

کاربردها

۱-۹ تغییر فرکانس نمونه برداری

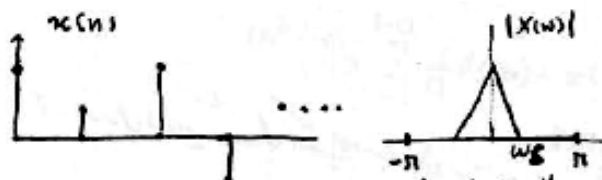
۱- در کاربردهای بسیاری اتفاق می افتد که سیگنالها با پهنای باندها متفاوت با یک پریود ثابت، نمونه گیری می گردند. از نظر محاسباتی خانه های مورد نظر حافظه بسیار مناسب خواهد بود که فرکانس نمونه گیری سیگنالهایی که پهنای باند کمتری دارند کاهش داده است همچنین برای اینکه نیازی به فیلتر ایده آل صفر ناکافی فرکانسی آنالوگ نباشد می توان از سیگنال آنالوگ فرامونه گیری کرده سپس بر آن فیلتر ضد درهم شدن فرکانسی دیجیتالی اعمال شده و سپس فرکانس نمونه گیری به مقدار حداقل خود کاهش داده می شود.

۲- برای بازسازی آنالوگ سیگنال دیجیتالی نیاز به فیلتر بازسازی ایده آل است. برای اینکه بتوان از فیلتر بازسازی آنالوگ ساده تری استفاده کرد می توان از سیگنال دیجیتال فرامونه گیری کرده و سپس آنرا به D/A اعمال کرد.

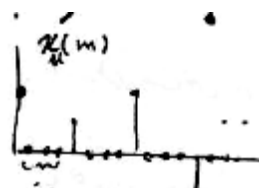
به کاهش فرکانس نمونه گیری $decimation$ و به افزایش فرکانس نمونه گیری $interpolation$ نیز گفته می شود. برای تغییر فرکانس نمونه گیری از F_s به $u/d F_s$ ابتدا فرکانس به $u F_s$ افزایش و سپس به $u/d F_s$ کاهش می یابین ترتیب فقط به یک فیلتر نیاز دارد و اگر معکوس عمل شود و دو فیلتر مورد نیاز می گردد. اگر $u=1$ باشد کاهش نمونه گیری صرف و اگر $d=1$ باشد افزایش نمونه گیری حاصل می گردد

۱-۱-۹ افزایش فرکانس نمونه گیری به نسبت u

۱- سیگنال $x(n)$ و طیف آن در شکل نشان داده شده است.



۲- $u-1$ صفر بین نمونه های $x(n)$ جای داده می شود.

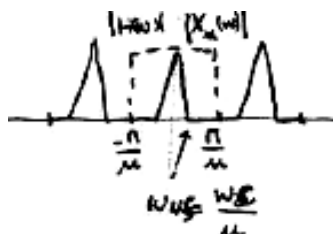


$$x'_u(m) = \begin{cases} x\left(\frac{m}{u}\right) & m = 0, \pm u, \pm 2u \\ 0 & \text{otherwise} \end{cases}$$

۳- با این کار، در میدان فرکانس طیف دیجیتالی فشرده شده و تصاویر سایه ایجاد می شود.

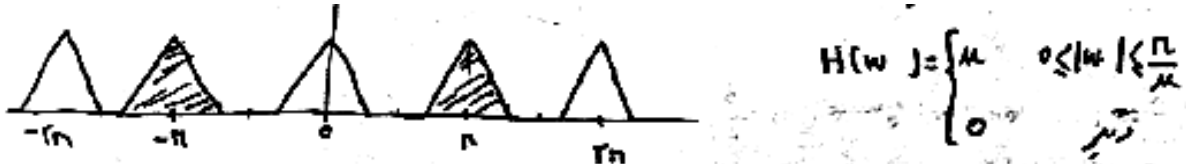
$$X'_u(\omega) = \sum_0^{\infty} x'_u(n) e^{-jn\omega} = \sum_0^{\infty} x\left(\frac{n}{u}\right) e^{-jn\omega} = \sum_0^{\infty} x(m) e^{-jmu\omega} = X(u\omega)$$

۴- در میدان Z تابع Z سیگنال $x(n)$ اینگونه تغییر می کند.



$$X'_u(z) = X(z^u)$$

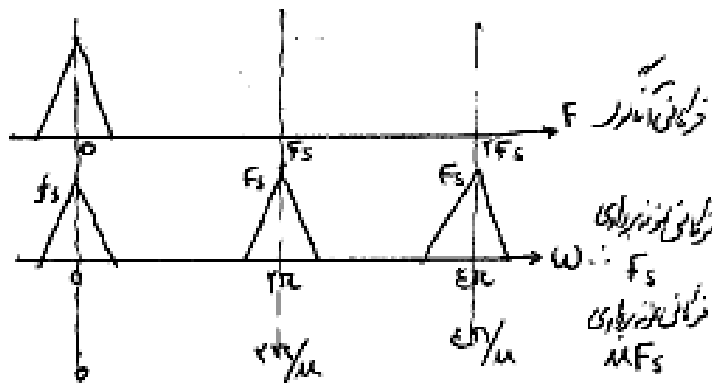
۵- میان یابی (interpolation) حالا باید سایه های $X(\omega)$ در طیف $X'_u(\omega)$ حذف شده و فقط $X(\omega)$ باند پایه باقی بماند. این کار باعث می شود در نقاطی که صفر قرار داده شده مقادیر مناسبی جای گیرد. با افزایش نمونه گیری به نسبت U تعداد $U-1$ طیف سایه شکل می گیرند. شکل به ازای $U=2$ را نشان می دهد. با اعمال فیلتر



طیفهای سایه حذف و اثر آن در میدان زمان میان بانی مناسب در نقاط اضافه شده است. خروجی این عمل فیلتر کردن برابر است با

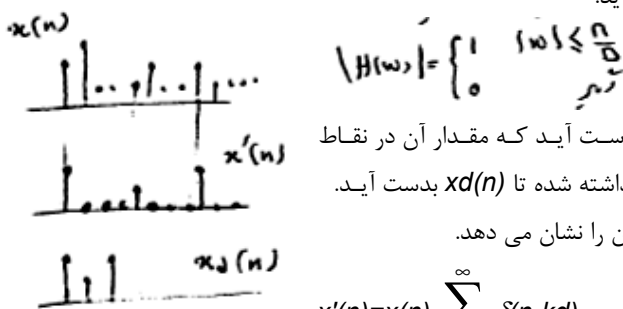
$$x_u(m) = \sum_{k=-\infty}^{\infty} h(m-k)x'_u(k) = \sum_{k=-\infty}^{\infty} h(m-ku)x(k)$$

این تساوی از آن رو برقرار است که $X'_u(k)$ فقط به ازای $k=\pm cu$ دارای مقدار است که C عدد صحیح باشد. و در بقیه نقاط صفر است. شکل ارتباط بین فرکانس آنالوگ، دیجیتال و u بار افزایش یافته را نشان می دهد.



۲-۱-۹ کاهش فرکانس نمونه گیری با نسبت d

۱- با کاهش فرکانس نمونه گیری امکان در هم شدن فرکانسی پدید می آید لذا سیگنال باید از فیلتر ضد درهم شدن فرکانسی عبور داده شده تا سیگنال $x(n)$ با طیف $X(\omega)$ بدست آید.



۲- $x(n)$ در دنباله ای از پالس ضرب می گردد تا $x'(n)$ بدست آید که مقدار آن در نقاط $n \neq kd$ صفر است. در نهایت $d-1$ صفر بین نمونه ها برداشته شده تا $x_d(n)$ بدست آید. شکل مراحل کاهش فرکانس نمونه گیری در میدان زمان را نشان می دهد.

$$x'(n) = x(n) \sum_{k=-\infty}^{\infty} \delta(n-kd)$$

۳- برای بدست آوردن تغییراتی که در طیف اتفاق می افتد، سری فوریه دنباله پالسها را در رابطه جایگزین کرده و در نهایت طیف $x_d(\omega)$ را بدست می آوریم.

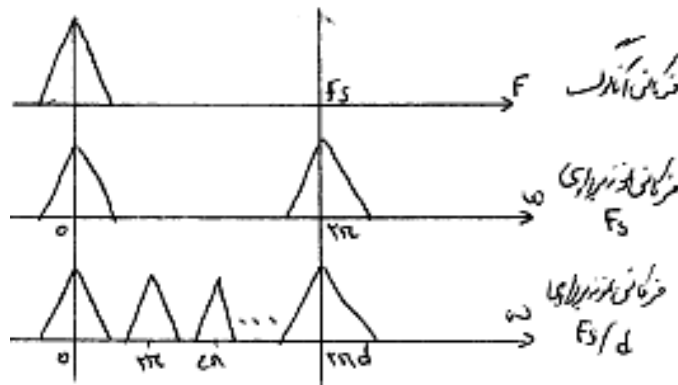
$$x'(n) = x(n) \frac{1}{d} \sum_{k=0}^{d-1} e^{j2\pi kn/d} \Rightarrow x_d(n) = x'(nd) = x(nd) * \frac{1}{d} \sum_{k=0}^{d-1} e^{j2\pi kn} \Rightarrow$$

$$X_d(\omega) = \sum_{n=-\infty}^{\infty} x_d(n) e^{-jn\omega} = \sum_{n=-\infty}^{\infty} x(nd) e^{-jn\omega} \frac{1}{d} \sum_{k=0}^{d-1} e^{j2\pi kn} \Rightarrow X_d(\omega) = \frac{1}{d} \sum_{k=0}^{d-1} X\left(\frac{\omega - 2\pi k}{d}\right)$$

۴- به طریق مشابهی می توان ارتباط بین تبدیل Z دو سیگنال را برقرار کرد.

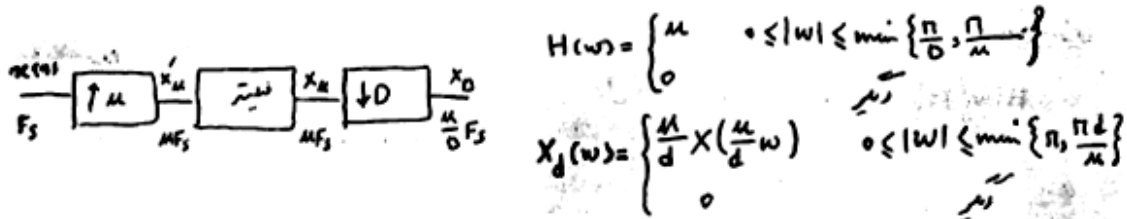
$$X_d(z) = \frac{1}{d} \sum_{k=0}^{d-1} X(e^{j2\pi k/d} z^{1/d})$$

۵- همچنانکه دیده می شود با کاهش فرکانس نمونه گیری طیف باز می شود و همین امر است که می تواند باعث ایجاد در هم شدن فرکانسی گردد.



۳-۱-۹ پیاده سازی فیلتر تغییر فرکانس نمونه گیری

برای تغییر فرکانس نمونه گیری عملیاتی مطابق بلوک دیاگرام شکل باید صورت گیرد تا $X(n)$ معوج نشود.

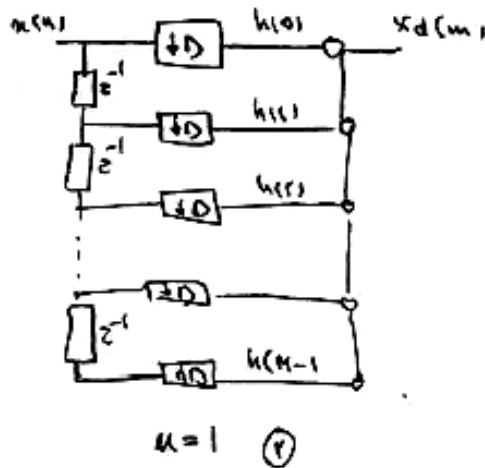
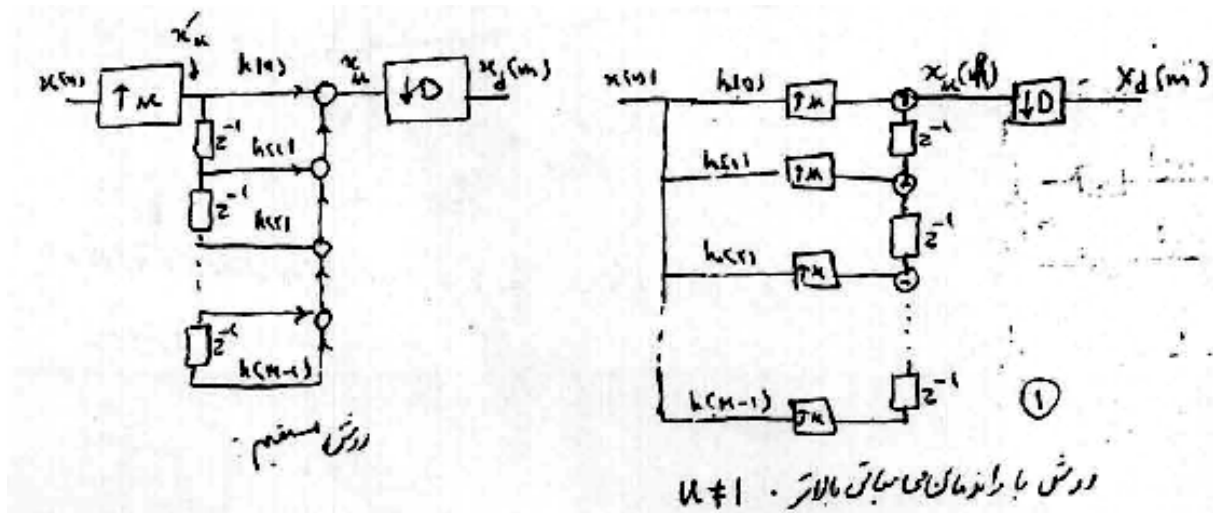


این پاسخ فرکانسی مربوط به سیگنال $x(n)$ وقتی فرکانس نمونه گیری با نسبت u/d F_s تغییر کند می باشد. ارتباط $X_d(\omega)$ با سیگنال آنالوگ مربوطه عبارت است از

$$X_d(\omega) = \frac{1}{d} F_s \sum_{k=-\infty}^{+\infty} X_a\left(\Omega - k \frac{2\pi}{d} \Omega_s\right)$$

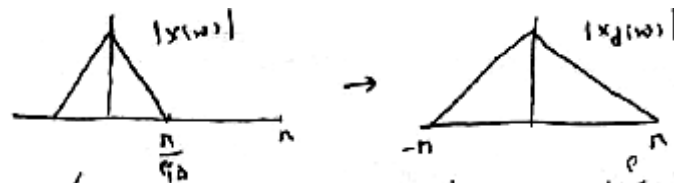
افزایش راندمان محاسباتی: در تبدیل فرکانس نمونه گیری x'_u دارای تعدادی صفر است که نیازی به دخالت آنها در محاسبات نیست. همچنین تمام نقاط x_u نیز در x_d مورد نیاز نیست که محاسبات مربوط به آن نقاط نیز برای بالا بردن راندمان باید کنار گذاشته شود.

بنابراین به جای پیاده کردن مستقیم بلوک دیاگرام شماتیک، تغییراتی در آن داده می شود تا راندمان محاسبات بالا رود. در این جا از فیلتر FIR خطی فاز با پاسخ ضربه $h(n)$ به طول M استفاده می شود.



در این طرح صفرهای اضافه شده به $x(n)$ ناشی از $u > 1$ از محاسبات خارج شده اند ولی خروجی فیلتر شده $x_u(k)$ نقاط بیشتری از نقاط مورد نیاز $x_d(n)$ را دارا است و از این نظر محاسبات اضافی انجام شده است. اگر $u=1$ باشد طرح مقابل طرح راندمانی است. اگر خطی فاز بودن ترها در نظر گرفته شوند برای هر مورد دو محاسبات ۵۰٪ کاهش می یابند. روشهایی با راندمان بالاتر برای این امر وجود دارند که در کتابهای مربوطه قابل دسترس هستند.

مثال: سیگنال $x(n)$ با فرکانس $\omega_s = \pi/3.5$ با پریود نمونه برداری F_s وجود دارد. سیگنال $x_d(m)$ را استخراج کنید که پریود نمونه برداری آن حداقل باشد.



برای این منظور فرکانس نمونه برداری باید در $1/3.5$ ضرب شود.

$$F_s' = \frac{M}{d} F_s \Rightarrow \frac{M}{d} = \frac{1}{3.5} = \frac{2}{7}$$

مقدار $\pi/D = \pi/7$ و $\pi/U = \pi/2$ است پس فیلتری با باند $\omega = \pi/7$ باید طرح شود مشخصات این فیلتر می تواند 0.1db رپیل گذر و 30db تضعیف در باند حذف باشد. از هر فیلتر خطی فازی می توان استفاده کرد که مشخصات فیلتر *kaiser* به این قرار است. دستورات **MATLAB** برای تغییر فرکانس نمونه برداری

$y = \text{decimate}(x, d)$

x بردار $x(n)$ ، d تعداد نقاط تقسیم است. این دستور بطور اتوماتیک از فیلتر چپی شف نوع ۱ درجه ۸ از نوع صفر فاز استفاده می کند. تا درهم شدن فرکانسی اتفاق نیفتد.

