Vysoká škola strojní a textilní v Liberci

nositelka Řádu práce

Fakulta strojní

obor 23-40-8

Automatizované systémy řízení výrobních procesů ve strojírenství

Katedra technické kybernetiky

Jan Wagner

Vedoucí práce : Prof.Ing. Bořivoj Hanuš CSc.,VŠST Liberec

Konzultant : Ing. Miroslav Olehla CSc.,VŠST Liberec

KTK - ASR SF - 019

Rozsah práce a příloh :

Počet stran : 50

Počet příloh : 14

Počet obrázků : 3

Počet tabulek : 1

KTK /ASŘ - S

Vysoká škola: VŠST Liberec

Katedra: technické kybernetiky

Fakulta: strojní

Školní rok: 1980/81

DIPLOMOVÝ ÚKOL

pro s. Jana W a g n e r a

obor automatizované systémy řízení výrobních procesů ve strojírenství

Protože jste splnil..... požadavky učebního plánu, zadává Vám vedoucí katedry ve smyslu směrnic ministerstva školství ~~a kultury~~ xxxxxxxx o státních závěrečných zkouškách tento diplomový úkol:

Název tématu: Řešení stability spojitých a diskrétních soustav na počítači

Pokyny pro vypracování:

1. Prostudujte z literatury numerické metody pro zjišťování stability dynamických soustav.
2. Sestavte a ověřte na příkladech programy pro zjišťování stability metodou

   pro spojité soustavy:

   - kořenů
   - Hurwitzovým kriteriem
   - Routh-Schurovým kriteriem

   pro diskrétní soustavy:

   - kořenů
   - převodů pomocí transformace
   - Mejerovým kriteriem
   - upraveným Routh-Schurovým kriteriem
3. Programy organizujte jako procedury ovládané hlavním programem.
4. Zhodnoťte časovou náročnost a přesnost programů podle jednotlivých metod.

SEVT - 49 395.0

JČT 9 - 910836-75

Rozsah grafických laboratorních prací:

Rozsah průvodní zprávy: **50 - 60 stran**

Seznam odborné literatury:

Ralsten,A.: Základy numerické matematiky. Praha, Academia 1973

Hanuš,B.: Základy technické kybernetiky. /Skripta./ Liberec, VŠST 1979

Hanuš,B.: Teorie automatického řízení I. /Skripta./ Liberec, VŠST 1980

Olehla,M. - Tišer,J.: Praktické použití FORTRANu. Praha, NADAS 1979

časopis: Algoritmy CACM

Vedoucí diplomové práce: **Prof.Ing. Bořivoj H a n u š, CSc.**

Konsultanti: **Ing. Miroslav O l e h l a, CSc.**

Datum zahájení diplomové práce: **15.9.1980**

Datum odevzdání diplomové práce: **12.6.1981**

VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ LIBEREC
fakulta strojního inženýrství

Doc.Ing. J.Alaxin,CSc.

Vedoucí katedry

Doc.RNDr B.Stříž,CSc.

Děkan

V **Liberci** dne **15.9.** 19**80**

Místopřísežně prohlašuji , že jsem diplomovou práci vypracoval samostatně a s použitím uvedené literatury .

V Liberci 11 . 6 . 1981

Obsah :

Použité označení

p	proměnná Laplaceova obrazu funkce proměnné
z	proměnná Z-obrazu funkce proměnné
$A(p)$	Laplaceův obraz spojité funkce
$A(z)$	Z-obraz nespojité funkce
$\binom{n}{k}$	binomický koeficient
$f'(x)$	derivace funkce $f(x)$
$\mathbb{H}$	označení matice
$\left(\frac{\partial A(z)}{\partial z}\right)_{z=z_i}$	hodnota derivace v bodě z_i
$x_{part}(t)$	partikulární řešení diferenciální rovnice
$x[kT]$	hodnota k-tého impulsu při periodě vzorkování
e	základ přirozených logaritmů
$\|\mathbb{H}\|$	determinant matice

1. Úvod

Rozvoj kybernetiky v současnosti přímo souvisí s rozvojem celého národního hospodářství v podmínkách nastupující vědecko technické revoluce . V podmínkách rozvinuté socialistické společnosti a tedy i ČSSR je hlavním úkolem současné hospodářské politiky zvyšování efektivnosti rozvoje národního hospodářství . Touto problematikou se zabývá i ÚV KSČ(v roce 1974 na květnovém zasedaní " K otázkám vědeckotechnického rozvoje čs.narodního hospodářství " byla jmenována kybernetizace výrobních a řídících procesů mezi rozhodujícími prostředky) a na nedávném XI.sjezdu v dubnu tohoto roku soudruh Lubomír Štrougal znovu zdůraznil ve Zprávě o hlavních směrech hospodářského rozvoje ČSSR na léta 1981-1985 význam rozvoje automatizace a kybernetizace .

Pod pojmem "kybernetizace" rozumíme zavádění matematických a matematickoekonomických metod v teorii řízení a současných technických prostředků (zejména počítačů) k řešení problémů řízení ve všech oblastech života naší společnosti .

Jedním ze základních kamenů kybernetiky je teorie zpětnovazebního automatického řízení , jejíž počátky můžeme vysledovat již koncem 19.století, kdy byla vypracována dodnes používaná algebraická kriteria stability (viz kapitola 3.).

Chování regulované soustavy je dáno technologickým procesem a konstrukcí technologického zařízení,popisu- ho jeme zpravidla diferenciální rovnicí či diferenční rovnicí nebo soustavou těchto rovnic . Tyto rovnice jsou pro odezvu lineárního časově invariantního regulačního obvodu obyčejné lineární s konstantními koeficienty . Toto chování regulačního obvodu je tedy dáno vlastnostmi bloků v něm obsažených(soustava,regulátor,..) a vlastnostmi signálů vstupujících do obvodu zvnějšku(porucha, řízení).

Podle průběhu regulačního pochodu posuzujeme jakost regulace.Nejzávažnějším kriteriem kvality regulačního pochodu je jeho stabilita. Smyslem zavádění automatické regulace je v naprosté většině technických případů dosažení stability regulačního pochodu .

Téma této práce tedy je podmíněno snahou o využití dostupné moderní výpočetní techniky pro řešení matematických metod pro stanovení podmínek stability vycházejících ze známé diferenciální či diferenční rovnice regulačního obvodu .

2. Stabilita systémů

Stabilita soustavy je dána podmínkou, aby se její výstupní veličina ustálila na konečné hodnotě, jestliže i porucha je konečná.

U lineární spojité soustavy se dosáhne stability, jestliže je tato soustava popsána diferenciální rovnicí, jíž příslušná charakteristická rovnice má kořeny obecně komplexní se zápornou reálnou částí, tj. leží v levé polovině Gaussovy roviny

$$\mathrm{Re}\, p_i < 0 \tag{2-1}$$

Levá strana charakteristické rovnice je při tom totožná se jmenovatelem obrazového přenosu upraveného na jednoduchý zlomek. Vyplývá to z řešení diferenciální rovnice příslušné lineární spojité soustavě, které je

$$x(t) = \sum_{i=1}^{n} c_i e^{p_i t} + x_{part}(t) \tag{2-2}$$

Pro konečnou hodnotu $x(t)$ v ustáleném stavu $(t \to \infty)$ je nutné, aby exponent $p_i t$ (kde p_i je kořenem charakteristické rovnice) měl zápornou reálnou složku. Pak platí

$$\lim_{t \to \infty} x(t) = \lim_{t \to \infty} x_{part}(t) \tag{2-3}$$

Obdobně můžeme vyjít při určování stability lineárních impulsních soustav z řešení průběhu výstupního signálu impulsní soustavy, který lze vyjádřit ve tvaru

$$x[kT] = \sum_{i=1}^{n} \frac{B(z_i)}{z_i A'(z_i)} z_i^{\,k} \tag{2-4}$$

kde $X(z)$ je obraz výstupního signálu

$$X(z) = \frac{B(z)}{A(z)}$$

a

$$A'(z) = \left(\frac{dA(z)}{dz} \right)_{z=z_i}$$

z_i jsou kořeny charakteristické rovnice $A(z) = 0$

Je zřejmé, že součet řady (2-4) bude konečný pro $k \to \infty$ jestliže bude splněna podmínka

$$|z_i| < 1 \tag{2-5}$$

tj. kořeny charakteristické rovnice příslušné impulsní soustavě leží uvnitř jednotkového kruhu se středem v počátku souřadnic Gaussovy roviny.

