Identifikace modelu náhodného signálu

doc. Ing. Petr Blaha, PhD.

Princip spektrální estimace

Odhad modelu náhodného signálu

- odhad spektra vstupního a výstupního signálu lze provést pomocí Fourierovy transformace
- výpočet přenosu

$$G(\jmath\omega) = \frac{1}{k}\sum_{i=1}^{k}\frac{Y_i(\jmath\omega)}{U_i(\jmath\omega)}$$

- podíl dvou veličin s normálním rozložením má Cauchyho rozdělení
  - definiční integrály pro střední hodnotu a rozptyl divergují
- Fourierova transformace není příliš vhodná pro analýzu náhodných signálů
- částečným řešením je použití vhodného okna
  - okno je nutné volit na základě přibližné znalosti spektra

Metoda nejmenších čtverců

- pokud je náhodný signál stacionární, jeho model je stabilní, s nulami uvnitř jednotkové kružnice
- metoda nejmenších čtverců negarantuje získání modelu s uvedenými vlastnostmi
- model diskrétního náhodného signálu ve většině případů nemůže odpovídat diskrétnímu ekvivalentu spojitého systému
- metody využívající vybělení chyby predikce nejsou jednorázové, vyžadují opakování výpočtu

Procesy AR

Definice AR procesu

AR - Auto Regressive

$$\sum_{k=0}^{p} a_p(k)x(n-k) = e(n)$$

Přenos v Z-transformaci

$$\frac{1}{A_p(z)} = \frac{X(z)}{E(z)}$$

Autokorelační metoda YW

Vyjdeme z diferenční rovnice $\sum_{k=0}^{p} a_p(k)x(n-k) = e(n)$
Střední hodnota

$$E\left\{\sum_{k=0}^{p} a_p(k)x(n-k)x(n-l)\right\} = E\left\{e(n)x(n-l)\right\}$$

pro $l > 0$ dostaneme Yule-Walkerovu rovnost

$$\sum_{k=0}^{p} a_p(k)r_{xx}(l-k) = 0 \qquad r_{xx}(i) = E\left\{x(n)x(n+i)\right\}$$

za předpokladu $a_p(0) = 1$, $r_{xx}(i) = r_{xx}(-i), l > 0$ dostaneme

$$r_{xx}(l) = -a_p(1)r_{xx}(l-1) - a_p(2)r_{xx}(l-2) - \cdots - a_p(p)r_{xx}(l-p)$$

Maticový zápis

$$\begin{pmatrix} r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(p-1) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(p-2) \\ \vdots & \vdots & \cdots & \vdots \\ r_{xx}(p-1) & r_{xx}(p-2) & \cdots & r_{xx}(0) \end{pmatrix} \begin{pmatrix} a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{pmatrix} = \begin{pmatrix} -r_{xx}(1) \\ -r_{xx}(2) \\ \vdots \\ -r_{xx}(p) \end{pmatrix}$$

$$\boldsymbol{R}_p \boldsymbol{a}_p = -\boldsymbol{r}_p$$

Parametry AR modelu lze vypočítat pomocí vztahu

$$\boldsymbol{a}_p = -\boldsymbol{R}_p^{-1} \boldsymbol{r}_p$$

Vzorec vypadá jednoduše, ale jeho výpočet je pro velké p (které je často vyžadováno) časově náročný.

Srovnání s IVM

$$x(k) = -a_p(1)x(k-1) - \cdots - -a_p(p)x(k-p) + C(q^{-1})e(k)$$

$$\varphi^T(k) = \begin{pmatrix} -y(k-1) & \cdots - y(k-p) \end{pmatrix}$$

$$z^T(k) = \begin{pmatrix} -y(k-n_c) & \cdots - y(k-p-n_c) \end{pmatrix}$$

$$\theta^T(k) = \begin{pmatrix} -a_p(1) & \cdots - a_p(p) \end{pmatrix}$$

$$\begin{pmatrix} r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(p) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(p-1) \\ r_{xx}(2) & r_{xx}(1) & \cdots & r_{xx}(p-2) \\ \vdots & \vdots & \cdots & \vdots \\ r_{xx}(p) & r_{xx}(p-1) & \cdots & r_{xx}(0) \end{pmatrix} \begin{pmatrix} 1 \\ a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{pmatrix} = \begin{pmatrix} V_p \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Matice nalevo je symetrická Toeplitzova matice a V_p je rozptyl šumu $e(n)$. Předpokládejme, že máme řešení $\boldsymbol{a}_p$ a chceme určit $\boldsymbol{a}_{p+1}$. Předchozí rovnice se rozšíří na

