

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra matematiky

## Diplomová práce

# Robustní přiřazení pólů stavovou a výstupní zpětnou vazbou

# Prohlášení

Prohlašuji, že jsem diplomovou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 22. května 2013

Jana Königsmarková

# Abstract

## Robust pole assignment by static state feedback and static output feedback

This master's thesis studies the problem of robust stabilization by static state and static output feedback. We look for an optimal feedback that assigns the required Jordan form to the closed-loop system and that makes the closed-loop system sufficiently robust or is not fragile. We describe the algorithms for solving real and complex stability radius and use these functions as the main criteria functions for optimizing properties of the closed-loop system. These functions together with the approach for computation of the explicit parametrization of state and output feedback creates a base for optimization. In this thesis we design a new algorithm for solving robust Jordan form assignment by output feedback. As a part of this thesis we create the library for solving the above problems and implement them in the Matlab toolbox.

**Keywords:** static state feedback, static output feedback, pole assignment, Jordan form, robust stability, fragility, singular values, optimization, Hamiltonian matrix

# Abstrakt

## Robustní přiřazení pólů stavovou a výstupní zpětnou vazbou

Tato diplomová práce se zabývá problémem robustní stabilizace stavovou a výstupní zpětnou vazbou. Chceme nalézt nejlepší zpětnou vazbu, která bude uzavřenému systému přiřazovat požadovanou Jordanovu formu a pro níž bude uzavřený systém dostatečně robustní nebo která nebude křehká. Seznámíme se s algoritmy na výpočet reálného a komplexního poloměru stability a využijeme je dále jako hlavní kritériální funkce při optimalizaci vlastností uzavřeného systému. Tyto funkce spolu s přístupem pro výpočet explicitní parametrizace slouží jako základ optimalizace. Dále navrhujeme nový algoritmus na výpočet robustního přiřazení Jordanovy formy výstupní zpětnou vazbou. Součástí práce je vytvoření knihovny funkcí v softwaru Matlab, pomocí které budeme moci řešit příklady z uvedené problematiky.

**Klíčová slova:** stavová zpětná vazba, výstupní zpětná vazba, přiřazení pólů, Jordanova forma, robustní stabilita, křehkost regulátoru, singulární čísla, optimalizace, Hamiltonova matice

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Přípravná kapitola</b>	<b>4</b>
2.1	Seznam značení . . . . .	4
2.2	Úvodní pojmy . . . . .	5
2.3	Hamiltonova matice, perturbace a pomocné věty . . . . .	7
2.4	Numerická analýza a výpočet vlastních čísel matice . . . . .	11
2.4.1	Numerický výpočet vlastních čísel Hamiltonovy matice . . . . .	14
2.5	Metody pro hledání optima . . . . .	16
2.5.1	Metoda zlatého řezu . . . . .	16
2.5.2	Line search metody . . . . .	18
2.5.3	Kvazi-Newtonovské metody, metoda BFGS . . . . .	18
2.6	Metody pro řešení nelineárních rovnic . . . . .	22
<b>3</b>	<b>Přiřazení Jordanovy formy zpětnou vazbou</b>	<b>25</b>
3.1	Stavová zpětná vazba . . . . .	25
3.2	Výstupní zpětná vazba . . . . .	31
<b>4</b>	<b>Robustnost uzavřeného systému</b>	<b>35</b>
4.1	Kritéria robustnosti . . . . .	36
4.2	Poloměr stability . . . . .	38
4.2.1	Komplexní poloměr stability . . . . .	38
4.2.2	Reálný poloměr stability . . . . .	52
4.3	Pseudospektrum . . . . .	58
4.3.1	Komplexní pseudospektrum . . . . .	58
4.3.2	Reálné pseudospektrum . . . . .	60
4.4	$H_\infty$ norma . . . . .	61
4.5	Ostatní uvažované funkce . . . . .	63
4.5.1	Lokace vlastních čísel . . . . .	63
4.5.2	Regulátor s omezenou strukturou . . . . .	64
<b>5</b>	<b>Návrh robustního přiřazení Jordanovy formy</b>	<b>65</b>
5.1	Přiřazení stavovou zpětnou vazbou . . . . .	68
5.2	Přiřazení výstupní zpětnou vazbou . . . . .	70

5.2.1	Návrh algoritmu . . . . .	72
<b>6</b>	<b>Vytvořený Toolbox v Matlabu</b>	<b>78</b>
<b>7</b>	<b>Příklady</b>	<b>86</b>
7.1	Tříhmotový systém . . . . .	86
7.2	Model robotu jednonožky - systém 10. řádu . . . . .	87
7.3	Model robotu jednonožky - zjednodušení . . . . .	93
<b>8</b>	<b>Závěr</b>	<b>97</b>
	<b>Literatura</b>	<b>100</b>

# 1 Úvod

Jedním z významných problémů v teorii lineárních systémů je přiřazení pólů zpětnou vazbou. Zpětnou vazbou se snažíme přiřadit systému námi požadované chování, které budeme určovat zvolenými póly. Uvažujme systém  $(A, B, C)$ , kde  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$ , popsany diferenciální rovnicí

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx,\end{aligned}$$

kde  $x$  reprezentuje stav systému,  $u$  představuje jeho vstup a  $y$  výstup. Vlastní dynamické chování tohoto systému je dáno jeho autonomní<sup>1</sup> částí (tj.  $\dot{x} = Ax$ ), konkrétně vlastními čísly a vlastními vektory matice  $A$ . Protože chceme získat stabilní systém, musíme všechny póly uzavřeného systému umístit do otevřené levé poloroviny komplexní roviny. K tomuto účelu můžeme využít zpětnou vazbu. V naší práci budeme navrhovat stavovou zpětnou vazbu  $F$  ( $u = Fx$ ) a výstupní zpětnou vazbu  $K$  ( $u = Ky$ ).

Pro uvedený systém tak chceme nalézt stavovou zpětnou vazbu  $F$ , resp. výstupní zpětnou vazbu  $K$ , aby matice dynamiky uzavřeného systému  $A_c = A + BF$ , resp.  $A_c = A + BKC$ , měla požadovanou Jordanovu formu. Takových zpětných vazeb je obecně nekonečně mnoho. V předkládané práci se budeme zabývat problémem návrhu stavové či výstupní zpětné vazby, která bude daný systém jednak stabilizovat a navíc v jistém smyslu optimalizovat.

Prvky matice dynamiky otevřeného systému jsou určeny hodnotami veličin reálného modelu, které neznáme přesně. Proto je rozumné uvažovat, že vstupní matice  $A$  může být zatížena jistými perturbacemi ( $\Delta$ ). Pro systém  $(A, B, C)$  tak nestačí jen najít zpětnou vazbu, která jej stabilizuje (která mu pomocí Jordanovy formy přiřadí požadované chování), ale je vhodné zajistit, aby i při působení co největších perturbací zůstal uzavřený systém stabilní. Budeme se tedy snažit vybrat takovou matici  $F$ , resp.  $K$ , aby póly uzavřené smyčky zůstaly v otevřené levé polorovině komplexní roviny i při uvažovaných poruchách. Požadujeme tak, aby pro všechny uvažované poruchy  $\Delta$  zůstala stabilní matice,

$$A_c + \Delta = (A + \Delta) + BF = (A + BF) + \Delta, \text{ resp.}$$

$$A_c + \Delta = (A + \Delta) + BKC = (A + BKC) + \Delta.$$

Je-li tato podmínka splněna, hovoříme o robustní stabilitě systému.

Obdobně uvažujme o přesnosti vypočtené zpětné vazby. Její realizace v číslicovém regulátoru je možná jen na určitý počet platných cifer daný formou zobrazení

---

<sup>1</sup>system bez vstupu

reálných čísel v použitém procesoru. Z tohoto důvodu bychom chtěli mít zajištěno, že i při určitých změnách v prvcích matice zpětné vazby se vlastní čísla matice dynamiky uzavřeného systému budou nacházet v otevřené levé polorovině komplexní roviny. Zpětnou vazbu, která by již při malých poruchách narušila stabilitu uzavřené smyčky, označujeme jako křehkou (fragilní<sup>2</sup>). Chceme-li tedy zabránit křehkosti zpětné vazby, požadujeme stabilitu matice

$$A_c + B\Delta = A + B(F + \Delta) = (A + BF) + B\Delta, \text{ resp.}$$

$$A_c + B\Delta C = A + B(K + \Delta)C = (A + BKC) + B\Delta C.$$

Pro posouzení, jak daleko od nestability se uzavřený systém nachází, budeme používat tzv. komplexní či reálný poloměr stability.

V článkách [7, 19] se autoři zabývají podobnou problematikou nalezení robustně stabilního systému. V [7] je navrhován regulátor, který stabilizuje daný systém se současnou minimalizací  $H_\infty$  normy přenosu uzavřené smyčky a dalším požadavkem na stabilní regulátor. Součástí těchto prací je také volně dostupný<sup>3</sup> program HIFOO<sup>4</sup> sestavený v softwaru Matlab. Optimalizovanými proměnnými jsou zde přímo všechny prvky matice výstupní zpětné vazby a může jich tak být obecně velmi mnoho. Navíc při změnách prvků zpětnovazební matice nemáme zaručeno, že uvažovaná kritériální funkce bude definována, protože uzavřený systém při změně prvků regulátoru nemusí zůstat stabilní. Podobný přístup je uvažován i v práci [19]. Zde se autoři snaží vyhnout této situaci přidáním vedlejší podmínky, která omezuje hodnotu největšího záporného vlastního čísla uzavřeného systému. Výpočtem některých základních robustních kritérií se zabývají i práce [31] a [15]. V [31] se autoři soustředí na vykreslení komplexního pseudospektra zejména pro velké řídké pevné matice a umožňují pro ně určit hodnotu pseudospektrální abscissy. V [15] je počítán například komplexní poloměr stability matic, ale bez návrhu regulátoru.

V naší práci zvolíme jiný přístup, založený na minimální parametrizaci množiny všech stavových, resp. výstupních, zpětných vazeb přiřazujících požadovanou Jordánovu formu, která vedle stability zajišťuje i požadované chování uzavřené smyčky. Kromě toho budeme uvažovat další požadavky na maximální robustnost uzavřeného systému, minimální křehkost regulátoru a některé další praktické požadavky.

Práce je členěna do čtyř hlavních částí. V první z nich je představena základní teorie potřebná pro uvedení do zmiňované problematiky, dále stručně pojednáváme o optimalizačních metodách využívaných při řešení nastíněných úloh.

V druhé části navážeme na návrh explicitní parametrizace z [25] a [16] a popíšeme nový algoritmus pro nalezení parametrické matice  $Q(\alpha)$ , která vede na minimální

---

<sup>2</sup>z lat. fragilis, e - křehký, nestálý

<sup>3</sup>[www.cs.nyu.edu/overton/software/hifoo](http://www.cs.nyu.edu/overton/software/hifoo)

<sup>4</sup>HIFOO : H-Infinity Fixed-Order Optimization

parametrizaci množiny zpětných vazeb a usnadňuje řešení problém. V třetí části si uvedeme zvolená kritéria pro posouzení robustnosti systému či fragility regulátoru a představíme možnosti jejich výpočtu.

Ve čtvrté části budeme formulovat konkrétní problém robustního přiřazení a navrhnout metody pro jeho řešení. Představíme návrh nového numerického algoritmu pro robustní přiřazení Jordanovy formy výstupní zpětnou vazbou a demonstrujeme jej na několika příkladech. Součástí předkládané práce bude vytvoření knihovny funkcí v softwaru Matlab pro řešení uvedené problematiky. V závěru práce budeme ilustrovat využití knihovny na několika praktických příkladech, například modelu robotu jednožky.



## 2 Přípravná kapitola

### 2.1 Seznam značení

$\square$	ukončení důkazu
$\triangleq$	definiční relace
$\mathbb{R}$	množina reálných čísel
$\mathbb{C}$	množina komplexních čísel
$\forall$	pro všechna
$\exists$	existuje
$\Rightarrow$	implikuje
$A \in \mathbb{R}^{n \times m}$	$A$ je reálná matice typu $n \times m$
$I_n$	jednotková matice typu $n \times n$
$A^{-1}$	inverzní matice k matici $A$
$A^T$	transponovaná matice $A$
$A^*$	hermitovsky sdružená matice $A$
$\det(A)$	determinant matice $A$
$\text{rank}(A)$	hodnost matice $A$
$\text{diag}\{L_1, \dots, L_k\}$	blokově diagonální matice s diagonálními bloky $L_1, \dots, L_k$
$\Lambda(A)$	spektrum matice $A$ , množina všech vlastních čísel matice $A$
$\sigma(A)$	singulární číslo matice $A$
$\sigma_1(A)$	největší singulární číslo matice $A$
$\sigma_{\min}(A)$	nejmenší singulární číslo matice $A$
$\text{Re}(x)$	reálná část komplexního čísla $x$
$\text{Im}(x)$	imaginární část komplexního čísla $x$
$\partial\mathbb{C}_g$	hranice oblasti $\mathbb{C}_g$
$\langle a, b \rangle$	uzavřený interval od $a$ do $b$
$o$	nulový vektor

## 2.2 Úvodní pojmy

V teorii řízení nás u systémů často zajímá, zda jsou stabilní, případně, jak daleko se nachází od nejbližšího nestabilního systému. O stabilitě systému vypovídají vlastní čísla matice dynamiky. Vzdálenost mezi systémy pak budeme posuzovat podle normy perturbací, které ze stabilního systému udělají nestabilní. Uved'me si proto nejprve některé základní pojmy, se kterými budeme dále pracovat.

**Definice 2.1.** (Norma vektoru) Necht'  $\mathcal{V}$  je vektorový prostor nad tělesem komplexních (resp. reálných) čísel. Normou vektoru  $x \in \mathcal{V}$  rozumíme funkci, která přiřadí každému vektoru  $x \in \mathcal{V}$  nezáporné číslo  $\|x\|$  a splňuje následující podmínky:

- 1)  $\|x\| > 0$  pro  $x \neq o$
- 2)  $\|\alpha x\| = |\alpha| \|x\|$ , pro  $\alpha \in \mathbb{C}$  (resp.  $\mathbb{R}$ )
- 3)  $\|x + y\| \leq \|x\| + \|y\|$ ,  $\forall x, y \in \mathcal{V}$ .

V této práci budeme používat Euklidovskou normu vektoru, definovanou jako

$$\|x\|_2 = \left( \sum_{i=1}^n |x_i|^2 \right)^{\frac{1}{2}}, \quad (2.1)$$

kde  $x_i$  je  $i$ -tá složka vektoru  $x \in \mathbb{C}^n$ .

Normu matice  $A \in \mathbb{C}^{n \times n}$  budeme měřit maticovou normou generovanou Euklidovskou normou (2.1)

$$\|A\|_2 = \sup_{x \neq o} \frac{\|Ax\|_2}{\|x\|_2} = \max_{\|x\|_2=1} \|Ax\|_2. \quad (2.2)$$

Pokud nebude specifikováno jinak, budeme pro matici v následujících kapitolách symbolem  $\|\cdot\|$  označovat její maticovou normu generovanou Euklidovskou normou.

**Definice 2.2.** Necht'  $A \in \mathbb{R}^{n \times n}$ .

- a) Matice  $A$  je symetrická, jestliže  $A = A^T$ .
- b) Matice  $A$  je ortogonální, jestliže  $A^T A = A A^T = I_n$ .

**Definice 2.3.** Necht'  $A \in \mathbb{C}^{n \times n}$ .

- a) Matice  $A$  je hermitovská, jestliže  $A = A^*$ .
- b) Matice  $A$  je unitární, jestliže  $A^* A = A A^* = I_n$ .

**Definice 2.4.**

- a) Necht'  $A \in \mathbb{R}^{n \times n}$ . Řekneme, že  $A = [a_{i,j}]_{i,j=1}^n$  je horní trojúhelníková matice, jestliže  $a_{i,j} = 0$  pro  $i > j$ .

- b) Necht'  $A \in \mathbb{R}^{n \times n}$ . Řekneme, že matice  $A$  je kvazi-trojúhelníková, je-li blokově horní trojúhelníková s bloky na diagonále typu  $1 \times 1$  nebo  $2 \times 2$ .
- c) Necht'  $A \in \mathbb{R}^{n \times n}$ . Řekneme, že matice  $A = [a_{i,j}]_{i,j=1}^n$  je v horním Hessenbergově tvaru, jestliže  $a_{i,j} = 0$  pro  $i > j - 1$ .
- d) Necht'  $A \in \mathbb{R}^{n \times n}$ . Řekneme, že matice  $A = [a_{i,j}]_{i,j=1}^n$  je v dolním Hessenbergově tvaru, jestliže  $a_{i,j} = 0$  pro  $i < j - 1$ .

**Definice 2.5.** Necht'  $A$  je čtvercová matice  $n$ -tého řádu. Polynom  $p(\lambda) = \det(\lambda I - A)$  nazýváme charakteristickým polynomem matice  $A$ . Kořeny polynomu  $p(\lambda)$  se nazývají vlastní čísla matice  $A$ . Soubor všech vlastních čísel matice  $A$  se nazývá spektrum matice  $A$  a označuje se  $\Lambda(A)$ . Necht'  $\lambda \in \Lambda(A)$ , potom nenulový vektor  $v \in \mathbb{C}^n$  takový, že  $(\lambda I - A)v = 0$ , tj.  $Av = \lambda v$ , je vlastním vektorem matice  $A$  příslušný vlastnímu číslu  $\lambda$ .

Vlastní čísla jsou tedy kořeny příslušného charakteristického polynomu. Protože pro výpočet kořenů polynomů pátého a vyššího stupně neexistuje obecný vzorec, nelze ani vlastní čísla matice těchto řádů vypočítat v konečném počtu kroků.

Numerické metody na výpočet vlastních čísel se liší podle toho, zda chceme nalézt všechna vlastní čísla (úplný problém), nebo jen některá (potom se jedná o částečný problém).

**Definice 2.6.** Řekneme, že matice  $A$  a  $B$  jsou podobné, jestliže existuje regulární matice  $T$  taková, že  $A = TBT^{-1}$ . Označujeme  $A \sim B$ .

**Věta 2.1.** *Jsou-li matice  $A, B$  podobné, mají stejná vlastní čísla.*

**Důkaz.** Jestliže jsou matice  $A, B$  podobné, existuje regulární matice  $T$  a lze psát  $A = TBT^{-1}$ . Potom  $\det(\lambda I - A) = \det(\lambda I - TBT^{-1}) = \det[T(\lambda I - B)T^{-1}] = \det(T) \det(\lambda I - B) \det(T^{-1}) = \det(\lambda I - B)$ .  $\square$

**Definice 2.7.** Necht'  $A$  je matice typu  $n \times m$ . Odmocniny z nenulových vlastních čísel matice  $A^*A$  (nebo  $AA^*$ ) nazýváme singulárními čísly  $A$  a označujeme  $\sigma(A)$ , tj.

$$\sigma_i = \sqrt{\lambda_i(A^*A)}, \quad (2.3)$$

kde  $i = 1, \dots, k$  a  $k = \text{rank}(A)$ .

Singulární čísla můžeme uspořádat sestupně :  $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots \geq \sigma_k > 0$ .

**Věta 2.2.** *Necht'  $A \in \mathbb{C}^{m \times n}$  a  $\text{rank}(A) = k$ . Potom existují unitární matice  $U \in \mathbb{C}^{m \times m}$ ,  $V \in \mathbb{C}^{n \times n}$  takové, že matici  $A$  lze vyjádřit jako součin*

$$A = U\Sigma V^*,$$

kde  $\Sigma = \text{diag}\{\sigma_1, \sigma_2, \dots, \sigma_k, 0, \dots, 0\} \in \mathbb{C}^{m \times n}$ .

**Definice 2.8.** Necht'  $v = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ ,  $p_1(v), p_2(v), \dots, p_m(v)$  představují polynomy s reálnými koeficienty v proměnných  $x_1, x_2, \dots, x_n$ . Varietou  $\mathcal{V}$  v  $\mathbb{R}^n$  nazýváme množinu

$$\mathcal{V} = \{v \in \mathbb{R}^n : p_i(v) = 0, i = 1, \dots, m\}.$$

## 2.3 Hamiltonova matice, perturbace a pomocné věty

V této podkapitole si uvedeme důležité věty, které dávají do souvislosti normu matic, singulární čísla a perturbace v prvcích matic.

Některé metody, které počítají hodnoty kritéria robustnosti, využívají vztahu mezi singulárními čísly jistých matic a vlastními čísly odpovídajících Hamiltonových matic. Zavedeme si proto Hamiltonovu matici a zmíníme některé její vlastnosti.

**Definice 2.9.** Matici  $M \in \mathbb{C}^{2n \times 2n}$ , která splňuje podmínku  $JM = (JM)^*$ , kde  $J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$  se nazývá Hamiltonova matice.

**Věta 2.3.** Jestliže  $\lambda$  je vlastním číslem Hamiltonovy matice  $M$ , potom je jejím vlastním číslem také  $-\lambda$ .

**Důkaz.** Mějme matici  $J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$ . V následujících rovnostech využijeme, že  $JM = -M^*J$ ,  $\det(J) = 1$  a pro  $A, B \in \mathbb{C}^{m \times m}$  je  $\det(AB) = \det(A)\det(B)$ .

$$\begin{aligned} p_M(\lambda) &= \det(\lambda I - M) = \det(J) \det(\lambda I - M) = \det(\lambda J - JM) = \\ &= \det(\lambda J + M^*J) = \det(\lambda I + M^*) \det(J) = \det(\lambda I + M^*) = \\ &= \det(\lambda I + M)^* = \det(M - (-\lambda I))^* = p_M(-\lambda). \end{aligned}$$

□

**Věta 2.4.** Necht'  $M \in \mathbb{C}^{2n \times 2n}$ ,  $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ , kde  $A, B, C, D$  jsou čtvercové matice řádu  $n$ . Potom  $M$  je Hamiltonova matice, pokud jsou matice  $B, C$  hermitovské a  $A + D^* = 0$ .

**Důkaz.**

$$JM = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} C & D \\ -A & -B \end{bmatrix},$$

$$(JM)^* = \begin{bmatrix} C^* & -A^* \\ D^* & -B^* \end{bmatrix} = \begin{bmatrix} C & D \\ -A & -B \end{bmatrix}.$$

Protože  $(JM) = (JM)^*$ , je matice  $M$  podle definice 2.9 Hamiltonova.  $\square$

**Věta 2.5.** (Sylvestrova věta o determinantech) Necht' matice  $M \in \mathbb{C}^{p \times q}$  a  $N \in \mathbb{C}^{q \times p}$ , potom platí

$$\det(I - MN) = \det(I - NM). \quad (2.4)$$

**Důkaz.** Vyjdeme z výrazu  $\det(I - MN)$ , který budeme dále upravovat,

$$\det(I - MN) = \det \left( \begin{bmatrix} I & 0 \\ M & I - MN \end{bmatrix} \right) = \det \left( \begin{bmatrix} I & N \\ M & I \end{bmatrix} \begin{bmatrix} I & -N \\ 0 & I \end{bmatrix} \right).$$

Nyní uijeme, že pro čtvercové matice  $A, B$  platí  $\det(AB) = \det(A)\det(B)$ . Potom

$$\det \left( \begin{bmatrix} I & N \\ M & I \end{bmatrix} \begin{bmatrix} I & -N \\ 0 & I \end{bmatrix} \right) = \det \left( \begin{bmatrix} I & N \\ M & I \end{bmatrix} \right) \det \left( \begin{bmatrix} I & -N \\ 0 & I \end{bmatrix} \right).$$

Dále upravme matici  $\begin{bmatrix} I & -N \\ 0 & I \end{bmatrix}$ . Vynásobíme druhý řádek této blokové matice maticí  $N$  zleva a přičteme k prvnímu řádku. Tím hodnotu determinantů nezměníme,

$$\det \left( \begin{bmatrix} I & N \\ M & I \end{bmatrix} \right) \det \left( \begin{bmatrix} I & -N \\ 0 & I \end{bmatrix} \right) = \det \left( \begin{bmatrix} I & N \\ M & I \end{bmatrix} \right) \det \left( \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \right) =$$

$$\det \left( \begin{bmatrix} I & N \\ M & I \end{bmatrix} \right) \det(I) \det(I) = \det \left( \begin{bmatrix} I & N \\ M & I \end{bmatrix} \right) \cdot 1 \cdot 1 = \det \left( \begin{bmatrix} I & N \\ M & I \end{bmatrix} \right).$$

Poslední matici opět upravíme vynásobením zleva maticí  $N$  druhého řádku a přičtením k prvnímu, dostaneme

$$\det \left( \begin{bmatrix} I & N \\ M & I \end{bmatrix} \right) = \det \left( \begin{bmatrix} I - NM & 0 \\ M & I \end{bmatrix} \right) = \det(I - NM).$$

Postupnými úpravami jsme získali vztah (2.4).  $\square$

**Věta 2.6.** Necht'  $A \in \mathbb{C}^{m \times n}$ , platí

$$\|A\|_2 = \sigma_1(A). \quad (2.5)$$

**Důkaz.** Ze vztahu (2.2) pro normu matice a věty 2.2 plyne

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2 = \max_{\|x\|_2=1} \|U\Sigma V^*x\|_2.$$

Protože  $U$  a  $V$  jsou unitární matice a po substituci  $y = V^*x$  ( $\|x\|_2 = \|Vy\|_2 = \|y\|_2$ ) dostaneme, že

$$\max_{\|x\|_2=1} \|U\Sigma V^*x\|_2 = \max_{\|y\|_2=1} \|\Sigma y\|_2.$$

Předchozí rovnost můžeme dále rozepsat (pro  $\text{rank}(A) = k$ ) podle vztahu (2.1)

$$\begin{aligned} \|A\|_2 &= \max_{\|y\|_2=1} \|\Sigma y\|_2 = \\ &= \max_{\|y\|_2=1} \sqrt{|\sigma_1 y_1|^2 + \dots + |\sigma_k y_k|^2 + |0 \cdot y_{k+1}|^2 + \dots + |0 \cdot y_n|^2} = \\ &= \max_{\|y\|_2=1} \sqrt{\sigma_1^2 |y_1|^2 + \dots + \sigma_k^2 |y_k|^2} \leq \max_{\|y\|_2=1} \sqrt{\sigma_1^2 (|y_1|^2 + \dots + |y_n|^2)} = \sigma_1(A). \end{aligned}$$

Nakonec pro vektor  $y = [1 \ 0 \ \dots \ 0]^T$ , tj.  $x = v_1$ , se maxima přímo dosahuje, tedy celkem  $\|A\|_2 = \sigma_1(A)$ .  $\square$

**Věta 2.7.** *Necht'  $A \in \mathbb{C}^{m \times n}$  a  $\text{rank}(A) = n$ , platí*

$$\min_{\|x\|_2=1} \|Ax\|_2 = \sigma_n(A). \quad (2.6)$$

**Důkaz.** Analogicky jako ve větě (2.6) bychom dostávali

$$\begin{aligned} \|Ax\|_2 &= \|U\Sigma V^*x\|_2 = \|\Sigma V^*x\|_2 = \|\Sigma y\|_2 = \\ &= \sqrt{|\sigma_1 y_1|^2 + \dots + |\sigma_n y_n|^2} = \sqrt{\sigma_1^2 |y_1|^2 + \dots + \sigma_n^2 |y_n|^2} \geq \sigma_n. \end{aligned}$$

Pro vektor  $y = [0 \ 0 \ \dots \ 1]^T$  se minima nabývá.  $\square$

**Věta 2.8.** *(Perturbace v součtu) Necht'  $M \in \mathbb{C}^{m \times n}$  a  $\text{rank}(M) = n$ , platí*

$$\min_{\Delta \in \mathbb{C}^{m \times n}} \{\|\Delta\|_2 : \text{rank}(M + \Delta) < n\} = \sigma_n(M). \quad (2.7)$$

**Důkaz.** Protože  $\text{rank}(M + \Delta) < n$ , existuje nenulový jednotkový vektor  $x$ , že  $(M + \Delta)x = o$ . Potom je  $\|Mx\|_2 = \|\Delta x\|_2$  a podle věty 2.7 získáme  $\|\Delta x\|_2 \leq \sigma_n(M)$ . Dále, protože platí  $\|\Delta\|_2 \|x\|_2 \geq \|\Delta x\|_2 \geq \sigma_n(M)$ , je  $\|\Delta\|_2 \geq \sigma_n(M)$ . Nakonec ukažme, že pro nějakou matici  $\Delta$  takovou, že  $\text{rank}(M + \Delta) < n$ , nastává rovnost. Pokud zvolíme  $\Delta = -\sigma_n u_n v_n^*$ , je

$$(M + \Delta)v_n = \sigma_n u_n - \sigma_n u_n v_n^* v_n = \sigma_n u_n - \sigma_n u_n = o,$$

tedy  $\text{rank}(M + \Delta) < n$  a pro matici  $\Delta$  platí i to, že  $\|\Delta\|_2 = \sigma_n(M)$ .  $\square$

**Věta 2.9.** (Perturbace v součinu) Necht'  $M \in \mathbb{C}^{m \times p}$ , platí

$$\min_{\Delta \in \mathbb{C}^{p \times m}} \{\|\Delta\|_2 : \text{rank}(I - M\Delta) \leq m - 1\} = \frac{1}{\sigma_1(M)}. \quad (2.8)$$

**Důkaz.** Matice  $(I - M\Delta) \in \mathbb{C}^{m \times m}$  a  $\text{rank}(I - M\Delta) \leq m - 1$ , tj. matice  $(I - M\Delta)$  je singulární. Existuje tedy vektor  $x \in \mathbb{C}^m$ ,  $x \neq o$  takový, že  $(I - M\Delta)x = o$ . Potom

$$\begin{aligned} Ix &= M\Delta x \quad \text{a} \\ \|x\|_2 &= \|M\Delta x\|_2 \leq \|M\|_2 \|\Delta x\|_2 = \sigma_1(M) \|\Delta x\|_2, \quad \text{tedy} \\ \|x\|_2 &\leq \sigma_1(M) \|\Delta x\|_2. \quad \text{Protože } x \neq o \text{ a podle (2.2) dostaneme} \\ \frac{1}{\sigma_1(M)} &\leq \frac{\|\Delta x\|_2}{\|x\|_2} \leq \|\Delta\|_2. \end{aligned}$$

Tím jsme obdrželi  $\|\Delta\|_2 \geq \frac{1}{\sigma_1(M)}$ . Nyní ukažme, že pro nějakou matici  $\Delta$  zde bude platit rovnost. Necht' singulární rozklad matice  $M$  je ve tvaru  $M = \sum_{i=1}^k \sigma_i u_i v_i^*$ , (pro  $\text{rank}(M) = k$ ). Zvolme  $\Delta = \frac{1}{\sigma_1(M)} v_1 u_1^*$ . Potom pro  $x = u_1$  máme

$$(I - M\Delta)x = (I - M\Delta)u_1 = \left( I - M \frac{v_1 u_1^*}{\sigma_1(M)} \right) u_1 = u_1 - \frac{Mv_1}{\sigma_1(M)}$$

a protože podle věty 2.2 je  $\frac{Mv_1}{\sigma_1(M)} = u_1$  bude

$$u_1 - \frac{Mv_1}{\sigma_1(M)} = u_1 - u_1 = o.$$

Z volby matice  $\Delta$  je  $\|\Delta\|_2 = \frac{1}{\sigma_1(M)}$  a dohromady s nerovností  $\|\Delta\|_2 \leq \frac{1}{\sigma_1(M)}$  dostaneme  $\min_{\Delta \in \mathbb{C}^{p \times m}} \{\|\Delta\|_2 : \text{rank}(I - M\Delta) \leq m - 1\} = \frac{1}{\sigma_1(M)}$ .  $\square$

**Věta 2.10.** (Rouchého věta pro polynomy) Necht'  $R(z)$  a  $S(z)$  jsou polynomy,  $\Gamma$  jednoduchá uzavřená křivka v  $\mathbb{C}$ . Je-li

$$|R(z)| > |S(z)| \quad \text{pro každé } z \in \Gamma, \quad (2.9)$$

potom  $R(z)$  a  $R(z) + S(z)$  mají stejný počet nulových bodů uvnitř  $\Gamma$ .

**Věta 2.11.** Mějme polynom

$$\begin{aligned} P(z) &= p_0 + p_1 z + p_2 z^2 + \dots + p_n z^n = \\ &= p_n (z - a_1)^{m_1} (z - a_2)^{m_2} \dots (z - a_r)^{m_r}, \quad (2.10) \end{aligned}$$

kde pro  $k = 1, 2, \dots, r$  jsou  $a_k$  kořeny polynomu  $P(z)$  a  $m_k$  jejich násobnosti. Necht'  $Q(z) = q_0 + q_1 z + q_2 z^2 + \dots + q_n z^n$ , kde  $q_i = p_i + \varepsilon_i$ , pro  $i = 1, 2, \dots, n$ , a  $D_k = \{z \in \mathbb{C} : |z - a_k| < r_k\}$ , kde  $0 < r_k < \min\{|a_k - a_j|\}$  a  $a_k, a_j$  jsou různé kořeny polynomu  $P(z)$ . Potom existuje  $\varepsilon > 0$  takové, že  $|\varepsilon_i| < \varepsilon$  (pro  $i = 1, 2, \dots, n$ ) implikuje, že polynom  $Q(z)$  má právě  $m_k$  nulových bodů uvnitř  $D_k$ .

**Důkaz.** Pro množinu  $D_k \triangleq \{z \in \mathbb{C} : |z - a_k| < r_k\}$  označme  $\partial D_k$  její hranici.  $\partial D_k$  je kompaktní množina v  $\mathbb{C}$ , na níž polynom  $P(z)$  není nulový. Následně tak bude existovat číslo  $\delta_k > 0$  takové, že i  $|P(z)| \geq \delta_k > 0$  pro  $z \in \partial D_k$ . Dále uvažujme  $Q(z) - P(z) \triangleq R(z) = \varepsilon_0 + \varepsilon_1 z + \dots + \varepsilon_n z^n$ . Pro  $z \in \partial D_k$  je

$$\begin{aligned} |R(z)| &= |\varepsilon_0 + \varepsilon_1 z + \dots + \varepsilon_n z^n| \leq |\varepsilon_0| + |\varepsilon_1||z| + \dots + |\varepsilon_n||z^n| \leq \\ &\leq |\varepsilon_0| + |\varepsilon_1|(|z - a_k| + |a_k|) + \dots + |\varepsilon_n|(|z - a_k| + |a_k|)^n \leq \\ &\leq |\varepsilon_0| + |\varepsilon_1|(r_k + |a_k|) + \dots + |\varepsilon_n|(r_k + |a_k|)^n \leq \\ &\leq \varepsilon + \varepsilon(r_k + |a_k|) + \dots + \varepsilon(r_k + |a_k|)^n \leq \varepsilon \sum_{j=0}^n (r_k + |a_k|)^j. \end{aligned}$$

Pro  $M_k \triangleq \sum_{j=0}^n (r_k + |a_k|)^j$  je  $|R(z)| \leq \varepsilon M_k$ . Zvolme  $\varepsilon < \frac{\delta_k}{M_k}$ ,  $M_k > 0, \varepsilon > 0, \varepsilon = \varepsilon(r_k)$ . Potom

$$|Q(z) - P(z)| = |R(z)| \leq \varepsilon M_k < \delta_k \leq |P(z)|,$$

tj.  $|Q(z) - P(z)| < |P(z)|$ . Podle věty 2.10, kde uvažujeme  $R(z) = P(z)$ ,  $S(z) = P(z) - Q(z)$  a  $\Gamma = \partial D_k$ , dostaneme, že polynom  $Q(z)$  má v  $D_k$  přesně  $m_k$  kořenů (právě tolik jako polynom  $P(z)$ ). Neboli, jestliže se koeficienty polynomu změní o perturbace velikosti nejvýše  $\varepsilon$ , posunou se příslušné kořeny perturbovaného polynomu do vzdálenosti ne více než  $r_k$  od původních kořenů  $a_k$ .  $\square$

**Důsledek 2.1.** *Předchozí věta říká, že kořeny polynomu závisí spojitě na jeho koeficientech. Protože vlastní čísla matice jsou podle definice 2.5 kořeny jejího charakteristického polynomu a koeficienty tohoto polynomu jsou spojitě funkce prvků matice, závisí vlastní čísla matice spojitě na jejích prvcích.*

## 2.4 Numerická analýza a výpočet vlastních čísel matice

Vlastní a singulární čísla matic musíme počítat iteračními metodami. Uvažujme pro jednoduchost případ, kdy matice  $A$  má  $n$  lineárně nezávislých vlastních vektorů  $v_1, v_2, \dots, v_n$  odpovídajících vlastním číslům  $\lambda_1, \lambda_2, \dots, \lambda_n$ . Potom je matice  $A$  diagonalizovatelná, tj.  $L = V^{-1}AV$ , kde  $V = [v_1, \dots, v_n]$  a  $L = \text{diag}\{\lambda_1, \dots, \lambda_n\}$ . Následující věta pro tento případ uvádí, že problém výpočtu vlastních čísel vede obecně na špatně podmíněnou úlohu.

**Věta 2.12.** *(Bauer – Fike) Je-li  $\mu$  vlastní číslo matice  $A + \Delta \in \mathbb{C}^{n \times n}$  a  $L = V^{-1}AV$ , kde  $L = \text{diag}\{\lambda_1, \dots, \lambda_n\}$ , potom  $\min_i |\lambda_i - \mu| \leq \kappa(V)\|\Delta\|$ , kde  $\kappa(V) = \|V^{-1}\|\|V\|$ .*



**Důkaz.** Je-li  $\mu$  vlastní číslo matice  $A + \Delta$  a  $x$  příslušný vlastní vektor, potom

$$\begin{aligned} (A + \Delta)x &= \mu x \\ (\mu I - A)x &= \Delta x \\ (\mu I - VDV^{-1})x &= \Delta x \\ V(\mu I - D)V^{-1}x &= \Delta x. \end{aligned}$$

Číslo  $\mu$  není vlastním číslem matice  $A$ , tedy

$$\begin{aligned} V^{-1}x &= (\mu I - D)^{-1}V^{-1}\Delta(VV^{-1})x \\ V^{-1}x &= (\mu I - D)^{-1}(V^{-1}\Delta V)V^{-1}x \\ \|V^{-1}x\| &= \|(\mu I - D)^{-1}(V^{-1}\Delta V)V^{-1}x\| \\ 1 &\leq \|(\mu I - D)^{-1}\| \|V^{-1}\| \|\Delta\| \|V\| \end{aligned}$$

a pro indukovanou normu

$$\begin{aligned} 1 &\leq \max_i \frac{1}{|\mu - \lambda_i|} \kappa(V) \|\Delta\| \\ \min_i |\mu - \lambda_i| &\leq \kappa(V) \|\Delta\|. \end{aligned}$$

□

Je-li matice symetrická, jsou vlastní vektory příslušející různým vlastním číslům ortogonální a vlastní vektory násobných vlastních čísel lze ortogonalizovat. Pro symetrickou matici tak existuje ortogonální matice  $V$ , pro kterou je  $\kappa(V) = 1$  a úloha na vlastní čísla symetrických matic (i singulárních čísel) vede na dobře podmíněnou úlohu.

