

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií

DIPLOMOVÁ PRÁCE

Liberec 2004

Martin Plešinger



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií

DIPLOMOVÁ PRÁCE

Vybrané problémy numerické lineární algebry
v teorii řízení

Selected Numerical Linear Algebra Problems
in Control Theory

Liberec 2004

Martin Plešinger

[Zadání] – to je ten papír, co podepsal Šembera s Maryškou

Anotace

Základním cílem diplomové práce bylo podat jednotný výklad základů teorie dynamických systémů a teorie řízení spolu s úlohami lineární algebry, které se v těchto oborech nejčastěji vyskytují – řešení soustav lineárních rovnic a problémy vlastních čísel. Důraz byl kladen na řešení velmi rozsáhlých úloh a na sledování potřebných vlastností s tím souvisejících metod. Seznámili jsme se s problémy, které nastávají při implementaci metod v počítači. Pokusili jsme se zde na příkladu výpočtu úlohy z teorie řízení (výpočet říditelnosti systému) poukázat na nutnost studia numerických vlastností algoritmů zejména u rozsáhlých úloh.

V druhé části diplomové práce jsme popsali další příklady z teorie řízení, které vedou na rozsáhlé algebraické problémy. První z těchto příkladů byl výpočet vlastních kmitů piezoelektrického rezonátoru, převedený na problém nalezení nul konkrétního dynamického systému. Druhým příkladem byl problém řízení tzv. „inteligentního materiálu“ realizovaného konkrétní strukturou (založeno taktéž na principu piezoelektrického jevu). Tento příklad v podstatě vede na nalezení optimálního (a fyzikálně realizovatelného) řízení. Ve třetím příkladu jsme se pokusili popsat obecnou lineární stacionární iterační metodu na řešení soustav lineárních rovnic jako systém s regulátorem a regulovanou soustavou. Tento příklad opět může vést na problém nalezení optimálního řízení.

Abstract

In this diploma project we wish to describe in a uniform way some basic theory of dynamic systems and control together with linear algebraic problems arising from these areas – solving linear algebraic systems and eigenvalue problems. We have emphasized solving very large problems and focused on properties of corresponding large-scale computational methods. We have researched numerical problems of computer implementations of various methods. We have attempted to present an example of the control theory problem (computing of system controllability), which illustrates necessity of analyzing numerical properties of algorithms in particular when solving large-scale problems.

In the second part we have described some other control problems giving rise to large-scale algebraic computations. In the first example eigenfrequencies of a piezoelectric resonator have been studied as zeros of some particular dynamic system. In the second example we have described the principle of control of so called “smart material structure”, based again on some piezoelectric properties. In the third example we have presented a general stationary iterative method for solving of linear algebraic system as a dynamical system. Here again the algebraic problem is related to optimal control.

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum

Podpis

OBSAH

Úvod	3
1. Základy teorie dynamických systémů a teorie řízení	5
1.1 Lineární dynamický systém	5
1.2 Vnitřní a vnější popis dynamického systému, přenos dynamického systému	6
1.3 Charakteristiky dyn. systému – další způsoby popisu systému	10
1.3.1 Přejchodová a impulsní charakteristika	10
1.3.2 Frekvenční charakteristiky systému	12
1.4 Póly a nuly systému	13
1.4.1 Přenosové póly a nuly systému a kanonický tvar přenosové matice	13
1.4.2 Určení kanonického tvaru matice pomocí minorů	14
1.4.3 Různé typy nul systému	15
1.4.4 Vztahy mezi různými typy nul	17
1.5 Základní pojmy z teorie řízení a vlastnosti systémů (-abilities)	18
1.5.1 Řiditelnost a dosažitelnost v systému (Controllability)	18
1.5.2 Pozorovatelnost a rekonstruovatelnost systému (Observability)	21
1.5.3 Poznámky k dosažitelnosti a pozorovatelnosti	22
1.5.4 Stabilita, stabilizovatelnost a detekovatelnost systému	24
2. Nástroje lineární algebry používané v teorii řízení	26
2.1 Soustavy lineárních rovnic	26
2.1.1 Soustavy lineárních rovnic v atypickém tvaru	27
2.1.2 Přímé metody řešení soustavy rovnic	29
2.1.3 Stacionární iterační metody řešení soustavy rovnic	31
2.1.4 Nestacionární iterační metody, projekční metody	33
2.2 Problém nejmenších čtverců	37
2.2.1 Existence řešení a jednoznačnost řešení úlohy	37
2.2.2 Řešení úlohy ve smyslu nejmenších čtverců a řešení minimální v normě	38
2.2.3 Metody řešení úloh nejmenších čtverců	38
2.2.4 QR rozklad	40
2.3 Problém vlastních čísel	42
2.3.1 Problém vlastních čísel	42
2.3.2 Mocninná metoda hledání vlastních čísel	42
2.3.3 Úplný problém vlastních čísel – QR algoritmus	44
2.3.4 Zobecněný problém vlastních čísel	45
2.4 Singulární rozklad	46
2.4.1 Singulární rozklad – singulární čísla	46
2.4.2 Ekonomická forma singulárního rozkladu	46
2.4.3 Vztah singulárních a vlastních čísel	47

3. Problémy s implementací – numerická lineární algebra	48
3.1 Reálná čísla v počítači	48
3.2 Relativní strojová přesnost	50
3.3 Numerická stabilita a podmíněnost	51
3.3.1 Příklad na začátek – Gaussova eliminace	51
3.3.2 Podmíněnost úlohy (Condition)	52
3.3.3 Podmíněnost matice	52
3.3.4 Příklad druhý – podmíněnost matice	53
3.3.5 Numerická stabilita algoritmu – přímá / zpětná stabilita	54
3.4 Nevhodné algoritmy v teorii řízení – příklad výpočtu říditelnosti systému	55
4. Příklady velkých úloh v teorii DS a teorii řízení	58
4.1 Stručný úvod do fyziky piezoelektrických materiálů	58
4.2 Příklad první – piezoelektrický rezonátor jako DS	60
4.3 Příklad druhý – aktivní piezoelektrické prvky	63
4.4 Příklad třetí – iterační metody jako DS se zpětnovazebním řízením	68
5. Závěr	71
LITERATURA	73

Úvod

V posledních desetiletích pokročil vývoj v oblasti výpočetní techniky výrazně kupředu. Tento pokrok umožnil provádět nejrůznější i velmi složité inženýrské výpočty, vedoucí často na algebraické úlohy se stovkami tisíc až miliony neznámých. Ač se jedná o nejrůznější úlohy (obecně lineární i nelineární), jádrem výpočetních algoritmů pro řešení většiny numerických úloh je řešení soustav lineárních rovnic a výpočet vlastních čísel matice.

Většina úloh, řešených v nejrůznějších inženýrských oborech, vede přímo na tyto úlohy lineární algebry. Stejně tak tomu je i v teorii dynamických systémů a teorii řízení, kterými se budeme podrobněji zabývat. Otevřeme-li učební text zabývající se touto problematikou, dříve či později na soustavy lineárních rovnic i na problém vlastních čísel narazíme. Nalezneme zde i podrobná vysvětlení, jakými cestami se k těmto problémům dospělo. Rovněž se často setkáme s různými příklady, jak takové úlohy řešit. Tyto ukázkové příklady bývají obvykle jednoduché, aby byly snadno pochopitelné a rychle řešitelné. Typická učebnicová úloha vede na soustavu několika lineárních rovnic s odpovídajícím počtem neznámých. Takovou soustavu pak lze snadno vyřešit například Gaussovou eliminací či jiným standardním konečným algoritmem. V učebnicích bývá pochopitelně ukázáno, že daný problém vede zcela obecně na soustavu lineárních rovnic, ovšem jak tuto soustavu řešit, sestává-li z velkého počtu rovnic, tam již většinou uvedeno není.

U takto rozsáhlých úloh (řádu desítek tisíc, stovek tisíc či miliónů) ovšem narážíme na zcela nový a nezanedbatelný problém. Obecně známým faktem je, že počítač není schopen pracovat s libovolnou numerickou přesností. Není schopen pracovat s reálnými čísly, a tak jejich množinu výrazně redukuje – používá pouze čísla patřící do jisté konečné podmnožiny čísel racionálních (obvykle říkáme, že počítač pracuje v aritmetice s konečnou přesností). V průběhu jakýchkoliv výpočtů tak dochází k nepřesnostem jednak při předávání vstupních dat počítači a jednak při vlastním výpočtu. Dochází k vzniku zaokrouhlovacích chyb. V průběhu numerického řešení se zaokrouhlovací chyby mohou kumulovat. Chceme-li řešit nějakou rozsáhlou úlohu lineární algebry, musíme být schopni ji efektivně algoritmizovat a výpočetní postup implementovat na daném počítači s plným vědomím nových nepřesností! K šíření zaokrouhlovacích chyb v průběhu výpočtu dochází v jakémkoliv algoritmu. Kvalitní algoritmy jsou ovšem schopny šíření zaokrouhlovacích chyb omezit. V ideálním případě je pak výsledek výpočtu zatížen chybou jen na úrovni základního zaokrouhlování (strojové přesnosti). U nestabilních algoritmů či implementací dochází ke kumulování zaokrouhlovacích chyb. Výsledek může být zatížen tak velkou chybou, že je zcela nepoužitelný. Analýzou stability jednotlivých algoritmů a citlivostí různých typů úloh lze ukázat, jak je výsledek výpočtu přesný. Tyto nástroje ve vzájemné souvislosti s analýzou konvergence iteračních metod tak hrají naprosto zásadní roli při návrhu algoritmů pro velmi rozsáhlé úlohy.

Bohužel v současné době není běžně k dispozici mnoho učebnic, které by například právě v kontextu teorie řízení poskytovaly uspokojivý návod, jak takové problémy řešit z pohledu numerické matematiky. Problém je často i v tom, že jazyk inženýrů a matematiků se mnohdy liší. Řešení praktického problému se často omezí jen na převedení daného problému do podoby matematických rovnic a na jeho řešení praktickým postupem bez hlubší analýzy všech numerických souvislostí! Stejně tak matematik často omezí svoji práci pouze na řešení striktně formulované matematické úlohy bez hlubší souvislosti s reálným problémem. A stejně k tomu může přistupovat i programátor, který se pak nezávisle na předchozích poznatcích snaží matematikem navrženou numerickou metodu

implementovat. Všechny tyto fáze řešení nějaké úlohy jsou bezesporu velmi důležité. Nesporným faktem je ovšem také to, že vyznáme-li se jako matematici v tom, co nám dal k dispozici technik nebo inženýr – řešitel daného problému, a známe-li strukturu reálného problému, můžeme s ohledem na to navrhnout postup řešení úlohy, který může být do jisté míry efektivnější. Jsme-li navíc schopni tuto metodu implementovat, může řešení celého problému vést na ještě lepší výsledky. Je zřejmé, že u rozsáhlých projektů není v silách jednoho člověka toto všechno obsáhnout. Důležitá je v tomto směru dobrá komunikace a spolupráce mezi jednotlivými členy možného řešitelského kolektivu a alespoň základní vhlad do práce ostatních řešitelů. To je ovšem podmíněno právě potřebou nalezení společného zájmu, motivace a jazyka.

V tomto textu se pokusíme popsat jednotícím způsobem vybrané základní úlohy, vyskytující se právě v teorii dynamických systémů, v teorii řízení a v matematice (zejména v numerické lineární algebře). Rovněž se pokusíme poukázat na důležitost analýzy numerické stability. Poté se budeme zabývat konkrétními příklady rozsáhlých úloh teorie řízení, například hledáním vlastních frekvencí piezoelektrického rezonátoru. Model piezoelektrického rezonátoru realizujeme pomocí metody konečných prvků – FEM. K tomuto modelu budeme přistupovat jako ke klasickému dynamickému systému, jak jej známe z teorie dynamických systémů. Vlastní frekvence rezonátoru budeme hledat jako nuly příslušného dynamického systému. Dále se stručně budeme věnovat řízení inteligentních materiálů (struktur) založených opět na bázi piezoelektrických materiálů. Jako dynamický systém také popíšeme obecnou lineární stacionární iterační metodu pro řešení soustav lineárních rovnic. V závěru se zmíníme o problémech při řešení Ljapunovových rovnic (speciální tvar soustavy lineárních rovnic, který se v teorii řízení často vyskytuje), jejichž studium by se mohlo stát východiskem pro další práci.

1. Základy teorie dynamických systémů a teorie řízení

1.1 Lineární dynamický systém

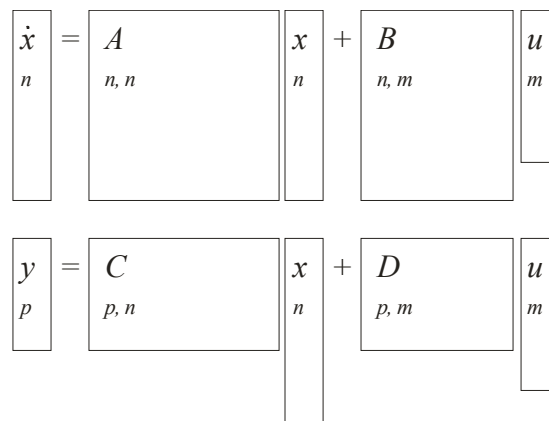
Dynamické systémy je možné třídit podle různých hledisek. Klasicky je možné systémy rozlišovat například na systémy typu *Single Input Single Output (SISO)* a *Multiple Input Multiple Output (MIMO)*. Nyní se budeme zabývat téměř výhradně systémy typu MIMO, což nebudeme dále zdůrazňovat (tyto systémy jsou obecnější z hlediska popisu). Další rozdělení, z našeho pohledu důležitější, je rozdělení na systémy spojité a nespojitě, respektive na *systémy se spojitým popisem* a na *systémy popsané diskretním způsobem*. V podstatě všechny důsledky, o kterých zde budeme dále hovořit z pohledu teorie dynamických systémů a teorie řízení, lze odvodit jak pro systémy spojité, tak pro systémy diskretní. My se budeme opět pro jednoduchost zabývat téměř výhradně systémy spojitými. A asi nejpodstatnější rozdělení je rozdělení na systémy lineární a na systémy nelineární.

Zde se budeme výhradně zabývat systémy lineárními, které jsou popsány standardním lineárním modelem [1.2; kap. 3.2; (3.1)]

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (1.1)$$

$$y(t) = Cx(t) + Du(t), \quad (1.2)$$

kde $x(t)$ je n -rozměrný *stavový vektor* (vektor vnitřních stavů systému), $u(t)$ je m -rozměrný *vstupní vektor* a $y(t)$ je p -rozměrný *výstupní vektor* systému. Matice A , B , C a D jsou obecně komplexní matice příslušných rozměrů, viz obrázek 1.1.



$$\begin{array}{c} \dot{x} \\ n \end{array} = \begin{array}{c} A \\ n, n \end{array} \begin{array}{c} x \\ n \end{array} + \begin{array}{c} B \\ n, m \end{array} \begin{array}{c} u \\ m \end{array}$$

$$\begin{array}{c} y \\ p \end{array} = \begin{array}{c} C \\ p, n \end{array} \begin{array}{c} x \\ n \end{array} + \begin{array}{c} D \\ p, m \end{array} \begin{array}{c} u \\ m \end{array}$$

Obrázek 1.1; Schematické znázornění rovnic (1.1) a (1.2).

Analogický standardní popis diskretního systému je [1.2; kap. 3.2; (3.8)]

$$x_{k+1} = (hA + I)x_k + (hB)u_k = \tilde{A}x_k + \tilde{B}u_k, \quad (1.1a)$$

$$y_k = Cx_k + Du_k, \quad (1.2a)$$

kde h je krok časové diskretizace a I je jednotková matice.

*Spojitém lineárním dynamickým systémem (LDS) nazveme dvojici rovnic (1.1), (1.2), jež je určena uspořádanou čtveřicí matic (A, B, C, D) . Matice A, B, C a D nejsou funkcemi času, pracujeme tedy se *stacionárním LDS*.*

Existuje řada složitějších modelů dynamických systémů, kterými se ovšem zabývat nebudeme, neboť výše popsaný systém je pro naše potřeby dostatečně obecný. Složitějším modelem je například *nestacionární LDS*, tedy model, v kterém jsou matice A, B, C a D obecné funkce času. Systém tak zůstane lineárním vůči vektoru vstupu a stavovému vektoru, výpočet se však výrazně zkomplikuje. Dále je možné do modelu zavést přímou závislost výstupního vektoru na derivaci (obecně libovolné derivaci) vstupního vektoru; jedná se tedy o jisté zobecnění matice D , viz [1.1; (1.0, 1)] nebo [1.2; kap. 3.6; př. 5]. Nejobecnějším popisem (ovšem již nelineárního dynamického systému) je obecná stavová rovnice dávající do souvislosti čas, vstupní vektor, stavový vektor, výstupní vektor a jejich derivace.

Jediným ze složitějších popisů, který zde uvedeme, bude popis tzv. *zobecněného* respektive *singulárního lineárního dynamického systému*. Tento popis systému nás zde bude zajímat zejména proto, že úloha hledání pólů a nul systému s tímto popisem vede na tzv. *zobecněný problém vlastních čísel*. Singulární lineární dynamický systém je popsán rovnicemi [1.2; kap. 3.5; (3.38)]

$$E\dot{x}(t) = Ax(t) + Bu(t), \quad (1.3)$$

$$y(t) = Cx(t) + Du(t). \quad (1.4)$$

Je zřejmé, že je-li matice E regulární, lze rovnici (1.3) snadno převést na rovnici (1.1) a tedy i celý popis dynamického systému lze vyjádřit standardním způsobem. Když je ovšem matice E singulární, systém popsat standardním způsobem nelze. Proto jsou také tyto systémy nazývány *singulární*, někdy též *implicitní* nebo *zobecněné* [1.2; kap. 3.5].

1.2 Vnitřní a vnější popis dynamického systému, přenos dynamického systému

Dynamický systém (1.1), (1.2) lze popsat mnoha různými způsoby. Tzv. *vnitřní popis systému* vyjadřuje relaci typu

$$\{\text{vstup} \rightarrow \text{stavový vektor} \rightarrow \text{výstup}\},$$

ukázkou vnitřního popisu je tedy i samotná soustava tzv. *stavových rovnic* (1.1) a (1.2). *Vnější popis systému* vyjadřuje pouze relaci typu

$$\{\text{vstup} \rightarrow \text{výstup}\}.$$

Vnější popis je tedy svým způsobem neúplný, neboť zde postrádáme informaci o stavu systému, kterou nelze obecně odvodit pouze ze znalosti vstupů a výstupů [1.2; kap. 3.3].

Při analýze stacionárních LDS lze s výhodou používat Laplaceových nebo Fourierových obrazů výše uvedené soustavy rovnic (1.1), (1.2). Často tedy budeme používat integrální transformace, zejména Laplaceovu a Fourierovu [1.3; kap. 28], [1.4]. Laplaceův obraz soustavy (1.1), (1.2) je následující [1.2; kap. 3.2; (3.3)]

$$sX(s) - x(0) = AX(s) + BU(s), \quad (1.5)$$

$$Y(s) = CX(s) + DU(s), \quad (1.6)$$

kde vektory $X(s)$, $U(s)$ a $Y(s)$ jsou Laplaceovy obrazy vektorů $x(t)$, $u(t)$ a $y(t)$ a vektor $x(0)$ odpovídá vektoru $x(t)$ v čase $t = 0$. Rovnice (1.5), (1.6) můžeme přepsat v maticové formě následujícím způsobem

$$\begin{bmatrix} sI - A & -B \\ C & D \end{bmatrix} \begin{bmatrix} X(s) \\ U(s) \end{bmatrix} \equiv P(s) \begin{bmatrix} X(s) \\ U(s) \end{bmatrix} = \begin{bmatrix} x(0) \\ Y(s) \end{bmatrix}. \quad (1.7)$$

Matice

$$P(s) = \begin{bmatrix} sI - A & -B \\ C & D \end{bmatrix}. \quad (1.8)$$

je nazývána *maticí systému (system matrix)*. Matice (1.8) je dalším příkladem vnitřního popisu systému, neboť obsahuje všechny informace o stavových rovnicích systému. Pokud z rovnice (1.5) vyjádříme vektor $X(s)$ a dosadíme ho do rovnice (1.6), dostaneme po úpravách následující vztah

$$Y(s) = [C(sI - A)^{-1}B + D]U(s) + [C(sI - A)^{-1}]x(0), \quad (1.9)$$

popisující systém (1.1), (1.2). Budeme-li uvažovat nulové počáteční podmínky (tedy $x(0) = 0$), vztah (1.9) se výrazně zjednoduší

$$Y(s) = G(s)U(s), \quad (1.10)$$

kde matice $G(s)$ je *Schurův doplněk podmatice $(sI - A)$* v matici $P(s)$, viz [2.3; (1.23)], tedy

$$G(s) \equiv C(sI - A)^{-1}B + D. \quad (1.11)$$

Matice $G(s)$ je nazývána *přenosovou maticí systému (transfer-function matrix)*. Přenosem rozumíme podíl Laplaceova obrazu výstupu ku Laplaceovu obrazu vstupu při nulových počátečních podmínkách. Libovolný prvek matice (1.11) je tedy podíl Laplaceových obrazů odpovídajícího výstupu ku odpovídajícímu vstupu za nulových počátečních podmínek. Funkce $G_{ij}(s)$, tedy ij -tý prvek přenosové matice (1.11), vyjadřuje přenos j -tého vstupu na i -tý výstup. Popis systému rovnicí (1.9), respektive maticí (1.11), je vnějším popisem systému.

Ze vztahu (1.11) lze ukázat, že prvky matice $G(s)$ jsou racionálně lomené funkce v proměnné s . Prvky G_{ij} jsou tvořeny podílem dvou polynomů v proměnné s ,

$$G_{ij}(s) = \frac{p_{ij;k}s^k + p_{ij;k-1}s^{k-1} + \dots + p_{ij;1}s + p_{ij;0}}{q_{ij;l}s^l + q_{ij;l-1}s^{l-1} + \dots + q_{ij;1}s + q_{ij;0}} \equiv \frac{P_{ij;k}(s)}{Q_{ij;l}(s)}.$$

Polynom čitatele $P_{ij;k}$ je polynom k -tého stupně, polynom jmenovatele $Q_{ij;l}$ je polynom l -tého stupně (poslední index vyjadřuje stupeň polynomu). Lze ukázat, že k i l jsou maximálně rovny n , přičemž n je počet řádků (sloupců) matice A (rozměr stavového vektoru $x(t)$ z rovnice (1.1)). Dále lze ukázat, že je-li matice D nulová, je číslo k maximálně rovno $n-1$. Systém s nulovou maticí D nazýváme *striktně ryzím systémem (proper system)*. Teoretické odvození tvarů polynomů čitatele a jmenovatele funkcí G_{ij} ze vztahu (1.11) je možné například takto:

- (i) Ze vztahu (1.11) vyjádříme inverzní matici $(sI-A)^{-1}$. Nejprve definujeme matici R ,

$$R \equiv (sI - A) = \begin{pmatrix} s - a_{11} & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & s - a_{22} & \cdots & -a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & s - a_{nn} \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{pmatrix}.$$

Vyjádříme inverzní matici k matici R [1.3; kap. 1.25; věta 6]

$$R^{-1} = \frac{1}{\det(R)} \begin{pmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{21} & \rho_{22} & \cdots & \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1} & \rho_{n2} & \cdots & \rho_{nn} \end{pmatrix},$$

kde ρ_{ij} jsou doplňky prvku r_{ij} [1.3; kap. 1.17; definice 2],

$$\rho_{ij} = (-1)^{i+j} \det(R_{ij}).$$

Matice R_{ij} vznikne z matice R vynecháním i -tého řádku a j -tého sloupce (je to tedy matice stupně $n-1$). Doplněk ρ_{ij} je determinant matice R_{ij} vynásobený číslem $(-1)^{i+j}$. Prvky inverzní matice R^{-1} vyjádříme

$$\left[R^{-1} \right]_{ij} = \left[(sI - A)^{-1} \right]_{ij} = (-1)^{i+j} \frac{\det(R_{ij})}{\det(R)}.$$

Determinant matice R je polynom stupně n v proměnné s a determinant matice R_{ij} je polynom stupně $n-1$ v proměnné s . Prvky této inverzní matice jsou tedy racionálně lomené funkce; podíly těchto dvou polynomů.

- (ii) Ze vztahu (1.11) dále vidíme, že inverzní matici $(sI-A)^{-1}$ násobíme zleva maticí C a zprava maticí B . Matice B ani matice C nejsou závislé na proměnné s ; obě jsou tvořeny pouze čísly. Z násobení matic je zřejmé, že prvky matice $C(sI-A)^{-1}B$ jsou lineární kombinace prvků matice $(sI-A)^{-1}$ s koeficienty tvořenými prvky matic B, C . Prvky matice $C(sI-A)^{-1}B$ jsou tedy také racionálně lomené funkce. Maximální stupeň polynomu čitatele $n-1$ a polynomu jmenovatele n se maticovým násobením nijak nezmění.

- (iii) Prvky matice $C(sI-A)^{-1}B+D$ (tedy prvky matice $G(s)$; viz (1.11)) můžeme zapsat tedy takto

$$G_{ij}(s) = \frac{\xi_{ij;n-1}s^{n-1} + \xi_{ij;n-2}s^{n-2} + \dots + \xi_{ij;1}s + \xi_{ij;0}}{\zeta_{ij;n}s^n + \zeta_{ij;n-1}s^{n-1} + \dots + \zeta_{ij;1}s + \zeta_{ij;0}} + d_{ij} \equiv \frac{\Xi_{ij;n-1}(s)}{Z_{ij;n}(s)} + d_{ij},$$

kde d_{ij} je příslušný prvek matice D . Provedeme-li součet převedením na společného jmenovatele, zapíšeme prvky G_{ij} jako

$$\begin{aligned} G_{ij}(s) &= \frac{d_{ij}\zeta_{ij;n}s^n + (d_{ij}\zeta_{ij;n-1} + \xi_{ij;n-1})s^{n-1} + \dots + (d_{ij}\zeta_{ij;1} + \xi_{ij;1})s + (d_{ij}\zeta_{ij;0} + \xi_{ij;0})}{\zeta_{ij;n}s^n + \zeta_{ij;n-1}s^{n-1} + \dots + \zeta_{ij;1}s + \zeta_{ij;0}} = \\ &= \frac{\Xi_{ij;n-1}(s) + d_{ij}Z_{ij;n}(s)}{Z_{ij;n}(s)} = \frac{P_{ij;n}(s)}{Q_{ij;n}(s)}. \end{aligned}$$

Je vidět, že prvky G_{ij} jsou racionálně lomené funkce. Dále je vidět, jak nulovost matice D ovlivňuje maximální stupeň polynomu čitatele v těchto funkcích.