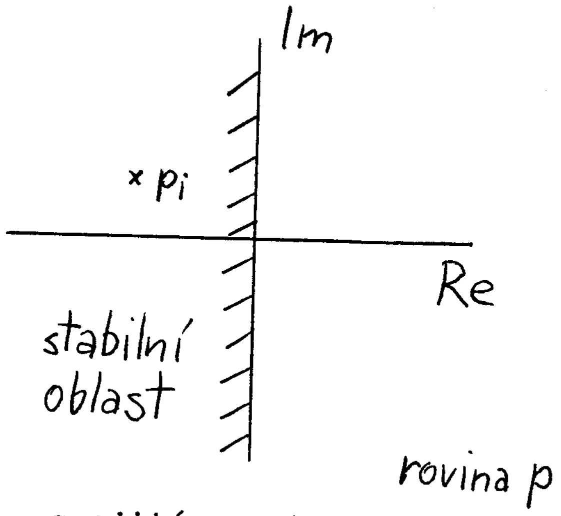
Obě podmínky (2-1) pro spojité soustavy a (2-5) pro impulsní soustavy spolu přímo souvisejí. Podle definice Z-transformace platí

$$z = e^{pt} \quad , \quad p = \frac{1}{T} \ln z \tag{2-6}$$

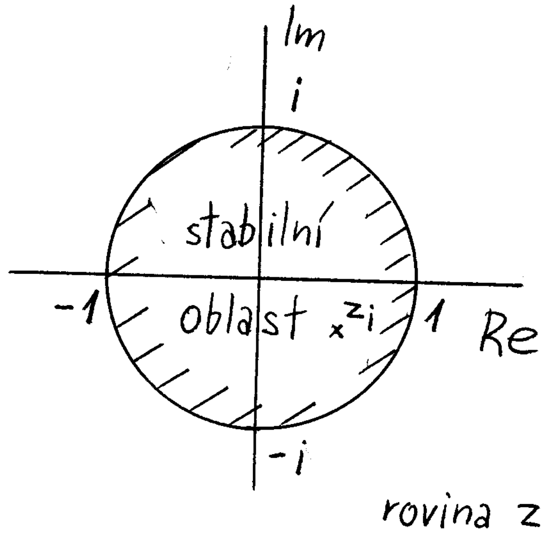
a tedy i

$$z_i = e^{p_i t} \tag{2-7}$$

tj. podmínka zápornosti reálné části kořenů p_i (2-1) je shodná s podmínkou na velikost absolutní hodnoty $|z_i|$ menší než jednotkovou (2-5). Hranici stability u spojité soustavy tvoří imaginární osa, u impulsní soustavy jednotková kružnice (obr. 2-1) .



spojitá soustava



impulsní soustava

Obr. 2-1

Z polohy kořenů charakteristické rovnice v rovině p či z lze určit zda je pochod stabilní či nestabilní, ale i zda je tlumeně kmitavý či aperiodický. Přehled o vlastnostech pochodu v závislosti na poloze kořenů dává následující tabulka.

Poloha kořenů charakt. rovnice v rovině p u spoj. soustavy	v rovině z u imp. soustavy	Pochod v časové oblasti
v levé polorovině	uvnitř jednotk. kružnice	stabilní
v pravé polorovině	vně jednotk. kružnice	nestabilní
na reálné ose	na kladné reál. ose	aperiodický
komplexně sdružené v levé polorovině	komplexně sdružené či reálné záporné uvnitř jedn. kružn.	kmitavý tlumený
na imaginární ose	na jednotkové kružnici	na mezi stability

3. Algebraická kriteria

Algebraická kriteria stability lineárních spojitých soustav umožňují rozhodnout o stabilitě bez výpočtu kořenů charakteristické rovnice soustavy, která je ve tvaru

$$f(p) = \sum_{i=0}^{n} a_i p_i = 0 \qquad (3-1)$$

3.1 Hurwitzovo kriterium stability

Z koeficientů $a_0, a_1, .. a_n$ charakteristického polynomu (3-1) sestavíme tzv. Hurwitzovu matici, která je rozměru $n \times n$.

$$H = \left[\begin{array}{ccc|c} a_{n-1} & a_{n-3} & a_{n-5} & 0 \\ a_n & a_{n-2} & a_{n-4} & 0 \\ 0 & a_{n-1} & a_{n-3} & \dot{0} \\ \hline 0 & 0 & 0 & a_0 \end{array}\right] \qquad (3-2)$$

Z Hurwitzovy matice počítáme Hurwitzovy determinanty Δ_i , které jsou rovny hlavním minorům matice H .Platí tedy

$$\Delta_1 = a_{n-1}, \ \Delta_2 = \begin{vmatrix} a_{n-1} & a_{n-3} \\ a_n & a_{n-2} \end{vmatrix}, \ldots, \ \Delta_n = a_0 \cdot \Delta_{n-1} \qquad (3-3)$$

O stabilitě soustavy můžeme rozhodnout na základě následujícího tvrzení.

<u>Věta 3.1-1.</u> Aby charakteristický polynom měl všechny kořeny se zápornou reálnou částí, je nutné a stačí,aby všechny Hurwitzovy determinanty byly kladné.

Hurwitzovo kriterium lze upravit do tvaru, který provedli Lienard a Chipard.

<u>Věta 3.1-2.</u> Jsou-li všechny koeficienty a_i charakteristického polynomu kladné,mají všechny kořeny charakteristického polynomu záporné reálné části tehdy a jen tehdy, když jsou všechny sudé nebo všechny liché Hurwitzovy determinanty kladné.

Stačí tedy vyšetřovat pouze sudé nebo liché Hurwitzovy determinanty, což usnadňuje výpočet.

3.2 Routh - Shurovo kriterium stability

Z koeficientů $a_0, a_1, \dots a_n$ charakteristického polynomu (3-1) sestavíme matici podle tohoto algoritmu

1. Do prvního řádku napíšeme postupně koeficienty a_n až a_0.
2. Každý druhý koeficient vynásobíme podílem prvého a druhého koeficientu a výsledek zapíšeme do druhého řádku s posunutím o jeden sloupec vlevo.
3. Odečteme prvky druhého řádku od prvního.
4. Jsou-li všechny prvky výsledného řádku kladné, naznačený postup opakujeme, až v posledním lichém řádku zbudou tři koeficienty.

$$
\begin{array}{cccccc|c}
 & a_n & a_{n-1} & a_{n-2} & a_{n-3} \; \dots & a_0 & \cdot \dfrac{a_n}{a_{n-1}} \\
- (& a_n & & a_{n-3}\dfrac{a_n}{a_{n-1}} & &) & \\
\hline
 & & a_{n-1} & a'_{n-2} & a_{n-3} \; \dots & a_0 & \cdot \dfrac{a_{n-1}}{a'_{n-2}} \\
- (& & a_{n-1} & & a'_{n-4}\dfrac{a'_{n-1}}{a'_{n-2}} &) & \\
\hline
 & & & a'_{n-2} & a''_{n-3} \; \dots & a_0 & \\
 & & & & & & \text{atd.} \quad (3-4)
\end{array}
$$

Věta 3.2-1. Kořeny charakteristického polynomu budou mít zápronou reálnou část, jestliže všechny prvky lichých řádků matice vytvořené podle zhora uvedeného algoritmu jsou kladné.

3.3 Metoda transformace

Konformní zobrazení roviny Z na rovinu p umožňuje řešit stabilitu impulsních soustav stejně jako u soustav spojitých a využít tak celého výpočtového aparátu, vypracovaného pro spojité soustavy.