Některé numerické metody pro výpočet vlastních čísel využívají podobnostní transformace a vztahu (2.1), případně se napřed převádí původní matice těmito transformacemi na tvar vhodnější pro výpočet. Uvažujme, že podobnostní transformaci aplikujeme na matici zatíženou chybami  $\Delta$ , tedy  $(A + \Delta)$  a  $X^{-1}(A + \Delta)X = X^{-1}AX + X^{-1}\Delta X = B$ , potom

$$\|X^{-1}\Delta X\| \leq \kappa(X) \|\Delta\|$$

a vlastní čísla matice  $B$  by se pro velké  $\kappa(X)$  velmi lišila od vlastních čísel matice  $A$ . Proto se používají pro podobnostní transformace unitární matice, tj.  $X \in \mathbb{C}^{n \times n}$ ,  $X^*X = XX^* = I$ , resp. ortogonální  $X \in \mathbb{R}^{n \times n}$ ,  $X^T X = XX^T = I$ .

Nyní uvedeme bez důkazu několik známých vět pro rozklady matic. Jejich důkazy lze nalézt např. v [30].

**Věta 2.13.** (Schurova) Pro libovolnou matici  $A \in \mathbb{C}^{n \times n}$  existuje unitární matice  $U \in \mathbb{C}^{n \times n}$  tak, že matice  $T = U^*AU$  bude horní trojúhelníková.

Na diagonále matice  $T$  potom nalezneme vlastní čísla matice  $A$ , která mohou být pro obecnou reálnou matici reálná i komplexní. Protože s komplexními čísly se pracuje obtížněji, můžeme uvažovat následující větu, která dává rozklad na reálné matice.

**Věta 2.14.** (*Schurova věta pro reálné matice*) Pro matici  $A \in \mathbb{R}^{n \times n}$  existují reálná ortogonální matice  $U$  a reálná kvazi-trojúhelníková matice  $T$  takové, že platí  $T = U^T A U$ .

Jednou z často používaných metod pro výpočet úplného problému vlastních čísel obecné matice (neřídké, nevelké dimenze) je  $QR$  algoritmus.

**Věta 2.15.** Necht' matice  $A \in \mathbb{R}^{n \times n}$  je regulární. Potom existují ortogonální matice  $Q \in \mathbb{R}^{n \times n}$  a horní trojúhelníková matice  $R \in \mathbb{R}^{n \times n}$  takové, že  $A = QR$ .

### Algoritmus 2.1. QR algoritmus

Vstup: matice  $A \in \mathbb{R}^{n \times n}$

$$A_0 = A$$

for  $k = 1, 2, \dots$

a) spočti QR rozklad matice  $A_{k-1}$ , tj.  $A_{k-1} = Q_k R_k$

b) sestav matici  $A_k = R_k Q_k$

end

Výstup: matice  $A_k$

Podle definice 2.6 je libovolná matice v  $k$ -té iteraci podobná původní matici  $A$ ,

$$\begin{aligned} A_k &= R_k Q_k = (Q_k^T Q_k) R_k Q_k = Q_k^T (Q_k R_k) Q_k = Q_k^T A_k Q_k = \\ &= Q_k^T R_{k-1} Q_{k-1} Q_k = Q_k^T (Q_{k-1}^T Q_{k-1}) R_{k-1} Q_{k-1} Q_k = \\ &= Q_k^T Q_{k-1}^T A_{k-2} Q_{k-1} Q_k = \dots = Q_k^T Q_{k-1}^T \dots Q_1^T A Q_1 \dots Q_{k-1} Q_k. \end{aligned}$$

Matice  $A_k$  se pro dostatečně velká  $k$  blíží horní (kvazi-)trojúhelníkové matici, vlastní čísla diagonálních bloků matice  $A_k$  jsou potom aproximacemi vlastních čísel matice  $A$ .

Pro urychlení  $QR$  algoritmu se matice  $A$  napřed převádí na horní Hessenbergerův tvar, např. pomocí Householderových reflexí či Givensových rotací [9]. Jiné možné urychlení se provádí pomocí tzv. shiftů (posunů). Navíc je vhodné pozorovat velikosti poddiagonálních prvků, a pokud by byly nulové, rozpadne se daná úloha na dva podproblémy. Podrobnější výklad je uveden v [9, 29].

**Algoritmus 2.2. QR algoritmus s posuny**

Vstup: matice  $A_0 \in \mathbb{R}^{n \times n}$ , (matice v horním Hessenbergerově tvaru získaná podobnostními transformacemi z matice  $A$ )

```

 $A_1 = A_0$ 
for  $k = 1, 2, \dots$ 
  a) urči posun  $\mu_k$  (např. prvek  $a_{nn}$  matice  $A_{k-1}$ )
  b) spočti QR rozklad matice  $A_k - \mu_k I$ , tj.  $A_k - \mu_k I = Q_k R_k$ 
  c)  $A_{k+1} = R_k Q_k + \mu_k I$ 
  d) if  $a_{n+1,n} = 0$  then pokračuj na podproblémech end if
end for

```

Výstup: matice  $A_k$

Výpočet singulárních čísel bude dle vztahu (2.3) dobře podmíněnou úlohou. Jednou z možností by bylo aplikovat QR algoritmus na matici  $A^*A$  a počítat její vlastní čísla. Tento přístup se ale nepoužívá, neboť formovat součin  $A^*A$  není vhodné. Pokud by byla matice  $A$  špatně podmíněná, bylo by  $\kappa(A^*A) = \kappa(A)^2$  velké a malá singulární čísla bychom mohli obdržet značně nepřesná.

Singulární čísla lze získat s relativní přesností odpovídající strojové přesnosti. K jejich určení se matice  $A$  nejprve převádí na bidiagonální tvar (např. Householderovými reflexemi) a následně se implicitním QR algoritmem nulují mimodiagonální prvky.

**2.4.1 Numerický výpočet vlastních čísel Hamiltonovy matice**

Při výpočtu vlastních čísel Hamiltonovy matice je důležité využít její speciální struktury a vlastností. Kdybychom použili například QR algoritmus, vlivem konečné aritmetiky bychom tyto vlastnosti ztratili.

V níže uvedených kapitolách uvádíme, že při výpočtu jednoho kritéria robustnosti je potřeba rozhodnout, zda má odpovídající Hamiltonova matice ryze imaginární vlastní čísla. Proto je dobré využít metody, které budou zachovávat například symetrii vlastních čísel. Zaokrouhlovací chyby potom způsobí, že ryze imaginární čísla se budou posouvat po imaginární ose. Nastíňme si nyní metodu pro výpočet vlastních čísel Hamiltonovy matice, kterou navrhli Benner a Kressner v [2].

Pro zachování struktury by měly podobnostní transformace být provedeny maticemi symplektickými (tj.  $U^T J U = J$ , kde  $J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$ ) a ortogonálními (tj.  $U^T U = U U^T = I$ ).

**Věta 2.16.** ([1]) Necht'  $H \in \mathbb{R}^{2n \times 2n}$  je Hamiltonova matice, potom existují ortogonální symplektické matice  $U, V \in \mathbb{R}^{2n \times 2n}$  takové, že

$$U^T H V = \begin{bmatrix} R_{1,1} & R_{1,2} \\ 0 & R_{2,2} \end{bmatrix}, \quad (2.11)$$

kde  $R_{i,j} \in \mathbb{R}^{n \times n}$  (pro  $i, j \in \{1, 2\}$ ) a matice  $R_{1,1}$  je horní trojúhelníková a  $R_{2,2}$  dolní Hessenbergerova.

**Tvrzení 2.1.** ([1]) Necht'  $H \in \mathbb{R}^{2n \times 2n}$  je Hamiltonova matice, potom existují ortogonální symplektické matice  $U, V \in \mathbb{R}^{2n \times 2n}$  takové, že

$$U^T H^2 U = \begin{bmatrix} -R_{1,1} R_{2,2}^T & R_{1,1} R_{1,2}^T - R_{1,2} R_{1,1}^T \\ 0 & -R_{2,2} R_{1,1}^T \end{bmatrix}, \quad (2.12)$$

kde  $R_{i,j} \in \mathbb{R}^{n \times n}$  (pro  $i, j \in \{1, 2\}$ ) a matice  $R_{1,1}$  je horní trojúhelníková a  $R_{2,2}$  dolní Hessenbergerova.

**Důkaz.** Podle věty 2.16 existují ortogonální symplektické matice  $U, V$  takové, že platí (2.11). Dále platí  $U^T U = I, V^T V = I, U^T J U = J$  a  $V^T J V = J$ , potom  $J H^T J = J$  a následně

$$\begin{aligned} U^T H^2 U &= U^T H H U = U^T H V V^T H U = U^T H V V^T J H^T J U = \\ &= U^T H V V^T V J V^T H^T U J U^T U = (U^T H V) J (V^T H^T U) J = \\ &= (U^T H V) J (U^T H V)^T J = \\ &= \begin{bmatrix} R_{1,1} & R_{1,2} \\ 0 & R_{2,2} \end{bmatrix} \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} \begin{bmatrix} R_{1,1}^T & 0 \\ R_{1,2}^T & R_{2,2}^T \end{bmatrix} \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} = \\ &= \begin{bmatrix} -R_{1,2} & R_{1,1} \\ -R_{2,2} & 0 \end{bmatrix} \begin{bmatrix} 0 & R_{1,1}^T \\ -R_{2,2}^T & R_{1,2}^T \end{bmatrix} = \begin{bmatrix} -R_{1,1} R_{2,2}^T & R_{1,1} R_{1,2}^T - R_{1,2} R_{1,1}^T \\ 0 & -R_{2,2} R_{1,1}^T \end{bmatrix}. \end{aligned}$$

□

Matice  $-R_{1,1} R_{2,2}^T$  je horní Hessenbergerova matice, její vlastní čísla jsou mocniny vlastních čísel Hamiltonovy matice  $H$ . K jejich výpočtu můžeme použít periodický QR algoritmus.

### Algoritmus 2.3. Výpočet vlastních čísel Hamiltonovy matice

Vstup:  $H$  Hamiltonova matice

- určí ortogonální symplektické matice  $U, V$ , aby  $U^T H V = \begin{bmatrix} R_{1,1} & R_{1,2} \\ 0 & R_{2,2} \end{bmatrix}$ ,  
(věta 2.16)
- periodickým QR algoritmem aplikovaným na  $-R_{1,1} R_{2,2}^T$  spočti vlastní čísla  $l_i, i = 1, \dots, n$
- vypočti vlastní čísla Hamiltonovy matice  $\lambda_i = \sqrt{l_i}, \lambda_{n+i} = -\sqrt{l_i}$ , pro  $i = 1, \dots, n$ .

Výstup:  $\lambda_i$ , pro  $i = 1, \dots, 2n$ .

## 2.5 Metody pro hledání optima

Při hledání zpětných vazeb (resp. odpovídajících parametrů) takových, že uzavřený systém bude mít nejlepší požadované vlastnosti, budeme potřebovat numerické optimalizační metody pro jednorozměrné i vícerozměrné funkce. Přehledný výklad optimalizačních metod je uveden v [10, 12].

**Definice 2.10.** Bod  $x^* \in \mathbb{R}^n$  se nazývá bodem lokálního minima funkce  $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ , pokud existuje  $\varepsilon > 0$  tak, že

$$f(x) \geq f(x^*), \quad \text{pro } \forall x \in \mathcal{U}(x^*, \varepsilon),$$

kde  $\mathcal{U}(x^*, \varepsilon) = \{x \in \mathbb{R}^n : 0 \leq \|x - x^*\| < \varepsilon\}$ .

**Definice 2.11.** Bod  $x^* \in \mathbb{R}^n$  je bodem globálního minima funkce  $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ , jestliže

$$f(x^*) \leq f(x), \quad \text{pro } \forall x \in \mathbb{R}^n.$$

**Definice 2.12.** Funkce  $f(x) : \mathbb{R} \rightarrow \mathbb{R}$  se nazývá kvazikonvexní funkcí na intervalu  $\langle a, b \rangle$ , když pro libovolné body  $x, y \in \langle a, b \rangle$  a pro každé  $\alpha \in (0, 1)$  platí, že:

$$f(\alpha x + (1 - \alpha)y) \leq \max\{f(x), f(y)\}.$$

**Definice 2.13.** Funkce  $f(x) : \mathbb{R} \rightarrow \mathbb{R}$  je unimodální funkcí na intervalu  $\langle a, b \rangle$ , když existuje  $c \in (a, b)$  takové, že  $f$  na intervalu  $\langle a, c \rangle$  klesá a na intervalu  $\langle c, b \rangle$  roste.

**Věta 2.17.** Necht' je funkce  $f(x) : \mathbb{R} \rightarrow \mathbb{R}$  unimodální funkcí v intervalu  $\langle a, b \rangle$  a nabývá svého minima v  $x^* \in \langle a, b \rangle$ . Potom pro  $x, y \in \langle a, b \rangle$ ,  $x < y$  platí, že

- jestliže  $f(x) < f(y)$ , potom  $x^* < y$ ,
- jestliže  $f(x) > f(y)$ , potom  $x^* > x$ ,
- jestliže  $f(x) = f(y)$ , potom  $x < x^* < y$ .

### 2.5.1 Metoda zlatého řezu

V případě jednorozměrné optimalizace unimodální funkce můžeme hledat minimum pomocí metody zlatého řezu. Minimum unimodální funkce je přímo jejím globálním minimem.

Uvažujme interval  $\langle a, b \rangle$ . Metoda vychází ze tří bodů, řekněme  $a, b, d$ , rozmístěných tak, aby  $a < d < b$  a poměr velikosti celého intervalu ku větší části byl stejný jako poměr velikosti větší části ku menší části aktuálního intervalu. K těmto bodům se konstruuje čtvrtý bod  $c$  a podle funkčních hodnot a věty 2.17 se vybírá trojice bodů pro další iteraci.

Pro interval  $\langle a, b \rangle$  vybíráme  $d$ , aby

$$\begin{aligned} \frac{b-a}{d-a} &= \frac{d-a}{b-d} = \gamma \\ \gamma &= \frac{d-a}{b-d} = \frac{b-a}{d-a} = \frac{(d-a) + (b-d)}{d-a} = 1 + \frac{b-d}{d-a} = 1 + \frac{1}{\gamma} \\ \gamma &= 1 + \frac{1}{\gamma} \\ \gamma^2 - \gamma - 1 &= 0 \\ \gamma_{1,2} &= \frac{1 \pm \sqrt{5}}{2}. \end{aligned}$$

Hodnota  $\gamma = \frac{1+\sqrt{5}}{2}$  se nazývá zlatý řez, odtud také název metody.

Další bod  $c$  chceme umístit tak, aby poměr délek zůstal zachován, ať vybereme jako další trojici body  $a, c, d$  nebo  $c, d, b$ . Proto

$$\begin{aligned} \frac{c-a}{d-c} &= \frac{d-a}{c-a} = \frac{b-c}{b-d} = \frac{b-d}{d-c} = \gamma \quad \text{a} \\ c-a &= \gamma(d-c) = b-d \\ c &= a + b - d. \end{aligned}$$

Délky intervalů obou uvažovaných trojic jsou shodné, ( $c - a = b - d$ ).

#### Algoritmus 2.4. Metoda zlatého řezu

Vstup:  $\text{tol}$  (tolerance)  $> 0$ ,  $\kappa = \frac{2}{1+\sqrt{5}}$ ,  $a_0 < b_0$ ,  $d_0 = (b_0 - a_0)\kappa + a_0$ ,  $c_0 = a_0 + b_0 - d_0$ ,  $k = 0$

```

while  $b_k - a_k > \text{tol}$ 
  if  $f(d_k) > f(c_k)$  then  $a_{k+1} = a_k$ ,  $b_{k+1} = d_k$ ,  $d_{k+1} = c_k$ ,  $c_{k+1} = a_{k+1} + b_{k+1} - d_{k+1}$ 
  else if  $f(d_k) < f(c_k)$  then  $a_{k+1} = c_k$ ,  $b_{k+1} = b_k$ ,  $c_{k+1} = d_k$ ,  $d_{k+1} = a_{k+1} + b_{k+1} - c_{k+1}$ 
  else  $a_{k+1} = c_k$ ,  $b_{k+1} = d_k$ ,  $d_{k+1} = (b_{k+1} - a_{k+1})\kappa + a_{k+1}$ ,  $c_{k+1} = a_{k+1} + b_{k+1} - d_{k+1}$ 
  end if
   $k = k + 1$ 
end while

```

Výstup:  $f(x_k)$ , kde  $x_k = \frac{a_{k+1} + b_{k+1}}{2}$

Dále nás bude zajímat nalezení optima nelineárních funkcí více proměnných, které budou reprezentovány vybranými kritérii robustnosti uvedenými v následujících kapitolách. V [7] navrhuji pro hledání extrémů těchto funkcí například metodu BFGS.

## 2.5.2 Line search metody

Mějme funkci  $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  a hledejme  $\min_{x \in \mathbb{R}^n} f(x)$ . Předpokládáme, že v  $k$ -té iteraci máme pro  $x_k$  daný směr poklesu  $h_k$ . Chceme-li se z bodu  $x_k$  dostat do nového bodu  $x_{k+1} = x_k + th_k$ , potřebujeme určit velikost kroku  $t$  ve směru  $h_k$ .

Označme  $g(t) = f(x_k + th_k)$ , potom

$$g'(t) = \nabla f(x_k + th_k)h_k. \quad (2.13)$$

Délku kroku chceme volit tak, abychom zajistili dostatečný, ale vhodný pokles. Dostaneme dvě podmínky, které budeme chtít splnit současně. První podmínka požaduje, aby byl pokles funkčních hodnot následných iterací úměrný velikosti kroku i  $g'(0)$ . Zabraňuje použití příliš velkých kroků, tedy těch, pro které je  $g(t) - g(0)$  mnohem větší než  $c_1 g'(0)$ , kde  $c_1 \in (0, 1)$ . Tato podmínka je nazývána Armijovo pravidlo,

$$g(t) \leq g(0) + c_1 t g'(0). \quad (2.14)$$

Protože podmínka (2.14) neurčuje, kdy je krok příliš malý, přidává se druhá podmínka. Pokud je hodnota  $g'(t)$  velká záporná, může hodnota funkce  $f$  v tomto směru značně poklesnout a nemusíme volit malé kroky. Hodnota  $g'(0)$  je záporná, vynásobením konstantou  $c_2 \in (c_1, 1)$  ji zvětšíme. Pokud požadujeme, aby  $g'(t) > c_2 g'(0)$ , vyloučíme malé kroky, protože v okolí  $t = 0$  bude mít  $g'(t)$  podobnou hodnotu.

Celkem tak získáme následujícího Wolfeho podmínky,

$$g(t) \leq g(0) + c_1 t g'(0) \quad (2.15)$$

$$g'(t) > c_2 g'(0), \quad (2.16)$$

kde  $0 < c_1 < c_2 < 1$ . Velikost kroku  $t$  potom volíme takovou, aby vyhovovala Wolfeho podmínkám.

## 2.5.3 Kvazi-Newtonovské metody, metoda BFGS

Kvazi-Newtonovské metody patří mezi gradientní metody numerické optimalizace. Uvažujme úlohu hledání minima funkce  $f(x)$ . Vyjdeme-li z nějakého bodu  $x_k$ , chceme najít bod  $x_{k+1}$ , pro který bude platit, že  $f(x_{k+1}) \leq f(x_k)$ . Protože máme k dispozici obecně více směrů, snažíme se vybrat nejvhodnější směr, ze kterého bychom měli pokračovat. Tím by mohl být směr odpovídající zápornému gradientu, protože funkce  $f(x)$  zde dosahuje největšího spádu.

Pro hledání minima se používá iterační předpis gradientních metod

$$x_{k+1} = x_k - \alpha_k g_k,$$

kde  $\alpha_k$  určuje velikost kroku a gradient  $g_k$  směr poklesu.

Kvazi-Newtonovské metody spadají jednak mezi metodu největšího spádu (základní gradientní metodu) a také mezi Newtonovu metodu pro hledání řešení nelineárních rovnic (resp. soustav).

Z metody největšího spádu, u níž je

$$x_{k+1} = x_k + \frac{g_k^T g_k}{g_k^T H g_k} g_k,$$

kde  $H$  je Hessova matice (matice druhých derivací), chceme využít vlastnosti konvergence.

Podle Newtonovy metody, kde uvažujeme

$$x_{k+1} = x_k + H_k^{-1} g_k, \quad (2.17)$$

resp. její modifikace

$$x_{k+1} = x_k + \alpha_k H_k^{-1} g_k,$$

pracují s Hessovou maticí  $H_k$ .

Kvazi-Newtonovské metody jsou založeny na aproximaci inverze Hessovy matice ( $B_k$ ). Potom vztah (2.17) přejde na

$$x_{k+1} = x_k - B_k g_k.$$

Aproximace  $B_k$  je konstruována tak, aby dobře odpovídala matici  $H_k^{-1}$ , její výpočet nebyl náročný a využíval znalostí (vypočtených matic a vektorů) z předchozích iterací. Tím se značně odlišují od klasické Newtonovy metody.

Uvažujme nejprve aproximaci přímo Hessovy matice  $H_k$

$$g_{k+1} - g_k = H_{k+1}(x_{k+1} - x_k).$$

Označme  $y_k = g_{k+1} - g_k$  a  $s_k = x_{k+1} - x_k$ , potom  $H_{k+1}s_k = y_k$ . Chceme, aby matice  $H_{k+1}$  byla symetrická, pozitivně definitní, splňovala rovnost  $H_{k+1}s_k = y_k$  a navíc byla v normě nejbliž matici  $H_k$ .

**Věta 2.18.** ([10]) *Necht'  $H \in \mathbb{R}^{n \times n}$ ,  $y = Hs$ ,  $y^T s > 0$  a  $H_k \in \mathbb{R}^{n \times n}$  je daná matice, potom platí*

$$\min_H \|H - H_k\| = \|\bar{H} - H_k\|,$$

kde

$$\bar{H} = H_k + \frac{(y - H_k s)^T s^T}{s^T s}.$$



Nyní můžeme odvodit formuli pro výpočet aproximace Hessovy matice  $H_{k+1}$ . Uvažujme Choleského rozklad matic  $H_k = LL^T$  z předchozí iterace a  $H_{k+1} = JJ^T$ . Požadujeme, aby byl splněn vztah  $JJ^T s = y$  a blízkost  $H_{k+1}$  k  $H_k$ .

Označme  $v = J^T s$ , potom

$$J = L + \frac{(y - Lv)v^T}{v^T v}. \quad (2.18)$$

Úpravami získáme

$$\begin{aligned} v &= L^T s + \frac{v(y - Lv)^T s}{v^T v} \\ v &= L^T s + \frac{v(y^T s - v^T L^T s)}{v^T v} \end{aligned} \quad (2.19)$$

a pro  $w = \frac{y^T s - v^T L^T s}{v^T v}$  bude  $L^T s = (1 - w)v$ . Chceme získat vztah pro  $v$ , označme  $p = \frac{1}{1-w}$ , potom  $v = pL^T s$ . Dosazením do (2.19) dostaneme

$$\begin{aligned} pL^T s &= L^T s + \frac{pL^T s(y - LpL^T s)^T s}{(pL^T s)^T (pL^T s)} \\ pL^T s &= p \frac{y^T s}{p^2 s^T LL^T s} L^T s \\ p^2 &= \frac{y^T s}{s^T LL^T s} = \frac{y^T s}{s^T H_k s}. \end{aligned}$$

Pokud bude  $y^T s > 0$ , můžeme vyjádřit

$$v = \sqrt{\frac{y^T s}{s^T H_k s}} L^T s. \quad (2.20)$$

Tuto úpravu můžeme provést díky (2.13) a (2.16). Dosazením (2.20) a (2.18) do  $H_{k+1} = JJ^T$  získáme

$$H_{k+1} = LL^T + \frac{yy^T}{v^T v} - \frac{Lv v^T L^T}{v^T v}. \quad (2.21)$$

Výrazy s vektorem  $v$  můžeme dále vyjádřit

$$v^T v = y^T s, \quad (2.22)$$

$$v v^T = \frac{y^T s}{s^T H_k s} L^T s s^T L. \quad (2.23)$$

Nakonec ze vztahů (2.21), (2.20), (2.22) obdržíme formuli pro aproximaci Hessovy matice  $H_{k+1}$

$$H_{k+1} = H_k + \frac{yy^T}{y^T s} - \frac{H_k s s^T H_k}{s^T H_k s}. \quad (2.24)$$

V metodě BFGS (Broydenově-Fletcherově-Goldfarbově-Shannoově kvazi-Newtonovské metodě) nás zajímá tvar aproximace inverze Hessovy matice. Tu získáme ze vztahu (2.24) pomocí Shermanovy-Morrisonovy-Woodburyho formule.

**Věta 2.19.** (Sherman–Morrison–Woodbury) Je-li  $M \in \mathbb{R}^{n \times n}$  nesingulární matice a  $u, v \in \mathbb{R}^n$  a  $N = M + uv^T$ , potom

$$N^{-1} = M^{-1} + \frac{M^{-1}uv^T M^{-1}}{1 + v^T M^{-1}u}.$$

Aplikujme tedy nyní větu 2.19 na matici  $H_{k+1}$

$$\begin{aligned} H_{k+1}^{-1} &= \left[ H_k + \frac{yy^T}{y^T s} \right]^{-1} + \frac{\left[ H_k + \frac{yy^T}{y^T s} \right]^{-1} \frac{H s s^T H}{s^T H s} \left[ H_k + \frac{yy^T}{y^T s} \right]^{-1}}{1 - \frac{s^T H \left[ H_k + \frac{yy^T}{y^T s} \right]^{-1} H s}{s^T H s}} = \\ &= H^{-1} - \frac{H^{-1} y y^T H^{-1}}{y^T s + y^T H^{-1} y} + \frac{\left[ H^{-1} - \frac{H^{-1} y y^T H^{-1}}{y^T s + y^T H^{-1} y} \right] \frac{H s s^T H}{s^T H s} \left[ H^{-1} - \frac{H^{-1} y y^T H^{-1}}{y^T s + y^T H^{-1} y} \right]}{1 - \frac{s^T H \left[ H^{-1} - \frac{H^{-1} y y^T H^{-1}}{y^T s + y^T H^{-1} y} \right] H s}{s^T H s}}, \end{aligned} \quad (2.25)$$

Po úpravě získáme formuli pro aproximaci inverze Hessovy matice  $B_{k+1}$

$$B_{k+1} = \left( I - \frac{sy^T}{y^T s} \right) B_k \left( I - \frac{ys^T}{y^T s} \right) + \frac{ss^T}{y^T s}. \quad (2.26)$$

Uvedené požadavky na konstrukci aproximace inverze Hessovy matice  $B_{k+1}$  jsou tedy splněny. Získáme ji s využitím předchozích iterací, a sice modifikací matice  $B_k$  (aproximace z předchozí iterace) maticemi o hodnoti jedna.

### Algoritmus 2.5. Hledání optima – metoda BFGS

Vstup:  $\text{tol} > 0$  (tolerance),  $B_0 = I$ ,  $x_0 \in \mathbb{R}^n$ ,  $g_0 = g(x_0)$ ,  $k = 0$

while  $\|g_k\| > \text{tol}$

- a) vypočti  $t_k = -B_k g_k$
- b) urči  $\alpha_k$  (podle Wolfeho podmínek (2.15), (2.16))
- c)  $x_{k+1} = x_k + \alpha_k t_k$
- d)  $s_k = x_{k+1} - x_k$
- e) spočti  $g_{k+1}$
- f)  $y_k = g_{k+1} - g_k$
- g) vypočti  $B_{k+1}$  (podle (2.26))
- h)  $k = k + 1$

end while

Výstup:  $f(x_k)$ ,  $x_k$

## 2.6 Metody pro řešení nelineárních rovnic

Uvažujme soustavu nelineárních rovnic  $f(x) = 0$ ,  $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ . Při řešení této soustavy můžeme využít přístupy pro řešení problému minimalizace metodou nelineárních nejmenších čtverců.

Chceme minimalizovat  $\|f(x)\|$ , pro  $x \in \mathbb{R}^n$ , resp. najít

$$x_{\text{opt}} = \arg \min_x \{F(x)\},$$

kde  $F(x) = \frac{1}{2}\|f(x)\|^2 = \frac{1}{2}f^T(x)f(x)$ .

Mezi metody, které se používají při řešení tohoto problému, patří například Gaussova-Newtonova metoda a Levenbergova-Marquardtova metoda. Vlastnostem uvedených metod se věnují například v [18, 33].

Uvažujme Taylorův rozvoj funkce  $f$  v bodě  $x + h$

$$f(x + h) = f(x) + J(x)h + O(\|h\|^2),$$

kde  $J$  představuje Jacobián, tj. matici prvních parciálních derivací.

Uvažované metody využívají lineární aproximaci funkce  $f(x)$

$$l(h) \triangleq f(x) + J(x)h,$$

tedy  $f(x + h) \approx l(h)$ . Potom

$$\begin{aligned} F(x + h) &= \frac{1}{2}f^T(x + h)f(x + h) \approx \frac{1}{2}l^T(h)l(h) = \\ &= \frac{1}{2}(f(x) + J(x)h)^T(f(x) + J(x)h) = \\ &= \frac{1}{2}f^T(x)f(x) + h^T J^T(x)f(x) + \frac{1}{2}h^T J^T(x)J(x)h = \\ &= F(x) + h^T J^T(x)f(x) + \frac{1}{2}h^T J^T(x)J(x)h \triangleq L(h) \end{aligned}$$

Gaussova-Newtonova metoda využívá iterační formuli

$$x_{k+1} = x_k + \alpha_k h_k^{\text{GN}},$$

kde  $\alpha_k$  je velikost kroku, kterou získáme metodou *line search*, nebo nastavíme pevnou, a  $h_k^{\text{GN}}$  představuje krok v  $k$ -té iteraci. Krok  $h_k^{\text{GN}}$  splňuje

$$h_k^{\text{GN}} = \arg \min\{L(h)\}.$$

Gradient funkce  $L(h)$  je  $\nabla L(h) = J^T(x)f + J^T(x)J(x)h$  a matice druhých derivací, tj. Hessián,  $\nabla^2 L(h) = J^T J$ . Matice  $\nabla^2 L(h)$  je symetrická, a pokud matice  $J$  bude mít plnou hodnot, bude pozitivně definitní a  $L(h)$  bude mít jediné minimum.

Potom

$$\begin{aligned}\nabla L(h) &= J^T f + J^T J h = 0 \\ J^T J h &= -J^T f\end{aligned}\tag{2.27}$$

a  $h^{\text{GN}}$  je řešením rovnice (2.27).

Z Gaussovy-Newtonovy metody vychází Levenbergova-Marquardtova metoda. Ta modifikuje aproximaci Hessiánu o diagonální matici s parametrem  $\mu \geq 0$ . Levenbergova-Marquardtova metoda (formulovaná s volbou kroku obdobně jako *trust region*) řeší úlohu

$$\min \frac{1}{2} \|f + Jh\|^2, \quad \text{s omezením, že } \|h\|^2 \leq \mu.\tag{2.28}$$

Krok Levenbergovy-Marquardtovy metody  $h^{\text{LM}}$  je dán řešením

$$h_k^{\text{LM}} = \arg \min \{L(h) + \frac{1}{2} \mu h^T h\},$$

kde člen  $\frac{1}{2} \mu h^T h$  představuje penalizaci omezující podmínky ve (2.28), která penalizuje velké kroky. Potom

$$\begin{aligned}L(h) &= F(x) + h^T J^T(x)f(x) + \frac{1}{2} h^T J^T(x)J(x)h + \frac{1}{2} \mu h^T h \\ \nabla L(h) &= J^T f + J^T J h + \mu h = 0\end{aligned}$$

a krok  $h^{\text{LM}}$  je řešením rovnice

$$(J^T J + \mu I)h = -J^T f\tag{2.29}$$

Parametr  $\mu$  zásadně ovlivňuje i další postup metody. Protože má platit (2.28), máme již dané omezení na velikost poklesu, proto zde už neuvažujeme *line search* metodu. V průběhu metody měníme hodnotu  $\mu$  podle velikosti hodnoty

$$\rho = \frac{F(x_k) - F(x_k + h^{\text{LM}})}{L(0) - L(h^{\text{LM}})},\tag{2.30}$$

přičemž  $F(x_k) - F(x_k + h^{\text{LM}})$  představuje aktuální změnu v dané iteraci a  $L(0) - L(h^{\text{LM}})$  předpokládanou změnu. Pro hodnoty  $\rho \approx 1$  parametr  $\mu$  zmenšíme, pro  $\rho$  blízko nule zvětšíme a jinak ponecháme stejný. Stručný nástin Levenbergovy-Marquardtovy metody uvádí následující algoritmus.

**Algoritmus 2.6. Levenbergova-Marquardtova metoda [18]**

Vstup:  $\mu_0, x_0, H_0 = J^T(x_0)J(x_0), g(x_0) = J^T(x_0)f, k = 0$

while  $\|g_k\| > \varepsilon$

a) vyřeš  $(H_k + \mu I)h_k^{\text{LM}} = -g_k$

b)  $x_{k+1} = x_k + h_k^{\text{LM}}$

c) vypočti  $\rho = \frac{F(x_k) - F(x_{k+1})}{L(0) - L(h_k^{\text{LM}})}$

d) if  $\rho > 0$

$$H_{k+1} = J^T(x_{k+1})J(x_{k+1}), g_{k+1} = J^T(x_{k+1})f(x_{k+1})$$

$$\mu = \mu \max\left\{\frac{1}{3}, 1 - (2\rho - 1)^3\right\}, \nu = 2$$

else  $\mu = \mu\nu, \nu = 2\nu$

end if

e)  $k = k + 1$

end while

Výstup:  $x_{k+1}, f(x_{k+1})$

## 3 Přiřazení Jordanovy formy zpětnou vazbou

Dynamické chování autonomního systému je dáno vlastními čísly matice dynamiky. Chceme-li změnit jeho vlastnosti, např. z nestabilního systému udělat stabilní, nebo z pomalého systému rychlý, zavedeme zpětnou vazbu. Podle toho, jaké hodnoty můžeme měřit, rozlišujeme stavovou a výstupní zpětnou vazbu. V této kapitole si nejprve připomeneme výsledky prací [23, 16, 25]. Dále uvedeme nový algoritmus na výpočet parametrické matice příslušné k Jordanově formě, kterou využíváme při výpočtech stavové i výstupní zpětné vazby.

Mějme model spojitého systému

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t),\end{aligned}\tag{3.1}$$

kde  $A \in \mathbb{R}^{n \times n}$  je matice dynamiky systému,  $B \in \mathbb{R}^{n \times m}$  vstupní matice (matice buzení),  $C \in \mathbb{R}^{p \times n}$  výstupní matice, vektor  $x(t) \in \mathbb{R}^n$  je stav systému, vektor  $u(t) \in \mathbb{R}^m$  vstup systému a vektor  $y(t) \in \mathbb{R}^p$  výstup systému.

### 3.1 Stavová zpětná vazba

Zavedme stavovou zpětnou vazbu

$$u(t) = Fx(t),\tag{3.2}$$

kde  $F \in \mathbb{R}^{m \times n}$ .

Definujme množinu všech stavových zpětných vazeb

$$\mathcal{F}_s(A, B, L) \triangleq \left\{ F \in \mathbb{R}^{m \times n} : (A + BF) \sim \begin{bmatrix} L & * \\ 0 & * \end{bmatrix} \right\},\tag{3.3}$$

kde  $L \in \mathbb{R}^{s \times s}$ ,  $s \leq n$  je daná reálná Jordanova forma a symboly  $*$  označují libovolné reálné matice. Jetliže  $s = n$  hovoříme o úplném přiřazení, pokud  $s < n$ , jedná se o neúplné přiřazení.

O nutné a postačující podmínce pro existenci stavové zpětné vazby pro případ, že  $s = n$ , hovoří Rosenbrockova věta. Rozšíření pro obecný případ  $s < n$  lze nalézt v [23].

**Věta 3.1.** (Rosenbrock) Necht'  $\mu(A, B) = \{\mu_i\}_{i=1}^m$  jsou indexy říditelnosti dvojice  $(A, B)$  a  $\nu(L) = \{\nu_i\}_{i=1}^k$  jsou stupně všech nekonstantních polynomů matice  $L$ . Potom množina  $\mathcal{F}_n(A, B, L)$  je neprázdná právě tehdy, jestliže  $k \leq m$  a pro všechna  $j = 1, \dots, k$  platí  $\sum_{i=1}^j (\nu_i - \mu_i) \geq 0$ .

Uvažujme pro lepší názornost zatím pouze úplné přiřazení. Ve vztahu (3.3) požadujeme potom přímo, aby  $(A + BF) \sim L$ . Musí tak existovat regulární matice  $X \in \mathbb{R}^{n \times n}$  taková, že

$$A + BF = XLX^{-1}.$$

Po úpravě dostaneme Sylvestrovu rovnici

$$AX - XL + BFX = 0. \quad (3.4)$$

Označme  $FX = H$ , kde  $H \in \mathbb{R}^{m \times n}$  je parametrická matice. Rovnice (3.4) přejde do tvaru

$$AX - XL + BH = 0. \quad (3.5)$$

Řešení  $X(H)$  Sylvestrové rovnice (3.5) závisí na prvcích matice  $H$ . Bude-li platit, že dvojice  $(A, B)$  je říditelná,  $\mathcal{F}_n(A, B, L) \neq \emptyset$  a  $\Lambda(A) \cap \Lambda(L) = \emptyset$ , potom bude matice  $X(H)$  pro skoro každou matici  $H$  regulární. Stavovou zpětnou vazbu, která systému  $(A, B)$  přiřadí požadovanou Jordanovu formu  $L$ , můžeme potom vyjádřit ve tvaru  $F(H) = HX^{-1}(H)$ .

Následující věta poskytuje redundantní explicitní parametrizaci množiny  $\mathcal{F}_s(A, B, L)$ , obecně pro  $s \leq n$ .