Zřejmě lze nalézt určitý polynom $d(s)$, pomocí kterého upravíme matici $G(s)$ následujícím způsobem

$$G(s) = \frac{1}{d(s)} F(s),$$

kde $F(s)$ je matice stejných rozměrů jako matice $G(s)$, ale je tvořena pouze polynomy. *Nejjednodušší jmenovatel matice $G(s)$* je polynom $d(s)$ s nejmenším stupněm, pro který toto platí; uvažujeme pouze *monické (normalizované) polynomy*, tedy polynomy s jednotkovým koeficientem u nejvyšší mocniny. Nejjednodušší jmenovatel je roven *nejmenšímu společnému násobku polynomů $Q_{ij}(s)$* . Lze ukázat, že nejjednodušší jmenovatel je polynom maximálně n -tého stupně a je-li polynomem n -tého stupně, je dán vztahem

$$d(s) = \det(sI - A).$$

Nejjednodušší jmenovatel $d(s)$ je tedy maximálně roven *charakteristickému polynomu systému $\det(sI-A)$* , respektive *charakteristickému polynomu matice A* (slovy lineární algebry). Je-li $\det(sI-A)$ nejjednodušším jmenovatelem, lze dále ukázat, že prvky $F_{ij}(s)$ matice $F(s)$ jsou polynomy maximálně n -tého stupně; v případě systému (1.1), (1.2) s nulovou maticí D maximálně stupně $n-1$.

V odstavci 1.4 ukážeme, že matice $G(s)$ a polynom $d(s)$ mají stěžejní význam při výpočtu tzv. pólů systému. Kořeny nejjednoduššího jmenovatele $d(s)$ jsou tzv. přenosové póly. Kořeny charakteristického polynomu jsou tzv. systémové póly (slovy teorie systémů), respektive vlastní čísla matice A (slovy lineární algebry, viz kapitolu 2.3).

Přenosovou matici je možné vyjádřit i u singulárního systému (1.3), (1.4),

$$G(s) \equiv C(sE - A)^{-1}B + D. \quad (1.12)$$

Když porovnáme matice (1.11) a (1.12) zjistíme, že byla jednotková matice I v (1.11) nahrazena singulární maticí E v (1.12). Charakteristický polynom singulárního systému bude odpovídat zobecněnému charakteristickému polynomu matice A . Nalezení systémových pólů singulárního LDS tak bude ekvivalentní zobecněnému problému vlastních čísel matice A (viz kapitolu 2.3.4).

U diskretních systémů (1.1a), (1.2a) lze vyjádřit matice $P(z)$ a $G(z)$, analogie matic $P(s)$ a $G(s)$. Postupujeme stejně jako u systému spojitého, pouze spojitou Laplaceovu transformaci nahradíme diskretní z -transformací [1.4; kap 15]. Všechny výrazy i obě matice vypadají v podstatě formálně stejně. Obraz soustavy rovnic (analogie rovnic (1.5), (1.6)) je

$$zX(z) - zx(0) = \tilde{A}X(z) + \tilde{B}U(z), \quad (1.5a)$$

$$Y(z) = CX(z) + DU(z). \quad (1.6a)$$

Matice $P(z)$ a $G(z)$ (analogie matic (1.8) a (1.11)) jsou

$$P(z) \equiv \begin{bmatrix} zI - \tilde{A} & -\tilde{B} \\ C & D \end{bmatrix}, \quad (1.8a)$$

$$G(z) \equiv C(zI - \tilde{A})^{-1} \tilde{B} + D; \quad (1.11a)$$

viz [1.2; kap. 3.2].

1.3 Charakteristiky dynamického systému – další způsoby popisu systému

1.3.1 Přechodová a impulsní charakteristika

U dynamických systémů se můžeme setkat kromě popisu soustavou rovnic (1.1), (1.2) nebo maticí (1.8), resp. (1.11) s popisem pomocí různých charakteristik či funkcí. Příkladem jsou *přechodová a impulsní (váhová) charakteristika (funkce)*. K jejich zavedení ovšem potřebujeme nejprve zmínit tzv. *Heavisideovu funkci* a tzv. *Diracův impuls*.

Heavisideovu funkci (jednotkovou skokovou funkci) značíme symbolem $\eta(t)$ a je popsána vztahy

$$\eta(t) = 0 \quad \text{pro } t \leq 0,$$

$$\eta(t) = 1 \quad \text{pro } t > 0.$$

Diracův impuls (Diracova distribuce, jednotkový impuls), který není funkcí, ale tzv. zobecněnou funkcí (distribucí) [1.5; kap. 4; str. 229], značíme symbolem $\delta(t)$ a je popsán vztahy

$$\delta(t) = 0 \quad \text{pro } t \neq 0,$$

$$\int_{-\infty}^{+\infty} \delta(t) dt = 1.$$

Mezi Heavisideovou funkcí a Diracovým impulsem je možné formálně (po zobecnění pojmu derivace) zavést vztah (viz [1.5; kap. 4.3 a 4.4; str. 232-237])

$$\delta(t) = \frac{d}{dt}\eta(t).$$

Laplaceovy obrazy Heavisideovy funkce a Diracovy distribuce jsou

$$\begin{aligned}\mathcal{L}\{\eta(t)\} &= \frac{1}{s}, \\ \mathcal{L}\{\delta(t)\} &= 1.\end{aligned}$$

Nyní zavedeme *přechodovou charakteristiku (funkci) systému* jako reakci systému na buzení Heavisideovou funkcí (jednotkovým skokem), resp. m -rozměrným vektorem těchto funkcí při nulových počátečních podmínkách. Přechodovou charakteristiku obvykle značíme $h(t)$ a získáme ji provedením zpětné Laplaceovy transformace [1.2; kap. 2.2.3],

$$u(t) = \eta(t), \tag{1.13}$$

$$h(t) = \mathcal{L}^{-1}\{G(s)U(s)\} = \mathcal{L}^{-1}\left\{G(s)\frac{1}{s}\right\}. \tag{1.14}$$

Dále zavedeme *impulsní (váhovou) charakteristiku (funkci) systému*. Impulsní charakteristiku systému obvykle značíme $g(t)$ a je odezvou systému na buzení *Diracovým impulsem*, respektive m -rozměrným vektorem těchto distribucí při nulových počátečních podmínkách [1.2; kap. 2.2.3],

$$u(t) = \delta(t), \tag{1.15}$$

$$g(t) = \mathcal{L}^{-1}\{G(s)U(s)\} = \mathcal{L}^{-1}\{G(s)\}. \tag{1.16}$$

Laplaceovým obrazem impulsní charakteristiky je tedy přímo přenosová matice systému.

Mezi přechodovou a váhovou charakteristikou systému a mezi jejich Laplaceovými obrazy lze formálně zavést vztahy [1.2; kap. 2.2.3]. Vztahy mezi charakteristikami jsou

$$g(t) = \frac{d}{dt}h(t), \quad \text{respektive} \quad h(t) = \int_0^t g(\tau) d\tau$$

Vztahy mezi Laplaceovými obrazy charakteristik jsou

$$G(s) = sH(s), \quad \text{respektive} \quad H(s) = \frac{1}{s}G(s).$$

1.3.2 Frekvenční charakteristiky systému

Frekvenční charakteristika systému popisuje chování systému při buzení harmonickým signálem na vstupu (pro jednoduchost zápisu uvažujeme systém SISO)

$$u(t) = \sin(\omega t) = \text{Im}\{\exp(i\omega t)\},$$

kde symbol i značí imaginární jednotku a ω značí úhlovou frekvenci harmonického signálu. Abychom mohli vypočítat odezvu systému $y(t)$ s impulsní charakteristikou $g(t)$, stačí najít vzor k Laplaceovu obrazu součinu funkcí $G(s)U(s)$ pro tuto konkrétní volbu funkce $u(t)$ respektive $U(s)$. Budeme-li předpokládat, že odezva systému bude *ustálená*, najdeme řešení relativně jednoduše. Vzorem součinu $G(s)U(s)$ je konvoluce impulsní charakteristiky $g(t)$ a budící funkce $u(t)$; viz [1.2; kap. 2.2.4].

$$\begin{aligned} y(t) &= \int_{-\infty}^{+\infty} g(\tau)u(t-\tau)d\tau = \int_{-\infty}^{+\infty} g(\tau)\text{Im}\{\exp(i\omega(t-\tau))\}d\tau = \\ &= \text{Im}\left\{\exp(i\omega t)\int_{-\infty}^{+\infty} g(\tau)\exp(-i\omega\tau)d\tau\right\} = \text{Im}\{\exp(i\omega t)G(i\omega)\} = \\ &= |G(i\omega)|\sin(\omega t + \arg(G(i\omega))) = \Gamma \sin(\omega t + \varphi). \end{aligned}$$

Jak je vidět, ustálená odezva systému na buzení harmonickým signálem je plně charakterizována komplexní funkcí $G(i\omega)$. U systémů MIMO se $G(i\omega)$ nazývá *maticí frekvenčních přenosů (charakteristik)*. Matici frekvenčních přenosů systému získáme jako hodnotu přenosové matice systému $G(s)$ v bodě $s = i\omega$.

Existují-li navíc *Fourierovy obrazy* obecného vstupního signálu $U(i\omega)$ a příslušného výstupního signálu $Y(i\omega)$, potom platí následující vztah

$$Y(i\omega) = G(i\omega)U(i\omega).$$

U frekvenčních charakteristik se často využívá jejich grafického zobrazení (u rozsáhlých MIMO systémů není možné celý systém popisovat graficky, nicméně u jeho vybraných charakteristických částí to může být vhodné a názorné). Používají se zejména dva způsoby grafického zobrazení. *Nyquistova charakteristika* – grafické zobrazení frekvenčního přenosu v komplexní rovině

$$G(i\omega) = \text{Re}\{G(i\omega)\} + i\text{Im}\{G(i\omega)\},$$

nebo zobrazení v polárních souřadnicích

$$G(i\omega) = |G(i\omega)|\exp(i\arg(G(i\omega))).$$

Druhým často používaným způsobem je *Bodeho graf* – závislost $\log|G(i\omega)|$ a $\arg(G(i\omega))$ na $\log(\omega)$. Vyjádření v logaritmických souřadnicích je výhodné zejména proto, že je zde možné amplitudovou i fázovou charakteristiku dobře aproximovat přímkovými asymptotami.

1.4 Póly a nuly systému

Póly a nuly systému jsou komplexní čísla (komplexní frekvence), která charakterizují vnitřní dynamiku daného systému a jeho vazbu na okolí. Póly i nuly mají specifický fyzikální význam. Póly systému popisují vnitřní dynamiku systému, charakterizují vnitřní rezonance systému. Jsou to komplexní frekvence, které je systém schopen za určitých okolností generovat, aniž by byly přímo obsaženy ve vstupním signálu. Systém může při vhodně zvolených počátečních podmínkách na výstupu generovat signály $\exp(i\lambda_i t)$, kde λ_i jsou póly systému, aniž by byl kterýkoliv z těchto signálů obsažen ve vstupním signálu. Nuly systému charakterizují vazbu systému na jeho okolí a odpovídají antirezonancím systému. Nuly představují komplexní frekvence, které systém utlumí, pokud jsou ve vstupním signálu obsaženy. Systém mající na vstupu signál $\exp(i\nu_i t)$, kde ν_i je nula systému, má při vhodně zvolených počátečních podmínkách nulový výstup, a to přesto, že vstupní i stavový vektor je nenulový.

1.4.1 Přenosové póly a nuly systému a kanonický tvar přenosové matice

Přenosové póly (*transmission poles*) a přenosové nuly (*transmission zeros*) systému jsou komplexní frekvence dané přenosovou maticí systému $G(s)$ – matice racionálně lomených funkcí v proměnné s . Klasický postup, jak určit přenosové póly a nuly systému, je založen na tzv. *Smithově-McMillanově formě* přenosové matice $G(s)$. Začneme postupnými úpravami přenosové matice, z nichž první jsme již naznačili dříve

$$G(s) = \frac{1}{d(s)} F(s). \quad (1.17)$$

Z přenosové matice je možné vytknout polynom $d(s)$, který je nejmenším společným násobkem polynomů $Q_{ij}(s)$, jmenovatelů funkcí $G_{ij}(s)$. Tento polynom jsme nazvali nejjednodušším jmenovatelem matice $G(s)$. Zde by bylo namísto připomenout: matice $G(s)$ a $F(s)$ jsou matice rozměrů $p \times m$ (stejně jako například matice D , viz (1.11)), hodnoty obecně $q \leq \min \{p, m\}$.

Polynomiální matici $F(s)$ je možné pomocí řádkových a sloupcových úprav vyjádřit v tzv. *Smithově formě*, někdy též říkáme v *kanonickém tvaru*

$$L(s)F(s)R(s) = S(s) = \begin{bmatrix} S^*(s) & 0 \\ 0 & 0 \end{bmatrix}, \quad (1.18)$$

kde matice $L(s)$, resp. $R(s)$ lze interpretovat jako řádkové, resp. sloupcové úpravy matice $F(s)$. Obě tyto matice jsou čtvercové a *unimodulární* (jejich determinant je konstantní a nezávisí na s). Provedením těchto řádkových a sloupcových úprav získáme z obecné polynomiální matice $F(s)$ matici $S(s)$, ta je polynomiální maticí v kanonickém tvaru. Matice $S(s)$ obsahuje nenulový diagonální blok $S^*(s)$ rozměrů $q \times q$, který má tvar

$$S^*(s) = \text{diag} [S_1(s), S_2(s), \dots, S_q(s)]. \quad (1.19)$$

Každý polynom $S_i(s)$ má tu vlastnost, že dělí všechny polynomy $S_j(s)$, kde $i \leq j \leq q$. Tyto polynomy nazýváme *invariantní faktory (polynomy)* matice $F(s)$. To tedy znamená, že počet kořenových činitelů polynomů $S_i(s)$ s rostoucím i neklesá [1.1; kap. 2.1], [1.2; kap. 3.4].

Analogicky lze pomocí řádkových a sloupcových úprav (matice reprezentující úpravy jsou stejné jako u (1.18)) odvodit kanonický tvar matice $G(s)$, tzv. *Smithovu-McMillanovu formu* (viz [1.1; kap. 2.3], [1.2; kap. 3.4])

$$L(s)G(s)R(s) = M(s) = \begin{bmatrix} M^*(s) & 0 \\ 0 & 0 \end{bmatrix}, \quad (1.20)$$

kde matice $M(s)$ je právě přenosová matice systému v Smithově-McMillanově tvaru. Mezi kanonickými tvary matic $F(s)$ a $G(s)$ zřejmě platí vztah

$$M(s) = \frac{1}{d(s)} S(s). \quad (1.21)$$

Kanonické tvary obou matic $G(s)$ a $F(s)$ mají také obdobnou strukturu. Matice $M(s)$ obsahuje nenulový diagonální blok $M^*(s)$ rozměru $q \times q$ tvořený racionálně lomenými funkcemi

$$\begin{aligned} M^*(s) &= \text{diag}[M_1(s), M_2(s), \dots, M_q(s)] = \\ &= \text{diag}\left[\frac{\varepsilon_1(s)}{\psi_1(s)}, \frac{\varepsilon_2(s)}{\psi_2(s)}, \dots, \frac{\varepsilon_q(s)}{\psi_q(s)}\right]. \end{aligned} \quad (1.22)$$

Pro polynomy $\varepsilon_i(s)$ a $\psi_i(s)$ platí: každý polynom $\varepsilon_i(s)$ dělí všechny polynomy $\varepsilon_j(s)$, kde $i \leq j \leq q$, a každý polynom $\psi_i(s)$ dělí všechny polynomy $\psi_k(s)$, kde $0 \leq k < i$. To znamená, že s rostoucím i počet kořenových činitelů u polynomů $\varepsilon_i(s)$ neklesá a u polynomů $\psi_i(s)$ neroste.

Přenosové póly systému λ_i jsou pak definovány jako kořeny polynomů $\psi_i(s)$ (při buzení systému signálem $\exp(\lambda_i t)$ bude odezva systému nekonečná). Součin polynomů $\psi_i(s)$ nazýváme *pólovým polynomem*, značíme

$$\lambda(s) = \prod_{i=1}^q \psi_i(s). \quad (1.23)$$

Obdobně přenosové nuly systému ν_i jsou definovány jako kořeny polynomů $\varepsilon_i(s)$ a součin těchto polynomů pak nazveme *nulovým polynomem*, značíme

$$\nu(s) = \prod_{i=1}^q \varepsilon_i(s). \quad (1.24)$$

1.4.2 Určení kanonického tvaru matice pomocí minorů

Kanonický tvar matice $G(s)$ lze získat i jiným postupem, než pomocí řádkových a sloupcových úprav matice, například pomocí minorů (subdeterminantů) matice. Nejprve z

přenosové matice $G(s)$ pomocí vztahu (1.17) vyjádříme polynomiální matici $F(s)$. Lze ukázat [1.2; kap. 3.4; (3.28)], že pro jednotlivé polynomy $S_i(s)$ kanonického tvaru (1.18) polynomiální matice $F(s)$ platí

$$S_i(s) = \frac{D_i(s)}{D_{i-1}(s)}, \quad (1.25)$$

kde $D_i(s)$ je největší společný dělitel všech minorů i -tého řádu dané matice $F(s)$, přičemž platí, že $D_0 \equiv 1$. Pro určení kanonického tvaru přenosové matice $G(s)$ – Smithovy-McMillanovy formy pak stačí použít vztah (1.21); platí tedy

$$M_i(s) = \frac{\varepsilon_i(s)}{\psi_i(s)} = \frac{S_i(s)}{d(s)} = \frac{D_i(s)}{d(s)D_{i-1}(s)}. \quad (1.26)$$

Tento postup je u rozsáhlejších systémů zřejmě velmi neefektivní, vzhledem k potřebě počítat velké množství determinantů (polynomiálních matic) a určit jejich největší společný dělitel.

1.4.3 Různé typy nul systému

Přenosové nuly systému, které jsme určili pomocí Smithovy-McMillanovy formy přenosové matice, nejsou všechny nuly, které lze v systému definovat a které se v systému vyskytují. Existuje celá řada dalších definic různých nul. Komplexní frekvence, které nazýváme nuly, mají společnou tu vlastnost, že způsobují jisté nulové chování systému – typicky nulový výstup. Nepříjemné je to, že různé definice různých nul se často překrývají a různí autoři často používají u různých definic stejnou terminologii, což občas působí zmatek.

Základním rozdílem při výpočtu různých typů nul je to, z jaké matice vycházíme. Přenosové nuly byly vypočteny z nejjednoduššího jmenovatele přenosové matice systému $G(s)$. Nuly lze počítat i z charakteristického polynomu systému nebo z matice systému $P(s)$. Matice systému $P(s)$ obsahuje větší množství informací o systému než matice přenosová. Rozdíl mezi oběma maticemi je právě v rozdílné množině nul, kterou definují. Obecně lze říci, že množina nul definovaných maticí $P(s)$ je větší, než množina nul definovaných maticí $G(s)$; pokud ovšem je systém takzvaně *kompletně říditelný* a *kompletně pozorovatelný*, obě množiny nul se shodují [1.1; kap. 4; Transmission zeros].

Nyní stručně naznačíme, s jakými typy nul se můžeme v teorii systémů setkat.

- (i) *Přenosové nuly (transmission zeros)* určíme z kanonického tvaru tzv. Smithovy-McMillanovy formy (1.22) přenosové matice $G(s)$, respektive z nulového polynomu (1.24) nalezením jeho kořenů. Kanonický tvar matice získáme pomocí řádkových a sloupcových úprav nebo pomocí minorů matice (1.25).
- (ii) *Invariantní nuly (invariant zeros)* jsou definovány pomocí minorů nejvyššího řádu matice systému $P(s)$; přesněji řečeno jedná se o kořeny polynomu, který je definován jako největší společný dělitel všech minorů nejvyššího řádu matice $P(s)$. Typické pro invariantní nuly je nulové chování systému na výstupu. Množina přenosových nul je podmnožinou množiny nul invariantních [1.1; kap 4].

- (iii) *Výstupní potlačené nuly (output decoupling zeros)* se vyskytují v systémech, které nejsou tzv. *kompletně pozorovatelné* (viz kapitolu 1.5.2). V podstatě tyto nuly existují, existuje-li minimálně jeden n -rozměrný vektor w (pravý vlastní vektor matice A), pro který platí $Cw = 0$; musí tedy platit

$$\begin{bmatrix} z_o I - A \\ C \end{bmatrix} w = 0, \quad (1.27)$$

kde z_o je *výstupní potlačená nula* [1.1; kap. 5.1; (5.1,3)]. Výstupní nuly lze počítat pomocí Smithovy formy matice $P_o(s)$, kde

$$P_o(s) \equiv \begin{bmatrix} sI - A \\ C \end{bmatrix}. \quad (1.28)$$

Lze ukázat, že systém nemá žádné výstupní potlačené nuly tehdy a jen tehdy, když Smithova forma matice $P_o(s)$ má tvar $[I_n, 0]^T$. Pokud matice nemá tento tvar Smithovy formy, lze nalézt unimodulární matici $Q_R(s)$, která je *pravým společným dělitelem* [1.9; str. 32; def. 11] matice $(sI-A)$ a matice C . Matice $Q_R(s)$ charakterizuje *nepozorovatelný subsystém* a obsahuje výstupní potlačené nuly zkoumaného systému [1.1; kap. 5.1].

- (iv) *Vstupní potlačené nuly (input decoupling zeros)* se vyskytují v systémech, které nejsou tzv. *kompletně říditelné* (viz kapitolu 1.5.1). Opět lze ukázat, že existence těchto nul souvisí s existencí n -rozměrného sloupcového vektoru v (levý vlastní vektor matice A), pro který platí $v^H B = 0$ (symbolem H značíme transponovaný, komplexně sdružený tedy tzv. *Hermitovsky sdružený*), musí platit

$$v^H [z_I I - A, \quad -B] = 0, \quad (1.29)$$

kde z_I je *vstupní potlačená nula* [1.1; kap. 5.2; (5.2,3)]. Vstupní nuly lze počítat pomocí Smithovy formy matice $P_I(s)$, kde

$$P_I(s) \equiv [sI - A, \quad -B]. \quad (1.30)$$

Lze ukázat, že systém nemá žádné vstupní potlačené nuly tehdy a jen tehdy, když Smithova forma matice $P_I(s)$ má tvar $[I_n, 0]$. Pokud matice tento tvar Smithovy formy nemá, lze nalézt unimodulární matici $Q_L(s)$, která je *levým společným dělitelem* [1.9; str. 32; def. 11] matice $(sI-A)$ a matice B . Matice $Q_L(s)$ charakterizuje *neříditelný subsystém* a obsahuje vstupní potlačené nuly zkoumaného systému [1.1; kap. 5.2].

- (v) *Vstupně-výstupní potlačené nuly (input-output decoupling zeros)*; může se stát, že bude existovat nějaké w , v^* a nějaké z_{IO} tak, že budou splněny oba vztahy (1.27) a (1.29) současně. Bude platit rovnost $z_I = z_o = z_{IO}$. Takové z_{IO} pak nazveme *vstupně-výstupní potlačenou nulou*. Je zřejmé, že vstupně-výstupní potlačená nula je taková nula, která patří současně do množiny výstupních potlačených nul i do množiny vstupních potlačených nul [1.1; kap. 5.3].

- (vi) *Systémové nuly (system zeros)* lze určit pomocí vhodně vybraných minorů matice systému $P(s)$, viz [1.1; kap. 6]. Obecně lze říci, že systémové nuly tvoří kompletní množinu všech nul systému.

1.4.4 Vztahy mezi různými typy nul

Mezi množinami různých typů nul definovanými v předchozím odstavci existuje velké množství vztahů. Zde zmíníme pouze několik základních vztahů. Různé množiny jsou definovány pomocí různých matic – přenosové nuly pomocí přenosové matice $G(s)$, invariantní a systémové nuly pomocí systémové matice $P(s)$ a vstupní, respektive výstupní potlačené nuly pomocí různých bloků systémové matice $P(s)$; viz (1.28) a (1.30).

Ze vztahu (1.11) pro výpočet přenosové matice systému (pro jednoduchost uvažujeme systém s nulovou maticí D)

$$G(s) = C(sI - A)^{-1} B$$

a ze vztahů definujících vstupní a výstupní potlačené nuly (1.27), (1.29) lze odvodit význam potlačených nul a jejich vliv na systém. V popisu výstupních, resp. vstupních potlačených nul v předchozím odstavci bylo řečeno, že existuje polynomiální matice $Q_R(s)$, resp. $Q_L(s)$, která je pravým, resp. levým společným dělitelem matice $(sI - A)$ a matice C , resp. B . Ze vztahu pro výpočet přenosové matice (1.11) je zřejmé, že při výpočtu přenosu dojde k vykrácení faktorů $Q_R(s)$ a $Q_L(s)$. Důsledkem je vykrácení potlačených nul, které byly v těchto faktorech obsaženy. Tyto nuly nijak neovlivňují přenos systému a nevyskytují se v množině nul přenosových (definované maticí $G(s)$). Oproti tomu například množina invariantních nul, definovaná pomocí matice $P(s)$, která obsahuje ještě nevykrácené informace, může některé z potlačených nul obsahovat. Množina invariantních nul obsahuje celou množinu nul přenosových a v případě, že má systém více výstupů než vstupů nebo naopak, obsahuje i část množiny výstupních nebo vstupních potlačených nul. Počet výstupních, resp. vstupních potlačených nul, které množina nul invariantních obsahuje, pak souvisí s počtem *nezpozorovatelných*, resp. *neřiditelných módů* systému, viz [1.1; kap. 9.1].

Rozdíl mezi množinou systémových nul a mezi množinou invariantních nul je v tom, že množina systémových nul obsahuje (kromě množiny nul přenosových) kompletní sadu vstupních i výstupních potlačených nul. Vztah mezi množinami lze jednoduše vyjádřit následujícím způsobem

$$\{\text{přenosové nuly}\} \subseteq \{\text{invariantní nuly}\} \subseteq \{\text{systémové nuly}\},$$

přičemž platí

$$\begin{aligned} \{\text{systémové nuly}\} \setminus \{\text{přenosové nuly}\} &= \\ &= \{\text{výstupní potlačené nuly}\} \cup \{\text{vstupní potlačené nuly}\}. \end{aligned}$$

V souvislosti s krácením potlačených nul při výpočtu přenosové matice $G(s)$ si všimneme tohoto faktu. To, s čím se potlačené nuly vykrátí, jsou póly systému, které vznikly inverzí matice $(sI-A)$ z potlačených nul, viz (1.11). Lze tak dodefinovat množinu *systémových pólů*. Přenosové póly jsou definovány přenosovou maticí $G(s)$, resp. pólovým polynomem (1.23), systémové póly jsou definovány maticí systému $P(s)$. Platí vztahy

$$\{\text{systémové nuly}\} = \{\text{přenosové nuly}\} \cup \\ \cup \{\text{výstupní potlačené nuly}\} \cup \{\text{vstupní potlačené nuly}\}$$

a

$$\{\text{systémové póly}\} = \{\text{přenosové póly}\} \cup \\ \cup \{\text{výstupní potlačené nuly}\} \cup \{\text{vstupní potlačené nuly}\}.$$

1.5 Základní pojmy z teorie řízení a vlastnosti systémů (-abilities)

V teorii řízení zkoumáme vlastnosti systémů, které nás zajímají v souvislosti s řízením těchto systémů. Řízením systému rozumíme generování vstupní veličiny $u(t)$ systému (1.1), (1.2), kterou nazýváme *řídící veličina*. Řídící veličinu generujeme tzv. *regulátorem*. Řízením se obecně snažíme dosáhnout toho, aby se výstup systému $y(t)$ choval podle našich představ. V praxi to znamená, že máme zadanou určitou *požadovanou hodnotu* výstupu, které se snažíme dosáhnout. Ta tvoří jeden ze vstupů regulátoru. Druhým vstupem regulátoru je hodnota skutečného výstupu systému, tj. hodnota výstupu, kterou systém skutečně generuje. Rozdíl mezi požadovanou hodnotou výstupu a skutečnou hodnotou výstupu nazýváme *regulační odchylka*. Regulátorem se snažíme na základě regulační odchylky generovat takový průběh řídicí veličiny $u(t)$, aby se regulační odchylka zmenšovala. V systému tak zavedeme zpětnovazební regulaci.