Logaritmická transformace vyplývající z (2-6) není však pro výpočetní zpracování vhodná. Používá se proto např. transformace

$$p = \frac{z+1}{z-1} \qquad (3-5)$$

resp. inversní transformace

$$z = \frac{p+1}{p-1} \qquad (3-6)$$

která je algebraická. Tuto transformaci lze použít pro vyšetřování stability např. pomocí kriteria Hurwitze nebo Routh - Schura, protože zobrazuje imaginární osu a levou polorovinu roviny na jednotkovou kružnici a vnitřek jednotkové kružnice roviny. Nelze ji však použít při vyšetřování aperiodicity, protože nezobrazuje zápornou reálnou osu roviny na interval (0;1) reálné osy roviny. Kořenům charakteristické rovnice

$$f(z) = \sum_{i=0}^{n} a_i z_i = 0 \qquad (3-7)$$

odpovídají kořeny rovnice

$$f(p) = \sum_{i=0}^{n} a_i \left(\frac{p+1}{p-1}\right)^i = \sum_{j=0}^{n} b_j p^j = 0 \qquad (3-8)$$

Platí

$$b_j = \sum_{k=0}^{n} a_k \sum_{i=0}^{n-j} (-1)^{n-j} \binom{k}{n-j-i} \binom{n-k}{i} \qquad (3-9)$$

kde $\quad 0 \leq j \leq n \qquad \binom{\alpha}{\beta} = \dfrac{\alpha!}{\beta!\,(\alpha-\beta)!}$

$$\binom{\alpha}{0} = 1 \;, \quad \binom{\alpha}{\beta} = 0 \quad \text{pro } \alpha < \beta,\ \beta < 0$$

Pro vyšetřování aperiodicity je výhodné použít transformace

$$z = \frac{1}{1-s}$$

která nám dává vztahy pro koeficienty

$$b_j = \sum_{i=0}^{n-j} a_i (-1)^j \binom{n-i}{j} .$$

3.4 Majerovo kriterium

Aperiodicitu lineární impulsní soustavy lze vyšetřit Majerovým kriteriem , které odvodili V.M.Majerov a I.Filip na základě Sturmovy věty o kořenech algebraických rovnic.

Výpočtový algoritmus spočívá v tom , žese sestaví řádka z koeficientů charakteristické rovnice (3-1) (tak jako při aplikaci Routh - Schurova kriteria při řešení stability lineární spojité soustavy) a z rovnice vzniklé derivací charakteristické rovnice podle tak , že za každý koeficient prvé rovnice se píše vždy odpovídající koeficient druhé rovnice. Na takto vzniklou řádku se aplikuje algoritmus ROUth - Schura . Jestliže kriterium Routh S chura je splněno , pak impulsní soustava je aperiodická .

Schema algoritmu Majerova :

$$
\begin{array}{l|l}
a_n \quad n a_n \quad a_{n-1} \quad (n-1)a_{n-1} \ \cdots \ a_1 \quad a_1 \quad a_0 & \dfrac{1}{n} \\
-\left(a_n \qquad \dfrac{(n-1)a_{n-1}}{n} \qquad \qquad \dfrac{a_1}{n} \qquad \qquad \right) & \\
\hline
\quad n a_n \quad \dfrac{a_{n-1}}{n} \quad (n-1)a_{n-1} \ \cdots \ \dfrac{n-1}{n}a_1 \quad a_1 \quad a_0 & \dfrac{n^2 a_n}{a_{n-1}} \\
-\left(\quad n a_n \qquad \dfrac{n^2 a_n}{a_{n-1}} a'_{n-2} \qquad \qquad \dfrac{n^2 a_n}{a_{n-1}} a_0 \right) & \\
\hline
\qquad \dfrac{a_{n-1}}{n} \quad a''_{n-1} \ \cdots \qquad a''_0 \quad a_0 & \text{atd.}
\end{array}
$$

3.5 Rozšířené Routh Schurovo kriterium

Stabilitu lineární impulsní soustavy můžeme vyšetřit i bez transformace z roviny z do roviny p. Použijeme k tomu obdobného kriteria jako v kapitole 4.1.

Sestavíme matici rozměru $(n+1)\times(2n+1)$ z koeficientů charakteristického polynomu takto

$$\left|\begin{array}{ccccccc} a_n & a_{n-1} & a_{n-2} & \dots & a_2 & a_1 & a_0 \\ a_0 & a_1 & a_2 & \dots & a_{n-2} & a_{n-1} & a_n \\ a'_{n-1} & a'_{n-2} & a'_{n-3} & \dots & a'_1 & a'_0 & 0 \\ a'_0 & a'_1 & a'_2 & \dots & a'_{n-2} & a'_{n-1} & 0 \\ a''_{n-2} & a''_{n-3} & & & & \text{atd.} & \end{array}\right| \begin{array}{l} \cdot a_n \\ \cdot a_0 \\ \cdot a'_{n-1} \\ \cdot a'_0 \\ \\ \end{array} \qquad (3\text{-}12)$$

Každý lichý řádek získáme z předcházejících dvou takto

1. $2n-1$ -tý řádek násobíme jeho prvním prvkem
2. $2n$ -tý řádek násobíme jeho prvním prvkem
3. Odečtením $2n$ -tého řádku od $2n-1$ -tého získáme hledaný $2n+1$ -tý řádek
4. Následující sudý řádek získáme přerovnáním dle (3-12).

Aby byla soustava stabilní je nutné, aby první prvky lichých řádků byly kladné.

4. Metoda kořenů

Algebraická kriteria nám umožňují rozhodnout o stabilitě soustavy , obecně však nemáme dobré apriorní informace o poloze všech kořenů. Z toho vyplývá potřeba nalézt všechny kořeny pomocí metody, která vždy konverguje.

4.1 Metoda Lehmer - Schurova

Jako základu metody pro určení kořenů rovnice (3-1) užijeme Schurova kriteria , podle něhož se dá zjistit existence nějakého kořene uvnitř jednotkového kruhu.

Nechť je dán polynom $f(p)$ vztahem (3-1).Definujeme

$$f^*(p) = p^n \bar{f}(\bar{p}^{-1}) = \bar{a}_n + \bar{a}_{n-1} p + \dots + \bar{a}_0 p^n \qquad (4-1)$$

kde pruh značí číslo komplexně sdružené (tuto metodu lze aplikovat jak v případech komplexních , tak v případě reálných kořenů). Definujeme dále

$$T[f(p)] = \bar{a}_0 f(p) - a_n f^*(p) \qquad (4-2)$$

takže speciálně

$$T[f(0)] = \bar{a}_0 a_0 - a_n \bar{a}_n = |a_0|^2 - |a_n|^2 \qquad (4-3)$$

je reálné číslo. Polynom $T[f(z)]$ je stupně nanejvýše $n-1$ takže definujeme-li

$$T^j[f(p)] = T\{T^{j-1}[f(p)]\} \qquad (4-4)$$

dostáváme posloupnost klesajících stupňů. Budiž k nejmenší celé číslo, pro které platí $T^k[f(0)] = 0$

Věta 4.1-1. Předpokládejme, že $f(0) \neq 0$. Je-li pro nějaké h takové, že $0 < h < k$, $T^h[f(0)] < 0$, má polynom $f(p)$ nejméně jeden kořen uvnitř jednotkového kruhu. Pokud však $T^i[f(0)] > 0$ pro $1 \leq i < k$ a $T^{k-1}[f(p)]$ je konstanta, neleží uvnitř jednotkového kruhu žádný kořen.

Abychom na základě této věty rozhodli zda leží, či neleží uvnitř jednotkového kruhu kořen, budeme postupovat takto : 1. Je-li $f(0)=0$, existuje kořen $p=0$. Je-li $f(0) \neq 0$, pokračujeme podle bodu 2.