**Věta 3.2.** Necht'  $\Lambda(A) \cap \Lambda(L) = \emptyset$  a  $\mathcal{F}_s(A, B, L) \neq \emptyset$ . Potom platí následující tvrzení:

- (i) Řešení  $X(H)$  rovnice (3.5) má plnou hodnotu pro skoro všechny matice  $H \in \mathbb{R}^{m \times s}$ .
- (ii) Libovolnou matici  $F \in \mathcal{F}_s(A, B, L)$  lze vyjádřit ve tvaru

$$F = \varphi(H, F_0) \triangleq H[X^T(H)X(H)]^{-1}X^T(H) + F_0 \quad (3.6)$$

kde  $H \in \mathcal{H}_s \triangleq \{H \in \mathbb{R}^{m \times s} : \det[X(H)] \neq 0\}$ ,  $X(H)$  je řešením rovnice (3.5) a pro  $F_0 \in \mathbb{R}^{m \times n}$  platí, že  $F_0X(H) = 0$ . Je-li  $s = n$ , potom  $F_0 = 0$ . Naopak, za uvedených předpokladů, matice  $\varphi(H, F_0)$  dána vztahem (3.6) patří do  $\mathcal{F}_s(A, B, L)$ .

Parametrizace pomocí matice  $H \in \mathbb{R}^{m \times s}$  je redundantní, tedy dává přebytečný počet parametrů ( $ms$ ), který by se zejména při následné numerické optimalizaci stal velmi nežádoucím. Proto je užitečné přejít k minimální parametrizaci, čímž se počet parametrů výrazně sníží.

Budeme-li opět uvažovat úplné přiřazení a vrátíme se ke vztahu (3.4), vidíme, že regulární matice  $X$  není určena jednoznačně. Uvažujme  $X_1, X_2, (X_1 \neq X_2)$ , potom dostaneme  $A + BF = X_1 L X_1^{-1}$  a  $A + BF = X_2 L X_2^{-1}$  a

$$X_1 L X_1^{-1} = X_2 L X_2^{-1}.$$

Jednoduchou úpravou získáme

$$X_2^{-1} X_1 L = L X_2^{-1} X_1.$$

Označme  $X_2^{-1} X_1 = Z$ , potom  $ZL = LZ$  a matice  $Z$  představuje regulární matici zaměnitelnou s maticí  $L$ . Množinu všech nesingulárních matic zaměnitelných s maticí  $L$  budeme značit  $\mathcal{Z}(L)$ .

Následující tvrzení nám pomohou při zavedení parametrické matice příslušné k matici  $L$ , která bude mít minimální počet parametrů.

**Lemma 3.1.** *Necht'  $\Lambda(A) \cap \Lambda(L) = \emptyset$ . Potom platí následující tvrzení:*

- (i) *Jestliže  $H_1 = H_2 Z$ ,  $Z \in \mathcal{Z}(L)$  a  $F_{01} = F_{02}$ , potom  $\varphi(H_1, F_{01}) = \varphi(H_2, F_{02})$ .*
- (ii) *Pokud  $\varphi(H_1, F_{01}) = \varphi(H_2, F_{02}) \triangleq F$  a  $A + BF$  je podobná matici  $M = \text{diag}\{L, S\}$ ,  $L \in \mathbb{R}^{s \times s}$ ,  $S \in \mathbb{R}^{(n-s) \times (n-s)}$ ,  $\Lambda(L) \cap \Lambda(S) = \emptyset$ , potom  $H_1 = H_2 Z$ ,  $Z \in \mathcal{Z}(L)$  a  $F_{01} = F_{02}$ .*

**Lemma 3.2.** *Necht'  $Q(\alpha) \in \mathbb{R}^{m \times s}$  je parametrická matice odpovídající matici  $L \in \mathbb{R}^{s \times s}$ , potom skoro každá matice  $H \in \mathbb{R}^{m \times s}$  může být jednoznačně vyjádřena ve tvaru*

$$H = Q(\alpha)Z, \tag{3.7}$$

kde  $Z \in \mathcal{Z}(L)$ .

Navažme opět na úplné přiřazení a Sylvestrovu rovnici (3.5). Podle předchozích lemmat můžeme psát

$$\begin{aligned} AX - XL + BH &= 0, \\ AX - XL + B\bar{H}Z &= 0, \\ AX - XL + BQ(\alpha) &= 0. \end{aligned} \tag{3.8}$$

Řešení rovnice (3.8) potom již závisí na parametru  $\alpha$ . Stavovou zpětnou vazbu získáme jako  $F(\alpha) = Q(\alpha)X^{-1}(\alpha)$ .

Následující věta uvádí explicitní parametrizaci stavových zpětných vazeb pomocí minimálního počtu návrhových parametrů.



**Věta 3.3.** *Necht'  $\mathcal{F}_s(A, B, L) \neq \emptyset$ ,  $\Lambda(A) \cap \Lambda(L) = \emptyset$  a  $\nu_1 \geq \nu_2 \geq \dots \geq \nu_k$  jsou stupně nekonstantních invariantních polynomů matice  $L$ , necht'  $Q(\alpha), \alpha \in \mathbb{R}^\gamma$  je parametrická matice odpovídající matici  $L$  a  $X(\alpha)$  jednoznačné řešení maticové rovnice*

$$AX - XL + BQ(\alpha) = 0. \quad (3.9)$$

Navíc, necht' pro  $F_0 \in \mathbb{R}^{m \times n}$  platí  $F_0X(\alpha) = 0$ ,

$$\psi(\alpha, F_0) = [\psi_{i,j}(\alpha, F_0)] \triangleq Q(\alpha)[X^T(\alpha)X(\alpha)]^{-1}X^T(\alpha) + F_0, \quad (3.10)$$

kde  $\alpha \in \mathcal{A} \triangleq \{\alpha \in \mathbb{R}^\gamma : \text{rank}[X(\alpha)] = s\}$ . Potom platí následující tvrzení:

- (i)  $\text{rank}[X(\alpha)] = s$  pro skoro všechna  $\alpha \in \mathbb{R}^\gamma$ .
- (ii) Existuje hustá podmnožina  $\mathcal{F}'_s$  množiny  $\mathcal{F}_s(A, B, L)$  taková, že libovolná matice  $F \in \mathcal{F}'_s$  může být vyjádřena ve tvaru  $F = \psi(\alpha, F_0)$ , kde  $\alpha \in \mathcal{A}$  a  $F_0$  splňuje podmínku  $F_0X(\alpha) = 0$ . Naopak, libovolná matice ve tvaru (3.10) patří do množiny  $\mathcal{F}_s(A, B, L)$ .
- (iii) Počet  $q$  všech parametrů v parametrizaci (3.10) ( $\gamma$  parametrů v  $\alpha$  a  $m(n-s)$  parametrů v  $F_0$ ) je dán

$$q = mn - \nu_1 - 3\nu_2 - \dots - (2k-1)\nu_k.$$

Podle věty 3.3 jsme schopni najít explicitní parametrizaci množiny všech stavových zpětných vazeb pomocí minimálního počtu parametrů. Při úplném přiřazení máme  $\gamma$  volných parametrů z matice  $Q(\alpha)$  a při neúplném přibude navíc  $m(n-s)$  parametrů. Tyto parametry mohou sloužit k tomu, abychom ze všech přípustných vazeb vybrali tu nejlepší.

Parametrická matice příslušející matici  $L$  je důležitým nástrojem při výpočtu zpětných vazeb. Algoritmy uvedené v [16] uvádí postup výpočtu explicitní parametrizace množiny všech stavových zpětných vazeb a rovněž algoritmus pro výpočet parametrické matice  $Q(\alpha)$ . Ten však vyžaduje i konstrukci zaměnitelné matice  $Z$ . Následující algoritmus popisuje nový způsob pro sestavení této parametrické matice, bez nutnosti vytváření zaměnitelné matice.

**Algoritmus 3.1. Výpočet parametrické matice  $Q(\alpha)$** 

vstup:  $L = \text{diag}\{L_1^1, L_1^2, \dots, L_1^{l_1} | L_2^1, L_2^2, \dots, L_2^{l_2} | \dots | L_k^1, L_k^2, \dots, L_k^{l_k}\} \in \mathbb{R}^{s \times s}$ ,  
 $L_i^j$  – reálné Jordanovy bloky velikosti  $n_i^j$  odpovídající  $\lambda_j \in \Lambda(L)$ ,  
 $n_1^j \geq n_2^j \geq \dots \geq n_k^j \geq 0, \quad \forall j \in \{1, 2, \dots, l = l_1\}$ ,  
 $\nu_i = \sum_{j=1}^{l_i} n_i^j, i = 1, 2, \dots, k, m$  počet vstupů (1),

výstup:  $Q(\alpha)$

- 1) přiřad' 0 do všech prvků  $Q(\alpha)$
- 2) přiřad' 1 následujícím prvkům  $Q(\alpha)$  (pro zajištění pozorovatelnosti  $(Q(\alpha), L)$  pro libovolný parametrický vektor  $\alpha \in \mathbb{R}^\gamma$ )

$$(1, 1), (1, n_1^1 + 1), \dots, (1, \sum_{i=1}^{l_1-1} n_1^i + 1);$$

$$(2, \nu_1 + 1), (2, \nu_1 + n_2^1 + 1), \dots, (2, \nu_1 + \sum_{i=1}^{l_1-1} n_2^i + 1);$$

...

$$(k, \sum_{i=1}^{k-1} \nu_i + 1), (k, \sum_{i=1}^{k-1} \nu_i + n_k^1 + 1), \dots, (k, \sum_{i=1}^{k-1} \nu_i + \sum_{i=1}^{l_k-1} n_k^i + 1);$$

- 3) přiřad' volné parametry do následujících prvků matice  $Q(\alpha)$

$$(j, 1), (j, 2), \dots, (j, n_1^1 - n_j^1);$$

$$(j, n_1^1 + 1), (j, n_1^1 + 2), \dots, (j, n_1^1 + n_1^2 - n_j^2);$$

...

$$(j, \sum_{i=1}^{l_1-1} n_1^i + 1), (j, \sum_{i=1}^{l_1-1} n_1^i + 2), \dots, (j, \sum_{i=1}^{l_1-1} n_1^i + n_1^{l_1} + n_j^{l_1});$$

pro  $j = 2, 3, \dots, k$ ;

$$(j, \nu_1 + 1), (j, \nu_1 + 2), \dots, (j, \nu_1 + n_2^1 - n_j^1);$$

$$(j, \nu_1 + n_2^1 + 1), (j, \nu_1 + n_2^1 + 2), \dots, (j, \nu_1 + n_2^1 + n_2^2 - n_j^2);$$

...

$$(j, \nu_1 + \sum_{i=1}^{l_2-1} n_2^i + 1), (j, \nu_1 + \sum_{i=1}^{l_2-1} n_2^i + 2), \dots, (j, \nu_1 + \sum_{i=1}^{l_2-1} n_2^i + n_2^{l_2} + n_j^{l_2});$$

pro  $j = 3, 4, \dots, k$ ;

$$(k, \sum_{i=1}^{k-2} \nu_i + 1), (k, \sum_{i=1}^{k-2} \nu_i + 2), \dots, (k, \sum_{i=1}^{k-2} \nu_i + n_{k-1}^1 - n_k^1);$$

$$(k, \sum_{i=1}^{k-2} \nu_i + n_{k-1}^1 + 1), (k, \sum_{i=1}^{k-2} \nu_i + n_{k-1}^1 + 2), \dots, (k, \sum_{i=1}^{k-2} \nu_i + n_{k-1}^2 - n_k^2);$$

...

$$(k, \sum_{i=1}^{k-2} \nu_i + \sum_{i=1}^{l_{k-1}-1} n_{k-1}^i + 1), (k, \sum_{i=1}^{k-2} \nu_i + \sum_{i=1}^{l_{k-1}-1} n_{k-1}^i + 2), \dots,$$

$$\dots (k, \sum_{i=1}^{k-2} \nu_i + \sum_{i=1}^{l_{k-1}-1} n_{k-1}^i + n_{k-1}^{l_{k-1}} - n_k^{l_{k-1}});$$

a  $(i, j)$ , pro  $i = k + 1, k + 2, \dots, m; j = 1, 2, \dots, s$ .

**Příklad 3.1.** Necht' matice  $B \in \mathbb{R}^{n \times m}$  a  $m = 3$ . Budeme chtít přiřadit následující Jordanovy bloky,

$$L_1 = \begin{bmatrix} -2 & 1 \\ 0 & -2 \end{bmatrix}, L_2 = \begin{bmatrix} -3 & 1 \\ 0 & -3 \end{bmatrix}, L_3 = \begin{bmatrix} -2 & 1 \\ 0 & -2 \end{bmatrix}, \\ L_4 = \begin{bmatrix} -3 \end{bmatrix}, L_5 = \begin{bmatrix} -2 \end{bmatrix},$$

tedy

$$L = \text{diag}\{L_1, L_2, L_3, L_4, L_5\}.$$

Nastiňme si, jak umístit parametry  $\alpha$  do matice  $Q(\alpha) \in \mathbb{R}^{3 \times 8}$ .

Blokům  $L_1, L_3, L_5$  přísluší vlastní číslo  $-2$ . Velikost největšího bloku je  $n_1 = 2$ , velikost druhého největšího, tj.  $L_3$ , je také  $n_3 = 2$ , proto v druhém řádku k vlastnímu číslu  $-2$ , tedy na pozicích  $Q_{2,1}, Q_{2,2}$  nebude žádný parametr,  $n_1 - n_3 = 0$ . Velikost třetího největšího bloku  $L_5$  pro vlastní číslo  $-2$  je  $n_5 = 1$ , tedy v třetím řádku bude  $n_1 - n_5 = 1$  parametr v prvku  $Q_{3,1}$  a  $n_3 - n_5 = 1$  parametr na pozici  $Q_{3,5}$ .

Vlastnímu číslu  $-3$  odpovídají bloky  $L_2$  a  $L_4$ , největší je  $L_2$ , jeho velikost je  $n_2 = 2$  a druhý největší je  $L_4$ , pro něj je  $n_4 = 1$ . V druhém řádku tak bude  $n_2 - n_4 = 1$  parametr na pozici  $Q_{2,3}$ . Nakonec, protože třetí blok, který by měl vlastní číslo  $-3$ , již není, bude od pozice  $Q_{3,2}$  následovat  $n_2 - 0 = 2$  parametrů a od  $Q_{3,7}$  potom  $n_4 - 0 = 1$  parametr  $\alpha$ . Celkem bude  $\alpha \in \mathbb{R}^6$  a matice  $Q(\alpha)$  bude mít tvar

$$Q(\alpha) = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_1 & 0 & 1 & 0 & 1 & 0 \\ \alpha_2 & 0 & \alpha_3 & \alpha_4 & \alpha_5 & 0 & \alpha_6 & 1 \end{bmatrix}.$$

**Příklad 3.2.** Uvažujme  $B \in \mathbb{R}^{n \times m}$ , kde  $m = 3$ . Budeme přiřazovat Jordanovy bloky uvedené v následující matici  $L$ ,

$$L = \text{diag} \left\{ \begin{bmatrix} -1 & 1 & 1 & 0 \\ -1 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} -1 \end{bmatrix} \right\}.$$

Potom parametrická matice bude následující

$$Q(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \alpha_1 & \alpha_2 & 0 & 0 & \alpha_3 & 0 & 1 & 0 & 1 \\ \alpha_4 & \alpha_5 & \alpha_6 & \alpha_7 & \alpha_8 & \alpha_9 & \alpha_{10} & \alpha_{11} & \alpha_{12} \end{bmatrix}.$$

**Příklad 3.3.** Mějme systém  $(A, B)$  daný maticemi

$$A = \begin{bmatrix} 1 & 0 & 1 & 3 & 0 \\ 0 & 1 & 1 & 2 & 1 \\ 1 & 0 & 1 & 3 & 3 \\ 2 & 2 & 0 & 2 & 1 \\ 0 & 1 & 2 & 2 & 3 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Chceme nalézt matici  $F \in \mathcal{F}_4$ , která přiřadí systému  $(A, B)$  Jordanovu formu

$$L = \text{diag} \left\{ \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix}, [-2], [-1] \right\}.$$

Jedná se tedy o neúplné přiřazení, ( $s < n$ ). Explicitní tvar matice stavové zpětné vazby je

$$F = \begin{bmatrix} r_1 & -\frac{1}{2} \frac{81\alpha_1 - 31\alpha_2 - 187 - 27\alpha_2\alpha_1 - 6r_1\alpha_1 - 52r_1\alpha_2 + 26r_1 + 12\alpha_1 r_1 \alpha_2}{-8\alpha_1 - 13\alpha_2 + 20 + 7\alpha_2\alpha_1} \\ r_2 & -\frac{1}{2} \frac{-6r_2\alpha_1 - 52r_2\alpha_2 + 26r_2 + 12\alpha_1 r_2 \alpha_2 - 103\alpha_1 - 289\alpha_2 + 293 + 107\alpha_2\alpha_1}{-8\alpha_1 - 13\alpha_2 + 20 + 7\alpha_2\alpha_1} \\ & \frac{170\alpha_1 r_1 \alpha_2 + 363\alpha_2\alpha_1 - 202r_1\alpha_1 - 465\alpha_1 - 309\alpha_2 + 1023 - 182r_1\alpha_2 + 406r_1}{4(-8\alpha_1 - 13\alpha_2 + 20 + 7\alpha_2\alpha_1)} \\ & \frac{327\alpha_2\alpha_1 - 202r_2\alpha_1 + 170\alpha_1 r_2 \alpha_2 - 393\alpha_1 + 406r_2 + 663 - 182r_2\alpha_2 - 201\alpha_2}{4(-8\alpha_1 - 13\alpha_2 + 20 + 7\alpha_2\alpha_1)} \\ & \frac{216\alpha_1 r_1 \alpha_2 + 459\alpha_2\alpha_1 - 270r_1\alpha_1 - 675\alpha_1 - 325\alpha_2 + 1517 - 208r_1\alpha_2 + 554r_1}{6(-8\alpha_1 - 13\alpha_2 + 20 + 7\alpha_2\alpha_1)} \\ & \frac{-267\alpha_1 + 247\alpha_2 + 277 + 183\alpha_2\alpha_1 - 270r_2\alpha_1 + 216\alpha_1 r_2 \alpha_2 + 554r_2 - 208r_2\alpha_2}{6(-8\alpha_1 - 13\alpha_2 + 20 + 7\alpha_2\alpha_1)} \\ & \frac{-1278r_1\alpha_1 + 1098\alpha_1 r_1 \alpha_2 + 2546r_1 - 1222r_1\alpha_2 + 6173 - 2835\alpha_1 + 2349\alpha_2\alpha_1 - 2131\alpha_2}{12(-8\alpha_1 - 13\alpha_2 + 20 + 7\alpha_2\alpha_1)} \\ & \frac{-1278r_2\alpha_1 + 1098\alpha_1 r_2 \alpha_2 + 2546r_2 - 1222r_2\alpha_2 + 4453 + 2241\alpha_2\alpha_1 - 1679\alpha_2 - 2619\alpha_1}{12(-8\alpha_1 - 13\alpha_2 + 20 + 7\alpha_2\alpha_1)} \end{bmatrix},$$

kde  $\alpha_1 \neq \frac{13\alpha_2 - 20}{-8 + 7\alpha_2}$ .

Návrhové parametry jsou  $\alpha_1, \alpha_2$  (z matice  $Q(\alpha)$ ) a  $r_1, r_2$  (vzniklé následkem neúplného přiřazení). Všechny tyto volné parametry můžeme využít k následné optimalizaci vlastností matice dynamiky uzavřeného systému, či regulátoru.

## 3.2 Výstupní zpětná vazba

Pokud nemáme k dispozici všechny složky stavového vektoru, ale jen některé z nich nebo jejich lineární kombinace, stává se problém přiřazení Jordanovy formy daleko obtížnější. Přiřazení Jordanovy formy (stejně jako přiřazení pólů) výstupní zpětnou vazbou spadá do třídy NP-těžkých problémů.

Pro systém (3.1) zavedeme výstupní zpětnou vazbu

$$u(t) = Ky(t), \quad (3.11)$$

kde  $K \in \mathbb{R}^{m \times p}$ .

Dále označme množinu všech výstupních zpětných vazeb

$$\mathcal{K}_s(A, B, L) \triangleq \left\{ K \in \mathbb{R}^{m \times p} : (A + BK) \sim \begin{bmatrix} L & * \\ 0 & * \end{bmatrix} \right\}, \quad (3.12)$$

kde  $L \in \mathbb{R}^{s \times s}$ ,  $s \leq n$ , je daná reálná Jordanova forma a symboly  $*$  označují libovolné reálné matice. Jestliže  $s = n$  hovoříme o úplném přiřazení, pokud  $s < n$ , jedná se o neúplné přiřazení.

V práci [16] jsou uvedeny tři přístupy k nalezení výstupní zpětné vazby. Kvůli následné optimalizaci vlastností systému (3.1) se budeme soustředit na přístup, který převádí problém nalezení výstupní zpětné vazby  $K \in \mathcal{K}_s$  na úlohu nalezení stavové zpětné vazby  $F \in \mathcal{F}_s$  (jiného ekvivalentního systému) takové, že  $F$  má  $m(n-p)$  nulových prvků.

Uvažujme systém (3.1) a definujme následující transformaci

$$z = Tx = \begin{bmatrix} C \\ D \end{bmatrix} x, \quad (3.13)$$

kde matice  $D \in \mathbb{R}^{(n-p) \times n}$  je volena tak, aby matice  $T = \begin{bmatrix} C \\ D \end{bmatrix}$  byla regulární.

Odvodíme ekvivalentní systém  $(\bar{A}, \bar{B}, \bar{C})$  pro vektor  $z$ ,

$$\begin{aligned} \dot{z} &= T\dot{x} = TAx + TBu = TAT^{-1}z + TBu \\ y &= CT^{-1}z. \end{aligned} \quad (3.14)$$

Protože matici  $T$  jsme volili regulární, můžeme uvažovat  $T^{-1}$ . Platí  $TT^{-1} = I$ , tj.  $\begin{bmatrix} C \\ D \end{bmatrix} T^{-1} = I$ , a potom je  $CT^{-1} = \begin{bmatrix} I_p & 0 \end{bmatrix}$ . Vztah (3.14) můžeme dále upravit

$$\begin{aligned} \dot{z} &= T\dot{x} = TAx + TBu = TAT^{-1}z + TBu \\ y &= CT^{-1}z = \begin{bmatrix} I_p & 0 \end{bmatrix} z. \end{aligned}$$

Následně obdržíme systém  $(\bar{A}, \bar{B}, \bar{C})$

$$\begin{aligned} \dot{z} &= \bar{A}z + \bar{B}u \\ y &= \bar{C}z, \end{aligned} \quad (3.15)$$

kde  $\bar{A} = TAT^{-1}$ ,  $\bar{B} = TB$ ,  $\bar{C} = \begin{bmatrix} I_p & 0 \end{bmatrix}$ ,  $I_p \in \mathbb{R}^{p \times p}$ .

Pro systém (3.15) máme

$$\begin{aligned} u &= Ky \\ u &= K\bar{C}z \\ u &= Fz, \quad \text{pro } F = K\bar{C}. \end{aligned}$$

**Poznámka 3.1.** Naznačme souvislost mezi systémy

$$\begin{aligned}\bar{A} + \bar{B}F &\sim L \\ TAT^{-1} + TBF &\sim L.\end{aligned}\tag{3.16}$$

Pro (3.16) tedy existuje regulární matice  $X$  taková, že

$$X^{-1}(TAT^{-1} + TBF)X = L.$$

Úpravami získáme

$$\begin{aligned}X^{-1}TAT^{-1}X + X^{-1}TBFX &= L \\ AT^{-1}X - T^{-1}XL + BFX &= 0.\end{aligned}$$

Pro  $F = KCT^{-1}$  (tj.  $F = K\bar{C}$ ) a  $P = T^{-1}X$  máme dále

$$\begin{aligned}AP - PL + BKCP &= 0 \\ P^{-1}(A + BKC)P &= L, \quad \text{neboli} \\ A + BKC &\sim L.\end{aligned}$$

Pokud tedy nalezneme  $F \in \mathcal{F}_s$  takovou, že pro  $F = \begin{bmatrix} F_1 & F_2 \end{bmatrix}$ , kde  $F_1 \in \mathbb{R}^{m \times p}$  a  $F_2 \in \mathbb{R}^{m \times (n-p)}$ , pro níž bude

$$F_2 = 0,\tag{3.17}$$

můžeme využít vztahů

$$\begin{aligned}F &= K\bar{C} = K \begin{bmatrix} I_p & 0 \end{bmatrix} = \begin{bmatrix} K & 0 \end{bmatrix} \quad \text{a} \\ F &= \begin{bmatrix} F_1 & 0 \end{bmatrix}\end{aligned}$$

a získáme, že výstupní zpětnou vazbu  $K$  můžeme vyjádřit jako  $K = F_1$ .

Výhodou tohoto přístupu je převod na výpočet stavové zpětné vazby. Naopak obtíž zde vzniká právě při nulování daných prvků, tj. prvků na pozicích  $f_{i,j}$  pro  $i = p + 1, p + 2, \dots, n$  a  $j = 1, \dots, m$ .

**Příklad 3.4.** Uvažujme systém  $(A, B, C)$ , kde

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Chceme navrhnout výstupní zpětnou vazbu  $K \in \mathcal{K}_4$ , která systému  $(A, B, C)$  bude přiřazovat Jordanovu formu

$$L = \text{diag} \left\{ \begin{bmatrix} -2 & 1 \\ 0 & -2 \end{bmatrix}, [-1], [-4] \right\}.$$

Podle uvedeného postupu dostaneme matici  $F(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ , pro kterou bude  $F_2 = 0$  pro

$$\alpha_3 = \frac{1}{2} \frac{350\alpha_4 + 49 + 300\alpha_4\alpha_1 - 417\alpha_1}{450\alpha_4 - 357\alpha_1 + 29},$$

$$\alpha_2 = -\frac{1}{540} \frac{-3213\alpha_1^2 + 2985\alpha_4\alpha_1 + 1545\alpha_1 - 98 - 1295\alpha_4}{5\alpha_4 - 1}.$$

Potom explicitní tvar výstupní zpětné vazby je

$$K = \begin{bmatrix} \frac{3(-1083\alpha_1 + 60\alpha_4\alpha_1 + 211 + 730\alpha_4)}{(3\alpha_4 - 1)(5\alpha_4 - 1)} & \frac{3(150\alpha_4 - 357\alpha_1 + 89)}{-5\alpha_4 - 3\alpha_1 + 1 + 15\alpha_4\alpha_1} \\ -\frac{2(-12834\alpha_1 + 675\alpha_4\alpha_1 + 2548 + 8425\alpha_4)}{50\alpha_4 - 153\alpha_1 + 41} & -49 \\ & -12 \\ & \frac{49 - 500\alpha_4 + 1350\alpha_4\alpha_1 - 117\alpha_1}{50\alpha_4 - 153\alpha_1 + 41} \end{bmatrix},$$

kde  $\alpha_1 \neq \frac{1}{5}$ ,  $\alpha_4 \neq \frac{1}{3}$ ,  $\alpha_1 \neq \frac{153}{50}\alpha_4 - \frac{41}{50}$ .

Volné parametry  $\alpha_1, \alpha_4$  mohou být voleny na základě různých kritérií, jak budeme uvažovat dále.

## 4 Robustnost uzavřeného systému

Podle předchozí kapitoly umíme explicitně vypočítat stavovou zpětnou vazbu pro systém (3.1). Získaný uzavřený systém je stabilní pro libovolnou stavovou zpětnou vazbu z množiny  $\mathcal{F}_s(A, B, L)$ . V matici stavové zpětné vazby se vyskytují návrhové parametry. Díky těmto parametrům můžeme zvolit matici  $F \in \mathcal{F}_s(A, B, L)$  tak, aby uzavřený systém byl v nějakém smyslu nejlepší.

Prvky matice dynamiky otevřeného systému jsou určeny hodnotami veličin reálného modelu. U mechanických soustav například hmotnosti, délky, tuhosti pružin, u elektrických soustav například hodnoty odporů, indukčností apod. Tyto hodnoty ovšem neznáme přesně. Proto chceme, aby i při perturbaci všech prvků matice dynamiky zůstal uzavřený systém stabilní. V tomto případě hovoříme o robustní stabilitě systému.

Podobně můžeme uvažovat perturbace ve zpětné vazbě. Často je totiž možné realizovat příslušný regulátor pouze na malý počet platných cifer. Pokud takový regulátor uijeme ke stabilizaci systému, budeme opět chtít, aby uzavřený systém byl stabilní. Tímto požadujeme, aby regulátor nebyl fragilní (křehký).

Uvažujme systém  $(A, B)$  a vypočtěme pro něj stavovou zpětnou vazbu. Matice uzavřeného systému  $A_c$  bude  $A_c = A + BF$ . Od uzavřeného systému požadujeme, aby zůstal stabilní i při určitých změnách, perturbacích, v prvcích uvažovaných matic. Perturbační matici budeme označovat  $\Delta$ .

Pro robustní stabilitu systému chceme, aby matice

$$(A + \Delta) + BF = (A + BF) + \Delta = A_c + \Delta$$

zůstala stabilní.

Pro zabránění fragilitě (křehkosti) požadujeme, aby systém

$$A + B(F + \Delta) = (A + BF) + B\Delta = A_c + B\Delta$$

zůstal stabilní.

Obdobně mějme systém  $(A, B, C)$  a spočtěme pro něj výstupní zpětnou vazbu. Matice uzavřeného systému je potom  $A_c = A + BKC$ .

Pro robustní systém chceme najít co nejlepší matici  $K$ , aby matice

$$(A + \Delta) + BKC = (A + BKC) + \Delta = A_c + \Delta$$

zůstala stabilní.



Pro zabránění fragilitě systému při perturbacích ve výstupní zpětné vazbě požadujeme, aby matice

$$A + B(K + \Delta)C = (A + BKC) + B\Delta C = A_c + B\Delta C$$

zůstala stabilní.

## 4.1 Kritéria robustnosti

V následujících podkapitolách si uvedeme významná kritéria robustnosti pro uzavřený systém se stavovou a výstupní zpětnou vazbou, jejich vlastnosti a metody pro výpočet. Kritériím robustnosti se věnují například v pracích [3, 13, 21, 28]. Nejprve si nyní nastíníme význam těchto kritérií na obecné stabilní matici  $X \in \mathbb{C}^{n \times n}$ .

Jelikož uvažujeme, že  $X$  je stabilní matice, leží prvky jejího spektra  $\Lambda(X)$  v otevřené levé polorovině komplexní roviny. Při posuzování, jak je daná matice blízko nestabilní matici, by prvním přirozeným kritériem mohlo být to, jak je vlastní číslo, které se nachází nejvíce vpravo, daleko od imaginární osy, čímž dospějeme k tzv. spektrální abscisse, označované  $\alpha(X)$ ,

$$\alpha(X) = \max\{\operatorname{Re} z : z \in \Lambda(X)\}.$$

Takové kritérium by ale nebylo příliš rozumné, protože  $\alpha(X)$  je velice citlivá na změny v prvcích matice. Převážně tedy chceme vědět, zda matice zatížena jistými perturbacemi zůstane stabilní, resp. které perturbace v prvcích matice již vedou k její nestabilitě. Vezměme si matici perturbací  $\Delta \in \mathbb{C}^{n \times n}$ ,  $\|\Delta\|_2 < \varepsilon$ , kterou přičítáme k  $X$ . Pokud všechny body spektra matice  $X + \Delta$  zůstanou v levé polorovině komplexní roviny, můžeme tyto perturbace v  $X$  povolit a matice  $X$  zůstane stabilní. Tímto přejdeme od spektra  $X$  k tzv. komplexnímu  $\varepsilon$ -pseudospektru

$$\Lambda_\varepsilon^{\mathbb{C}}(X) = \{z : z \in \Lambda(X + \Delta) \text{ pro } \Delta \in \mathbb{C}^{n \times n}, \|\Delta\|_2 \leq \varepsilon\}.$$

Prvek této množiny, jehož reálná část je nejbližší imaginární ose, by vedl na další kritérium, tzv.  $\varepsilon$ -pseudospektrální abscissu  $\alpha_\varepsilon(X)$ :

$$\alpha_\varepsilon(X) = \max\{\operatorname{Re} z : z \in \Lambda_\varepsilon(X)\}.$$

Pokud bychom povolili pouze reálné perturbace, tj.  $\Delta \in \mathbb{R}^{n \times n}$ , dospěli bychom k reálnému  $\varepsilon$ -pseudospektru

$$\Lambda_\varepsilon^{\mathbb{R}}(X) = \{z : z \in \Lambda(X + \Delta) \text{ pro } \Delta \in \mathbb{R}^{n \times n}, \|\Delta\|_2 \leq \varepsilon\}.$$

Pro zjištění, jak daleko je matice od nestability, by bylo vhodné určit velikost perturbací  $\varepsilon$ , pro kterou by nějaký bod množiny  $\Lambda_\varepsilon(X)$  ležel na imaginární ose. To nás vede k velice významnému kritériu – poloměru stability:

$$r_F = \inf\{\|\Delta\| : \Delta \in F^{n \times n} \text{ a } X + \Delta \text{ je nestabilní}\},$$

kde pro  $F = \mathbb{C}$  získáme komplexní a pro  $F = \mathbb{R}$  reálný poloměr stability.

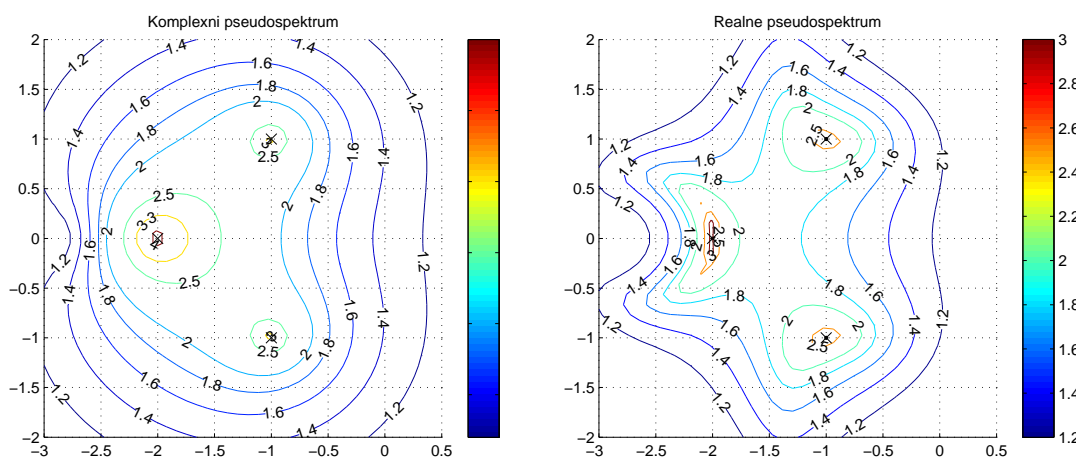
V praktických příkladech nás zajímají převážně reálné perturbace, ale  $r_{\mathbb{R}}(X)$  je výpočetně mnohem složitější. Komplexní poloměr stability má však také svůj význam. Nejenže dává dolní odhad pro reálný poloměr stability, ale také např. souvisí s  $\varepsilon$ -pseudospektrální abscissou v tom smyslu, že

$$r_{\mathbb{C}}(X) \leq \varepsilon \Leftrightarrow \alpha_{\varepsilon}(X) \geq 0.$$

**Příklad 4.1.** Ukažme si význam jednotlivých kritérií robustní stability na matici

$$X = \begin{bmatrix} 4.0000 & 2.1017 & 14.7288 & -28.1017 \\ 4.0000 & -1.4237 & 5.7966 & -11.5763 \\ 7.0000 & 1.6780 & 19.5254 & -40.6780 \\ 5.0000 & 1.1017 & 13.7288 & -28.1017 \end{bmatrix}.$$

Na obrázku 4.1a a 4.1b je vykreslené komplexní a reálné pseudospektrum. Hodnoty jednotlivých hladin jsou uvedené pro lepší názornost v logaritmickém měřítku (o základu 10).



(a) Komplexní pseudospektrum.

(b) Reálné pseudospektrum.

Obrázek 4.1: Vykreslení pseudospekter.

Spektrum matice  $X$  je  $\Lambda(X) = \{-1 + j, -1 - j, -2, -2\}$ , spektrální abscissa je  $\alpha(X) = -1$ . Vypočtená hodnota komplexního poloměru stability je 0.0404. Rovněž u komplexního pseudospektra vidíme, že hladina s hodnotou přibližně 1.394 v logaritmickém měřítku, tj. 0.0404, se jako první dotýká imaginární osy.  $\varepsilon$ -pseudospektrální abscissa pro tuto hodnotu  $\varepsilon$  je nula a body, v nichž dosahuje svého maxima jsou  $z = \pm 0.8589j$ .

Reálný poloměr stability je přibližně 0.0602. Rovněž na pseudospektru vidíme, že hladina s hodnotou přibližně 1.2201 v logaritmickém měřítku, tj. 0.0602, se dotýká imaginární osy.

## 4.2 Poloměr stability

Jedním z kritérií pro posouzení robustnosti či fragility systému je poloměr stability. Rozeznáváme poloměr stability komplexní ( $r_{\mathbb{C}}$ ) a reálný ( $r_{\mathbb{R}}$ ). K získání robustního systému či málo fragilního regulátoru se snažíme najít takové řízení, aby výsledný systém měl maximální poloměr stability. Pokud budeme uvažovat robustnost systému, jedná se o poloměr stability, pokud nás bude zajímat fragilita regulátoru, budeme hovořit o strukturovaném poloměru stability.

### 4.2.1 Komplexní poloměr stability

Uvažujme stabilní matici  $X \in \mathbb{R}^{n \times m}$ ,  $n \leq m$  a  $\text{rank}(X) = n$ . Komplexní poloměr stability nám dává informaci o tom, jak daleko je matice  $X$  od nejbližší nestabilní matice.

Podobný problém představuje známá věta o aproximaci matice maticí nižší hodnosti, kde zkoumáme vzdálenost matice  $X$  a nejbližší matice dané hodnosti  $k \neq n$ . Libovolnou matici můžeme podle (2.2) rozložit na  $X = \sum_{i=1}^n \sigma_i(X) u_i(X) v_i^*(X)$ , neboli na součet nezávislých matic hodnosti jedna. Nejbližše se k matici  $X$  přiblížíme, budeme-li vhodně kopírovat tuto její strukturu. Uvažujme tak matici  $X_{\text{aprox}} = \sum_{i=1}^k \sigma_i(X) u_i(X) v_i^*(X)$  s hodností  $k$ . Vzdálenost mezi  $X$  a  $X_{\text{aprox}}$  bude odpovídat největšímu singulárnímu číslu matice  $(X - X_{\text{aprox}})$ , tedy  $\sigma_{k+1}(X)$ . Například vzdálenosti mezi maticí  $X$  a nejbližší maticí nižší hodnosti  $k = n - 1$  pak bude odpovídat velikost nejmenšího singulárního čísla  $\sigma_n(X)$ .