$y = \text{interp}(x, u)$

x بردار ورودی u تعداد نقاط اضافه شده این دستور از فیلتر بهینه حداقل مربعات استفاده می کند

$y = \text{resample}(x, \mu, D)$

در این دستور x بردار ورودی و فرکانس نمونه گیری در u/d ضرب می شود. این دستور از فیلتر *kaiser* استفاده می کند.

۴-۱-۹ ساختار چند فازی Polyphase Representation

عملکرد مدل مستقیم در کاهش و افزایش فرکانس نمونه برداری با راندمان نیست. از اینرو مدل دیگری برای محاسبات ارائه می گردد. فرض کنید هدف، کاهش فرکانس نمونه برداری به $d=2$ باشد.

تابع تبدیل زیر را در نظر بگیرید

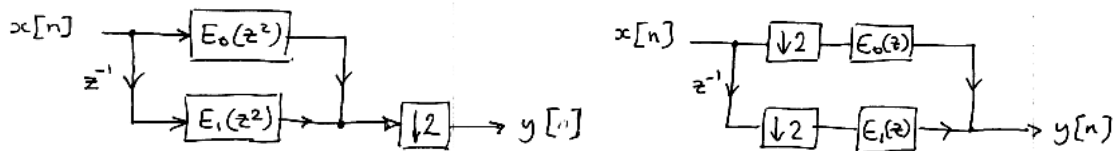
$$H(z) = 1 + h(1)z^{-1} + h(2)z^{-2} + \dots$$

آنرا میتوان بصورت

$$H(z) = \sum_k h(2k)z^{-2k} + z^{-1} \sum_k h(2k+1)z^{-2k} = E_0(z^2) + z^{-1}E_1(z^2)$$

نوشت. در این رابطه جمله اول ورودیهای زوج و جمله دوم ورودیهای فرد را شامل می شود. به این، تجزیه 2-phase تابع تبدیل $H(z)$ می گویند. به این ترتیب بجای مدل مستقیم محاسباتی می توان از مدل با راندمان استفاده کرد. در مدل با راندمان فیلتر کردن در ریت پایین انجام می شود.

$$H(z) = E_0(z^2) + z^{-1}E_1(z^2)$$



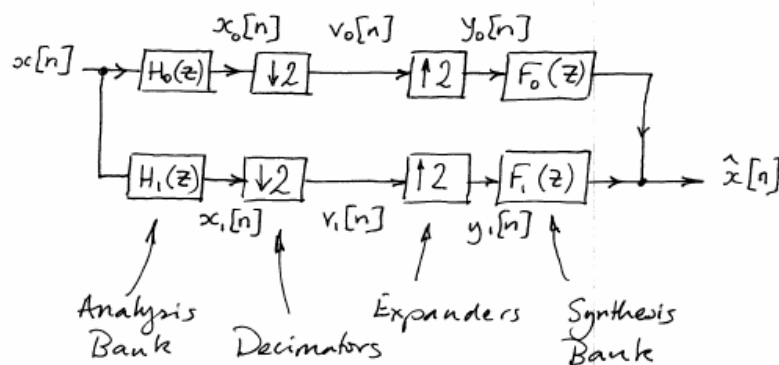
دقت کنید که $E_0(z)$ فقط با ورودیهای زوج و $E_1(z)$ فقط روی ورودیهای فرد عمل می کند.

در حالت کلی که کاهش فرکانس نمونه برداری با نسبت $d=M$ است باید از M -phase استفاده شود که روابط آن را به یکی از ۲ فرم ذیل می توان نوشت.

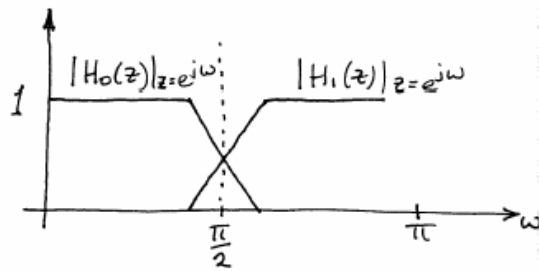
$$\begin{aligned}
 H(z) &= \sum_{k=-\infty}^{\infty} h(kM) z^{-kM} \\
 &+ z^{-1} \sum_{k=-\infty}^{\infty} h(kM+1) z^{-kM} \\
 &\vdots \\
 &+ z^{-(M-1)} \sum_{k=-\infty}^{\infty} h(kM+M-1) z^{-kM} \\
 &\Rightarrow H(z) = \sum_{l=0}^{M-1} z^{-l} E_l(z^M) \\
 E_l(z) &= \sum_{k=-\infty}^{\infty} e_l(k) z^{-k} \\
 e_l(n) &= h(kM+l) \quad 0 \leq l \leq M-1
 \end{aligned}$$

۵-۱-۹ بانک فیلترها

می توان سیگنال را به ۲ کانال مطابق شکل تجزیه و روی هریک با ریت پایین تری عمل کرد.



شکل طیف سیگنال هر کانال را نشان می دهد.



که H_0 و H_1 تصاویر آینه ای یکدیگر حول $\pi/2$ هستند. و به آنها **Quadrature Mirror Filter (QMF)** می گویند
 $H_1(z)$ را از $H_0(z)$ اینگونه می توان ساخت

$$H_1(z) = H_0(-z)$$

$$h_1[n] = h_0[n] \cdot (-1)^n$$

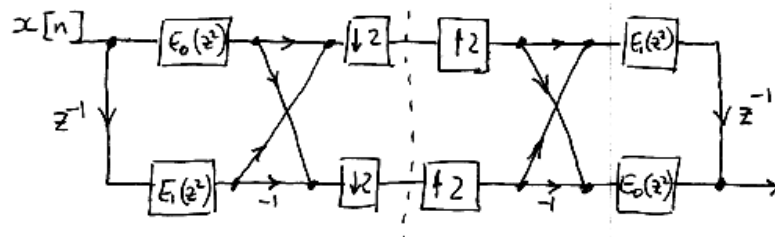
این روش در معرض ۳ خطا است که عبارتند از

درهم شوندگی فرکانس

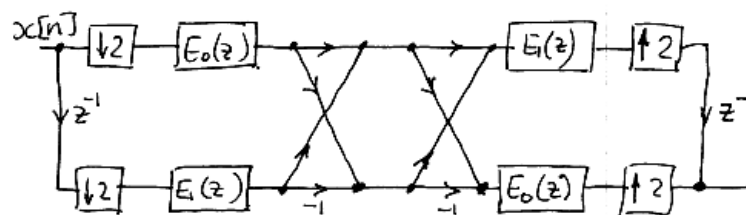
اعوجاج دامنه

اعوجاج فاز

مدل مستقیم



مدل معادل با راندمان

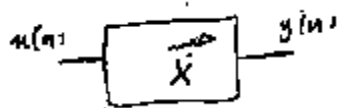


دقت کنید که فیلترها در ریت پایین انجام می گیرند.

۲-۹ مقدمه ای بر شناسایی سیستمها

منظور از شناسایی سیستم تخمین سیستم بر اساس ورودی و خروجی است. در یک کاربرد، سیگنال خروجی از کانال نامشخص دریافت شده، و هدف بدست آوردن تخمین سیگنال ورودی است. در این حالت کانال با سیگنالهای مشخص تحریک و مدل کانال استخراج می گردد، سپس براساس مدل کانال، می توان در حالت واقعی ورودی اصلی به کانال را تخمین زد. به دنبال آشنائی مختصر با شناسایی سیستمها ارائه می شود.

۱-۲-۹ بر اساس تبدیل Z



اگر خروجی سیستم معلوم باشد، براساس اطلاع از ورودی و یا مشخص بودن سیستم، می توان سیستم و یا ورودی مجهول را بدست آورد.

$$Y(z)=X(z)H(z) \Rightarrow X(z)=Y(z)/H(z) \quad \text{یا} \quad H(z)=Y(z)/X(z)$$

عملا در اکثر موارد این روش قابل استفاده نیست، زیرا خروجی به فرم بسته ریاضی $Y(z)$ وجود ندارد.

در عمل وقتی $h(n)$ پاسخ ضربه کانال، معلوم است. معکوس آن را به روش ذیل می توان بدست آورد.

$$H(z)H(z^{-1})=1 \Rightarrow h(n) \otimes h_i(n)=\delta(n)$$

با فرض $h(n)=0$ برای $n < 0$ می نویسیم $h_i(0)=h(0)$ و بقیه تابع ضربه سیستم معکوس را با رابطه

$$h_i(n) = \sum_{k=1}^n \frac{h(n)h_i(n-k)}{h(0)} \quad n \geq 1$$

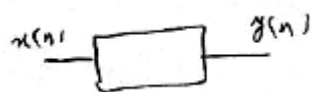
بدست می آوریم. در این روش فرض می شود که سیستم و معکوس آن علی و پایدار است در این روش اگر کانال حداقل فاز نباشد (صفر خارج دایره واحد داشته باشد) معکوس آن پایدار نیست. ولی از آنجائیکه برای سیستم حداقل فاز رابطه

$$|H_{\max \text{phase}}| = |H_{\text{allpass}}| \cdot |H_{\min \text{phase}}|$$

صادق است می توان دامنه سیگنال ورودی را استخراج کرد.

۲-۲-۹ بر اساس همبستگی متقابل

در این روش سیستم با ورودی تحریک می گردد که تمام مولفه های فرکانسی باند مورد نظر را داشته باشد. اگر تابع همبستگی ورودی $r_{xx}(m)$ باشد. تابع همبستگی متقابل آن.



$$r_{yx}(m) = \sum_{k=0}^{\infty} h(k)r_{xx}(m-k) = h(n) \otimes r_{xx}(m)$$

می باشد که کانولوشن تابع ضربه سیستم با $r_{xx}(m)$ است. لذا می توان نوشت

$$S_{yx}(\omega) = H(\omega)|X(\omega)|^2 \Rightarrow H(\omega) = \frac{S_{yx}(\omega)}{|X(\omega)|^2}$$

این روش یکی از روشهای موثر برای شناسایی سیستمهای مجهول است. اگر $x(n)$ به نحوی انتخاب شود که مثلا تابع ضربه باشد آنوقت می تواند $S_{yx}(\omega)=H(\omega)$ باشد.

۳-۲-۹ جداسازی همومورفیک Homomorphic Deconvolution

خروجی سیستمی را در نظر بگیرید.

$$Y(z)=X(z)H(z)$$

اگر از خروجی \ln بگیریم بدست می آید

$$\ln(Y(z))=\ln(X(z))+\ln(H(z))=C_y(z)=C_x(z)+C_h(z)$$

اگر از $C_x(z)$ و بقیه عکس تبدیل Z گرفته شود دنباله ای بدست می آید که به آن سپترم Complex Cepstrum می گویند.

$$C_x(n) = \frac{1}{2\pi j} \int_{\Gamma} \ln x(z) z^{n-1} dz, \quad C_x(\tau) = \ln x(e) = \sum_{-\infty}^{+\infty} C_x(n) z^{-n}$$

سپستریم $x(n)$ وجود دارد اگر دایره واحد جزء ناحیه همگرایی $X(z)$ باشد به این ترتیب می توان نوشت که

$$C_x(\omega) = \ln x(\omega) = \sum_{-\infty}^{+\infty} C_x(n) e^{-j\omega n}$$

$$C_x(n) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \ln[x(\omega)] e^{j\omega n} d\omega = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \ln(x(\omega)) e^{j\omega n} d\omega +$$

$$\frac{1}{2\pi} \int_{-\pi}^{+\pi} j\theta(\omega) e^{j\omega n} d\omega \Rightarrow C_x(n) = C_{mx}(n) + jC_{\theta x}(n)$$

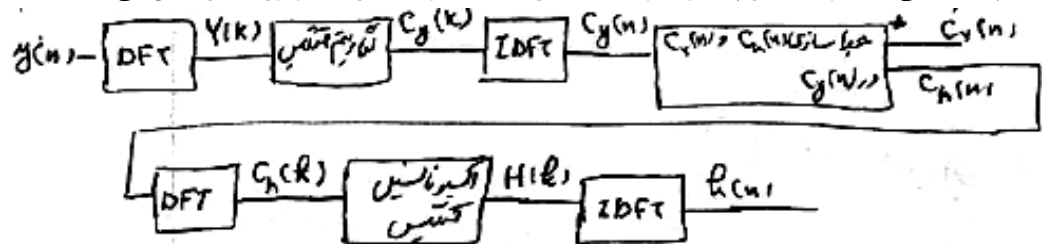
حال اگر برگردیم به تابع تبدیل سیستم مجهول داشتیم که

$$C_y(z) = C_x(z) + C_h(z)$$

که متناظرا

$$c_y(n) = c_x(n) + c_h(n)$$

بدست می آید. به این ترتیب کانولوشن در سیگنال در میدان $y(n) = x(n) \otimes h(n)$ به مجموع دو سیگنال در میدان سپستریم بدل می شود که $c_y(n) = c_x(n) + c_h(n)$ است. حال اگر $x(n)$ و $h(n)$ سپستریمهایی قابل تفکیک داشته باشند امکان دست یابی به $H(z)$ یا $h(n)$ سیستم بدست می آید. بلوک دیاگرام شکل نحوه مبادله سازی عملیات و جداسازی همومورفیک را نشان می دهد.



این روش در کاربردهایی مانند پردازش صورت زلزله کاربرد دارد که محل $C_x(n)$ و $C_h(n)$ در سپستریم Y قابل تفکیک هستند.

۴-۲-۹ روش های پارامتریک شناسایی سیستمها:

حداقل مربعات *Least Squares*

فرض کنید سیگنال u به سیستمی اعمال و خروجی y بدست آمده باشد هدف تعیین پارامترهای سیستمی با تابع تبدیل

$$G(z) = \frac{\sum_{i=0}^p b_i z^{-i}}{\sum_{j=0}^q a_j z^{-j}}$$

است. برای سیستم ابتدا می بایستی درجات p, q را انتخاب نمود.

معادله دیفرنس این سیستم برابر است با

$$y(n) + a_1 y(n-1) + \dots + a_p y(n-p) = b_0 u(n) + \dots + b_q u(n-q)$$

معادله دیفرنس این سیستم را می توان به صورت برداری اینگونه نوشت.

$$y(n) = \Phi'(n) \theta$$

$$\theta = [a_1, \dots, a_p, b_0, \dots, b_q]'$$

$$\Phi(n) = [-y(n-1), \dots, -y(n-p), u(n), \dots, u(n-q)]'$$

بردار θ بردار پارامترهای مجهول سیستم است و بردار Φ ، مقادیر اندازه گیری شده از ورودی u و خروجی y است. اگر $\theta = \theta^*$ مقدار واقعی θ

باشد بدیهی است که

$$y(n) - \varphi'(n)\theta = y(n) - \varphi'(n)\theta_t = 0$$

می گردد. ولی اگر θ تخمین از θ_t باشد.

$$y(n) - \varphi'(n)\theta = e(n, \theta)$$

می گردد که آلا ما صفر نیست. اگر طول مقادیر اندازه گیری شده N باشد، N معادله بدست می آید.

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} 0 & x(0) & 0 \\ -y(0) & x(0) & x(0) \\ \vdots & \vdots & \vdots \\ -y(N-1) & x(N-1) & \vdots \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_1 \end{bmatrix} = \begin{bmatrix} e(0, \theta) \\ e(1, \theta) \\ \vdots \\ e(N-1, \theta) \end{bmatrix} \Rightarrow Y - \Phi\theta = E(\theta)$$

برای بدست آوردن تخمین θ ، θ به نحوی محاسبه می گردد که مجموع مربعات خطاها حداقل شود. برای این منظور می نویسیم.

$$J(\theta) = \sum_{k=0}^{N-1} e^2(k, \theta) = E'(\theta)E(\theta)$$

می توان برای خطاها وزن نیز در نظر گرفت که در این صورت $Weighted LS$ خوانده می شود.

$$J(\theta) = \sum_{k=0}^{N-1} w(k) e^2(k, \theta) = E'WE$$

برای اینکه $J(\theta)$ حداقل گردد مشتق آن نسبت به θ باید صفر شود.

$$J(\theta) = (Y - \Phi\theta)'(Y - \Phi\theta) = Y'Y - \theta'\Phi'Y - Y'\Phi\theta + \theta'\Phi'\Phi\theta \Rightarrow$$

$$\frac{dJ(\theta)}{d\theta} = 0 \Rightarrow -Y'\Phi + \theta'\Phi'\Phi = 0 \Rightarrow \theta_{LS} = (\Phi'\Phi)^{-1}\Phi'Y$$

به این تخمین، تخمین حداقل مربعات برای پارامترهای θ گفته می شود.