Zmenšování regulační odchylky je velmi široký pojem a ke zmenšování může docházet mnoha různými regulátory. Regulátor může například systém řídit tak, abychom dosáhli požadovaného výstupu v minimálním čase. Existuje tak celá řada různých algoritmů regulace – *časově optimální řízení, kvadraticky optimální řízení*, atd., viz [1.6]. Možnost řídit daný systém je také závislá na vlastnostech tohoto systému. Zde se budeme velmi stručně zabývat výhradně vlastnostmi systémů, které řízení ovlivňují (ne tedy problémy vlastní realizace regulátoru).

1.5.1 Řiditelnost a dosažitelnost v systému (Controllability) [1.2; kap. 8]

Jak jsme již zmínili, jednou ze základních úloh teorie řízení je najít takový průběh řídicí veličiny $u(t)$, která způsobí změnu stavu x , resp. výstupu y z dané počáteční hodnoty x_0 , resp. y_0 v určitou předem zvolenou hodnotu stavu x_1 , resp. výstupu y_1 . Prvním problémem, který se vyskytne v souvislosti s řešením této úlohy, je samotná *existence takového řízení*. Jsou systémy, u kterých nelze některých stavů dosáhnout žádným průběhem řídicí veličiny. Pokud takové řízení existuje, může jich naopak existovat více (existuje tedy více průběhů řídicí veličiny) takových, které systém dovedou k námi požadované hodnotě výstupu. Z těchto úvah vyplývá *problém optimálního řízení systému* [1.2].

Dosažitelnost stavu

V předchozím odstavci jsme zmínili, že některých stavů systému nelze dosáhnout. Nyní přesně definujeme pojem *dosažitelnosti stavu v systému*. Máme-li nějaký stav x_D , kterého chceme dosáhnout, je tento stav dosažitelný tehdy a jen tehdy, existuje-li takové řízení $u(t)$, které systém do tohoto stavu převede z počátečního stavu x_0 v konečném čase. O celém systému pak můžeme říci, že je *systémem dosažitelným*, jsou-li dosažitelné všechny jeho stavy.

Řiditelnost stavu

V jistém smyslu opačný pojem k pojmu dosažitelnost stavu v systému je pojem *řiditelnosti stavu v systému*. Máme-li systém v nějakém stavu x_R , hledáme nějaké řízení $u(t)$, které převede systém do stavu počátečního (nulového) x_0 v konečném čase. Existuje-li takové řízení, řekneme, že stav x_R systému je říditelný. Jsou-li všechny stavy systému říditelné, řekneme o systému, že je *systémem říditelným*.

Řiditelnost a dosažitelnost stavu v nestacionárním systému

Složitější je situace u nestacionárních systémů. Nestacionární systémy jsou takové systémy, u kterých jsou obecně matice A , B , C a D závislé na čase. Takové systémy mění pochopitelně své vlastnosti v závislosti na čase. Stav, který je říditelný, resp. dosažitelný v nějakém čase t_1 , nemusí být obecně říditelný, resp. dosažitelný v nějakém jiném čase t_2 . U nestacionárních systémů tak nezkoumáme vlastnosti stavu x , ale obecně vlastnosti události (t_i, x) , viz [1.2; kap. 8.6].

Říkáme, že událost (t_2, x) je dosažitelná, existuje-li konečný čas $t_1 \leq t_2$ a řízení $u(t)$, přičemž $t_1 \leq t \leq t_2$ takové, že událost $(t_1, 0)$ jím lze převést na událost (t_2, x) . Platí-li toto tvrzení pro libovolný stav x , pak je systém dosažitelný v čase t_2 .

Říkáme, že událost (t_2, x) je říditelná, existuje-li konečný čas $t_3 \geq t_2$ a řízení $u(t)$, přičemž $t_2 \leq t \leq t_3$ takové, že událost (t_2, x) jím lze převést na událost $(t_3, 0)$. Platí-li toto tvrzení pro libovolný stav x , pak je systém tzv. říditelný v čase t_2 .

Kritéria říditelnosti a dosažitelnosti stavu

Mohlo by se zdát, že oba pojmy (řiditelnost i dosažitelnost) jsou identické a že říditelnost získáme z dosažitelnosti pouze „inverzí času“ a naopak. U tzv. *reverzibilních systémů* tyto pojmy opravdu splývají. Stacionární lineární dynamické systémy, tedy systémy, kde matice A , B , C , D jsou nezávislé na čase, jsou systémy reverzibilní.

Nyní je potřeba rozhodnout o tom, zda daný lineární dynamický systém (stacionární, spojitý) popsáný rovnicemi (1.1), (1.2) je říditelný, respektive dosažitelný. Odvození kritérií říditelnosti a dosažitelnosti viz [1.2; kap. 8.4, 8.5]; zde uvedeme pouze výsledky, tedy jednotlivá kritéria.

O nějakém stavu x_R řekneme, že je stavem říditelným, pokud leží v prostoru generovaném sloupci matice B a maticí A , jak je vidět z následujícího výrazu (připomeňme, že matice A má rozměry $n \times n$ a matice B má rozměry $n \times m$),

$$x_R \in \text{span} \{b_1, b_2, \dots, b_m, Ab_1, Ab_2, \dots, Ab_m, \dots, A^{n-1}b_1, A^{n-1}b_2, \dots, A^{n-1}b_m\},$$

kde b_i jsou jednotlivé sloupce matice B . Lze tedy říci, že lineární dynamický systém (stacionární, spojitý) je říditelný, resp. dosažitelný, pokud platí

$$\text{rank} [B, AB, A^2B, \dots, A^{n-1}B] \equiv \text{rank} [Q_C] = n, \quad (1.31a)$$

kde matice Q_C má rozměry $n \times mn$, viz [1.2; kap. 8.4; (8.25)].

Existují i další kritéria říditelnosti, respektive dosažitelnosti systému. Například test pomocí vlastních vektorů. Systém popsaný rovnicemi (1.1), (1.2) je říditelný, pokud neexistuje žádný nenulový sloupcový vektor q , který je levým vlastním vektorem matice A a současně je kolmý na všechny sloupce matice B . Pokud jsou splněny obě následující rovnosti

$$q^H A = \lambda q^H, \quad q^H B = 0$$

pro nějaký nenulový vektor q^H a pro příslušné vlastní číslo λ , systém říditelný není. Viz rovnici (1.29) a odstavec o vstupních potlačených nulách.

Další způsob, jak popsat říditelnost systému, je test pomocí hodnoty. Lineární dynamický systém (1.1), (1.2) je říditelný pouze tehdy, platí-li následující rovnost

$$\text{rank} [\lambda_i I - A, B] = n \quad (1.31b)$$

pro všechna $\lambda_i \in \lambda(A)$.

Říditelnost výstupu a kritéria říditelnosti výstupu

Podobně jako jsme definovali říditelnost, resp. dosažitelnost stavu, je možné definovat i říditelnost výstupu. Ptáme se, zda je možné převést výstup z nějaké libovolné hodnoty y_0 na jinou libovolnou hodnotu y_1 . Respektive ptáme se na to, zda existuje takové řízení $u(t)$, které by změnu výstupu realizovalo v konečném čase.

Říditelnost výstupu systému lze popsat a vyjádřit řadou různých způsobů; uvedeme pouze základní kritérium, které jsme uvedli i u popisu říditelnosti stavu systému. Říkáme, že lineární dynamický systém má říditelný výstup, pokud platí

$$\text{rank} [D, CB, CAB, \dots, CA^{n-1}B] \equiv \text{rank} [Q_{Cy}] = p, \quad (1.32)$$

kde matice Q_{Cy} má rozměry $p \times m(n+1)$. Říkáme také, že lineární dynamický systém (stacionární, spojitý) systém vyjádřený pomocí přenosové matice $G(s)$ má říditelný výstup, jestliže jsou řádky této matice lineárně nezávislé, viz [1.2; kap. 8.7].

1.5.2 Pozorovatelnost a rekonstruovatelnost systému (Observability) [1.2; kap. 9]

Stavové veličiny systému jsou v podstatě „ukryté uvnitř systému“ a obvykle nejsou měřitelné, na rozdíl od veličin vstupních a výstupních. Důležitou otázkou tedy je, zda lze měřením vstupních a výstupních veličin určit stav, v kterém se systém nachází.

Pozorovatelnost systému

O systému řekneme, že je *pozorovatelný*, je-li možno změřením vstupní a výstupní veličiny na konečném intervalu jednoznačně určit stav systému na začátku tohoto intervalu. Nemůžeme-li určit stav systému jednoznačně, pak systém obsahuje tzv. *nepozorovatelné stavy*. To jsou takové stavy systému, které se vůbec neprojeví na výstupu.

Rekonstruovatelnost systému

O systému řekneme, že je *rekonstruovatelný*, je-li možno změřením vstupní a výstupní veličiny na konečném intervalu jednoznačně určit stav systému na konci tohoto intervalu. Rekonstruovatelnost systému se také někdy nazývá *obnovitelností* nebo *identifikovatelností* systému. Pozorovatelnost a rekonstruovatelnost systému mají velký význam při tzv. *experimentální identifikaci systému*.

Pozorovatelnost a rekonstruovatelnost systému jsou velmi blízké pojmy. Pokud jsme schopni určit stav systému na počátku nějakého intervalu (tedy počáteční podmínku stavu v tomto intervalu), jsme vždy schopni určit i stav systému na konci tohoto intervalu. Z pozorovatelnosti systému tedy obecně přímo vyplývá i rekonstruovatelnost systému (opačné tvrzení neplatí). Oba pojmy splývají u systémů reverzibilních stejně jako tomu je v případě říditelnosti a dosažitelnosti. U nestacionárních systémů jsou oba pojmy pozorovatelnost i rekonstruovatelnost obecně závislé na čase.

Kritéria pozorovatelnosti a rekonstruovatelnosti systému

Podobně jako u říditelnosti a dosažitelnosti lze ukázat, že lineární dynamický systém (stacionární, spojitý) je pozorovatelný, resp. rekonstruovatelný, pokud platí

$$\text{rank} \begin{bmatrix} C^T & (CA)^T & (CA^2)^T & \dots & (CA^{n-1})^T \end{bmatrix} \equiv \text{rank}[Q_o] = n, \quad (1.33a)$$

kde matice Q_o má rozměry $n \times pn$, viz [1.2; kap. 9.3.1].

Dalším ze základních kritérií říditelnosti je kritérium pomocí vlastních vektorů matice A . Systém (1.1), (1.2) je nepozorovatelný pouze tehdy, je-li některý z vlastních vektorů r matice A kolmý na řádky matice C . Tedy pokud platí obě rovnosti

$$Ar = \lambda r, \quad Cr = 0$$

pro nějaký nenulový vektor r a pro příslušné vlastní číslo λ , systém pozorovatelný není. Vektor r , který splňuje výše uvedenou rovnost, se pak nazývá *nepozorovatelný vlastní vektor*,

jemu odpovídající vlastní číslo λ se nazývá *nepozorovatelné vlastní číslo* a dohromady tvoří tzv. *nepozorovatelný mód*. Viz rovnici (1.27) a odstavec o výstupních potlačených nulách.

Posledním kritériem, které uvedeme, je kritérium pomocí hodnosti. Systém (1.1), (1.2) je pozorovatelný pouze tehdy, pokud platí

$$\text{rank}[\lambda_i I - A^T, C^T] = n \quad (1.33b)$$

pro všechna $\lambda_i \in \lambda(A)$.

1.5.3 Poznámky k dosažitelnosti a pozorovatelnosti

Dualita pojmů

Dosažitelnost (řiditelnost) a pozorovatelnost (rekonstruovatelnost) systému spolu úzce souvisejí. Zavedeme-li nějaký systém $S_1 = (A, B, C, 0)$ a systém $S_2 = (A^T, C^T, B^T, 0)$, který nazveme *systémem duálním* k systému S_1 , pak platí, že je-li systém S_1 dosažitelný (řiditelný), pak je systém S_2 pozorovatelný (rekonstruovatelný) a naopak [1.2; kap. 9.3.3].

Nedosažitelné a nepozorovatelné systémy

Přenos systému (prvek přenosové matice) je racionálně lomená funkce. Je-li každý přenos systému tvořen dvěma nesoudělnými polynomy (nemají stejné kořenové činitele), říkáme, že je systém v *minimální realizaci*. Pokud je alespoň jeden z přenosů systému rozšířen (polynom čitatele a polynom jmenovatele mají nevykrácený stejný kořenový činitel), systém v minimální realizaci není, říkáme, že systém je v *neminimální realizaci*. Nevykrácený kořenový činitel polynomu čitatele/jmenovatele bude v systému představovat nulu/pól. Neminimální realizace není vhodná pro analýzu dosažitelnosti a pozorovatelnosti systému. Nevhodná neminimální realizace může způsobit, že systém nebude plně dosažitelný nebo pozorovatelný.

Obdobný mechanismus ztráty dosažitelnosti resp. pozorovatelnosti je v systémech, které jsou tvořeny více subsystémy např. v sériovém zapojení. Přenos systému tvořeného sériovou kombinací subsystémů je dán součinem přenosů jednotlivých subsystémů. Má-li jeden ze subsystémů přenosovou nulu v takovém bodě, v kterém má jiný subsystém přenosový pól, dojde k vykrácení pólu s nulou mezi subsystémy. Takto vytvořený systém pak bude také obsahovat nedosažitelné resp. nepozorovatelné části, viz [1.2; kap. 9.4].

Grammova matice (gramián) řiditelnosti, pozorovatelnosti

Grammova matice (gramián) řiditelnosti, resp. *pozorovatelnosti* má velký význam pro určení řiditelnosti (dosažitelnosti), resp. pozorovatelnosti (rekonstruovatelnosti) systému pro lineární, zejména nestacionární, ale i stacionární dynamické systémy.

Gramián řiditelnosti W_C je pro stacionární systémy definován vztahem

$$W_C = \int_0^{\infty} \exp(At) BB^T \exp(A^T t) dt \quad \text{pro spojité systémy,} \quad (1.34)$$

respektive $W_C = \sum_{i=0}^{\infty} \tilde{A}^i \tilde{B} \tilde{B}^T (\tilde{A}^T)^i \quad \text{pro diskrétní systémy} \quad (1.35)$

a vyhovuje Ljapunovově rovnici

$$AW_C + W_C A^T = -BB^T \quad \text{pro spojité systémy,} \quad (1.36)$$

respektive $\tilde{A}W_C \tilde{A}^T - W_C = -\tilde{B} \tilde{B}^T \quad \text{pro diskrétní systémy,} \quad (1.37)$

viz [1.2; kap. 8.6.2]. Jsou-li všechna vlastní čísla matice A v levé polorovině komplexní roviny, pak je gramián řiditelnosti pozitivně definitní pouze tehdy, je-li systém plně dosažitelný a řiditelný.

Gramián pozorovatelnosti W_O je pro stacionární systémy definován vztahem

$$W_O = \int_0^{\infty} \exp(A^T t) C^T C \exp(At) dt \quad \text{pro spojité systémy,} \quad (1.38)$$

respektive $W_O = \sum_{i=0}^{\infty} (\tilde{A}^T)^i C^T C \tilde{A}^i \quad \text{pro diskrétní systémy} \quad (1.39)$

a vyhovuje Ljapunovově rovnici

$$A^T W_O + W_O A = -C^T C \quad \text{pro spojité systémy,} \quad (1.40)$$

respektive $\tilde{A}^T W_O \tilde{A} - W_O = -C^T C \quad \text{pro diskrétní systémy,} \quad (1.41)$

viz [1.2; kap 9.5.2]. Jsou-li všechna vlastní čísla matice A v levé polorovině komplexní roviny, pak je gramián pozorovatelnosti pozitivně definitní pouze tehdy, je-li systém plně pozorovatelný a rekonstruovatelný.

V kapitole 3.4 se k problémům výpočtu řiditelnosti systému vrátíme. Řiditelnost spočítáme na konkrétním systému pomocí (1.31a). V kontextu numerické lineární algebry ukážeme nevhodnost postupu, který z (1.31a) plyne.

1.5.4 Stabilita, stabilizovatelnost a detekovatelnost systému [1.2; kap. 10]

Stabilita je pojem, který všichni určitým způsobem intuitivně chápeme, bohužel naše intuitivní představy o významu nějakého pojmu mohou být zdrojem nejasností. V různých oborech lidské činnosti se často používají stejné termíny pro vyjádření odlišného významu. Stejně je tomu i s pojmem stabilita. V teorii systémů se vyskytuje celá řada různých definic pojmu stabilita. Princip, na kterém je pojem stabilita postaven, je však většinou stejný. Vyvolá-li malá (omezená) změna $d\alpha < \varepsilon$ nějaké veličiny α malou (omezenou) změnu $d\beta < \delta$ nějaké jiné β , pak je veličina β určitým způsobem stabilní vzhledem k veličině α .

Klasicky se v teorii řízení zavádí tzv. *Ljapunovská stabilita*, která reprezentuje omezenost změn stavové veličiny x dynamického systému vzhledem k omezené změně počátečních podmínek v okolí nějakého *rovnovážného stavu* systému x_L . Rovnovážným stavem systému se stavovým popisem $\dot{x} = f(x)$ rozumíme stav x_R , pro který platí $f(x_R) = 0$. Dále se definuje tzv. *kvaziasymptotická stabilita*, která reprezentuje schopnost stavu x dynamického systému pro čas rostoucí nade všechny meze konvergovat k nějakému rovnovážnému stavu x_K . Ljapunovská stabilita o konvergenci stavu nic neříká, naopak kvaziasymptotická stabilita nic neříká o jeho omezenosti v libovolném daném čase. Je-li však nějaký rovnovážný stav x_S systému Ljapunovsky stabilní a zároveň je i kvaziasymptoticky stabilní, říkáme o něm, že je *asymptoticky stabilní*. Asymptotická stabilita rovnovážných stavů má v praxi asi největší význam. Pro lineární systém lze ukázat, že je daným způsobem stabilní tehdy, je-li stejným způsobem stabilní jeho nulový bod (tedy stav $x = 0$); viz [1.2; kap. 10]

Dále se definuje celá řada dalších stabilit systémů. Podobně jako jsme popsali stabilitu stavu systému, je možné definovat stabilitu výstupu systému. Pokud například systém není plně pozorovatelný (existují stavy, které se neprojeví na výstupu), pak je možné, že přestože je stav systému nestabilní, výstup může být stabilní. Stabilita výstupu systému vzhledem k vstupu systému se také někdy nazývá *stabilitou vnějšího chování* systému nebo také *Bounded Input Bounded Output (BIBO)* – každému omezenému vstupu odpovídá omezený výstup systému.

Pro určení stability lineárních systémů se používá a existuje celá řada kritérií (například *Routhovo kritérium stability* [1.2; kap. 10.4.1], *Hurwitzovo kritérium stability* [1.2; kap. 10.4.2] nebo *Routhovo-Schurovo kritérium stability* [1.2; kap. 10.4.3]). Základním kritériem stability lineárního systému, ze kterého většina kritérií vychází, je fakt, že vlastní čísla matice A musí mít záporné reálné části, tedy že vlastní čísla matice A leží v levé polorovině komplexní roviny. Leží-li nějaké vlastní číslo matice A přímo na imaginární ose a má tedy reálnou část nulovou, říkáme o systému, že je *na mezi stability*. Připomeňme, že vlastní čísla matice A (kořeny charakteristického polynomu systému) úzce souvisí s přenosovými póly systému. Výše jmenovaná kritéria se často snaží rozhodnout o stabilitě systému a obejít přitom problém s hledáním vlastních čísel, tedy problém faktorizace charakteristického polynomu systému.

Většina kritérií stability rozhoduje o stabilitě stacionárních lineárních dynamických systémů. Chceme-li tedy taková kritéria použít i pro systém nestacionární nebo nelineární, systém v daném bodě a daném čase linearizujeme a stabilitu počítáme pouze v okolí tohoto bodu a v tomto čase. Jedním z kritérií stability systému jsou *Ljapunovovy věty o stabilitě*; viz [1.2; kap. 10.2]. Ljapunovovy metody určení stability systému lze aplikovat i na systémy

nelineární. Tyto metody jsou tedy obecnější a v analýze nelineárních systémů jsou nejpoužívanější. Ljapunovovy metody jsou také založeny na principu linearizace systému.

Z Ljapunovovy analýzy stability [1.2; kap. 10.2] vyplývají důležitá tvrzení a závěry i pro lineární dynamické systémy. Některá si zde uvedeme. Máme-li lineární dynamický systém daný rovnicí

$$\dot{x}(t) = Ax(t) \quad \text{spojitý}$$

nebo $x_{k+1} = \tilde{A}x_k$ diskrétní,

pak tento systém má stav $x = 0$ asymptoticky stabilní tehdy, když k libovolné symetrické pozitivně definitní matici Q existuje symetrická pozitivně definitní matice P taková, že platí Ljapunovova rovnice [1.2; kap. 10.2; (10.11), (10.13)]

$$A^T P + PA = -Q \quad \text{pro spojitý,} \quad (1.42)$$

respektive $\tilde{A}^T P \tilde{A} - P = -Q$ pro diskretní systém. (1.43)

S ohledem na jednoduchost řešení často za matici Q volíme jednotkovou matici I , rovnice pak mají tvar

$$A^T P + PA = -I \quad \text{pro spojitý,} \quad (1.44)$$

respektive $\tilde{A}^T P \tilde{A} - P = -I$ pro diskretní systém. (1.45)

Stabilizovatelnost a detekovatelnost

Podobně jako je možné každý systém rozložit na například dosažitelnou a nedosažitelnou část, je možné systém také rozdělit na jeho část stabilní a na část nestabilní. O systému pak říkáme, že je *stabilizovatelný* v tom případě, že celá jeho nestabilní část (všechny jeho nestabilní módy) je obsažena v jeho dosažitelném podprostoru. To znamená, že každý nestabilní stav systému je dosažitelný, tedy nedosažitelné mohou být pouze takové stavy, které jsou stabilní. Je zřejmé, že je-li systém dosažitelný nebo stabilní, pak je také stabilizovatelný. Obdobně můžeme o systému říci, že je *detekovatelný*, jestliže všechny jeho nestabilní stavy jsou pozorovatelné (projeví se ve výstupu). To znamená, že nepozorovatelné jsou pouze takové stavy, které jsou stabilní. Je zřejmé, že je-li systém pozorovatelný nebo stabilní, pak je tento systém také detekovatelný; viz [1.2; kap. 10.6].

2. Nástroje lineární algebry používané v teorii řízení

Ze základních lineárních algebraických úloh, které se v teorii řízení a v teorii dynamických systémů nejčastěji vyskytují, zmíníme čtyři. Řešení soustavy lineárních rovnic (viz kapitolu 2.1), řešení zobecněné soustavy lineárních rovnic, tzv. problému nejmenších čtverců (viz kapitolu 2.2), problém vlastních čísel matice (viz kapitolu 2.3) a problém singulárních čísel matice, respektive tzv. singulární rozklad (viz kapitolu 2.4). Tyto algebraické úlohy a nástroje na jejich řešení jsou často v technické literatuře popsány jen okrajově. Nástroje na řešení jednoduchých učebnicových úloh mnohdy nejsou použitelné k řešení úloh, které se v praxi vyskytují (z důvodu rozměrů reálných úloh; typickým příkladem jsou soustavy lineárních rovnic s řádově až miliony neznámých).

Nejprve si zavedeme některé pojmy lineární algebry, které budeme později používat. Mějme matici A rozměrů $n \times m$ (má tedy n m -rozměrných řádkových vektorů, resp. m n -rozměrných sloupcových vektorů), pak $R(A)$ označuje *obor hodnot matice A (range A)*. Obor hodnot je množina všech n -rozměrných vektorů y , které generuje matice A ; píšeme

$$R(A) = \{y \in R^n : y = Ax, \forall x \in R^m\} = \text{span}\{a_1, \dots, a_m\},$$
$$R(A) \subseteq R^n,$$

kde sloupcové vektory a_i jsou sloupce matice A . Obor hodnot matice A tvoří podprostor prostoru R^n . Dále $N(A)$ označuje *nulový prostor matice A (null space A)*. Nulový prostor je množina všech m -rozměrných vektorů x takových, které matice A zobrazí na nulový vektor; píšeme

$$N(A) = \{x \in R^m : Ax = 0\},$$
$$N(A) \subseteq R^m.$$

Nulový prostor matice A tvoří podprostor prostoru R^m . Mezi oborem hodnot matice a nulovým prostorem matice platí vztahy

$$R(A) \cup N(A^T) = R^n, \quad R(A) \perp N(A^T),$$
$$R(A^T) \cup N(A) = R^m, \quad R(A^T) \perp N(A),$$

kde symbol \perp značí, že prostory, resp. báze prostorů jsou vzájemně ortogonální; viz [2.1; kap. 7.2.1].

2.1 Soustavy lineárních rovnic

Řešení soustavy lineárních rovnic je jedna z nejznámějších úloh lineární algebry. V té nejjednodušší podobě se se soustavami rovnic setkáváme již na základní škole. Soustavy rovnic jsou také jednou z nejčastěji řešených algebraických úloh v technické a inženýrské praxi. V první kapitole, týkající se systémů a teorie řízení, jsme na problém řešení soustav lineárních rovnic také několikrát narazili. Soustavy lineárních rovnic jsou například tzv. *Ljapunovovy rovnice*, s kterými jsme se již setkali (viz kapitolu 1.5, problém říditelnosti a pozorovatelnosti dynamického systému, Ljapunovovy věty o stabilitě).

Soustavu n lineárních rovnic

$$\sum_{j=1}^n a_{ij}x_j = b_i, \quad \text{pro } i=1,2,\dots,n$$

pro n neznámých x_i zapisujeme symbolicky v maticovém tvaru

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

respektive

$$Ax = b, \quad (2.1)$$

kde A je matice rozměru $n \times n$ – tzv. *matice soustavy*, b je n -rozměrný vektor – tzv. *vektor pravé strany* (zkráceně *pravá strana*) a x je n -rozměrný vektor neznámých. Pokud má matice A jednotlivé řádky, resp. sloupce lineárně nezávislé, říkáme, že je *regulární* (*invertovatelná*). K regulární matici A existuje inverzní matice A^{-1} , platí

$$A^{-1}A = AA^{-1} = I,$$

kde I je jednotková matice. Je-li matice A regulární, jednotlivé rovnice soustavy jsou lineárně nezávislé a soustava rovnic (2.1) má jednoznačné řešení x . Pomocí inverzní matice řešení vyjádříme jako

$$x = A^{-1}b. \quad (2.2)$$

2.1.1 Soustavy lineárních rovnic v atypickém tvaru

Ljapunovovy rovnice

Ne vždy je však soustava lineárních rovnic zapsaná ve tvaru (2.1). V kapitole 1.5 jsme se setkali s *Ljapunovými rovnicemi* (např. rovnice (1.36), (1.37)), tyto rovnice jsou také soustavy lineárních rovnic, ovšem zapsané ve speciálním tvaru. Nejběžnější Ljapunovova rovnice je

$$MX + XM^T = -Q, \quad (2.3)$$

kde M , Q jsou známé čtvercové matice, X je neznámá čtvercová matice; všechny matice jsou řádu $n \times n$. Vztah (2.3) popisuje soustavu lineárních rovnic se speciální strukturou, lze ji obecně převést na soustavu popsanou vztahem (2.1), naopak soustavu popsanou vztahem (2.1) na soustavu popsanou vztahem (2.3) obecně převést nelze. V praxi je ovšem často výhodné soustavu popsanou vztahem (2.3) na soustavu popsanou vztahem (2.1) nepřevádět; viz například [2.6].

