekne nic a museli bychom použít jinou analýzu. [18]

4.4 YALMIP

YALMIP je modelovací jazyk pro pokročilé modelování a řešení konvexní a nekonvexních optimalizačních problémů. Je implementován jako volný (bezplatný) toolbox pro MATLAB.

Hlavním důvodem pro používání YALMIPu je rychlý vývoj algoritmů. Programovací jazyk je v souladu se syntaxemi MATLABu, díky tomu je pro každého, kdo je obeznámen s příkazy v MATLABu, jednoduché používat tento toolbox.

Další výhodou YALMIPu je, že implementuje velké množství způsobů modelování, což umožňuje uživateli soustředit se na modely vyšších řádů, zatímco YALMIP se stará o syntézu nízkých řádů. Samozřejmě snaží se o získání co nejefektivnějších a numericky nejvýhodnějších modelů.

Modelovací jazyk podporuje velké množství optimalizačních tříd, jako je lineární, kvadratická, semidefinitní, geometrická, lokální a globální polynom a samozřejmě multiparametrické a robustní programování.

Jednou z ústředních myšlenek v YALMIPu je zaměřit se na programovací jazyk a algoritmy vyšších řádů. Zároveň se opírá o vnější řešitele pro samotné výpočty. Nicméně YALMIP také implementuje interní algoritmy pro globální optimalizaci a smíšené celočíselné programování, multiparametrické programování a robustní optimalizaci. Tyto algoritmy jsou obvykle založeny na nízkourovňovém programovacím jazyku, který je k dispozici v YALMIPu a řeší dílčí problémy pomocí externích řešitelů.

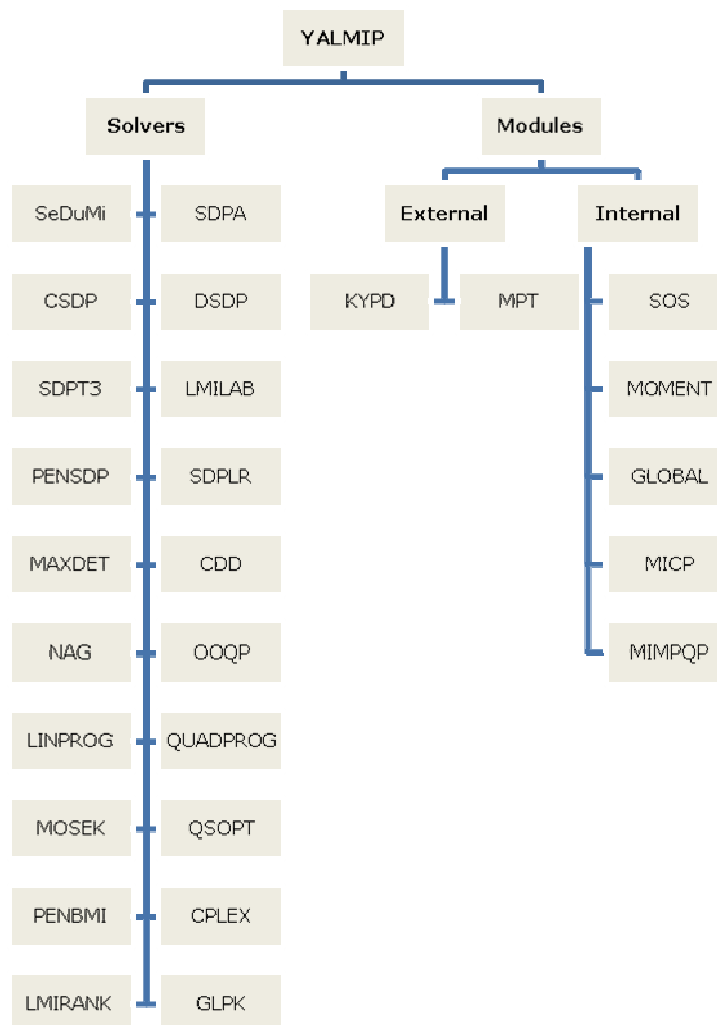
YALMIP je výhradně založen na m-kódu, a proto se také lehce instaluje. Odstraňte všechny staré verze YALMIPu, pokud máte již nějaký v PC, rozbalte soubor yalmip.zip, který stáhnete na jeho domovských webových stránkách [13] a přidejte následující adresáře do MATLABu:

```
/yalmip
/yalmip/extras
/yalmip/demos
/yalmip/solvers
/yalmip/modules
/yalmip/modules/parametric
/yalmip/modules/moment
/yalmip/modules/global
/yalmip/modules/sos
/yalmip/operators
```