مثال: فرض کنید به سیستمی ورودی پله $u=[1,1,1,1]'$ اعمال و خروجی $y=[1,1.5,1.75,1.875]'$ بدست آمده باشد، پارامترهای

سیستمی با درجات $p=1$ و $q=0$ که معادله دیفرانس آن $y(n)=ay(n-1)+bu(n)$ است را برای تخمین بزنید.

$$\begin{aligned} y(0) &= ax(0) + bx(1) + e(0, \theta) = 1 \\ y(1) &= ax(1) + bx(1) + e(1, \theta) = 1.5 \\ y(2) &= ax(2) + bx(1) + e(2, \theta) = 1.75 \\ y(3) &= ax(3) + bx(1) + e(3, \theta) = 1.875 \end{aligned} \Rightarrow \begin{bmatrix} 1 \\ 1.5 \\ 1.75 \\ 1.875 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1.5 & 1 \\ 1.75 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} e(0, \theta) \\ e(1, \theta) \\ e(2, \theta) \\ e(3, \theta) \end{bmatrix}$$

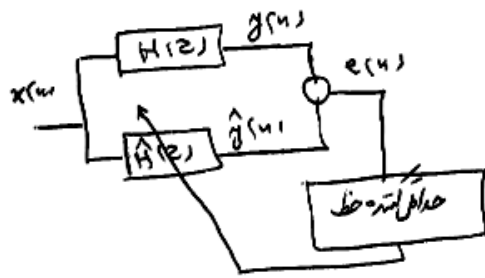
$$\Rightarrow \theta_{LS} = (\Phi'\Phi)^{-1}\Phi'Y = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1.5 & 1 \\ 1.75 & 1 \end{bmatrix}' \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1.5 & 1 \\ 1.75 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1.5 \\ 1.75 \\ 1.875 \end{bmatrix}$$

$$\theta_{LS} = \begin{pmatrix} 7/12 & 5/12 \\ 5/12 & 1/12 \end{pmatrix}^{-1} \begin{bmatrix} 7/12 & 5/12 \\ 5/12 & 1/12 \end{bmatrix} \begin{bmatrix} 1 \\ 1.5 \\ 1.75 \\ 1.875 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

این ضرایب دقیقاً مربوط به سیستم $y(n)=0.5y(n-1)+u(n)$ است که سیگنال خروجی فوق را تولید کرده بود.

شناسایی مدل FIR بر اساس LS

در این روش مدل سیستم



$$H(z) = \sum_{k=0}^{N-1} b_k z^{-k}$$

فرض می شود. ورودی $x(n)$ و خروجی $y(n)$ وجود دارد، هدف بدست آوردن تخمینی برای $H(z)$ است بنابراین

$$e(n) = y(n) - \hat{y}(n) = y(n) - \sum_{k=0}^{N-1} b_k x(n-k)$$

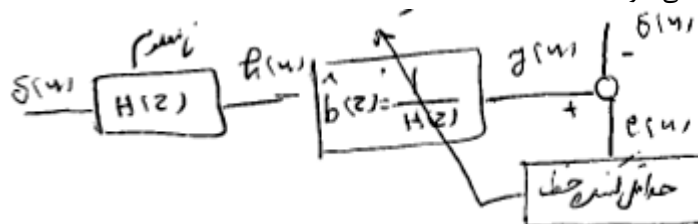
نوشته و مجموع مربعات خطا حداقل می گردد. براساس تعداد مشاهدات از خروجی و ورودی، و M طول فیلتر، M معادله بدست می آید که هر یک N پارامتر دارند لذا نوشته می شود. $E = Y - \Phi' \theta$ که در آن E بردار M در 1 ، Y بردار M در 1 ، Φ' ماتریس $M \times N$ و $\theta = [b_0, \dots, b_{N-1}]'$ بردار پارامترهاست. با حداقل سازی $E'E$ مقدار

$$\theta = (\Phi \Phi')^{-1} \Phi Y = R_{xx}^{-1} R_{yx}$$

بدست می آید.

در این رابطه R_{xx} ماتریس خود همبستگی X و R_{yx} بردار همبستگی متقابل Y و X است.

شناسایی مدل تمام قطب بر اساس معکوس سیستم

برای بدست آوردن مدل تمام قطب، براساس پاسخ ضربه، از مدل شکل می توان استفاده کرد. که در آن $h(n)$ معلوم است و ضرایب مخرج $H(z) = 1/b(z)$ تخمین زده می شود

روش LPC

این روش ضرایب α مربوط به تابع تبدیل

$$H(z) = \frac{1}{1 + \sum_{i=0}^{N-1} a_i z^{-i}}$$

را از روی پاسخ ضربه این سیستم تعیین می کند. خروجی پاسخ ضربه سیستم برای $n > 0$ برابر است با

$$h(n) = \sum_{k=1}^{N-1} a_k h(n-k)$$

تابع خطا نوشته شده و به روش LS حل می گردد.

$$h(n) - \sum_{k=1}^{N-1} a_k h(n-k) = e(n, a) = h(n) - \Phi'(n) a \Rightarrow a_{LS} = (\Phi' \Phi)^{-1} \Phi' h$$

دیده می شود که $\Phi' h$ و بردارهایی ساخته شده از تابع خودهمبستگی h هستند

$$R_{hh} \Phi' \Phi = \begin{bmatrix} r_{hh}(0) & r_{hh}(1) & \dots & r_{hh}(p-1) \\ r_{hh}(1) & r_{hh}(0) & \dots & r_{hh}(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{hh}(p-1) & r_{hh}(p-2) & \dots & r_{hh}(0) \end{bmatrix}, \quad \Phi' h = \begin{bmatrix} r_{hh}(1) \\ \vdots \\ r_{hh}(p) \end{bmatrix}$$

به این ترتیب رابطه را می توان بصورت

$$R_{hh} a = r_{hh}$$

به این معادله، معادله نرمال یا معادله *Yule Walker* می گویند. حل این معادله به P^3 محاسبه نیاز دارد.

روش محاسبه خاصی با توجه به تقارن در R_{hh} وجود دارد که به نام *Levinson-Durbin* شناخته می شود. تعداد محاسبات این روش متناسب با P^2 است لذا در اغلب الگوریتمها مورد استفاده قرار می گیرد.

MATLAB این الگوریتم را با دستور

$$a = \text{levinson}(r, p)$$

اجرا می کند. در این دستور P ابعاد ماتریس R_{hh} و r تابع خود همبستگی سیگنال h است.

دستور **MATLAB** برای اجرای دستور *LPC* اینگونه است.

$$a = \text{lpc}(h, p)$$

در این دستور h بخشی از بردار پاسخ ضربه مربوط به $p, n > 0$ درجه مخرج و α ضرایب مخرج است.

الگوریتم پرونی Prony

این الگوریتم بر اساس پاسخ ضربه سیستم تابع تبدیل

$$H(z) = \frac{\sum_{i=1}^p b_i z^{-i}}{-\sum_{j=1}^p a_j z^{-j}}$$

برای سیستم تقریب می زند. پاسخ این سیستم را به دو قسمت می توان تقسیم کرد.

$$n \leq q \Rightarrow h(n) = \sum_{i=1}^p a_i h(n-i) + b_n$$

$$n > q \Rightarrow h(n) = \sum_{i=1}^p a_i h(n-i)$$

برای معادله دوم تابع خطا تشکیل، مشتق مجموع توان دوم آنها مطابق روش *LS* صفر گردد.

$$\begin{aligned} J(a) &= \sum_{n=q+1}^{\infty} \left(h(n) - \sum_{j=1}^p a_j h(n-j) \right)^2 \\ \Rightarrow \frac{dJ(a)}{da} &= 0 = \sum_{n=q+1}^{\infty} \left(\sum_{\ell=1}^p h(n-\ell) \right) \left(\sum_{j=1}^p (h(n) - a_j h(n-j)) \right) = 0 \\ &= \sum_{\ell=1}^p \sum_{n=q+1}^{\infty} h(n) h(n-\ell) = \sum_{\ell=1}^p \sum_{j=1}^p \sum_{n=q+1}^{\infty} h(n-\ell) h(n-j) a_j \\ &= \sum_{i=1}^p a_i r_{hh}(i, i) = r_{hh}(i, 0) \quad i=1, 2, \dots, p \quad h(n)=0 \quad n < 0 \end{aligned}$$

$$\begin{bmatrix} r_{hh}(1,p) & r_{hh}(1,p) \\ r_{hh}(p,p) & r_{hh}(p,p) \end{bmatrix} \begin{bmatrix} a_1 \\ a_p \end{bmatrix} = \begin{bmatrix} r_{hh}(1,0) \\ r_{hh}(p,0) \end{bmatrix} = R_{hh} a = r_{hh}$$

$$\Rightarrow a = R_{hh}^{-1} r_{hh}$$

محاسبه اخیر را دستور *levinson* می تواند انجام دهد. و کل محاسبات برای تخمین α را دستور *lpc* انجام می دهد که h بخشی از پاسخ ضربه مربوط به $n > q$ است. برای تعیین ضرایب b_i تعداد $q+1$ معادله نوشته می شود تا ضرایب معین گردند.

$$b_n = h(n) - \sum_{j=1}^p a_j h(n-j) \quad n > q$$

$$\begin{bmatrix} b_0 \\ b_q \end{bmatrix} = \begin{bmatrix} h(0) \\ h(q) \end{bmatrix} - \begin{bmatrix} h(-1) & \dots & h(-p) \\ h(q-1) & \dots & h(q-p) \end{bmatrix} \begin{bmatrix} a_1 \\ a_p \end{bmatrix} \quad \therefore h(n) \text{ for } n < 0$$

به این ترتیب در این روش پارامترهای a بر اساس *LS* و پارامترهای b بر اساس محاسبه مستقیم تعیین می گردند. این دستور *MATLAB* الگوریتم فوق را اجرا می کنند.

$$[b,a] = \text{prony}(h,nb,na)$$

در این دستور تمامی پاسخ ضربه $na=p, nb=q$ است و در نهایت b, α ضرایب تابع تبدیل بدست می آید.

الگوریتم شانک *Shank*

الگوریتم شانک مانند پرونی است با این تفاوت که روش محاسبه b ها نیز بر اساس *LS* می گردد.

برای این منظور $H(z)$ به دو قسمت تمام قطب و تمام صفر مطابق شکل تقسیم می شود.

$$\delta(n) \rightarrow \left[\frac{1}{-\sum a_j z^{-j}} \right] \rightarrow v(n) \rightarrow \left[\sum b_i z^{-i} \right] \rightarrow h(n)$$

بر اساس مقادیر تخمین α مقدار $v(n)$ تعیین می شود.

$$v(n) = \sum_{j=0}^q a_j v(n-j) + \delta(n) \quad n > q$$

سپس رابطه ای بین $h(n), v(n)$ برقرار می گردد.

$$h(n) = \sum_{j=0}^q b_j v(n-j)$$

در نهایت تابع خطا معرفی و مشتق مجموع توان دوم آنها مطابق روش *LS* صفر می گردد.

$$\mathcal{E} = \sum_{n=-\infty}^{\infty} \left(h(n) - \sum_{j=0}^q b_j v(n-j) \right)^2$$

$$\frac{d\mathcal{E}}{db} = 0 = \sum_{n=-\infty}^{\infty} -2 \sum_{j=0}^q v(n-j) \left(h(n) - \sum_{l=0}^q b_l v(n-l) \right) = 0$$

$$\sum_{j=0}^q \sum_{n=-\infty}^{\infty} v(n-j) h(n) = \sum_{j=0}^q \sum_{l=0}^q \sum_{n=-\infty}^{\infty} b_l v(n-j) v(n-l)$$

$$\sum_{j=0}^q r_{hv}(j) = \sum_{j=0}^q \sum_{l=0}^q b_l r_{vv}(j, l)$$

$$\Rightarrow \begin{bmatrix} r_{vv}(0) & r_{vv}(1) \\ r_{vv}(2) & r_{vv}(3) \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} r_{uv}(0) \\ r_{uv}(1) \end{bmatrix} \Rightarrow R_{vv} b = r_{uv} \Rightarrow b = R_{vv}^{-1} r_{uv}$$

نکات: اگر در الگوریتم پرونی $nb=0$ منظور شود. مدل تمام قطب برای سیستم محاسبه می گردد.

مدل تمام صفر FIR

در الگوریتم نک ورودی فیلتر FIR، سیگنال $v(n)$ و خروجی $h(n)$ بود. در آن روابط اگر بجای $v(n)$ تابع ضربه $\delta(n)$ و خروجی $h(n)$ پاسخ ضربه باشد مدل تمام صفر FIR برای سیستم بدست می آید.

$$\delta(n) \rightarrow \left[\sum b_j z^{-j} \right] \rightarrow h(n)$$

این کار را برای هر ورودی مانند $x(n)$ و خروجی $y(n)$ نیز می توان انجام داد. در این حال تخمین b برابر است با

$$b = R_{xx}^{-1} r_{yx}$$

مثال: پاسخ ضربه سیستمی داده شده است با دستور prony با $na=nb=4$. تابع تبدیلی برای آن تخمین بزنید.

$$h = [0.0048, 0.0007, 0.0104, 0.1279, 0.2549, 0.2748, 0.1935, 0.1238, 0.0694, \\ -0.0085, -0.0407, -0.0484, -0.0329, -0.0187, -0.0013, 0.1, 0.016, 0.0121, \\ 0.0071, 0.0017, -0.0052, -0.0020, -0.0039, -0.0024, -0.0010, 0.0004]$$

$$[bb, aa] = \text{prony}(h, \epsilon, \epsilon)$$

$$bb = [0.0048, 0.0193, 0.0219, 0.0193, 0.0048] \quad aa = [1, -1.2329z^{-1}, 1.2140, -1.0247, 0.1874] \Rightarrow H(z) = \frac{1.0(z^4 + 1.93z^3 + 2.19z^2 + 1.93z + 0.48)}{1 - 1.2329z^{-1} + 1.2140z^{-2} - 1.0247z^{-3} + 0.1874z^{-4}}$$

تذکر: $H(z)$ بدست آمده دقیقاً همان تابع تبدیلی است که h آن طبق دستورات ذیل ساخته شده بود

$$[b, a] = \text{butter}(\epsilon, 0.2), R = \text{filter}(b, a, \text{zeros}(1, 100));$$

که h پاسخ ضربه فیلتر است.

نکته دیگر اینکه در این مثال h آلوده به نویز نیست و دقیقاً پاسخ ضربه سیستم بوده و دیگر درجات فیلتر نیز دقیق مانند فیلتر اصلی انتخاب شده بودند. وقتی سیگنال ورودی دستور نویزی باشد و درجات نیز نامشخص باشند با سعی و خطا می توان به جواب نزدیک شد. مثال: سیستم ذیل و پاسخ ضربه آن را در نظر بگیرید. از پاسخ ضربه که نویز به آن اضافه شده است سیستم را مجهول فرض کرده، شناسایی کنید.

$$H(z) = \frac{1 - z^{-1}}{1 + z^{-1} + 0.2z^{-2}} \Rightarrow h(n) = -h(n-1) + 0.5h(n-2) + \delta(n) - 0.2\delta(n-1) \\ \Rightarrow h(n) = \{1, -3, 2, 0, -1, -0.25, \dots\} \\ -h(n) = -h(n-1) + 0.5h(n-2) \Rightarrow \{1, 2, -1.1, 1.1, -0.7, -0.25, \dots\} \\ H(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} \Rightarrow \text{Reverse} \quad \begin{aligned} h(0) &= b_0 = 1.1 \\ h(1) &= -1.2a_1 + b_1 = -2.1 \end{aligned}$$

$$Y = \begin{bmatrix} 1.1 \\ -0.5 \\ -0.25 \end{bmatrix} = \begin{bmatrix} 1.1 & -1.2 \\ -1.1 & 1.1 \\ 0.5 & -1.1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \Leftrightarrow \begin{cases} R(1) = +1.1a_1 - 1.2a_2 = 1.1 \\ R(2) = -1.1a_1 + 1.1a_2 = -0.5 \\ R(3) = 0.5a_1 - 1.1a_2 = -0.25 \end{cases}$$

$$\Rightarrow Y = \Phi \theta \Rightarrow E = Y - \Phi \hat{\theta} \Rightarrow E'E = (Y - \Phi \hat{\theta})'(Y - \Phi \hat{\theta}) = Y'Y - Y'\Phi \hat{\theta} - \hat{\theta}'\Phi'Y + \hat{\theta}'\Phi'\Phi \hat{\theta}$$

$$\frac{dE'E}{d\theta} = 0 \Rightarrow -Y'\Phi - \Phi'Y + 2\Phi'\Phi \hat{\theta} = 0 \Rightarrow \hat{\theta} = (\Phi'\Phi)^{-1}\Phi'Y$$

$$\Rightarrow \hat{\theta} = \left(\begin{bmatrix} 1.1 & -1.1 & 0.5 \\ -1.2 & 1.1 & -1.1 \end{bmatrix} \begin{bmatrix} 1.1 & -1.2 \\ -1.1 & 1.1 \\ 0.5 & -1.1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1.1 & -1.1 & 0.5 \\ -1.2 & 1.1 & -1.1 \end{bmatrix} \begin{bmatrix} 1.1 \\ -0.5 \\ -0.25 \end{bmatrix}$$

$$= \begin{bmatrix} 1.71 & -1.71 \\ -1.71 & 1.71 \end{bmatrix}^{-1} \begin{bmatrix} 1.1 & -1.2 \\ -1.1 & 1.1 \end{bmatrix} \begin{bmatrix} 1.1 \\ -0.5 \\ -0.25 \end{bmatrix} = \frac{1}{1.71} \begin{bmatrix} 1.1 & -1.2 \\ -1.1 & 1.1 \end{bmatrix} \begin{bmatrix} 1.1 \\ -0.5 \\ -0.25 \end{bmatrix} = \begin{bmatrix} 1.05 \\ 0.50 \end{bmatrix} = \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \end{bmatrix}$$

$$\text{نسبت اولی} \Rightarrow \hat{b}_2 = 1.2, -1.2\hat{a}_1 + \hat{b}_1 = -1.1 \Rightarrow \hat{b}_1 = -1.59$$

$$\Rightarrow H(z) = \frac{1.2 - 1.59z^{-1}}{1 + 1.1z^{-1} + 0.501z^{-2}}$$

تخمین b بر اساس الگوریتم شانک

$$\hat{b}(n) = \frac{\hat{a}(n)}{\sum \hat{a}_k \varepsilon_k}, \quad \hat{a}(n) = \sum b_j \varepsilon_j = h(n)$$

$$\Rightarrow \hat{a}(n) = -1.1\hat{a}(n-1) + 0.501(\hat{a}(n-2) + \hat{b}(n))$$

$$\Rightarrow \hat{a}(0) = \{1, -1.1, 0.505, -0.005, -0.259, \dots\}$$

$$\Rightarrow h(n) = b_0 \hat{a}(n) + b_1 \hat{a}(n-1) \Rightarrow h(0) = 1.2 = b_0$$

$$h(1) = -1.1b_0 + b_1 = -1.1$$

$$h(2) = 0.505b_0 - 1.1b_1 = 1.1$$

$$h(3) = -0.005b_0 + 0.505b_1 = -0.5$$

$$h(4) = -0.259b_0 - 0.005b_1 = -0.25$$

$$\Rightarrow Y = \begin{bmatrix} 1.2 \\ -1.1 \\ 1.1 \\ -0.5 \\ -0.25 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1.1 & 1 \\ 0.505 & -1.1 \\ 0.005 & 0.505 \\ 0.259 & -0.005 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

$$\Rightarrow \hat{\theta} = \begin{bmatrix} \hat{b}_0 \\ \hat{b}_1 \end{bmatrix} = (\Phi'\Phi)^{-1}\Phi'Y = \begin{bmatrix} 1.222 & -1.01 \\ -1.01 & 1.222 \end{bmatrix}^{-1} \begin{bmatrix} 0.00847 \\ -0.9473 \end{bmatrix} = \frac{1}{1.947} \begin{bmatrix} 1.222 & 1.01 \\ 1.01 & 1.222 \end{bmatrix} \begin{bmatrix} 0.00847 \\ -0.9473 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \hat{b}_0 \\ \hat{b}_1 \end{bmatrix} = \begin{bmatrix} 1.2 \\ -1.185 \end{bmatrix}$$