Druhou Ljapunovovou rovnicí je

$$NXN^T - X = -Q, \quad (2.4)$$

kde N , Q jsou známé čtvercové matice, X je neznámá čtvercová matice; všechny matice jsou řádu $n \times n$. Vztah (2.4) stejně jako vztah (2.3) popisuje soustavu lineárních rovnic se speciální strukturou, lze ji obecně převést na soustavu popsanou vztahem (2.1), opačný převod obecně provést nelze.

Rovnice (2.3) a (2.4) v teorii řízení vycházejí z různých modelů dynamického systému, rovnice (2.3) je svázána se spojitým popisem; rovnice (2.4) je svázána s diskrétním popisem; viz například kapitolu 1.5.4, vztahy (1.42), (1.43).

Sylvestrový rovnice

Další typ nestandardního zápisu soustavy lineárních rovnic používají takzvané *Sylvestrový rovnice*. První Sylvestrova rovnice je

$$MX + XE = -Q, \quad (2.5)$$

druhá Sylvestrova rovnice je

$$NXF - X = -Q, \quad (2.6)$$

kde M , N jsou známé čtvercové matice řádu $n \times n$, matice E , F , jsou známé čtvercové matice řádu $m \times m$, matice Q je známá matice řádu $n \times m$, matice X je neznámá matice řádu $n \times m$. Ze zápisu rovnic je zřejmé, že jsou zobecněním rovnic Ljapunových.

Soustavu lineárních rovnic popsanou vztahem (2.5), resp. (2.6) lze opět obecně převést na soustavu lineárních rovnic popsanou vztahem (2.1), opačný převod obecně provést nelze.

Zobecnění Ljapunových a Sylvestrových rovnic

Existují ještě obecné tvary Ljapunových a Sylvestrových rovnic. Někdy se tak můžeme setkat se *symetricky sdruženou Ljapunovovou rovnicí*

$$MXE^T + EXM^T = -Q,$$

resp. s *obecnou Sylvestrovou rovnicí*

$$\sum_i N_i X F_i^T = -Q.$$

Oba vztahy opět popisují soustavu lineárních rovnic, lze je tak obecně přepsat na soustavu lineárních rovnic ve tvaru (2.1).

Riccatiho (nelineární) rovnice

V teorii řízení se často vyskytují ještě takzvané *Riccatiho rovnice* (například v souvislosti s problémem tzv. optimálního řízení [1.6; kap. 2.4]). Riccatiho rovnice je

$$MX + XM^T - XGX = -Q.$$

Tento vztah ovšem již nepopisuje soustavu lineárních rovnic, ale soustavu kvadratických rovnic. Rovnici zde zmiňujeme pouze proto, že lineární část této rovnice (tedy rovnice bez kvadratických členů) je právě Ljapunovovou rovnicí; viz [1.2], [1.6].

2.1.2 Přímé metody řešení soustavy rovnic (2.1)

Řešení soustavy lineárních rovnic klasickým tvaru, tedy pomocí matice soustavy a pomocí vektoru pravých stran, je možné nalézt mnoha způsoby. Hlavní rozdělení metod řešení (2.1) je rozdělení na tzv. *přímé metody* a na *iterační metody*. Přímé metody vycházejí z transformací matice soustavy a vektoru pravých stran, které provádíme tak dlouho, až nalezneme řešení. U metod iteračních se vychází z počátečního odhadu řešení, který se postupně zpřesňuje na základě odchylky tohoto odhadu od skutečného řešení. Iterační metody tak ponechávají matici soustavy v původním tvaru. To je základní rozdíl mezi přístupy jednotlivých typů metod.

Cramerovo pravidlo

Jako první metodu řešení (2.1) můžeme uvést tzv. *Cramerovo pravidlo*. Tuto metodu řešení uvádíme mezi přímými metodami, přestože není typickým představitelem přímých metod. Metodu zde zařazujeme spíše z historických důvodů. Touto metodou hledáme jednotlivé složky vektoru řešení $x = (x_1, x_2, \dots, x_i, \dots, x_n)^T$ soustavy rovnic (2.1) takto

$$x_i = \frac{D_i}{D},$$

kde D je determinant matice A této soustavy a D_i je determinant matice, která vznikne z matice A nahrazením i -tého sloupce vektorem pravých stran. Tato metoda je v praxi jen velmi těžko použitelná, výpočet $n+1$ determinantů matic řádu $n \times n$ je pro velká n úloha aritmeticky velmi náročná (pro n v řádech stovek ji lze již považovat za prakticky neproveditelnou).

Cramerovo pravidlo je typickým příkladem metody, která je velmi snadno aplikovatelná na malé úlohy (např. soustava tří rovnic) a zároveň je absolutně nepoužitelná pro úlohy rozsáhlé. Metoda je naprosto nepoužitelná i při řešení takových úloh na počítači. Zejména z důvodu velké spotřeby strojového času (obrovské množství aritmetických operací). Druhým důvodem je problém numerické stability (viz kapitolu 3.3) algoritmu, který by tuto metodu realizoval. Při použití nevhodného postupu výpočtu determinantu může být výsledek (vlivem zaokrouhlovacích chyb) zatížen neúměrně velkou chybou. Nalezené řešení soustavy (2.1) tak může být zcela nepoužitelné.

Gaussova eliminace (LU faktorizace)

Gaussova eliminace systematicky převádí původní soustavu (2.1) na ekvivalentní soustavu (na soustavu, která má stejné řešení jako soustava původní), která je v *horním*, resp. *dolním trojúhelníkovém tvaru*. Systematický převod je tvořen posloupností elementárních úprav soustavy (vhodných lineárních kombinací jednotlivých rovnic). Typicky se soustava převádí na horní trojúhelníkový tvar, dále tedy budeme uvažovat pouze tento tvar. Řešení soustavy s trojúhelníkovou maticí dostaneme tzv. *zpětným chodem* (nejprve nalezneme neznámou x_n vektoru neznámých x triviálním způsobem z poslední rovnice trojúhelníkové soustavy, pak obdobným způsobem nalezneme neznámou x_{n-1} a tak dále až nalezneme neznámou x_1). Všechny úpravy původní soustavy, vedoucí na horní trojúhelníkovou soustavu, lze zapsat jako transformační matici, ta má dolní trojúhelníkový tvar. V každé regulární matici A lze provést jistou pivotaci (vzájemnou záměnu několika řádků, resp. sloupců) tak, že výsledná matice má všechny levé horní čtvercové podmatice regulární (všechny hlavní rohové minory jsou nenulové). Takovou matici nazýváme *silně regulární*. Pro každou silně regulární matici A platí (viz [2.3; věta 1.24])

$$A = LU, \quad (2.7)$$

kde L je jednotková dolní trojúhelníková matice (na diagonále jsou jedničky, poddiagonální prvky jsou obecně nenulové), U je horní trojúhelníková matice. Tento rozklad je jednoznačný, nazýváme ho *LU faktorizací*. Soustavu rovnic (2.1) lze (po případné pivotaci) přepsat

$$LUx = b$$

a řešení soustavy pomocí Gaussovy eliminace lze přepsat

$$Ax = b, \quad (2.8a)$$

$$L^{-1}Ax = L^{-1}b, \quad (2.8b)$$

$$Ux = c, \quad (2.8c)$$

kde matice L^{-1} je jednotková dolní trojúhelníková matice, která reprezentuje úpravy původní matice soustavy, U je matice horní trojúhelníkové soustavy a c je vektor pravých stran horní trojúhelníkové soustavy. Gaussova eliminace realizuje LU faktorizaci. Pokud známe přímo faktorizaci matice soustavy (známe matice L , U), řešení soustavy lze přepsat

$$LUx = b, \quad (2.9a)$$

$$Lc = b, \quad (2.9b)$$

$$Ux = c. \quad (2.9c)$$

Oba postupy jsou formálně zcela ekvivalentní, pouze u Gaussovy eliminace (2.8) v průběhu výpočtu provádíme LU faktorizaci, kterou v případě (2.9) již známe. Hledání řešení je rozloženo do několika kroků – nalezení faktorizace, řešení dolní trojúhelníkové soustavy *dopředným (přímým) chodem* (2.9b), řešení horní trojúhelníkové soustavy *zpětným chodem* (2.8c), resp. (2.9b). Hledání faktorizace a řešení horní trojúhelníkové soustavy je u Gaussovy eliminace vzájemně provázáno v jednom kroku (2.8b).

Bunch-Parletův a Choleského rozklad (LL^T faktorizace)

Je-li matice A soustavy (2.1) symetrická, je možné této symetrie využít a místo LU rozkladu (2.7) má matice rozklad

$$A = LDL^T, \quad (2.10)$$

kde L je jednotková dolní trojúhelníková matice, D je diagonální matice. Tento rozklad je jednoznačný, nazýváme ho *Bunch-Parletův rozklad (faktorizace)*.

Je-li matice A soustavy (2.1) pozitivně definitní (pro libovolné x platí $x^T Ax > 0$, viz [2.3; kap. 2]), má rozklad

$$A = LL^T, \quad (2.11)$$

kde L je jednotková dolní trojúhelníková matice. Tento rozklad je jednoznačný, nazýváme ho *Choleského rozklad*.

Provedením Bunch-Parletova, resp. Choleského rozkladu matice řešíme soustavu (2.1) obdobně jako u Gaussovy eliminace.

2.1.3 Stacionární iterační metody řešení soustavy rovnic (2.1)

Při iteračním řešení soustav lineárních rovnic se nejprve pokusíme nějakým způsobem odhadnout přesné řešení x aproximací řešení $x^{(0)}$. Tento odhad postupně v jednotlivých krocích zpřesňujeme. Obecné schéma zpřesňování odhadu je u lineárních iteračních metod

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + H^{(k)} r^{(k)} = x^{(k)} + H^{(k)} (b - Ax^{(k)}) = \\ &= (I - H^{(k)} A) x^{(k)} + H^{(k)} b = B^{(k)} x^{(k)} + H^{(k)} b, \end{aligned} \quad (2.12)$$

kde horní index v závorkách označuje číslo iterace, vektor r je tzv. *reziduum* a matice $H^{(k)}$, resp. $B^{(k)}$ je matice charakterizující jednotlivé metody. Matice $B^{(k)}$ se nazývá *iterační matice*.

Reziduum v k -tém kroku (iteraci) je rozdíl k -té aproximace řešení $x^{(k)}$ a skutečného řešení x úlohy zleva násobený maticí A , píšeme

$$r^{(k)} = A(x - x^{(k)}) = Ax - Ax^{(k)} = b - Ax^{(k)}.$$

Pomocí matice $H^{(k)}$, resp. $B^{(k)}$ také rozdělujeme metody do dvou základních skupin. Jsou to *lineární stacionární a nestacionární iterační metody*. Stacionární metody jsou takové, u kterých se tyto matice nemění v jednotlivých iteracích, platí tedy $H^{(k)} = H^{(k+1)}$, resp. $B^{(k)} = B^{(k+1)}$. U nestacionárních metod se tyto matice mění závislosti na iteraci.

Stacionární iterační metody

U stacionárních metod tedy platí, že H , resp. B jsou konstantní matice, nezávislé na iteraci. Vztah (2.12) přepíšeme

$$x^{(k+1)} = x^{(k)} + Hr^{(k)} = (I - HA)x^{(k)} + Hb = Bx^{(k)} + Hb. \quad (2.12a)$$

Nyní definujeme vhodný rozklad matice A

$$A = M - N,$$

kde matice M je snadno invertovatelná. Pro soustavu (2.1) platí

$$\begin{aligned} Ax &= b, \\ Mx &= Nx + b, \end{aligned}$$

definujeme iterační vzorec

$$\begin{aligned} Mx^{(k+1)} &= Nx^{(k)} + b, \\ x^{(k+1)} &= M^{-1}Nx^{(k)} + M^{-1}b. \end{aligned} \quad (2.12b)$$

Z porovnání iteračních vzorců (2.12a), (2.12b) vyplývá vztah mezi maticemi H , M . Pomocí matice B ověříme konzistenci obou vzorců,

$$\begin{aligned} H &= M^{-1}, \\ B &= I - HA = I - M^{-1}(M - N) = I - I + M^{-1}N = M^{-1}N. \end{aligned}$$

Matici H , resp. M nelze zvolit libovolně, obzvlášť má-li být snadno invertovatelná. Rozborem iteračního vzorce (2.12a) lze pro matici H odvodit omezení. Chceme, aby platilo

$$\lim_{k \rightarrow \infty} x^{(k)} = x,$$

musí být tedy splněna podmínka

$$\lim_{k \rightarrow \infty} (I - HA)^k = \lim_{k \rightarrow \infty} B^k = 0. \quad (2.13)$$

Tato podmínka je splněna, pokud

$$\rho(B) \equiv \max \{ \lambda : Bx = \lambda x, x \neq 0 \} < 1, \quad (2.14)$$

tedy pokud je *spektrální poloměr* ρ matice B (v absolutní hodnotě největší vlastní číslo matice B) menší než jedna.

Konkrétní stacionární iterační metody

Než zmíníme některé konkrétní stacionární iterační metody, zavedeme rozklad matice A soustavy (2.1) na součet tří matic,

$$A = D - L - U, \quad (2.15)$$

kde D je diagonální matice (obsahuje pouze diagonální prvky matice A) a L , resp. U je striktně dolní, resp. striktně horní trojúhelníková matice (obsahuje pouze pod-, resp. naddiagonální záporně vzaté prvky matice A).

Nyní uvedeme tvar matice H pro některé klasické stacionární iterační metody:

Richardsonova metoda $H = \omega I = \text{diag}[\omega, \omega, \dots, \omega], \quad \text{kde } \omega > 0,$

Jacobiho metoda $H = D^{-1} = \text{diag}[1/a_{11}, 1/a_{22}, \dots, 1/a_{nn}]$

Gauss-Seidlova metoda $H = (D - L)^{-1},$

SOR – Superrelaxační metoda $H = (D - \omega L)^{-1}, \quad \text{kde } \omega \in (0, 2),$

(matice musí být regulární).

Analýza konvergence pro jednotlivé metody viz [2.1; kap. 10.2], [2.3; kap. 12]. Stacionárních metod se postupem času vytvořilo velké množství, existují varianty metod, specializované pro strukturované úlohy (například *SSOR* pro symetrické úlohy).

2.1.4 Nestacionární iterační metody, projekční metody

Nestacionární a semiiterační metody – urychlení konvergence

Výše popsané iterační metody většinou k přesnému řešení konvergují *asymptoticky*. To znamená, že metoda dokončuje k řešení (teoreticky) po nekonečně mnoha iteracích. V praxi nám stačí řešení s nějakou konečnou přesností, proto je možné metodu ukončit po konečně mnoha krocích. Nicméně právě asymptotická konvergence stacionárních metod může při požadavku rozumné přesnosti řešení vést na velké množství počítaných iterací. Konvergence tak může být pro rozsáhlé úlohy velmi zdlouhavá. Významné urychlení konvergence přinášejí *nestacionární a semiiterační metody*.

Nestacionární metody jsou charakteristické tím, že se matice $H^{(k)}$, resp. $B^{(k)}$ mění v závislosti na iteraci. Příkladem takové nestacionární metody může být varianta výše popsané superrelaxační metody (SOR), v níž použijeme parametr ω_k , který se bude měnit v závislosti na iteraci. Hlavním cílem variace parametrů u nestacionárních a semiiteračních metod je urychlení konvergence těchto metod, viz [2.1; kap. 10.4].

Projekční metody – základní princip

Základní princip *projekčních metod* je zobrazení rozsáhlé úlohy na úlohu menší, snadněji řešitelnou. Provádíme tedy projekci úlohy z původního n -rozměrného prostoru do prostoru menšího, m -rozměrného, kde tuto úlohu řešíme. Snažíme se, aby platilo

$$m \ll n.$$

Řešení redukované m -rozměrné úlohy zpětně zobrazíme do původního n -rozměrného prostoru, tím získáme řešení původní n -rozměrné úlohy.

Nyní popíšeme projekční metody konkrétně. Máme úlohu (2.1)

$$Ax = b, \quad \text{přičemž} \quad A \in R^{n,n},$$

hledáme aproximaci řešení x_K v m -rozměrném podprostoru K prostoru R^n (dolní index K zde neznačí iteraci ale podprostor, v kterém aproximaci hledáme). Pro určení této aproximace x_K potřebujeme m nezávislých podmínek. Tyto podmínky jsou podmínky ortogonality rezidua r_K na bázi určitého m -rozměrného podprostoru L . Reziduum r_K je tedy kolmé na podprostor L . Tyto podmínky se nazývají *Petrov-Galerkinovy podmínky*, píšeme

$$\begin{aligned} x_K \in K, & \quad \text{kde} \quad K = R(U) = \text{span}\{u_1, \dots, u_m\}, \\ b - Ax_K \perp L, & \quad \text{kde} \quad L = R(V) = \text{span}\{v_1, \dots, v_m\}. \end{aligned}$$

Matice U , resp. V jsou matice typu $n \times m$, jsou tedy tvořeny m n -rozměrnými sloupcovými vektory. Aproximace řešení x_K lze pomocí matice U vyjádřit

$$x_K = U\hat{x},$$

kde \hat{x} je řešení redukované soustavy. Z ortogonality rezidua a podprostoru L dostaneme

$$V^T(b - Ax_K) = 0, \quad \text{po roznásobení} \quad V^T b - V^T A U \hat{x} = 0.$$

Pro výpočet aproximace řešení x_K původní n -rozměrné soustavy nyní potřebujeme nalézt řešení m -rozměrné soustavy rovnic

$$\hat{b} - \hat{A}\hat{x} = 0, \quad \text{kde} \quad \hat{b} = V^T b, \quad \hat{A} = V^T A U;$$

viz obrázek 2.1.

Princip projekce nyní jednoduchým způsobem zakomponujeme do schématu iteračních metod. Nyní budeme značit x přesné řešení úlohy, x_K aproximaci řešení v podprostoru K (tu se snažíme nalézt), $x^{(p)}$ aproximaci řešení v podprostoru K v p -té iteraci. Můžeme psát

$$x_K = x^{(p)} + d^{(p)}, \quad \text{kde} \quad d^{(p)} = A^{-1}r^{(p)},$$

přičemž $r^{(p)}$ je reziduum v p -té iteraci.

Předpokládejme, že platí

$$d^{(p)} \in K \quad \text{a} \quad b - A(x^{(p)} + d^{(p)}) \perp L.$$

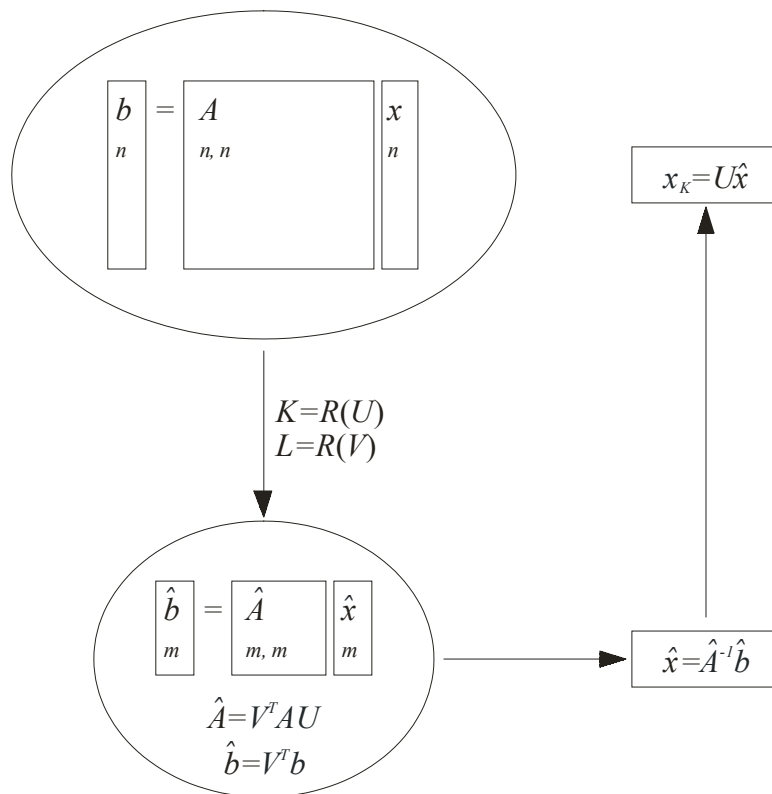
Platí následující vztahy

$$x^{(p)} = U\hat{x}^{(p)}, \quad \hat{r}^{(p)} = \hat{b} - \hat{A}\hat{x}^{(p)}, \quad \hat{d}^{(p)} = \hat{A}^{-1}\hat{r}^{(p)} = (V^T A U)^{-1} V^T r^{(p)}.$$

Iterační vzorec je následující

$$x^{(p+1)} = x^{(p)} + U\hat{d}^{(p)} = x^{(p)} + U(V^T A U)^{-1} V^T r^{(p)}.$$

Obecně u iterační metody mohou být podprostory L a K různé v různých iteracích, tedy $L = L^{(p)}$ a $K = K^{(p)}$ pro $p = 1, 2, \dots$. Je-li matice soustavy A symetrická, oba podprostory mohou být stejné (tedy $L = K$ a $U = V$); viz například [2.1; kap. 10.5].



Obrázek 2.1; Základní schéma fungování projekčních metod.

Metody Krylovových podprostorů – metoda konjugovaných gradientů (CG)

Metody Krylovových podprostorů (někdy též krylovovské metody) a zejména metody založené na CG jsou v současnosti standardem pro řešení soustav lineárních rovnic, jsou vhodné pro řešení rozsáhlých řídkých symetrických úloh. Jedná se o projekční metody, u kterých posloupnosti podprostorů $L^{(k)}$ a $K^{(k)}$ nahradíme posloupností *Krylovových podprostorů*

$$K_k(r^{(0)}, A) = \text{span}\{r^{(0)}, Ar^{(0)}, A^2r^{(0)}, \dots, A^{k-1}r^{(0)}\},$$

tedy $L^{(k)} = K^{(k)} = K_k(r^{(0)}, A)$, $k = 1, 2, \dots$, kde $r^{(0)}$ je počáteční reziduum (tedy $r^{(0)} = b - Ax^{(0)}$).

Jednou z těchto metod je *metoda konjugovaných gradientů – CG*. Tato metoda je fundamentální metodou pro řešení soustav rovnic se symetrickou pozitivně definitní maticí. Metoda CG hledá řešení v každém kroku (v každé iteraci) ve směru $p^{(k)}$, který je dán lineární kombinací rezidua a předchozího směru $p^{(k-1)}$. Platí tedy

$$p^{(k+1)} = r^{(k+1)} + \beta^{(k)} p^{(k)}, \quad k = 0, 1, 2, \dots, \quad \text{přičemž} \quad p^{(0)} = r^{(0)}.$$

Z požadavku vzájemné A -ortogonalita jednotlivých směrů $p^{(k)}$

$$(p^{(k)}, Ap^{(i)}) = 0, \quad i = 0, 1, 2, \dots, k-1.$$

vyplývá požadavek na parametr $\beta^{(k)}$. Vektory $p^{(k-1)}$ a $r^{(k-1)}$ leží v Krylovově podprostoru $K_k(r^{(0)}, A)$, přičemž vektory $p^{(0)}, \dots, p^{(k-1)}$ tvoří A -ortogonální bázi tohoto podprostoru. Lze ukázat, že k -té reziduum $r^{(k)}$ je ortogonální na podprostor $K_k(r^{(0)}, A)$, všechna rezidua jsou tedy vzájemně ortogonální

$$(r^{(k)}, r^{(i)}) = 0, \quad i = 0, 1, 2, \dots, k-1.$$

Tento vztah může platit maximálně pro n nenulových reziduí, pro $r^{(k)} \neq 0$, $k = 0, 1, \dots, n-1$. Pro $k = n$ může být splněn pouze pokud $r^{(n)} = 0$. Metoda CG tedy konverguje k řešení maximálně v n krocích; viz [2.1; kap. 10.6].

Algoritmus realizující metodu CG:

```

 $r^{(0)} = b - Ax^{(0)}; \quad p^{(0)} = r^{(0)};$ 
for  $k = 0, 1, 2, \dots$  while  $\|r^{(k)}\|_2 > \varepsilon$  do
     $\alpha^{(k)} = (r^{(k)}, r^{(k)}) / (p^{(k)}, Ap^{(k)});$ 
     $x^{(k+1)} = x^{(k)} + \alpha^{(k)} p^{(k)};$ 
     $r^{(k+1)} = r^{(k)} - \alpha^{(k)} Ap^{(k)};$ 
     $\beta^{(k)} = (r^{(k+1)}, r^{(k+1)}) / (r^{(k)}, r^{(k)});$ 
     $p^{(k+1)} = r^{(k+1)} + \beta^{(k)} p^{(k)};$ 
end.

```

Číslo ε v algoritmu označuje přesnost, se kterou budeme řešení hledat. Toto číslo zde pozbývá smyslu, neboť v přesné aritmetice je již n -té reziduum vždy nulové, můžeme tedy dát $\varepsilon = 0$. Je-li algoritmus implementován v aritmetice s konečnou přesností (tedy na počítači), je $\varepsilon > 0$ (viz kapitolu 3).

Metoda CG byla poprvé popsána Hestensem a Stiefelem v článku [2.4]. V tomto článku byl také poprvé popsán výše zmiňovaný algoritmus. Více podrobností k metodě CG nalezneme právě v [2.4], nebo například v [2.5]. Popis metody včetně algoritmu nalezneme také v [2.1; kap. 10.6].

2.2 Problém nejmenších čtverců

Problém nejmenších čtverců je zobecněním soustavy lineárních rovnic. Úlohu, kterou je zde potřeba řešit, zapíšeme takto

$$Ax \approx b. \quad (2.16)$$

Úloha (2.16) je zobecněním úlohy (2.1), matice A je obdélníková rozměrů $n \times m$, jejíž hodnost je $r \leq \min \{n, m\}$, vektor x je m -rozměrný vektor neznámých a vektor b je n -rozměrný vektor pravých stran. Pokud je $m > n$, říkáme, že soustava (2.16) je *nedourčená* (má více neznámých než rovnic). Pokud je $m < n$, říkáme, že soustava (2.16) je *přeuročena* (má více rovnic než neznámých).