Pro kontrolu své instalace použijte příkaz `yalmiptest`.

Externí řešitele musíme také nainstalovat. Postupy pro instalaci jednotlivých řešitelů jsou popsány v jejich manuálech. Hlavní zásadou je však hlavně nezapomenout přidat cestu k jejich umístění do MATLABu.

Na obrázku (Obr. 17) můžeme vidět kompletní seznam aktuálně využívaných externích řešitelů a interních modulů.



Obr. 17 Přehled modulů a externích řešitelů pro YALMIP

Samozřejmě není třeba vlastnit všechny tyto řešitele. YALMIP zvládne řešit zadanou úlohu více způsoby, pokud nenalezne příslušného řešitele, vytvořeného přímo pro řešenou úlohu. Např. lineární programování může být řešeno kvadratickým programováním, které může být zase řešeno kónickým programováním druhého řádu a to nakonec můžeme znovu obejít semidefinitním programováním. Teoreticky tedy potřebujeme jen řešitele na semidefinitní programování. Pokud však chceme řešit větší problémy nebo jiné druhy

problémů je lepší stáhnout více řešitelů a vyzkoušet, který nejlépe zvládne zadaný problém. [13]

4.5 Ostatní toolboxy pro robustní řízení a Simulink

Jak bylo zmíněno na začátku této kapitoly, toolboxů pro robustní řízení existuje celá řada. Ty nejnávštěvnější jsme zmínili v předešlých podkapitolách. Ale samozřejmě za zmínku stojí i ty ostatní, které jsou méně sofistikované a nenabízejí uživateli tolik možností pro analýzu. Vytvoříme si zde takový menší seznam těchto méně známých toolboxů s několika větičkami a odkazem, na kterém je možnost získat více informací.

- **SeDuMi** - jeden z nejoblíbenějších volně dostupných solverů pro semidefinitní programování a LMI. Dostupný zdarma, vyvinut nedávno předčasně zesnulým nizozemským matematikem Josem Sturmem. Vývoj pokračuje, sledovat a stáhnout tento software můžete na jeho domácích internetových stránkách [19].
- **Paradise Toolbox** - volně dostupný toolbox pro návrh robustních regulátorů pro parametrické neurčitosti vyvíjený výzkumníky v DLR v Německu. Bohužel vývoj dále nepokračuje, takže toolbox viditelně stárne. [10]
- **Linear Fractional Representation Toolbox (LFRT) for Matlab** - volně dostupný toolbox pro manipulaci s LFT vyvíjený J.-F.Magni (Onera, Toulouse). Existuje i varianta pro Scilab. [17]
- **Skew μ Toolbox** - volně dostupný toolbox pro vychýlenou μ -analýzu a syntézu vyvíjený J.-M.Biannicem (Onera, Toulouse). [17]
- **Slicot** - sada špičkových numerických kódů napsaných ve Fortranu. Pro řadu funkcí ale již existují verze pro Matlab, případně není těžké si tyto „wrappery“ napsat sám. Některé funkce dokonce použity v poslední verzi Matlabu. [16]
- **Octave** - klon Matlabu pod GNU GPL licencí. Obsahuje i sadu funkcí pro návrh H_{∞} regulátorů. [7]

Ještě na závěr několik málo slov k platformě Simulink. Je to velmi univerzální a důležitý nástroj pro velkou škálu simulací navržených regulátorů pro různé systémy.