2. Vypočteme $T[f(p)]$. Je-li $T[f(0)] < 0$, existuje uvnitř jednotkového kruhu kořen. Je-li $T[f(0)] \geq 0$, pokračujeme podle bodu 3.

3. Vypočítáme postupně $T^j[f(p)]$ pro $j = 1, 2 \ldots$, až bude buď $T^j[f(0)] < 0$ pro některé $j < k$, nebo $T^k[f(0)] = 0$. V prvním případě existuje uvnitř jednotkového kruhu aspoň jeden kořen. Nastane-li druhý případ a je-li polynom $T^{k-1}[f(p)]$ konstantní, neexistuje uvnitř kruhu žádný kořen.

Mezeru ve větě 4.1-1 tj. je-li $T^k[f(0)] = 0$, ale $T^{k-1}[f(p)]$ není konstantní vyplníme později.

Abychom aplikovali uvedenou větu k nalezení kořenů rovnice (3-1) všimneme si, že má-li polynom $f(p)$ kořen uvnitř kruhu $|p - c| = \rho$, má polynom

$$g(p) = f(\rho p + c) \qquad (4\text{-}5)$$

kořen uvnitř jednotkového kruhu (polynom $g(p)$ může mít komplexní koeficienty, i když má polynom $f(p)$ reálné koeficienty). Budeme postupovat takto :

1. Nemá-li polynom $f(p)$ kořen uvnitř jednotkového kruhu , budeme uvažovat polynom $g(p) = f(2p)$ a zkoumat, má-li kořen uvnitř jednotkového kruhu. Nemá-li , budeme uvažovat polynom $f(2^2 p)$. Budeme-li takto pokračovat nalezneme dříve nebo později mezikruží

$$R = 2^j \leq |p| < 2^{j+1} = 2R \tag{4-6}$$

takové , že polynom $f(p)$ má kořen v tomto mezikruží a žádný kořen uvnitř kruhu $|p| = R$. Má-li polynom $f(p)$ kořen uvnitř jednotkového kruhu , budeme půlit poloměr tak dlouho , až nalezneme kruh , v němž neleží žádný kořen . Opět dostaneme pro mezikruží , ve kterém leží kořen , nerovnosti tvaru (4-6).

2. Totož mezikruží lze úplně pokrýt osmi překrývajícími se kruhy se středy v bodech

$$\frac{3R}{2\cos\frac{1}{8}\pi} \cdot e^{2\pi i \frac{k}{8}} \quad , \; k = 0, 1, \ldots 7 \quad , \; i = \sqrt{(-1)} \tag{4-7}$$

a o poloměrech $\frac{4}{5}R$ (tento výsledek pochází od G.R.Hagena a zlepšuje původní Lehmerův výsledek). Zkoušíme-li tyto kruhy po řadě pomocí věty 4.1-1 , nalezneme aspoň jeden , uvnitř kterého leží kořen rovnice (3-1). Jsou-li koeficienty reálné , musí ležet kořen v jednom z kruhů pro $k = 0, 1, 2, 3, 4$.

3. Označíme střed tohoto kruhu C_1 a pokračujeme jako v bodě 1 , s tím rozdílem , že nyní v každém kroku půlíme poloměr , přičemž začneme s poloměrem $\frac{4}{5}R$. Tím způsobem nalezneme mezikruží

$$R_1 = \frac{4}{5} R \cdot 2^{-j_1} \leq |p - c_1| < \frac{4}{5} R \cdot 2^{-(j_1 - 1)} = 2R_1$$

(4-8)

které obsahuje kořen polynomu $f(p)$. Stejně jako v bodě 2 pokryjeme toto mezikruží osmi kruhy a opakujeme body 2 a 3 tak dlouho , jak je třeba.

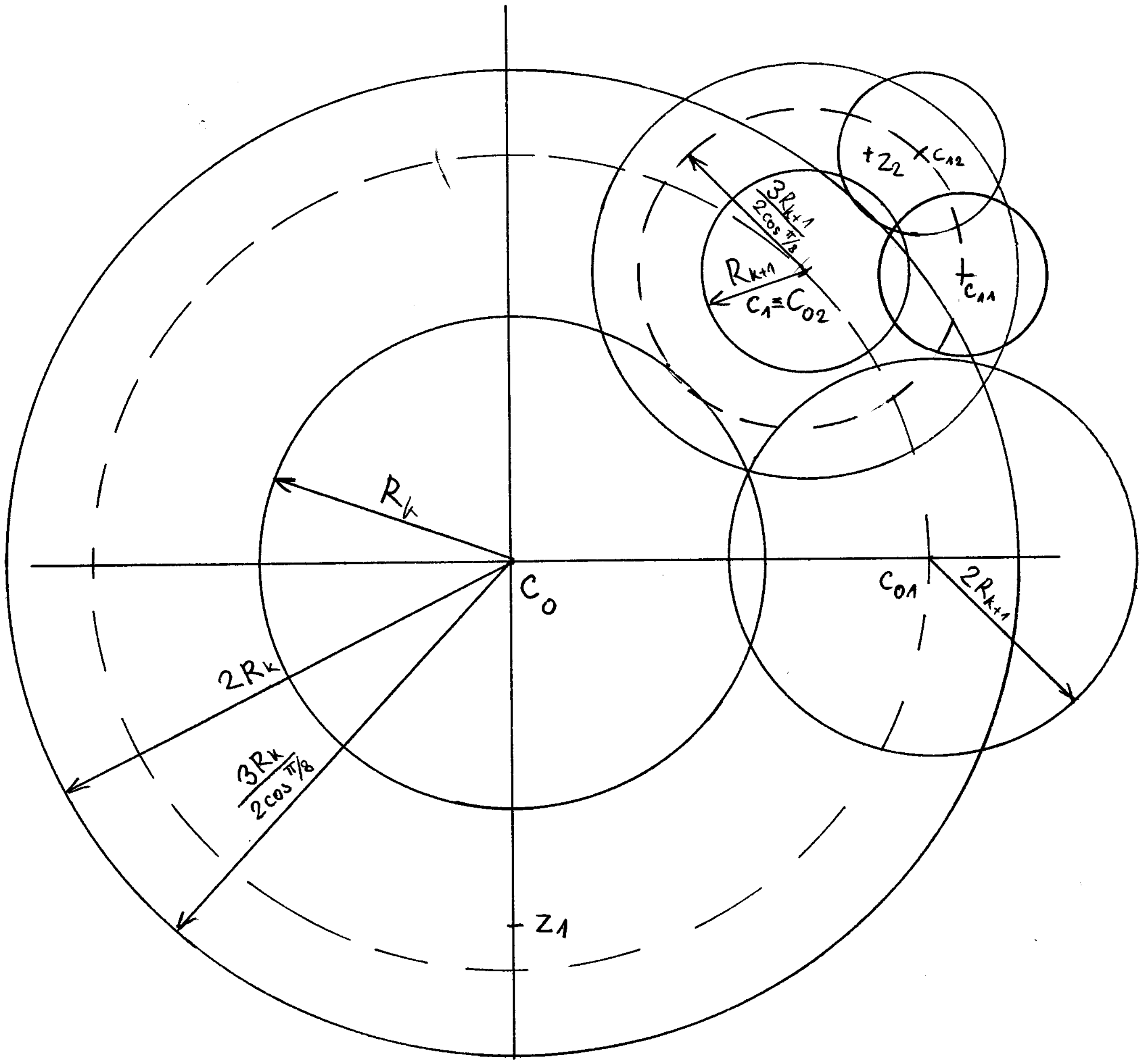
Metoda je znázorněna graficky na obr. 4 -1. Tato metoda nemusí konvergovat ke kořenu α_1 , který vymezil mezikruží (4-6) , neboť zakončení každé fáze může být určeno kořenem různým od kořene , pomocí něhož byla ukončena předešlá fáze . Je-li však R poloměr v (4-6) , dá se ukázat , že absolutní hodnota kořene , ke kterému uvažovaný proces konverguje , nepřevýší číslo $\frac{5R}{2\cos\frac{1}{8}\pi}$.