Mezi stabilní maticí a nejbližšími nestabilními maticemi můžeme očekávat obdobný vztah. Matice, které budou destabilizovat matici  $X$  budou dány opět singulárními vektory a vzdálenost bude určena singulárním číslem nějaké vhodné matice. Ta bude záviset na volbě stabilní oblasti, resp. její hranici. Protože nás opět zajímá vzdálenost k nejbližší nestabilní matici, budeme vybírat minimální hodnotu singulárního čísla vhodné matice mezi všemi maticemi danými zvolenou oblastí.

**Definice 4.1.** Komplexní poloměr stability matice dynamiky uzavřeného systému  $(A_c, B, C)$ ,  $A_c \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$  je

$$r_{\mathbb{C}}(A_c) = \inf \{ \|\Delta\| : \Delta \in \mathbb{C}^{n \times n} \text{ a } A_c + \Delta \text{ je nestabilní} \}.$$

Matice  $A_c$  představuje matici uzavřeného systému zpětnou vazbou, který je již stabilní. Pro stavovou zpětnou vazbu by bylo

$$A_c = A + BF, \quad (4.1)$$

pro výstupní bychom měli

$$A_c = A + BKC. \quad (4.2)$$

**Poznámka 4.1.** V dalším textu budeme maticí  $A_c$  označovat stabilní matici danou buď vztahem (4.1), nebo (4.2).

Pro dosažení systému, který nebude fragilní uvažujeme následující modifikaci.

**Definice 4.2.** Strukturovaný komplexní poloměr stability systému  $(A_c, B, C)$  definujeme jako

$$r_{\mathbb{C}}(A_c, B, C) \triangleq \inf\{\|\Delta\| : \Delta \in \mathbb{C}^{m \times p} \text{ a } A_c + B\Delta C \text{ je nestabilní}\}, \quad (4.3)$$

kde  $A_c \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$ .

Systém  $(A, B, C)$  bude nestabilní, pokud některé z vlastních čísel matice dynamiky  $A$  bude ležet v pravé polorovině komplexní roviny. Případně o systému  $(A, B, C)$  řekneme, že je na mezi stability, pokud některá vlastní čísla matice dynamiky budou ležet na imaginární ose a ostatní v otevřené levé polorovině. Obecně uvažujeme oblast  $\mathbb{C}_g$ , pro níž je systém stabilní, potom tedy bude nestabilní pro  $\mathbb{R}^2 \setminus \mathbb{C}_g \triangleq \mathbb{C}_b$ , tedy již i pro vlastní čísla na  $\partial\mathbb{C}_g$ .

**Věta 4.1.**

$$r_{\mathbb{C}}(A_c, B, C) = \inf\{\|\Delta\| : \Delta \in \mathbb{C}^{m \times p}, \Lambda(A_c + B\Delta C) \cap \partial\mathbb{C}_g \neq \emptyset\} \quad (4.4a)$$

$$= \inf_{s \in \partial\mathbb{C}_g} \inf_{\Delta} \{\|\Delta\| : \Delta \in \mathbb{C}^{m \times p}, \det(sI - A_c - B\Delta C) = 0\} \quad (4.4b)$$

$$= \inf_{s \in \partial\mathbb{C}_g} \inf_{\Delta} \{\|\Delta\| : \Delta \in \mathbb{C}^{m \times p}, \det[I - \Delta C(sI - A_c)^{-1}B] = 0\} \quad (4.4c)$$

$$= \left\{ \sup_{s \in \partial\mathbb{C}_g} \sigma_1[C(sI - A_c)^{-1}B] \right\}^{-1} \quad (4.4d)$$

**Důkaz.** Komplexní poloměr stability (strukturovaný) jsme definovali jako  $\inf\{\|\Delta\| : \Delta \in \mathbb{C}^{m \times p}, A_c + B\Delta C \text{ je nestabilní}\}$ . Matice  $A_c + B\Delta C$  je nestabilní, jestliže její vlastní čísla nenáleží do množiny  $\mathbb{C}_g$ . Ze spojitosti vlastních čísel matice v závislostech na perturbacích na vstupech (podle věty 2.10) plyne, že vlastní čísla přecházející z množiny  $\mathbb{C}_g$  do množiny  $\mathbb{C}_b$  musí ležet právě na  $\partial\mathbb{C}_g$ . Proto lze požadavek  $A_c + B\Delta C$

je nestabilní matice nahradit požadavkem, aby  $A_c + B\Delta C$  měla vlastní čísla na  $\partial\mathbb{C}_g$ . Tím jsme získali vztah (4.4a).

Nyní ukažme odvození vztahu (4.4b). Vyjděme z (4.4a) a necht'  $s \in \partial\mathbb{C}_g$ . Potom  $s$  bude vlastním číslem matice  $A_c + B\Delta C$  a  $\det(sI - A_c - B\Delta C) = 0$ . Tento vztah budeme postupně upravovat, přičemž k předposlední rovnosti dojdeme podle věty 2.5.

$$\begin{aligned} \det(sI - A_c - B\Delta C) &= \det[(sI - A_c)(I - (sI - A_c)^{-1}B\Delta C)] = \\ &= \det(sI - A_c) \det[I - (sI - A_c)^{-1}B\Delta C] = \\ &= \det(sI - A_c) \det[I - \Delta C(sI - A_c)^{-1}B] = 0. \end{aligned}$$

O matici  $A_c$  předpokládáme, že je stabilní, proto  $\det(sI - A_c) \neq 0$ . Potom bude ale

$$\det[I - \Delta C(sI - A_c)^{-1}B] = 0. \quad (4.5)$$

Tím jsme získali předpis (4.4c).

Důkaz vztahu (4.4d) budeme provádět ověřením obou nerovností  $r_C \geq \{\sup_{s \in \partial\mathbb{C}_g} \sigma_1[C(sI - A_c)^{-1}B]\}^{-1}$  a  $r_C \leq \{\sup_{s \in \partial\mathbb{C}_g} \sigma_1[C(sI - A_c)^{-1}B]\}^{-1}$ . Uvažujme nejprve první nerovnost. Podle vztahu (4.4a) zvolme  $s \in \partial\mathbb{C}_g \cap \Lambda(A_c + B\Delta C)$ . Potom existuje  $x$  takové, že

$$(A_c + B\Delta C)x = sx.$$

Roznásobením dostaneme

$$A_c x + B\Delta C x = s x. \quad (4.6)$$

Protože matice dynamiky uzavřeného systému je stabilní, musí být  $s \notin \Lambda(A_c)$ , tedy  $A_c x \neq s x$ ,  $B\Delta C x \neq 0$  a  $Cx \neq 0$ . Označme  $y = Cx$  a upravujme rovnost (4.6).

$$\begin{aligned} (A_c + B\Delta C)x &= s x, \\ A_c x + B\Delta y &= s x, \\ B\Delta y &= (sI - A_c)x, \\ (sI - A_c)^{-1}B\Delta y &= x, \quad /C, \\ C(sI - A_c)^{-1}B\Delta y &= Cx, \\ C(sI - A_c)^{-1}B\Delta y &= y. \end{aligned}$$

Necht'  $G(s) \triangleq C(sI - A_c)^{-1}B$ , potom  $G(s)\Delta y = y$  a

$$\begin{aligned} \|y\| &= \|G(s)\Delta y\| \leq \|G(s)\| \|\Delta\| \|y\|, \\ 1 &\leq \|G(s)\| \|\Delta\|, \\ \frac{1}{\|G(s)\|} &\leq \|\Delta\|. \end{aligned}$$

Protože dané vztahy platí pro libovolné  $s \in \Lambda(A_c + B\Delta C) \cap \partial\mathbb{C}_g$  získáme, že

$$\|\Delta\| \geq \frac{1}{\{\sup_{s \in \partial\mathbb{C}_g} \sigma_1[C(sI - A_c)^{-1}B]\}^{-1}} \quad \text{a}$$

$$r_{\mathbb{C}} \geq \frac{1}{\{\sup_{s \in \partial\mathbb{C}_g} \sigma_1[C(sI - A_c)^{-1}B]\}^{-1}}.$$

Naopak, označme opět  $G(s) \triangleq C(sI - A_c)^{-1}B$  a necht' suprema  $\|G(s)\|$  je pro  $s \in \partial\mathbb{C}_g$  dosaženo v  $s_0$ . Předpokládejme nyní  $m \geq p$  (pro  $m < p$  bychom postupovali analogicky). Singulární rozklad matice  $G(s_0)$  (při  $\text{rank}[G(s_0)] = r$ ) bude

$$G(s_0) = \sum_{j=1}^r s_j u_j v_j^*,$$

kde  $s_j$  jsou singulární čísla,  $u_j$  levé singulární vektory a  $v_j$  pravé singulární vektory. Definujme  $\bar{\Delta} = \frac{1}{s_1} v_1 u_1^*$ . Upravíme výraz  $G(s_0)\bar{\Delta}u_1$  s využitím unitarity singulárních vektorů ( $v_j^* v_i = 0$  pro  $j \neq i$  a  $v_j^* v_j = 1$ )

$$\begin{aligned} G(s_0)\bar{\Delta}u_1 &= \sum_{j=1}^r s_j u_j v_j^* \frac{1}{s_1} v_1 u_1^* u_1 = \sum_{j=1}^r s_j \frac{1}{s_1} u_j v_j^* v_1 = \\ &= s_1 \frac{1}{s_1} u_1 v_1^* v_1 + s_2 \frac{1}{s_1} u_2 v_2^* v_1 + \dots + s_r \frac{1}{s_1} u_r v_r^* v_1 = u_1. \end{aligned}$$

Potom

$$\begin{aligned} G(s_0)\bar{\Delta}u_1 &= u_1, \\ C(s_0 I - A_c)^{-1} B \bar{\Delta} u_1 &= u_1, \quad / B \bar{\Delta} \\ B \bar{\Delta} C (s_0 I - A_c)^{-1} B \bar{\Delta} u_1 &= B \bar{\Delta} u_1, \\ B \bar{\Delta} C (s_0 I - A_c)^{-1} B \bar{\Delta} u_1 &= (s_0 I - A_c) (s_0 I - A_c)^{-1} B \bar{\Delta} u_1, \\ (B \bar{\Delta} C + A) (s_0 I - A_c)^{-1} B \bar{\Delta} u_1 &= s_0 (s_0 I - A_c)^{-1} B \bar{\Delta} u_1, \\ (A + B \bar{\Delta} C) [(s_0 I - A_c)^{-1} B \bar{\Delta} u_1] &= s_0 [(s_0 I - A_c)^{-1} B \bar{\Delta} u_1], \end{aligned}$$

pro  $z = (s_0 I - A_c)^{-1} B \bar{\Delta} u_1$  máme

$$(A + B \bar{\Delta} C) z = s_0 z.$$

Mezi vlastní čísla matice  $A + B \bar{\Delta} C$  patří  $s_0$ . Celkem je tak  $s_0 \in \Lambda(A_c + B \bar{\Delta} C) \cap \partial\mathbb{C}_g$ . Podle (4.4a) a protože  $\sigma_1[C(s_0 I - A_c)^{-1}B] \leq \sup_{s \in \partial\mathbb{C}_g} \sigma_1[C(sI - A_c)^{-1}B]$  dostaneme

$$\begin{aligned} r_{\mathbb{C}} &= \inf\{\|\Delta\| : \Delta \in \mathbb{C}^{m \times p}, \Lambda(A_c + B\Delta C) \cap \partial\mathbb{C}_g \neq \emptyset\} \leq \\ &\leq \|\bar{\Delta}\| = \frac{1}{s_1} = \frac{1}{\sigma_1[C(s_0 I - A_c)^{-1}B]} \leq \frac{1}{\sup_{s \in \partial\mathbb{C}_g} \sigma_1[C(sI - A_c)^{-1}B]}. \end{aligned}$$

□

Vztah (4.4d), který ukazuje, že  $r_{\mathbb{C}}$  lze vyjádřit pomocí singulárního čísla matice  $C(sI - A_c)^{-1}B$ , se nejlépe hodí při výpočtech.

**Poznámka 4.2.** Při odvozování rovnosti (4.4d) ve větě 4.1 můžeme využít vztahu mezi singulárními čísly a hodnotí matic (věta 2.9). Platí

$$\inf\{\|\Delta\| : \Delta \in \mathbb{C}^{m \times p}, \det[I - \Delta C(sI - A_c)^{-1}B] = 0\}.$$

Podle věty 2.5 je  $\det[I - \Delta C(sI - A_c)^{-1}B] = \det[I - C(sI - A_c)^{-1}B\Delta]$  a podle věty 2.9 pro  $M = C(sI - A_c)^{-1}B$  budeme mít

$$\begin{aligned} r_{\mathbb{C}}(A_c, B, C) &= \inf_{s \in \partial\mathbb{C}_g} \inf_{\Delta} \{\|\Delta\| : \Delta \in \mathbb{C}^{m \times p}, \det[I - \Delta C(sI - A_c)^{-1}B] = 0\} = \\ &= \inf_{s \in \partial\mathbb{C}_g} \frac{1}{\sigma_1[C(sI - A_c)^{-1}B]} = \frac{1}{\sup_{s \in \partial\mathbb{C}_g} \sigma_1[C(sI - A_c)^{-1}B]}. \end{aligned}$$

Pokud  $B = C = I_n$  bude mít vztah (4.4d) tvar

$$r_{\mathbb{C}}(A_c) = \frac{1}{\sup_{s \in \partial\mathbb{C}_g} \sigma_1[(sI - A_c)^{-1}]} = \inf_{s \in \partial\mathbb{C}_g} \sigma_1[(sI - A_c)^{-1}] = \inf_{s \in \partial\mathbb{C}_g} \sigma_{\min}[(A_c - sI)].$$

**Věta 4.2.** Necht'  $\mathbb{C}_g = \{s \in \mathbb{C} : \operatorname{Re}(s) < 0\}$  a  $B = C = I_n$ , potom  $r_{\mathbb{C}}(A_c) \leq \sigma_{\min}(A_c)$ .

**Důkaz.** Necht' se pro nějaké  $s_{\text{opt}} \in \partial\mathbb{C}_g = \{s \in \mathbb{C} : \operatorname{Re}(s) = 0\}$  nabývá optima. Potom pro libovolné  $s \in \partial\mathbb{C}_g$ , tedy i pro  $s = 0$ , budeme mít

$$r_{\mathbb{C}}(A_c) = \sigma_{\min}[(A_c - s_{\text{opt}}I)] \leq \sigma_{\min}(A_c).$$

□

**Věta 4.3.** Necht'  $\mathbb{C}_g = \{s \in \mathbb{C} : \operatorname{Re}(s) < 0\}$ , potom

$$r_{\mathbb{C}} = \left\{ \sup_{\omega \in (0, \omega_{\max})} \sigma_1[C(j\omega I - A_c)^{-1}B] \right\}^{-1}, \quad (4.7)$$

kde  $\omega \in \mathbb{R}$  a  $|\omega_{\max}| \leq \sigma_1(A_c) + \frac{\sigma_1(C)\sigma_1(B)}{\sigma_1(CA_c^{-1}B)}$ .

**Důkaz.** Necht'  $\mathbb{C}_g = \{s \in \mathbb{C} : \operatorname{Re}(s) < 0\}$ , potom  $\partial\mathbb{C}_g = \{j\omega \in \mathbb{C} : \omega \in \mathbb{R}\}$ . Vztah (4.7) plyne přímo ze vztahu (4.4d). Podle (4.4a) je matice  $j\omega I - (A_c + B\Delta C)$  singulární. Následně podle věty 2.8 (pro  $M = j\omega I$ , kde pro  $\omega \neq 0$  je  $M$  nesingulární) a protože  $\|(j\omega I)^{-1}\|^{-1} = |\omega|$  platí, že  $|\omega| \leq \|A_c + B\Delta C\|$ . Dalšími úpravami získáme

$$|\omega| \leq \|A_c + B\Delta C\| \leq \|A_c\| + \|B\Delta C\| \leq \|A_c\| + \|B\|\|\Delta\|\|C\|.$$

Dále pro  $\|\Delta_{\text{opt}}\|$  (matici z věty 4.1, pro níž ve vztahu (4.4a) získáme hodnotu  $r_{\mathbb{C}}$ ) a  $s = j\omega$  podle (4.4d) platí, že

$$\|\Delta_{\text{opt}}\| \leq \frac{1}{\sigma_1[C(j\omega I - A_c)^{-1}]} \leq \frac{1}{\sigma_1(CA_c^{-1}B)}.$$

Celkem s využitím věty 2.6 dostaneme

$$|\omega_{\max}| \leq \sigma_1(A_c) + \frac{\sigma_1(B)\sigma_1(C)}{\sigma_1(CA_c^{-1}B)}.$$

□

Pro výpočet komplexního poloměru stability potřebujeme umět spočítat (4.4d). Jak jsme uvedli, zajímá nás případ, že  $\partial\mathbb{C}_g$  je imaginární osa, tedy ve vztahu (4.4d) je  $s = j\omega$ , kde  $\omega \in \mathbb{R}$ . Výsledné kritérium – komplexní strukturovaný poloměr stability (resp. pro  $B = C = I_n$  komplexní poloměr stability matice dynamiky uzavřeného systému) – bude mít tvar

$$r_C(A_c, B, C) = \frac{1}{\sup_{\omega \in \mathbb{R}} \sigma_1[C(j\omega I - A_c)^{-1}B]}. \quad (4.8)$$

Pro praktické postupy musíme nejprve zvolit nějaký interval  $\langle a, b \rangle$ , na kterém se budeme pohybovat. Podle věty 4.3 a můžeme volit interval  $\langle 0, \omega_{\max} \rangle$ , kde  $\omega_{\max}$  je určeno ve větě 4.3. Nyní si uveďme metody, pomocí nichž můžeme vypočítat hodnotu komplexního poloměru stability.

Jednou z možností je použití hrubé síly a na podintervalech celkového intervalu aplikovat optimalizační metody. Vhodnější metody pro výpočet (4.4d) potom navrhl Byers a Boyd s Balakrishnanem.

### Byersova metoda bisekce

Byers v [8] navrhl metodu bisekce pro případ, že  $B = C = I_n$ . V [5] Boyd a Balakrishnan rozšířili tuto metodu pro obecný stabilní systém  $(A_c, B, C)$ . Byersova metoda bisekce, podobně jako metoda bisekce pro hledání kořenů rovnice, vychází z počátečního intervalu a v každé iteraci modifikuje jednu z jeho mezí podle rozhodovacího kritéria. Jedná se o efektivnější metodu, než ty, které vyčíslují  $\sigma_1[C(j\omega I - A_c)^{-1}B]$  pro  $\omega \in \mathbb{R}$  a hledají maximum optimalizačními metodami. Je založena na vztahu mezi singulárními čísly matice v (4.8) pro  $\omega \in \mathbb{R}$  a imaginárními vlastními čísly příslušné Hamiltonovy matice, který uvádí následující dvě věty.

Pro daný systém  $(A_c, B, C)$  a  $\gamma \in \mathbb{R}$  zavedeme matici

$$\begin{aligned} M_\gamma &= \begin{bmatrix} A_c & 0 \\ 0 & -A_c^T \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & -C^T \end{bmatrix} \begin{bmatrix} 0 & \gamma I \\ \gamma I & 0 \end{bmatrix}^{-1} \begin{bmatrix} C & 0 \\ 0 & B^T \end{bmatrix} = \\ &= \begin{bmatrix} A_c & \frac{1}{\gamma} B B^T \\ -\frac{1}{\gamma} C^T C & -A_c^T \end{bmatrix}. \quad (4.9) \end{aligned}$$

Matice  $M_\gamma$  je podle věty 2.4 Hamiltonova.



**Věta 4.4.** Předpokládejme, že matice  $A_c$  nemá imaginární vlastní čísla,  $\gamma > 0$  a  $\omega_0 \in \mathbb{R}$ . Potom  $\gamma$  je singulární číslo matice  $C(j\omega_0 I - A_c)^{-1}B$  právě tehdy, když  $M_\gamma - j\omega_0 I$  je singulární.

**Důkaz.** Ukažme platnost implikace zleva doprava. Matice  $A_c$  nemá imaginární vlastní čísla, tedy pro imaginární číslo  $s = j\omega$ , kde  $\omega \in \mathbb{R}$ , můžeme uvažovat výraz  $(j\omega I - A_c)^{-1}$ . Necht'  $\gamma$  je singulárním číslem matice  $C(j\omega_0 I - A_c)^{-1}B$ , potom necht'  $u, v$  jsou příslušné singulární vektory,  $u \neq 0$ ,  $v \neq 0$ .

$$\begin{aligned} C(j\omega_0 I - A_c)^{-1}Bu &= \gamma v \\ [C(j\omega_0 I - A_c)^{-1}B]^* v &= \gamma u, \end{aligned}$$

$$\begin{aligned} C(j\omega_0 I - A_c)^{-1}Bu &= \gamma v \\ B^T(j\omega_0 I - A_c)^{-*}C^T v &= \gamma u, \end{aligned}$$

$$\begin{aligned} C(j\omega_0 I - A_c)^{-1}Bu &= \gamma v \\ B^T(-j\omega_0 I - A_c^T)^{-1}C^T v &= \gamma u, \end{aligned}$$

označme  $r = (j\omega_0 I - A_c)^{-1}Bu$  a  $s = (-j\omega_0 I - A_c^T)^{-1}C^T v$ , tj.

$$\begin{aligned} Cr &= \gamma v \text{ a } B^T s = \gamma u \\ u &= \frac{1}{\gamma} B^T s \text{ a } v = \frac{1}{\gamma} Cr. \end{aligned}$$

Potom

$$\begin{aligned} \begin{bmatrix} u \\ v \end{bmatrix} &= \begin{bmatrix} 0 & \frac{1}{\gamma} \\ \frac{1}{\gamma} & 0 \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix}, \\ \begin{bmatrix} u \\ v \end{bmatrix} &= \begin{bmatrix} 0 & \frac{1}{\gamma} \\ \frac{1}{\gamma} & 0 \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & B^T \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix}, \\ \begin{bmatrix} u \\ v \end{bmatrix} &= \begin{bmatrix} 0 & \gamma I \\ \gamma I & 0 \end{bmatrix}^{-1} \begin{bmatrix} C & 0 \\ 0 & B^T \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix}, \\ \begin{bmatrix} u \\ v \end{bmatrix} &\neq \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \text{ proto i } \begin{bmatrix} r \\ s \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \end{aligned} \tag{4.10}$$

Pokračujme dále od vztahu pro  $r$  a  $s$ :

$$\begin{aligned} r &= (j\omega_0 I - A_c)^{-1}Bu \\ s &= (-j\omega_0 I - A_c^T)^{-1}C^T v, \end{aligned}$$

$$\begin{aligned} (j\omega_0 I - A_c)r &= Bu \\ (-j\omega_0 I - A_c^T)s &= C^T v, \end{aligned}$$

$$\begin{aligned} j\omega_0 r - A_c r &= B u \\ -j\omega_0 s - A_c^T s &= C^T v, \end{aligned}$$

Celkem

$$\begin{bmatrix} j\omega_0 I & 0 \\ 0 & j\omega_0 I \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} + \begin{bmatrix} -A_c & 0 \\ 0 & A_c^T \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} B & 0 \\ 0 & -C^T \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}.$$

Dosazením vztahu (4.10) získáme

$$\begin{aligned} \begin{bmatrix} j\omega_0 I & 0 \\ 0 & j\omega_0 I \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} + \begin{bmatrix} -A_c & 0 \\ 0 & A_c^T \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} &= \\ \begin{bmatrix} B & 0 \\ 0 & -C^T \end{bmatrix} \begin{bmatrix} 0 & \gamma I \\ \gamma I & 0 \end{bmatrix}^{-1} \begin{bmatrix} C & 0 \\ 0 & B^T \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix}, & \end{aligned}$$

$$\begin{aligned} \begin{bmatrix} A_c & 0 \\ 0 & -A_c^T \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & -C^T \end{bmatrix} \begin{bmatrix} 0 & \gamma I \\ \gamma I & 0 \end{bmatrix}^{-1} \begin{bmatrix} C & 0 \\ 0 & B^T \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} &= j\omega_0 \begin{bmatrix} r \\ s \end{bmatrix}, \\ \left\{ \begin{bmatrix} A_c & 0 \\ 0 & -A_c^T \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & -C^T \end{bmatrix} \begin{bmatrix} 0 & \gamma I \\ \gamma I & 0 \end{bmatrix}^{-1} \begin{bmatrix} C & 0 \\ 0 & B^T \end{bmatrix} \right\} \begin{bmatrix} r \\ s \end{bmatrix} &= j\omega_0 \begin{bmatrix} r \\ s \end{bmatrix}, \end{aligned}$$

a protože

$$M_\gamma = \begin{bmatrix} A_c & 0 \\ 0 & -A_c^T \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & -C^T \end{bmatrix} \begin{bmatrix} 0 & \gamma I \\ \gamma I & 0 \end{bmatrix}^{-1} \begin{bmatrix} C & 0 \\ 0 & B^T \end{bmatrix}$$

je

$$M_\gamma \begin{bmatrix} r \\ s \end{bmatrix} = j\omega_0 \begin{bmatrix} r \\ s \end{bmatrix}$$

a tedy  $j\omega_0$  je vlastním číslem matice  $M_\gamma$  a  $\begin{bmatrix} r \\ s \end{bmatrix}$  příslušný vlastní vektor.

Při důkazu opačné implikace postupujeme analogicky. □

Z věty 4.4 plyne následující věta, která nám dává rozhodovací kritérium, zda máme přenastavit dolní, nebo horní mez.

**Věta 4.5.** ([5]) *Nechť  $A_c$  je stabilní,  $\gamma > 0$ . Potom  $\sup_{\omega \in \mathbb{R}} \sigma_1[C(j\omega I - A_c)^{-1}B] \geq \gamma$  právě tehdy, když  $M_\gamma$  má imaginární vlastní číslo.*

**Důkaz.** Necht'  $M_\gamma$  má alespoň jedno imaginární číslo, tj.  $\exists j\bar{\omega}$ ,  $\bar{\omega} \in \mathbb{R} : (M_\gamma - j\bar{\omega}I)$  je singulární. Podle věty 4.4  $\sigma_i[C(j\bar{\omega}I - A_c)^{-1}B] = \gamma$ , potom  $\sup_{\omega \in \mathbb{R}} \sigma_1[C(j\omega I - A_c)^{-1}B] \geq \gamma$ . Naopak necht'  $\sup_{\omega \in \mathbb{R}} \sigma_1[C(j\omega I - A_c)^{-1}B] \geq \gamma$ . Protože  $\sigma_1[C(j\omega I -$

$A_c)^{-1}B]$  je spojitou funkcí v  $\omega$ , existuje  $\bar{\omega} : \sigma_1[C(j\bar{\omega}I - A_c)^{-1}B] = \gamma$ . Potom  $(M_\gamma - j\bar{\omega}I)$  je singulární a  $M_\gamma$  má imaginární vlastní číslo.  $\square$

Podmínku, zda  $M_\gamma$  nemá imaginární vlastní čísla, není nutné ověřovat jejich přímým výpočtem. Vlastní čísla matice  $M_\gamma$  nemůžeme přesně spočítat v konečném počtu kroků. Proto je lepší využít vlastností Hamiltonových matic.

Označme charakteristický polynom matice  $M_\gamma$  jako  $r(s)$ , tj.  $r(s) = \det(sI - M_\gamma)$ . Protože  $M_\gamma$  je Hamiltonova matice, má podle věty 2.3 vlastní čísla symetrická podle imaginární osy.

Potom můžeme rozepsat charakteristický polynom  $r(s)$  následovně

$$\begin{aligned} r(s) &= r_n(s - \lambda_1)(s + \lambda_1)(s - \lambda_2)(s + \lambda_2) \dots (s - \lambda_n)(s + \lambda_n) = \\ &= r_n(s^2 - \lambda_1^2)(s^2 - \lambda_2^2) \dots (s^2 - \lambda_n^2). \end{aligned} \quad (4.11)$$

Charakteristický polynom matice  $M_\gamma$  je polynomem v proměnné  $s^2$ , lze tak zavést polynom  $p(-s^2)$ .

**Věta 4.6.** *Matice  $M_\gamma$  má imaginární vlastní číslo právě tehdy, když polynom  $p(-s^2)$  má reálné nezáporné kořeny.*

**Důkaz.** Plyne bezprostředně ze vztahu (4.11), definujeme-li  $p(-s^2) = r_n(-1)^{-n}(s^2 + \lambda_1^2) \dots (s^2 + \lambda_n^2)$ .  $\square$

**Poznámka 4.3.** V článku [5] je k určení, zda  $M_\gamma$  má imaginární vlastní číslo, využívána Sturmova posloupnost polynomů získaných opakovanou aplikací Euklidova algoritmu. Nejprve se dělí charakteristický polynom jeho derivací a poté vždy předchozí dělitel polynomem vzniklým jako zbytek po dělení. Následně se využije toho, že polynom  $p(-s^2)$  nemá reálné nezáporné kořeny, když  $v_0 = v_\infty$ , kde  $v_0$  udává počet změn znamének polynomů ve Sturmově posloupnosti pro 0 a  $v_\infty$  témuž pro  $\infty$ .

Při numerickém výpočtu dělení polynomů však může docházet k nestabilitě a výsledky bychom získávali nepřesné. Proto v této práci navážeme při rozhodování o vlastních číslech  $M_\gamma$  na větu 4.6. Uvažujme, že polynom má reálné kořeny. Pokud budeme předpokládat malé nepřesnosti ve vstupních datech a reálné kořeny budou od sebe dostatečně vzdálené, bude se měnit nejprve jejich poloha na reálné ose. Dvě reálná vlastní čísla se k sobě budou přibližovat po reálné ose, až dosáhnou stejného bodu, odchýlí se do imaginární části komplexní roviny a stanou se komplexně sdruženými vlastními čísly. Pro velmi blízké kořeny může samozřejmě i pro malé perturbace nastat posun do imaginární části. Protože ale nepotřebujeme spočítat přesné hodnoty kořenů, ale jen určit, zda jsou některé z nich reálné, budeme se spoléhat na uvedený přístup.

### Algoritmus 4.1. Výpočet komplexního poloměru – Byersova metoda bisekce, modifikovaná verze

Vstup:  $A_c, (B, C), \gamma_l, \gamma_u, \text{tol}$  (tolerance)

while  $\gamma_u + \gamma_l > 2\text{tol}\gamma_l$

a)  $\gamma = \frac{\gamma_u + \gamma_l}{2}$

b) sestav  $M_\gamma$  (4.9)

c) sestav modifikovaný charakteristický polynom  $p(-s^2)$  ((4.11), např. Leverierův-Faddějevův algoritmus ([23]))

d) vypočti kořeny  $p_i$  polynomu  $p(-s^2)$  (např. převodem na výpočet vlastních čísel companion matice ([11]))

f) if  $\max_i \{\text{Re}(p_i)\} > 0$  then (tj.  $M_\gamma$  má imaginární vlastní číslo)

$\gamma_l = \gamma$

else

$\gamma_u = \gamma$

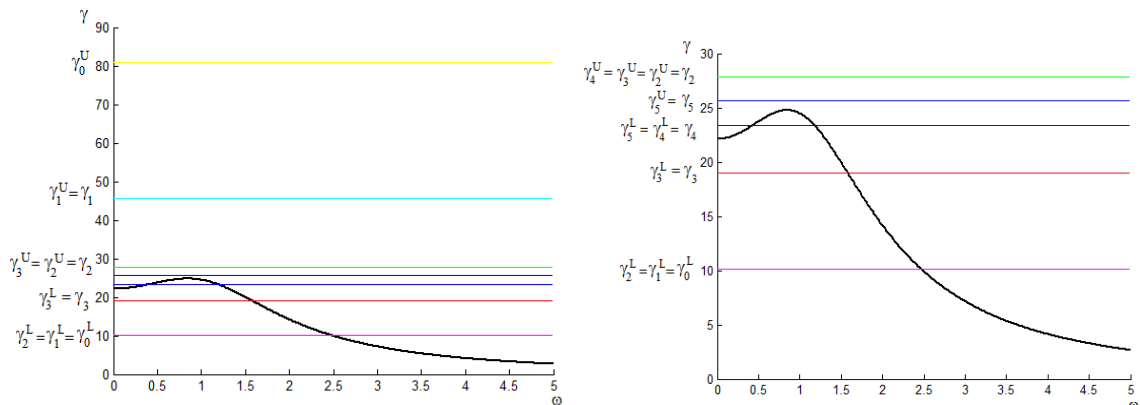
end

end

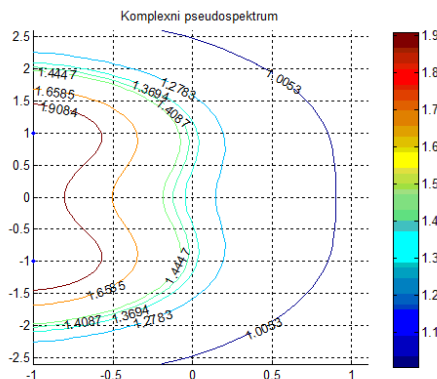
Výstup:  $r_{\mathbb{C}} = \gamma^{-1}$

**Příklad 4.2.** Pro matici  $X$  z příkladu 4.1 a startovací interval  $\langle 10.1235, 80.9878 \rangle$  znázorňují obrázky 4.2a a 4.2b několik prvních iterací metody bisekce,  $\gamma_k^L$  označuje dolní odhad v  $k$ -té iteraci,  $\gamma_k^U$  označuje horní odhad v  $k$ -té iteraci.

Podobně, jak znázorňuje obrázek 4.3, bychom viděli i na pseudospektru, které hladiny při iteracích volíme.



Obrázek 4.2: Byersova metoda bisekce, (a) několik prvních iterací, (b) detailní výřez.



Obrázek 4.3: Metoda bisekce – pseudospektrum.

### Boydova-Balakrishnanova metoda

Metoda v [4] navržená Boydem a Balakrishnanem využívá toho, že algoritmem pro výpočet vlastních čísel Hamiltonovy matice (podkapitola 2.4.1) můžeme vypočítat imaginární vlastní čísla matice  $M_\gamma$  (4.9). Řekněme tedy, že vypočteme požadovaná  $j\omega_1, \dots, j\omega_r$ . Tím se dostaneme na imaginární osu a potom již musíme jen určit způsob, jak se po ní pohybovat, abychom našli  $\sup_{\omega \in \mathbb{R}} \sigma_1[C(j\omega I - A_c)^{-1}B]$ . Označujme dále opět matici  $G(j\omega) \triangleq C(j\omega I - A_c)^{-1}B$ .

Díky symetrii okolo reálné osy se můžeme pro  $\omega$  soustředit jen na interval  $\langle 0, \infty \rangle$ , resp. podle věty 4.3 jen na  $\langle 0, \omega_{\max} \rangle$ . V souvislosti s definicí pseudospektra 4.5, pokud najdeme hladinu  $h_1$ , která protíná v několika bodech imaginární osu, chceme v dalším kroku najít hladinu  $h_2$  (s větší hodnotou než  $h_1$ ), až bychom postupně dospěli k hladině  $h_k$ , pro kterou bude v bodě  $\omega$  (odpovídající vlastnímu číslu  $j\omega$ , resp.  $-\omega$  pro vlastní číslo  $-j\omega$ ) imaginární osa její tečnou.

Nejprve potřebujeme odhad  $\gamma_0 \leq \sup_{\omega \in \mathbb{R}} \sigma_1[G(j\omega)]$ . Pro toto  $\gamma_0$  bude mít  $M_\gamma$  obecně několik ryze imaginárních vlastních čísel  $j\omega_l$  takových, že hodnota nějakého singulárního čísla matice  $G(j\omega_l)$  je rovna  $\gamma_0$ .

Jedna z metod, jak vybrat správné intervaly, vychází z následující úvahy a vede na Boydův Balakrishnanův algoritmus.

**Pozorování 4.1.** Pro  $\gamma_k$ , (kde  $k$  je příslušná iterace), je  $\gamma_k^2 I - G^*(j\omega_l)G(j\omega_l)$  pozitivně semidefinitní matice právě tehdy, když  $\sigma_1[G(j\omega_l)] = \gamma_k$ .

**Důkaz.** Nejprve vezměme  $\sigma_1^2[G(j\omega_l)] = \gamma_k^2 = \lambda_1[G^*(j\omega_l)G(j\omega_l)]$ , potom  $\lambda_1[\gamma_k^2 I - G^*(j\omega_l)G(j\omega_l)] \geq 0$ , pro každé  $i = 1, \dots, r$ . Naopak uvažujme, že  $\gamma_k^2 I - G^*(j\omega_l)G(j\omega_l)$  je pozitivně semidefinitní a že  $\sigma_m[G(j\omega_l)] = \gamma_k$  pro nějaké  $m > 1$  a  $\sigma_m[G(j\omega_l)] \neq$

$\sigma_1[G(j\omega_l)]$ . Potom, protože  $\sigma_1 > \sigma_m$ , je  $\lambda_1[\gamma_k^2 I - G^*(j\omega_l)G(j\omega_l)] < 0$ , což vede ke sporu s pozitivní semidefinitností.  $\square$

Protože nás zajímají  $\sigma_1[G(j\omega)]$ , plyne z předchozího pozorování, že z vypočtených ryze imaginárních vlastních čísel  $(j\omega_1, \dots, j\omega_r)$  vyřadíme ta  $j\omega_l$ , pro která  $\gamma_k^2 I - G^*(j\omega_l)G(j\omega_l)$  není pozitivně semidefinitní matice. Pro zbylá  $j\omega_l, \dots, j\omega_p$  určují hodnoty  $\omega_l, \dots, \omega_p$  body, v nichž křivka  $\sigma_1[G(j\omega)]$  bude protínat přímkou  $\gamma = \gamma_k$ . Dále zbývá určit, které hodnoty  $\omega$  určují meze intervalů pro další iteraci.

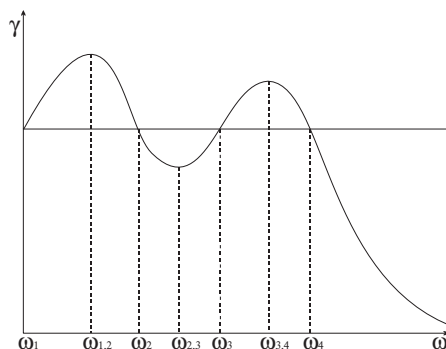
**Pozorování 4.2.** Pro  $\omega_l, \omega_m \in \{\omega_1, \omega_2, \dots, \omega_p\}$  taková, že  $\omega_l < \omega_r$  má matice  $\gamma_k^2 I - G^* [j\frac{1}{2}(\omega_l + \omega_m)] G [j\frac{1}{2}(\omega_l + \omega_m)]$  záporné vlastní číslo, právě když  $\sigma_1[G(j\frac{1}{2}(\omega_l + \omega_m))] > \gamma_k$ .