$$\begin{pmatrix} r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(p) & r_{xx}(p+1) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(p-1) & r_{xx}(p) \\ r_{xx}(2) & r_{xx}(1) & \cdots & r_{xx}(p-2) & r_{xx}(p-1) \\ \vdots & \vdots & \cdots & \vdots & \\ r_{xx}(p+1) & r_{xx}(p) & \cdots & r_{xx}(1) & r_{xx}(0) \end{pmatrix} \begin{pmatrix} 1 \\ a_p(1) \\ a_p(2) \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} V_p \\ 0 \\ 0 \\ \vdots \\ \alpha_p \end{pmatrix}$$

Poslední řádek definuje α_p. Pro symetrickou Toeplitzovu matici platí, že se dá napsat ve tvaru

Levinson Durbin algoritmus

$$\begin{pmatrix} r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(p) & r_{xx}(p+1) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(p-1) & r_{xx}(p) \\ r_{xx}(2) & r_{xx}(1) & \cdots & r_{xx}(p-2) & r_{xx}(p-1) \\ \vdots & \vdots & \cdots & \vdots & \\ r_{xx}(p+1) & r_{xx}(p) & \cdots & r_{xx}(1) & r_{xx}(0) \end{pmatrix} \begin{pmatrix} 0 \\ a_p(p) \\ a_p(p-1) \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_p \\ 0 \\ 0 \\ \vdots \\ V_p \end{pmatrix}$$

Předchozí dvě rovnice sečteme s tím, že druhou vynásobíme $\rho_p = -\frac{\alpha_p}{V_p}$

$$\boldsymbol{R}_{p+1} \begin{pmatrix} 1 \\ a_p(1) + \rho_p a_p(p) \\ a_p(2) + \rho_p a_p(p-1) \\ \vdots \\ \rho_p \end{pmatrix} = \begin{pmatrix} V_p + \rho_p \alpha_p \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Levinson Durbin algoritmus

Vektor na levé straně je vlastně

$$\begin{pmatrix} 1 \\ a_p(1) + \rho_p a_p(p) \\ a_p(2) + \rho_p a_p(p-1) \\ \vdots \\ \rho_p \end{pmatrix} = \begin{pmatrix} 1 \\ a_{p+1}(1) \\ a_{p+1}(2) \\ \vdots \\ a_{p+1}(p+1) \end{pmatrix}$$

Levinson Durbin algoritmus

Odtud dostáváme rekurzivní vzorečky

$$\hat{a}_{p+1}(k) = \hat{a}_p(k) + \hat{\rho}_p \hat{a}_p(p-k+1)$$

$$\hat{a}_{p+1}(p+1) = \hat{\rho}_p$$

$$V_{p+1} = V_p + \hat{\rho}_p \alpha_p$$

$$\hat{\rho}_p = \frac{-\alpha_p}{V_p}$$

$$\alpha_p = r_{xx}(p+1) + \sum_{k=1}^{p} \hat{a}_p(k) r_{xx}(p+1-k)$$

S počátečními podmínkami

$$V_1 = r_{xx}(0) - \frac{r_{xx}(1)^2}{r_{xx}(0)}$$

$$\hat{a}_1(1) = \frac{-r_{xx}(1)}{r_{xx}(0)}$$

Levinson Durbin algoritmus

Jedna iterace rekurze vyžaduje $4p+2$ sčítání a násobení a jedno dělení. Výpočet $\hat{\boldsymbol{a}}_{p+1}$ vyžaduje $2p^2$ operací. Autokorelační metoda vyžaduje naproti tomu počet operací úměrný p^3.