Protože j_1 v bodě ~~2~~ 3 je kladné , je $2R_1 \leq \frac{4}{5} R$ $2R_2 \leq \frac{4}{5} R_1$, a obecně , po k krocích leží kořen uvnitř kruhu o poloměru $2R_k$, přičemž platí

$$R_k \leq \left(\frac{2}{5}\right)^k R \qquad (4-9)$$

a pravděpodobně je R_k mnohem menší než tento jeho horní odhad .

Koeficienty polynomů (4-4) se postupně velmi rychle zvětšují nebo zmenšují , abychom se tomuto jevu vyhnuli , je třeba koeficienty normalizovat ; například tak ,že absolutní člen každého polynomu $T^j[f(p)]$ učiníme v absolutní hodnotě rovný jedné .



Obr. 4-1

Mezera ve větě 4.1-1 se při náhodném výběru koeficientů polynomu (3-1) projeví s pravděpodobností nula , nicméně se však projeví při $a_0 = a_n$ a v některých jednoduchých příkladech celočíselných koeficientů.

Nastane-li např. při provádění bodu 1 pro poloměr R případ , že $T^{k}[g(0)]=0$ a $T^{k-1}[g(p)]$ není konstanta , je nejjednodušší vzít nový poloměr βR , kde $\frac{1}{2} < \beta < 1$, např. $\beta = \frac{3}{4}$, a pokračovat s touto hodnotou poloměru , přičemž za další poloměr vezmeme poloměr $2\beta R$. Nastane-li tento případ v bodě 3 , použijeme za příslušné β hodnotu , pro kterou platí $1 < \beta < 2$, a pokračujeme zřejmým způsobem .

Tento postup , jehož základní myšlenka pochází od Schura a který byl rozpracován Lehmerem , nutně konverguje k nějakému kořenu rovnice (3-1) . Rychlost konvergence není na rozdíl od jiných metod (Graeffova , Laguerrova aj.) v žádném směru ovlivněna násobností kořenů ani tím, jsou-li kořeny libovolným způsobem nahromaděny .

Když jsme nalezli jeden kořen , můžeme jej odstranit z rovnice pomocí syntetického dělení a pokračovat v hledání dalších kořenů , přičemž bereme za počáteční poloměr hodnotu .

Dělíme-li polynom (3-1) výrazem $(p-p_j)$ je zbytek R_j roven číslu $f(p_j)$. platí tedy

$$f(p) = (p-p_j)\left(b_{n-1}p^{n-1} + b_{n-2}p^{n-2} + \dots + b_0\right) + R_j \qquad (4\text{-}10)$$

Abychom odvodili rekurence pro syntetické dělení , porovnáme stejné mocniny ve výrazech (3-1) a (4-10). Dostaneme

$$a_n = b_{n-1}$$

$$a_{n-1} = b_{n-2} - p_j b_{n-1}$$

. . .

$$a_k = b_{k-1} - p_j b_k$$

. . .

$$a_0 = R_j - p_j b_0 \qquad (4\text{-}11)$$

takže koeficienty b_k lze počítat z rekurencí

$$b_k = a_{k+1} + p_j b_{k+1} \quad , \; k = 0, 1 \ldots n-1 \; ; \; b_n = 0 \qquad (4\text{-}12)$$

Z poslední rovnice (4-11) dostáváme

$$R_j = a_0 + p_j b_0 \qquad (4\text{-}13)$$

Vzhledem k tomu , že koeficienty rovnice (3-1) jsou čísla reálná,a tedy její kořeny jsou buď reálné nebo komplexně sdružené , můžeme místo lineárním činitelem $(p - p_j)$ dělit kvadratický činitelem $(p^2 + q_j p + r_j)$. Můžeme tedy psát

$$f(p) = (p^2 + q_j p + r_j)(b_{n-2} p^{n-2} + \ldots + b_0) + R_j p + S_j \qquad (4\text{-}14)$$

Dostaneme rekurentní vztahy

$$b_k = a_{k+2} - q_j b_{k+1} - r_j b_{k+2} \quad , k = n-2, \dots 0$$

$$b_{n-1} = b_n = 0$$

(4-15)

kde

$$q_j = -2\,\mathrm{Re}(p_j)$$

$$r_j = |p_j|^2$$

(4-16)

Pokud po vydělení obdržíme polynom stupně $n = 2$ nebo $n = 1$, zbývající kořeny nebo kořen vypočítáme pomocí známých vztahů pro výpočet kvadratické či lineární rovnice .

5. Přesnost programů a jejich časová náročnost

Pmro posuzování programů můžeme použít dvě kriteria : přesnost a časovou náročnost programu . Na obě působí tyto faktory - metoda výpočtu a jeho algoritmizace pro sestavení programu , počet prováděných operací při výpočtu výsledku , rychlost prováděných operací počítačem , délka slova v počítačové paměti a další.

Zdroje chyb můžeme rozdělit zhruba do dvou částí . První typ chyb spočívá v tom , že často nahrazujeme nekonečný proces procesem konečným . Tento typ chyby ve všech jeho podobách označujeme jako typ metody . Druhý důležitý zdroj chyb je způsoben skutečností že téměr nikdynelze aritmetické výpočty provést s úplnou přesností . Většina čísel má nekonečné dekadické vyjádření , které musí být zaokrouhleno . Ale i v tom případě , kdy lze vstupní údaje úlohy vyjádřit přesně konečným dekadickým vyjádřením , můžeme dělením získat čísla , které je nutno zaokrohlit , a násobaní může dávat více desetiných míst , nez je možné uložit v paměti počítače . Chyba , které se dopouštíme zaokrouhlením čísla , se nazývá zaokrouhlovací chyba . Má náhodný charakter , a proto je nesnadné ji vyšetřit . Při výpočtu na samočiném počítači EC 1033 lze zaokrouhlovací chybu omezit použitím proměných s dvojnásobnou délkou slova , avšak za cenu zvětšení rozsahu použité paměti a rychlosti výpočtu .

Při hodnocení časové náročnosti programů se nezajímame ani tak o absolutní rychlost , jako spíše o relativní rychlost , kterou jsou prováděny různé typy výpočtů.

5.1 Přesnost výpočtu algebraických kriterií

Při výpočtech prováděných podprogramy HURW , SHURE, MAJER,X ISHURE , tedy podprogramů využívajících algoritmu Hurwitzova (viz kapitola 3.1) , Routh - Schurova (viz kapitola 3.2) , Majerova (viz kapitola 3.4) , a rozšířeného kriteria Routh - Schurova (viz kapitola 3.5) lze očekávat chybný výsledek vlivem zaokrouhlovací chyby v průběhu jednotlivých výpočtů , jestliže zkoumané soustavy budou v oblasti meze stability či aperiodicity, nebo v jejich blízkosti . Jak již bylo výše uvedeno možnost chybného výsledku lze úspěšně omezit použitím dvojnásobné délky slova . Chyby jednotlivých metod jsou prakticky shodné . Je to způsobeno obdobným algoritmem výpočtu .

Při výpočtu prováděných podprogramy TRANS a TRANA také dochází v průběhu výpočtuk zaokrouhlovacím chybám. Vznik chyby je vzhledem k použitým algoritmům pravděpodobnější při provádění podprogramu TRANS (viz kapitola 3.3) . Tato chyba se ještě znásobuje použitím následujícím podprogramu HURW nebo SHURE , které provádíme pro zjištění stability soustavy . Pro kontrolu stability lineární impulsní soustavy je proto z hlediska větší přesnosti výhodnější použít přímý výpočet pomocí podprogramu ISHURE.

Je zřejmé , že s rostoucím stupněm polynomu pores- počtu te,vzhledem k zvýšení na sebe navazujících operací, poroste i chyba výpočtu .