Simulink je softwarový balíček, který umožňuje modelovat, simulovat a analyzovat všechny druhy systémů. Pro modelování Simulink poskytuje grafické uživatelské rozhraní (GUI), ve kterém si uživatel sestaví vlastní model z jednotlivých bloků knihovny pomocí tzv. click-and-drag operace myši, což je kliknutí na blok a tažení bloku na místo, kde se blok má v modelu nacházet. Simulink zahrnuje rozsáhlou knihovnu bloků jako je knihovna zdrojů, knihovna lineárních a nelineárních prvků a mnoho dalších. Dále si uživatel může vytvořit i vlastní bloky. Simulink může být využit ke zkoumání chování celé řady dynamických systémů reálného světa, včetně elektrických, mechanických a termodynamických systémů. První vydání Simulinku obsahovala verze MATLABu 3.5 v roce 1990.

Simulovat dynamický systém se skládá z dvou kroků. Za prvé, uživatel vytvoří blokové schéma s využitím Simulink model editoru, který graficky zobrazuje časové závislé matematické vztahy mezi vstupy, výstupy a stavy. Blokové schéma se sestaví z bloků, které zahrnuje knihovna Simulinku. Po vytvoření blokového schématu nebo modelu uživatel přikáže Simulinku, aby provedl simulaci systému, který je zastoupený modelem za určitých počátečních a koncových hodnot času. Pro zobrazení výsledku simulace se nejčastěji používá blok Scope z knihovny Sinks.

ZÁVĚR

Diplomová práce se zabývá problematikou specializovaných toolboxů pro robustní řízení. Klade si za cíl vytvořit přehled existujících toolboxů s menší ukázkou jejich možností a funkcí pro usnadnění analýzy a syntézy dynamických neurčitých systémů.

Práce se nejprve věnuje podrobnějšímu vysvětlení základních pojmů souvisejících s robustní stabilitou a její problematikou v otázce analýzy robustní stability systémů s neurčitostmi. Dále popisuje klasifikaci neurčitých systémů s parametrickou a neparametrickou neurčitostí spolu s nadefinovanými pojmy týkající se vyšetřování robustní stability systémů pro jednotlivé struktury neurčitosti. U některých struktur neurčitosti jsou uvedeny její typické analytické nástroje pro testování robustní stability.

Pro usnadnění těchto analýz a syntéz neurčitých systému jsou v praktické části představeny různé typy specializovaných toolboxů. Nejsou opomíjeny ani toolboxy dostupné zdarma, ale jsou představeny jen stručně, přeci jen např. komerční Robust Control Toolbox nabízí mnohem větší škálu nástrojů pro různé druhy neurčitostí a není jen dedikován na jeden typ neurčitosti, jakož tomu je u některých volně dostupných toolboxů. Avšak to, že komerční toolbox nabízí větší škálu funkcí, nemusí nutně znamenat, že bude vždy a všech směrech nejlepší. Úlohy robustní stability jsou tak různorodé, že není možné sestrojít toolbox, který zvládne řešit všechny tyto úlohy nejlepším způsobem.

ZÁVĚR V ANGLIČTINĚ

Diploma thesis deals with the problem of specialized toolboxes for robust control. It aims to create an overview of existing toolboxes with a small example of their capabilities and functions to facilitate the analysis and synthesis of uncertain dynamic systems.

First, we discuss a more detailed explanation of basic concepts related to the robust stability and the issues regarding robust stability analysis of systems with uncertainties. It also describes the classification of uncertain systems with parametric and non-parametric uncertainties along with the concepts relating to the investigation of robust stability of systems for each uncertainty structure. For some uncertainty structures are set out its typical analytical tools for the robust stability testing.