۱. معادله دیفرنس

$$y(n) = a_1 y(n-1) + \dots + a_p y(n-p) + b_1 u(n-k) + \dots + b_q u(n-q+1) + e(n) + c_1 e(n-1) + \dots + c_m e(n-m)$$

را در نظر بگیرید. این معادله را بصورت

$$A(q) y(t) = B(q) u(t-k) + C(q) e(t)$$

می توان نوشت که $y(n)$ خروجی، $u(n)$ ورودی قابل اندازه گیری و $e(n)$ نویز گوسی با میانگین صفر است.

دستور

$$m = \text{armax}(\text{data}, 'na', p, 'nb', q, 'nc', m, 'nk', k)$$

ضرایب a, b و c را تخمین می زند در این دستور data با دستور

$$\text{data} = \text{iddata}(y, u, e, Ts)$$

تولید می شود که y خروجی ، u ورودی ، e دنباله سیگنال نویز و Ts فرکانس نمونه گیری است. خروجی m حاوی ضرایب تخمین a, b و c و Ts است.

$$m = \text{idpoly}(A, B, C, D, f, \text{noisevariance}, Ts)$$

$$A(z)g(t) = \frac{B(z)}{F(z)} u(t-k) + \frac{C(z)}{D(z)} e(t)$$

در مدل گسسته

$$A(z)g(n) = \frac{B(z)}{F(z)} u(n-k) + \frac{C(z)}{D(z)} e(n)$$

A, B, C, D با ۱ شروع می شوند و B با k تا صفر که نشانه تاخیر k است شروع می گردد.

۳-۹ تخمین طیف Spectral Estimation

در تخمین طیف ورودی قابل اندازه گیری نیست و فقط در باره مشخصات آماری آن مثل ایستایی، میانگین و واریانس و تابع توزیع فرضیهایی اعمال می شود.

مسائل دنیای واقعی تخمین طیف عبارتند از:

- تخمین طیف از سیگنال آلوده به نویز $E[|X(\omega)|^2]$
- تخمین طیف توانی فرایندهای تصادفی $rx(k) = E[x(n)x^*(n+k)] \Rightarrow Px(\omega) = DTFT(rxx(k))$

بدست آوردن طیف سیگنال معین، همان بررسی طیف سیگنال است که با DFT صورت می گرفت بدلیل اعمال پنجره بر سیگنال، طول پنجره رزولوشن طیف و نوع پنجره برآمدگیهای کناری را تعیین می کند که عوامل موثر در بررسی طیف سیگنال می باشند. وقتی سیگنال مورد نظر $x(n)$ تصادفی یا مخلوط سیگنال معین و تصادفی (نویز) باشد، دیگر سیگنال در گروه سیگنالهای انرژی محدود نمی تواند قرار گیرد و برای آن تبدیل فوری وجود ندارد. به همین ترتیب برای سیگنال پریودیک، پریود مشخصی نمی توان متصور شد، لذا سیگنال DFS ندارد. اگرچه ممکن است توان متوسط سیگنال محدود باشد. تخمین طیف مستقیماً از طریق فوریه سیگنال امکان پذیر نیست باید راهی برای تخمین طیف پیدا کرد..

۱-۳-۹ روش خود همبستگی: پریودوگرام

پریودوگرام: سیگنالهای نویزی ایستا دارای توان محدود هستند. برای این سیگنالها می توان تابع خودهمبستگی را نوشت و سپس اطلاعات سیگنال را از طیف تابع همبستگی خودی بدست آورد.

$$r_{xx}(\tau) = E(x(\tau)x^*(\tau+\tau))$$

از تئوری *Wiener-Khinchine*، طیف فوریه تابع همبستگی خودی است.

$$P_{xx}(\omega) = \int_{-\infty}^{\infty} r_{xx}(\tau) e^{-j\omega\tau} d\tau$$

اگر فرایند ارگودیک فرض شود از یک تابع نمونه می توان همبستگی خودی و طیف آن را بدست آورد.

تخمین تابع همبستگی

در صورتیکه تابع خود همبستگی به صورت ذیل تعریف شود.

$$r'_{xx}(m) = \frac{1}{N-|m|-1} \sum_{n=0}^{N-|m|-1} x^*(n)x(n+m)$$

می توان نشان داد اگر $E(r'_{xx}(m)) = r_{xx}(m)$ و $\lim_{N \rightarrow \infty} \text{var}(r'_{xx}(m)) = 0$ باشد، $\sum_{m=-\infty}^{\infty} |r_{xx}(m)| < \infty$ به این معنی

که $r'_{xx}(m)$ تخمین معتبری از تابع همبستگی خودی $r_{xx}(m)$ است.

نتیجه گیری فوق برای m های بزرگ صادق نیست، لذا از رابطه

$$r_{xx}(m) = \frac{1}{N} \sum_{n=0}^{N-|m|} x^*(n)x(n+m)$$

استفاده می گردد. این رابطه تخمین بایاس داری از $r_{xx}(m)$ ارائه می کند.

$$E(r_{xx}(m)) = \left(1 - \frac{|m|}{N}\right) r_{xx}(m)$$

ولی واریانس آن برای m های بزرگ کوچکتر است. به نحویکه

$$\lim_{N \rightarrow \infty} E(r_{xx}(m)) = r_{xx}(m)$$

تخمین طیف: برای تخمین طیف اگر از رابطه ی

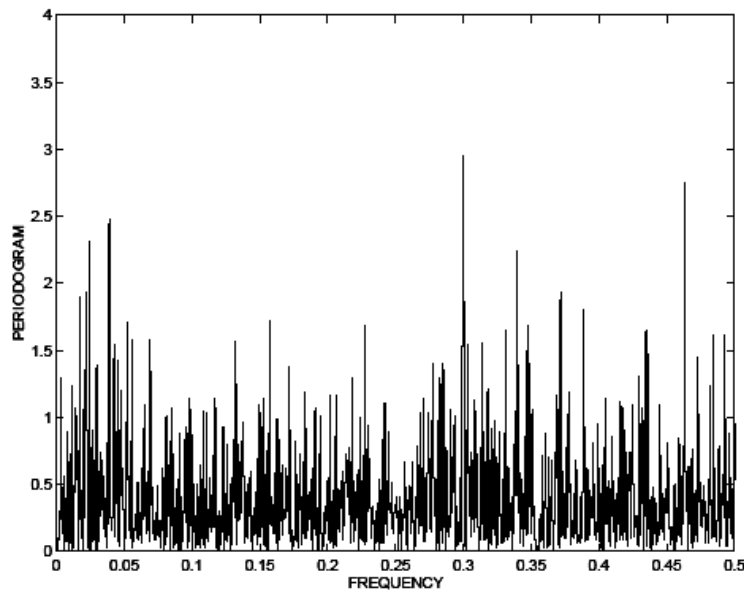
$$P_{xx}(f) = \sum_{m=-(N-1)}^{N-1} r_{xx}(m) e^{-j2\pi f m}$$

استفاده شود، به رابطه

$$P_{xx}(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n) e^{-j2\pi f n} \right|^2 = \frac{1}{N} |X(f)|^2$$

منجر می شود که به نام **پریودوگرام** خوانده می شود. نشان داده می شود که این تخمین به ازای $\infty \rightarrow f_{XX}(f)$ میل می کند ولی واریانس آن به حد قابل قبولی کاهش نمی یابد. لذا پریودوگرام تخمین معتبری برای طیف نیست.

مثال: شکل تخمین طیف سیگنال $x(n) = A \exp(j2\pi f_s n) + \xi(n)$ که در آن $f_s = 0.3$ و نویز سفید با واریانس ۱ با طول $N=10000$ و $A=0.1$ است را نشان میدهد.

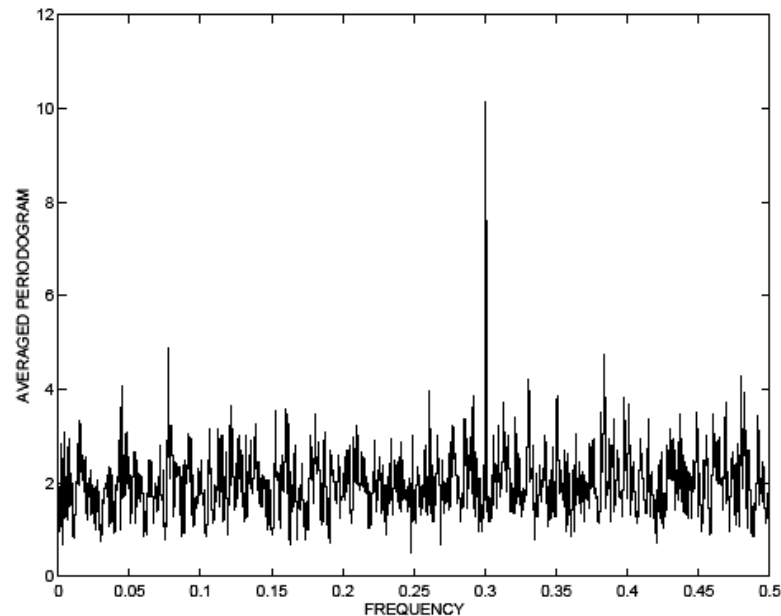


۲-۳-۹ روش بارتلت *Bartlett*: میانگین پریودوگرامها

در این روش دنباله به k دنباله مستقل M تایی تقسیم می گردد و هر دنباله در پنجره مثلثی بارتلت ضرب می گردد. سپس $P_{xx}(f)$ از میانگین $P_{xxi}(f)$ هر یک از k دنباله M تایی بدست می آید. به دلیل استفاده از پنجره رزولوشن طیف کاهش می یابد ولی واریانس تخمین کاهش می یابد. مقدار کاهش واریانس به نسبت $1/k$ است.

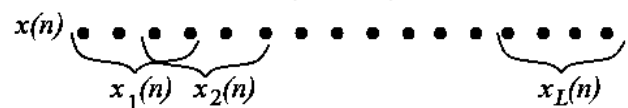
$$\begin{aligned} & \overbrace{x(n)}^{x_1(n)} \quad \overbrace{x(n)}^{x_2(n)} \quad \dots \quad \overbrace{x(n)}^{x_L(n)} \\ & \hat{S}_{xx}(\omega) = \frac{1}{k} \sum_{k=1}^k \hat{S}_{xxk}(\omega) \end{aligned}$$

شکل طیف با $A=0.1$ و $M=1000$ و $N=10000$ است.

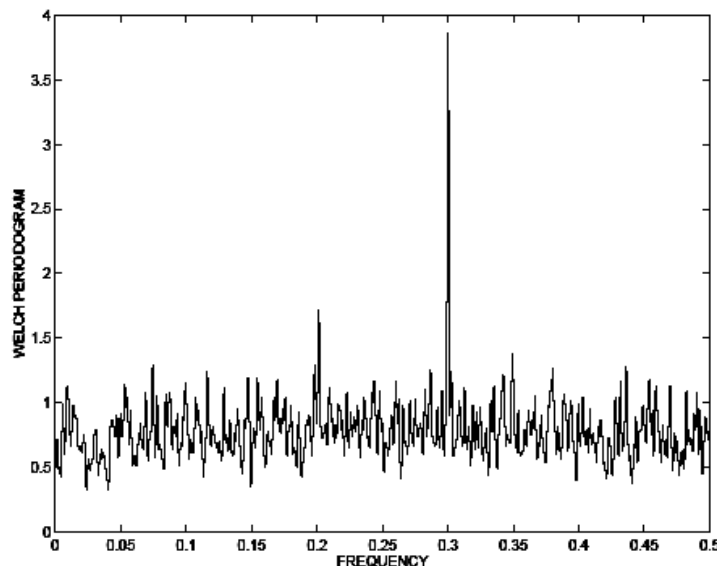


۳-۳-۹ روش ولش *Welch*: میانگین بهبود یافته پریودوگرامها

در این روش دنباله اصلی به k دنباله M تایی تجزیه می گردد. سپس اجازه داده می شود که دنباله ها به هنگام محاسبه طیف هم پوشانی داشته باشند. اگر مقدار هم پوشانی 50% باشد تعداد طیف هایی که محاسبه می گردد. $L=2k$ می رسد که در نهایت به کاهش $1/L=1/2$ واریانس در همپوشانی 50% می شود.



شکل طیف را با $A=0.1$, $N=10000$, $M=1000$ و $2/3$ همپوشانی نشان می دهد.



$$[P_{xx}, w] = \text{pwelch}(x, \text{window}, \text{nooverlap}, \text{nfft}, fs)$$

x : سیگنال مشاهدات، window : پنجره، nooverlap : تعداد نقاط هم پوشش، P_{xx} : طیف تخمین زده شده

w : محدوده فرکانس، nfft : تعداد نقاط برای محاسبه FFT، fs : فرکانس نمونه گیری

تخمین تابع تبدیل به tfe

$$T = \text{tfe}(x, y, \text{nfft}, fs, \text{window}, \text{nonoverlap})$$

x : سیگنال ورودی، y : سیگنال خروجی، nfft : تعداد نقاط برای محاسبه FFT، fs : فرکانس نمونه گیری

noverlap: تعداد نقاط هم پوشش

window: پنجره ای که باید استفاده شود،

دستور **tfe** (تخمین تابع تبدیل) این عملیات را انجام می دهد

$$tfe = (x, y, n, FFT, Ft, window)$$

$$h = \text{firls}(30, 0.2);$$

$$x = \text{rand}(20000, 1);$$

$$y = \text{filter}(h, 1, x);$$

$$tfe(x, y, 1024, [1, 1], 512)$$

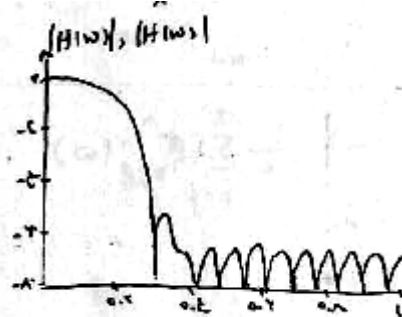
فیلتر FIR

نمونه برداری $x(n)$

نمونه برداری $y(n)$ و $x(n)$

نمونه برداری $y(n)$

نمونه برداری x و y



۴-۳-۹ تخمین پارامتریک طیف

در تخمین پارامتریک طیف بر سیگنال پنجره اعمال نمی گردد. لذا مسئله نشتی **leakage** وجود ندارد. ضمن اینکه رز ولوشن خوب حتی وقتی طول اطلاعات کوتاه است ارائه می دهد.

۱. در این روش برای سیگنالهای خروجی $y(n)$ مدل

$$H(z) = \frac{\sum_{k=0}^p b_k z^{-k}}{1 + \sum_{k=1}^p a_k z^{-k}}$$

در نظر گرفته می شود که فرض شده با سیگنال تصادفی با میانگین صفر و واریانس σ_w^2 تحریک گردیده است.

$$y(n) = - \sum_{k=1}^p a_k y(n-k) + \sum_{k=0}^p b_k w(n-k)$$

سیگنال تصادفی ورودی قابل اندازه گیری نیست ولی اگر خروجی $y(n)$ را فرآیند تصادفی ایستا در نظر بگیریم انهنگام ورودی $w(n)$ نیز ایستا باید باشد. در این حالت **PSD** سیگنال خروجی و واریانس ورودی مطابق رابطه با یکدیگر

$$\Gamma_{yy}(f) = |\Gamma_{ww}(f)|^2 = \sigma_w^2 |H(f)|^2$$

ارتباط پیدا می کنند. چون در اینجا m_w صفر در نظر گرفته شده ، همبستگی و کو واریانس و به همین ترتیب طیفهای آنها معادل یکدیگر می باشند.