5.2 Přesnost výpočtu metody kořenů

Algoritmus hledání kořene polynomu (3.1) zakončujeme jestliže platí

$$2R_k \leq \varepsilon \qquad (5.1)$$

kde volíme na základě použité jednoduché aritmetiky $\varepsilon = 10^{-6}$. Při následujícím syntetickém dělení (viz kapitola 4.1) vzniká problém zaokrouhlovacích chyb. Koeficienty nově vzniklého polynomu nejsou přesnými koeficienty ~~xxjmkm~~ polynomu , z něhož vypočítáváme další kořeny původního polynomu . Tato chyba v koeficientech nutně ovlivňuje přesnost vypočtených kořenů .

Nechť přesný polynom je

$$F(p) = \sum_{i=0}^{n} A_i z^i \qquad (5.2)$$

a definujeme

$$\delta_i = A_i - a_i \qquad (5.3)$$

kde $i = 0, 1, 2 .. n$

Je-li vypočtený kořen Z_0 a přesný kořen z_0 , je

$$Z_0 = z_0 + \varepsilon_0 \tag{5.4}$$

kde ε_0 je obecně komplexní číslo , které potřebujeme odhadnout . Dosadíme-li pak (5.4) a (5.3) do (5.2) dostaneme (viz literatura (3))

$$\sum_{i=0}^{n} \delta_i z_0^i + \varepsilon_0 f'(z_0) \approx 0 \tag{5.5}$$

a tedy platí

$$|\varepsilon_0| \approx \frac{\left|\sum_{i=0}^{n} \delta_i z_0^i\right|}{|f'(z_0)|} \tag{5.6}$$

Tento odhad má jedno zřejmé omezení , které se projeví jestliže $|f'(z_0)| \doteq 0$. V případě polynomů vyšších stupňů však ε_0 může vzrůst nad únosnou mez (viz literatura (3)) . Polynomy,u kterých malá změna v koeficientech může způsobit velkou změnu v jednom nebo více kořenů , nazýváme špatně podmíněné . Tato změna kořenů je výraznější u kořenů s větší absolutní hodnotou , metoda Lehmer - Schurova dává proto lepší výsledky , protože se v jejím případě vypočítávají kořeny v pořadí vzrůstající absolutní hodnoty .

V literatuře (2) je uveden podprogram , který vyrovnává tuto chybu metodou Peterse a Wilkinsona .

5.3 Časová náročnost

Již při sestavení algoritmů pro jednotlivé metody je zřejmé , že jejich časová náročnost bude značně rozdílná . Algebraická kriteria jsou časově mnohem méně náročnější proti metodě kořenů jak při kontrole stability lineárních spojitých soustav, tak lineárních impulsních soustav . Rozdíl však je v kvalitě informace o stabilitě soustavy , jak jsme již uvedli v kapitole 2..

Ze tří porovnávaných metod pro řešení stability lineárních spojitých soustav je"nejrychlejší"Routh - Schurovo kriterium , následuje kriterium Hurwitzovo a "nejpomalejší" je použití metody kořenů . Časová náročnost těchto metod je také závislá na stupni zkoumaného polynomu n . U Routh - Schurova kriteria je tato závislost zhruba kvadratická

$$t_n \doteq a \cdot n^2 \tag{5.7}$$

u Hurwitzova kubická

$$t_n \doteq b n^3 \tag{5.8}$$

a u metody kořenů je lineární

$$t_n \doteq c n \tag{5.9}$$

Ze vztahu (4-9) můžeme odvodit odhad počtu nutných iterací pro dosažení kořene s přesností . Dosadíme-li do výrazu (4-9) tyto odhady vyplývající přímo z algoritmu

$$2R_k \leq \varepsilon \quad , \quad R \leq |p_i| \tag{5.10}$$

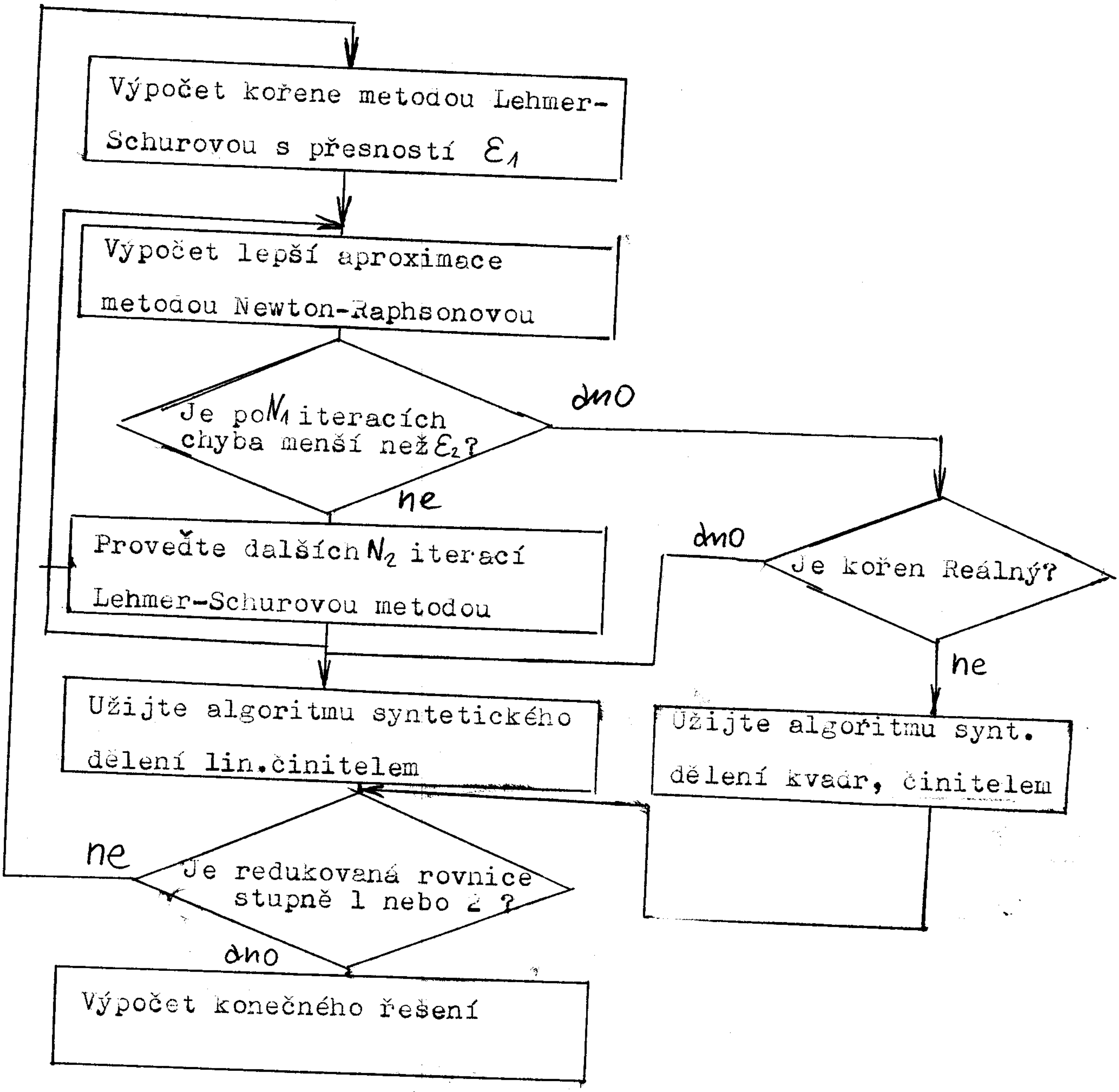
dostaneme vztah

$$\frac{\varepsilon}{2} \leq \left(\frac{2}{5}\right)^k |p_i| \tag{5.11}$$

který můžeme upravit na nerovnost

$$k \leq \alpha \left(\ln|p_i| - \ln\varepsilon\right) + \beta$$
$$\alpha \doteq 1{,}09 \qquad \beta \doteq 0{,}76 \tag{5.12}$$

ve které je jasně vidět závislost časové náročnosti na velikosti kořene ke kterému algoritmus konverguje a na přesnosti výpočtu . Pokud $|p_i| \leq \varepsilon$ algoritmus dává výsledek $p_i = 0$.