Levinson Durbin algoritmus

Nevýhody autokorelačního algoritmu

- nutnost výpočtu inverze matice
- nelze snadno zvýšit řád modelu, nutno vše počítat znovu

Levinson Durbin algoritmus

- iterativní algoritmus
- v každém kroku je zvýšen řád modelu o 1

Procesy MA

Definice MA procesu

MA - Moving Average

$$x(n) = \sum_{k=0}^{q} b_q(k) e(n-k)$$

Přenos v Z-transformaci

$$B_q(z) = \frac{X(z)}{E(z)} = \sum_{k=0}^{q} b_q(k) z^{-k}$$

Teorémy

Wold - ARMA nebo AR systém s konečným řádem je ekvivalentní systému MA s nekonečným řádem.
Kolmogorov - ARMA nebo MA systém s konečným řádem je ekvivalentní systému AR s nekonečným řádem.

Durbinova metoda

MA systém q-tého řádu lze nahradit AR modelem p-tého řádu, kde $p >> q$

$$B_q(z) \approx \frac{1}{A_p(z)} = \frac{1}{\sum_{k=0}^{q} a_p(k) z^{-k}}$$

Platí rovněž

$$A_p(z) = \sum_{k=0}^{q} a_p(k) z^{-k} \approx \frac{1}{B_q(z)}$$

AR model $\frac{1}{B_q(z)}$ q-tého řádu může být považován za model procesu tvořeného řadou koeficientů $[a_p(0), a_p(1), \ldots, a_p(p)]$ „dlouhého“ AR modelu $A_p(z)$.

Durbinův algoritmus

1. Pomocí autokorelačního nebo Levinson Durbin algoritmu aplikovaného na změřená data $x(n)$ určíme koeficienty AR modelu $a_p(k)$ $p >> q$
2. Získané koeficienty $[a_p(0), a_p(1), ..., a_p(p)]$ považujeme za novou řadu dat, na kterou aplikujeme autokorelační metodu pro získání AR modelu. Koeficienty takto získaného modelu odpovídají koeficientům MA modelu původních dat $b_q(k)$.

Procesy ARMA

Definice ARMA procesu

Kombinace AR a MA

$$\sum_{i=1}^{p} a_p(i)x(n-i) = \sum_{j=1}^{q} b_q(j)e(n-j)$$

Přenos v z-transformaci

$$\frac{B_q(z)}{A_p(z)} = \frac{X(z)}{E(z)}$$

Podle Kolmogorovova teorému platí

$$\frac{B_q(z)}{A_p(z)} = \frac{1}{A_\infty(z)}$$

přibližně platí pro velké l

$$\frac{B_q(z)}{A_p(z)} \approx \frac{1}{A_l(z)}$$

Postup

1. Pomocí YW metody určíme AR model $\dfrac{1}{A_l(z)}$
2. Na základě vztahu $\hat{e} = A_l(z)x$ odhadneme signál na vstupu modelu
3. ARMA model určíme metodou nejmenších čtverců pro systém $A_p(z)x = B_q(z)\hat{e}$

Principal druhé Durbinovy metody

1. Vypočten „dlouhý“ AR model $\dfrac{1}{A_l(z)}$ pomocí Durbinovy metody
2. Počáteční odhad AR části modelu pomocí první Durbinovy metody
3. Iterace
   (a) Dělení $D^j(z) = \dfrac{A_l(z)}{A^{j-1}(z)}$
   (b) Výpočet MA modelu $B^j(z)$ z $D^j(z)$ pomocí Durbinova MA algoritmu
   (c) Pomocí MA modelu $B^j(z)w = x$ určíme hodnoty signálu w
   (d) Odhadneme AR model $A^j(z)$ pomocí YW algoritmu aplikovaného na řadu dat $w(n)$
   (e) Opakujeme dokud se řešení neustálí

Prostředky pro výpočet

MATLAB Simulink

- Signal Processing Toolbox
- DSP Blockset

ARMASA

- volně šiřitelný Toolbox pro Matlab
- lze stáhnout z http://www.dcsc.tudelft.nl/Research/Software/index.html
- umožňuje výpočet jednotlivých typů modelů náhodných signálů (a tedy i výpočet spektra)
- automatická volba řádu modelu