۲. ارتباط بین تابع همبستگی و پارامترهای مدل: طرفین معادله دیفرنس را در $x^*(n-m)$ ضرب و امید ریاضی رابطه را حساب می کنیم.

$$\hat{\gamma}_{yy}(m) = E(y(n) y^*(n-m)) = - \sum_{k=1}^p a_k E(y(n-k) y^*(n-m)) + \sum_{k=0}^p b_k E(w(n-k) y^*(n-m))$$

$$\hat{\gamma}_{yy}(m) = - \sum_{k=1}^p a_k \hat{\gamma}_{yy}(m-k) + \sum_{k=0}^p b_k \hat{\gamma}_{wy}(m-k)$$

جمله دوم سمت راست رابطه را می توان اینگونه بسط داد.

$$\sum_{k=0}^p b_k E(w(n-k) \sum_{l=0}^{\infty} h_1(l) w(n-m-l))$$

$$= \sum_{k=0}^q b_k \sum_{\ell=0}^{\infty} h_1(\ell) E(\omega^*(n-k) \omega(n-m-\ell)) = \begin{cases} \sum_{k=0}^q b_k \sum_{\ell=0}^{\infty} h_1(\ell) \sigma_w^2 & k=m+\ell \\ 0 & \text{غیر} \end{cases}$$

در نوشتن رابطه فوق از سفید بودن نویز $w(n)$ استفاده شد. در نهایت

$$= \sum_{k=0}^q b_k h(k-m) = \sum_{k=0}^{q-m} b_{k+m} h_1(k)$$

حال می توان نوشت:

$$y_{yy}(m) = \begin{cases} -\sum_{k=1}^p a_k y_{yy}(m-k) + \sigma_w^2 \sum_{k=0}^{q-m} h_1(k) b_{k+m} & 0 \leq m \leq q \\ -\sum_{k=1}^p a_k y_{yy}(m-k) & m > q \\ y_{xx}^*(-m) & m < 0 \end{cases}$$

روابط اخیر برای مدل $ARMA$ است برای مدل AR ($q=0$) و MA ($p=0$) این روابط به شکل ذیل ساده می شوند.

$$AR \Rightarrow \begin{cases} -\sum_{k=1}^p a_k y_{yy}(m-k) + \sigma_w^2 & m=0 \\ -\sum_{k=1}^p a_k y_{yy}(m-k) & m>0 \\ y_{xx}^*(-m) & m<0 \end{cases}, MA \Rightarrow \begin{cases} \sigma_w^2 \sum_{k=0}^{q-m} b_{k+m} & 0 \leq m \leq q \\ 0 & m > q \\ y_{xx}^*(-m) & m < 0 \end{cases}$$

۳. تخمین ack ها

برای سیگنال $y(n)$ تخمین همبستگی خودی را از اطلاعات موجود محاسبه می کنیم

$$r_{yy}(m) = \frac{1}{N} \sum_{n=0}^{N-m-1} y^*(n) y(n+m) \quad m \geq 0$$

برای $m > q$ قبلا نوشتیم

$$y_{yy}(m) = -\sum_{k=1}^p a_k y_{yy}(m-k) \Rightarrow$$

حال بجای $y_{yy}(m)$ مقدار $r_{yy}(m)$ را در رابطه قرار می دهیم

$$\begin{bmatrix} r_{xx}(q) & \dots & r_{xx}(q-p+1) \\ \vdots & & \vdots \\ r_{xx}(N-1) & \dots & r_{xx}(N-p) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} r_{xx}(q+1) \\ \vdots \\ r_{xx}(N) \end{bmatrix} \Rightarrow R_{xx} a = -r_{xx}$$

تخمین LS پارامترها اینگونه بدست می آید.

$$R_{xx} a + r_{xx} = e \Rightarrow \hat{a} = -(R_{xx}' R_{xx})^{-1} R_{xx}' r_{xx}$$

اگر قرار بود مدل AR استخراج کنیم کافی بود $q=0$ منظور می شد و واریانس را رابطه

$$r_{xx}(0) = \sum_{k=1}^p \hat{a}_k r_{xx}(-k) + \sigma_w^2$$

در اختیار قرار می داد.

نکته: راه حل فوق برگرفته از بهینه سازی رابطه $y = H\theta + e$ است که در آن H می باید ماتریس ثابتها، θ بردار پارامترهای معین (غیر تصادفی)، y بردار مشاهدات تصادفی و e نویز گوسی با میانگین صفر است در این شرایط

$$\hat{\theta} = (H' H)^{-1} H' y$$

تخمین بهینه از θ را ارائه می کنند.

۴. تخمین bk ها

بعد از تخمین ak ها مشاهدات $y(n)$ از معکوس فیلتر تخمین زده شده AR

عبور داده شده تا $\hat{v}(n)$ بدست آید. این خروجی تخمین خروجی بخش MA مدل به ازای ورودی تصادفی است.

$$\hat{v}(n) = g(n) + \sum_{k=1}^p \hat{a}_k g(n-k) \quad n=0, 1, \dots, N-1$$

در این قسمت ابتدا بجای تخمین MA مدل AR با درجه $p > q$ تخمین زده می شود که معادل MA باشد. این کار مشابه بند ۳ صورت می گیرد که در نهایت

$$\beta = -[R_{vv}^{-1} R_{vu}]^{-1} R_{vu}^{-1} r_{vu}$$

را تولید می کنند. بعد از محاسبه β نوبت تخمینی برای مدل MA با درجه q می رسد

$$MA(q) = AR(p') \Rightarrow B(z) = \frac{1}{\beta(z)} = \frac{1}{1 + \sum_{k=1}^p \beta_k z^{-k}} \quad \frac{B(z)}{\beta(z)} = 1$$

این تساویها را می توان برقرار کرد اگر $\beta p'$ به صفر نزدیک شود. از $B(z)\beta(z) = 1$ به جمع کانولوشن می رسم.

$$\sum_{k=0}^n b_k \beta_{n-k} = \delta(n) \Rightarrow \sum_{k=0}^q b_k \beta_{n-k} = \delta(n) \quad \therefore \begin{cases} \beta_k = 0 & k < 0 \\ \beta_0 = 1 \end{cases}$$

$$\Rightarrow n=0 \Rightarrow b_0 \beta_0 = 1 \Rightarrow \boxed{b_0 = 1} \Rightarrow \beta_n + \sum_{k=1}^q b_k \beta_{n-k} = \delta(n)$$

حال با تشکیل تابع خطا و حداقل کردن آن تخمین b بدست می آید.

$$\varepsilon = \sum_{n=0}^r \left[\beta_n + \sum_{k=1}^q b_k \beta_{n-k} \right]^2$$

$$\Rightarrow \frac{d\varepsilon}{db_k} = 0 \Rightarrow \boxed{\hat{b} = -R_{\beta\beta}^{-1} r_{\beta\beta}}$$

$$R_{\beta\beta}(1-z^{-1}) = \sum_{n=0}^p \beta_n \beta_{n+1-z^{-1}}$$

$$x(z) = 1, z^{-1}, \dots, z^{-p}$$

$$r_{\beta\beta}(z) = \sum_{n=0}^p \beta_n \beta_{n+1}$$

$$x = 1, z^{-1}, \dots, z^{-p}$$

به این ترتیب براساس سیگنال $y(n)$ مدل

$$H(z) = \frac{\sum_{k=0}^q \hat{b}_k z^{-k}}{1 + \sum_{k=1}^p \hat{a}_k z^{-k}}$$

بدست می آید. این مدل بعنوان مدل های $ARMA$ شناخته می شود.

۲. تخمین طیف براساس مدل سازی AR با درجه p برای سیگنال مشاهدات x به روش *yule walker*

$$P_{xx}=pyulear(x,p)$$

۳. تخمین طیف براساس مدل سازی AR با درجه p به روش *burg* برای سیگنال مشاهدات x

$$P_{xx}=pburg(x,p)$$

۴. تخمین مدل AR به روش *yule Walker* برای سیگنال مشاهدات x

$$[a,e]=aryule(x,p)$$

p : در چه چند جمله ای مدل AR ، α : تخمین ضرایب مدل e : واریانس نویز ورودی به مدل

$$H(z) = \frac{1}{1 + \sum_{k=1}^p a_k z^{-k}}$$

۵. تخمین مدل AR با درجه p برای سیگنال مشاهدات x به روش *burg*

$$[a,e]=arburg(x,p)$$

۴-۹ فیلترهای وفقی Adaptive Filters

- در فیلترهای کلاسیک، فیلتر بر اساس اطلاعات ثابت از قبل معین در باره سیگنال و نویز طرح و مورد استفاده قرار می گیرد.
- در فیلترهای بهینه *optimal*، سیگنال و نویز ناشی از یک فرایند تصادفی با مشخصات آماری معلوم فرض می گردند. هدف فیلتر آن است که تفاوت بین سیگنال نویزی s و سیگنال مطلوب d را طبق معیاری به حداقل برساند $s=d+e$. این معیار ها می تواند مجموع قدر مطلق انحرافات دوسیگنال L^1 ، مجموع مربعات انحرافات L^2 ، ماکزیمم انحراف و غیره باشد که باید به حداقل برسد. طرح فیلتر بهینه مربوط به *Kalmogroph(1939)* و *Wiener(1942)* می باشد که به نام فیلتر وینر شناخته می شود.
- فیلتر وفقی: در بسیاری از کاربردها سیستم ممکن است ناشناخته و یا متغیر با زمان باشد. در این موارد فیلتر وفقی مشخصات خود را تغییر می دهد تا بتواند تغییرات ایجاد شده در سیستم را دنبال کند. فیلترهای وفقی کاربرد گسترده ای در مخابرات دارند.

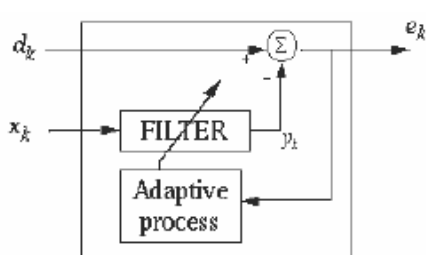
کاربرد فیلترهای وفقی

فیلترهای وفقی. را می توان به راههای متنوع در کاربردهای وسیعی بکار گرفت. نمونه کاربردهای آن

- *noise cancellation*
- *adaptive system identification*
- *channel equalisation*
- *prediction*

می باشند.

۱-۴-۹ مدل کلی فیلتر وفقی



شکل ساختمان کلی فیلتر وفقی را نشان می دهد که از ۶ جزء تشکیل یافته است.

- x_k - Input signal
- d_k - Desired signal
- y_k - Output signal
- e_k - Error signal
- *FILTER* - Filtering process
- *Adaptive process* - Some kind of algorithm

هدف فیلتر وفقی، فیلتر کردن سیگنال ورودی x_k است تا به نحوی سیگنال فیلتر

شده y_k مشابه سیگنال مطلوب d_k گردد. سیگنال خطای e_k معیار مقدار شباهت بین y_k و d_k است. از e_k طبق الگوریتمی استفاده می شود تا پارامترهای فیلتر تنظیم گردد. فیلتر هر نوعی ممکن است باشد ولی اغلب از *FIR* بدلیل پایداری و سادگی استفاده می شود.

$$y(k) = \sum_{l=0}^{N-1} w(l)x(k-l) = W'X_k, \quad W' = [w_0, \dots, w_{N-1}], \quad X = [x_k, \dots, x_{k-N+1}]'$$

$$J = E(e(k)^2) = E \left(\left[d(k) - \sum_{l=0}^{N-1} w(l)x(k-l) \right]^2 \right) = E \left([d(k) - W'X_k]^2 \right) =$$

$$P = \begin{bmatrix} R_{dx}(0) \\ \vdots \\ R_{dx}(N-1) \end{bmatrix}, \quad R = \begin{bmatrix} R_{xx}(0) & \dots & R_{xx}(N-1) \\ \vdots & \ddots & \vdots \\ R_{xx}(N-1) & \dots & R_{xx}(0) \end{bmatrix}$$

با حداقل کردن J روابط ذیل بدست می آید.

$$\Delta J/dW = -(2P) + 2RW$$

$$d(A^TW)/dw = A^T \quad d(WMW)/dw = 2MW \text{ for symmetric } M$$

$$(W_{opt}R=P, \Rightarrow W_{opt}=R^{-1}P)$$

R ماتریس همبستگی خودی است، لذا غیر منقی است و برای حالت مثبت مینیمم منحصر بفرد دارد.

۲-۴-۹ معیار مقایسه بین روشهای مختلف بهینه سازی

- همگرایی الگوریتم *convergence*
- قابلیت وفقی داشتن *tracking*
- پایداری *stability* صحت *Accuracy* و ثبات عملکرد *robustness* در محاسبات
- پیچیدگی و حجم محاسبات
- ساختار سخت افزار مورد نیاز

ملاحظات محاسباتی

محاسبه θ به $O(N^3)$ محاسبه نیاز دارد. ولی از آنجائیکه ماتریس R ساختار *Toeplitz* دارد با روش *Levinson-Durbin* با حجم $O(N^2)$ محاسبه قابل حل است. این حجم محاسبه برای کاربردهای بلا درنگ زیاد است زیرا محاسبه خروجی فیلتر نیز $O(N)$ محاسبه دیگر نیاز دارد. ۲ روش اساسی برای حل مشکل وجود دارد.

1. Take advantage of the fact that R^{k+1} is only slightly changed from R^k to reduce the computation to $O(M)$; these algorithms are called **Fast Recursive Least Squares algorithms**; all methods proposed so far have stability problems and are dangerous to use.

Recursively! $r_{xx}^k(l) = r_{xx}^{k-1}(l) + x_k x_{k-l} - x_{k-N} x_{k-N-l}$ This is critically stable, so people usually do $(1 - \alpha) r_{xx}^k(l) = \alpha r_{xx}^{k-1}(l) + x_k x_{k-l}$

2. Find a different approach to solving the optimization problem that doesn't require explicit inversion of the correlation matrix such as steepest decent method and LMS.

Note: Adaptive algorithms involving the correlation matrix are called *Recursive least Squares (RLS)* algorithms. Historically, they were developed after the LMS algorithm, which is the simplest and most widely used approach $O(M)$. $O(M^2)$ RLS algorithms are used in applications requiring very fast adaptation.

$$\begin{aligned}
 e &= d(n) - \hat{d}(n) = d(n) - \sum_{k=1}^N a_k \phi(n-k) \\
 &= d(n) - \phi' \theta \\
 J &= E(e'e) = \sigma_d^2 - \theta' \phi_d - \phi_d' \theta + \theta' (\phi \phi') \theta \\
 &\Rightarrow \hat{\theta} = (\phi \phi')^{-1} \phi_d = R^{-1} p \\
 \theta(n+1) &= \theta(n) + \frac{1}{r} \mu \left(- \frac{dJ}{d\theta} \right) \\
 &\Rightarrow \theta(n+1) = \theta(n) + \frac{1}{r} \mu (r p - R \theta) \\
 &\Rightarrow \theta(n+1) = \theta(n) + \mu (p - R \theta_n) \\
 &\Rightarrow \theta(n+1) = \theta(n) + \mu \phi^T (d - \phi' \theta_n) \\
 \theta &= [a_1, \dots, a_N] \\
 \phi' &= [\phi(n-1), \dots, \phi(n-N)]
 \end{aligned}$$

۳-۴-۹ تندترین شیب Steepest Descent

پارامترهای روش بهینه LS تابعی مربعی می سازند که اگر R *positive definite* باشد، سطح تابع N بعدی $E\{e^2\}$ کاسه ای در R^N می سازد که هدف یافتن ته آن است. برای سطح خطای مربعی ته آن را با محاسبه $R^{-1}p$ در یک گام می توان بدست آورد. در سیستم هایی که بازمان تغییر می کنند، موقعیت ته کاسه تغییر می کند، ولی اگر تغییرات کند باشد می توان آنرا دنبال کرد.

ساده ترین و قدیمیترین روش بازگشتی برای یافتن مینیمم روش تندترین شیب است. در این روش از نقطه ای اولیه شروع و در جهت عکس تندترین شیب حرکت تا به مینیمم دسترسی پیدا شود.

عملکرد این روش به انتخاب μ اندازه گام بستگی دارد.

if μ is too large, the iterations could bounce

back and forth up out of the bowl. However, if μ is too small, it could take many iterations to approach the bottom. In summary, the gradient descent algorithm for solving the Weiner filter problem is:


```

Guess  $W^0$ 
do  $i = 1, \infty$ 
 $\nabla^i = -(2P) + 2RW^i$ 
 $W^{i+1} = W^i - \mu \nabla^i$ 
repeat
 $W_{opt} = W^\infty$ 

```

بار محاسباتی این روش محاسبه شیب است $\nabla = 2RW - 2P$ است که محاسبه P و R را بدنبال دارد که حجم محاسباتی آن $O(N^2)$ است که برای استفاده بلا درنگ مناسب نیست..