In the practical part are introduced different types of specialized toolboxes for facilitation of the analysis and synthesis of uncertain systems. There are not forgotten even toolboxes available for free, but they are presented only briefly, after all for example the commercial Robust Control Toolbox provides a much wider range of tools for different types of uncertainties and it is not only dedicated to the one type of uncertainty, as is the case with some freely available toolboxes. But that a commercial toolbox offers a greater range of features does not necessarily mean that it is always and in all aspects the best. Robust stability tasks are so diverse that it is not possible to construct a toolbox that can solve all.

SEZNAM POUŽITÉ LITERATURY

- [1] BARMISH, B. *New tools for robustness of linear systems*. New York: Maxwell Macmillan International, c1994, xvi, 394 p. ISBN 00-230-6055-7.
- [2] BHATTACHARYYA, S. P., H. CHAPPELLAT a L. H. KEEL. *Robust control: The parametric approach*. New Jersey: Prentice Hall, Englewood Cliffs, 1995.
- [3] BIALAS, S. A necessary and sufficient condition for stability of interval matrices. *International Journal of Control*, vol. 37, 1983, pp. 717n722.
- [4] BOYD, S., L. EL GHAOU, E. FERON a V. BALAKRISHNAN. *Linear Matrix Inequalities in System and Control Theory*. Philadelphia: Society for Industrial and Applied Mathematics, 1994. SIAM studies in applied mathematics, 15. ISBN 0-89871-334-X.
- [5] DOSTÁL, P. a R. MATUŠŮ. *Stavová a algebraická teorie řízení*. Zlín: Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2010. ISBN 978-80-7318-991-4.
- [6] DULLERUD, G. E. a F. G. PAGANINI. *A course in robust control theory: a convex approach*. New York: Springer, c2000, xx, 417 p. ISBN 0387989455.
- [7] *GNU Octave* [online]. © 1998-2012 [cit. 2012-05-27]. Dostupné z: <http://www.gnu.org/software/octave/>
- [8] HENRION, D. *Courses for robust control* [online]. Toulouse, 2005 [cit. 2012-05-25]. Dostupné z: <http://homepages.laas.fr/henrion>
- [9] INRIA. *Scilab website* [online]. © 1989- 2011 [cit. 2012-05-27]. Dostupné z: <http://www.scilab.org/>
- [10] INSTITUT FÜR ROBOTIK UND MECHATRONIK. *DLR - Institut für Robotik und Mechatronik* [online]. © 2012 [cit. 2012-05-27]. Dostupné z: <http://www.dlr.de/rm-neu/desktopdefault.aspx/tabid-4021/>
- [11] KAWAMURA, T. a M. SHIMA. Robust stability of characteristic polynomials with monotonicity. *Nonlinear Analysis: Theory, Methods & Applications*. 1997, Vol. 30, No. 8, pp. 5109-5119.
- [12] KHARITONOV, V. L. Asymptotic stability of an equilibrium position of a family of systems of linear differential equations. *Differential Equations*. 1979, č. 14, pp. 1483-1485.