Obr . 5 . 1

Na obrázku 5.1 je blokové schema kombinované metody pro výpočet kořenů polynomu , která konverguje rychleji než vlastní metoda Lehmer-Schurova,tj. její časová náročnost je nižší . Používá se v ní rychle , ale ne vždy konvergující metody Newton - Raphsonovy . Tento nedostatek je odstraněn kombinací s Lehmer - Schurovou metodou , pomocí které získáme aproximaci kořene takovou , která vede ke konvergenci Newton - Raphsonovy metody .

Jistou nevýhodou je však závislost konstant na řešeném polynomu . Tímto algoritmem bychom mohli získat podprogram časově méně náročný s prakticky stejnou přesností .

5.4 Kontrolní příklady

Pro kontrolní příklady byly použity příklady soustav zadaných v literatuře (6) a (7) tj. s charakteristickou rovnicí soustavy lin. spojité

$$3p^4 + 5p^3 + 4p^2 + 3p + 1 = 0$$

a s charakteristickou rovnicí lin.impulsní soustavy

$$z^3 - \frac{7}{4} z^2 + \frac{7}{8} z - \frac{1}{8} = 0$$

které jsou stabilní .

Výsledky kontrolních příkladů jsou uvedeny v příloze II.

6 . Popis programů

V této části jsou popsány jednotlivé podprogramy použité pro výpočet stability lineárních soustav . Jejich výpis je uveden v příloze .

6.1 SZERO

Použití : kontrola nezápornosti reálných částí kořenů polynomu

$$a_n p^n + a_{n-1} p^{n-1} + \ldots + a_1 p + a_0 = 0$$

LOGICAL FUNCTION SZERO (N , A , KOREN)

Logická funkce nabývá hodnoty .TRUE. , jestliže mají všechny kořeny polynomu zápornou reálnou část , a hodnoty .FALSE. v ostatních případech.

N je jednoduchá proměnná typu INTEGER , jako její velikost zadává uživatel stupeň polynomu n .

A je jednorozměrné pole typu REAL , do kterého uživatel umístí koeficienty polynomu tak , že koeficient a_i bude umístěn ~~xxx~~ jako indexovaná proměnná A(I+1) .

KOREN je jednorozměrné pole typu COMPLEX , do kterého podprogram ukládá vypočtené hodnoty kořenů polynomu . Je rozměru N .

Podprogram využívá podprogram ZERO pro výpočet kořenů polynomu a kontroluje jejich reálnou část .

6.2 DZERO

Použití : kontrola absolutní hodnoty kořenů polynomu s reálnými koeficienty

$$a_n z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0 = 0$$

LOGICAL FUNCTION DZERO (N , A , KOREN)

Logická funkce nabývá hodnoty .TRUE. , jestliže mají všechny kořeny polynomu absolutní hodnotu menší než jedna , a hodnoty .FALSE. v ostatních případech .

N je jednoduchá proměnná typu INTEGER , jako její velikost zadává uživatel stupeň polynomu n .

A je jednorozměrné pole typu REAL , do kterého uživatel umístí koeficienty polynomu tak , že koeficient a_i bude umístěn jako indexovaná proměnná A(I+1) .

KOREN je jednorozměrné pole typu COMPLEX , do kterého podprogram ukládá vypočtené hodnoty kořenů polynomu . Je rozměru N .

Podprogram využívá podprogram ZERO pro výpočet kořenů polynomu a kontroluje jejich absolutní hodnotu .

6.3 ZERO

Použití : nalezení všech kořenů polynomu s reálnými koeficienty

$$\sum_{i=0}^{n} a_i p^i = 0$$

SUBROUTINE ZERO (N , A , KOREN , EPS)

N   je jednoduchá proměnná typu INTEGER , jako její velikost zadává uživatel stupeň polynomu n .

A   je jednorozměrné pole typu REAL , do kterého uživatel umístí koeficienty polynomu tak , že koeficient a_i bude umístěn jako indexovaná proměnná A(I+1) .

KOREN   je jednorozměrné pole typu COMPLEX , do kterého podprogram ukládá vypočtené hodnoty kořenů polynomu . Je rozměru N .

EPS   je požadovaná přesnost výpočtu , volíme ji vzhledem k použité metodě a použitému typu proměnných 10^{-6} .

Podprogram vypočítává všechny kořeny polynomu s reálnými koeficienty metodou Lehmer-Schurovou . Podprogram používá ve výpočtu podprogram KRLS a podprogram BINOM .

Popis metody Lehmer-Schurovy je v kapitole 4.1 .

6.4 KRLS

Použití : zjištění existence kořene polynomu s komplexními koeficienty

$$\sum_{i=0}^{m-1} b_i z^i = 0$$

uvnitř kruhu o poloměru ρ a středu c .

LOGICAL FUNCTION KRLS (M , B , RAD , CEN , ZM)

M je jednoduchá proměnná typu INTEGER , jako její hodnotu uživatel zadává hodnotu m , kde $m=n+1$ je stupeň polynomu .

B je jednorozměrné pole typu COMPLEX , do kterého uživatel zadává koeficienty polynomu tak , že koeficient b_i bude umístěn jako indexovaná proměnná B(I+1) . Je rozměru M .

RAD je jednoduchá proměnná typu REAL , jako její hodnotu uživatel zadává hodnotu ρ .

CEN je jednoduchá proměnná typu COMPLEX , jako její hodnotu uživatel zadává hodnotu c .

ZM je jednoduchá proměnná typu LOGICAL ,jako její hodnotu uživatel zadává .TRUE. ,jestliže v případě nejednoznačnosti použitého kriteria má podprogram uvažovat hodnotu $\frac{3}{4}R$ místo R , a hodnotu .FALSE.,pokud má uvažovat hodnotu $\frac{3}{2}R$.

Podprogram používá upraveného Schurova kriteria dle kapitoly 4.1 . Ve výpočtu používá podprogram BINOM .

6.5 BINOM

Použití : výpočet binomických koeficientů

$$\binom{k}{l} = \frac{k!}{l!\,(k-l)!}$$

$$\binom{k}{l} = 1 \text{ pro } l=0 \text{ a } \binom{k}{l} = 0 \text{ pro } k<l \text{ a } l<0$$

INTEGER FUNCTION BINOM (K , L)

K jednoduchá proměnná typu INTEGER

L jednoduchá proměnná typu INTEGER

6.6 DET

Použití : výpočet determinantu.

REAL FUNCTION DET (N , A)

N jednoduchá proměnná typu INTEGER , kterou uživatel zadává stupeň determinantu .

A je dvourozměrné pole typu REAL o dimenzi do kterého uživatel umístí prvky matice .

Podprogram používá k výpočtu determinantu metodu eliminace s výběrem hlavního prvku. Je podrobně popsán v literatuře (11) .

6.7 MINOR

Použití : kontrola nezápornosti hlavních minorů stupně i čtvercové matice stupně n.

LOGICAL FUNCTION MINOR (N , I , H , SUB)

Logická funkce nabývá hodnoty .TRUE. pokud hlavní minor matice H ,která je stupně n , je kladný.

N je jednoduchá proměnná typu INTEGER , kterou uživatel udává rozměr čtvercové matice n .

I je jednoduchá proměnná typu INTEGER ,kterou uživatel zadává stupeň zkoumaného hlavmího minoru matice i .

H je dvourozměrné pole typu REAL o dimenzi $n \times n$ do kterého uživatel umístí prvky matice H

SUB je dvourozměrné pole typu REAL o dimenzi $i \times i$ do kterého podprogram vytváří prvky minoru matice .

Podprogram vytváří z prvků matice H její hlavní minor stupně i . Dále vypočítá pomocí podprogramu DET velikost determinantu takto vzniklé matice a zjišťuje jeho kladnost .