Pozorování 4.2 nám pomáhá určit, v jakých intervalech bude

$$\omega_{\text{opt}} = \arg \max_{\omega \in \mathbb{R}} \sigma_1[C(j\omega I - A_c)^{-1}B].$$

Tyto intervaly budou  $(\omega_l, \omega_m)$ , pro které je splněno, že  $\sigma_1[G(j\frac{1}{2}(\omega_l + \omega_m))] > \gamma_k$ . Hodnotu  $\max \{ \sigma_1 [G(j\frac{\omega_l + \omega_m}{2})] \}$  budeme v této práci volit za  $\gamma_{k+1}$ . V [4] volí za hodnotu  $\gamma_{k+1}$  největší singulární číslo v  $j\bar{\omega}$ , kde  $\bar{\omega}$  je střed největšího intervalu.

Průběh jedné iterace algoritmu je znázorněn na obrázku 4.4. Hodnota  $\gamma_k$  v iteraci



Obrázek 4.4: Boydova-Balakrishnanova metoda, volba intervalů na základě definitnosti matice.

$k$  protíná křivku  $\sigma_1[G(j\omega)]$  v bodech  $\omega_1, \omega_2, \omega_3, \omega_4$ . V bodech  $\omega = \omega_{1,2}, \omega = \omega_{3,4}$  má matice  $\gamma_k^2 I - G^*(j\omega)G(j\omega)$  záporné vlastní číslo, bod  $\omega_{2,3}$  vyloučíme. Za  $\gamma_{k+1}$  pro další iteraci volíme hodnotu  $\sigma_1[G(j\omega_{1,2})]$ .

### Algoritmus 4.2. Výpočet komplexního poloměru – Boydova - Balakrishna- nova metoda

Vstup:  $A_c, (B, C), \gamma_0, \text{tol}$  (tolerance),  $k = 0$

- a) pro  $\gamma = \gamma_k$  sestav  $M_\gamma$  (podle vztahu (4.9))
- b) vypočti ryze imaginární vlastní čísla  $j\omega_i$  matice  $M_\gamma: \Lambda_I(M_\gamma)$  (algoritmus 2.3), pokud  $\Lambda_I(M_\gamma) = \emptyset$ , pokračuj na krok g)
- c) z množiny  $\Lambda_I(M_\gamma)$  odeber ta vlastní čísla, pro něž je matice  $\gamma_k^2 I - G^*(j\omega_i)G(j\omega_i)$  pozitivně definitní, pokud odebereme všechna, jdi na g)
- d) urči všechna  $\omega_s = \frac{\omega_i + \omega_{i+1}}{2}$  (kde  $j\omega_i \in \Lambda_I(M_\gamma)$ ), pro která  $\gamma_k^2 I - G^*(j\omega_s)G(j\omega_s)$  má záporné vlastní číslo
- e)  $\gamma_{k+1} = \max_s \sigma_1[G(j\omega_s)]$ , kde  $\omega_s$  jsou body z kroku d)
- f)  $k = k + 1$ , pokračuj na krok a)
- g)  $\gamma_{k+1} = \gamma_k$ , konec

Výstup:  $r_{\mathbb{C}} = \gamma_{k+1}^{-1}$

V [4] je ukázáno, že konvergence této metody je vždy alespoň kvadratická.

Jiná metoda nalezení správných intervalů pro další iteraci vychází z [17] a [26] a její princip je následující. Seřaďme sestupně imaginární části příslušných ryze imaginárních vlastních čísel matice  $M_\gamma$

$$\omega_1 > \omega_2 > \omega_3 > \dots \omega_r.$$

Opět chceme zjistit, jakému  $\sigma_i[G(j\omega_m)]$  vlastní číslo  $j\omega_m$  náleží. Uvažujme obecně křivku  $\gamma_k = \sigma_i[G(j\omega)]$ ,  $\omega \in \mathbb{R}$ , která v okolí bodu  $\omega_m$  odpovídá některému  $\sigma_i[G(j\omega_m)]$ . V bodě  $\omega_1$  bude křivka  $\gamma_k$  klesat.

Abychom zjistili, které křivce  $\sigma_i[G(j\omega_m)]$  odpovídají další body  $\omega_m$ , kde  $m > 1$ , zajímá nás, zda je křivka  $\gamma$  v tomto bodě rostoucí, či klesající. To můžeme zjistit podle derivace  $\frac{\partial \gamma}{\partial \omega}(\omega_m)$ .

Vzhledem k tomu, že  $\omega_m$  jsme získali jako vlastní čísla  $M_\gamma$ , je  $\omega$  závislé na hodnotě  $\gamma$ . Potom lze hledanou derivaci vyjádřit jako

$$\frac{\partial \gamma}{\partial \omega}(\omega_m) = \left( -j \frac{\partial \lambda}{\partial \gamma}(\gamma_k) \right)^{-1}, \quad (4.12)$$

pro  $\lambda_m = j\omega_m$ , přičemž  $\frac{\partial \lambda}{\partial \gamma}(\gamma_k)$  lze podle [27] vyjádřit následovně

$$\frac{\partial \lambda_m}{\partial \gamma}(\gamma_k) = u_m^* \frac{\partial M_\gamma}{\partial \gamma}(\gamma_k, A_c, B, C) v_m, \quad (4.13)$$

kde  $v_m$  je normalizovaný vlastní vektor odpovídající vlastnímu číslu  $\lambda_m$ ,  $u_m^*$  je jeho levý vlastní vektor matice  $M_\gamma$ .

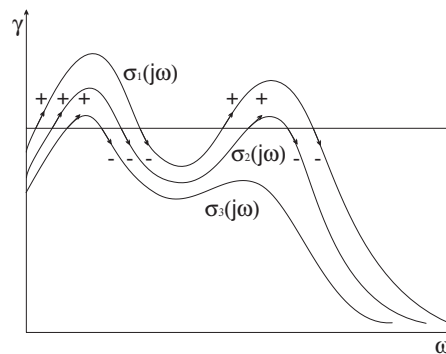
Víme, že v bodě  $\omega_1$  bude  $\frac{\partial \gamma}{\partial \omega}(\omega_1)$  záporná. Pokud by v  $\omega_2$  byla derivace kladná, vyplyne ze spojitosti singulárních čísel a jejich vzájemného uspořádání, že  $\omega_2$  přísluší křivce  $\sigma_1[G(j\omega)]$ . Pokud by ale v  $\omega_2$  byla derivace záporná, musí  $\omega_2$  odpovídat  $\sigma_2[G(j\omega)]$ . Analogicky bychom postupovali dále pro další body  $\omega_m$ . Dále vypočítáme, že pokud znaménka ve dvou po sobě jdoucích  $\omega$  jsou různá, budou náležet těmto  $\sigma_i[G(j\omega)]$ .

Označme nyní  $s_m = \text{sgn} \frac{\partial \gamma}{\partial \omega}(\omega_m)$  znaménko derivace v  $\omega_m$ , pro  $m = 1, 2, \dots, r$  a  $c_m$  pro  $m = 1, 2, \dots, r$  pořadí singulárního čísla, tj.  $c_m = 1$  určuje křivku  $\sigma_1$ . Potom víme, že  $c_r = 1$ , a dále postupujeme podle následujících pravidel:

- je-li  $s_{m+1} < 0$  a  $s_m < 0$ , potom  $c_m = c_{m+1} + 1$ ,
  - je-li  $s_{m+1} < 0$  a  $s_m > 0$ , potom  $c_m = c_{m+1}$ ,
  - je-li  $s_{m+1} > 0$  a  $s_m < 0$ , potom  $c_m = c_{m+1}$ ,
  - je-li  $s_{m+1} > 0$  a  $s_m > 0$ , potom  $c_m = c_{m+1} - 1$ .
- (4.14)

Takto bychom vybrali všechna  $\omega_m$ , pro něž je  $c_m = 1$ , a získali bychom příslušné intervaly  $(\omega_{r-1}, \omega_r), (\omega_{r-3}, \omega_{r-2}), \dots$ , které odpovídají největšímu singulárnímu číslu  $\sigma_1[G(j\omega)]$ . Obrázek 4.5 ukazuje, jak podle pravidel (4.14) získáme požadovaná  $\omega_i$ .

**Poznámka 4.4.** Kladné znaménko u  $\omega_1$ , tj.  $s_1 > 0$ , by znamenalo, že dolní mez daného intervalu by byla v bodě  $-\omega_1$ , a protože jsme na intervalu  $\langle 0, \omega_{\max} \rangle$ , volili bychom interval  $\langle 0, \omega_1 \rangle$ .



Obrázek 4.5: Volba intervalů na základě znaménka derivace, Sreedharův přístup.



**Algoritmus 4.3. Výpočet komplexního poloměru – Sreedharův přístup**

Vstup:  $A_c, (B, C), \gamma_0, \text{tol}$  (tolerance),  $k = 0$

- a) pro  $\gamma = \gamma_k$  sestav  $M_\gamma$  (podle vztahu (4.9))
- b) vypočti ryze imaginární vlastní čísla  $j\omega_i$  matice  $M_\gamma$ :  $\Lambda_I(M_\gamma)$  (algoritmus 2.3), pokud  $\Lambda_I(M_\gamma) = \emptyset$ , pokračuj na krok h)
- c) vypočti  $s_i$ , pro  $i = 1, \dots, r$
- d) urči  $\omega_i$ , pro něž  $c_i = 1$  a sestav intervaly, pokud je délka největšího intervalu menší než  $\text{tol}$ , pokračuj na h)
- e) spočti středy  $\omega_p$  intervalů z předchozího bodu
- f)  $\gamma_{k+1} = \max_p \{\sigma_1[G(j\omega_p)]\}$
- g)  $k = k + 1$
- h)  $\gamma_{k+1} = \gamma_k$ , konec

Výstup:  $r_{\mathbb{C}} = \gamma_{k+1}^{-1}$

Komplexní (strukturovaný) poloměr stability můžeme tak počítat pomocí algoritmů 4.1, 4.2, 4.3. Algoritmem 4.1 můžeme poměrně snadno (s malým počtem vyčíslování hodnot) vypočítat přibližnou hodnotu  $r_{\mathbb{C}}$ . Algoritmy 4.2, 4.3 jsou potom vhodné pro přesnější výpočty, navíc jsou i rychleji konvergentní a nevyžadují velký počet iterací. V ukázkových příkladech v této práci je komplexní (strukturovaný) poloměr stability počítán pomocí algoritmu 4.3.

**4.2.2 Reálný poloměr stability**

Jak již bylo řečeno, reálný poloměr stability můžeme definovat obdobně jako komplexní, jen uvažujeme pouze reálné perturbace.

**Definice 4.3.** Reálný poloměr stability matice dynamiky uzavřeného systému  $(A_c, B, C)$ ,  $A_c \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$  je

$$r_{\mathbb{R}}(A_c) = \inf \{ \|\Delta\| : \Delta \in \mathbb{R}^{n \times n} \text{ a } A_c + \Delta \text{ je nestabilní} \}.$$

Matice  $A_c$  je opět maticí dynamiky uzavřeného systému zpětnou vazbou, který je již stabilní. Pro stavovou zpětnou vazbu bychom měli  $A_c = A + BF$ , pro výstupní  $A_c = A + BKC$ .

Pro posuzování fragility regulátoru vezměme opět následující modifikaci.

**Definice 4.4.** Strukturovaný reálný poloměr stability systému  $(A_c, B, C)$ ,  $A_c \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$

$$r_{\mathbb{R}}(A_c, B, C) \triangleq \inf \{ \|\Delta\| : \Delta \in \mathbb{R}^{m \times p} \text{ a } A_c + B\Delta C \text{ je nestabilní} \}. \quad (4.15)$$

Obdobně jako u věty 4.1 bychom také mohli zavést vztahy pro reálný poloměr stability, s výjimkou bodu (4.4d), který zde neplatí.

**Věta 4.7.**

$$\begin{aligned} r_{\mathbb{R}}(A_c, B, C) &= \\ &= \inf\{\|\Delta\| : \Delta \in \mathbb{R}^{m \times p}, \Lambda(A_c + B\Delta C) \cap \partial C_g \neq \emptyset\} \\ &= \inf_{s \in \partial C_g} \inf_{\Delta} \{\|\Delta\| : \Delta \in \mathbb{R}^{m \times p}, \det(sI - A_c - B\Delta C) = 0\} \\ &= \inf_{s \in \partial C_g} \inf_{\Delta} \{\|\Delta\| : \Delta \in \mathbb{R}^{m \times p}, \det[I - \Delta C(sI - A_c)^{-1}B] = 0\}. \end{aligned}$$

Výpočetní tvar pro  $r_{\mathbb{R}}$  našli Qui, Bernhardsson, Rantzer a spol. a jeho důkaz je uveden v [22]. Zde ho pro stručnost vypouštíme.

**Věta 4.8.** ([22])

$$r_{\mathbb{R}}(A_c, B, C) = \left[ \max_{s \in \partial C_g} \mu_R[C(sI - A_c)^{-1}B] \right]^{-1}, \quad (4.16)$$

kde

$$\begin{aligned} \mu_R[C(sI - A_c)^{-1}B] &= \\ &= \inf_{\gamma \in (0,1)} \sigma_2 \left( \begin{bmatrix} \operatorname{Re} C(sI - A_c)^{-1}B & -\gamma \operatorname{Im} C(sI - A_c)^{-1}B \\ \frac{1}{\gamma} \operatorname{Im} C(sI - A_c)^{-1}B & \operatorname{Re} C(sI - A_c)^{-1}B \end{bmatrix} \right). \end{aligned}$$

**Věta 4.9.** ([22])

Funkce  $\sigma_2 \left( \begin{bmatrix} \operatorname{Re} C(sI - A_c)^{-1}B & -\gamma \operatorname{Im} C(sI - A_c)^{-1}B \\ \frac{1}{\gamma} \operatorname{Im} C(sI - A_c)^{-1}B & \operatorname{Re} C(sI - A_c)^{-1}B \end{bmatrix} \right)$  je kvazikonvexní.

Kvazikonvexní funkce má konvexní množinu řešení a její lokální minimum bude tedy i globálním minimem. Hodnotu  $\mu_R[C(sI - A_c)^{-1}B]$  můžeme tak určit snadno například metodou zlatého řezu (nebo také metodou bisekce apod.).

Opět nás bude zajímat případ, kdy  $\partial C_g = \{s \in \mathbb{C} : \operatorname{Re}(s) = 0\}$ , tj.  $\{j\omega \in \mathbb{C}, \omega \in \mathbb{R}\}$ . Vztah (4.16) přejde do tvaru

$$r_{\mathbb{R}}(A_c, B, C) = \left[ \max_{\omega \in \mathbb{R}} \mu_R[C(j\omega I - A_c)^{-1}B] \right]^{-1}, \quad (4.17)$$

kde

$$\begin{aligned} \mu_R[C(j\omega I - A_c)^{-1}B] &= \\ &= \inf_{\gamma \in (0,1)} \sigma_2 \left( \begin{bmatrix} \operatorname{Re} C(j\omega I - A_c)^{-1}B & -\gamma \operatorname{Im} C(j\omega I - A_c)^{-1}B \\ \frac{1}{\gamma} \operatorname{Im} C(j\omega I - A_c)^{-1}B & \operatorname{Re} C(j\omega I - A_c)^{-1}B \end{bmatrix} \right). \end{aligned} \quad (4.18)$$

Díky symetrii můžeme uvažovat jen  $\omega \in \langle 0, \infty \rangle$ . Při výpočtu se však musíme omezit na nějaký konečný interval  $\langle 0, \omega_{\max} \rangle$ . V [32] bylo dokázáno, že  $\omega_{\max}$  lze volit (analogicky jako u komplexního poloměru stability ve větě 4.3) za  $\sigma_1(A_c) + \frac{\sigma_1(C)\sigma_1(B)}{\sigma_1(CA_c^{-1}B)}$ .

**Věta 4.10.** ([32])

$$r_{\mathbb{R}}(A_c, B, C) = \left[ \max_{\omega \in \langle 0, \omega_{\max} \rangle} \mu_R[C(j\omega I - A_c)^{-1}B] \right]^{-1}, \quad (4.19)$$

kde  $\omega_{\max} = \sigma_1(A_c) + \frac{\sigma_1(C)\sigma_1(B)}{\sigma_1(CA_c^{-1}B)}$ .

Nyní bychom již mohli uvažovat naivní algoritmus pro výpočet  $r_{\mathbb{R}}(A_c, B, C)$ .

**Algoritmus 4.4. Výpočet reálného poloměru – naivní přístup**

Vstup:  $A_c, (B, C)$

- rozděl interval  $\langle 0, \omega_{\max} \rangle$  na podintervaly  $\langle \omega_i, \omega_{i+1} \rangle$
- na každém intervalu  $\langle \omega_i, \omega_{i+1} \rangle$  nalezni maximum  $\xi_i$  z hodnot  $\mu_R[C(j\omega I - A_c)^{-1}B]$  (např. metodou zlatého řezu, (algoritmus 2.4))
- vyber  $\xi^* = \max_i \xi_i^*$

Výstup:  $r_{\mathbb{R}} = (\xi^*)^{-1}$

Označme  $G(j\omega) = C(j\omega I - A_c)^{-1}B$ , potom vztah (4.18) přejde na

$$\mu_R[G(j\omega)] = \inf_{\gamma \in (0,1)} \sigma_2 \left( \begin{bmatrix} \operatorname{Re} G(j\omega) & -\gamma \operatorname{Im} G(j\omega) \\ \frac{1}{\gamma} \operatorname{Im} G(j\omega) & \operatorname{Re} G(j\omega) \end{bmatrix} \right).$$

Přestože  $\mu_R[G(j\omega)]$  můžeme spočítat pro unimodální funkci pomocí metody zlatého řezu, stále je obtížné nalézt  $\sup_{\omega \in \langle 0, \omega_{\max} \rangle} \mu_R[G(j\omega)]$ .

V [26] se k výpočtu (4.17) využívá, stejně jako v předchozím odstavci, vztah mezi singulárními čísly a vlastními čísly odpovídající Hamiltonovy matice.

Označme

$$\tilde{P}(\gamma, G(j\omega)) \triangleq \begin{bmatrix} \operatorname{Re} G(j\omega) & -\gamma \operatorname{Im} G(j\omega) \\ \frac{1}{\gamma} \operatorname{Im} G(j\omega) & \operatorname{Re} G(j\omega) \end{bmatrix}.$$

**Lemma 4.1.**

$$\tilde{P}(\gamma, G(j\omega)) \sim \begin{bmatrix} \operatorname{Re} G(j\omega) & j\gamma \operatorname{Im} G(j\omega) \\ j\frac{1}{\gamma} \operatorname{Im} G(j\omega) & \operatorname{Re} G(j\omega) \end{bmatrix}.$$

**Důkaz.**

$$\begin{aligned} \begin{bmatrix} I & 0 \\ 0 & jI \end{bmatrix} \begin{bmatrix} \operatorname{Re} G(j\omega) & -\gamma \operatorname{Im} G(j\omega) \\ \frac{1}{\gamma} \operatorname{Im} G(j\omega) & \operatorname{Re} G(j\omega) \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & jI \end{bmatrix} &= \\ &= \begin{bmatrix} \operatorname{Re} G(j\omega) & j\gamma \operatorname{Im} G(j\omega) \\ j\frac{1}{\gamma} \operatorname{Im} G(j\omega) & \operatorname{Re} G(j\omega) \end{bmatrix}. \end{aligned}$$

Protože  $\begin{bmatrix} I & 0 \\ 0 & jI \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & -jI \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$ , jsou příslušné matice podle definice 2.6 podobné.  $\square$

$$\text{Necht' nyní } P(\gamma, G(j\omega)) = \begin{bmatrix} \operatorname{Re} G(j\omega) & j\gamma \operatorname{Im} G(j\omega) \\ j\frac{1}{\gamma} \operatorname{Im} G(j\omega) & \operatorname{Re} G(j\omega) \end{bmatrix}.$$

**Lemma 4.2.**

$$P(\gamma, G(j\omega)) = T_\gamma \begin{bmatrix} G & 0 \\ 0 & \bar{G} \end{bmatrix} T_\gamma,$$

$$\text{kde } T_\gamma = \begin{bmatrix} I & 0 \\ 0 & \gamma^{-1}I \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} I & I \\ I & -I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \gamma I \end{bmatrix}.$$

**Důkaz.**

$$\begin{aligned} & \begin{bmatrix} I & 0 \\ 0 & \gamma^{-1}I \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} I & I \\ I & -I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \gamma I \end{bmatrix} \begin{bmatrix} G & 0 \\ 0 & \bar{G} \end{bmatrix} \\ & \begin{bmatrix} I & 0 \\ 0 & \gamma^{-1}I \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} I & I \\ I & -I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \gamma I \end{bmatrix} = \\ & = \begin{bmatrix} I & 0 \\ 0 & \gamma^{-1}I \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} I & I \\ I & -I \end{bmatrix} \begin{bmatrix} G & 0 \\ 0 & \bar{G} \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} I & I \\ I & -I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \gamma I \end{bmatrix} = \\ & = \begin{bmatrix} I & 0 \\ 0 & \gamma^{-1}I \end{bmatrix} \frac{1}{2} \begin{bmatrix} G + \bar{G} & G - \bar{G} \\ G - \bar{G} & G + \bar{G} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \gamma I \end{bmatrix} = \\ & = \begin{bmatrix} I & 0 \\ 0 & \gamma^{-1}I \end{bmatrix} \begin{bmatrix} \operatorname{Re}(G) & \operatorname{Im}(G) \\ \operatorname{Im}(G) & \operatorname{Re}(G) \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \gamma I \end{bmatrix} = \begin{bmatrix} \operatorname{Re} G(j\omega) & j\gamma \operatorname{Im} G(j\omega) \\ j\frac{1}{\gamma} \operatorname{Im} G(j\omega) & \operatorname{Re} G(j\omega) \end{bmatrix}. \end{aligned}$$

$\square$

Uvedená lemmata nám pomohou při důkazu následující věty.

**Věta 4.11.** ([26]) *Necht'  $\gamma \in (0, 1)$  a  $\xi > 0$ . Potom pro  $\omega \in \mathbb{R}$ , je  $\xi$  singulární číslo matice  $P(\gamma, G(j\omega))$  právě, když  $j\omega$  je vlastním číslem matice*

$$H(\xi, \tilde{A}_c, \tilde{B}_\gamma, \tilde{C}_\gamma) = \begin{bmatrix} \tilde{A}_c & \frac{1}{\xi} \tilde{B}_\gamma \tilde{B}_\gamma^T \\ -\frac{1}{\xi} \tilde{C}_\gamma^T \tilde{C}_\gamma & -\tilde{A}_c^T \end{bmatrix},$$

$$\text{kde } \tilde{A}_c = \begin{bmatrix} A_c & 0 \\ 0 & -A_c \end{bmatrix}, \tilde{B}_\gamma = \frac{1}{\sqrt{2}} \begin{bmatrix} B & \gamma B \\ -\gamma^{-1}B & B \end{bmatrix}, \tilde{C}_\gamma = \frac{1}{\sqrt{2}} \begin{bmatrix} C & \gamma C \\ \gamma^{-1}C & -C \end{bmatrix}.$$

**Důkaz.** Matice  $H(\xi, \tilde{A}_c, \tilde{B}_\gamma, \tilde{C}_\gamma)$  je Hamiltonova matice sestavená z matic  $\tilde{A}_c, \tilde{B}_\gamma, \tilde{C}_\gamma$ . S využitím věty 2.16 stačí ukázat, že  $P(\gamma, G(j\omega)) = \tilde{C}_\gamma(j\omega I - \tilde{A}_c)^{-1}\tilde{B}_\gamma$ .

$$\begin{aligned}
P(\gamma, G(j\omega)) &= T_\gamma \begin{bmatrix} G & 0 \\ 0 & \bar{G} \end{bmatrix} T_\gamma = \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} I & \gamma I \\ \gamma^{-1}I & -I \end{bmatrix} \begin{bmatrix} C(j\omega I - A_c)^{-1}B & 0 \\ 0 & C(-j\omega I - A_c)^{-1}B \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} I & \gamma I \\ \gamma^{-1}I & -I \end{bmatrix} = \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} I & \gamma I \\ \gamma^{-1}I & -I \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} (j\omega I - A_c)^{-1} & 0 \\ 0 & (j\omega I + A_c)^{-1} \end{bmatrix} \\
&\quad \begin{bmatrix} B & 0 \\ 0 & -B \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} I & \gamma I \\ \gamma^{-1}I & -I \end{bmatrix} = \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} C & \gamma C \\ \gamma^{-1}C & -C \end{bmatrix} \begin{bmatrix} j\omega I - A_c & 0 \\ 0 & j\omega I + A_c \end{bmatrix}^{-1} \frac{1}{\sqrt{2}} \begin{bmatrix} B & \gamma B \\ -\gamma^{-1}B & B \end{bmatrix} = \\
&\quad = \tilde{C}_\gamma(j\omega I - \tilde{A}_c)^{-1}\tilde{B}_\gamma.
\end{aligned}$$

□

Podle věty 4.11 můžeme postupovat analogicky jako u komplexního poloměru stability, jen s tím rozdílem, že zde musíme navíc zohlednit parametr  $\gamma$  a zajímá nás druhé největší singulární číslo.

Uvažujme  $k$ -tou iteraci a označme  $\xi_k = \inf_{\gamma \in (0,1)} \sigma_2[P(\gamma, G(j\omega_k))]$ , tj.  $\xi_k = \sigma_2[P(\gamma_k^*, G(j\omega_k))]$ .

**Lemma 4.3.** *Necht' pro pevné  $\hat{\omega} \in \mathbb{R}_0^*$  je  $\xi = \sigma_2[P(\hat{\gamma}^*, G(j\hat{\omega}))]$ , potom pro každé  $\omega \in \langle 0, \infty \rangle$  je  $\sigma_2[P(\hat{\gamma}^*, G(j\omega))] \geq \inf_{\gamma \in (0,1)} \sigma_2[P(\gamma, G(j\omega))]$ .*

**Důkaz.** Plyne přímo ze vztahu (4.18) a vlastnosti infima. □

Pro dané  $\omega_k$  spočteme  $\xi_k$  a  $\gamma_k^*$ . Vhodné intervaly pro další iteraci můžeme opět určit výpočtem ryze imaginárních vlastních čísel matice  $H(\xi_k, \tilde{A}_c, \tilde{B}_\gamma, \tilde{C}_\gamma)$  a určením těch  $\tilde{\omega}$ , která přísluší druhému největšímu singulárnímu číslu, například pomocí znaménka derivace odpovídající křivky v bodě  $\tilde{\omega}$ .

**Algoritmus 4.5. Výpočet reálného poloměru – Sreedharova metoda [26]**

Vstup:  $A_c, (B, C), \omega_0, \xi_0, \Omega_0 = \langle 0, \omega_{\max} \rangle$ , tol (tolerance),  $k = 0$

- pro  $\omega_{k+1}$  spočti  $\xi_{k+1} = \inf_{\gamma \in (0,1)} \sigma_2[P(\gamma, G(j\omega_{k+1}))]$  a nalezní  $\gamma_{k+1}^*$
- je-li  $\xi_{k+1} > \xi_k$  vyber  $\xi_{k+1}$ , jinak  $\xi_{k+1} = \xi_k$
- vypočti ryze imaginární vlastní čísla  $j\omega$  matice  $H(\xi_{k+1}, \tilde{A}_c, \tilde{B}_{\gamma_{k+1}^*}, \tilde{C}_{\gamma_{k+1}^*})$ , pokud žádná nejsou, pokračuj na j)
- pro každé  $\omega$  z předchozího bodu urči  $s_i$  a  $c_i$  podle (4.14)
- vyber  $\omega_i$ , pro něž je  $c_i = 2$ , pokud žádné není, jdi na j)

- f) urči  $\hat{\Omega}_{k+1} = (\omega_r, \omega_{r-1}) \cup (\omega_{r-2}, \omega_{r-3}) \cup \dots$ , pokud je délka největšího intervalu menší než  $\text{tol}$ , pokračuj na j)
- g) nastav  $\Omega_{k+1} = \hat{\Omega}_{k+1} \cup \Omega_k$
- h) vyber  $\omega_{k+1}$  jako střed největšího intervalu v  $\Omega_{k+1}$
- i)  $k = k + 1$
- j)  $\xi_{k+1} = \xi_k$ , konec

Výstup:  $r_{\mathbb{R}} = \xi_{k+1}^{-1}$

**Příklad 4.3.** Pro matice

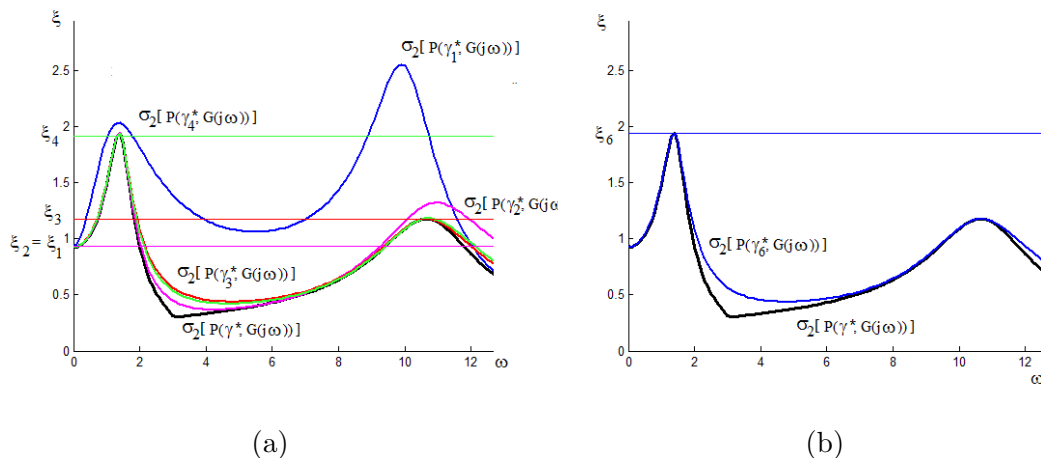
$$A = \begin{bmatrix} 79.0000 & 20.0000 & -30.0000 & -20.0000 \\ -41.0000 & -12.0000 & 17.0000 & 13.0000 \\ 167.0000 & 40.0000 & -60.0000 & -38.0000 \\ 33.5000 & 9.0000 & -14.5000 & -11.0000 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.2190 & 0.9347 \\ 0.0470 & 0.3835 \\ 0.6789 & 0.5194 \\ 0.6793 & 0.8310 \end{bmatrix}, \quad C = \begin{bmatrix} 0.0346 & 0.5297 & 0.0077 & 0.0668 \\ 0.0535 & 0.6711 & 0.3834 & 0.4175 \end{bmatrix}$$

a startovací interval  $S_0 = \langle 0, 216.8366 \rangle$  znázorňuje obrázek 4.6a princip metody pro výpočet reálného strukturovaného poloměru stability, kde  $\xi_k$  označuje odhad hodnoty reálného poloměru v  $k$ -té iteraci. Křivka  $\sigma_2[P(\gamma^*, G(j\omega))]$  představuje pro optimální  $\gamma^*$  křivku, pro níž se nabývá přibližně v bodě  $\omega = 1.37675$  optima. Křivky  $\sigma_2[P(\gamma_k^*, G(j\omega))]$  jsou horními aproximacemi  $\sigma_2[P(\gamma^*, G(j\omega))]$  v  $k$ -tých iteracích. Odtud je rovněž vidět, že  $\Omega_{k+1} \subset \Omega_k$ . Na obrázku 4.6b je znázorněna křivka  $\sigma_2[P(\gamma_6^*, G(j\omega))]$  v poslední iteraci. Hodnota reálného strukturovaného poloměru stability vychází  $r_{\mathbb{R}} = 0.514144$ . Tabulka 4.1 ukazuje, jak se mění hodnoty v jednotlivých iteracích.

Iterace	$\omega_k$	$\gamma_k^*$	$\xi_k$
1	0	0.999999	0.931699
2	0	0.136582	0.931699
3	10.65244	0.227994	1.177244
4	1.310585	0.205987	1.919030
5	1.375544	0.227033	1.944970
6	1.376751	0.227043	1.944979

Tabulka 4.1: Hodnoty v jednotlivých iteracích při výpočtu reálného strukturovaného poloměru.



Obrázek 4.6: Sreedharova metoda, (a) průběh druhých singulárních čísel v závislosti na  $\omega$ , (b) druhé singulární číslo v poslední iteraci.

## 4.3 Pseudospektrum

Spektrum matice  $A_c$  tvoří podle definice 2.5 body  $s \in \mathbb{C}$ , pro něž je  $\det(sI - A_c) = 0$ , neboli body, ve kterých není definován výraz  $(sI - A_c)^{-1}$ , (lze jej dodefinovat jako  $\infty$ ). Pseudospektrum matice  $A_c$  budou pak tvořit body, v nichž je číslo  $\|(sI - A_c)^{-1}\|$  konečné, ale řekněme velmi velké. Zaměříme-li se opět na perturbace v prvcích matice a povolíme perturbace omezené v normě, budou do pseudospektra matice  $A_c$  příslušet vlastní čísla té perturbované matice.

### 4.3.1 Komplexní pseudospektrum

**Definice 4.5.** Necht'  $A_c \in \mathbb{R}^{n \times n}$ ,  $\varepsilon > 0$ , potom

$$\Lambda_\varepsilon^{\mathbb{C}}(A_c) = \{s \in \mathbb{C} : s \in \Lambda(A_c + \Delta) \text{ pro } \Delta \in \mathbb{C}^{n \times n}, \|\Delta\|_2 < \varepsilon\} \quad (4.20)$$

je  $\varepsilon$ -pseudospektrum matice  $A_c$ .

Opět můžeme uvažovat perturbace v prvcích regulátoru, což zobecněním (4.20) vede k definici strukturovaného pseudospektra.

**Definice 4.6.** Necht'  $A_c \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$ ,  $\varepsilon > 0$ , potom

$$\Lambda_\varepsilon^{\mathbb{C}}(A_c, B, C) = \{s \in \mathbb{C} : s \in \Lambda(A_c + B\Delta C) \text{ pro } \Delta \in \mathbb{C}^{m \times p}, \|\Delta\|_2 < \varepsilon\} \quad (4.21)$$

je strukturované  $\varepsilon$ -pseudospektrum systému  $(A_c, B, C)$ .

**Věta 4.12.**

$$\Lambda_\varepsilon^{\mathbb{C}}(A_c, B, C) = \left\{ s \in \mathbb{C} : \sigma_1(C(sI - A_c)^{-1}B) > \frac{1}{\varepsilon} \right\}. \quad (4.22)$$

**Důkaz.** Podle definice 4.6 je  $s \in \Lambda_\varepsilon^{\mathbb{C}}(A_c, B, C)$ , když  $s \in \Lambda(A_c + B\Delta C)$ , kde  $\|\Delta\|_2 < \varepsilon$ . To znamená, že  $\det(sI - (A_c + B\Delta C)) = 0$ . Podle (4.5) je  $\det(I - \Delta C(sI - A_c)^{-1}B) = 0$ , tedy  $1 \in \Lambda(\Delta C(sI - A_c)^{-1}B)$  a  $[\Delta C(sI - A_c)^{-1}B]v = v$ , pro  $v \in \mathbb{C}^n$ , kde  $\|v\|_2 = 1$ . S využitím těchto vztahů získáme dohromady následující

$$\begin{aligned} 1 = \|v\| &= \|\Delta C(sI - A_c)^{-1}B\| \leq \|\Delta\| \|C(sI - A_c)^{-1}B\| \\ \frac{1}{\|\Delta\|} &\leq \|C(sI - A_c)^{-1}B\| \\ &\text{a protože } \|\Delta\| < \varepsilon \\ \frac{1}{\varepsilon} &< \frac{1}{\|\Delta\|} \leq \|\Delta C(sI - A_c)^{-1}B\|, \\ \frac{1}{\varepsilon} &< \sigma_1[\Delta C(sI - A_c)^{-1}B]. \end{aligned}$$

□

**Poznámka 4.5.** Pro případ  $B = I_n, C = I_n$  lze (4.22) vyjádřit jako

$$\Lambda_\varepsilon^{\mathbb{C}}(A_c) = \{s \in \mathbb{C} : \sigma_{\min}(sI - A_c) \leq \varepsilon\}. \quad (4.23)$$

Při výpočtu hodnot pro znázornění pseudospektra matice  $A_c$  můžeme vycházet ze vztahu (4.22) nebo (4.23). V komplexní rovině si vygenerujeme síť bodů ( $\nu \times \nu$ ) a v každém tomto bodě bychom potřebovali spočítat hledané singulární číslo. Výpočet klasického pseudospektra bychom mohli urychlit, aplikujeme-li na matici  $A_c$  Schurův rozklad a nahradíme ji unitárně ekvivalentní horní (kvazi-)trojúhelníkovou maticí. Hodnoty singulárních čísel budou stejné, ale počet potřebných operací se sníží. U výpočtu strukturovaného pseudospektra však nelze využít stejného zjednodušení jako u klasického a musíme se zde zatím spokojit s přímou metodou výpočtu hodnoty ze vztahu (4.22) v každém bodě mřížky a následném vykreslení vrstevnic.

#### Algoritmus 4.6. Komplexní pseudospektrum

Vstup:  $A_c \in \mathbb{R}^{n \times n}$ ,  $(B, C)$ , (meze  $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ ),  
(pokud  $B = C = I$ , nahrad'  $A_c$  horní (kvazi-) trojúhelníkovou maticí  $T$  z Schurova rozkladu (dle věty 2.14))

- vygeneruj mřížku  $\nu \times \nu$
- spočti  $\sigma_1[C(sI - A_c)^{-1}B]$  pro každý bod  $s$  mřížky  $\nu \times \nu$
- vykresli vrstevnice pro  $\varepsilon$

Výstup: komplexní pseudospektrum  $\Lambda_\varepsilon^{\mathbb{C}}$



### 4.3.2 Reálné pseudospektrum

Pro reálné perturbace získáme reálné pseudospektrum.

**Definice 4.7.** Necht'  $A_c \in \mathbb{R}^{n \times n}$ ,  $\varepsilon > 0$ , potom

$$\Lambda_\varepsilon^{\mathbb{R}}(A_c) = \{s \in \mathbb{C} : s \in \Lambda(A_c + \Delta) \text{ pro } \Delta \in \mathbb{R}^{n \times n}, \|\Delta\|_2 < \varepsilon\} \quad (4.24)$$

je  $\varepsilon$ -pseudospektrum matice  $A_c$ .

Při perturbacích v prvcích regulátoru zavedeme strukturované reálné pseudospektrum.

**Definice 4.8.** Necht'  $A_c \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$ ,  $\varepsilon > 0$ , potom

$$\Lambda_\varepsilon^{\mathbb{R}}(A_c, B, C) = \{s \in \mathbb{C} : s \in \Lambda(A_c + B\Delta C) \text{ pro } \Delta \in \mathbb{R}^{m \times p}, \|\Delta\|_2 < \varepsilon\} \quad (4.25)$$

je strukturované reálné  $\varepsilon$ -pseudospektrum systému  $(A_c, B, C)$ .