۴-۴-۹ تقریب شیب: Least Mean Squares

ایده اصلی در LMS بکار گیری تقریب شیب بجای محاسبه دقیق آن است.

Note that e_k^2 itself is a very noisy approximation to $E(e_k^2)$. We can get a noisy approximation to the gradient by finding the gradient of e_k^2 ! Widrow and Hoff first published the LMS algorithm, based on this clever idea, in 1960.

$$\nabla^k = \frac{\partial}{\partial W} (e_k^2) = 2e_k \frac{\partial}{\partial W} (d_k - W^k X^k) = 2e_k (-X^k) = -(2e_k X^k)$$

این تقریب $\hat{\nabla} = -2e_k X_k$ جایگزین محاسبه دقیق در روش تندترین شیب می گردد

$$\theta(n+1) = \theta(n) + 2\mu \phi(n) e(n)$$

این فرمول **Widrow-Hoff LMS Algorithm** است که پایدار و حجم محاسباتی آن N ضرب و N جمع برابر $O(N)$ است. سرعت همگرایی آن کمتر از RLS است. μ سرعت همگرایی و پایداری را تعیین می کند.

Example 1: The LMS Adaptive Filter Algorithm

$$e_k = d_k - y_k = d_k - \sum_{i=0}^{M-1} (w_i x_{k-i}) = d_k - X^k T W^k \quad X^k = \begin{pmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-M+1} \end{pmatrix} \quad W^k = \begin{pmatrix} w_0^k \\ w_1^k \\ \vdots \\ w_{M-1}^k \end{pmatrix}$$

$$1. y_k = W^{kT} X^k = \sum_{i=0}^{M-1} (w_i^k x_{k-i})$$

$$2. e_k = d_k - y_k$$

$$3. W^{k+1} = W^k - \mu \hat{\nabla}^k = W^k - \mu (-2e_k X^k) = W^k + 2\mu e_k X^k \quad (w_i^{k+1} = w_i^k + 2\mu e_k x_{k-i})$$

The LMS algorithm is often called a **stochastic gradient algorithm**, since $\hat{\nabla}^k$ is a noisy gradient. This is by far the most commonly used adaptive filtering algorithm, because

- it was the first
- it is very simple
- in practice it works well (except that sometimes it converges slowly)
- it requires relatively little computation
- it updates the tap weights every sample, so it continually adapts the filter
- it tracks slow changes in the signal statistics well

Computational Cost of LMS

| To Compute | y_k | e_k | W^{k+1} | = Total |
|------------|-------|-------|-----------|----------|
| multiplies | M | 0 | $M + 1$ | $2M + 1$ |

| | | | | |
|------|-----|---|---|----|
| adds | M 1 | 1 | M | 2M |
|------|-----|---|---|----|

So the LMS algorithm is $O(M)$ per sample. In fact, it is nicely balanced in that the filter computation and the adaptation require the same amount of computation.

Note that the parameter μ plays a very important role in the LMS algorithm. It can also be varied with time, but usually a constant μ ("convergence weight factor") is used, chosen after experimentation for a given application.

Tradeoffs

large μ : fast convergence, fast adaptivity

small μ : accurate $W \Rightarrow$ less misadjustment error, stability

Complex LMS

LMS for complex data and coefficients (such as quadrature communication systems) takes the form

$$y_k = W_k^H X_k$$

$$e_k = d_k - y_k$$

$$W_{k+1} = W_k + 2\mu e_k^* X_k$$

It is derived in exactly the same way as LMS, using the following complex vector differentiation formulas

$$\frac{d}{dW} (P^H W) = 0$$

$$\frac{d}{dW} (W^H P) = 2P$$

$$\frac{d}{dW} (W^H R W) = 2RW$$

or by differentiating with respect to the real and imaginary parts separately and recombining the results.

Normalized LMS

In "normalized" LMS, the gradient step factor μ is normalized by the energy of the data vector:

$\mu_{NLMS} = \frac{\alpha}{X_k^H X_k + \sigma}$ where α is usually $\frac{1}{2}$ and σ is a very small number introduced to prevent division by zero if $X_k^H X_k$ is very small.

$$W_{k+1} = W_k + \frac{1}{X_k^H X_k + \sigma} e_k X_k$$

The normalization has several interpretations

- corresponds to the 2nd-order convergence bound
- makes the algorithm independent of signal scalings
- adjusts W^{k+1} to give zero error with current input: $W^{k+1} X_k = d_k$
- minimizes mean effort at time $k + 1$
- NLMS usually converges much more quickly than LMS at very little extra cost; NLMS is very commonly used. In some applications, normalization is so universal that "we use the LMS algorithm" implies normalization as well.

LMS VARIANTS: Block LMS

The weight vector of the FIR filter is held constant for a few iterations while an improved estimate of the performance surface gradient is obtained.

LMS VARIANTS: Variable Step Size

The value of μ is chosen large at the beginning and then is progressively reduced to a smaller size to iterate closer to the optimum value.

LMS VARIANTS: Leaky LMS

This variation is addressed to systems with small wordlengths where round-off noise is fed back to adaptive weights and accumulates in time without bound leading to overflow.

A small bias factor b , which is slightly less than one, is built in to bias each weight toward zero on each iteration counteracting the effect of noise build up:

$$\mathbf{W}_{k+1} = b \mathbf{W}_k + 2\mu \varepsilon_k \mathbf{X}_k$$

LMS VARIANTS: Sign Error LMS

The computation needed by the adaptive algorithm can be reduced to zero multiplications and N additions using only the sign of the error signal (and making μ be a power of two):

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \mu \text{sign}[\varepsilon_k] \mathbf{X}_k$$

۵-۴-۹ روش بازگشتی حداقل مربعات Recursive Least Squares

در این روش محاسبات بر مبنی تقریب P و R و استفاده از فرمول بازگشتی است. RLS به $O(N^2)$ محاسبه در هر بار نیاز دارد و می توان نشان داد که اگر فیلتر FIR باشد، این محاسبات به $O(N)$ کاهش می یابد. در این روش متفاوت با روش قبلی نیاز به اطلاعات خروجی و ورودی برای تخمین پارامترها نسبت عکس با فراهم شدن هر اندازه گیری جدید از خروجی و ورودی می توان اقدام به تصحیح تخمین پارامترها نمود. حل رابطه وینر $W=R^{-1}P$ به $O(N^3)$ عملیات نیاز دارد. برای کاهش عملیات از این ایده استفاده می شود که بجای هر بار محاسبه R^{-1} و P آنها بصورت بازگشتی محاسبه گردند

$$R(n+1)=R(n)+\varphi(n+1)\varphi'(n+1), \quad P(n+1)=P(n)+\varphi(n+1)y(n+1),$$

سپس بر اساس قضیه معکوس ماتریس

$$(A+BCD)^{-1}=A^{-1}-A^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1}$$

می توان نوشت

$$R(n+1)^{-1}=[R(n)+\varphi_{n+1}\varphi'_{n+1}]^{-1}=R(n)^{-1}-R(n)^{-1}\varphi_{n+1}\frac{1}{1+\varphi'_{n+1}R(n)^{-1}\varphi_{n+1}}\varphi'_{n+1}R(n)^{-1}$$

حال می توان نوشت.

$$\theta(n+1)=\theta(n)+k(n+1)\cdot e(n+1)$$

$$e(n)=y(n)-\varphi'(n)\theta(n-1)$$

در این رابطه θ بردار پارامتر و φ بردار مشاهدات بر اساس رابطه

$$y(n)=\varphi'(n)\theta(n)$$

بر همین اساس رابطه ای برای P می توان نوشت. در نتیجه بصورت بازگشتی با $O(N^2)$ قابل محاسبه می شود.

$$e(n)=[y(n+1)-\varphi'(n+1)\theta(n)]$$

در این رابطه k به نام $Kalman Gain$ و مقدار خطا است. از فرمول ذیل برای محاسبه بازگشتی k می توان استفاده کرد.

$$k(n+1)=[R(n+1)]^{-1}\varphi_{n+1}=\frac{R(n)^{-1}}{1+\varphi'_{n+1}R(n)^{-1}\varphi_{n+1}}K_{n+1}$$

شرایط اولیه الگوریتم $R=\alpha I$ و $\theta=\theta_0$ در نظر گرفته می شود. می توان نشان داد که این شرایط اولیه به معنی تغییر در تابع معیار به شکل ذیل است.

$$J=\alpha^2\|\theta-\theta_0\|_2^2+\sum_{l=1}^ke_l^2$$

از آنجاییکه سیستمها ممکن است فقط در زمانهای کوتاه مشخصات ثابتی داشته باشند برای آنها از $Windowed RLS$ یا $sliding window$ RLS یا $exponentially weighted RLS$ استفاده می گردد.

الگوریتم اجرای این برنامه از اینقرار است.

(۱) مقادیر اولیه: به γ مقدار اولیه مثلاً $\gamma=1$ ، به θ حدس اولیه در رابطه با پارامترها مثلاً $\theta=[0,\dots,0]$ و به P مقدار اولیه مثلاً که $P=\alpha I$

که α عددی بزرگ باشد را اعمال کنید. اگر مدل

$$y(n)=\sum_{j=1}^p a_j y(n-j)+\sum_{i=0}^q b_i u(n-i)$$

باشد طول بردار θ ، برابر $l=p+q+1$ ، ماتریس P ابعاد $l \times l$ دارد.

(۲) شمارنده را $i=1$ قرار دهید

(۳) بردار φ را بسازید.

$$\varphi(i)=[y(i-1), \dots, y(i-p), u(i), \dots, u(i-q)]'$$

مقادیر $y(n)$, $u(n)$ برای $n < 0$ برابر صفر است.

(۴) k را حساب کنید

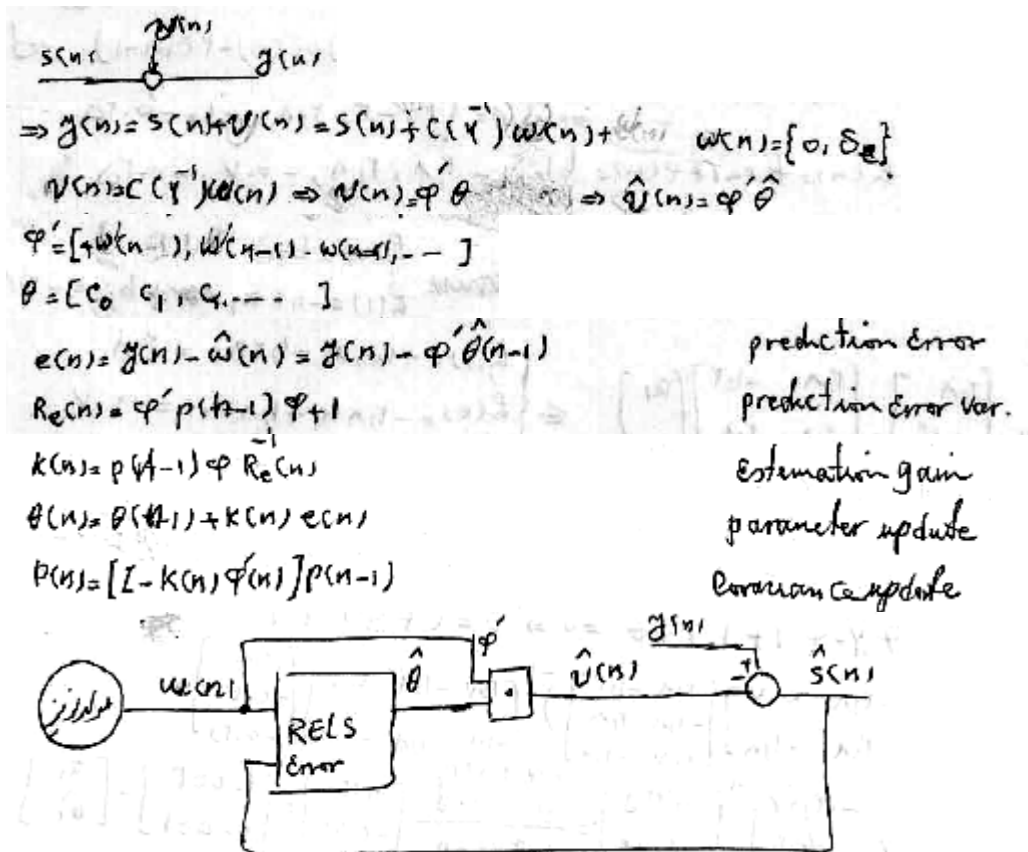
$$k(n) = \frac{p(n-1)}{\gamma} \varphi(n-1) \left(1 + \varphi(n-1) \frac{p(n-1)}{\gamma} \varphi(n-1)' \right)^{-1}$$

(۵) تخمین جدید پارامتر

$$\begin{aligned} \hat{\theta}(n) &= \hat{\theta}(n-1) + k(n) [y(n) - \varphi(n-1)' \hat{\theta}(n-1)] \\ p(n) &= \frac{1}{\gamma} (I - k(n) \varphi(n-1) \varphi(n-1)') p(n-1) \end{aligned}$$

(۶) $i=i+1$ و بازگشت به ۳

مثال: حذف نویز از سیگنال بصورت افقی



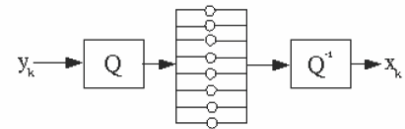
Beyond LMS ۶-۴-۹

FIR adaptive filter algorithms with faster convergence. Since the Wiener solution can be obtained on one step by computing $W_{opt} = R^{-1}P$, most RLS algorithms attempt to estimate R^{-1} and P and compute W_{opt} from these. There are a number of $O(N^2)$ algorithms which are stable and converge quickly. A number of $O(N)$ algorithms have been proposed, but these are all unstable except for the lattice filter method. This is described to some extent in the text.

The adaptive lattice filter converges quickly and is stable, but reportedly has a very high noise floor. Many of these approaches can be thought of as attempting to "orthogonalize" R , or to rotate the data or filter coefficients to a domain where R is diagonal, then doing LMS in each dimension separately, so that a fast-converging step size can be chosen in all directions.

Frequency-domain methods

Frequency-domain methods implicitly attempt to do this: If QRQ^{-1} is a diagonal matrix, this yields a fast algorithm. If Q is chosen as an FFT matrix, each channel becomes a different frequency bin. Since R is Toeplitz and not a circulant, the FFT matrix will not exactly diagonalize R , but in many cases it comes very close and frequency domain methods converge very quickly. However, for some R they perform no better than LMS. By using an FFT, the transformation Q becomes inexpensive $O(N \log N)$. If one only updates on a block-by-block basis (once per N samples), the frequency domain methods only cost $O(\log N)$ computations per sample, which can be important for some applications with large N . (Say 16,000,000)



Adaptive IIR filters

Adaptive IIR filters are attractive for the same reasons that IIR filters are attractive: many fewer coefficients may be needed to achieve the desired performance in some applications. However, it is more difficult to develop stable IIR algorithms, they can converge very slowly, and they are susceptible to local minima. Nonetheless, adaptive IIR algorithms are used in some applications (such as low frequency noise cancellation) in which the need for IIR-type responses is great. In some cases, the exact algorithm used by a company is a tightly guarded trade secret.

Most adaptive IIR algorithms minimize the prediction error, to linearize the estimation problem, as in deterministic or block linear prediction.

Due to the slow convergence and the difficulties in tweaking the algorithm parameters to ensure stability, IIR algorithms are used only if there is an overriding need for an IIR-type filter.

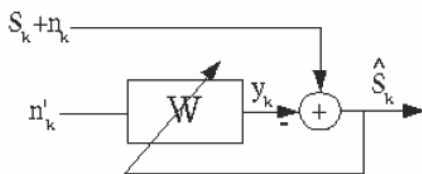
۷-۴-۹ حذف وفتی نویز و تداخل Adaptive Interference (Noise) Cancellation

هدف حذف نویز و تداخل تا حد ممکن بصورت وفتی است. برای مثال به حذف صدای موتور ماشین و غیره از رسیدن به اتاق و مسافران می توان اشاره کرد.