2.2.1 Existence řešení a jednoznačnost řešení úlohy (2.16)

Pokud je splněna podmínka

$$b \in R(A), \quad (2.17)$$

pak lze úlohu (2.16) řešit přesně, úloha má *klasické řešení*, existuje tedy takové x , že platí $Ax = b$. Je-li navíc splněna podmínka

$$N(A) = \{0\}, \quad (2.18)$$

pak řešení x je *řešením jednoznačným*. Pokud podmínka (2.18) splněna není a matice A má netriviální nulový prostor, existuje nekonečně mnoho klasických řešení. Libovolné z těchto *obecných řešení*, značíme x_O , lze rozložit na součet

$$x_O = x_P + x_N, \quad x_P \perp N(A), \quad x_N \in N(A),$$

kde x_P je *partikulární řešení*, x_N je vektor z nulového prostoru matice, platí

$$Ax_O = A(x_P + x_N) = Ax_P + Ax_N = b + 0 = b.$$

2.2.2 Řešení úlohy (2.16) ve smyslu nejmenších čtverců a řešení minimální v normě

V případě, že vektor b neleží v prostoru hodnot matice A (není splněna podmínka (2.17)), přesné řešení úlohy (2.16) neexistuje. Tato situace může nastat u přeuredených soustav. Hledáme takové řešení x , aby byla splněna podmínka minimálního rezidua

$$\|b - Ax\| = \min_{u \in R^m} \|b - Au\|. \quad (2.19)$$

Toto řešení x nazveme *řešením ve smyslu nejmenších čtverců*. Je-li splněna podmínka (2.18) pak je toto řešení jednoznačné, pokud splněna není, existuje nekonečně mnoho řešení ve smyslu nejmenších čtverců (k danému x , které splňuje podmínku (2.19), lze přičíst libovolný vektor x_N z nulového prostoru matice A , tento součet podmínku (2.19) splní též).

Z důvodů jednoznačnosti řešení úlohy (2.16) zavádíme *řešení ve smyslu nejmenších čtverců minimální v normě*. To je takové řešení x splňující podmínku (2.19), které má minimální normu

$$\|x\| = \min \left\{ \|v\| : \|b - Av\| = \min_{u \in R^m} \|b - Au\| \right\}. \quad (2.20)$$

Takto definované řešení x úlohy (2.16) je vždy jednoznačné a platí $x \perp N(A)$.

2.2.3 Metody řešení úloh nejmenších čtverců

Úlohu nejmenších čtverců (2.16) lze různými způsoby převést na soustavu lineárních rovnic (2.1), tedy na úlohu, kterou již umíme řešit. Různé způsoby převodu úlohy realizují jednotlivé metody. Jednou z fundamentálních metod je *řešení rozšířené soustavy*.

Řešení rozšířené soustavy

Nejprve definujme reziduum

$$r \equiv b - Ax, \quad (2.21)$$

kde x je řešení ve smyslu nejmenších čtverců minimální v normě (vektor splňující (2.20)). Reziduum (2.21) bude nenulové tehdy, pokud vektor b neleží v oboru hodnot matice A , tedy pokud nelze nalézt klasické řešení soustavy (2.16). Vektor b lze rozložit na část, která leží v prostoru hodnot matice A , a na část, která leží v nulovém prostoru matice A^T . Mezi vektorem pravých stran b a reziduem r platí vztah

$$b = b|_{R(A)} + b|_{N(A^T)} = b|_{R(A)} + r.$$

Reziduum je tedy ta část vektoru b , která leží v nulovém prostoru matice A^T , platí tedy

$$A^T r = 0. \quad (2.22)$$

Rovnice (2.21), (2.22) lze zapsat v maticovém tvaru, výsledkem zápisu je soustava lineárních rovnic (2.1)

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}. \quad (2.23)$$

Matrice soustavy (2.23) je symetrická obecně indefinitní matice rozměru $(n+m) \times (n+m)$. Vyřešením soustavy (2.23) nalezneme řešení úlohy (2.16) ve smyslu nejmenších čtverců minimální v normě (2.20) a reziduum (2.21). Na řešení této soustavy je možné použít například Bunch-Parletův rozklad (viz kapitolu 2.1.2).

Řešení systému normálních rovnic

Další metodou je použití tzv. *systému normálních rovnic*. Tento postup je velmi výhodný pro řešení úloh, pro které platí $m \ll n$. Postup je takový, že celou soustavu (2.16) vynásobíme zleva maticí A^T . Protože reziduum (2.21) r je část vektoru b ležící v nulovém prostoru matice A^T , platí

$$A^T b = A^T b|_{R(A)} + A^T b|_{N(A^T)} = A^T b|_{R(A)} + A^T r = A^T b|_{R(A)} + 0$$

a soustava lineárních rovnic

$$A^T A x = A^T b. \quad (2.24)$$

již půjde vždy řešit jednoznačně. Řešení x soustavy (2.24) je řešením ve smyslu nejmenších čtverců minimální v normě.

Pokud platí $m \ll n$, tak jsme původní soustavu převedením na soustavu (2.24) velmi výrazně zmenšili. Systém normálních rovnic (2.24) je soustava lineárních rovnic s maticí soustavy řádu $m \times m$, tato matice je symetrická pozitivně definitní. K řešení soustavy (2.24) lze použít například Choleského rozklad (viz kapitolu 2.1.2).

Řešení systému normálních rovnic vede na řešení menší soustavy lineárních rovnic navíc se symetrickou pozitivně definitní maticí. Systém normálních rovnic má i svoje nevýhody. Pokud je matice A soustavy (2.16) špatně podmíněná (viz kapitolu 3.3.3), pak je špatně podmíněná i matice soustavy (2.24), neboť platí $\kappa(A^T A) = \kappa^2(A)$.

2.2.4 QR rozklad

Pomocí *QR rozkladu* lze řešit nejen úlohy nejmenších čtverců (2.16), ale lze pomocí něj řešit i soustavy lineárních rovnic (2.1). Řešení soustavy rovnic pomocí QR rozkladu se přitom řadí mezi přímé metody. Libovolnou matici A řádu $n \times m$ lze rozložit

$$A = QR, \quad (2.25)$$

kde Q je ortogonální matice (má vzájemně ortonormální sloupce, platí tedy $Q^T Q = I$, respektive $Q^{-1} = Q^T$) a R je horní trojúhelníková matice. QR rozklad matice lze provést velmi snadno (ukážeme vzápětí).

Mějme čtvercovou matici A a matice Q a R , všechny řádu $n \times n$ takové, že platí (2.25). Označme a_i sloupce matice A a q_i sloupce matice Q . Z násobení matic ve vztahu (2.25) jasně vyplývá, že

$$\begin{aligned} a_1 &= q_1 r_{11}, \\ a_2 &= q_1 r_{12} + q_2 r_{22}, \\ &\dots \\ a_n &= \sum_{i=1}^n q_i r_{in}, \end{aligned}$$

platí tedy vztah

$$\text{span}\{a_1, \dots, a_i\} = \text{span}\{q_1, \dots, q_i\}, \quad \forall i = 1, 2, \dots, n.$$

Postup výpočtu matic Q, R při provádění rozkladu matice A je následující,

$$\begin{aligned} r_{11} &= \|a_1\|, & q_1 &= a_1 / r_{11}, \\ r_{12} &= q_1^T a_2, & r_{22} &= \|a_2 - q_1 r_{12}\|, & q_2 &= (a_2 - q_1 r_{12}) / r_{22}, \\ &\dots & &\dots & &\dots \\ r_{in} &= q_i^T a_n, & r_{nn} &= \left\| a_n - \sum_{i=1}^{n-1} q_i r_{in} \right\|, & q_n &= \left(a_n - \sum_{i=1}^{n-1} q_i r_{in} \right) / r_{nn}. \end{aligned}$$

Řešení úlohy (2.1) pomocí QR rozkladu

Řešení soustavy lineárních rovnic (2.1) i úlohy nejmenších čtverců (2.16) pomocí QR rozkladu je snadné. Začneme nejprve soustavou lineárních rovnic, platí

$$Ax = QRx = b.$$

Pokud známe QR rozklad matice soustavy A , celou soustavu vynásobíme zleva maticí Q^T ,

$$Q^T QRx = Q^T b. \quad (2.26a)$$

$$Rx = c. \quad (2.26b)$$

Postup je tedy, známe-li rozklad, obdobný jako u LU faktorizace (viz kapitolu 2.1.2). Řešení soustavy lineárních rovnic se tak redukuje na řešení horní trojúhelníkové soustavy (2.26b).

Řešení úlohy (2.16) pomocí QR rozkladu

V případě řešení úlohy nejmenších čtverců je postup trochu složitější, nejprve je potřeba si ujasnit, jak bude vypadat QR rozklad v případě, že matice A má rozměry $n \times m$. Pokud $n > m$, můžeme psát

$$A = QR = [Q_1, \quad Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1,$$

přičemž matice Q je čtvercová matice $n \times n$, matice R je obdélníková matice rozměrů $n \times m$. Matice Q_1 je pak obdélníkový blok $n \times m$ a matice R_1 je horní trojúhelníkový blok $m \times m$. Pokud $n < m$, můžeme psát

$$A = QR = Q [R_2, \quad R_3],$$

přičemž matice Q je čtvercová matice $n \times n$, matice R je obdélníková matice rozměrů $n \times m$. Matice R_2 je horní trojúhelníkový blok $n \times n$.

Řešení úlohy (2.16) například pro $n > m$ je již formálně stejné. Pokud známe QR rozklad matice A , celou soustavu vynásobíme zleva maticí $Q^T = [Q_1, \quad Q_2]^T$

$$Q^T QRx = Q^T b, \quad (2.27a)$$

$$\begin{bmatrix} R_1 \\ 0 \end{bmatrix} x = \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} b, \quad (2.27b)$$

$$R_1 x = Q_1^T b, \quad (2.27c)$$

$$R_1 x = c. \quad (2.27d)$$

Matice Q_1 a Q_2 jsou vzájemně ortogonální, platí $R(Q_1) = R(A)$ a $R(Q_2) = N(A^T)$; viz [2.1; kap. 8.5.2]. Nalezení řešení minimálního v normě (2.20) se tak opět redukuje na řešení horní trojúhelníkové soustavy (2.27d).

2.3 Problém vlastních čísel

2.3.1 Problém vlastních čísel

Problém vlastních čísel je problém nalezení takových čísel λ , která splňují rovnost

$$Ax = \lambda x, \quad \text{pro } x \neq 0, \quad (2.28)$$

respektive

$$y^H A = \lambda y^H, \quad \text{pro } y \neq 0. \quad (2.29)$$

Tato čísla λ nazýváme *vlastní čísla matice A*. Vektory x nazýváme (*pravé*) *vlastní vektory matice A* a vektory y nazýváme *levé vlastní vektory matice A*. Množinu vlastních čísel matice A nazýváme *spektrum matice A*.

Pro libovolnou čtvercovou matici A řádu $n \times n$ definujeme *charakteristický polynom*

$$\det(A - \lambda I) = 0 \quad (2.30)$$

v proměnné λ . Vlastní čísla matice A jsou kořeny charakteristického polynomu. Matice má n vlastních čísel, ne všechna musí být různá. Nalezení vlastních čísel faktorizací polynomu ovšem není obecně možné.

2.3.2 Mocninná metoda hledání vlastních čísel

Mocninná metoda

Mocninná metoda je iterační postup, jak určit dominantní vlastní číslo matice (metodu lze použít v případě, že některé z vlastních čísel je dominantní, tedy v absolutní hodnotě výrazně větší, než jsou všechna ostatní). Předpokládejme, že vlastní čísla λ_i matice A řádu $n \times n$ splňují tuto nerovnost

$$|\lambda_1| \gg |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|. \quad (2.31)$$

Normalizované vlastní vektory x_i diagonalizovatelné matice A tvoří bázi prostoru R^n , libovolný vektor q je tedy možné vyjádřit jako lineární kombinaci těchto vektorů

$$q = \sum_{i=1}^n \alpha_i x_i. \quad (2.32)$$

Vynásobíme-li vektor q maticí A , platí

$$Aq = \sum_{i=1}^n \alpha_i Ax_i = \sum_{i=1}^n \alpha_i \lambda_i x_i.$$

Budeme-li vektor q násobit maticí A k -krát, platí

$$A^k q = \sum_{i=1}^n \alpha_i A^k x_i = \sum_{i=1}^n \alpha_i \lambda_i^k x_i \approx \alpha_1 \lambda_1^k x_1. \quad (2.33)$$

Díky tomu, že platí (2.31), získáme dominantní vlastní číslo λ_1 .

Inverzní iterace s posunutím (Shift + Inverse) a odhad vlastních čísel

Mocninnou metodu lze použít jen v případě, že mezi vlastními čísly matice platí vztah (2.31). Všechna vlastní čísla $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n$ obecné matice A však tímto postupem spočítat nelze. Pro určení takových vlastních čísel se používají tzv. *inverzní iterace s posunutím*. Podstata této metody je založena na vytvoření dominance ve spektru matice. Pokud máme odhad σ nějakého vlastního čísla matice A , spočítáme matici

$$A' = (A - \sigma I)^{-1}. \quad (2.34)$$

Pokud σ je dobrý odhad vlastního čísla matice A , matice $(A - \sigma I)$ má jedno vlastní číslo velmi blízké nule a inverzní matice A' má jedno dominantní vlastní číslo (absolutní hodnoty vlastních čísel matice odpovídají převráceným hodnotám absolutních hodnot vlastních čísel inverzní matice). Aplikováním mocninné metody na matici A' získáme vlastní číslo λ matice A , jemuž bylo číslo σ odhadem.

Otázka je, jak zvolit σ takové, aby bylo blízko nějakého konkrétního vlastního čísla matice A . Nejprve si zvolíme libovolný nenulový vektor q a pak určíme odhad σ vlastního čísla tak, aby tento odhad minimalizoval následující vztah

$$\min_{\sigma} \|Aq - \sigma q\|^2. \quad (2.35)$$

Tento výraz bude minimální pro odhad σ daný *Rayleighovým kvocientem*

$$\sigma_{\min} = \frac{(Aq, q)}{(q, q)} \quad (2.36)$$

a navíc pro vhodně zvolený vektor q (blízký vlastnímu vektoru matice A) bude σ blízko odpovídajícího vlastního čísla matice A . V praxi se Rayleighův kvocient σ_{\min} počítá iteračně

$$\sigma^{(k+1)} = \frac{(Aq^{(k)}, q^{(k)})}{(q^{(k)}, q^{(k)})}, \quad (2.37a)$$

kde vektor $q^{(k)}$ je normalizovaný vektor $w^{(k)}$,

$$q^{(k)} = \frac{w^{(k)}}{\|w^{(k)}\|}, \quad (2.37b)$$

vektor $w^{(k)}$ je řešením soustavy rovnic

$$(A - \sigma^{(k)}I)w^{(k)} = q^{(0)}, \quad (2.37c)$$

vektor $q^{(0)}$ je počáteční odhad.

2.3.3 Úplný problém vlastních čísel – QR algoritmus

Podobnostní a unitární transformace matice

Problém nalezení vlastních čísel matice A není možné řešit analyticky. Tento problém se tedy řeší iteračně pomocí tzv. *QR algoritmu*. Před popisem tohoto algoritmu je potřeba na úvod zavést několik pojmů. Matici A , jejíž vlastní čísla hledáme, je možné transformovat operací

$$PAP^{-1} = B$$

na určitou matici B , matice P je regulární *transformační matice*. Tato transformace se nazývá *podobnostní transformací*. Podobnostní transformace nemění charakteristický polynom matice, ani spektrum matice. Vlastní čísla matice A i matice B jsou stejná. Je-li transformační matice ortogonální, značíme Q , píšeme

$$QAQ^H = B,$$

pak je tato transformace *unitární transformací*. Existuje taková unitární transformace, která matici A převede na horní trojúhelníkovou matici,

$$QAQ^H = R, \quad (2.38)$$

kde R je horní trojúhelníková matice. Protože matice R má však stejná vlastní čísla jako původní matice A , leží tato vlastní čísla právě na její diagonále. Transformační matici Q není možné obecně analyticky nalézt, neboť bylo-li by to možné, šlo by obecně nalézt kořeny libovolného polynomu.

QR algoritmus

QR algoritmus je metoda, která se v podstatě snaží najít transformaci ze vztahu (2.38). Tuto transformaci hledá iteračně a používá k tomu QR rozklad (viz kapitulu 2.2.3). Nejprve matici A , jejíž vlastní čísla hledáme, konečným počtem podobnostních transformací (např. pomocí *Householderových reflexí* viz [2.1; kap. 8.5]) převedeme na matici B , která je tvořena horní trojúhelníkovou maticí a má navíc nenulovou první poddiagonálu. O takové matici řekneme, že je v *horním Hessenbergově tvaru*. Pokud má první poddiagonála všechny prvky nenulové, řekneme o matici, že je *úplným horním Hessenbergově tvaru* (*Proper Upper Hessenberg – PUH*). Pokud matice B není v PUH tvaru, lze ji přepsat jako blokově horní trojúhelníkovou matici, jejíž diagonální prvky jsou bloky v PUH tvaru.

Nyní máme matici B , která je v PUH tvaru a má stejná vlastní čísla jako matice A (pokud B v PUH tvaru není, následující algoritmus aplikujeme na jednotlivé diagonální bloky, které v PUH tvaru jsou). QR algoritmus popíšeme pomocí rekurzivních vztahů

$$B^{(0)} := B, \quad (2.39a)$$

$$B^{(k)} := R^{(k)} Q^{(k)}, \quad (2.39b)$$

kde matice $R^{(k)}$ a $Q^{(k)}$ jsou matice, které jsme získali pomocí QR rozkladu z matice $B^{(k-1)}$, tedy

$$B^{(k-1)} = Q^{(k)} R^{(k)}. \quad (2.39c)$$

Vztah mezi dvěma po sobě jdoucími maticemi $B^{(k)}$, $B^{(k-1)}$,

$$B^{(k)} = R^{(k)} Q^{(k)} = R^{(k)} B^{(k-1)} (R^{(k)})^{-1} = (Q^{(k)})^H B^{(k-1)} Q^{(k)}. \quad (2.40)$$

Vztah (2.40) je podobnostní, resp. unitární transformací, tudíž všechny matice $B^{(k)}$ mají stejná vlastní čísla jako matice A . Platí limita

$$\lim_{k \rightarrow \infty} B^{(k)} = R, \quad (2.41)$$

kde matice R je horní trojúhelníková matice ze vztahu (2.38); viz [2.1; kap. 9.7].

2.3.4 Zobecněný problém vlastních čísel

Klasický problém vlastních čísel je hledání takových čísel λ , která splní rovnost (2.28)

$$Ax = \lambda x, \quad \text{pro } x \neq 0.$$

Čísla λ jsou vlastní čísla matice A , jsou to kořeny polynomu (2.30)

$$\det(A - \lambda I) = 0.$$

Existuje také *zobecněný problém vlastních čísel*. Hledáme čísla λ , která splní rovnost

$$Ax = \lambda Ex, \quad \text{pro } x \neq 0. \quad (2.42)$$

Řekneme, že zobecněný problém vlastních čísel je regulární, resp. singulární, pokud je matice $(A - \lambda E)$ regulární, resp. singulární. Čísla λ , která jsou kořeny polynomu

$$\det(A - \lambda E) = 0, \quad (2.43)$$

jsou *zobecněná vlastní čísla* matice A . Matice A , E jsou řádu $n \times n$, polynom (2.43) je obecně stupně $p \leq n$ (v případě singulárního problému se $p = 0$). Řekneme, že matice A v problému (2.42) má $n-p$ zobecněných vlastních čísel $\lambda = \infty$. Zobecněný problém vlastních čísel lze řešit například mocninou metodou s inverzní iterací (viz [2.1; kap. 9.9.4]).

2.4 Singulární rozklad

2.4.1 Singulární rozklad – singulární čísla

Dalším z často užívaných nástrojů lineární algebry je takzvaný *singulární rozklad* (*Singular Values Decomposition – SVD*). Máme-li nějakou obecnou obdélníkovou matici A řádu $n \times m$, pak lze tuto matici A rozložit na součin tří matice

$$A = U \Sigma V^T, \quad (2.44)$$

kde matice U a V jsou ortogonální matice, matice U je řádu $n \times n$, matice V je řádu $m \times m$. Matice Σ je řádu $n \times m$ a obsahuje pouze čtvercový diagonální blok Σ_R řádu $r \times r$, platí

$$\Sigma_R = \text{diag}[\sigma_1, \sigma_2, \dots, \sigma_r], \quad (2.45)$$

přičemž

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0.$$

Číslo r je *hodnota matice A (rank)*, tedy počet lineárně nezávislých sloupců, resp. řádků. Pro hodnota matice platí

$$r \leq \min\{n, m\}. \quad (2.46)$$

Nenulová čísla σ_i nazýváme *singulární čísla* matice A .

2.4.2 Ekonomická forma singulárního rozkladu

Matice Σ obsahuje velké množství nulových řádků (pro $n > m \geq r$), respektive sloupců (pro $r \leq n < m$). Singulární rozklad (2.44) se často přepisuje v tzv. *ekonomické formě*

$$A = U_R \Sigma_R V_R^T, \quad (2.47)$$

kde matice U_R je matice $n \times r$ a je tvořena prvními r sloupci matice U , matice V_R je matice rozměrů $m \times r$ a je tvořena prvními r sloupci matice V . Matice Σ_R je blok (2.45) matice Σ , má rozměry $r \times r$. Ekonomickou formu singulárního rozkladu je možné rozepsat pro jednotlivé vektory u_i a v_i matic U, V

$$A = \sum_{i=1}^r u_i \sigma_i v_i^T = \sum_{i=1}^r \sigma_i u_i v_i^T \equiv \sum_{i=1}^r \sigma_i A_i. \quad (2.48)$$

Matici A tak rozložíme na součet r matic, které jsou tvořeny vnějšími součiny vektorů u_i, v_i a singulárními čísly σ_i . Vztahem (2.48) matici A vyjádříme jako součet řezů $\sigma_i A_i$ s hodnotí 1 a s normou danou velikostí příslušného singulárního čísla σ_i .

2.4.3 Vztah singulárních a vlastních čísel

Vztah (2.44) je možné přepsat ve tvaru

$$AV = U \Sigma . \quad (2.49a)$$

Pokud vztah (2.44) nejprve transponujeme a vynásobíme jej maticí U zprava, je možné ho přepsat ve tvaru

$$A^T U = V \Sigma^T . \quad (2.49b)$$

Nyní (2.49a), (2.49b) vyjádříme v ekonomickém tvaru a přepíšeme je pro jednotlivé vektory, dostaneme tak (postupně)

$$Av_i = \sigma_i u_i , \quad (2.50a)$$

$$A^T u_i = \sigma_i v_i , \quad \text{pro } i = 1, 2 \dots r . \quad (2.50b)$$

Pokud (2.50a), resp. (2.50b) vynásobíme zleva maticí A^T , resp. maticí A , dostaneme po několika málo dalších úpravách vztahy

$$A^T Av_i = \sigma_i^2 v_i \quad (2.51a)$$

$$AA^T u_i = \sigma_i^2 u_i . \quad \text{pro } i = 1, 2 \dots r \quad (2.51b)$$

Čísla σ_i^2 (druhé mocniny singulárních čísel) jsou tedy vlastní čísla matic $A^T A$, AA^T , vektory v_i , resp. u_i (sloupce matice U , resp. V) jsou vlastní vektory matice $A^T A$, resp. AA^T , (viz (2.28)).

3. Problémy s implementací – numerická lineární algebra

Úlohy lineární algebry jsou základní nástroje nejen v teorii řízení a teorii dynamických systémů, ale i v mnoha dalších oblastech inženýrské činnosti. Velké množství reálných fyzikálních problémů vede na tak složité úlohy, že je neumíme analyticky řešit. Tyto úlohy (většinou spojité v čase i v prostoru) diskretizujeme a po částech linearizujeme, například pomocí FEM. Diskretizovaná úloha pak vede na rozsáhlou soustavu lineárních rovnic nebo na problém vlastních čísel. Základními matematickými postupy, jak tyto lineární algebraické úlohy řešit, jsme se zabývali v druhé kapitole. Matice, figurující algebraických úlohách, jsou často strukturované (důsledkem vhodné diskretizace), a díky snaze co nejpřesněji popsat fyzikální problém jsou tyto matice navíc velmi rozsáhlé. Strukturovanost matic, s kterými pracujeme, lze často v algoritmech využít. Vznikají tak algoritmy specializované na algebraické úlohy s konkrétní strukturou, viz například [2.1; kap. 7.8]. Díky velkému rozsahu algebraických úloh je potřeba studovat numerické vlastnosti vyvíjených algoritmů.

Numerické vlastnosti algoritmů studujeme proto, že tyto algebraické úlohy řešíme s využitím výpočetní techniky a každý počítač pracuje jen s omezenou přesností. Při všech operacích dochází k vzniku zaokrouhlovacích chyb, které se šíří výpočtem a projeví se v řešení. U kvalitních algoritmů, resp. implementací je chyba řešení pouze na úrovni základního zaokrouhlování (tzv. strojová přesnost). Řešení nalezené nekvalitním algoritmem může být zatíženo tak velkou chybou, že je zcela nepoužitelné.

V této kapitole budeme hovořit právě o zobrazování reálných čísel v počítači, o analýze stability algoritmů (kapitola 3.3.5) a o podmíněnosti dat vstupujících do algoritmu (kapitola 3.3.2). Analýzu algoritmů a jejich optimalizaci, studium vzniku a šíření zaokrouhlovacích chyb, studium numerických vlastností matic, atd. nazýváme *numerickou lineární algebrou*.

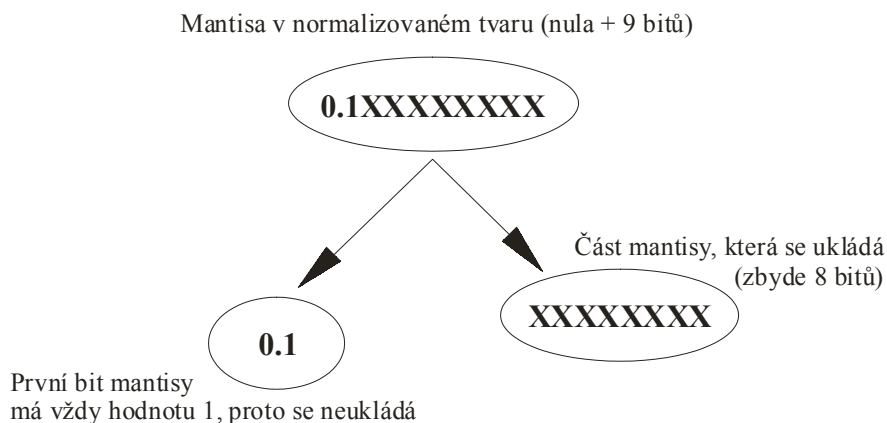
V textu této kapitoly je řešeno několik konkrétních triviálních příkladů zejména z důvodů přehledné prezentace základních pojmů numerické lineární algebry. Řadu informací a příkladů včetně analýzy konkrétních algoritmů nalezneme v literatuře, viz [3.3], [3.4].