6.8 HURW

Použití : kontrola nezápornosti reálných částí kořenů polynomu

$$\sum_{i=0}^{n} a_i p^i = 0$$

LOGICAL FUNCTION HURW (N , A)

Logická funkce nabývá hodnoty .TRUE. , jestliže mají všechny kořeny polynomu zápornou reálnou část , a hodnoty .FALSE. v ostatních případech .

N   je jednoduchá proměnná typu INTEGER , jako její velikost zadává uživatel stupeň polynomu n .

A   je jednorozměrné pole typu REAL ,do kterého uživatel umístí koeficienty polynomu tak , že koeficient a_i bude umístěn jako indexovaná proměnná A(I+1) .

Podprogram kontroluje zápornost reálné části kořenů polynomu pomocí Hurwitzova kriteria . Popis tohoto algebraického kriteria je uveden v kapitole 3.1.

Podprogram využívá při výpočtu podprogramu MINOR.

6.9 SHURE

Použití : kontrola zápronosti reálných částí kořenů polynomu

$$\sum_{i=0}^{n} a_i p^i = 0$$

LOGICAL FUNCTION SHURE (N , A)

Logická funkce nabývá hodnoty .TRUE. , jestliže mají všechny kořeny polynomu zápornou reálnou část , a hodnoty .FALSE. v ostatních případech .

N   je jednoduchá proměnná typu INTEGER , jako její velikost zadává uživatel stupeň polynomu n .

A   je jednorozměrné pole typu REAL , do kterého uživatel umístí koeficienty polynomu tak , že koeficient a_i bude umístěn jako indexovaná proměnná A(I+1) .

Podprogram kontroluje zápornost reálné části kořenů polynomu pomocí ROuth - Schurova kriteria . Popis tohoto kriteria je uveden v kapitole 3.2 .

6.10 TRANS

Použití :pro transformaci polynomu s reálnými koeficienty

$$\sum_{i=0}^{n} a_i z^i = 0$$

z roviny z do roviny p pomocí transformace

$$z = \frac{p+1}{p-1}$$

SUBROUTINE TRANS (N , A , B)

N je jednoduchá proměnná typu INTEGER , jako její hodnotu zadává uživatel stupeň polynomu n .

A je jednorozměrné pole typu REAL , do kterého uživatel umístí koeficienty polynomu tak , že koeficient a_i bude umístěn jako indexovaná proměnná A(I+1) .

B je jednorozměrné pole typu REAL , do kterého podprogram vypočítává koeficienty transformovaného polynomu .

Podprogram vypočítává koeficienty transformovaného polynomu pomocí algoritmu uvedeného v kapitole 3.3. Tato transformace je výhodná pro další kontrolu zápornosti kořenů polynomu pomocí podprogramu HURW nebo SHURE .

6.11 TRAXNA

Použití : pro transformaci polynomu s reálnými koeficienty

$$\sum_{i=0}^{n} a_i z^i = 0$$

z roviny z do roviny p pomocí transformace

$$z = \frac{1}{1-s}$$

SUBROUTINE TRANA N (N , A , B)

N je jednoduchá proměnná typu INTEGER , jako její hodnotu zadává uživatel stupeň polynomu n .

A je jednorozměrné pole typu REAL , do kterého uživatel umístí koeficienty polynomu tak , že koeficient a_i bude umístěn jako indexovaná proměnná A(I+1) .

B je jednorozměrné pole typu REAL , do kterého podprogramm vypočítává koeficienty transformovaného polynomu .

Pod program vypočítává koeficienty transformovaného polynomu pomocí algoritmu uvedeného v kapitole 3.3. Tato transformace je vhodná pro následující kontrolu aperiodicity pomocí podprogramu MAJER .

Podprogram využívá při výpočtu podprogram BINOM .

6.12 ISHURE

Použití : kontrola absolutní hodnoty kořenů polynomu s reálnými koeficienty

$$\sum_{i=0}^{n} a_i z^i = 0$$

pomocí rozšířeného Routh - Schurova kriteria .

LOGICAL FUNCTION ISHURE (N , A)

Logická funkce nabývá hodnoty .TRUE. , jestliže mají všechny kořeny polynomu ~~zápornou reálnou~~ absolutní hodnotu menší než jedna , a hodnoty .FALSE. v ostatních případech .

N je jednoduchá proměnná typu INTEGER , jako její hodnotu zadává uživatel stupeň polynomu n .

A je jednorozměrné pole typu REAL , do kterého uživatel umístí koeficienty polynomu tak , že koeficient a_i bude umístěn jako indexovaná proměnná A(I+1) .

Podprogram kontroluje velikost absolutní hodnoty kořenů polynomu pomocí rozšířeného Routh - Schurova kriteria uvedeného v kapitole 3.5 .

6.13 MAJER

Použití : zjištění existence reálného kořene polynomu

$$\sum_{i=0}^{n} a_i p^i = 0$$

LOGICAL FUNCTION MAJER (N , A)

Logická funkce nabývá hodnoty .TRUE. , jestliže kořeny polynomu jsou reálné , a hodnotu .FALSE. v ostatních případech .

N	je jednoduchá proměnná typu INTERGER , jako její hodnotu zadává uživatel stupeň polynomu n .

A	je jednorozměrné pole typu REAL , do kterého uživatel umístí koeficienty polynomu tak , že koeficient a_i bude umístěn jako indexovaná proměnná A(I+1) .

Podprogram zjišťuje , zda jsou kořeny polynomu reálné pomocí Majerova kriteria uvedeného v kapitole 3.4.

7. Závěr

V práci je uveden přehled numerických metod pro zjišťování stability dynamických soustav , vypracovaný na základě studia literatury . Pro řešení stability metodou kořenů byla vybrána numerická metoda hledání kořenů Lehmer - Schurova , jejíž výhodou je , že je vždy konvergentní .

Pro všechny uvedené metody byly sestaveny programy v jazyce FORTRAN , které byly zpracovávány na samočinném počítači EC 1033 . Programy byly zorganizovány jako procedury ovládané hlavním programem . Byla zhodnocena přesnost a časová náročnost jednotlivých metod a uvedeny některé možnosti zlepšení stávajících algoritmů .

8. Seznam příloh

I. Výpisy programů

jméno podprogramu	číslo přílohy
SZERO	I/1
DZERO	I/2
ZERO	I/3
KRLS	I/4
BINOM	I/5
DET	I/6
MINOR	I/7
HURW	I/8
SHURE	I/9
TRANS	I/10
TRANA	I/11
ISHURE	I/12
MAJER	I/13

II. Výsledky kontrolních příkladů

Literatura :

knihy :

(1) Aramovič J. G., Lunc G. L., Elsgolc L. E. : Funkce komplexnej premennej, Operátorový počet, Teorie stability
ALFA, Bratislava 1973

(2) Madsen K., Reid J.K. : Fortran subroutines for finding polynomial zeros
AERE Harwell, Oxford 1975

(3) Ralston A. : Základy numerické matematiky
Academia, Praha 1978

(4) Švec J., Kotek Zd. a kolektiv : Teorie automatického řízení
SNTL, Praha 1969

(5) Vogel J. : Programování v jazyku FORTRAN
SNTL, Praha 1976

skripta

(6) Hanuš B. : Základy technické kybernetiky I.
VŠST Liberec 1979

(7) Hanuš B. : Základy teorie lineárního impulsního obvodu I.
VŠST Liberec 1972

(8) Nekvinda M., Šrubař J., Vild J. : Numerická matematika
VŠST Liberec 1976

(9) Olehla M. : Programování, programovací jazyky a operační systémy
Král F.
Švarc I.
Tišer J.
VŠST Liberec 1980

(10) Olehla M. : FORTRAN IV
Tišer J.
VŠST Liberec 1975

(11) Olehla M. : Použití FORTRANU v praxi
Tišer J.
VŠST Liberec

(12) Strejc V. : Teorie automatického řízení diskrétních systémů
ČVÚT Praha 1979

časopisy, příručky :

(15) časopis Communications of the ACM
(různá čísla a ročníky)

(16) separát Collected algorithms from CACM

(17) Pokyny pro uživatele EC 1033 VŠST Liberec