روش حذف تداخل

به شکل ذیل توجه کنید.

$$E[\epsilon_k^2] = E[(s_k + n_k - y_k)^2] = E[s_k^2] + 2E[s_k(n_k - y_k)] + E[(n_k - y_k)^2]$$



We assume s_k , n_k , and n'_k are zero-mean signals, and that s_k is independent of n_k and n'_k . Then

$$E[s_k(n_k - y_k)] = E[s_k]E[n_k - y_k] = 0$$

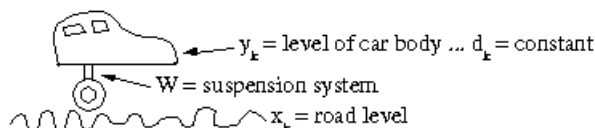
$$E[\epsilon_k^2] = E[s_k^2] + E[(n_k - y_k)^2]$$

Since the input signal has no information about s_k in it, minimizing $E[\epsilon_k^2]$ can only affect the second term, which is

$$W = R_{nn'}^{-1} P_{nn'}$$

the standard Wiener filtering problem, with solution

سیستم تعلیق ماشین active suspension system



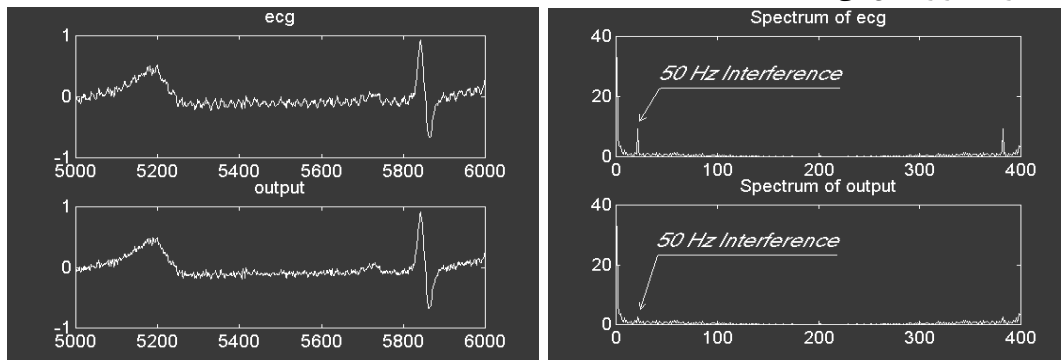
رفتار سیستم تعلیق بهینه تحت شرایط جاده و وزن مسافران و بار تغییر می کند

حذف تداخل ۵۰ هرتز CANCELLING 50 HZ

INTERFERENCE IN ECGs

ECG دستگاه ثبت سیگنالهای الکتریکی قلب است که از طریق پربهایی که به قفسه سینه چسبانده می شود دریافت می کند. این سیگنالها در معرض نویز و تداخلهای زیاد/ از جمله ۵۰ هرتز برق شهر هستند. این تداخل را با فیلتر ناچ می توان حذف کرد ولی برای اینکه پهنای بان بسیار کمی داشته باشد، درجه آن و متعاقبا تعداد المانهای مورد نیاز افزایش می یابد. روش دیگر استفاده از فیلتر **FIR** درجه ۲ است. شکل سیستم استفاده شده، سیگنال دریافتی و طیف آنرا نشان می دهد.

شکل زیر عملکرد فیلتر را نشان می دهد.



The signal comes from a man's chest, taken at a sample rate of 1 KHz. The output of the adaptive system was obtained using the following parameters:

Algorithm: LMS

Order of the FIR filter: 2

Stability factor (μ): $1e-6$

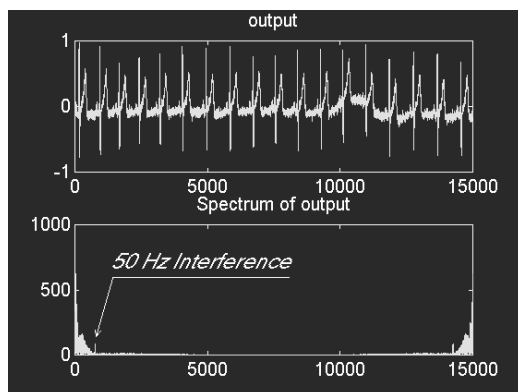
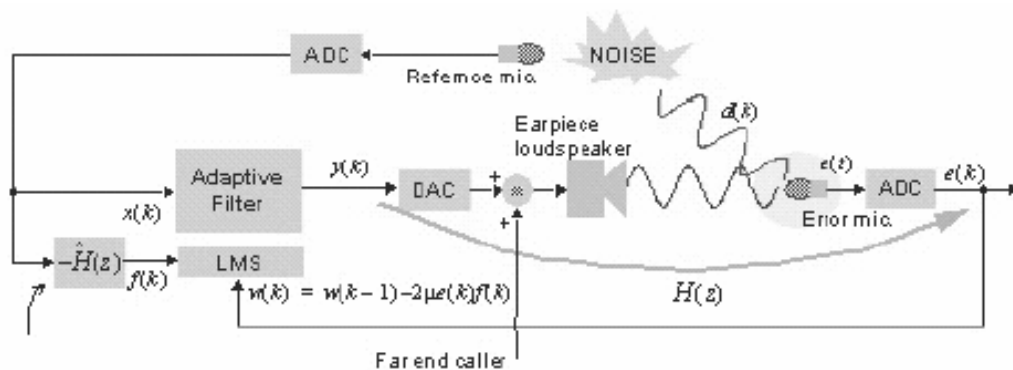
Initial weights vector: $w1 = 0.052$

$w2 = -0.020$

Noise reference input: 50 Hz cosinewave

حذف صدای نویزی

اصل اساسی این کار اندازه گیری صدای نویزی در نقاط مختلف با میکروفون و تولید صدای دیگری با بلندگو است که بتواند صدای اولیه را خنثی کرده و نویز را از بین ببرد. از کاربردهای آن می توان به استفاده در کابین هواپیما، ماشین و غیره اشاره کرد. در حالیکه طرح آن ساده بنظر می رسد ولی پیاده سازی آن پیچیده است. تکنولوژی که می تواند در این امر توفیق داشته باشد، حذف وقتی نویز است.

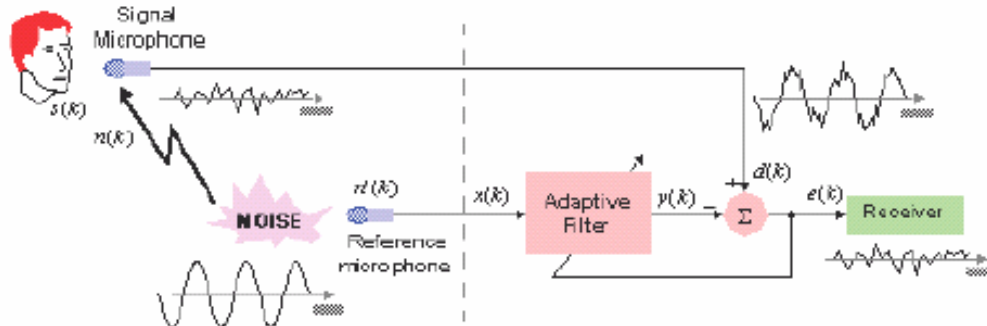


ECG



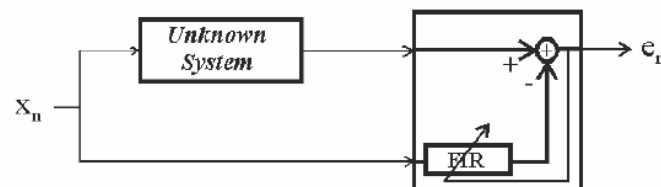
کاهش نویز محیط در ارسال سیگنال صوتی

Noise Cancellation. The speech signal to be transmitted (perhaps a mobile phone in a car) is spectrally masked by noise, from, for example a car engine. By using an adaptive filter, we can attempt to minimise the error by finding the correlation between the noise at the signal microphone and the (correlated) noise at the reference microphone.



۸-۴-۹ شناسایی وفقی سیستمها

هدف تعیین وفقی مدل سیستم از اطلاعات ورودی و خروجی است که ساختار آن در شکل نشان داده شده است.



For instance, suppose the 'unknown system' is a fifth order FIR filter with the following weights:

$$w_1 = 0.3835$$

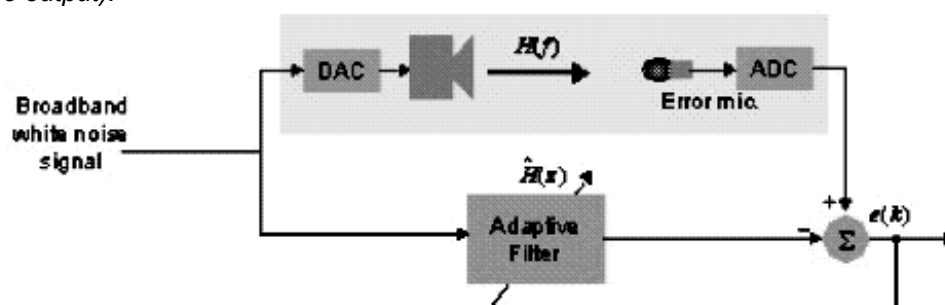
$$w_2 = 0.1926$$

$$w_3 = -0.4994$$

$$w_4 = 0.3649$$

$$w_5 = -0.7682$$

Now suppose we apply a white noise input signal ('x' in the picture above).. The system will adapt itself yielding zero output).

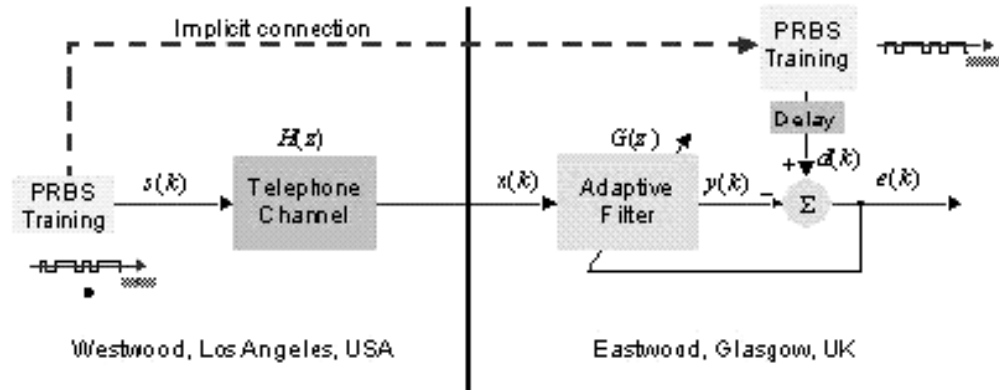


This works perfectly when the system to be identified has got a frequency response that matches that of a certain FIR filter. But what happens if the 'unknown system' is a all-pole filter? Then the FIR filter will try its best

۹-۴-۹ Inverse system Identification

Channel Equalisation. In order to equalise (or inverse identify the telephone channel) an identical pseudo-random binary sequence is set up at both ends of the telephone call. This is equivalent to the implicit connection shown. If indeed the error does adapt to $e(k) = 0$ then $d(k) = y(k) = s(k-D)$ where D is the delay of the signal through the channel. Therefore after an initial training mode

we can switch to data transmission mode and the adaptive filter will perform (zero-forcing) equalisation. Data equalisation for cable and mobile channel equalisation has a number of varied implementations and additional processing elements to produce decision feedback equalisers, blind

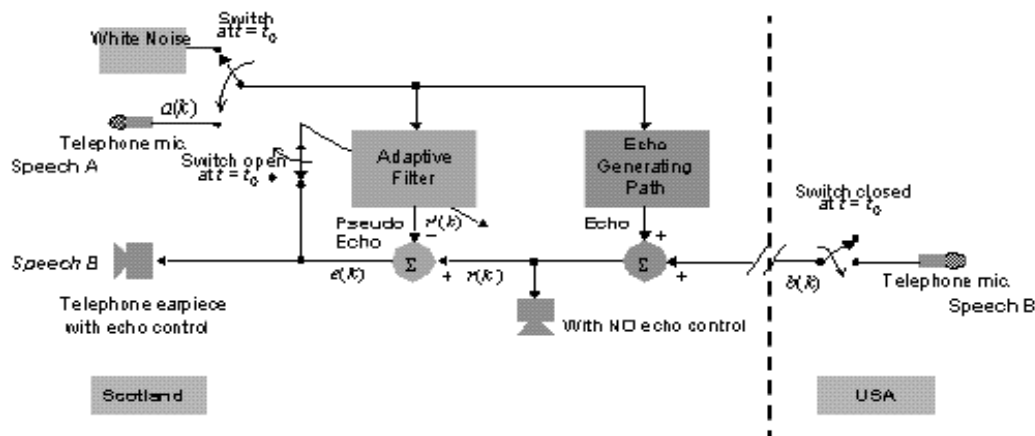


equalisers using decision directed mode and so on.

Prediction ۱۰-۴-۹

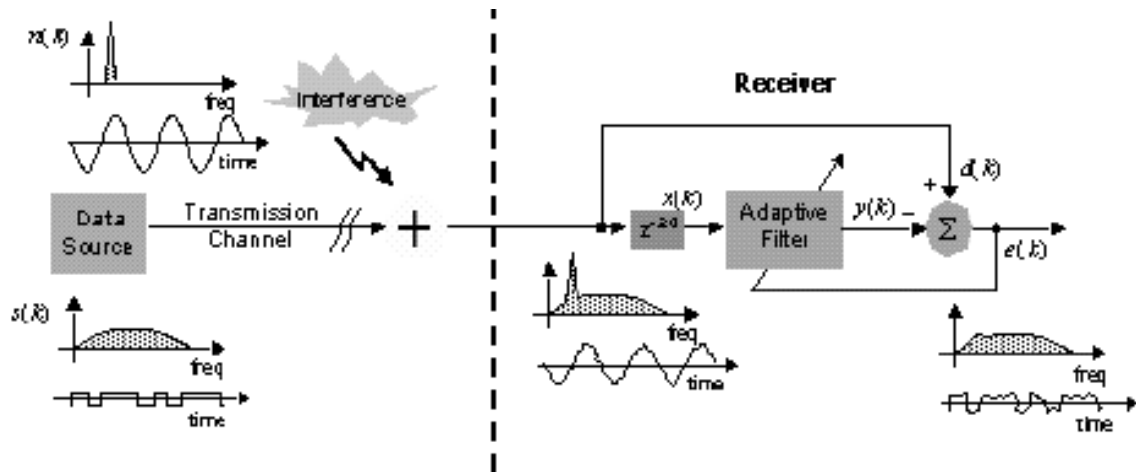
حذف اکو Adaptive Echo Cancellation

The adaptive filter is first trained by passing white noise through the telephone channel, ensuring the far-end signal $b(k)=0$, and thus performing an adaptive system identification on the echo path. After adaption at time if the error $e(k)=0$ then the adaptation is switched off and the signal at the earpiece is essentially: $e(k) = \text{Echo} - \text{PseudoEcho} + b(k) \approx b(k)$ and the echo has been cancelled. Clearly the echo canceller has to be of sufficient impulse response to model the echo path, and the adaption time must be short enough to be an acceptable to the callers - these are the decisions of the DSP engineer. This type of echo control is also used in V32 modems where the microphones are replaced by binary data sources.



CDMA interference suppression.

If a broadband (stochastic) signal has interference from a narrowband (periodic) source we can use a prediction architecture such that when the adaptive filter attempts to find the correlation between $d(k)$ and $y(k)$ only the narrowband signal can be correlated. Hence when performing $d(k) - y(k)$ the narrowband signal is attenuated and we find that $e(k) \approx s(k)$. One application for this type of predictive architecture for broadband spread spectrum CDMA signals interfered with by a narrowband jammer.



Summary of Adaptive Filtering Methods ۱۱-۴-۹

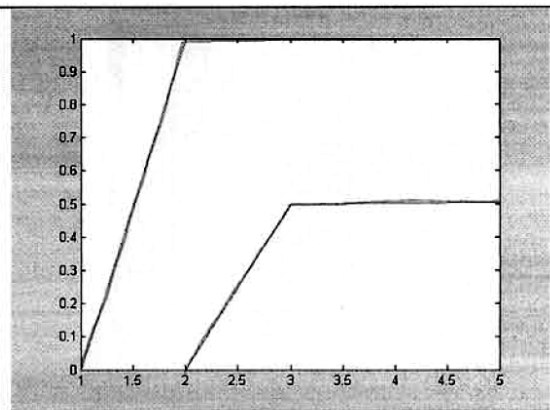
1. LMS - remains the simplest and best algorithm when slow convergence is not a serious issue (typically used) $O(N)$
 2. NLMS - simple extension of the LMS with much faster convergence in many cases (very commonly used) $O(N)$
 3. Frequency-domain methods - offer computational savings ($O(\log N)$) for long filters and usually offer faster convergence, too (sometimes used; very commonly used when there are already FFTs in the system)
 4. Lattice methods - are stable and converge quickly, but cost substantially more than LMS and have higher residual EMSE than many methods (very occasionally used) $O(N)$
 5. RLS - algorithms that converge quickly and are stable exist. However, they are considerably more expensive than LMS. (almost never used) $O(N)$
 6. Block RLS - (least squares) methods exist and can be pretty efficient in some cases. (occasionally used) $O(\log N)$, $O(N)$, $O(N^2)$
 7. IIR - methods are difficult to implement successfully and pose certain difficulties, but are sometimes used in some applications, for example noise cancellation of low frequency noise (very occasionally used)
 8. CMA - very useful when applicable (blind equalization); CMA is the method for blind equalizer initialization (commonly used in a few specific equalization applications) $O(N)$
- Note: In general, getting adaptive filters to work well in an application is much more challenging than, say, FFTs or IIR filters; they generally require lots of tweaking!