3.1 Reálná čísla v počítači

Tento text si neklade za cíl věnovat se důkladněji otázkám zobrazení reálných čísel v počítači. Přesnější popis ukládání a práce s reálnými čísly si může zájemce nalézt v literatuře. Běžné počítače PC používají formát daný standardem IEEE, jiné počítače (respektive procesory) mohou používat i jiný způsob ukládání. My se budeme zabývat pouze způsobem, který je definován normou IEEE, a navíc velmi okrajově (viz např. [3.1], [3.4]).

Množinu reálných čísel nelze v počítači celou zobrazit. Počítač pracuje pouze s konečnou přesností, místo reálných čísel používá čísla racionální. Ale ani množinu racionálních čísel nelze v počítači zobrazit celou. Množinu reálných čísel tak v počítači nahrazujeme jistou konečnou podmnožinou množiny čísel racionálních, to je základem vzniku zaokrouhlovacích chyb při výpočtech. Tuto konečnou množinu budeme nazývat *množinou čísel zobrazitelných* na počítači, čísla z této množiny budeme nazývat *zobrazitelnými čísly*.

Ukládání zobrazitelných čísel v počítači je možné v *jednoduché (normální) přesnosti (single)*, resp. v *dvojitě přesnosti (double)*. Pro uložení takového čísla jsou použity čtyři, resp. osm bytů; číslo je tedy uloženo na 32, resp. 64 bitech. Ukládá v tzv. normalizovaném tvaru, ten je definován. Číslo je v tomto tvaru složeno ze tří částí *znaménko*, *mantisa* a *exponent*. Hodnota takto zakódovaného čísla se určí jako součin mantisy a základu umocněného na exponent (samozřejmě s ohledem na příslušné znaménko). Jako základ se u počítače používá číslo 2. Mantisa čísla v normalizovaném tvaru má desetinnou tečku je před svým nejvyšším nenulovým řádem. Vzhledem k tomu, že mantisa je uložena přirozeně v dvojkové soustavě, tento nejvyšší nenulový řád má vždy hodnotu 1, proto se neukládá (viz obrázek 3.1).



Obrázek 3.1; Mantisa vyjádřená ve dvojkové soustavě.

Exponent je kladné, resp. záporné celé číslo, které specifikuje o kolik řádů je potřeba desetinnou čárku v mantise posunout vpravo, resp. vlevo (absolutní hodnota zobrazovaného čísla je větší, resp. menší než 1). Důležité je upozornit, že exponent nemůže být u normalizovaných čísel tvořen samými jedničkami, resp. samými nulami. Tyto dva speciální exponenty se využívají u tzv. nenormalizovaných čísel (o nich se zmíníme v této kapitole později). Pro uložení znaménka čísla se používá 1 bit (nultý) v normální i v dvojitě přesnosti. Pro uložení exponentu se použije 8, resp. 11 bitů. Mantisa má 24, resp. 53 bitů, pro její uložení však stačí pouze 23, respektive 52 bitů (viz obrázek 3.1).

Kromě normalizovaných čísel existuje ještě řada čísel nenormalizovaných. Existují dvě speciální „čísla“. Prvním speciálním „číslem“ je *Inf (infinite)* a značí plus resp. minus nekonečno – exponent obsahuje samé jedničky, mantisa samé nuly a znaménkový bit značí, zda jde o plus, resp. minus nekonečno. Toto „číslo“ vznikne při přetečení v průběhu výpočtu (výsledek operace je v absolutní hodnotě větší než největší normalizované číslo), nebo při dělení nenulového čísla nulou. Druhým speciálním „číslem“ je *NaN (not a number)* a značí, že výsledek operace není číslo, resp. reálné číslo. Toto „číslo“ je tedy výsledkem při vyčíslení operací $0/0$, $0 \cdot \infty$, ∞/∞ , resp. při vyčíslení operace $\sqrt{-1}$. Kódování tohoto „čísla“ je následující – exponent obsahuje samé jedničky, mantisa obsahuje libovolnou nenulovou kombinaci, znaménkový bit je také libovolný. Tato dvě čísla využívají jednu ze zakázaných kombinací exponentu (viz normalizovaná čísla) – exponent tvořený samými jedničkami.

Dále existují čísla, která se ukládají v nenormalizovaném tvaru, tzv. *subnormální čísla*. Jsou to velmi malá čísla, v absolutní hodnotě menší než nejmenší normalizované číslo. U subnormálních čísel se znaménkový bit využívá standardně, exponent v podstatě také, ovšem v exponentu jsou samé nuly. Mantisa čísla se ukládá celá (u normalizovaných čísel se

mantisa ukládala bez prvního bitu). Při ukládání subnormálních čísel v normalizovaném tvaru by došlo k podtečení exponentu. Subnormální čísla využívají druhou ze zakázaných kombinací exponentu (viz normalizovaná čísla) – exponent tvořený samými jedničkami. Výjimku tvoří číslo 0, to je také subnormální číslo, v normalizovaném tvaru zřejmě nelze vyjádřit. Číslo 0 se však kóduje jako 32, resp. 64 nul.

V počítači uložená čísla nazýváme čísla *s plovoucí desetinnou čárkou (floating point)*. Operace s nimi v aritmetice s konečnou přesností nazýváme *operace v plovoucí desetinné čárce (floating point operation)*. Obraz reálného čísla a v počítači značíme

$$\tilde{a} = \text{fl}(a). \quad (3.1)$$

Vyčíslený výsledek nějaké operace (např. sčítání) provedené na dvou reálných číslech a, b zobrazených v počítači značíme

$$\tilde{c} = \text{fl}(\tilde{a} + \tilde{b}). \quad (3.2)$$

Další informace ohledně ukládání čísel a manipulace s čísly v počítači (množiny zobrazitelných čísel, vyčíslení základních aritmetických operací, atd.) viz [3.4].

3.2 Relativní strojová přesnost

Při zobrazování reálných čísel na čísla zobrazitelná v počítači dochází k vzniku chyb. Existuje zobrazitelné číslo $\varepsilon > 0$, pro které platí, že jeho součet s jedničkou v aritmetice s konečnou přesností je roven opět jedničce. Existuje právě jedno číslo ε_M , největší z čísel ε . Toto číslo nazýváme *relativní strojovou přesností (relative machine precision)*, platí

$$\text{fl}(1 + \varepsilon_M) = \text{fl}(1) = 1. \quad (3.3)$$

Relativní strojová přesnost klasického počítače (PC) při použití dvojitě přesnosti je přibližně $\varepsilon_M \approx 2^{-53}$, tedy $\varepsilon_M \approx 1.11 \times 10^{-16}$.

Zobrazením reálného čísla a do množiny čísel zobrazitelných se dopustíme relativní chyby δ a platí

$$\tilde{a} = a(1 + \delta), \quad \text{přičemž} \quad |\delta| \leq \varepsilon_M. \quad (3.4)$$

Pro vyčíslený výsledek nějaké operace (např. násobení) provedené na dvou číslech zobrazených v počítači platí

$$\text{fl}(\tilde{a} \cdot \tilde{b}) = \tilde{a} \cdot \tilde{b} \cdot (1 + \delta), \quad \text{přičemž} \quad |\delta| \leq \varepsilon_M; \quad (3.5)$$

podrobněji viz [3.2; str. 4].

3.3 Numerická stabilita a podmíněnost

S pojmem stabilita jsme se v tomto textu již setkali v první kapitole týkající se teorie řízení. V teorii řízení a v teorii dynamických systémů stabilitu definujeme omezeností reakce na omezený podnět. V numerické lineární algebře budeme pojem stabilita chápat odlišně (viz kapitolu 3.3.5).

3.3.1 Příklad na začátek – Gaussova eliminace [3.2; str. 6; př. 2]

Na úvod spočítáme jednoduchý příklad. Pokusíme se pomocí Gaussovy eliminace bez pivotace (klasický, nejjednodušší algoritmus Gaussovy eliminace) nalézt řešení jednoduché soustavy lineárních rovnic

$$Ax = b.$$

O tomto algoritmu je známo, že je numericky nestabilní, viz [3.2]. Na tomto příkladu budeme ilustrovat důsledky numerické nestability algoritmu. Mějme tedy

$$A = \begin{bmatrix} 1.000 \times 10^{-4} & 1.000 \\ 1.000 & -1.000 \end{bmatrix}, \quad b = \begin{bmatrix} 1.000 \\ 0.000 \end{bmatrix}.$$

Uvažujme nyní aritmetiku, v níž má libovolné číslo pouze čtyři platné číslice (v desítkové soustavě). Výsledky všech výpočtů tak budeme zaokrouhlovat na čtyři platné číslice. Pro počítač s takovou aritmetikou platí $\varepsilon_M = 0.0001$. Řešení této úlohy v dané aritmetice je (přesné řešení následuje)

$$x = A^{-1}b = \begin{bmatrix} 0.9999 \\ 0.9999 \end{bmatrix} \dots \begin{bmatrix} 0.99990001 \\ 0.99990001 \end{bmatrix}.$$

Použijeme-li k řešení úlohy Gaussovu eliminaci bez pivotace, postup bude následující:

$$[2. \text{ řádek}] := [2. \text{ řádek}] - 10000 \times [1. \text{ řádek}];$$

získáme tak soustavu v horním trojúhelníkovém tvaru

$$\begin{bmatrix} 1.000 \times 10^{-4} & 1.000 \\ 0.000 & -1.000 \times 10^4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.000 \\ -1.000 \times 10^4 \end{bmatrix}.$$

Začneme postupně vyčíslovat řešení horní trojúhelníkové soustavy. Platí $x_2 = 1.000$, což je dobrá aproximace přesného řešení, chyba je dokonce menší než relativní strojová přesnost. Nyní dosadíme takto vypočtené x_2 zpět a pokusíme se nalézt x_1 . Platí $x_1 = 0.000$, což je velmi špatná aproximace přesného řešení, je chyba téměř desetitisícinásobek relativní strojové přesnosti. Tak velká chyba je důsledek numerické nestability použitého algoritmu (Gaussovy eliminace bez pivotace).

Jak jsme na předchozím triviálním příkladu ukázali, i u velmi jednoduché úlohy lze nalézt velmi špatné řešení, pokud zvolíme nevhodný algoritmus výpočtu. Pro ilustraci nestability tohoto algoritmu na reálném počítači stačí za prvek a_{11} v matici A dosadit relativní strojovou přesnost daného počítače.

Přesnost nalezeného řešení nesouvisí jen s *numerickou stabilitou* použitého algoritmu. Závisí také na *podmíněnosti vstupních dat*, na která jsme tento algoritmus aplikovali, respektive na *podmíněnosti úlohy*. Jednotlivými pojmy se budeme postupně zabývat.

3.3.2 Podmíněnost úlohy (Condition)

Podmíněnost úlohy je dána pouze vstupními daty. Mějme matematickou úlohu, tu budeme obecně reprezentovat funkcí f . Tato funkce bude zobrazovat jistá vstupní data x na výstupní data $f(x)$. Vstupní data specifikují konkrétní úlohu, výstupní data jsou přesným řešením této konkrétní úlohy. Nyní funkcí f zobrazíme nějaká vstupní data x^* na výstupní data $f(x^*)$. Předpokládejme, že data x^* jsou aproximací dat x . Je-li pak řešení $f(x^*)$ „blízko“ řešení $f(x)$ říkáme, že úloha je dobře podmíněná a naopak, pokud se $f(x^*)$ výrazně liší od $f(x)$ úloha je špatně podmíněná. Je ovšem potřeba definovat ono „blízko“.

Definujme prostor \mathbf{X}_{IN} jako prostor všech přípustných vstupů x funkce f a v tomto prostoru definujeme metriku, budeme ji značit d_{IN} . Obdobně definujeme prostor \mathbf{Y}_{OUT} všech možných výstupů $f(x)$ funkce f a v tomto prostoru definujeme metriku, budeme ji značit d_{OUT} . Podmíněnost úlohy definujeme takto, viz [3.2; str. 4; (7)],

$$\kappa[f(x)] = \lim_{d_{IN}(x, x^*) \rightarrow 0} \left(\frac{d_{OUT}(f(x), f(x^*))}{d_{IN}(x, x^*)} \right). \quad (3.6)$$

Pokud je podmíněnost relativně malá, říkáme, že úloha je *dobře podmíněná*. Naopak, pokud je podmíněnost relativně velká, říkáme, že úloha je *špatně podmíněná*. Je-li podmíněnost nekonečná, pak říkáme, že úloha je *špatně postavená*. Vztah (3.6) v podstatě říká, že podmíněnost úlohy je dána jakousi derivací funkce f podle jejích vstupních dat. Velikost této derivace, je-li omezená, rozhoduje o tom, zda je úloha dobře nebo špatně podmíněná. Omezenost, resp. neomezenost derivace charakterizuje dobře, resp. špatně postavenou úlohu; viz [3.2], [3.3], [3.4].

3.3.3 Podmíněnost matice

Pohybujeme se zejména v oblasti lineární algebry a jednou z nejčastějších úloh je řešení soustavy lineárních rovnic. Vstupní data jsou často matice. V analýze podmíněnosti úlohy soustavy lineárních rovnic pak definujeme *číslo podmíněnosti matice A* (stručně jen *podmíněnost matice A*). Podmíněnost regulární matice A definujeme jako

$$\kappa(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = \frac{\sigma_{\max}}{\sigma_{\min}}, \quad (3.7)$$

kde použitá maticová norma je 2-norma (spektrální norma); viz [3.3; kap. 1.6.8; (1.6.39)]. Číslo σ_{max} a σ_{min} je největší, respektive nejmenší singulární číslo matice A .

Říkáme, že matice A je špatně podmíněná, je-li číslo podmíněnosti relativně velké, a naopak matice A je dobře podmíněná, je-li číslo podmíněnosti relativně malé, tedy blízké jedné. Ze vztahu (3.7) je zřejmé, že číslo podmíněnosti rovné jedné má jednotková matice I . Číslo podmíněnosti matice odráží algebraické vlastnosti matice; viz například [3.3; kap. 2.4]. V následující kapitole uvedeme příklad vlivu velké podmíněnosti matice na řešení soustavy rovnic.

3.3.4 Příklad druhý – podmíněnost matice [3.4; str. 6]

Mějme jednoduchou soustavu dvou rovnic pro dvě neznámé; soustavu s maticí A a pravou stranou b , kde

$$A = \begin{bmatrix} 2 & 6 \\ 2 & 6.00001 \end{bmatrix}, \quad b = \begin{bmatrix} 8 \\ 8.00001 \end{bmatrix}.$$

Přesné řešení této soustavy je

$$x = [1, 1]^T.$$

Dále mějme druhou soustavu s nepatrně perturbovanými prvky a_{22} a b_2

$$A' = \begin{bmatrix} 2 & 6 \\ 2 & 5.99999 \end{bmatrix}, \quad b' = \begin{bmatrix} 8 \\ 8.00002 \end{bmatrix}.$$

Přesné řešení druhé perturbované soustavy je

$$x' = [10, -2]^T.$$

Je vidět, že velmi malá relativní změna vstupních dat (řádově 10^{-6}) vyvolala velkou relativní změnu řešení (řádově 10^0 až 10^1). Z příkladu je patrné, že tato úloha řešení soustavy lineárních rovnic je velmi citlivá na perturbace vstupních dat.

Vysoká citlivost na změnu vstupních dat zde není naružena od prvního příkladu dána nevhodným algoritmem (pracovali jsme s přesnou aritmetikou). Důvod velké citlivosti této konkrétní soustavy rovnic je velká podmíněnost matice A . Podmíněnost matice A je $\kappa = 4 \times 10^6$. Řádky matice A jsou téměř lineárně závislé, proto má matice tak vysoké číslo podmíněnosti. Říkáme, že matice A je blízko matici singulární. V prvním příkladu (kapitola 3.3.1) je naopak matice soustavy podmíněná velmi dobře (číslo podmíněnosti je 2.677). K chybnému výsledku v prvním příkladu vede použití numericky nestabilního algoritmu (převedení matice na horní trojúhelníkový tvar bez předchozí pivotace způsobí velký nárůst podmíněnosti matice soustavy, číslo podmíněnosti horní trojúhelníkové matice je již 1×10^8).

3.3.5 Numerická stabilita algoritmu – přímá / zpětná stabilita (Forward / Backward stability)

Při analýze numerické stability nějakého algoritmu zjišťujeme, jakým způsobem se algoritmem šíří chyby. Můžeme postupovat v zásadě dvěma způsoby.

Přímá analýza chyb, přímá stabilita algoritmu

První způsob je tzv. *přímá analýza chyb*, ta definuje *přímou stabilitu algoritmu*. Při přímé analýze procházíme algoritmus od začátku do konce a snažíme se odhadnout, jakým způsobem se chyby šíří. Tento postup je však velmi náročný a složitý, většinou neproveditelný. Přímé analýzy chyb lze využít pouze u jednoduchých algoritmů, které se vyskytují jako stavební kameny algoritmů složitějších. Přímou analýzou velikosti chyby lze například při výpočtu skalárního součinu vektorů, nebo při násobení matice vektorem. Přímá analýza je většinou možná jen s konkrétními daty. Takto analyzovat algoritmus je tedy potřeba s různými daty zvlášť. Přímá analýza se v praxi nepoužívá.

Zpětná analýza chyb, zpětná stabilita algoritmu

Druhým způsobem je tzv. *zpětná analýza chyb*. Zpětnou analýzou jsme schopni popsat stabilitu algoritmu obecně, odděleně od konkrétních vstupních dat. Díky tomu je možné analýzu libovolné konkrétní úlohy rozdělit na dva základní kroky, na analýzu *zpětné stability algoritmu* (vzápětí ji popíšeme) a na podmíněnost vstupních dat (kapitola 3.3.2 a 3.3.3).

Mějme danou matematickou úlohu. V kapitole 3.3.2 jsme pro tuto úlohu zavedli funkci, která ji přesně řeší, nazvali jsme ji f . Budeme-li tuto úlohu řešit numericky, použijeme algoritmus (proceduru implementovanou v počítači s konečnou aritmetikou), který bude funkci f v počítači reprezentovat. Algoritmus označíme f^* . Numerické řešení konkrétní úlohy se vstupními daty x je $f^*(x)$. Pokud existují nějaká vstupní data x^* , která leží „blízko“ vstupních dat x , taková, že $f^*(x) = f(x^*)$, pak o algoritmu f^* řekneme, že je (*numericky*) *zpětně stabilní*. Jinými slovy algoritmus f^* je zpětně stabilní, pokud řešení, které nalezneme, je přesné řešení nějaké konkrétní úlohy x^* „blízké“ původní úloze x .

Jsou-li vstupní data x konkrétní úlohy, kterou chceme řešit, navíc dobře podmíněná a jsou-li data x^* aproximací dat x , znamená to, že $f(x^*)$ leží „blízko“ $f(x)$. Zřejmě tedy pak platí, že numerické řešení $f^*(x) = f(x^*)$ původního problému x nalezené algoritmem f^* leží „blízko“ skutečného řešení $f(x)$ úlohy.

Na přesnost výpočtu mají tedy vliv dva různé faktory, stabilita algoritmu a podmíněnost dat. Nelze proto očekávat, že použití zpětně stabilního algoritmu na špatně podmíněná data povede obecně na dobrou aproximaci přesného řešení. Viz příklad v kapitole 3.3.4 s velmi špatně podmíněnou maticí ($\kappa = 4 \times 10^6$). I velmi malá perturbace dat má nezanedbatelný vliv na výsledek (algoritmus zde není potřeba analyzovat, neboť máme přesná řešení). Stejně tak použití nestabilního algoritmu na dobře podmíněná data nepovede obecně na dobrou aproximaci přesného řešení. Viz příklad v kapitole 3.3.1, kde řešíme soustavu lineárních rovnic s relativně dobře podmíněnou maticí ($\kappa = 2.6177$) nestabilním algoritmem (Gaussova eliminace bez pivotace).

Analýza stability konkrétních algoritmů je velmi složitá, příklady analýzy stability a další informace týkající se stability a podmíněnosti nalezneme v literatuře [3.2], [3.3], [3.4].

3.4 Nevhodné algoritmy v teorii řízení – příklad výpočtu říditelnosti systému

Velký problém (i v teorii řízení) je ten, že pro řešení často velmi citlivých úloh se špatně podmíněnými daty se použije ten nejjednodušší algoritmus, který je k dispozici. Takový algoritmus může být numericky nestabilní a řešení úlohy se nám nepovede aproximovat s požadovanou přesností. Příkladem takového nejjednoduššího algoritmu je právě Gaussova eliminace bez pivotace pro řešení soustav lineárních rovnic (viz příklad v kapitole 3.3.1).

Zde ukážeme příklad takového nevhodného algoritmu pro výpočet říditelnosti systému. Mějme spojitý lineární dynamický systém daný dvojicí rovnic (1.1), (1.2), přesně popsany uspořádanou čtveřicí matic (A, B, C, D) . Připomeňme, že matice A má rozměry $n \times n$ a matice B má rozměry $n \times m$. Pro výpočet říditelnosti jsme uvedli v kapitole 1 několik kritérií. Bylo to kritérium (1.31a) založené na výpočtu hodnoty matice a kritérium (1.31b) založené na výpočtu vlastních čísel matice a hodnoty matice. První kritérium (1.31a) se zdá snáze algoritmizovatelné, výpočet říditelnosti tedy postavíme na něm. Dynamický systém je říditelný tehdy, když platí

$$\text{rank} \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \equiv \text{rank} [Q_C] = n. \quad (3.8)$$

Platí tedy

$$\bigcup_{j=1}^m K_n(b_j, A) = K_n(b_1, A) \cup K_n(b_2, A) \cup \dots \cup K_n(b_m, A) = R^n,$$

kde R^n je n rozměrný prostor nad tělesem reálných čísel a

$$K_i(b_j, A) = \text{span} \{b_j, Ab_j, A^2b_j, \dots, A^{i-1}b_j\}$$

je i -tý Krylovův podprostor daný vektorem b_j a maticí A (pozor, symbol \cup sjednocení je zde chápán ve smyslu sjednocení prostorů, nikoliv pouze ve smyslu sjednocení množin). Problém říditelnosti systému tedy pomocí (3.8) převedeme na problém určení hodnoty (počtu lineárně nezávislých řádků, resp. sloupců) matice Q_C rozměrů $n \times mn$.

Nejjednodušší algoritmus postavený na kritériu (3.8) nejprve spočítá matici Q_C , pak provede její singulární rozklad (SVD). Hodnost matice určíme (viz kapitolu 2.4) jako počet nenulových singulárních čísel matice. Výpočet provedeme pro konkrétní dynamický systém. Mějme tedy příklad [3.5] jednoduchého dynamického systému (stačí nám matice A, B)

$$A = \begin{bmatrix} 1 & & & & \\ & 2^{-1} & & & \\ & & 2^{-2} & & \\ & & & \ddots & \\ & & & & 2^{1-n} \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

Matice A je relativně špatně podmíněná, její podmíněnost je $\kappa = 2^{n-1}$. Pro jednotlivé prvky matice Q_C dané vztahem (3.8) platí

$$q_{ij} = 2^{(i-1)(1-j)}.$$

Pro přesnější představu nyní položíme $n = 10$ a spočítáme hodnost matice Q_C . Při výpočtu SVD na počítači s konečnou aritmetikou ovšem neurčíme skutečnou hodnost matice, ale tzv. *numerickou hodnost matice*. Numerická hodnost je vždy menší nebo rovna skutečné hodnosti matice. Rozdíl je způsoben konečnou přesností čísel uložených v počítači. Všechna singulární čísla menší než relativní strojová přesnost nejsou při výpočtu na počítači akceptována, proto může být numerická hodnost matice menší než hodnost skutečná, viz [3.5].

Nalezneme tedy singulární rozklad matice Q_C dané vztahem (3.8) pro $n = 10$. Tři nejmenší singulární čísla této matice jsou

$$\sigma_8 = 0.71247 \times 10^{-7}, \quad \sigma_9 = 0.36403 \times 10^{-9}, \quad \sigma_{10} = 0.61289 \times 10^{-12}.$$

Pokud bychom o problému říditelnosti rozhodovali pomocí algoritmu postaveného na kritériu (3.8) na méně přesném počítači, například na počítači s $\varepsilon_M > 10^{-12}$ (tedy na počítači s větší hodnotou relativní strojové přesnosti, řádově větší než 10^{-12}), závěr by byl, že tento systém není říditelný. Tento systém ovšem říditelný je. Přestože jsme řekli, že matice A je relativně špatně podmíněná, pro malá n je podmíněnost poměrně uspokojivá. Pro $n = 10$ je $\kappa = 512$. Podmíněnost matice A zde ovšem výsledek výrazně neovlivní. Aby se tento systém (A, B, \dots, \dots) stal neříditelným vlivem perturbací vstupních dat, muselo by dojít k perturbacím řádu 10^{-3} . To je vzhledem k relativní strojové přesnosti 10^{-12} velmi velká perturbace; viz [3.5].

Takto postavený algoritmus na kritériu (3.8), resp. (1.31a) nemůže nikdy fungovat správně. Rozhodnout o říditelnosti systému ovšem lze i podle kritéria (1.31b). Dynamický systém je říditelný, pokud platí

$$\text{rank}[\lambda_i I - A, \quad B] \equiv \text{rank}[P_i] = n, \quad \text{pro } i = 1, \dots, n, \quad (3.9)$$

kde λ_i jsou vlastní čísla matice A . V algoritmu tedy musíme nejprve řešit problém vlastních čísel matice A , následně problém hodnosti matic P_i .

U prvního algoritmu postaveného na kritériu (3.8) počítáme $n-1$ součinů matice s vektorem a jeden singulární rozklad. U druhého algoritmu postaveného na kritériu (3.9) počítáme vlastní čísla matice a n singulárních rozkladů. Druhý algoritmus je výpočetně složitější, mohlo by se tedy zdát, že je méně výhodný pro řešení úlohy říditelnosti systému. Opak je ovšem pravdou, lze ukázat [3.5], že pro nejmenší singulární číslo σ_n matice P_i platí

$$\sigma_n > 2^{-(n+1)}, \quad \text{pro } i = 1, \dots, n.$$

Navážeme-li na příklad, kde $n = 10$, pak zřejmě platí že

$$\sigma_{10} > 0.48828 \times 10^{-3}.$$

O říditelnosti systému, při použití algoritmu postaveného na kritériu (3.9), rozhodneme správně už na počítače s hodnotou relativní strojové přesnosti řádově 10^{-5} ; viz [3.5].

Algoritmus postavený na kritériu	(3.8)	(3.9)
Minimální hodnota ε_M pro dosažení správného výsledku (řádově)	10^{-12}	10^{-5}

Tabulka 3.1; Porovnání minimálních řádových hodnot relativních strojových přesností pro dosažení správného výsledku při použití různých algoritmů.

Z tabulky 3.1 je vidět, že při použití prvního algoritmu (3.8) potřebujeme provádět výpočty s daleko větší přesností. Současné počítače mají při použití dvojitě přesnosti $\varepsilon_M \approx 10^{-16}$, použití obou postupů by tak vedlo ke správnému výsledku (pro $n = 10$).