Podle výpočetního vztahu (4.8) můžeme počítat i reálné pseudospektrum.

**Věta 4.13.** ([14])

$$\Lambda_\varepsilon^{\mathbb{R}}(A_c, B, C) = \left\{ s \in \mathbb{C} : \mu_{\mathbb{R}}[C(sI - A_c)^{-1}B] > \frac{1}{\varepsilon} \right\}. \quad (4.26)$$

#### Algoritmus 4.7. Reálné pseudospektrum

Vstup:  $A_c \in \mathbb{R}^{n \times n}$ ,  $(B, C)$ , (meze  $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ )

(pokud  $B = C = I_n$ , nahrad'  $A_c$  horní (kvazi-)trojúhelníkovou maticí  $T$  z Schurova rozkladu dle věty 2.14)

- vygeneruj mřížku  $\nu \times \nu$
- spočti  $\mu_{\mathbb{R}}[C(sI - A_c)^{-1}B]$  pro každý bod  $s$  mřížky  $\nu \times \nu$
- vykresli vrstevnice pro  $\varepsilon$

Výstup: reálné pseudospektrum  $\Lambda_\varepsilon^{\mathbb{R}}$

## 4.4 $H_\infty$ norma

Uvažujme model systému

$$\begin{aligned} \dot{x} &= Ax + B_2w + B_1u \\ z &= C_2x \\ y &= C_1x, \end{aligned} \tag{4.27}$$

kde  $A, x$  mají stejný význam jako u (3.1), vektor  $w \in \mathbb{R}^m$  představuje poruchy nebo testovací signály působící na systém,  $u \in \mathbb{R}^m$  je vstup, řízení systému, vektor  $z \in \mathbb{R}^p$  řízená proměnná (vyjadřuje kvalitu toho, čeho chceme dosáhnout) a  $y \in \mathbb{R}^p$  je výstupní proměnná.

Pro matici uzavřeného systému  $A_c = A + B_1KC_1$  zavádíme přenos systému z  $w$  na  $z$  jako  $G(s) = C_2(sI - A_c)^{-1}B_2$ . Zobrazení z prostoru matic přenosu  $G(s)$  do  $\mathbb{R}^+$  definované

$$\|G(s)\|_{\mathcal{H}_\infty} = \sup_{s \in \partial\mathcal{C}_g} \sigma_1(G(s)) \tag{4.28}$$

nazýváme  $H_\infty$  normou. Někdy se norma přenosu značí  $\|G\|_{\mathcal{H}_\infty}$ , kde  $\mathcal{H}_\infty$  je Hardyho prostor [20].

My se budeme opět soustředit na případ, že  $\partial\mathcal{C}_g$  je imaginární osa, potom bude

$$\|G(j\omega)\|_{\mathcal{H}_\infty} = \sup_{\omega \in \mathbb{R}} \sigma_1(G(j\omega)). \tag{4.29}$$

Norma  $H_\infty$  udává, jaký vliv mají poruchy  $w$  na řízenou veličinu  $z$ . Protože by tento vliv měl být co nejmenší, chceme dosáhnout minimální  $H_\infty$  normy.

Podle vztahu (4.29) je patrné, že výpočet lze provádět stejnými metodami 4.1, 4.2, 4.3 jako u strukturovaného komplexního poloměru stability, pro vhodně zvolené matice  $(A_c, B, C)$ , konkrétně  $(A_c, B_2, C_2)$ .

Pro systémy s jedním vstupem a výstupem přejde tvar (4.29) na

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

V amplitudové frekvenční charakteristice, která vyjadřuje závislost  $|G(j\omega)|$  na frekvenci  $\omega$ , tomu odpovídá bod s největší amplitudou  $G_{\max}$  pro frekvenci  $\omega_{\max}$ , tedy  $\|G\|_{\mathcal{H}_\infty} = G_{\max}$ .

**Příklad 4.4.** Mějme systém  $(A, B_1, C_1, B_2, C_2)$  s jedním vstupem a jedním výstupem daný maticemi

$$A = \begin{bmatrix} -2 & 0 & 1 \\ 2 & -2 & -3 \\ 3 & 1 & 1 \end{bmatrix}, B_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, C_1 = [1 \ 0 \ 1],$$

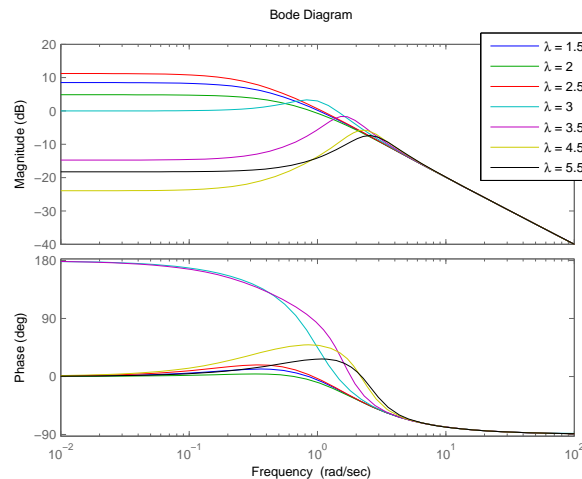
$$B_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, C_2 = [0 \ 1 \ 0].$$

Pro tento systém neobdržíme žádné parametry v matici  $Q(\alpha)$ . Volný parametr budeme tedy uvažovat v Jordanově bloku  $L$ . Chceme najít výstupní zpětnou vazbu, která přiřadí uzavřenému systému  $A + B_1 K C_1$  vlastní číslo  $-\lambda$ .

Výstupní zpětná vazba bude mít tvar

$$K = \left[ -\frac{\lambda^3 - 3\lambda^2 + 6}{\lambda^2 - 4\lambda + 9} \right].$$

Na obrázku 4.7 je znázorněn průběh frekvenční charakteristiky  $G(j\omega)$  pro různé hodnoty přiřazovaného vlastního čísla. Podle tohoto obrázku bychom si vybrali přiřazení vlastního čísla  $\lambda = 5.5$ , pro které bychom dostali ze všech uvedených možností minimální hodnotu (4.29).



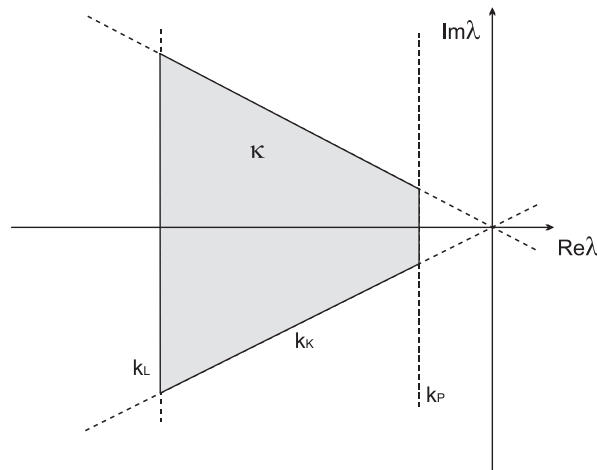
Obrázek 4.7: Bodeho logaritmičká amplitudová a fázová frekvenční charakteristika.

## 4.5 Ostatní uvažované funkce

### 4.5.1 Lokace vlastních čísel

V případě neúplného přiřazení ( $s < n$ ) přiřazujeme jen některá vlastní čísla v Jordanově blocích. Ostatní vlastní čísla budou potom záviset na parametrech. Protože chceme původní systém stabilizovat, požadujeme, aby i pro ostatní vlastní čísla platilo  $\text{Re}(\lambda) < 0$ .

Rovněž můžeme chtít, aby všechna vlastní čísla ležela nalevo od vhodné přímky, tj.  $\text{Re}(\lambda) \leq k_P$ , nebo aby  $\text{Re}(\lambda) \geq k_L$  a systém tak nebyl příliš rychlý. Také můžeme chtít potlačit kmitavost systému a umístit vlastní čísla tak, aby  $\frac{\text{Im}(\lambda)}{\text{Re}(\lambda)} \leq k_K$ . Podle těchto omezení požadujeme, aby se ostatní vlastní čísla, která jsme nepřidali, nacházela v nějaké uzavřené oblasti  $\kappa$  zobrazené na obrázku 4.8.



Obrázek 4.8: Umístění vlastních čísel.

Zaved'me si funkce

$$\begin{aligned} f_P &= \max_i \{\text{Re}(\lambda_i)\}, \\ f_L &= \min_i \{\text{Re}(\lambda_i)\}, \\ f_K &= \max_i \left\{ \frac{\text{Im}(\lambda_i)}{\text{Re}(\lambda_i)} \right\}. \end{aligned}$$

K předchozím funkcím tak lze přidat následující optimalizační (účelovou) funkci

$$f_l = p_K \max\{0, f_K - k_K\} + p_P \max\{0, f_P - k_P\} + p_L \max\{0, k_L - f_L\}, \quad (4.30)$$

kde hodnoty  $p_K, p_P, p_L$  představují penalizace.

Pokud budou vlastní čísla ležet v požadované oblasti, bude hodnota funkce  $f_l$  nulová. V případě, že se vlastní čísla budou nacházet vně  $\kappa$ , bude se hodnota funkce  $f_l$  zvětšovat s rostoucí vzdáleností od hranice oblasti.

## 4.5.2 Regulátor s omezenou strukturou

V teorii řízení se používá také regulátor, který přiřazuje danou Jordanovu formu a má zvolenou strukturu nenulových prvků. Prvky matice zpětné vazby jsou racionálními funkcemi v návrhových parametrech  $\alpha$ . Od některých prvků tak můžeme požadovat, aby byly nulové.

Potom můžeme minimalizovat funkci

$$\sum_{[i,j] \in \mathcal{N}_F} p_{k_{i,j}} |f_{i,j}|,$$

kde  $f_{i,j}$  je prvek matice zpětné vazby na pozici  $i, j$ , množina  $\mathcal{N}_F$  obsahuje vybrané prvky k vynulování a  $p_{k_{i,j}}$  odpovídající penalizace. Ta by měla být volena velmi velká. Následná kontrola, zda je vypočtená zpětná vazba správná, může být ověřena nastavením zvolených prvků napevno na nulu a spočtením vlastních čísel matice uzavřeného systému. Jestli budou blízko přiřazovaným vlastním číslům, získali jsme zpětnou vazbu s omezenou strukturou.

## 5 Návrh robustního přiřazení Jordanovy formy

Pro daný systém budeme navrhovat stavovou či výstupní zpětnou vazbu tak, abychom optimalizovali vlastnosti matice dynamiky uzavřeného systému, nebo vlastnosti příslušného regulátoru. Optimalizaci budeme provádět vzhledem k parametrům  $\alpha$  parametrické matice  $Q(\alpha)$  z věty 3.3. Pokud navíc nebudeme požadovat přiřazení Jordanovy formy s pevnými (konkrétními) vlastními čísly, ale jen její strukturu s volnými prvky, získáme další parametry (označované  $q$ ), které lze také optimalizovat. Další volné parametry (označované  $r$ ) obdržíme v případě neúplného přiřazení (věta 3.3).

Podle toho, zda budeme požadovat, aby výsledný systém byl co nejvíce robustní ve stabilitě, nebo regulátor co nejméně fragilní, budeme řešit následující problémy.

**Problém 5.1.** Robustnost uzavřeného systému

Pro daný systém  $(A, B)$ , resp.  $(A, B, C)$ , nalezněte stavovou zpětnou vazbu  $F \in \mathbb{R}^{m \times n}$ , resp. výstupní zpětnou vazbu  $K \in \mathbb{R}^{m \times p}$ , takovou, aby výsledný uzavřený systém byl stabilní a aby

- řízený systém měl maximální komplexní poloměr stability anebo
- řízený systém měl maximální reálný poloměr stability anebo
- řízený systém měl minimální  $H_\infty$  normu (v případě, že jsou zadány matice  $B_2, C_2$  jako v (4.27)) anebo
- zvolené prvky matice zpětné vazby byly nulové anebo
- vlastní čísla matice dynamiky uzavřeného systému ležela v předepsané oblasti (pokud máme volné prvky v blocích Jordanovy formy nebo uvažujeme neúplné přiřazení).

**Problém 5.2.** Fragilita zpětné vazby

Pro daný systém  $(A, B)$ , resp.  $(A, B, C)$ , nalezněte stavovou zpětnou vazbu  $F \in \mathbb{R}^{m \times n}$ , resp. výstupní zpětnou vazbu  $K \in \mathbb{R}^{m \times p}$ , takovou, aby výsledný uzavřený systém byl stabilní a aby

- řízený systém měl maximální strukturovaný komplexní poloměr stability anebo
- řízený systém měl maximální strukturovaný reálný poloměr stability anebo

- řízený systém měl minimální  $H_\infty$  normu (v případě, že jsou zadány matice  $B_2, C_2$  jako v (4.27)) anebo
- zvolené prvky matice zpětné vazby byly nulové anebo
- vlastní čísla matice dynamiky uzavřeného systému ležela v předepsané oblasti (pokud máme volné prvky v blocích Jordanovy formy anebo uvažujeme neúplné přiřazení).

**Poznámka 5.1.** V případě, že maximalizujeme (strukturovaný) komplexní poloměr stability, řešíme úlohu

$$\begin{aligned} \max_{[\alpha, q, r]} r_{\mathbb{C}}(A, B, C) &= \max_{[\alpha, q, r]} \frac{1}{\max_{\omega \in \langle 0, \omega_{\max} \rangle} \sigma_1[C(j\omega I - A_c)^{-1}B]} = \\ &= \min_{[\alpha, q, r]} \max_{\omega \in \langle 0, \omega_{\max} \rangle} \sigma_1[C(j\omega I - A_c)^{-1}B]. \end{aligned}$$

**Poznámka 5.2.** V případě, že maximalizujeme (strukturovaný) reálný poloměr stability, řešíme úlohu

$$\begin{aligned} \max_{[\alpha, q, r]} r_{\mathbb{R}}(A, B, C) &= \max_{[\alpha, q, r]} \frac{1}{\max_{\omega \in \langle 0, \omega_{\max} \rangle} \mu_R[C(j\omega I - A_c)^{-1}B]} = \\ &= \min_{[\alpha, q, r]} \max_{\omega \in \langle 0, \omega_{\max} \rangle} \mu_R[C(j\omega I - A_c)^{-1}B], \end{aligned}$$

kde  $\mu_R[C(j\omega I - A_c)^{-1}B]$  je určeno podle (4.18).

**Poznámka 5.3.** Ke každé funkci, kterou optimalizujeme, můžeme přidat vedlejší podmínky. Jako vedlejší podmínky v této práci uvažujeme, aby

- hodnota (strukturovaného) komplexního poloměru byla alespoň zadaná hodnota  $b_c$ , tj.  $r_{\mathbb{C}} \geq b_c$ ,
- hodnota (strukturovaného) reálného poloměru byla alespoň  $b_r$ , tj.  $r_{\mathbb{R}} \geq b_r$ ,
- hodnota  $H_\infty$  normy byla nejvýš  $b_h$ , tj.  $f_h \leq b_h$ .

Výsledná úloha může mít potom následující podobu.

$$\min_{[\alpha, q, r]} \max \{p_1 f_{r_{\mathbb{C}}}, p_2 f_{r_{\mathbb{R}}}, p_3 f_h, p_4 f_l, p_5 f_z\},$$

kde funkce  $f_{r_{\mathbb{C}}}$  je převrácená hodnota (strukturovaného) komplexního poloměru stability,  $f_{r_{\mathbb{R}}}$  je převrácená hodnota (strukturovaného) reálného poloměru stability,  $f_h$  je

$H_\infty$  norma systému  $(A_c, B_2, C_2)$ ,  $f_l$  funkce umísťující vlastní čísla do dané oblasti a  $f_z$  funkce s hodnotami prvků matice zpětné vazby, které chceme vynulovat, a  $p_i$ , pro  $i = 1, \dots, 5$  jsou příslušné penalizace.

Obdobně s omezením tak můžeme například uvažovat úlohu

$$\min_{[\alpha, q, r]} \max\{p_1 f_{r_C}, p_2 f_l\}, \quad \text{za podmínky } f_h \leq \frac{1}{2}.$$

### Algoritmus 5.1. Nalezení robustní zpětné vazby

Vstup: systém  $A, B, (C)$ , přiřazované Jordanovy bloky, optimalizované funkce  $f_i$  pro  $i = 1, \dots, v$ , penalizace  $p_i$  pro  $i = 1, \dots, v$ , omezení – funkce vedlejších podmínek a jim odpovídající mezní hodnota, počáteční vektor  $x_0$  ( $[\alpha_0, q_0, r_0]$ ),

- a) vypočti hodnoty funkcí  $f_i$  a omezení  $g_j^i$
- b) sestav  $\{p_1 f_1, \dots, p_v f_v\}$
- c) vypočti  $\min_x \max_{\{p_i f_i\}} \{p_i f_i\}$

Výstup:  $x_{\text{opt}}$  ( $[\alpha_{\text{opt}}, q_{\text{opt}}, r_{\text{opt}}]$ ), funkční hodnoty  $f_i(x_{\text{opt}})$ , zpětná vazba

**Poznámka 5.4.** Pokud budeme řešit úlohu

$$\min_{x \in \mathbb{R}^n} f(x) \tag{5.1a}$$

$$\text{s omezením, že } g_i(x) \leq 0, \quad \text{pro } i = 1, \dots, m, \tag{5.1b}$$

převédeme ji na úlohu

$$\min_{x \in \mathbb{R}^n} \left\{ f(x) + \sum_{i=1}^m \phi_i(\nu_i, g_i) \right\}, \tag{5.2}$$

kde pro  $i = 1, \dots, m$  představuje  $\phi_i(\nu_i, g_i)$  penalizační funkci pro funkci  $g_i(x)$  a  $\nu_i$  je vhodná pokuta. Funkce  $\phi_i(\nu_i, g_i)$  je volena tak, aby při nesplnění nerovnosti (5.1b) měla pokuta velký vliv a jinak funkce  $f(x)$  nebyla vůbec pokutována.

Pokud je vedlejší funkce komplexní (strukturovaný) poloměr stability  $r_C$ , tj.  $r_C \geq b_c$ , neboli  $f_{r_C} = \frac{1}{r_C} \leq \frac{1}{b_c}$ , volíme

$$\phi(\nu, r_C) = \nu \left( \max \left\{ 0, f_{r_C} - \frac{1}{b_c} \right\} \right)^4.$$

Obdobnou penalizaci volíme i v případě reálného (strukturovaného) poloměru stability  $r_{\mathbb{R}}$ , kdy pro  $r_{\mathbb{R}} \geq b_r$ , neboli pro  $f_{r_{\mathbb{R}}} = \frac{1}{r_{\mathbb{R}}} \leq \frac{1}{b_r}$ , volíme

$$\phi(\nu, r_{\mathbb{R}}) = \nu \left( \max \left\{ 0, f_{r_{\mathbb{R}}} - \frac{1}{b_r} \right\} \right)^4.$$



Necht' je vedlejší funkcí norma  $H_\infty$ , tj.  $f_h$  a požadujeme, aby  $f_h \leq b_h$ . Potom budeme uvažovat

$$\phi(\nu, f_h) = \max\{0, \nu e^{f_h - b_h} - \nu\}.$$

## 5.1 Přiřazení stavovou zpětnou vazbou

Při hledání stavové zpětné vazby takové, aby hodnoty kriteriální funkce pro uzavřený systém byly co možná nejlepší, můžeme využívat zejména větu 3.3 a vztah (3.10). Ty nám dávají explicitní parametrizaci množiny  $\mathcal{F}_s$  pomocí návrhového parametru  $\alpha$ . V každé iteraci si tak můžeme pro příslušný vektor vypočítat matici  $F$  i hodnoty zadaných funkcí.

### Algoritmus 5.2. Robustní přiřazení Jordanovy formy stavovou zpětnou vazbou

Vstup: systém  $A, B$ , přiřazované Jordanovy bloky, optimalizované funkce  $f_i$ , pro  $i = 1, \dots, v$ , penalizace  $p_i$ , pro  $i = 1, \dots, v$ , omezení – funkce vedlejších podmínek a jejich mezní hodnota, počáteční vektor  $x_0$  ( $[\alpha_0, q_0, r_0]$ ),

- a) vypočti hodnoty funkcí  $f_i$  a omezení  $g_j^i$  (vypočti  $F$  podle (3.10) a hodnoty funkcí z kapitoly 4.1)
- b)  $\min_x \max_{\{p_i, f_i\}} \{p_i f_i(x)\}$  (pomocí optimalizační metody BFGS (alg. 2.5))

Výstup:  $x_{\text{opt}}$  ( $[\alpha_{\text{opt}}, q_{\text{opt}}, r_{\text{opt}}]$ ), funkční hodnoty  $f_i(x_{\text{opt}})$ , stavová zpětná vazba

**Příklad 5.1.** Pro systém  $(A, B)$  a Jordanovu formu  $L$  z příkladu 3.3, kde jsme přiřazovali bloky

$$\begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix}, [-2], [-1],$$

navrháme stavovou zpětnou vazbu  $F \in \mathcal{F}_4(A, B, L)$  takovou, aby výsledný uzavřený systém byl stabilní a aby řízený systém měl maximální strukturovaný reálný poloměr stability.

Protože se jedná o neúplné přiřazení, musíme zde zajistit, aby všechna vlastní čísla uzavřeného systému ležela v otevřené levé polorovině komplexní roviny. Můžeme si navíc vybrat oblast (jako na obrázku 4.8), kam budeme chtít umístit zbylé neurčené vlastní číslo. V tomto případě budeme tedy minimalizovat jednak funkci  $f_l$  a jednak převrácenou hodnotu strukturovaného reálného poloměru stability  $f_r$ . Tím se snažíme získat co nejlepší regulátor.

Budeme řešit úlohu

$$\min_{[\alpha, r]} \max\{p_1 f_{r_{\mathbb{R}}(A_c, B)}([\alpha, r]), p_2 f_l([\alpha, r])\}, \quad (5.3)$$

kde  $A_c = A + BF$ ,  $F$  je maticí stavové zpětné vazby.

Zvolme si  $p_1 = 1, p_2 = 10^5$  a  $f_l = 10^5 \max\{0, f_P + 1\} + 10^5 \max\{0, -10 - f_L\}$ . a počáteční vektor  $[10, -8, 2, -4]$ . V průběhu optimalizace je nejdříve v několika prvních iteracích uzavřený systém nestabilní a strukturovaný reálný poloměr stability není definován, minimalizuje se tak hodnota funkce  $f_l$ . Poté, co se vlastní čísla dostanou do požadované oblasti, je funkce  $f_l$  nulová a minimalizuje se funkce  $f_{r_{\mathbb{R}}}$ . Takto postupně dospějeme k hodnotě  $r_{\mathbb{R}} = 1.0281$  a výsledné spektrum matice dynamiky uzavřeného systému bude  $\Lambda(A + BF_{\text{opt}}) = \{-2, -1, -1, -1, -9.9987\}$ . Neurčené vlastní číslo se tedy přiřadilo do bodu  $-9.9987$  a daný systém je stabilní. Matice zpětné vazby má tvar

$$F_{\text{opt}} = \begin{bmatrix} -2.2221 & 4.1685 & -0.1873 & -0.0494 & -0.5127 \\ -4.6940 & -4.1302 & -17.7996 & -20.7766 & -36.6222 \end{bmatrix}.$$

**Příklad 5.2.** Pro systém  $(A, B)$  navíjecího ústrojí [23] daný maticemi

$$A = \begin{bmatrix} -1 & 0 & -1 & 1 \\ 0 & -1 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix},$$

navrhněme stavovou zpětnou vazbu  $F \in \mathcal{F}_4(A, B, L)$  takovou, aby výsledný uzavřený systém byl stabilní a aby matice dynamiky uzavřeného systému měla maximální komplexní poloměr stability.

Pro stabilizaci systému  $(A, B)$  přiřazujme čtyřnásobný pól  $-q$ , pro  $q \in (0, \infty)$ . Jordanova forma bude mít následující podobu

$$\begin{bmatrix} -q & 1 & 0 & 0 \\ 0 & -q & 1 & 0 \\ 0 & 0 & -q & 1 \\ 0 & 0 & 0 & -q \end{bmatrix}. \quad (5.4)$$

Jedná se o úplné přiřazení a navíc zde máme volný parametr odpovídající vlastnímu číslu. Jednak chceme zajistit, aby všechna vlastní čísla uzavřeného systému ležela v otevřené levé polorovině komplexní roviny, jednak si můžeme určit oblast (jako na obrázku 4.8), odkud budeme volný parametr  $q$  uvažovat.

Chceme tedy minimalizovat funkci  $f_{r_{\mathbb{C}}}$  (převrácená hodnota komplexního poloměru stability) a rovněž funkci  $f_l$ , aby vlastní číslo náleželo oblasti  $\kappa$ .

Budeme řešit úlohu

$$\min_{[\alpha, q]} \max\{f_{r_{\mathbb{C}}(A_c)}([\alpha, q]), f_l([\alpha, q])\}. \quad (5.5)$$

Zvolme si  $f_l = 10 \max\{0, f_P + 1\} + 10 \max\{0, -30 - f_L\}$ ,  $f_P = -1$ ,  $f_L = -30$ .

Hodnota lokálního minima uvažované funkce i průběh optimalizace budou samozřejmě záležet na volbě počátečního vektoru  $x_0$ . Ukažme si v následující tabulce 5.1 výsledky pro různá  $x_0$ . Ve všech případech je hodnota funkce  $f_l$  nulová, pohybujeme

$x_0$	$r_{\mathbb{C}}(A_c)$	$x_{\text{opt}}$	Počet iterací
[-5, -3, 1, 2, 5]	0.6245	[-0.4012, -3.8235, 4.7934, -4.0401, 3.4930]	244
[3, 2, 1, 8, 5]	0.6244	[-0.4009, 3.3361, 5.1997, 9.6984, 3.4928]	363
[2, -1, 4, -3, 2]	0.5923	[66.4465, 214.4580, 401.4178, -420.4438, 3.4304]	304
[2, 6, -1, -3, 3.4]	0.6241	[-2.8709, 1.0830, 0.9015, 4.9269, 3.4337]	276
[4, 1, 3, 3, 9]	0.6255	[16.8448, 15.0064, 18.2195, 23.1024, 3.8749]	536

Tabulka 5.1: Výsledky optimalizace komplexního poloměru stability.

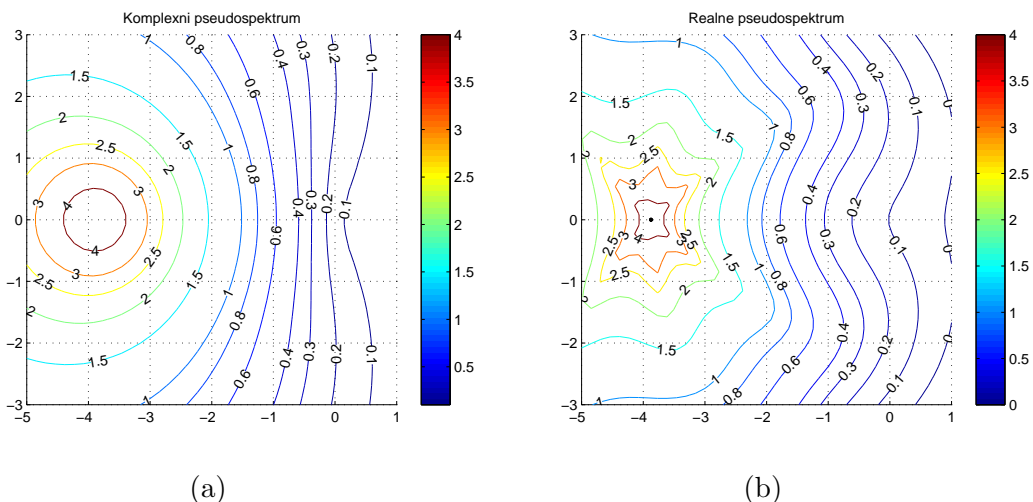
se ve stabilní oblasti a zároveň i ve zvolené oblasti  $\kappa$ . Z výsledků získáváme zajímavé pozorování. Hodnota  $q$ , pro kterou dostáváme lokální optima komplexního poloměru stability, se pohybuje okolo 3.8. Pro stabilizaci bychom si tedy vybrali vlastní číslo Jordanova bloku (5.4) přibližně  $-3.87$ . Pro úplnost uveďme některou výslednou stavovou zpětnou vazbu. Například pro poslední řádek tabulky 5.1 dostaneme

$$F_{\text{opt}} = \begin{bmatrix} -7.0056 & 7.4632 & -0.4718 & -15.4568 \\ -17.3237 & 8.7696 & -7.4940 & -58.0636 \end{bmatrix}.$$

Výsledky jsou sice pro některé počáteční vektory odlišné, ale to jsme očekávali, získáváme totiž lokální minima. Složitost problému je zde dána jeho větší dimenzí a také tím, že přiřazujeme jedno čtyřnásobné vlastní číslo. Na obrázku 5.1 jsou znázorněna pseudospektra matice dynamiky uzavřeného systému  $A_c = A + BF_{\text{opt}}$ , pro uvedenou stavovou zpětnou vazbu  $F_{\text{opt}}$ . Hodnota reálného poloměru stability pro tuto matici  $A_c$  je  $r_{\mathbb{R}}(A_c) = 0.6383$ .

## 5.2 Přiřazení výstupní zpětnou vazbou

Robustní přiřazení Jordanovy formy výstupní zpětnou vazbou je mnohem těžším problémem než přiřazení stavovou zpětnou vazbou. Při výpočtu výstupní zpětné vazby sestavujeme podle odstavce 3.2 matici stavové zpětné vazby a následně provádíme nulování prvků. Potom ale nevíme, jaká je efektivní dimenze návrhového



Obrázek 5.1: Komplexní a reálné pseudospektrum matice dynamiky uzavřeného systému  $A + BF$ .

vektoru  $\alpha$ , protože při řešení soustavy nelineárních rovnic (3.17) se některé parametry stanou pevně závislými na ostatních (některé zůstanou volné) a volnost ve vektoru  $\alpha$  se sníží.

Označme nyní  $F$  stavovou zpětnou vazbu, kterou dostaneme po transformaci (3.13) a  $\alpha$  návrhový vektor v matici  $Q(\alpha)$  z věty 3.3. Požadujeme, aby v matici  $F$  bylo posledních  $n - p$  sloupců nulových, tj. podle (3.17)  $F_2 = 0$ . Označme  $\mathcal{M} \subset \mathbb{R}^\gamma$  množinu všech vektorů  $\alpha$ , které jsou řešením rovnice  $F_2 = 0$ , tj.

$$\mathcal{M} = \{\alpha \in \mathbb{R}^\gamma : F_2(\alpha) = 0\}.$$

Matice výstupní zpětné vazby  $K$  potom tedy existuje jen pro  $\alpha \in \mathcal{M}$ . Hlavním problémem při numerickém výpočtu je, že o množině  $\mathcal{M}$  nemáme dopředu potřebné informace. Neznáme její předpis, ani efektivní dimenzi návrhového vektoru  $\alpha$ , a při následné optimalizaci pomocí klasických metod se můžeme dostat snadno do značných obtíží, když jako bod do další iterace obdržíme bod  $\alpha \notin \mathcal{M}$ .

Pomocí symbolických výpočtů popsanych v [16] se ukazuje, že množina  $\mathcal{M}$  je nějaká varieta (podle definice 2.8) v prostoru  $\mathbb{R}^\gamma$ , resp. sjednocení variet. Ve většině případů bude množina  $\mathcal{M}$  nekonečná a skoro všude spojitá. Tato vlastnost je velice důležitá. Pokud se nacházíme v bodě  $\alpha$  na varietě  $\mathcal{M}$ , víme, že v nějakém jeho okolí  $\mathcal{U}(\alpha)$  bude existovat další bod  $\bar{\alpha}$  příslušející také této množině. V některých případech se může samozřejmě stát, že množinu  $\mathcal{M}$  bude tvořit konečný počet izolovaných bodů, ale v takovém případě nemá smysl provádět optimalizaci vybraných kritérií.

### 5.2.1 Návrh algoritmu

Problému robustní stabilizace výstupní zpětnou vazbou se současnou optimalizací kriteriálních funkcí je v současné době věnována velká pozornost. V práci [19] řeší autoři problém návrhu regulátoru, který minimalizuje  $H_\infty$  normu matice přenosu, za předpokladu, že regulátor je stabilizující. Optimalizované proměnné jsou zde přímo prvky matice zpětné vazby, je jich tedy tolik, jaký je rozměr matice. Obdobný přístup je uveden i v článku [7]. Pokud však optimalizujeme hodnotu  $H_\infty$  normy pomocí všech prvků v matici výstupní zpětné vazby, nemáme zaručeno, že v každé iteraci bude  $H_\infty$  norma definována, protože regulátor nemusí být při změně svých prvků stabilizující. Aby uvedená situace nenastala, využívá se v [19] při optimalizaci vedlejší podmínka omezující hodnotu spektrální abscissy při změně prvků regulátoru.

V námi navrhovaném algoritmu přiřazení Jordanovy formy výstupní zpětnou vazbou využíváme k optimalizaci kriteriálních funkcí minimální počet parametrů ( $\alpha \in Q(\alpha)$ ). Dimenze návrhového parametru tak může být výrazně nižší, než kdybychom uvažovali všechny prvky regulátoru. Zároveň, pokud nalezneme stabilizující výstupní zpětnou vazbu  $K$ , budeme chtít zajistit, aby všechny zpětné vazby v dalších iteracích byly stabilizující a nedostali bychom se do oblasti, kde nejsou kriteriální funkce definovány.

Uvedený algoritmus je stavěn tak, aby byl jednoduchý a navíc nezávislý na volbě (počtu) kriteriálních funkcí, resp. uměl řešit problémy 5.1 a 5.2. Výsledné hodnoty budou odpovídat lokálnímu optimu. Nyní si nastíníme, jak můžeme postupovat při řešení problémů 5.1 a 5.2.

Úlohu pro nalezení nejlepšího vektoru  $\alpha_{\text{opt}} \in \mathcal{M}$  takového, že hodnota kriteriální funkce bude minimální, rozdělíme do dvou kroků.

V prvním kroku hledáme nejprve počáteční bod pro optimalizaci, tedy takový bod, pro nějž je  $F_2 = 0$ . Jedná se o samostatný úkol, kde např. pomocí Levenbergovy-Marquardtovy metody (viz odstavec 2.6) hledáme řešení soustavy nelineárních rovnic.

V druhém kroku následně startujeme optimalizaci z výsledného bodu  $\alpha$  z prvního kroku. Úkolem je najít další body na varietě  $\mathcal{M}$  blízké výchozímu bodu (s menší funkční hodnotou).

Uvažujme, že vycházíme z bodu  $\alpha_k$  na varietě  $\mathcal{M}$ . Chceme zajistit, aby další vybraný bod  $\alpha_{k+1}$  splňoval  $F_2(\alpha_{k+1}) = 0$  a byl od  $\alpha_k$  ve vzdálenosti  $h$ . Uvedený požadavek na vzdálenost můžeme analogicky formulovat tak, že bod  $\alpha_{k+1}$  bude ležet na  $\gamma$ -rozměrné sféře  $B_\gamma(\alpha_k, h)$  se středem v bodě  $\alpha_k$  a o poloměru  $h$ , tj.

$$B_\gamma(\alpha_k, h) = \{\alpha \in \mathbb{R}^\gamma : \|\alpha - \alpha_k\| = h\}.$$

Takových bodů bude obecně nekonečně mnoho. Následující bod  $\alpha_{k+1}$  chceme tedy celkově určit tak, aby náležel do průniku  $B_\gamma(\alpha_k, h)$  a  $\mathcal{M}$ . Označme

$$\mathcal{P} = B_\gamma(\alpha_k, h) \cap \mathcal{M}.$$

Pro lepší názornost předpokládejme nyní na chvíli nekonečně malý krok  $h$ . Body, které náležejí varietě  $\mathcal{M}$ , budou pro tento případ zároveň ležet na tečné nadrovině  $\mathcal{T}$  variety  $\mathcal{M}$  určené v bodě  $\alpha_k$ . Libovolný bod na tečné nadrovině  $\mathcal{T}$  lze vyjádřit lineární kombinací jejích bázových vektorů. Počet bázových vektorů určuje dimenzi tečné nadroviny a shoduje se s lokální dimenzí variety  $\mathcal{M}$ . Dimenzi variety  $\mathcal{M}$  však neznáme, víme jen, že  $\dim \mathcal{M} \leq \gamma$ . Protože bází lze vygenerovat všechny body nadroviny  $\mathcal{T}$ , lze pomocí ní též vygenerovat i hledané body množiny  $\mathcal{P}$ . Těchto bodů bude opět obecně nekonečně mnoho. Z tohoto důvodu budeme hledat bázové vektory, jejichž počet je konečný. Bázové vektory si zvolíme takové, aby jejich počátečním bodem byl bod  $\alpha_k$  a koncové body ležely v průniku (označme jej  $\mathcal{R}$ ) tečné nadroviny a  $\gamma$ -rozměrné sféry, tedy

$$\mathcal{R} = B_\gamma(\alpha_k, h) \cap \mathcal{T}.$$

Navíc si pro následné snazší určení bázových vektorů vybereme ortogonální bázi.

Každý bázový vektor  $v_i$  je určen bodem  $\alpha_k$  a  $\alpha_{k+1}^i$ . První bod  $\alpha_{k+1}^1$  získáme jako libovolný bod z  $\mathcal{R}$ . Další bod  $\alpha_{k+1}^2$  vybíráme opět z množiny  $\mathcal{R}$ , ale navíc nově uvažujeme, že vektory  $v_1, v_2$  jsou kolmé. Bod  $\alpha_{k+1}^2$  ( $\alpha_{k+1}^2 - \alpha_k = v_2$ ) musí tedy splňovat přidanou rovnici

$$v_1^T v_2 = 0. \tag{5.6}$$

Pro získání dalších bodů  $\alpha_{k+1}^i$  bychom postupovali analogicky. Stále bychom požadovali, aby další bázový vektor  $v_i$  byl kolmý na všechny předchozí  $v_{i-1}, v_{i-2}, \dots, v_1$ . Vektorů  $v_i$  a jejich odpovídajících bodů  $\alpha_{k+1}^i$  bude tolik, kolik je dimenze tečné nadroviny  $\mathcal{T}_n$ .

Ve všech získaných bodech  $\alpha_{k+1}^1, \alpha_{k+1}^2, \dots$  potom spočteme hodnotu kritériální funkce  $I$  a pro další iteraci vybereme bod  $\alpha_{k+1}$ , v němž funkce nabývá nejmenší hodnoty, tj.

$$\alpha_{k+1} = \arg \min_i I(\alpha_{k+1}^i).$$

Při optimalizaci samozřejmě nelze uvažovat nekonečně malé kroky  $h$ . Zvolme tedy krok  $h$  o libovolné velikosti. V takovém případě již neplatí analogie s tečnou nadrovinou a nelze hovořit ani o její bázi a není tak ani možné vygenerovat všechna řešení.