- [13] LÖFBERG, J. *YALMIP Wiki* [online]. 2003 [cit. 2012-05-25]. Dostupné z: <http://users.isy.liu.se/johanl/yalmip/>
- [14] MATUŠŮ, R. Aplikace robustních algoritmů při řízení systémů s periodicky perturbovanými parametry. In: *Instruments and control: proceedings of XXVIII seminary ASR'03 : Ostrava, May 6, 2003*. 1. vyd. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, Fakulta strojní, Katedra automatizační techniky a řízení, 2003, s. 213. ISBN 80-248-0326-7.
- [15] MUNRO, N. *Symbolic methods in control system analysis and design*. London: Institution of Electrical Engineers, c1999, xviii, 393 p. ISBN 08-529-6943-0.
- [16] NICONET. *NICONET* [online]. 1998 [cit. 2012-05-27]. Dostupné z: <http://www.icm.tu-bs.de/NICONET/index.html>
- [17] ONERA. *ONERA - the French Aerospace Lab* [online]. © 2012 [cit. 2012-05-27]. Dostupné z: <http://www.onera.fr/toulouse/>
- [18] POLYX, Ltd. *The Polynomial Toolbox for MATLAB* [online]. 1999 [cit. 2012-05-27]. Dostupné z: <http://www.polyx.com/>
- [19] *SeDuMi* [online]. 2004 [cit. 2012-05-27]. Dostupné z: <http://sedumi.ie.lehigh.edu/>
- [20] SKOGESTAD, S. a I. POSTLETHWAITE. *Multivariable feedback control: analysis and design*. 2nd repr. ed. Chichester: Wiley, 2008, xiv, 574 s. ISBN 0-470-01167-X.
- [21] ŠEBEK, M. *Robustní systémy: PDF podklady pro předmět robustní systémy* [online]. ČVUT Praha, 2002 [cit. 25. května 2012]. Dostupné z: http://www.polyx.com/_robust/
- [22] ŠEBEK, M., M. HROMČÍK a J. JEŽEK. Polynomial toolbox 2.5 and systems with parametric uncertainties. *Proceedings of 3rd IFAC Symposium on Robust Control Design*. Prague, 2000.
- [23] THE MATHWORKS, Inc. *Control System Toolbox* [online]. ©1984-2012 [cit. 2012-05-25]. Dostupné z: <http://www.mathworks.com/products/control/>
- [24] THE MATHWORKS, Inc. *Robust Control Toolbox - Documentation* [online]. ©1984-2012 [cit. 2012-05-25]. Dostupné z: <http://www.mathworks.com/help/toolbox/robust/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

$A(q)$	Neurčitá matice.
α_i	Sloupcový vektor.
α_i^T	Transponovaná matice.
β_i	Skalární číslo.
C	Konvexní množina.
$C(s)$	Neznámá matice.
$\text{conv}\{C\}$	Konvexní obal množiny.
\det	Determinant.
δ	Obecná frekvenční proměnná.
$D(s)$	Stabilní polynomiální matice.
Im	Imaginární část komplexní roviny.
K_i	Kharitonovův polynom.
P	Rodina polynomů.
p_0	Polynom.
P	Polytop.
q	Vektor neurčitých parametrů.
q_i	Reálný neurčitý parametr.
Q	Množina omezující parametry.
ω	Frekvence polynomu.

SEZNAM OBRÁZKŮ

<i>Obr. 1 Příklad konvexní a nekonvexní množiny</i>	27
<i>Obr. 2 Příklady konvexních obalů</i>	27
<i>Obr. 3 Hlavní okno programu MATLAB</i>	37
<i>Obr. 4 Přechodové charakteristiky modelu systému a uzavřené smyčky</i>	43
<i>Obr. 5 Analýza odezvy otevřené smyčky na jednotkový skok</i>	45
<i>Obr. 6 Bodeho graf otevřené smyčky stejnosměrného motoru</i>	46
<i>Obr. 7 Analýza robustnosti uzavřené smyčky</i>	47
<i>Obr. 8 Amplituda citlivostní funkce S</i>	48
<i>Obr. 9 Kompenzace poruchy ve tvaru jednotkového skoku</i>	49
<i>Obr. 10 Graf pro jmenovitý a nejhorší případ kompenzace poruchy ve tvaru jednotkového skoku.....</i>	50
<i>Obr. 11 Amplituda jmenovité a nejhorší citlivosti</i>	51
<i>Obr. 12 Graf umístění kořenů</i>	53
<i>Obr. 13 Kharitonovovy obdélníky.....</i>	55
<i>Obr. 14 Přiblížená množina hodnot polynomu</i>	56
<i>Obr. 15 Kharitonovovy obdélníky.....</i>	58
<i>Obr. 16 Extremální polygony.....</i>	60
<i>Obr. 17 Přehled modulů a externích řešitelů pro YALMIP</i>	62