Tento příklad ukazuje důležitost výběru vhodného algoritmu a důležitost studia chování algoritmů. Při použití nevhodného algoritmu můžeme výsledek získat snadno, ale tento výsledek vůbec nemusí souviset se skutečným řešením úlohy. Přestože obě kritéria (3.8), (3.9) říditelnosti systému jsou správná, použití kritéria (3.8) jako vzorce pro výpočet říditelnosti je vysloveně nevhodné. Slabým místem algoritmu postaveného na kritériu (3.8) je počítání mocnin matice A , každým mocněním matice se zvyšuje číslo podmíněnosti. Pro matici A z tohoto příkladu dokonce platí $\kappa(A^k) = \kappa^k(A)$. Matice A^k se s rostoucím k velmi rychle blíží k matici singulární.

Toto slabé místo ovšem není skryto přímo v kritériu (3.8), ale v tom, jakým způsobem z něj byl vytvořen algoritmus. Vhodnější algoritmus postavený na (3.8) by jednotlivé sloupce matice Q_C generoval a postupně vzájemně ortogonalizoval a normalizoval. Výsledkem algoritmu by byla matice Q_C' rozměrů $n \times q$, kde $q \leq n$ je hodnost matice Q_C' , resp. Q_C . Sloupce matice Q_C' tak tvoří ortonormální bázi q -rozměrného podprostoru prostoru R^n .

V úvodu tohoto příkladu jsme vyjádřili kritérium (3.8) pomocí sjednocení tzv. Krylovových podprostorů. Tyto podprostory se používají mj. (a také jsme na ně narazili) při řešení soustav lineárních rovnic u projekčních metod (kapitola 2.1.4), přesněji řečeno u metod Krylovových podprostorů (krylovovských metod), například metoda konjugovaných gradientů CG. U těchto metod se báze podprostoru generuje právě tak, že se jednotlivé vektory postupně vzájemně ortogonalizují a normalizují.

4. Příklady velkých úloh v teorii dynamických systémů a teorii řízení

V předchozích třech kapitolách jsme se věnovali zejména teoretickým otázkám, formulacím problémů v teorii dynamických systémů a v teorii řízení a jejich matematickým popisům. Tím jsme zredukovali úlohy, které nás zajímají, na několik málo úloh lineární algebry. Lineární algebraické úlohy jsme si roztřídili na základní typy úloh a naznačili jsme, jak lze tyto úlohy řešit. Upozornili jsme na problémy při numerickém řešení lineárních algebraických úloh. Seznámili jsme se se základními pojmy z analýzy stability algoritmů.

V celém textu jsme neustále upozorňovali na to, že se chceme zabývat řešením úloh značně rozsáhlých. Rozsáhlost úloh, nezanedbatelným způsobem ovlivňuje metody, které můžeme k jejich řešení používat. Rozsáhlost úloh vyplývá z potřeby zpřesňování výpočtů, je způsobena stále jemnější diskretizací zkoumaných oblastí, snahou o co největší komplexnost zkoumaných úloh. Nyní se seznámíme s některými problémy z teorie dynamických systémů a teorie řízení, které vedou na velmi rozsáhlé úlohy.

4.1 Stručný úvod do fyziky piezoelektrických materiálů

Piezoelektrika jsou materiály vykazující tzv. *piezoelektrický jev*. Jsou to materiály, v nichž existuje *elektromechanická vazba*. Konkrétně se jedná o to, že u takového materiálu můžeme mechanickým namáháním vyvolat odezvu v podobě vzniku elektrického pole v materiálu. Naopak, pokud materiál vložíme do elektrického pole, můžeme v materiálu vyvolat mechanickou odezvu. Tento fyzikální jev lze dobře popsat pomocí termodynamiky. Existence jevu závisí zejména na vnitřní struktuře a symetrii materiálu. Symetrie materiálů vychází ze studia symetrie krystalů a popisujeme ji pomocí grup symetrie. Fyzikálně jev pak popisujeme pomocí tenzorových veličin, které charakterizují materiálové vlastnosti příslušného materiálu. Podrobný rozbor symetrie krystalů, vlivu symetrie na vlastnosti materiálů a popis různých tenzorových veličin nalezneme v literatuře, například v [4.7]. Popis piezoelektrik a jejich použití pak nalezneme například v [4.6].

Zde uvedeme pouze běžně užívaný popis piezoelektrik pomocí stavových rovnic odvozených z termodynamiky. Budeme-li uvažovat pouze izotermické děje (nebo pouze adiabatické děje), piezoelektrický materiál můžeme popsat pomocí dvojice rovnic

$$T_{ij} = c_{ijkl}^{(E)} S_{kl} + e_{kij} E_k, \quad \text{kde } i, j, k, l = 1 \dots 3 \quad (4.1)$$

$$D_i = e_{ikl} S_{kl} - \varepsilon_{ik}^{(S)} E_k. \quad (4.2)$$

První rovnice (4.1) je rovnicí mechanickou a druhá rovnice (4.2) je rovnice elektrická. Jednotlivé fyzikální veličiny, vystupující v rovnicích, jsou

T_{ij}	– tenzor napětí,	S_{ij}	– tenzor relativní deformace,
D_i	– vektor elektrické indukce a	E_i	– vektor intenzity elektrického pole.

Materiálové konstanty vystupující v rovnicích jsou pak

$c_{ijkl}^{(E)}$ – tenzor elastických modulů (při konstantní intenzitě elektrického pole),
 e_{ijk} – tenzor piezoelektrických modulů
 $\varepsilon_{ij}^{(S)}$ – tenzor permitivity (při konstantní deformaci).

4.2 Příklad první – piezoelektrický rezonátor jako dynamický systém

Nyní se budeme zabývat vlastnostmi tzv. piezoelektrického rezonátoru. Jedná se o velmi jemný výbrus z piezomateriálu. Tento výbrus je buzen například harmonickým elektrickým polem a vlivem elektromechanické vazby dojde k jeho rozkmitání. Tyto piezoelektrické rezonátory se v praxi velmi často používají, neboť mají mimořádně stabilní chování v čase (jejich rezonanční frekvence se s časem téměř nemění). Je ovšem nutné pro konkrétní rezonátor znát vlastní frekvence a tvary kmitů. Tyto frekvence rezonátoru můžeme měřit v laboratoři, některé jednoduché úlohy lze samozřejmě řešit analyticky. Tento problém lze také modelovat na počítači. *Rezonanční frekvence* je vlastní frekvence rezonátoru, na které je rezonátor buzen, pojem *vlastní frekvence* definujeme později v této kapitole.

Všechny předchozí rovnice, které jsme si uváděli, popisují statické rozložení mechanické (elastické) a elektrické energie v materiálu – jsou to stavové rovnice. Říkají nám, kolik energie „konzervované“ v materiálu je uloženo v podobě mechanické potenciální energie, kolik je uloženo v podobě elektrické energie a kolik energie jsme schopni „přesouvat“ pomocí piezoelektrických koeficientů (poměry těchto energií jsou popsány *koeficientem elektromechanické vazby*, viz např. [4.6; str. 29]). K popisu rezonátoru potřebujeme rovnice, které budou popisovat jeho dynamiku. Potřebujeme tedy získat pohybovou rovnici kontinua – piezoelektrického materiálu. Pohybovou rovnici vyjádříme dosazením stavových rovnic piezoelektrika (4.3), (4.4) do rovnice energetické bilance, tzv. *Hamiltonova energetického principu* [4.9; kap. 6.4]. Ten říká, že rozdíl kinetické a potenciální energie je v každém okamžiku roven práci vnějších sil. Nahradíme-li navíc elektrickou indukci hustotou plošného náboje a dosadíme-li za elektrickou intenzitu gradient potenciálu (viz [4.10; kap. 1.5]), získáme po diskretizaci úlohy například pomocí FEM dvojici rovnic

$$M\ddot{U} + K_M U + K_P^T \Phi = F, \quad (4.6)$$

$$K_P U + K_E \Phi = \sigma. \quad (4.7)$$

Celé odvození rovnic (4.6), (4.7) z rovnic (4.3), (4.4) včetně diskretizace například viz [4.1], [4.2], [4.3]. Tato dvojice rovnic vychází ze stavového popisu, jednotlivé použité fyzikální veličiny (matice K_i) tak nepřímo souvisí s fyzikálními veličinami v materiálové matici Θ . Stejně tak jako ve stavových rovnicích se zde vyskytují mechanické a elektrické veličiny. Jsou to vektory

U – deformace materiálu (posunutí), F – vnější síla (např. mechanické buzení),
 Φ – elektrický potenciál a σ – plošná hustota náboje.

Dále jsou to matice

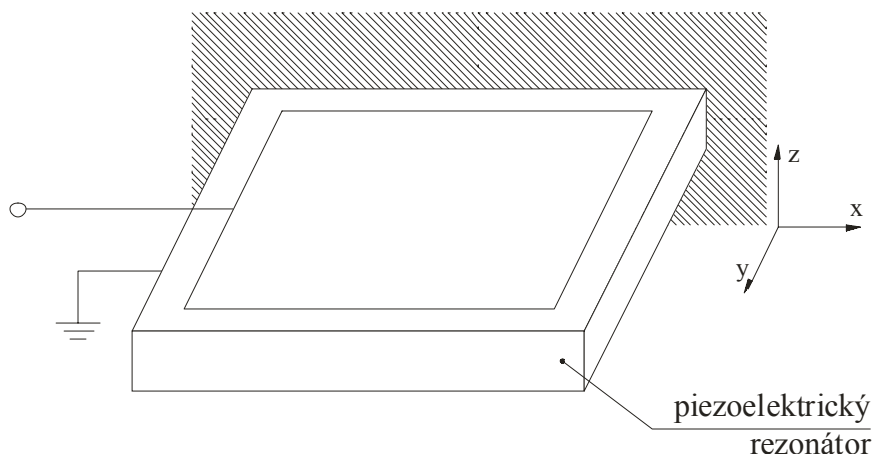
K_M – matice reprezentující mechanickou tuhost,
 K_P – matice reprezentující elektromechanickou vazbu a
 K_E – matice reprezentující elektrickou kapacitu.

Tyto matice obsahují příslušné materiálové koeficienty podobně jako materiálová matice Θ a obsahují informace o způsobu diskretizace rezonátoru. Poslední koeficient je matice

M – hmotnostní matice.

Tato matice popisuje setrvačnost rezonátoru, také odráží strukturu úlohy po diskretizaci.

Do modelu jsou také zahrnuty okrajové podmínky pro U , v místě uchycení rezonátoru je nulové posunutí, a pro Φ , zadaný potenciál na elektrodách rezonátoru. Přesný a kompletní popis aplikace FEM pro modelování piezoelektrických rezonátorů (variační odvození rovnic, zahrnutí okrajových podmínek, atd.) nalezneme v literatuře, viz například články [4.1]...[4.4] a nebo například disertační práce V. Pieforta [4.5].



Obrázek 4.2; Piezoelektrický rezonátor je výbrus z piezoelektrického materiálu. Pro nalezení jeho vlastních frekvencí lze použít například FEM model. V místě uchycení rezonátoru (zadní stěna) je zadána Dirichletova podmínka – nulové posunutí.

Předpokládejme, že posunutí U , které je v reálném problému funkcí prostoru a času, bude v rovnicích (4.6), (4.7) harmonickou stojatou vlnou,

$$U = u(x, y, z) \cdot \exp(i\omega t).$$

Pro druhou derivaci posunutí platí

$$\ddot{U} = -\omega^2 U, \quad (4.8)$$

kde ω je právě úhlová frekvence kmitů. Po dosazení vztahu (4.8) do rovnice (4.6) dostáváme

$$(K_M - \omega^2 M)U + K_P^T \Phi = F, \quad (4.9)$$

$$K_P U + K_E \Phi = \sigma. \quad (4.10)$$

Vlastní frekvence ω_R je rezonátor schopen v ideálním případě generovat bez buzení, jsou tedy řešením soustavy rovnic (4.9), (4.10) s neznámou ω s nulovou pravou stranou ($F = 0, \sigma = 0$). Vynásobíme-li rovnici (4.9) zleva maticí $-M^{-1}$ a nahradíme-li

$$\omega^2 = \lambda, \quad (4.11)$$

vyjádříme dvojici rovnic (4.9), (4.10) v maticovém zápisu

$$\begin{bmatrix} \lambda I - M^{-1} K_M & -M^{-1} K_P^T \\ K_P & K_E \end{bmatrix} \begin{bmatrix} U \\ \Phi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (4.12)$$

Hledáme taková $\lambda_R = \omega_R^2$ a vektory $[U_R, \Phi_R]^T$, která rovnici (4.12) řeší, ω_R jsou tedy *vlastní frekvence* rezonátoru, vektor U_R odpovídá tvaru mechanických kmitů.

V kapitole 1.2 jsme pro popis dynamických systémů zavedli matici systému $P(s)$. Blokovaná matice rovnice (4.12) je maticí systému FEM modelu piezoelektrického rezonátoru. Systémová matice (1.8) je

$$P(\lambda) = \begin{bmatrix} \lambda I - A & -B \\ C & D \end{bmatrix} = \begin{bmatrix} \lambda I - M^{-1}K_M & -M^{-1}K_P^T \\ K_P & K_E \end{bmatrix}. \quad (4.13a)$$

Ze systémové matice je možné vyjádřit přenosovou matici $P(s)$ systému (viz kapitolu 1.2). Přenosová matice (1.11) je

$$G(\lambda) = B(\lambda I - A)^{-1}C + D = K_P(\lambda I - M^{-1}K_M)^{-1}M^{-1}K_P^T + K_E. \quad (4.13b)$$

FEM model piezoelektrického rezonátoru je tedy v rovnici (4.12), resp. (4.13a), (4.13b) popsán jako lineární dynamický systém. Dynamický systém včetně vstupů a výstupů je pomocí matice $P(s)$ popsán viz (1.7),

$$P(\lambda) \begin{bmatrix} X(\lambda) \\ U(\lambda) \end{bmatrix} = \begin{bmatrix} \lambda I - A & -B \\ C & D \end{bmatrix} \begin{bmatrix} X(\lambda) \\ U(\lambda) \end{bmatrix} = \begin{bmatrix} x(0) \\ Y(\lambda) \end{bmatrix},$$

kde $X(s)$ je stavový vektor systému, $U(s)$ je vstup systému, $Y(s)$ je výstup systému a $x(0)$ je počáteční podmínka stavového vektoru. Tedy vstupem dynamického systému (4.12) je vektor Φ , stavovým vektorem je vektor deformace U , výstupem je vektor hustoty plošného náboje σ .

Přenosové póly lineárního dynamického systému jsou taková λ_p , která splňují rovnici

$$(\lambda I - A) = 0.$$

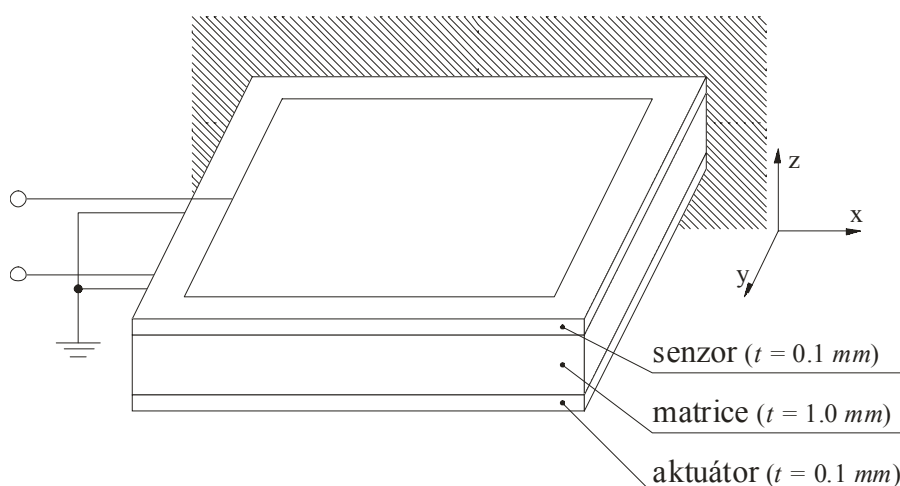
Přenosové póly jsou vlastní čísla matice A . Invariantní nuly (viz kapitolu 1.4) jsou nuly charakterizované nulovým chováním systému na výstupu (nulovým $Y(\lambda)$), jsou tedy řešením rovnice (4.12). Invariantní nuly jsou kořeny polynomu, který je největším společným dělitelem všech minorů nejvyššího řádu matice $P(\lambda)$ (1.8), resp. (4.13a).

Póly dynamického systému (4.12) jsou kvadráty vlastních frekvencí mechanického subsystému tohoto systému. Vlastní frekvence piezoelektrického rezonátoru ω_R jsou dány množinou nul dynamického systému (4.12).

Hledání vlastních frekvencí piezoelektrického rezonátoru s využitím diskretizace pomocí FEM může vést na velmi rozsáhlý dynamický systém (4.12). Zde tedy poprvé narážíme na příklad rozsáhlé úlohy v teorii dynamických systémů. Je zřejmé, že problém nalezení vlastních frekvencí rezonátoru, tedy nul dynamického systému, není možné řešit hledáním kořenů polynomu a že ani není možné tento polynom určit jako největší společný dělitel minorů matice $P(s)$. V podstatě všechny operace nutné k výpočtu nul klasickým způsobem vedou na problém faktorizace polynomu, tedy na problém vlastních čísel.

4.3 Příklad druhý – aktivní piezoelektrické prvky

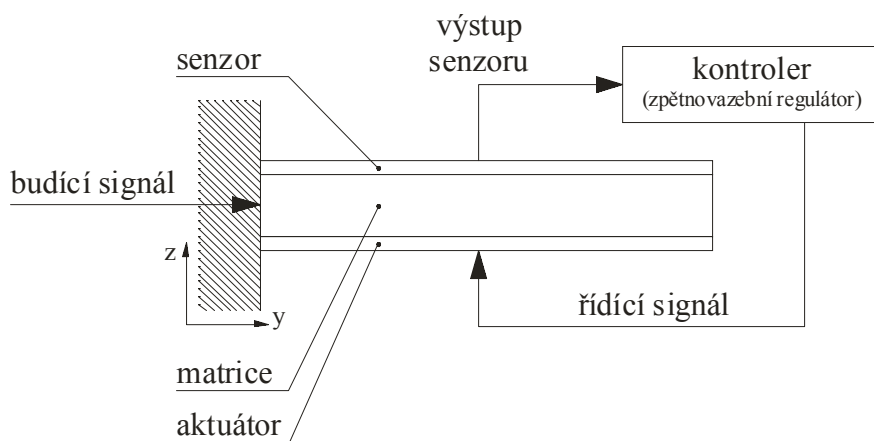
V druhém příkladu rozsáhlé úlohy teorie řízení se zaměříme na studium speciálního aktivního piezoelektrického prvku. V současné době se již asi každý setkal s pojmem *inteligentní materiál* (*Smart Material*). Aktivní piezostukturu, kterou se nyní budeme zabývat, bychom mohli mezi tyto materiály zařadit. Tento materiál je tvořen elastickou maticí, která piezoelektrické vlastnosti nevykazuje. Na matici jsou pak ze dvou protilehlých stran vrstvy piezomateriálu. V praxi musí být ještě každá vrstva piezomateriálu opatřena dvojicí elektrod. Elektrody jsou potřeba proto, abychom mohli na vrstvu piezomateriálu přivést nějaké napětí, nebo abychom mohli měřit náboj, který zde vzniká deformací materiálu vlivem zpětného piezoelektrického jevu. Příklad takové struktury je na obrázku 4.3.



Obrázek 4.3; Piezoelektrický aktivní prvek tvořený třemi vrstvami. Prostřední (nejsilnější) vrstva je matrice tvořená běžným elastickým materiálem. Tato vrstva je obklopená vrstvami z piezoelektrických materiálů – senzorem a aktuátorem. Tloušťky jednotlivých vrstev jsou čistě ilustrační.

Každou piezoelektrickou vrstvu takového materiálu budeme využívat jiným způsobem. Jednu vrstvu (na obrázku horní vrstvu) budeme využívat jako senzor, druhou vrstvu (spodní) budeme využívat jako aktuátor. Pokusme se nyní vysvětlit, jak můžeme takový vrstevnatý materiál využít a co nás vede k tomu, že ho nazýváme „inteligentním“ materiálem. Prostřední elastickou vrstvu můžeme rozkmitat pomocí nějakého budícího signálu. Elastická vrstva (respektive celá struktura) může být realizována například jako křídlo letadla nebo fotovoltaický článek vesmírné družice. Obzvláště v kosmonautice a aeronautice se takové problémy řeší velmi často. Plochá konstrukce křídla se rozkmitá poměrně snadno a je zřejmé, že to není žádoucí. Otázkou tedy je, jak tyto kmity, tedy vlastní kmity elastické konstrukce, eliminovat. Opatříme-li tenkou elastickou vrstvu (matici) velmi slabou vrstvou piezoelektrického materiálu s elektrodami, můžeme teoreticky relativně dobře měřit deformaci matrice. Deformace matrice, způsobená jejím kmitáním, pochopitelně způsobí deformaci piezovrstvy a vlivem zpětného piezoelektrického jevu dojde k hromadění náboje na elektrodách této vrstvy. Změřením napětí na elektrodách této vrstvy v různých bodech tak získáme představu o deformaci matrice. Takovou vrstvu můžeme tedy považovat za jakýsi distribuovaný senzor. Druhá piezovrstva je aktuátor. I na té dochází vlivem deformací matrice k hromadění náboje, zde ovšem nebudeme napětí na elektrodách měřit. Právě naopak. Na elektrody aktuátoru přivedeme zesílené napětí ze senzoru, samozřejmě s opačným znaménkem (resp. na aktuátor přivedeme napětí ze senzoru obecně

transformované nějakým řídicím algoritmem \mathbf{R}). Napětí zesílíme například stokrát, abychom mohli napětí vzniklé na aktuátoru vlivem zpětného piezoelektrického jevu zanedbat. Přivedením napětí s opačným znaménkem dojde k deformaci aktuátoru opačným směrem vzhledem k matici. Budeme-li opět aktuátor považovat za aktuátor s distribuovanými parametry, můžeme se tak pokusit aktivně tlumit vibrace a kmity matrice. Princip funkce materiálu je schematicky znázorněn na obrázku 4.4.



Obrázek 4.4; Schéma funkce „inteligentního“ piezoelektrického prvku pro aktivní regulaci a řízení vibrací.

Při modelování takového materiálu na počítači například pomocí FEM do modelu zahrneme i řídicí algoritmus \mathbf{R} (na obrázku realizován kontrolerem). Hledání řídicího algoritmu je pak pochopitelně velmi složitou a velmi rozsáhlou úlohou teorie řízení (v prvním příkladu jsme řešili pouze úlohu teorie dynamických systémů).

Zde se nebudeme zabývat úlohou hledání optimálního řízení takové aktivní struktury. Pouze se pokusíme s využitím prvního příkladu popsat dynamiku této struktury pro určitý konkrétní řídicí algoritmus. Tato aktivní struktura obsahuje tři vrstvy, z toho jsou dvě z piezoelektrického materiálu. Piezovrstvy popíšeme rovnicemi analogickými s rovnicemi (4.6) a (4.7) z prvního příkladu [4.1; (10)], [4.3; (26)].

$$M_S \ddot{U}_S + S_S \dot{U}_S + K_{S,M} U_S + K_{S,P}^T \Phi_S = F_S, \quad (4.14)$$

$$K_{S,P} U_S + K_{S,E} \Phi_S = \sigma_S. \quad (4.15)$$

$$M_A \ddot{U}_A + S_A \dot{U}_A + K_{A,M} U_A + K_{A,P}^T \Phi_A = F_A, \quad (4.16)$$

$$K_{A,P} U_A + K_{A,E} \Phi_A = \sigma_A. \quad (4.17)$$

První dvojice rovnic (4.14) a (4.15) popisuje senzor, druhá dvojice (4.16) a (4.17) pak aktuátor (viz dolní indexy S/A u veličin a materiálových koeficientů). Význam jednotlivých veličin i koeficientů byl již popsán v prvním příkladu s výjimkou koeficientu

S – matice tlumení.

Tato matice popisuje tlumení, tedy jistou disipaci energie v materiálu (viz [4.1], [4.3]). Pro úplné popsání modelu této struktury bychom ještě potřebovali pohybovou rovnici elastické matrice. Ta by v podstatě vypadala stejně, jako pohybové rovnice senzoru/aktuátoru – rovnice (4.14) a (4.16) – ovšem s nulovým koeficientem K_P . Pohybovou rovnici matrice bychom potřebovali například pro výpočet vlastních kmitů celé struktury. My ovšem pouze popíšeme chování aktuátoru (resp. odpovídajícího dynamického systému) v závislosti na chování senzoru (resp. odpovídajícího dynamického systému) a v závislosti na algoritmu řízení \mathbf{R} . V dalším popisu budeme vždy uvažovat, že pro senzor i aktuátor byl použit stejný materiál se stejnými materiálovými koeficienty a se stejnými parametry (rozměry). Rovněž budeme uvažovat stejnou diskretizaci obou piezoelektrických vrstev. Díky těmto předpokladům budou matice koeficientů senzoru a aktuátoru stejné. Potom tedy platí a budeme psát

$$\begin{aligned} K_{S,M} &= K_{A,M} = K_M, & K_{S,P} &= K_{A,P} = K_P, & K_{S,E} &= K_{A,E} = K_E, \\ S_S &= S_A = S, \\ M_S &= M_A = M. \end{aligned}$$

Mechanické a elektrické veličiny (tedy U , F , Φ a σ) jsou pochopitelně u senzoru a u aktuátoru různé. U vrstvy realizující senzor nedochází k buzení elektrickým polem, platí tedy $\sigma_S = 0$. Z rovnice elektrické bilance senzoru (4.15) pak můžeme vyjádřit napětí na elektrodách senzoru

$$\Phi_S = -K_E^{-1} K_P U_S. \quad (4.18)$$

Po dosazení do pohybové rovnice senzoru dostáváme rovnici, která popisuje dynamiku piezoelektrické vrstvy realizující senzor

$$M\ddot{U}_S + S\dot{U}_S + (K_M - K_P^T K_E^{-1} K_P)U_S = F_S. \quad (4.19)$$

U piezoelektrické vrstvy realizující aktuátor již máme obecně nenulové buzení elektrickým polem. Toto buzení je dokonce prostřednictvím daného řídicího algoritmu \mathbf{R} závislé na napětí na senzoru. Jak jsme již zmínili, optimální algoritmus řízení nebudeme hledat, neboť je to pravděpodobně velmi složitá úloha závisající na mnoha parametrech. Kritérium optimality řídicího algoritmu by byla zřejmě minimální výchylka celé struktury z nulové polohy. Algoritmů řízení této struktury je v literatuře k dispozici několik [4.1]. Tyto algoritmy jsou ale většinou relativně jednoduché a intuitivní. Chceme-li nyní popsat dynamiku piezoelektrické vrstvy realizující aktuátor, musíme některý z těchto algoritmů použít. Nejprve tedy vyjádříme buzení aktuátoru σ_A , tedy pravou stranu rovnice (4.17), uvedeme si dva algoritmy řízení. Oba algoritmy jsou založeny na tom, že aktuátor působí proti směru pohybu matrice (resp. celé struktury). Aktuátorem se tak snažíme potlačit jakékoliv vibrace, ke kterým v matici dochází.