Přesto však můžeme postupovat obdobně, jen nalezené vektory neodpovídají bázi tečné nadroviny, ale představují vhodné ortogonální vektory  $s_i$ , které jsou určeny

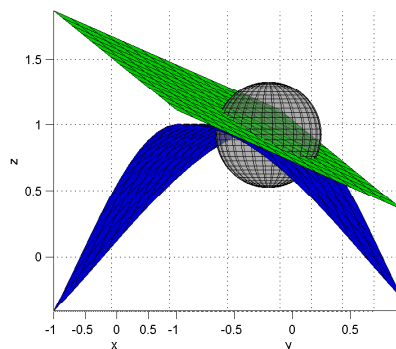
body  $\alpha_k$  a  $\alpha_{k+1}^i$ . Bod  $\alpha_{k+1}^1$  určíme řešením soustavy o rovnicích  $\|\alpha - \alpha_k\| = h$  a  $F(\alpha) = 0$ , tedy  $\alpha_{k+1}^1 \in \mathcal{P}$ . Pro další body  $\alpha_{k+1}^i$  budeme k těmto rovnicím přidávat podmínku, aby nový vektor  $s_i$  byl kolmý na všechny předchozí vektory,

$$s_i^T s_j = 0 \quad \text{pro } j = i - 1, i - 2, \dots, 1.$$

Dalším významným problémem této metody je volba kroku  $h$ . V řadě optimalizačních metod vícerozměrné optimalizace se provádí pro zjištění velikosti kroku  $h$  jednorozměrná optimalizace. Ta zde ale není možná, protože nemáme zaručené, že body generované při jednorozměrné optimalizaci budou ležet na varietě  $\mathcal{M}$ .

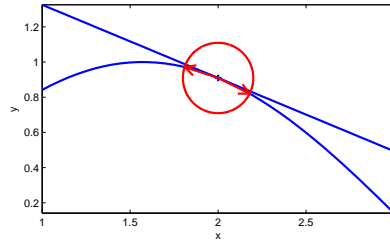
Krok  $h$  volíme v první iteraci pevný, v dalších iteracích jej můžeme modifikovat. V naší práci jsme si vybrali úpravu kroku podle relativní změny funkční hodnoty. Označme si  $f_z = \frac{f(\alpha_k) - f(\alpha_{k+1})}{f(\alpha_k)}$ , (uvažujeme minimalizační úlohu,  $f(\alpha_{k+1}) < f(\alpha_k)$ ). Pokud  $f_z < 0.1$ , předpokládáme, že se funkce pomalu mění, funkční hodnoty klesají pomalu a můžeme si dovolit krok  $h$  zvětšit (např. zdvojnásobit). Jestliže  $f_z > 0.5$ , očekáváme, že se příslušná funkce mění příliš rychle, proto zmenšíme krok (např. na polovinu). Speciálním případem bude, když  $f(\alpha_{k+1}) > f(\alpha_k)$ . Potom nenalezneme lepší řešení, než současné, proto zůstaneme v aktuálním bodě  $\alpha_{k+1} = \alpha_k$  a krok také zmenšíme.

**Poznámka 5.5.** Kdybychom v popsaném návrhu algoritmu uvažovali případ, že  $\gamma = 3$  a množina  $\mathcal{M}$  je dvoudimenzionální varieta ve třírozměrném prostoru. Potom  $B_\gamma = B_3$  bude představovat sféru. V průniku  $\mathcal{R}$  bude ležet nekonečně mnoho bodů, které budou tvořit kružnici. Bázové vektory tečné nadroviny nalezneme dva. Uvedený případ je nastíněn na obrázku 5.2.



Obrázek 5.2: Ukázka průniku sféry s tečnou nadrovinou a varietou ( $\gamma = 3$ ).

**Poznámka 5.6.** Ve speciálních případech, když  $\mathcal{M}$  bude jednorozměrná varieta, bude  $\mathcal{R}$  i  $\mathcal{P}$  tvořen pouze dvěma body. Bázový vektor  $v$ , a stejně tak směr  $s$ , nalezneme potom jeden. Tento případ je pro  $\gamma = 2$  znázorněn na obrázku 5.3.



Obrázek 5.3: Ukázka průniku sféry s tečnou nadrovinou a varietou ( $\gamma = 2$ ).

**Poznámka 5.7.** V případě neúplného přiřazení Jordanovy formy výstupní zpětnou vazbou je vhodnější nejprve lokalizovat všechna vlastní čísla do nějaké vybrané stabilní oblasti a teprve z vypočteného bodu z této fáze provádět optimalizaci dalších zadaných funkcí.

**Poznámka 5.8.** Navržený algoritmus samozřejmě není ideální. Poskytuje jen heuristický přístup pro nalezení lepších výstupních zpětných vazeb. Jeho nevýhodou je, že nemusí najít řešení hned prvního bodu a následnou optimalizaci nelze provádět. Přesto se ukazuje, že v řadě případů po několika iteracích nalezneme řešení, které můžeme považovat za lokální optimum, a odpovídající výstupní zpětnou vazbu  $K$ .

**Poznámka 5.9.** Budeme-li řešit úlohu, u které se dimenze návrhového parametru výrazně sníží, bude obtížné najít řešení prvního kroku. Následná optimalizace by ale byla snazší a rychlejší.

### Algoritmus 5.3. Robustní přiřazení Jordanovy formy výstupní zpětnou vazbou

Vstup: systém  $A, B, C$ , přiřazované Jordanovy bloky, optimalizované funkce  $f_i$  pro  $i = 1, \dots, v$ , penalizace  $p_i$  pro  $i = 1, \dots, v$ , omezení – funkce vedlejších podmínek a mezní hodnota, počáteční vektor  $x_0$  ( $[\alpha_0, q_0, r_0]$ ),

- nalezni bod  $x_1 \in \mathcal{M}$  (řešením soustavy  $F_2(x) = 0$ , (3.17))
- nalezni body  $x_{k+1}^1, x_{k+1}^2, \dots, x_{k+1}^t$
- vypočti hodnotu kriteriální funkce  $I(x) = \max_{i=1, \dots, v} \{p_i f_i(x)\}$  pro každý bod  $x_{k+1}^i$
- zvol  $x_{k+1}$  jako bod s nejmenší funkční hodnotou vypočtenou v bodech  $x_{k+1}^1, x_{k+1}^2, \dots, x_{k+1}^t$ , tj.  $x_{k+1} = \arg \min_i \{I(x_{k+1}^i)\}$

Výstup:  $x_{\text{opt}}$  ( $[\alpha_{\text{opt}}, q_{\text{opt}}, r_{\text{opt}}]$ ), funkční hodnoty  $f_i(x_{\text{opt}})$ , výstupní zpětná vazba



**Příklad 5.3.** Mějme matice  $(A, B, C)$  a Jordanovu formu  $L$  jako v příkladu (3.4). Přiřazujeme tedy bloky

$$\begin{bmatrix} -2 & 1 \\ 0 & -2 \end{bmatrix}, [-1], [-4],$$

jedná se o úplné přiřazení.

Navrhněme nyní výstupní zpětnou vazbu  $K \in \mathcal{K}_n(A, B, L)$ , takovou, aby výsledný uzavřený systém byl stabilní a aby měl maximální komplexní poloměr stability. Počáteční bod  $x_0$  volíme  $[4, 3, 5, -1]$ . Budeme tak minimalizovat funkci  $f_{rc}$ .

V první fázi navrhovaného algoritmu získáme bod

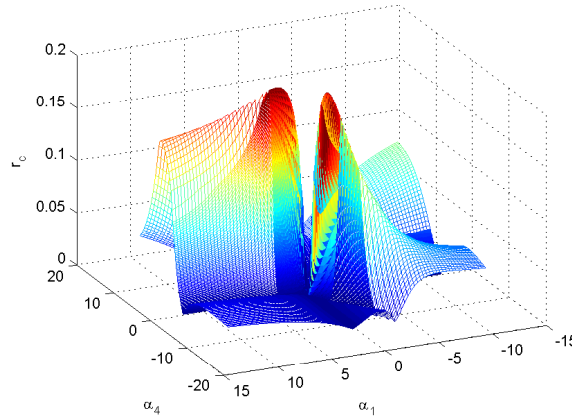
$$x_1 = [4.2656188, 0.3544547, 6.4121276, 4.2082775].$$

V druhé fázi startujeme z bodu  $x_1$ , další postup ukazuje tabulka 5.2. Hodnoty jsou uvedeny zaokrouhlené na sedm desetinných míst.

Číslo iterace	Vektor $x$ , Vektor $s$	Funkční hodnota
1	[4.2656185, 0.3544545, 6.4121261, 4.2082774], [-0.0593660, 0.0169701, 0.00559406, -0.0784562]	7.6337828
2	[4.2062525, 0.3714246, 6.41772017, 4.1298212], [-0.0597700, 0.0164305, 0.0057432, -0.0782593]	7.4099339
3	[4.1464825, 0.3878551, 6.4234633, 4.0515619], [-0.1207478, 0.0311837, 0.0119292, -0.1559013]	6.9722570
4	[4.0257347, 0.4190388, 6.4353926, 3.8956606], [-0.2463425, 0.0552541, 0.0255342, -0.3092092]	6.1381316
5	[3.7793922, 0.4742930, 6.4609268, 3.5864515], [-0.2527817, 0.0449938, 0.0274931, -0.3054846]	5.3611800
6	[3.5266105, 0.5192868, 6.4884199, 3.2809669], [-0.2591059, 0.0338538, 0.0292667, -0.3014320]	5.2363395
7	[3.2675047, 0.5531406, 6.5176866, 2.9795349], [-0.0813019, -0.0951753, -0.7900397, -0.0130007]	5.2154617
⋮	⋮	⋮
26	[2.5245300, 0.2516529, 3.8169277, 2.4292345], [-0.0075191, -0.0087083, -0.0486560, -0.0004718]	5.1089384
27	[2.5170109, 0.2429446, 3.7682717, 2.4287627], [-0.0076335, -0.0088570, -0.0486110, -0.0004168]	5.1081366

Tabulka 5.2: Průběh druhé fáze algoritmu 5.3.

Hodnota komplexního poloměru stability je přibližně  $r_C = \frac{1}{5.108} = 0.196$ . Průběh komplexního poloměru stability znázorňuje obrázek 5.4.



Obrázek 5.4: Komplexní poloměr stability pro příklad 5.3.

**Poznámka 5.10.** Matice stavové i výstupní zpětné vazby můžeme symbolicky vypočítat v explicitním tvaru pomocí knihovny *JordanFormAssignment*<sup>1</sup> sestavené v softwaru Maple 13, vytvořené v rámci práce [16]. Pokud bychom si zpětnou vazbu takto vypočetli, prováděla by se následná optimalizace vzhledem k volným parametrům  $\alpha$  v matici  $F(\alpha)$ , či  $K(\alpha)$ . Potom by se zpětné vazby nemusely počítat numericky v každé iteraci, což by bylo výhodné zejména u výstupní zpětné vazby. Optimalizovat bychom mohli samozřejmě i parametry  $q$  zastupující vlastní čísla a  $r$  získané při neúplném přiřazení. Oproti tomu, při velkém řádu systému a většího počtu návrhových parametrů  $\alpha$  je výpočet racionálních polynomiálních rovnic při symbolickém přístupu nesmírně složitý a je lepší využít numerický výpočet.

<sup>1</sup>Knihovna *JordanFormAssignment* pro software Maple 13 je volně dostupná na internetové adrese [control.zcu.cz/~schlegel/JFA.zip](http://control.zcu.cz/~schlegel/JFA.zip)

## 6 Vytvořený Toolbox v Matlabu

Součástí práce je vytvoření knihovny v softwaru Matlab realizující uvedené navržené algoritmy.

Algoritmy jsou implementovány ve stylu toolboxu, s názvem `Jordan_form_assignment_opt`, který lze do Matlabu přidat importováním složky `JORDAN_FORM_ASSIGNMENT_OPT` včetně všech podsložek (pomocí menu `File` → `Set Path`).

Implementace je rozdělena do čtyř částí:

- UTIL
- STATE\_FEEDBACK
- OUTPUT\_FEEDBACK
- GUI.

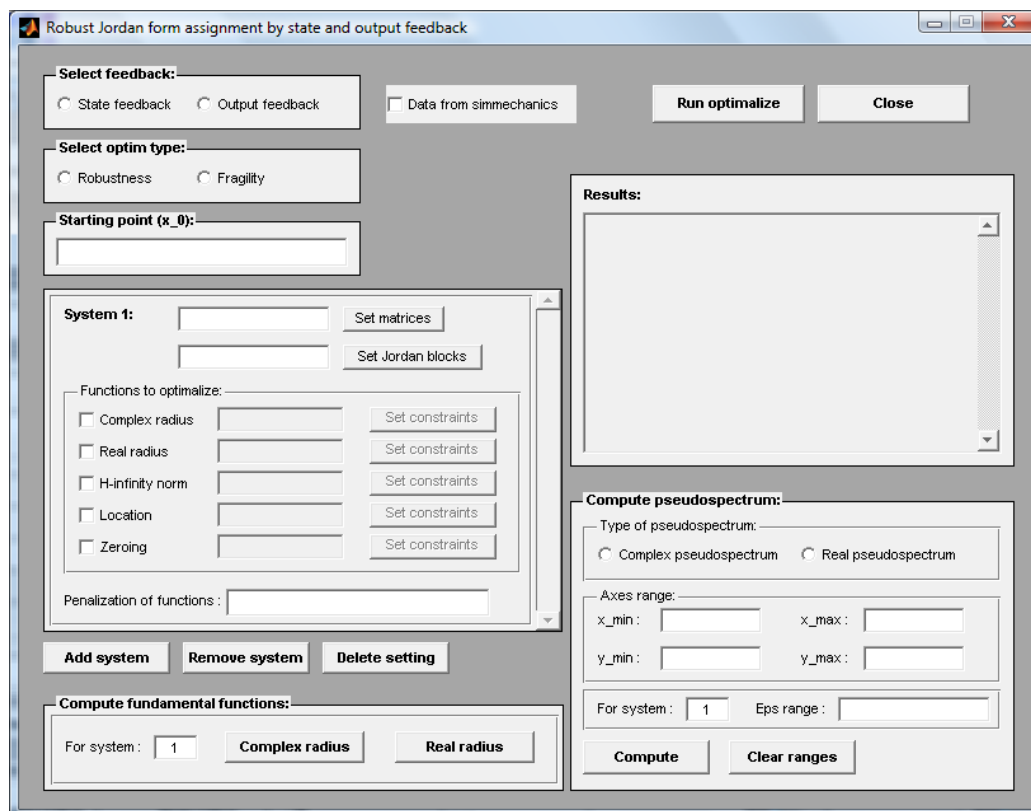
Do složky UTIL jsou zařazeny funkce společné oběma optimalizačním problémům 5.1, 5.2 a ostatní základní funkce. Mezi ně patří výpočet stavové a výstupní zpětné vazby pro úplné i neúplné přiřazení (podle kapitoly 3), algoritmy výpočtu kritérií robustnosti (dle kapitoly 4) a ostatní pomocné funkce (například metoda zlatého řezu). Dále jsou zde uvedeny funkce pro výpočet, resp. vykreslení, komplexního a reálného pseudospektra matice systému.

Modul STATE\_FEEDBACK zahrnuje funkce k nalezení optimální stavové zpětné vazby vzhledem k požadovaným kritériím (podle algoritmu 5.2).

Složka OUTPUT\_FEEDBACK obsahuje funkce pro hledání nejlepší výstupní zpětné vazby. Při optimalizaci se využívá algoritmus navržený v podkapitole 5.2.1, jehož implementace je uvedena také v tomto modulu.

Ve složce GUI je umístěno vytvořené uživatelské rozhraní, které umožňuje pohodlnější používání sestavených funkcí. Uživatelské rozhraní znázorňuje obrázek 6.1.

Využíváme-li rozhraní, je nejprve nutné určit, zda budeme přiřazovat stavovou, nebo výstupní zpětnou vazbu, a také jestli budeme optimalizovat vlastnosti uzavřeného systému vzhledem k jeho robustní stabilitě, nebo fragilitě zpětné vazby. Dále je potřeba zadat matice systému, přiřazované Jordanovy bloky, kritériální funkce a případně jejich omezení. Také můžeme zadat hodnotu počátečního bodu pro optimalizaci, pokud ji neurčíme, vygeneruje se náhodný bod. Jednotlivé kritériální funkce je možné penalizovat.



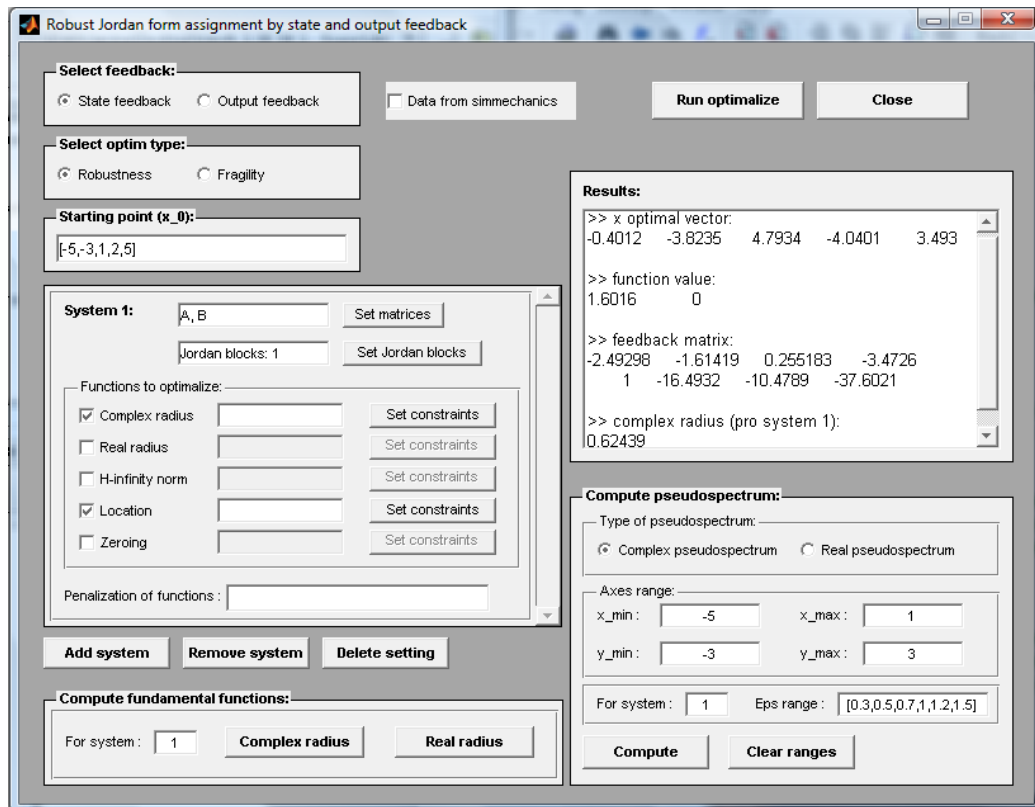
Obrázek 6.1: Grafické uživatelské rozhraní s vyplněnými vstupními daty.

Grafické rozhraní dále umožňuje stabilizovat a optimalizovat vlastnosti současně více systémů, například, kdybychom měli systém s parametry, které by mohly nabývat za určitých podmínek různých hodnot, a jejich stavový popis by se tak lišil. Potom mohou být tedy různé jednak matice systému, rovněž tak kritériální funkce, ale Jordanovy bloky přiřazujeme všem systémům stejné.

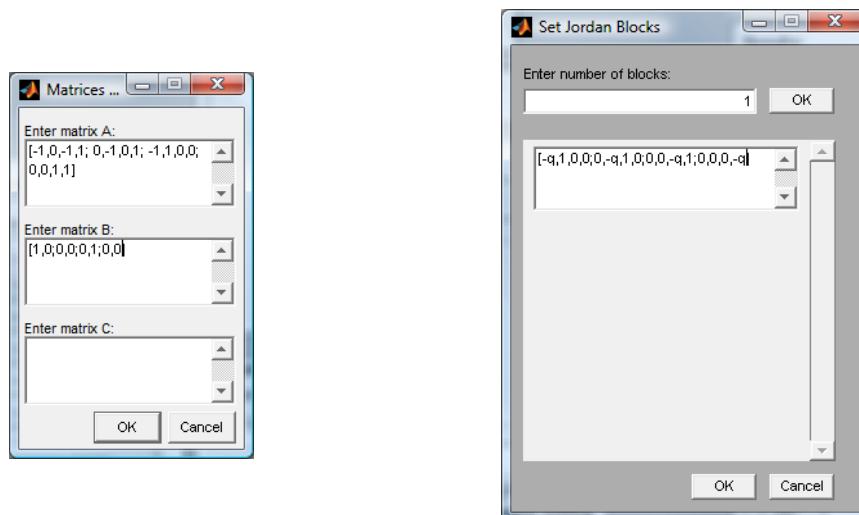
Dále v uživatelském rozhraní poskytujeme možnost výpočtu základních funkcí (komplexního a reálného poloměru) a vykreslení komplexního a reálného pseudospektra pro výsledný uzavřený systém.

Všechny vstupy v rozhraní lze zadávat číselně i pomocí proměnných z pracovní plochy Matlabu. Výstupy (optimální bod, funkční hodnoty a matice zpětné vazby) se jednak zobrazí v panelu pro výsledky a rovněž se uloží do pracovní plochy Matlabu (do proměnných `x_opt_m`, `fval_m`, `feedback_matrix_m`).

Uživatelské rozhraní se spouští příkazem `jordan_form_assignment` z pracovní plochy Matlabu. Obrázek 6.2 znázorňuje vyplněné vstupní hodnoty v uživatelském rozhraní. Na obrázku 6.3 je ukázáno zadávání matic systému a Jordanových bloků.



Obrázek 6.2: Příklad vyplněného grafického uživatelského rozhraní.



Obrázek 6.3: Ukázka zadávání vstupních matic (Set matrices) a Jordanových bloků (Set Jordan blocks).

Uvedme si nyní příklady volání základních funkcí toolboxu bez uživatelského rozhraní.

Jednou z možností, jak zadat matice systému, je definovat matice v samostatné struktuře – `struct('A',A,'B',B,'C',C,'B2',B2,'C2',C2)`, kde  $A, B, C$  představují matice z (3.1) a  $B_2, C_2$  odpovídají maticím podle (4.27).

Další možností je využití formátu `ss`, definovaném v Toolboxu `Control System`, potom například bude – `ss(A,Bi,Co,0,'InputGroup', struct('U1', [1,2], 'U2', [3]), 'OutputGroup', struct('Z1', [1], 'Z2', [2,3]))`, čímž specifikujeme, že matice  $B$  bude tvořena prvním a druhým sloupcem a  $B_2$  třetím sloupcem vstupní matice  $B_i$  a matice  $C$  potom prvním a  $C_2$  druhým a třetím řádkem matice  $C_o$ .

Jordanovy bloky se zadávají ve formě `cell`, například

```
- jordan_blocks = cell(1,n)
- jordan_blocks{1,1}=[-3,1;0,-3], ..., jordan_blocks{1,n}=[-2].
```

Zadávané matice mohou obsahovat nedefinované symbolické proměnné určené k optimalizaci.

Kriteriální funkce jsou dány vektorem znaků, například

```
- crit_functions = ['c','r','h','l','z'],
```

kde `c` odpovídá komplexnímu poloměru stability, `r` reálnému, `h` normě  $H_\infty$  (pro kterou musíme zadat matice  $B_2, C_2$ ), `l` funkci umísťující vlastní čísla uzavřeného systému do dané oblasti, `z` nulování vybraných prvků.

Při zadání funkce `l` je nutné určit meze a případně definovat penalizaci. Tyto hodnoty můžeme umístit do přídatné struktury nastavení, kterou lze přidat k hlavním optimalizačním funkcím. Jako příklad uvedme

```
- options_fun = struct('kappa_vec', [-1, -15, 2], ...
                    'kappa_penalize', [1000, 1000, 10]),
```

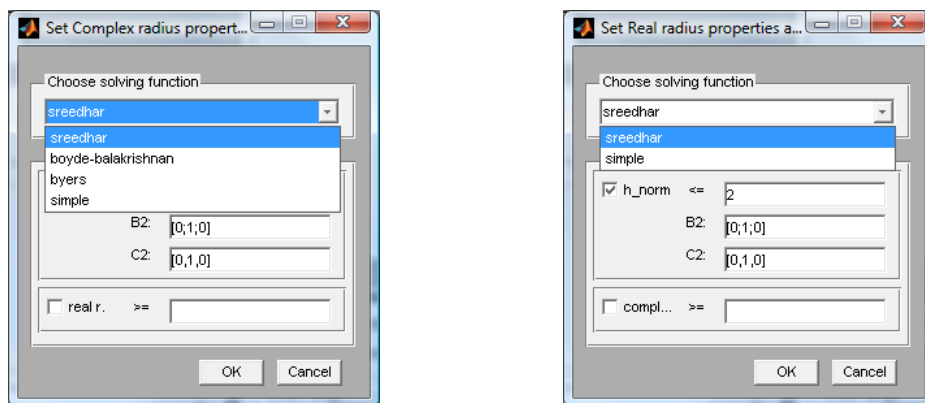
kde v `kappa_vec` jsou hodnoty (jak uvádí podkapitola 4.5.1) po řadě – pravá mez (přímka  $\text{Re}(\lambda) = -1$ ), levá mez (přímka  $\text{Re}(\lambda) = -15$ ) a směrnice přímky, tj.  $\text{Im}(\lambda) = 2\text{Re}(\lambda)$ , a v `kappa_penalize` jsou příslušné penalizace.

V případě zadání znaku `z` se volí, které prvky se mají nulovat. Do pomocné nepovinné struktury nastavení (`options_fun`) potom přidáme pole s názvem `zero_elements` a pozice prvků a případně pole `weight` s vektorem požadovaných vah. Například při nulování prvku na pozici `[2,1]` a váhu 10000 budeme mít

- `options_fun = struct('zero_elements',{[2,1]},'weight',[10000]).`

Komplexní poloměr stability je implicitně počítán podle algoritmu 4.3. Pokud by si uživatel chtěl zvolit jinou metodu, lze si jej v uživatelském rozhraní vybrat v nabídce algoritmů uložené pod tlačítkem **Set constraints**. Z pracovní plochy Matlabu by místo 'c' (či 'c1'), čemuž přísluší algoritmus 4.3, zadal 'c2' pro algoritmus 4.2, 'c3' pro 4.1 a případně 'c4' pro naivní verzi algoritmu k výpočtu komplexního poloměru stability. Obdobně pro výpočet reálného poloměru stability implicitně užíváme algoritmus 4.5 a pro 4.4 bychom museli specifikovat 'r2'.

Nyní uveďme ještě obrázky 6.4a a 6.4b, kde vybíráme funkce pro výpočet a nastavujeme případná omezení.



(a)

(b)

Obrázek 6.4: Ukázka zadávání vedlejších omezení u komplexního (a) a reálného (b) poloměru stability.

V případě, že hledáme stavovou zpětnou vazbu, která systému  $(A, B)$  přiřadí Jordanovy bloky, aby hodnoty kritériálních funkcí pro matici dynamiky uzavřeného systému byly optimální, může mít příkaz k zavolání odpovídající funkce následující obecnou podobu

- `state_feedback_R_opt(vstupní_systém(y), ...  
                          názy_kriteriálních_funkcí, Jordanovy_bloky, ...  
                          počáteční_bod, nastavení).`

Pokud bychom požadovali málo fragilní regulátor, potom

- `state_feedback_F_opt(vstupní_systém(y), ...  
                          názy_kriteriálních_funkcí, Jordanovy_bloky, ...  
                          počáteční_bod, nastavení).`

Uvažujme pro lepší názornost konkrétní příklad, u něhož bude počáteční bod  $x_0 = [1, 2, 3, 4]$ , systém  $(A, B)$ , Jordanovy bloky  $\begin{bmatrix} -2 & 1 \\ 0 & -2 \end{bmatrix}$ ,  $[-1]$  a požadujeme maximální reálný poloměr stability matice uzavřeného systému. Optimalizaci bychom spustili příkazem

```
- [x_opt, func, feedback] = ...
    state_feedback_R_opt(struct('A',A,'B',B),['r'],...
        {[-2,1;0,-2],[-1]},{1,2,3,4}).
```

Obdobně, pokud hledáme vhodnou výstupní zpětnou vazbu, přiřazující systému  $(A, B, C)$  Jordanovy bloky, takovou, aby hodnoty kritériálních funkcí pro matici dynamiky uzavřeného systému byly optimální, budeme používat příkaz

```
- output_feedback_R_opt(vstupní_systém(y),...
    názvy_kritériálních_funkcí, Jordanovy_bloky,...
    počáteční_bod, nastavení).
```

Pro výběr co nejméně fragilního systému bychom volali

```
- output_feedback_F_opt(vstupní_systém(y),...
    názvy_kritériálních_funkcí, Jordanovy_bloky,...
    počáteční_bod, nastavení).
```

Důležitými funkcemi jsou potom také vykreslení komplexního, či reálného pseudospektra, které můžeme volat následujícím stylem

- `psesp_c(struct('A',A+BF))` (pro komplexní pseudospektrum matice dynamiky uzavřeného systému ve smyslu definice 4.5)
- `psesp_c(struct('A',A+BKC,'B',B,'C',C), [-4,1,-6,6])` (pro strukturované komplexní pseudospektrum podle definice 4.6 se zadanými mezemi pro vykreslování)
- `psesp_r(struct('A',A+BF))` (pro reálné pseudospektrum podle definice 4.7)
- `psesp_r(struct('A',A+BKC,'B',B,'C',C), [-4,1,-6,6])` (reálné pseudospektrum ve smyslu definice 4.8 se zadanými mezemi).



**Příklad 6.1.** Ukažme si strukturu volání optimalizační funkce z pracovní plochy Matlabu na příkladě. Hledáme lokálně optimální stavovou zpětnou vazbu, aby výsledný uzavřený systém měl maximální reálný poloměr stability.

```
>> A = [0, 0, 0, 1, 0, 0; 0, 0, 0, 0, 1, 0; 0, 0, 0, 0, 0, 1; ...
        -1, 1, 0, 0, 0, 0; 1, -2, 1, 0, 0, 0; 0, 1, -1, 0, 0, 0];
>> B = [0, 0; 0, 0; 0, 0; 1, 0; 0, 0; 0, -1];
>> jordan_blocks = cell(1,3);
>> jordan_blocks{1,1}=[-2, 1; 0, -2];
>> jordan_blocks{1,2}=[-3, 1; 0, -3];
>> jordan_blocks{1,3}=[-2, 1; 0, -2];
>> system_matrix = struct('A', A, 'B', B);
>> crit_functions = ['r'];
>> x0 = [-5, -3];
>> [x_opt, fval_opt, F_opt] = ...
        state_feedback_R_opt(system_matrix, crit_functions, ...
        jordan_blocks, x0)

>> pspes_r(struct('A', A + B*F_opt))
```

**Příklad 6.2.** Další příklad uvádí kód pro optimalizaci výstupní zpětné vazby  $K$  takové, aby matice zpětné vazby byla co nejméně křehká, maximalizujeme komplexní poloměr stability. Ukážeme si dále volání výpočtu parametrické matice  $Q(\alpha)$  a vykreslení komplexního strukturovaného pseudospektra matice uzavřeného systému v daných mezích.

```
>> A = [1, 1, 0, 0; 0, 0, 0, 1; 0, 1, 1, 0; 1, 0, 1, 1];
>> B = [0, 0; 0, 0; 1, 0; 0, 1];
>> C = [1, 0, 0, 0; 0, 1, 0, 0; 0, 0, 1, 0];
>> jordan_blocks = cell(1,3);
>> jordan_blocks{1,1} = [-2, 1; 0, -2];
>> jordan_blocks{1,2} = [-4];
>> jordan_blocks{1,3} = [-1];
>> system_matrix = struct('A', A, 'B', B, 'C', C);
>> x0 = [1, -2, -3, 1]

>> [x_opt, fval_opt, K_opt] = ...
        output_feedback_F_opt(system_matrix, ['c'], jordan_blocks, x0)

>> Q = matrixQ(B, jordan_blocks);

>> pspes_c(struct('A', A + B*K_opt*C, 'B', B, 'C', C), [-6, 0, -3, 3])
```

**Příklad 6.3.** Nyní naznačíme optimalizaci stavové zpětné vazby pro případ neúplného přiřazení s lokací vlastních čísel do vymezené oblasti. Budeme maximalizovat hodnotu komplexního poloměru počítaného algoritmem 4.2.

```
>> A = [1, 0, 1, 3, 0; 0, 1, 1, 2, 1; 1, 0, 1, 3, 3;...
        2, 2, 0, 2, 1; 0, 1, 2, 2, 3];
>> B = [1, 0; 0, 0; 0, 0; 0, 1; 0, 0];
>> jordan_blocks = cell(1,3);
>> jordan_blocks{1,1} = [-1, 1; 0, -1];
>> jordan_blocks{1,2} = [-2];
>> jordan_blocks{1,3} = [-1];
>> system_matrix = struct('A', A, 'B', B);
>> x0 = [1, 3, 5, -3, -8, 3];
>> options_fun = struct('kappa_vec', [-1/2, -10, 0],...
                        'kappa_penalize', [10000, 10000, 0]);
>> [x_opt, fval_opt, F_opt] = ...
    state_feedback_R_opt(system_matrix, ['c2', '1'],...
                        jordan_blocks, x0, options_fun)
```

## 7 Příklady

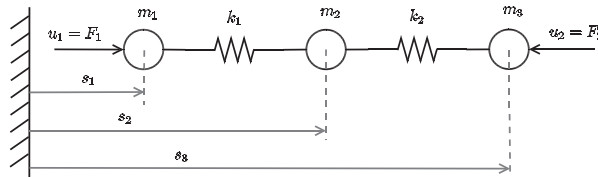
Navržené metody a algoritmy aplikujeme nyní na několik reálných příkladů. Při odvozování stavového popisu systému budeme využívat Lagrangeovu metodu [24], ve které sestavujeme Lagrangeovy rovnice

$$\frac{d}{dt} \left( \frac{\delta L}{\delta \dot{q}_i} \right) - \left( \frac{\delta L}{\delta q_i} \right) = Q_i, \quad 1 \leq i \leq n, \quad (7.1)$$

kde  $L = T - V$  je tzv. Lagrangián,  $T$  kinetická energie soustavy,  $V$  potenciální energie,  $q_i$  představují zobecněné souřadnice a  $Q_i$  zobecněné síly.

### 7.1 Tříhmotový systém

**Příklad 7.1.** Uvažujme tříhmotový systém se dvěma pružinami a se dvěma vstupy, jak ukazuje obrázek 7.1



Obrázek 7.1: Tříhmotový systém.

Pro hodnoty  $m_1 = m_2 = m_3 = 1$  a  $k_1 = k_2 = 1$ , získáme následující stavový popis

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}.$$

Chceme-li přiřadit Jordanovu formu

$$L = \text{diag} \left\{ \begin{bmatrix} -2 & 1 \\ 0 & -2 \end{bmatrix}, \begin{bmatrix} -3 & 1 \\ 0 & -3 \end{bmatrix}, \begin{bmatrix} -2 & 1 \\ 0 & -2 \end{bmatrix} \right\},$$

parametrická matice  $Q(\alpha)$  bude

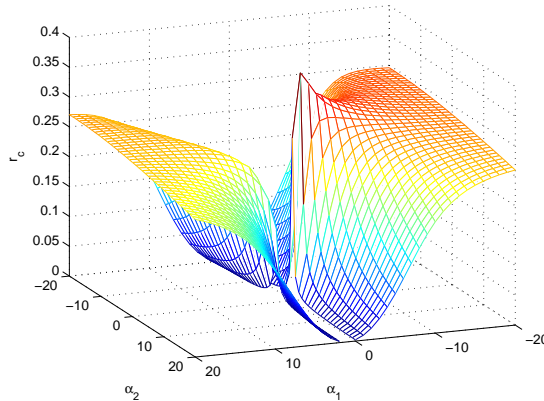
$$Q(\alpha) = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \alpha_1 & \alpha_2 & 1 & 0 \end{bmatrix}.$$

Pro náš systém dostaneme explicitní parametrizaci množiny  $\mathcal{F}_n(A, B, L)$  danou dvěma parametry. Její předpis je dán následovně

$$F = \left[ \begin{array}{cc} \frac{62\alpha_1^2 - 925\alpha_1 - 1890\alpha_2 + 863}{-25(1-2\alpha_1 + \alpha_1^2)} & \frac{23\alpha_1^2 + 800\alpha_1 + 540\alpha_2 - 823}{-25(1-2\alpha_1 + \alpha_1^2)} \\ \frac{788\alpha_1^2 - 775\alpha_1 - 13 + 1890\alpha_2}{25(-1 + \alpha_1)^2} & \frac{823\alpha_1^2 - 800\alpha_1 + 540\alpha_2 - 23}{-25(1-2\alpha_1 + \alpha_1^2)} \\ \frac{13\alpha_1^2 + 775\alpha_1 + 1890\alpha_2 - 788}{25(-1 + \alpha_1)^2} & \frac{197\alpha_1^2 - 700\alpha_1 - 540\alpha_2 + 503}{-50(1-2\alpha_1 + \alpha_1^2)} \\ \frac{863\alpha_1^2 - 925\alpha_1 + 1890\alpha_2 + 62}{25(1-2\alpha_1 + \alpha_1^2)} & \frac{3(101\alpha_1^2 - 100\alpha_1 + 180\alpha_2 - 1)}{50(1-2\alpha_1 + \alpha_1^2)} \\ \frac{31\alpha_1^2 + 1000\alpha_1 + 3780\alpha_2 - 1031}{25(1-2\alpha_1 + \alpha_1^2)} & \frac{3(-101 + \alpha_1^2 + 180\alpha_2 + 100\alpha_1)}{50(1-2\alpha_1 + \alpha_1^2)} \\ \frac{1031\alpha_1^2 - 1000\alpha_1 - 31 + 3780\alpha_2}{25(1-2\alpha_1 + \alpha_1^2)} & \frac{503\alpha_1^2 - 700\alpha_1 + 540\alpha_2 + 197}{50(1-2\alpha_1 + \alpha_1^2)} \end{array} \right],$$

kde  $\alpha_1 \neq 1$ .