Algoritmus řízení s konstantním zesílením (Constant Gain Feedback Control)

Tento algoritmus (resp. regulátor) reaguje na velikost změny (časové derivace) napětí na senzoru, zesílenou a s opačným znaménkem ji přivede na aktuátor. Zesílení je konstantní a záporné (z důvodu změny znaménka). Algoritmus popíšeme vztahem (viz [4.1])

$$\sigma_A = \Gamma \cdot \frac{\partial}{\partial t} \Phi_S = \Gamma \cdot \dot{\Phi}_S = -\Gamma \cdot K_E^{-1} K_P \dot{U}_S, \quad (4.20)$$

kde Γ je zesílení (resp. matice zesílení) regulátoru. Z rozměrové analýzy této rovnice je zřejmé, že matice zesílení Γ je v jednotkách $[(C \cdot m^{-2}) / (V \cdot s^{-1})] = [F \cdot s \cdot m^{-2}]$.

Ljapunovovo řízení (Lyapunov Control – Constant Amplitude Feedback Control)

Druhý algoritmus reaguje pouze na směr změny napětí na senzoru (reaguje pouze na to, zda se napětí zvětšuje nebo naopak). Na aktuátor je přiveden konstantní budicí signál, pouze s různým znaménkem, respektive nulový budicí signál v případě, že se napětí na senzoru nemění. Algoritmus popíšeme vztahem (viz [4.1])

$$\sigma_A = \Gamma \cdot \operatorname{sgn} \left(\frac{\partial}{\partial t} \Phi_S \right) = \Gamma \cdot \operatorname{sgn} (\dot{\Phi}_S) = -\Gamma \cdot K_E^{-1} K_P \operatorname{sgn} (\dot{U}_S). \quad (4.21)$$

Matice Γ je opět zesílení regulátoru a funkce $\operatorname{sgn}(x)$ je tzv. *signum funkce*. Tedy funkce, která vrátí znaménko argumentu. Funkce tedy vrátí hodnotu -1 pro $x < 0$, hodnotu 0 pro $x = 0$ a hodnotu $+1$ pro $x > 0$.

V příkladu použijeme řídicí algoritmus popsany vztahem (4.20). S pomocí tohoto vztahu přepíšeme pohybovou a elektrickou rovnici aktuátoru; tedy rovnice (4.16), (4.17)

$$M\ddot{U}_A + S\dot{U}_A + K_M U_A + K_P^T \Phi_A = F_A, \quad (4.22)$$

$$K_P U_A + K_E \Phi_A = -\Gamma \cdot K_E^{-1} K_P \dot{U}_S. \quad (4.23)$$

Nyní z rovnice (4.23) vyjádříme napětí na elektrodách aktuátoru stejně, jako jsme to dělali u senzoru. Platí pak

$$\Phi_A = -K_E^{-1} K_P U_A - K_E^{-1} \cdot \Gamma \cdot K_E^{-1} K_P \dot{U}_S, \quad \text{respektive} \quad (4.24)$$

$$\Phi_A = \Phi'_A - K_E^{-1} \cdot \Gamma \cdot K_E^{-1} K_P \dot{U}_S,$$

kde Φ'_A je napětí, které na aktuátoru vznikne vlivem zpětného piezoelektrického jevu (tedy napětí, které na aktuátoru vznikne vlivem jeho vlastní deformace). Napětí na aktuátoru Φ_A je tedy dáno jako součet napětí z regulátoru a právě napětí Φ'_A . Napětí Φ'_A tak můžeme v podstatě považovat za jakousi *regulační odchylku*.

Po dosazení rovnice (4.24) do pohybové rovnice aktuátoru (4.22) dostaneme rovnici

$$M\ddot{U}_A + S\dot{U}_A + KU_A = F_A + \Gamma' \cdot \dot{U}_S, \quad (4.25)$$

kde K je matice tvořená materiálovými koeficienty a Γ' je matice tvořená řídicím algoritmem a materiálovými koeficienty

$$K \equiv K_M - K_P^T K_E^{-1} K_P$$

$$\Gamma' \equiv K_P^T K_E^{-1} \cdot \Gamma \cdot K_E^{-1} K_P \dot{U}_S.$$

Na závěr aktuátor vyjádříme jako dynamický systém popsany klasickými rovnicemi (1.1), (1.2). Aktuátor tedy popíšeme rovnicemi ve tvaru

$$\dot{x} = Ax + Bz$$

$$y = Cx + Dz,$$

kde x je stavová veličina, z je vstup systému a y je výstup. Definujme

$$x \equiv \begin{bmatrix} U_A \\ \dot{U}_A \end{bmatrix}, \quad z \equiv [F_A + \Gamma' \cdot U_S] \quad \text{a} \quad y \equiv [\Phi'_A].$$

Stavovým vektorem je výchylka a rychlost, vstupem mechanické buzení a elektrické buzení realizované regulátorem, výstupem pak regulační odchylka napětí. Dynamický systém FEM modelu aktuátoru popíšeme dvojicí rovnic

$$\dot{x} = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}S \end{bmatrix} x + \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix} z \quad (4.26)$$

$$y = [-K_E^{-1}K_P, \quad 0]x, \quad (4.27)$$

kde I je jednotková matice; v maticovém zápisu

$$\begin{bmatrix} \dot{U}_A \\ \ddot{U}_A \\ \Phi'_A \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ -M^{-1}K & -M^{-1}S & M^{-1} \\ -K_E^{-1}K_P & 0 & 0 \end{bmatrix} \begin{bmatrix} U_A \\ \dot{U}_A \\ F_A + \Gamma' \cdot U_S \end{bmatrix}. \quad (4.28)$$

Pomocí rovnic (4.26), (4.27), respektive pomocí rovnice (4.28) jsme popsali aktuátor aktivní piezostuktury (viz obrázek 4.4) jako dynamický systém.

4.4 Příklad třetí – iterační metody jako dynamické systémy se zpětnovazebním řízením

V předchozích dvou příkladech jsme se zabývali popisem konkrétních fyzikálních jevů z pohledu teorie dynamických systémů. Ve třetím příkladu popíšeme z pohledu teorie řízení a teorie dynamických systémů obecnou lineární stacionární iterační metodu pro řešení soustavy rovnic (2.1)

$$Ax = b. \quad (4.29)$$

Soustava rovnic (4.29), kterou chceme řešit, realizuje určitou *regulovanou soustavu* a lineární iterační metoda, kterou k řešení použijeme, pak zastává funkci *regulátoru* (přesný tvar regulované soustavy i regulátoru odvodíme později v této kapitole). Hledání v jistém smyslu optimální metody řešení soustavy (4.29) je z pohledu teorie řízení ekvivalentní úloze hledání optimální regulace.

Je-li matice A soustavy rovnic (4.29) regulární, lze tuto soustavu řešit jednoznačně, existuje právě jedno řešení; značíme

$$x = A^{-1}b. \quad (4.30)$$

Při řešení soustavy (4.29) pomocí stacionárních iteračních metod (viz kapitolu 2.1.3) vycházíme z dané počáteční aproximace řešení (počáteční podmínky) $x^{(0)}$ a neustále se v jednotlivých krocích postupnými úpravami aproximace řešení přibližujeme k skutečnému řešení (4.30). Pro jednotlivé úpravy platí

$$x^{(k+1)} = x^{(k)} + Hr^{(k)}, \quad (4.31)$$

kde H je matice specifická pro každou iterační metodu a r je reziduum. Reziduum lze vyjádřit pomocí rozdílu mezi skutečným řešením a jeho aproximací; platí

$$r^{(k)} = b - Ax^{(k)} = A(x - x^{(k)}), \quad (4.32)$$

po dosazení do (4.31) dostaneme

$$x^{(k+1)} = x^{(k)} + H(b - Ax^{(k)}) = (I - HA)x^{(k)} + Hb = Bx^{(k)} + Hb. \quad (4.33)$$

Nyní definujeme dva další pomocné vektory, které budeme později potřebovat

$$u^{(k)} \equiv Hr^{(k)}, \quad (4.34)$$

$$y^{(k)} \equiv Ax^{(k)}. \quad (4.35)$$

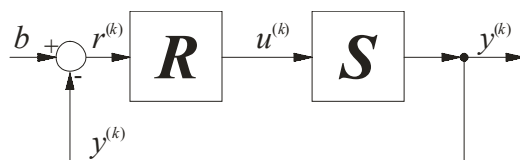
Za předpokladu, že uvažovaná metoda konverguje, platí $x^{(k)} \rightarrow x$, $r^{(k)} \rightarrow 0$, pak pro pomocné vektory platí $u^{(k)} \rightarrow 0$, $y^{(k)} \rightarrow b$.

Chceme-li iterační metody popisovat jako dynamické systémy, musíme použít diskretní popis, neboť iterační metody nejsou spojité v čase (hodnoty se mění skokově v jednotlivých iteracích). Diskretní dynamický systém lze obecně popsat dvojicí rovnic (1.1a) – stavovou rovnicí a (1.2a) – rovnicí výstupu;

$$\begin{aligned}x_{k+1} &= \tilde{A}x_k + \tilde{B}u_k, \\y_k &= Cx_k + Du_k.\end{aligned}$$

Aby nedocházelo k záměnám jednotlivých vektorů, bylo zachováno původní značení z kapitoly 1.1 (vektor $x^{(k)}$ je aproximace řešení soustavy (4.29) a x_k stavový vektor dynamického systému, atd.). Diskretní dynamický systém je přesně popsán uspořádanou čtveřicí matic ($\tilde{A}, \tilde{B}, C, D$).

Nyní přepíšeme rovnice obecné lineární stacionární iterační metody tak, aby byly kompatibilní s popisem diskretního lineárního dynamického systému. Zachováme přitom již naznačenou strukturu – oddělíme vlastní metodu od soustavy rovnic. Schéma řešení soustavy rovnic (4.29) pomocí obecné iterační metody je na obrázku 4.5.



Obrázek 4.5; Schéma řešení soustavy (4.29) pomocí iterační metody. Metodu charakterizuje regulátor R , úloha je reprezentována regulovanou soustavou S .

Jako jednotlivé *signály* (viz obrázek) používáme vektory z rovnic popisujících obecnou lineární iterační metodu

$y^{(k)}$	– regulovaná veličina	(pomocný vektor (4.35)),
b	– požadovaná hodnota regulované veličiny	(pravá strana soustavy (4.29)),
$r^{(k)}$	– regulační odchylka	(k -té reziduum),
$u^{(k)}$	– regulační veličina	(pomocný vektor (4.34)).

Aproximaci řešení – vektor $x^{(k)}$ nepoužijeme jako signál ale jako stavový vektor dynamického systému S (regulované soustavy); tedy

$x^{(k)}$	– stavový vektor regulované soustavy	(k -tý odhad neznámé),
$x^{(0)}$	– počáteční podmínka	(počáteční odhad neznámé).

Regulátor R i regulovaná soustava S na obrázku 4.5 jsou obecné diskretní dynamické systémy. Z obrázku je patrné, co jsou vstupy a co jsou výstupy u obou systémů. Regulovaná soustava S představuje úlohu, kterou popíšeme dvojicí rovnic (stavovou rovnicí a rovnicí výstupu)

$$x^{(k+1)} = Ix^{(k)} + Iu^{(k)}, \quad (4.36)$$

$$y^{(k)} = Ax^{(k)}. \quad (4.37)$$

Vektor $u^{(k)}$ je vstupem regulované soustavy, vektor $y^{(k)}$ je výstupem. Vektor $x^{(k)}$ (aproximace řešení) je stavovým vektorem (registrem) regulované soustavy. Popíšeme-li regulovanou soustavu uspořádanou čtveřicí matic, platí $\mathcal{S} = (I, I, A, 0)$. Regulátor popíšeme (v případě lineární stacionární metody) jen jednou rovnicí (rovnici výstupu)

$$u^{(k)} = Hr^{(k)}. \quad (4.38)$$

Vektor rezidua $r^{(k)}$ je vstupem a výstupem vektor $u^{(k)}$. Stavový vektor (registr) regulátoru u stacionární metody nepotřebujeme. Regulátor v podstatě slouží pouze jako zesilovač se zesílením H . Popíšeme-li regulátor uspořádanou čtveřicí matic, platí $\mathbf{R} = (0, 0, 0, H)$. Nyní ověříme správnost popisu. Do rovnice (4.38) dosadíme za reziduum. Výpočet rezidua ovšem není obsažen ani v popisu regulátoru ani v popisu regulované soustavy, reziduum se počítá jako rozdíl $r^{(k)} = b - y^{(k)}$ ve sčítacím členu (viz obrázek 4.5). Rovnici (4.38) s vyjádřeným reziduem dosadíme do rovnice (4.36), po dosazení rovnice (4.37) vyjde vztah (4.33)

$$x^{(k+1)} = (I - HA)x^{(k)} + Hb$$

popisující obecnou lineární stacionární iterační metodu.

Regulovaná soustava $\mathcal{S} = (I, I, A, 0)$ je popsána vždy stejně, nezávisle na konkrétní popisované iterační metodě – regulovaná soustava popisuje samotný problém, který řešíme (obsahuje v podstatě pouze matici A soustavy (4.29)). Popis regulátoru $\mathbf{R} = (0, 0, 0, H)$ je přímo závislý na použité metodě – regulátor tedy tuto metodu reprezentuje (obsahuje matici specifickou pro různé metody). Do takto vytvořeného popisu jsme schopni zahrnout všechny lineární stacionární iterační metody pro řešení soustav lineárních rovnic (4.29). Regulátor se strukturou $\mathbf{R} = (0, 0, 0, H)$ lze použít například u Richardsonovy, Jacobiho, Gauss-Seidlový nebo superrelaxační (SOR) metody (viz kapitolu 2.1.3).

Nestacionární a semiiterační metody pro řešení soustavy rovnic (4.29) lze schematicky znázornit také pomocí obrázku 4.5. Struktura regulované soustavy \mathcal{S} zůstává i nadále stejná, struktura regulátoru \mathbf{R} se mění. Všechny čtyři matice regulátoru jsou obecně nenulové, viz [4.8]. Využíváme stavového vektoru (registru) regulátoru, a tím do systému regulátoru zavedeme paměť. Regulátor reprezentující nestacionární metody je plnohodnotným dynamickým systémem (u stacionárních metod fungoval regulátor pouze jako zesilovač).

Na závěr zbývá jen zopakovat, že problém optimální metody řešení soustavy rovnic (4.29) můžeme z pohledu teorie řízení formulovat jako problém nalezení optimálního řízení (regulace) soustavy \mathcal{S} . Hledáme tedy regulátor \mathbf{R} optimální v takovém smyslu, aby byla regulační odchylka $r^{(k)}$ minimální, v co nejmenším počtu iterací, a zároveň, aby byl celý systém $(\mathbf{R}, \mathcal{S})$ stabilní a robustní. Problém numerické stability takto hledaného algoritmu pak můžeme v jistém smyslu převést na problém stability dynamického systému regulátoru \mathbf{R} . Dobrá podmíněnost úlohy (resp. v úloze řešení rovnice (4.29) dobrá podmíněnost matice A) pak souvisí se stabilitou regulované soustavy \mathcal{S} . Za předpokladu numerické stability algoritmu (tedy stability regulátoru \mathbf{R}) a dobré podmíněností úlohy (tedy stability soustavy \mathcal{S}) můžeme hovořit o stabilitě celého systému $(\mathbf{R}, \mathcal{S})$; viz [4.8].

5. Závěr

V této diplomové práci jsme se pokusili ukázat vybrané problémy z teorie dynamických systémů a teorie řízení také ze stránky numerické lineární algebry. V celé práci byl kladen velký důraz (vycházející z motivace této práce) na potřebu řešit stále rozsáhlejší úlohy korespondující s úlohami z teorie dynamických systémů a teorie řízení. Rešerše těchto znalostí se stala těžištěm celé první části práce.

V návaznosti na teoretickou první část diplomové práce bylo v druhé části práce uvedeno několik praktických příkladů vedoucích na rozsáhlé algebraické úlohy, zejména z oblasti fyziky piezoelektrických materiálů. Na řešení rozsáhlých algebraických úloh klasické metody založené na finitních rozkladech bohužel často nestačí. Je potřeba hledat nové především iterační algoritmy řešení algebraických problémů teorie dynamických systémů a teorie řízení. Připomeňme si v tomto směru první z příkladů – hledání vlastních frekvencí piezoelektrického rezonátoru. Tuto úlohu můžeme v zásadě řešit buď pomocí hledání vlastních čísel matice (což je relativně dobře zvládnutelný problém například pomocí QR algoritmu) nebo klasicky pomocí faktorizace polynomu (tedy hledání pólů a nul systému, což je naopak velmi obtížně řešitelný problém). Je zřejmé, že pro malé úlohy a tedy polynomy nízkého řádu jsme schopni faktorizaci provést, nikoliv však již pro úlohy, jejichž rozměry jsou v řádech několika tisíc či více.

Potřeba hledat nové postupy a algoritmy je přirozeně podmíněna znalostmi ze všech zúčastněných technických a matematických oborů. Tato diplomová práce by měla být základem stručně shrnujícím to nejdůležitější z výše uvedených oborů a měla by tak sloužit pro další práci v oboru.

Příkladem problému, na kterém lze ukázat otevřené otázky a možnosti další práce, je iterační řešení Ljapunovových rovnic (viz kapitoly 1.5.3, 1.5.4, 2.1.1)

$$AX + XA^T = -Q, \quad (5.1)$$

kde A , X , Q jsou čtvercové matice rozměrů $n \times n$. Cílem je nalézt dobrou aproximaci matice X . Jednou z možností je převedení rovnice (5.1) do tvaru standardní lineární algebraické soustavy rovnic s vektorem neznámých a vektorem pravé strany a následně použití některé vhodné krylovovské metody. Převedení na standardní tvar můžeme provést například takto

$$(I \otimes A + A \otimes I)x_V \equiv A_V x_V = -q_V, \quad (5.2)$$

kde symbolem \otimes značíme tzv. *Kroneckerův maticový součin* (*Kronecker product*), vektor x_V je vektor obsahující sloupce matice X a vektor q_V je vektor obsahující sloupce matice Q . Kroneckerův maticový součin můžeme slovně interpretovat takto: každý ij -tý prvek matice, která je vlevo, nahradíme celou maticí, která je vpravo. Tuto „vkládanou“ matici (pravou matici) však nejprve vynásobíme právě ij -tým prvkem levé matice. Matice A_V je maticí rozměrů $n^2 \times n^2$. Je také zřejmé, že vektory x_V a q_V jsou vektory délky n^2 . Soustavu rovnic pak schematicky můžeme zapsat takto

$$\left\{ \begin{bmatrix} A & 0 & \cdots & 0 \\ 0 & A & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A \end{bmatrix} + \begin{bmatrix} a_{11}I & a_{12}I & \cdots & a_{1n}I \\ a_{21}I & a_{22}I & \cdots & a_{2n}I \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}I & a_{n2}I & \cdots & a_{nn}I \end{bmatrix} \right\} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = - \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix},$$

neboli

$$\begin{bmatrix} A + a_{11}I & a_{12}I & \cdots & a_{1n}I \\ a_{21}I & A + a_{22}I & \cdots & a_{2n}I \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}I & a_{n2}I & \cdots & A + a_{nn}I \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = - \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix},$$

kde x_i , respektive q_i jsou jednotlivé sloupce matice X , respektive Q . Soustavu n^2 rovnic pro n^2 neznámých (5.2) můžeme řešit například pomocí metody GMRES. Tato metoda patří mezi metody Krylovových podprostorů, tedy mezi projekční metody. S Krylovovými podprostory i s projekčními metodami jsme se již setkali (viz kapitoly 2.1.4, 3.4). Principem projekčních metod je redukce úlohy pomocí projekce na vhodné podprostory. Úlohu rozměru n^2 , kterou je potřeba řešit, projektujeme do Krylovova podprostoru, čímž vznikne redukovaná úloha rozměru $m \ll n^2$. Redukovanou úlohu vyřešíme a řešení pak zobrazíme zpět do původního prostoru. Zdůrazňujeme, že takto popsáný matematický princip krylovovských metod (a obecně projekčních metod) není naprosto patrný z jejich algoritmičké realizace, představované obvykle rekurentními vztahy pro iterační výpočet příslušných vektorových a skalárních veličin.

Nevýhodou uvedeného postupu je to, že zobrazením úlohy na úlohu redukovanou ztratíme strukturu původní Ljapunovovy rovnice. Tedy v redukované soustavě už nelze jednoduše objevit Ljapunovovu rovnici, stala se z ní standardní soustava rovnic bez Ljapunovské struktury. Původ tohoto jevu je v tom, že použitá výpočetní metoda (realizující tuto redukci úlohy) nevyužívá podstatnou informaci o struktuře úlohy a v důsledku toho promíchá strukturu sloupce x_i . Bylo by tedy zřejmě vhodné použít takovou metodu, aby došlo k zachování struktury úlohy na redukované úrovni. Vhodné by bylo použít projekční metodu takovým způsobem, aby redukovaná úloha byla opět Ljapunovovou rovnicí.

Využití krylovovských metod pro řešení Ljapunovových rovnic se zdá být hodné dalšího zkoumání. Krylovovské metody při redukci úlohy (modelu) postupně extrahují pomocí Krylovových podprostorů dominantní informaci obsaženou v dané úloze.

LITERATURA

Kapitola 1

- [1.1] **MacFarlane, A. G. J. – Karcaniyas, N.:** *Poles and Zeros of Linear Multivariable Systems: A Survey of the Algebraic, Geometric and Complex-Variable Theory*, Int. J. Control, 1976, vol. 24, pp. 33-74;
- [1.2] **Štecha, J. – Havlena, V.:** *Teorie dynamických systémů*, Skriptum, ČVUT, FEL, 1995;
- [1.3] **Rektorys, K.:** *Přehled užití matematiky I + II*, Prometheus, s. r. o., 1995;
- [1.4] **Veit, J.:** *Integrální transformace*, SNTL – MVŠT sešit XIV, 1979;
- [1.5] **Kolmogorov, A. N. – Fomin, S. V.:** *Základy teorie funkcí a funkcionální analýzy*, SNTL – TKI, 1975;
- [1.6] **Štecha, J. – Havlena, V.:** *Moderní teorie řízení*, Skriptum, ČVUT, FEL, 2000;
- [1.7] **Kučera, V.:** *Discrete Linear Control: The Polynomial Equation Approach*, Academia, Praha a Wiley, Chichester, 1979;
- [1.8] **Kučera, V.:** *Analysis and Design of Discrete Linear Control Systems*, Academia, Praha a Prentice-Hall, London, 1991;
- [1.9] **Kučera, V.:** *Algebraická teorie diskrétního lineárního řízení*, Academia, Praha, 1978.

Kapitola 2

- [2.1] **Björck, Åke – Dahlquist, Germund:** *Numerical Mathematics and Scientific Computation – Volume II*, <http://www.math.liu.se/~akbjo/NMbook.html>;
- [2.2] **Patel, R. V. – Laub, A. J. – Van Dooren, P. M.:** *Numerical Linear Algebra Techniques for Systems and Control*, IEEE Press, New York, 1994;
- [2.3] **Fiedler, M.:** *Speciální matice a jejich použití v numerické matematice*, SNTL, Praha 1981;
- [2.4] **Hestenes, M. R. – Stiefel, E.:** *Methods of Conjugate Gradients for Solving Linear Systems*, J. Res. Nat. Bur. Standards 49, 1952, pp. 409-436;
- [2.5] **Strakoš, Z.:** *Convergence and Numerical Behavior of The Krylov Space Methods*, NATO ASI Institute Algorithms for Large Scale Linear Algebraic Systems: The State of the Art and Applications in Science and Engineering, G. Winter Althaus and E. Spedicato eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998, pp. 175-197;

- [2.6] **Jaimoukha, Imad M. – Kasenally, Ebrahim M.:** *Krylov Subspace Methods for Solving Large Lyapunov Equations*, SIAM Journal on Numerical Analysis, vol. 31, no. 1, Feb. 1994, pp. 227-251.

Kapitola 3

- [3.1] **Töpfer, P.:** *Algoritmy a programovací techniky*, Prometheus, s. r. o., 1995;
- [3.2] **Patel, R. V. – Laub, A. J. – Van Dooren, P. M.:** *Numerical Linear Algebra Techniques for Systems and Control*, IEEE Press, New York, 1994;
- [3.3] **Björck, Åke – Dahlquist, Germund:** *Numerical Mathematics and Scientific Computation – Volume I*, <http://www.math.liu.se/~akbj/NMbook.html>;
- [3.4] **Drkošová, J. – Strakoš, Z.:** *Úvod do teorie citlivosti a stability v numerické lineární algebře*, Skriptum, AVČR, 1996;
- [3.5] **Paige, Ch. C.:** *Properties of Numerical Algorithms Related to Computing Controllability*, IEEE Trans. Automat. Contr., vol. AC-26, no. 1, pp. 130-138, Feb. 1981; (reprinted in [3.2], pp. 186-194);
- [3.6] **Van Dooren, P. M.:** *The Generalized Eigenstructure Problem in Linear System Theory*, IEEE Trans. Automat. Contr., vol. AC-26, no. 1, pp. 111-129, Feb. 1981; (reprinted in [3.2], pp. 109-127);
- [3.7] **Emami-Naeini, A. – Van Dooren, P. M.:** *Computation of Zeros of Linear Multivariable Systems*, Automatica, vol. 18, no. 4, pp. 415-430, 1982; (reprinted in [3.2], pp. 271-286).

Kapitola 4

- [4.1] **Tzou, H. S. – Tseng, C. I.:** *Distributed Modal Identification and Vibration Control of Continua: Piezoelectric Finite Element Formulation and Analysis*, Journal of Dynamics Systems, Measurement, and Control, vol. 113, September 1991;
- [4.2] **Hwang, W.-S. – Park, H. Ch.:** *Finite Element Modeling of Piezoelectric Sensors and Actuators*, AJAA Journal, vol. 31, no. 5, May 1993;
- [4.3] **Chen, S.-H. – Wang, Z.-D. – Liu, X.-H.:** *Active Vibration Control and Suppression for Intelligent Structures*, Journal of Sound and Vibration, 1997;
- [4.4] **Piefort, V. – Preumont, A.:** *Modeling of Smart Piezoelectric Shell Structures with Finite Elements*, ISMA 25 – Leuven – Belgium, September 2000;
- [4.5] **Piefort, V.:** *Finite Element Modeling of Piezoelectric Active Structures*, Université Libre de Bruxelles – Faculty of Applied Sciences – Active Structures Laboratory, Department of Mechanical Engineering and Robotics, 2001;

- [4.6] **Zelenka, J.:** *Piezoelektrické rezonátory a jejich použití*, Academia, Praha, 1980;
- [4.7] **Kraus, I.:** *Struktura a vlastnosti krystalů*, Academia, Praha, 1993;
- [4.8] **Bhaya, A. – Kaszkurewicz, E.:** *Iterative Methods as Dynamical Systems with Feedback Control*, 42nd IEEE Conference on Decision and Control (CDC), Maui, HI, 2003;
- [4.9] **Brdička, M. – Samek, L. – Sopko, B.:** *Mechanika kontinua*, Academia, 2000;
- [4.10] **Sedlák, B. – Štoll, I.:** *Elektrina a magnetismus*, Academia, 2002.