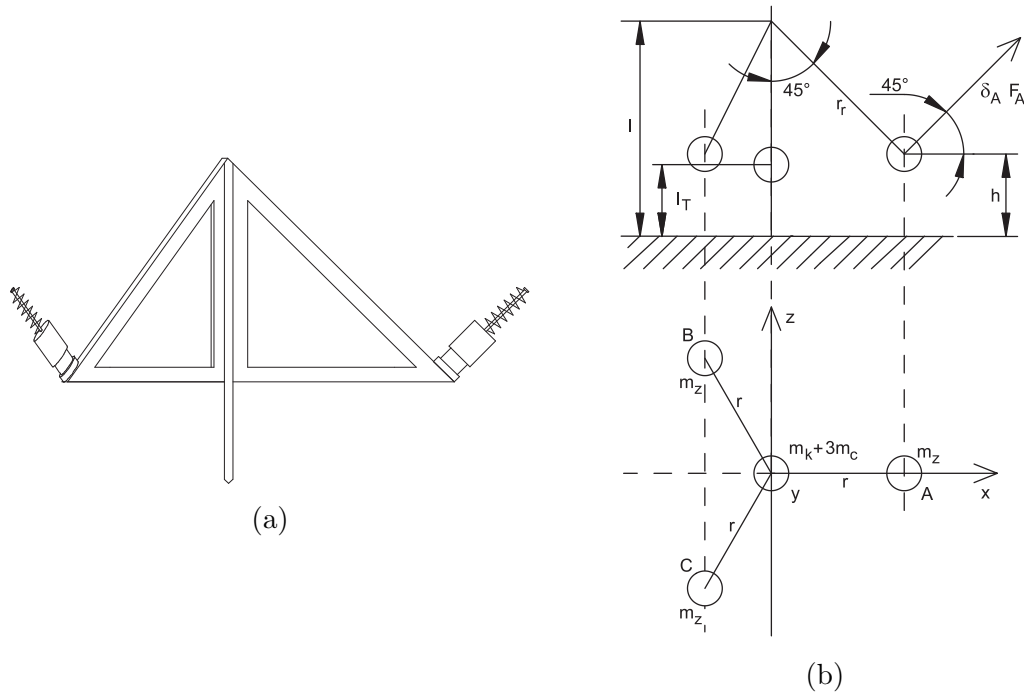
Volné parametry  $\alpha_1, \alpha_2$  chceme zvolit tak, aby uzavřený systém měl maximální komplexní poloměr stability. Pomocí uvedených algoritmů zjistíme, že se optima nabývá pro parametry  $\alpha_1 = -1$  a  $\alpha_2 = 0$  a jeho hodnota je  $r_C = 0.38028$ . Obrázek 7.2 ukazuje průběh komplexního poloměru stability  $r_c$  v závislosti na  $\alpha_1$  a  $\alpha_2$ .



Obrázek 7.2: Komplexní poloměr stability pro příklad 7.1.

## 7.2 Model robotu jednožky - systém 10. řádu

Robot jednožky se skládá ze základní svislé části (nožky) a tří ramen, která mezi sebou svírají úhel  $120^\circ$ . Kolmo na každé rameno je připevněn akční člen v podobě *voice coil* aktuátoru. Model robotu znázorňuje obrázek 7.3. Vzhledem k symetrii konstrukce robotu se jeho těžiště v klidovém stavu nachází ve vertikální ose robotu. Posuny pohyblivých částí aktuátorů vychylujeme těžiště robotu, abychom jej ustabilizovali v požadované poloze.



Obrázek 7.3: Model robotu jednožky.

Za stavové veličiny si vezmeme úhel  $\varphi$  rotace kolem osy  $z$  a jeho derivaci  $\dot{\varphi}$ , úhel  $\psi$  rotace kolem osy  $x$  a jeho derivaci  $\dot{\psi}$ , dále polohy  $\delta_A$ ,  $\delta_B$  a  $\delta_C$  hmot magnetů  $m_z$  a jejich derivace  $\dot{\delta}_A$ ,  $\dot{\delta}_B$  a  $\dot{\delta}_C$ . Vstupy systému budou odpovídat silám  $F_A$ ,  $F_B$  a  $F_C$ . Protože budeme moci měřit jen  $\varphi$ ,  $\psi$ ,  $\dot{\varphi}$  a  $\dot{\psi}$ , budeme navrhovat výstupní zpětnou vazbu.

Zvolíme si následující hodnoty veličin modelu:  $l = 0.209$  m,  $l_T = 0.076$  m,  $r_r = 0.2176$  m,  $d_0 = 0.035$  m,  $m_k = 0.166$  kg,  $m_z = 0.0475$  kg,  $m_c = 0.0165$  kg,  $g = 9.81$  ms<sup>-2</sup>,  $k = 20$  kgs<sup>-2</sup>. Vzhledem k velkému řádu systému i velkému počtu návrhových parametrů nebudeme uvádět symbolický stavový popis nebo následně všechny výstupy získané při optimalizaci.

Úkolem je najít vhodnou výstupní zpětnou vazbu, která robot po vychýlení z rovnovážné polohy ustabilizuje v horní vzpřímené poloze.

Pro uvedené hodnoty veličin dostaneme následující stavový popis:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{1745}{4} & \frac{38}{5} & \frac{38}{5} & \frac{1434}{253} & \frac{108}{11} & -\frac{463}{21} & \frac{96}{193} & \frac{96}{193} & 0 & 0 \\ \frac{38}{5} & -\frac{1745}{4} & \frac{38}{5} & \frac{1434}{253} & -\frac{108}{11} & \frac{96}{193} & -\frac{463}{21} & \frac{96}{193} & 0 & 0 \\ \frac{38}{5} & \frac{38}{5} & -\frac{1745}{4} & -\frac{1417}{125} & 0 & \frac{96}{193} & \frac{96}{193} & -\frac{463}{21} & 0 & 0 \\ -\frac{871}{8} & -\frac{871}{8} & \frac{871}{4} & \frac{4852}{77} & 0 & -\frac{57}{8} & -\frac{57}{8} & \frac{57}{4} & 0 & 0 \\ -\frac{1320}{7} & \frac{1320}{7} & 0 & 0 & \frac{4852}{77} & -\frac{543}{44} & \frac{543}{44} & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\frac{96}{193} & -\frac{96}{193} & \frac{463}{21} \\ -\frac{96}{193} & \frac{463}{21} & -\frac{96}{193} \\ \frac{463}{21} & -\frac{96}{193} & -\frac{96}{193} \\ -\frac{57}{4} & \frac{57}{8} & \frac{57}{8} \\ 0 & -\frac{543}{44} & \frac{543}{44} \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Vlastní čísla matice dynamiky otevřeného systému jsou

$$\Lambda(A) = \{-10.5263 \pm 17.6139j, -11.1854 \pm 17.5464j, -11.1854 \pm 17.5464j, \\ -7.5738, -7.5738, 7.3996, 7.3996\}.$$

Jedná se tedy o nestabilní systém (dva jeho póly jsou kladné). Budeme chtít daný systém stabilizovat výstupní zpětnou vazbou – všechna vlastní čísla přesunout do levé otevřené poloroviny komplexní roviny.

Uvažujme, že budeme přiřazovat Jordanovy bloky

$$L_1 = \begin{bmatrix} -3 & 1 \\ 0 & -3 \end{bmatrix}, L_2 = \begin{bmatrix} -3 & 1 \\ 0 & -3 \end{bmatrix},$$

jedná se tedy o neúplné přiřazení. Výsledný systém je 10. řádu, se třemi vstupy a čtyřmi výstupy. Pro dané Jordanovy bloky získáme příslušnou parametrickou matici

$$Q(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{bmatrix}.$$

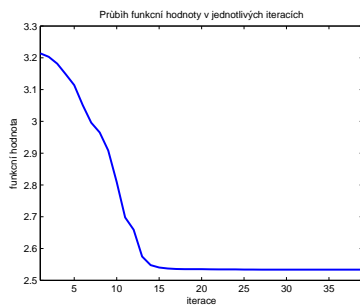
Protože je jedná o neúplné přiřazení, obdržíme dalších  $m(n - s) = 3(10 - 4) = 18$  návrhových parametrů. Celkem tak budeme mít návrhový vektor  $[\alpha, q] \in \mathbb{R}^{22}$ .

Za kritériální funkci si zvolme reálný poloměr stability, který budeme maximalizovat za účelem nalezení vhodné výstupní zpětné vazby. Výsledky pro různě volené počáteční vektory uvádí tabulka 7.1. Na obrázcích 7.4 a 7.5 jsou pro výsledky prvního

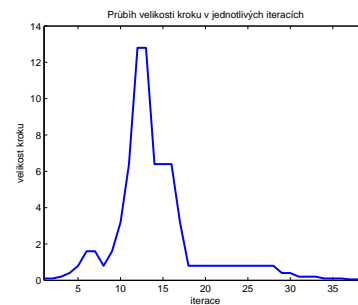
	$r_{\mathbb{R}}(A_c)$	počet iterací
1.	0.3947	48
2.	0.3901	21
3.	0.3935	25
4.	0.3905	34
5.	0.3940	24

Tabulka 7.1: Hodnoty reálného poloměru stability pro různé počáteční vektory.

a druhého řádku tabulky 7.1 znázorněny průběh poklesu funkční hodnoty (převrácené hodnoty reálného poloměru stability) a průběh velikosti kroku v závislosti na iteracích.

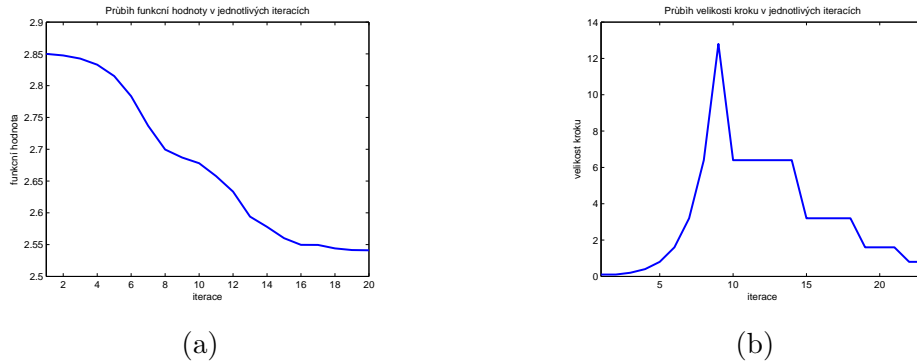


(a)



(b)

Obrázek 7.4: Průběh funkční hodnoty (a) a velikosti kroku (b) v závislosti na aktuální iteraci, řádek 1 tabulky 7.1.



Obrázek 7.5: Průběh funkční hodnoty (a) a velikosti kroku (b) v závislosti na aktuální iteraci, řádek 2 tabulky 7.1.

Hodnota reálného poloměru stability pro uvedený model nám tak vychází přibližně 0.3947. Při výpočtu prvního řádku tabulky 7.1, u něhož je hodnota reálného poloměru největší, vycházíme z počátečního vektoru

$$x_0 = [-7.8798, 16.3969, -2.9954, 0.3695, -12.8467, 6.9088, -2.4908, \\ 18.3542, 19.7248, 8.7715, 13.9481, 12.5950, -6.0301, 4.6822, -1.9249, \\ -12.6968, 3.5055, -1.0625, -15.1864, 11.8541, -10.2447, -18.0192].$$

Výsledný vektor a matice výstupní zpětné vazby jsou

$$x_{\text{opt}} = [-7.0422, 13.4466, -5.2929, 9.5530, 2.4957, 2.1219, -5.0479, \\ 1.3106, 2.2039, 7.7473, 2.1890, -5.6561, 4.0578, -3.3312, 4.5424, \\ -3.3030, -5.2395, 3.3640, 1.2940, 3.4037, -6.6476, -5.0170],$$

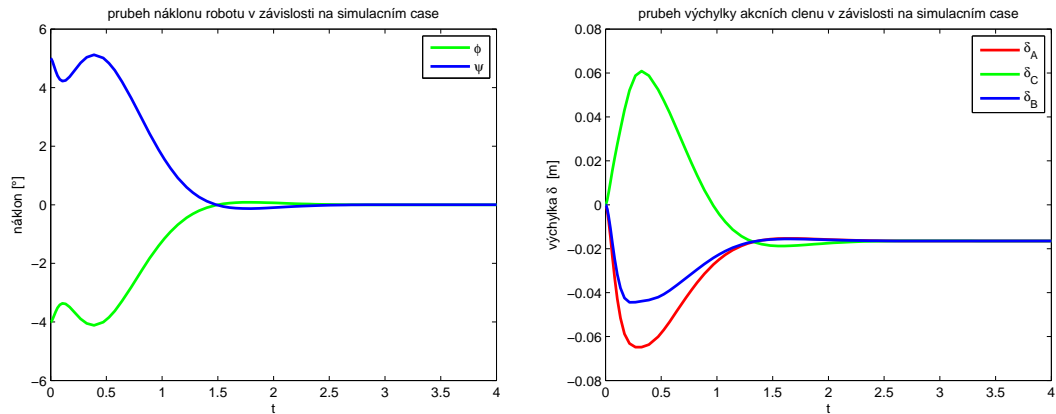
$$K_{\text{opt}} = \begin{bmatrix} -0.3497 & 0.5989 & 12.4505 & 2.3336 \\ 10.7487 & 2.0759 & -6.7724 & -0.2246 \\ -11.4480 & -0.8781 & -6.7724 & -0.2246 \end{bmatrix}. \quad (7.2)$$

Vlastní čísla matice dynamiky uzavřeného systému  $A + BK_{\text{opt}}C$  jsou přibližně hodnoty

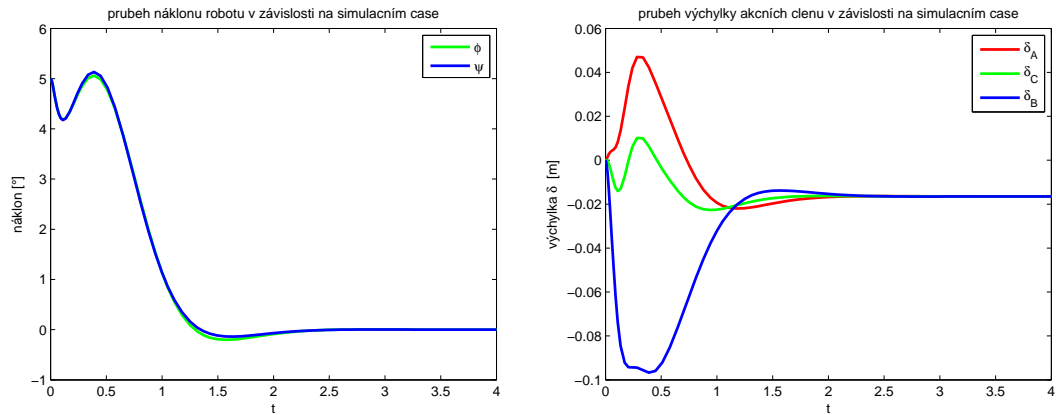
$$\Lambda(A + BK_{\text{opt}}C) = \{-10.5263 \pm 17.6139j, -45.8489, -45.8489, -7.1490, -7.1490, \\ -3.0000, -3.0000, -3.0000, -3.0000\}.$$

Následující obrázky 7.6, 7.7 znázorňují průběh stabilizace nejprve při vychýlení  $5^\circ$  okolo osy  $x$  a  $-4^\circ$  okolo osy  $z$  pro uvedené vypočtené výstupní hodnoty, další potom vychýlení  $5^\circ$  okolo osy  $x$  a  $5^\circ$  okolo osy  $z$  pro porovnání. Na obrázku 7.8 jsou vykreslena pseudospektra matice  $A + BK_{\text{opt}}C$ , kde  $K_{\text{opt}}$  je matice (7.2)

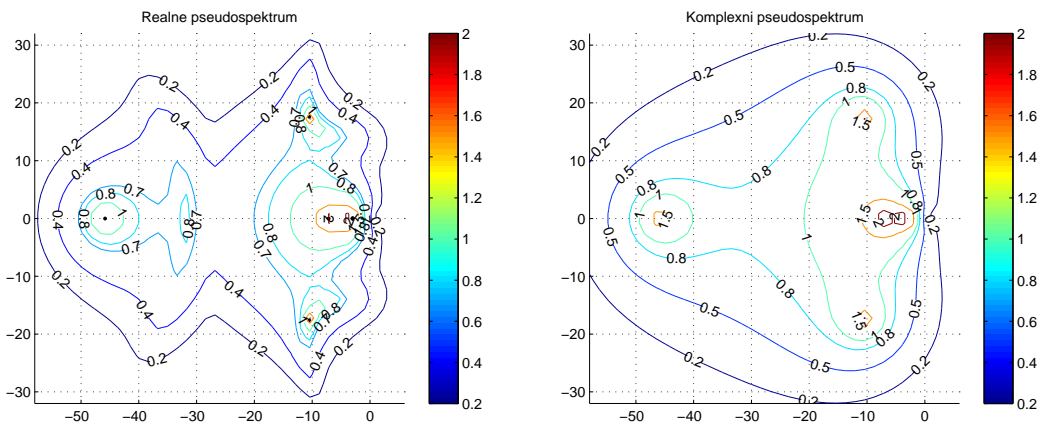




Obrázek 7.6: Průběh stabilizace modelu robotu jednozářky pro vychýlení  $5^\circ$  okolo osy  $x$  a  $-4^\circ$  okolo osy  $z$ .



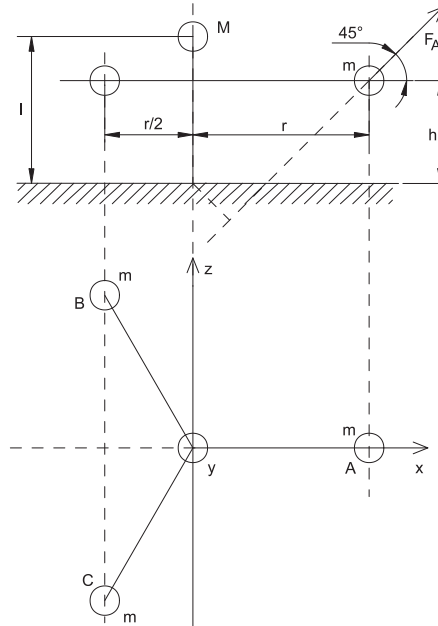
Obrázek 7.7: Průběh stabilizace modelu robotu jednozářky pro vychýlení  $5^\circ$  okolo osy  $x$  a  $5^\circ$  okolo osy  $z$ .



Obrázek 7.8: Reálné a komplexní pseudospektrum matice dynamiky uzavřeného systému.

### 7.3 Model robotu jednoočky - zjednodušení

Uvažujme zjednodušený model robotu jednoočky podle obrázku 7.9. Nebudeme-li uvažovat hmoty  $m_z$  získáme model 4. řádu. Protože opět budeme moci měřit veličiny  $\varphi$ ,  $\dot{\varphi}$ ,  $\psi$ ,  $\dot{\psi}$ , máme k dispozici celý stav a budeme navrhovat stavovou zpětnou vazbu.



Obrázek 7.9: Model robotu jednoočky – zjednodušení.

Pro uvažované obecné veličiny získáme matice stavového popisu systému robotu ve tvaru

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ a_1 & 0 & 0 & 0 \\ 0 & a_2 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & b_1 & b_2 \\ b_3 & b_4 & b_5 \end{bmatrix},$$

kde

$$a_1 = \frac{72gh^3m^2 + 24Mghl^2m + 12Mglmr^2 + 36ghm^2r^2 + 24Mgh^2lm + 8M^2gl^3}{2(12Ml^2mr^2 + 36m^2h^2r^2 + 24Ml^2mh^2 + 36m^2h^4 + 4M^2l^4 + 9m^2r^4)},$$

$$a_2 = \frac{36ghm^2r^2 + 8M^2gl^3 + 24Mghl^2m + 12Mglmr^2 + 72gh^3m^2 + 24Mgh^2lm}{2(12Ml^2mr^2 + 36m^2h^2r^2 + 24Ml^2mh^2 + 36m^2h^4 + 4M^2l^4 + 9m^2r^4)},$$

$$b_1 = \frac{-6\sqrt{6}h^3m + 3\sqrt{6}mr^3 - 2M\sqrt{6}hl^2 + 2M\sqrt{6}l^2r + 6\sqrt{6}h^2mr - 3\sqrt{6}hmr^2}{2(12Ml^2mr^2 + 36m^2h^2r^2 + 24Ml^2mh^2 + 36m^2h^4 + 4M^2l^4 + 9m^2r^4)},$$

$$\begin{aligned}
b_2 &= \frac{6\sqrt{6}h^3m - 3\sqrt{6}mr^3 + 2M\sqrt{6}hl^2 - 2M\sqrt{6}l^2r - 6\sqrt{6}h^2mr + 3\sqrt{6}hmr^2}{2(12Ml^2mr^2 + 36m^2h^2r^2 + 24Ml^2mh^2 + 36m^2h^4 + 4M^2l^4 + 9m^2r^4)}, \\
b_3 &= \frac{4Ml^2h\sqrt{2} - 4Ml^2r\sqrt{2} - 12mh^2r\sqrt{2} + 6mr^2h\sqrt{2} - 6mr^3\sqrt{2} + 12mh^3\sqrt{2}}{2(12Ml^2mr^2 + 36m^2h^2r^2 + 24Ml^2mh^2 + 36m^2h^4 + 4M^2l^4 + 9m^2r^4)}, \\
b_4 &= \frac{3mr^3\sqrt{2} - 6mh^3\sqrt{2} - 2Ml^2h\sqrt{2} + 2Ml^2r\sqrt{2} + 6mh^2r\sqrt{2} - 3mr^2h\sqrt{2}}{2(12Ml^2mr^2 + 36m^2h^2r^2 + 24Ml^2mh^2 + 36m^2h^4 + 4M^2l^4 + 9m^2r^4)}, \\
b_5 &= \frac{3mr^3\sqrt{2} - 6mh^3\sqrt{2} - 2Ml^2h\sqrt{2} + 2Ml^2r\sqrt{2} + 6mh^2r\sqrt{2} - 3mr^2h\sqrt{2}}{2(12Ml^2mr^2 + 36m^2h^2r^2 + 24Ml^2mh^2 + 36m^2h^4 + 4M^2l^4 + 9m^2r^4)}.
\end{aligned}$$

Vlastní čísla matice dynamiky otevřeného systému jsou přibližně hodnoty

$$\Lambda(A) = \{7.62334, 7.62334, -7.62334, -7.62334\}.$$

Parametry modelu si zvolme (obdobně jako u příkladu robotu jednoožky 10. řádu) podle obrázku 7.9, tj.  $g = 9.81 \text{ ms}^{-2}$ ,  $m = 0.064 \text{ kg}$ ,  $M = 0.166 \text{ kg}$ ,  $l = 0.076 \text{ m}$ ,  $h = 0.04 \text{ m}$ ,  $r = 0.15 \text{ m}$ . Pro uvedený systém budeme navrhovat stavovou zpětnou vazbu, abychom optimalizovali vybrané vlastnosti systému. Postupně si uvedeme několik různých přiřazení Jordanových bloků. Nyní si zvolme Jordanovy bloky

$$L_1 = \begin{bmatrix} -p & 1 \\ 0 & -p \end{bmatrix}, \quad L_2 = \begin{bmatrix} -p & 1 \\ 0 & -p \end{bmatrix}, \quad (7.3)$$

nejprve s konkrétní hodnotou  $p = 10$  a maximalizujeme komplexní poloměr stability.

Při řešení uvedeného problému vychází stále konstantní hodnota (pro libovolné parametry návrhového vektoru  $\alpha$ ) komplexního poloměru stability,  $r_{\mathbb{C}}(A + BF) = 0.7157$ , (tj. funkční hodnota při minimalizaci 1.3973).

Analogicky bychom pro zkoumaný systém  $(A, B)$  a pevně zvolené libovolné číslo  $p$  dospěli rovněž ke stejné hodnotě. Potom bude komplexní poloměr stability pro všechny zpětné vazby totožný, (pro všechny parametry  $\alpha$  dostaneme stejné ohodnocení). K tomuto závěru bychom dospěli i pro případ kritéria v podobě reálného poloměru stability. Navíc hodnota komplexního a reálného poloměru zde pro každé pevné  $p$  vyjdou stejné. Abychom mohli vybrat zpětnou vazbu s lepšími vlastnostmi než některé ostatní, přidáme další parametr – vlastní číslo  $p$ .

V takovém případě budeme řešit úlohu

$$\min_{[\alpha, p]} \max\{f_{r_{\mathbb{C}}}, f_l\},$$

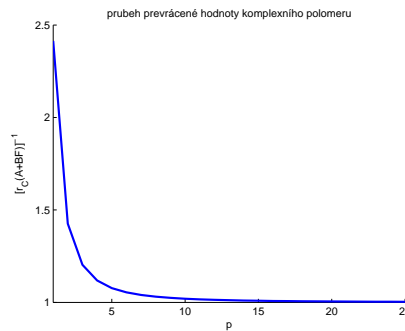
kde funkce  $f_l$  je přidána za účelem lokace čísla  $p$  do požadované oblasti a zajištění stability uzavřeného systému.

Získáme očekávané výsledky, podle kterých se převrácená hodnota komplexního poloměru zmenšuje s větší vzdáleností vlastního čísla od imaginární osy. Potom pro oblast vymezenou přímkami  $\operatorname{Re}(\lambda) = -20$  a  $\operatorname{Re}(\lambda) = -1$ , počáteční vektor  $x_0 = [-2, -5, 1, 1, 6]$  dostaneme  $r_{\mathbb{C}}(A + BF_{\text{opt}}) = 0.9950$  a

$$x_{\text{opt}} = [-2.00, -5.00, 1.00, 1.00, 19.99],$$

$$F_{\text{opt}} = \begin{bmatrix} -272.03117 & 150.33702 & -14.04278 & 7.52061 \\ -283.67099 & 130.17625 & -15.05948 & 5.75963 \\ -260.39134 & 130.17625 & -13.02608 & 5.75963 \end{bmatrix}.$$

Jak vidíme, parametry  $\alpha$  z parametrické matice  $Q(\alpha)$  se nemění (skutečně nemají na hodnotu poloměru vliv). Mění se tak jen hodnota vlastního čísla. Protože převrácená hodnota komplexního poloměru klesá se vzdáleností od imaginární osy, parametr pro vlastní číslo se tak bude pohybovat kolem hranice oblasti, kterou jsme si zvolili. Opět zde tak klesá význam optimalizace. Závislost převrácené hodnoty komplexního poloměru stability na volbě čísla  $p$  znázorňuje obrázek 7.10.



Obrázek 7.10: Převrácená hodnota komplexního poloměru stability.

Jako další možnost uvažujme nyní jeden přiřazovaný Jordanův blok se čtyřnásobným vlastním číslem

$$L = \begin{bmatrix} -p & 1 & 0 & 0 \\ 0 & -p & 1 & 0 \\ 0 & 0 & -p & 1 \\ 0 & 0 & 0 & -p \end{bmatrix}. \quad (7.4)$$

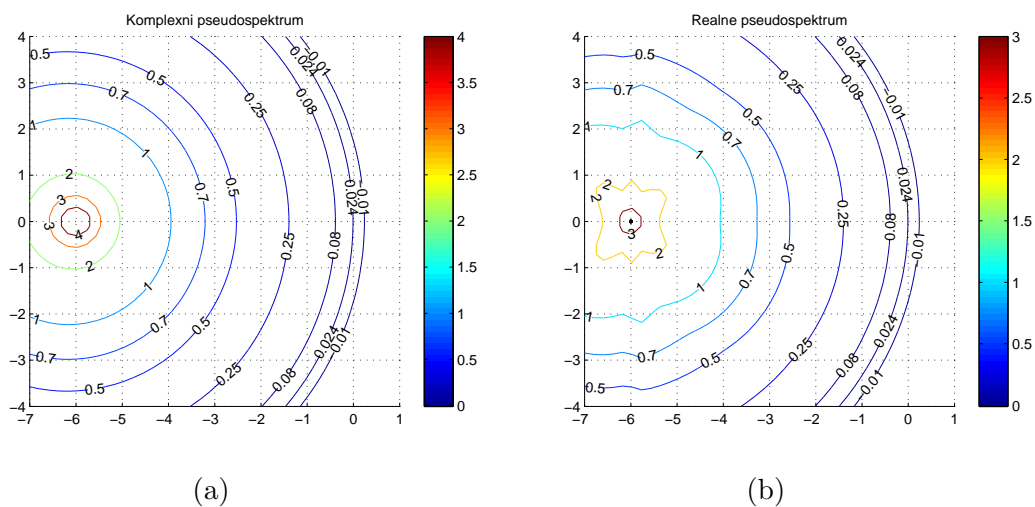
Vybereme si hodnotu  $p = 6$  (pro tu je v předchozí části příkladu již dostatečně malá hodnota  $[r_{\mathbb{C}}(A + BF)]^{-1}$ ). Z několika (přibližně třiceti) náhodně vygenerovaných počátečních podmínek jsme získali nejlepší hodnotu komplexního poloměru stability  $r_{\mathbb{C}}(A + BF_{\text{opt}}) = 0.9486$ , kde

$$F_{\text{opt}} = \begin{bmatrix} 0.7590 & -1.6346 & 0.1337 & -0.3712 \\ -1.5823 & -5.8761 & -0.1540 & -0.9318 \\ 3.1507 & -5.7770 & 0.4384 & -0.8987 \end{bmatrix}. \quad (7.5)$$

Výsledný optimální vektor je

$$x_{\text{opt}} = [-2.9272, -3.0844, 1.9252, 1.0065, 1.1244, 0.0977, -1.9152, -2.1059].$$

Budeme-li optimalizovat reálný poloměr stability, získáme opět stejnou hodnotu jako u komplexního poloměru stability, tj.  $r_{\mathbb{R}}(A + BF_{\text{opt}}) = r_{\mathbb{C}}(A + BF_{\text{opt}})$ . Obrázky 7.11 znázorňují pseudospektra matice dynamiky uzavřeného systému  $A + BF_{\text{opt}}$ .



Obrázek 7.11: Komplexní a reálné pseudospektrum matice  $A + BF_{\text{opt}}$ , kde  $F_{\text{opt}}$  je matice (7.5).

## 8 Závěr

Práce pojednává o problému robustního přiřazení Jordanovy formy stavovou a výstupní zpětnou vazbou. Při výběru nejlepší matice zpětné vazby, která systému přiřazuje Jordanovu formu, využíváme optimalizace zejména komplexního a reálného poloměru stability.

Jednou z výhod našeho přístupu a zároveň z přínosů práce je využití minimální parametrizace všech zpětných vazeb přiřazujících požadovanou Jordanovu formu. Optimalizace se tak nemusí provádět vzhledem ke všem prvkům matice zpětné vazby, ale vystačíme s mnohem méně parametry, což je při následné numerické optimalizaci velice výhodné. Dále kromě komplexního poloměru stability, který je uvažován také v práci HIFOO, přidáváme další kritériální funkce. Z praktických důvodů používáme též reálný poloměr stability a také umožňujeme navrhovat regulátory s omezenou strukturou. V případě neúplného přiřazení, kdy přiřazujeme jen některá vlastní čísla v Jordanově blocích, bude poloha ostatních vlastních čísel záviset na dalších volných parametrech. Poněvadž nemáme zaručeno, že se budou nacházet v otevřené polovině levé komplexní roviny, zařazujeme dodatečně požadavek na stabilitu uzavřeného systému. Hlavní nové výsledky této práce jsou:

- Nový algoritmus pro výpočet parametrické matice  $Q(\alpha)$ , ve kterém si při konstrukci její struktury vystačíme se znalostí geometrické násobnosti vlastních čísel a velikostí jednotlivých bloků, a nemusíme přímo vytvářet pomocnou zaměnitelnou matici.
- Sestavení algoritmu pro nalezení stavové zpětné vazby, která přiřazuje systému požadovanou Jordanovu formu a představuje lokálně optimální stavovou zpětnou vazbu na základě vybraných kritérií robustnosti.
- Nový numerický algoritmus pro výpočet lokálně optimální výstupní zpětné vazby podle výše uvedených kritérií. Navržený algoritmus nalezne lokální minimum a lze jej pro naše účely považovat za dostatečný. Doba potřebná k výpočtu je však v některých případech velká. V další práci se pokusíme navržený algoritmus upravit za účelem snížení jeho časové náročnosti. Rovněž bychom mohli vylepšit volbu velikosti kroku.
- Nástroj v programovém systému Matlab pro robustní přiřazení Jordanovy formy stavovou a výstupní zpětnou vazbou, ve kterém jsou implementovány výše uváděné algoritmy. Součástí implementace je také uživatelské rozhraní pro příjemnější ovládání vytvořených programů.

Navržené algoritmy byly testovány na řadě příkladů, včetně reálných úloh. Pomocí těchto algoritmů se nám podařilo například stabilizovat praktický příklad modelu robotu jednožky.

# Literatura

- [1] BENNER, Peter; MEHRMANN, Volker; XU, Hongguo. A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils. *Numerische Mathematik*, 1998, 78.3: 329-358.
- [2] BENNER, Peter; KRESSNER, Daniel. Algorithm 854: Fortran 77 subroutines for computing the eigenvalues of Hamiltonian matrices II. *ACM Transactions on Mathematical Software (TOMS)*, 2006, 32.2: 352-373.
- [3] BENNER, Peter; VOIGT, Matthias. Numerical computation of structured complex stability radii of large-scale matrices and pencils. In: *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012. p. 6560-6565.
- [4] BOYD, Stephen; BALAKRISHNAN, Venkataramanan. A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its  $L_\infty$ -norm. *Systems & Control Letters*, 1990, 15.1: 1-7.
- [5] BOYD, S.; BALAKRISHNAN, V.; KABAMBA, P. A bisection method for computing the  $H_\infty$  norm of a transfer matrix and related problems. *Mathematics of Control, Signals and Systems*, 1989, 2.3: 207-219.
- [6] BURKE, James V.; LEWIS, Adrian S.; OVERTON, Michael L. A nonsmooth, nonconvex optimization approach to robust stabilization by static output feedback and low-order controllers. In: *Proc. IFAC Symp. Robust Control Design, Milan, Italy*. 2003.
- [7] BURKE, J. V., et al. HIFOO – A MATLAB package for fixed-order controller design and  $H_\infty$  optimization. In: *Fifth IFAC Symposium on Robust Control Design, Toulouse*. 2006.
- [8] BYERS, Ralph. A bisection method for measuring the distance of a stable matrix to the unstable matrices. *SIAM Journal on Scientific and Statistical Computing*, 1988, 9.5: 875-881.
- [9] ČERMÁK, Libor. *Vybrané statě z numerických metod*. Brno, 2012.

- [10] DOSTÁL, Zdeněk; BEREMLIJSKI, Petr. *Metody optimalizace*. Plzeň, 2011.
- [11] GOLUB, Gene H.; VAN LOAN, Charles F. *Matrix computations*. ISBN: 0-8018-5414-8, JHUP, 1996.
- [12] GRASMAIR, M.; *Continuous Optimisation*, Lecture Notes, Computational Science Center, University of Vienna, 2011.
- [13] CHEN, S.; WU, J.; LI, G. Two approaches based on pole sensitivity and stability radius measures for finite precision digital controller realizations, *Systems & Letters*, 2002, p. 321-329.
- [14] KAROW, Michael; KOKIOPOULOU, Effrosyni; KRESSNER, Daniel. On the computation of structured singular values and pseudospectra. *Systems & Control Letters*, 2010, 59.2: 122-129.
- [15] KILIC, Mustafa; MENGI, Emre; YILDIRIM, Emre Alper. Numerical optimization of eigenvalues of hermitian matrix functions. *arXiv preprint arXiv:1109.2080*, 2011. <http://home.ku.edu.tr/~emengi/software/eigopt.html>
- [16] KÖNIGSMARKOVÁ, Jana. *Modální řízení konečněrozměrných lineárních systémů*. Plzeň, 2010. Bakalářská práce. ZČU.
- [17] LAWRENCE, Craig T.; TITS, André L.; DOOREN, Paul Van. A fast algorithm for the computation of an upper bound on the  $\mu$ -norm. *Automatica*, 2000, 36.3: 449-456.
- [18] MADSEN, K.; NIELSEN, H. B.; TINGLEFF, O. Methods for non-linear least squares problems, *Informatics and Mathematical Modelling*, Technical University of Denmark, 2004.
- [19] MAMMADOV, M.; ORSI, R.  $H_\infty$  Synthesis via a Nonsmooth, Nonconvex Optimization Approach, *Pacific Journal of Optimization*, vol. 1, 2005, p. 405-420.
- [20] MILLSTONE, Marc A. HIFOO 1.5: *Structured control of linear systems with non-trivial feedthrough*. 2006. PhD Thesis. New York University.
- [21] OVERTON, M. L. *Optimization of Polynomial Roots, Eigenvalues and Pseudospectra*, Courant Institute of Mathematical Sciences, NYU, CalTech, Halloween, 2011.
- [22] QIU, Li, et al. A formula for computation of the real stability radius. *Automatica*, 1995, 31.6: 879-890.
- [23] SCHLEGEL, Miloš. *Modální řízení konečněrozměrných lineárních systémů*. Plzeň, 1984. Disertační práce, Vysoká škola strojní a elektrotechnická.



- [24] SCHLEGEL, Miloš. *Mechanické systémy*. Plzeň, 2007. Pomocný učební text. ZČU.
- [25] SCHLEGEL, Miloš; KÖNIGSMARKOVÁ, Jana. Parametric Jordan Form Assignment Revisited, *Asian Journal of Control*, vol. 16, no. 1, pp.1-12, doi: 10.1002/asjc.713, January 2014.
- [26] SREEDHAR, J.; VAN DOOREN, Paul; TITS, André L. A fast algorithm to compute the real structured stability radius. In: *Stability Theory*. Birkhäuser Basel, 1996. p. 219-230.
- [27] STEWART, G. W; SUN, J. *Matrix perturbation theory*, New York, Academic Press, 1990.
- [28] VAN LOAN, Charles. *How near is a stable matrix to an unstable matrix?*. Cornell University, 1984.
- [29] VONDRÁK, Vít; POSPÍŠIL, Lukáš. *Numerické metody I*. Vysoká škola báňská – Technická univerzita Ostrava a Západočeská univerzita v Plzni, 2011.
- [30] WATKINS, David S. *Fundamentals of matrix computations*. Wiley-Interscience, New York, 2002. Second edition.
- [31] WRIGHT T.G. EigTool: a graphical tool for nonsymmetric eigenproblems, 2002. Oxford University Computer Laboratory. <http://web.comlab.ox.ac.uk/pseudospectra/eigtool/>
- [32] YANG, Y.; TITS, A. L.; QIU, L. On the Computation of the Real Stability Radius. 1993.
- [33] YUAN, Y. X. Recent advances in numerical methods for nonlinear equations and nonlinear least squares, Numerical algebra, *Control and Optimization*, vol. 1, number 1, 2011, pp. 15-34.