مثال: برنامه ارائه شده از روی سیگنال پله ورودی با طول ۵ و خروجی تولیدی از سیستم $H(z) = 1/(1-0.5z^{-1})$ پارامترهای آنرا به روش RLS تخمین می زند. همانطور که دیده می شود بعد از ۳ بار تکرار پارامترها به مقدار نهایی خود رسیده اند.

```

clear all
%input and output vectors of the assumed unknown system
n=5;u=ones(1,n);y=filter(1,[1,-.5],u);
%model structure
%y(n)=a1*y(n-1)+.....+ap*y(n-p)+b0*u(n)+.....+bq*u(n-q)
%for this example
p=1;q=0;l=p+q+1;phi=zeros(1,1);
%initial value for tet and pee
tet=zeros(1,1);pee=100*eye(1);tet=zeros(1,n);
%phi vector arrangement
for i=1:n
    for j=1:p
        if i-j<1 phi(j)=0; else phi(j)=y(i-j);end
    end
    for j=0:q
        if i-j<1 phi(p+j)=0;else phi(p+j+1)=u(i-j);end
    end
    tet(:,i)=tet ;
    %l vector calculation
    K=pee*phi/(1+phi'*pee*phi(
    %new estimate for tet
    tet=tet+1*K*(y(i)-phi'*tet(
    pee=(eye(1)-K*phi')*pee
end
plot(tet')

```



۵-۹ ویولت

- توابع و سیگنالها را می توان بصورت ترکیب خطی توابع دیگر نوشت.

$$f(t) = \sum_k a_k \phi_k(t)$$

اگر این رابطه *unique* باشد به $\phi(t)$ تابع پایه *Basis* می گویند. اگر ϕ

$$\langle \phi_k, \phi_l \rangle = \int \phi_k \phi_l dt = \begin{cases} a & k = l \\ 0 & k \neq l \end{cases}$$

باشد می توان

$$a_k = \int f(t) \phi_k dt$$

را بدست آورد. به رابطه f رابطه سنتز و به a_k رابطه تحلیلی می گویند.

- در سری فوریه متعامد نرمال است ولی در بسط تبلور متعامد نیست.
- در سری فوریه سیگنال به مجموعه ای از فرکانس اصلی و هارمونیکهای آن تجزیه می شود، به عبارت دیگر بسط تابع پایه، هارمونیکهای تابع پایه هستند و در بسط تبلور بسط تابع پایه، انتگرالهای آن هستند.
- در سری فوریه اطلاعات زمانی وجود ندارد و فقط اطلاعات فرکانسی مدل می گردد زیرا تابع پایه و هارمونیکهای آن سیگنالهای توان هستند که از $-\infty$ تا ∞ ادامه دارند لذا بسط فاقد اطلاعات زمانی است و منحصر اطلاعات فرکانسی را ارائه می دهد.
- برای سیگنالهای غیر ایستا اطلاعات فوریه مفاسب نیست از اینرو از *Short Time Fourier Transform (STFT)* استفاده می شود. مشکل *STFT* در آن است که اگر طول سیگنال بزرگ باشد اطلاعات زمانی و اگر کوچک باشد اطلاعات فرکانسی بدلیل کمبود رزولوشن مخدوش می گردند.

ویژگیهای ویولت

- در بسط ویولت، تابع بسط یک سیگنال انرژی است که بخش مهم انرژی آن در محدوده کوچک زمانی متمرکز است. این شرط از آنروست که قابلیت ارائه همزمان مشخصات زمانی و فرکانسی سیگنال را ارائه کند. بنابراین برای سیگنالهای غیر ایستا و گذرا بسیار مناسب است در حالیکه سری فوریه برای سیگنالهای ایستا و با مشخصات ثابت بسیار مفید است. از این نظر ویولت مشکل *STFT* را حل می کند زیر چند رزولوشنی *Multi resolution* است. مشخصات زمانی که $f(t)$ و مشخصات فرکانسی که $F(\omega)$ عرضه می دارند. شکل تابع ویولت و محدودیتهای آن این امر را نشان می دهد.
- تبدیل ویولت، تبدیل دو بعدی است که یک بعد آن زمان (برای سیگنال در بازه ی زمانی) و بعدی دیگر آن رزولوشن فرکانسی یا اشل است برای این منظور برای تابع ویولت، تابع انرژی تعریف می گردد تا بتواند اطلاعات زمانی سیگنال را ارائه کند.
- روش محاسباتی سریع برای آن وجود دارد و برای *DWT* مقدار $O(N)$ است

بسط ویولت گسسته

- در بسط ویولت گسسته از دو تابع اشل ϕ که بنام ویولت پدر خوانده می شود و تابع ویولت ψ ویولت مادر استفاده می شود.
- تابع اشل *scale function* یا **ویولت پدر**: منحصر به یک تابع خاص نیست و هر تابعی که شرایط را داشته باشد کاندیدی برای آن است. این تابع دوبعدی است. یک بعد آن k زمانی و بعد دوم آن j مرتبط با درجه تفکیک فرکانسی است.

$$f(t) = \sum_k \sum_j a_{j,k} \phi_{j,k}(t)$$

- توابع زیادی مانند *Haar*, *Daubechies*، مثلثی و غیره تعریف شده و مورد استفاده قرار گرفته اند. این توابع، توابع انرژی هستند و بعضی از آنها بصورت پنجره هستند و فقط در محدوده خاص زمانی مقدار دارند.
- این توابع آزاد نیستند و بر اساس تعاریف دارای محدودیتهای ذیل در موجک ۲ باندی می باشد. برای M باندی جای عدد ۲ در روابط با

M عوض می شود.

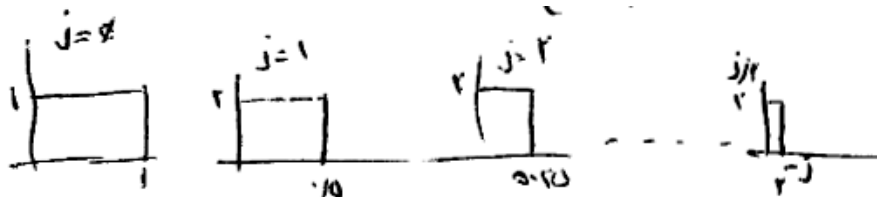
- شرط ۱ از اینقرار است:

$$\varphi_{j,k}(t) = 2^{j/2} \varphi(2^j t - k) = 2^{j/2} \varphi[2^j (t - \frac{k}{2^j})]$$

که

$$f(t) = \sum_k \sum_j c_{j,k} \varphi_{j,k}(t)$$

می باشد. در این رابطه k شیفت زمانی و j درجه اشل است. هرچه j بزرگتر شود پریود موج کوچکتر می شود و رزولوشن بهتر می گردد. شکل تابع اشل مربوط به تابع $Haar$ را نشان می دهد.



با ضرب $\sqrt{2}$ انرژی توابع مختلف بسط ثابت نگه داشته می شود.

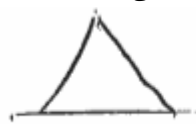
- شرط ۲: برای اینکه $\varphi(t)$ برای کار قابل قبول باشد. به این معنی که بسط $\varphi(t)$ حاوی اطلاعات $\varphi_{j-1}(t)$ باشد،

$$\varphi(t) = \sum_n h(n) \sqrt{2} \varphi(2t - n)$$

$$h = [\frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}}]$$

و برای اشل مثلی

تعریف می شود. برای اشل $Haar$.



$$[\frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}}]$$

و به همین ترتیب برای بقیه است.

$$\sum h(n) = \sqrt{2}$$

این تابع باید شرایطی را تامین کند که به این ترتیب بسط $\varphi(2t)$ حاوی اطلاعات بسط جملات $\varphi(t)$ است.

- با افزایش j در تابع $Haar$ و اکثر توابع اشل دیگر، بسط تابع به ضربه نزدیک می شود. و منجر به

$$a_{j,k} = \int_{-\infty}^{\infty} f(t) \psi_{j,k}(t) dt = f(t) \delta(t - k)$$

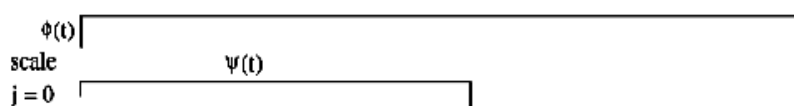
می شود، زیرا تابع φ به ضربه میل می کنند. این بدان معنی است که ضرایب ویولت همان نمونه های سیگنال باشند.

تابع ویولت ψ wavelet function

- **ویولت مادر**: به جای بسط دادن سیگنال با جملات φ ، تابع ویولت Ψ تعریف می گردد. که

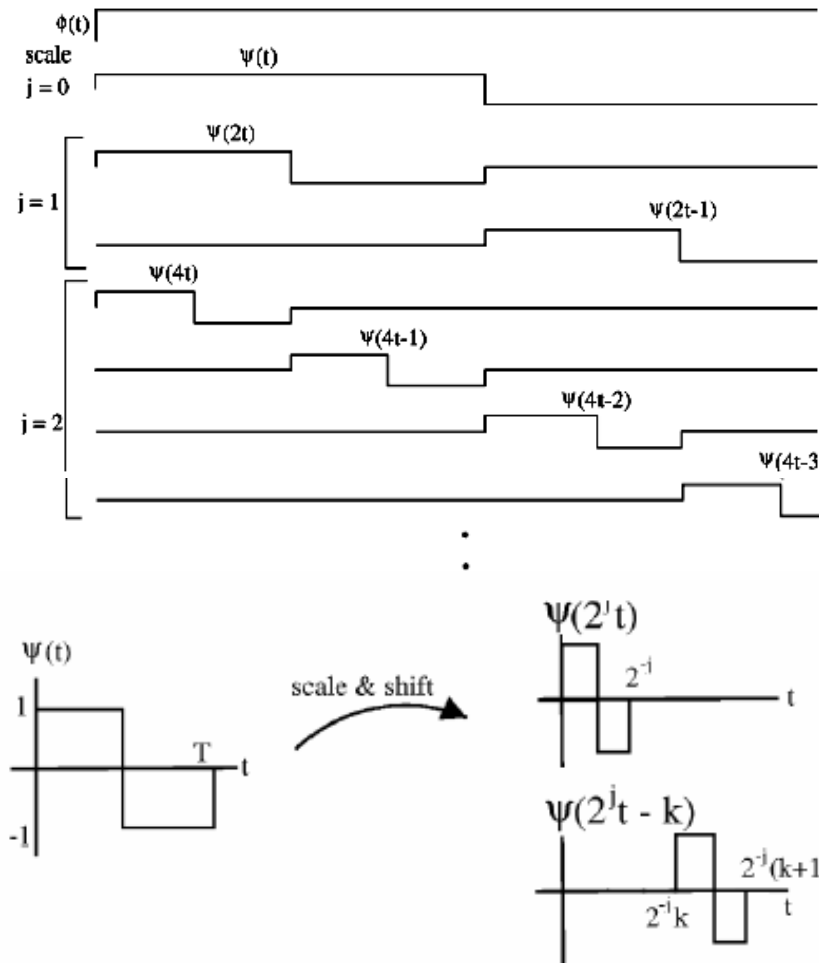
$$\psi(t) = \sum_n h_1(n) \sqrt{2} \varphi(2t - n), \quad h_1(n) = (-1)^n h(N-1-n), \quad \psi_{j,k} = 2^{j/2} \psi(2^j t - k)$$

است. برای تابع $Haar$ ، ویولت مطابق شکل از تابع اشل بدست می آید.



این بسط دارای این مزیت است که تعداد زیادی از پارامترها کوچک می گردند و قابل حذف شدن می شوند برای مثال اگر تابع اشل $Haar$ استفاده شود و تابع ویولت آن تفاضل میانگین دو نیمه هر خانه در حالیکه تابع اشل میانگین آن خانه را بدست می دهد به این

ترتیب توابع پایه شیف و اشل شده تابع اصلی هستند. به شکل توجه کنید.



- وقتی تابع f را به Ψ مدل کنیم دنباله j ها تا بی نهایت ادامه می یابد برای محدود کردن آن پائین ترین j را با تابع اشل نوشته تا دنباله محدود گردد. به این ترتیب

$$f(t) = \sum_{k=-\infty}^{\infty} c_{j_0}(k) \varphi_k(t) + \sum_{k=-\infty}^{\infty} \sum_{j=j_0}^{j_m} d_j(k) \psi_{j,k}(t)$$

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) = 2^{j/2} \psi[2^j (t - \frac{k}{2^j})]$$

$$\varphi_{j,k}(t) = 2^{j/2} \varphi(2^j t - k) = 2^{j/2} \varphi[2^j (t - \frac{k}{2^j})]$$

در این رابطه j_m بالاترین رزولوشن است که در آن $c_{jm}(k)$ همان نمونه های سیگنال است.

- اگر φ و Ψ متعامد باشند آنگاه

$$d_j(k) = \int f(t) \psi_{j,k}(t) dt \quad c_{j_0}(k) = \int f(t) \varphi_{j_0,k}(t) dt$$

بدست می آید. در بعضی حالات شرط کامل متعامد بودن را حذف می کنند در این حالات انعطاف بیشتری حاصل می شود.

- برای محاسبه DWT نیاز به اصل اطلاعات تابع اشل یا ویولت نیست بلکه تابع $h(n)$ کفایت می کند. در این حالت تابع سنتز

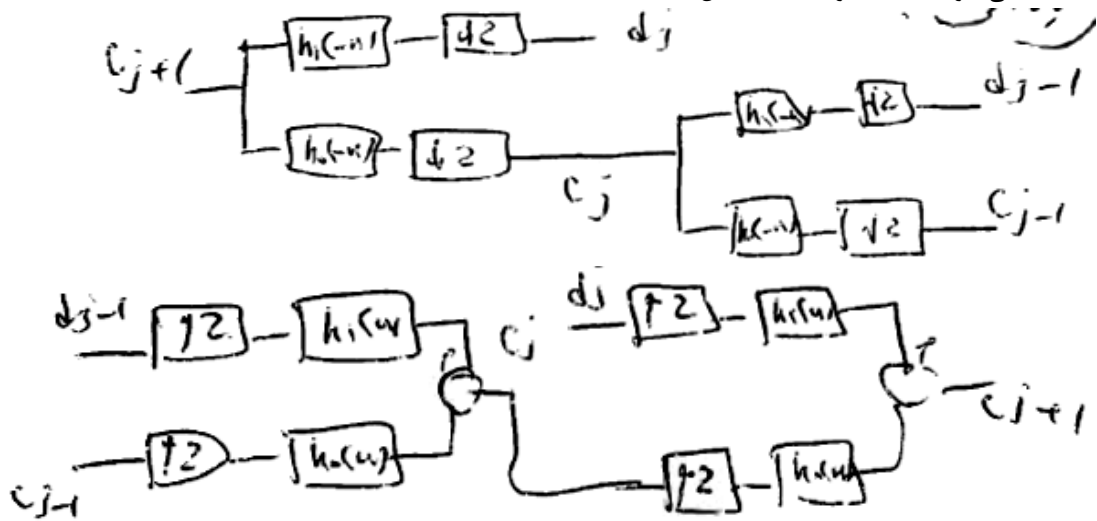
$$c_{j+1}(k) = \sum_m c_j(m) h(k - 2m) + \sum_m d_j(m) h_1(k - 2m)$$

است که از ضرایب تابع، c_{jm} (سیگنال نمونه برداری شده) بدست می آید.

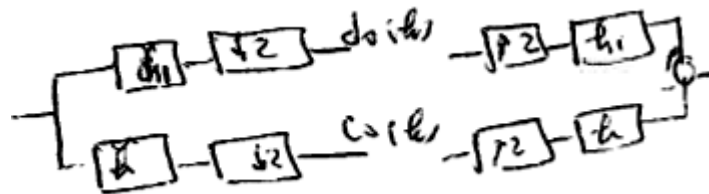
برای تحلیل از رابطه

$$c_j(k) = \sum_m h(m - 2k) c_{j+1}(m) \quad d_j(k) = \sum_m h_1(m - 2k) c_{j+1}(m)$$

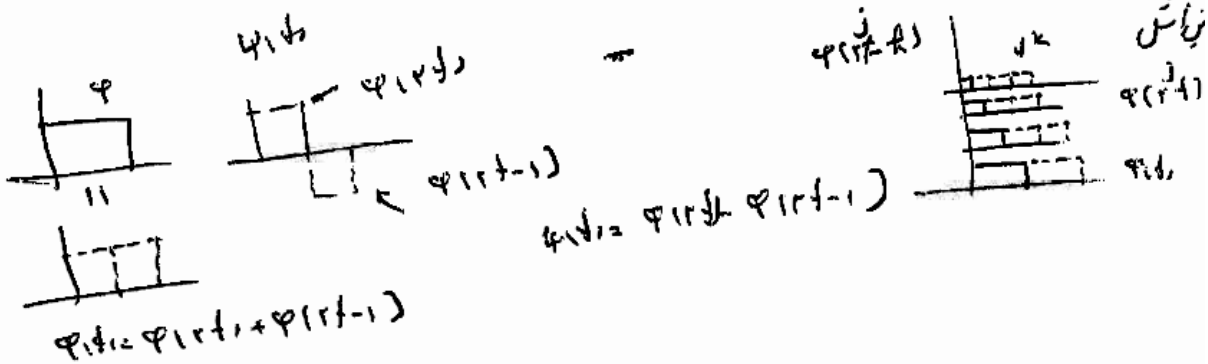
استفاده می شود که cjm نمونه های سیگنال است.



- محاسبات DWT در محدوده $O(N)$ است.
- مکمل متعامد V_j در V_{j+1} به نام W_j خوانده می شود. به این ترتیب تمام اعضا V_j نسبت به اعضا W_j متعامد هستند.



- افزایش اشل : آنچه در نتیجه افزایش اشل اتفاق می افتد از اینقرار است.

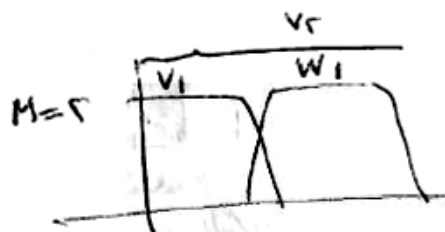


موضوعات دیگر مربوط به ویولت

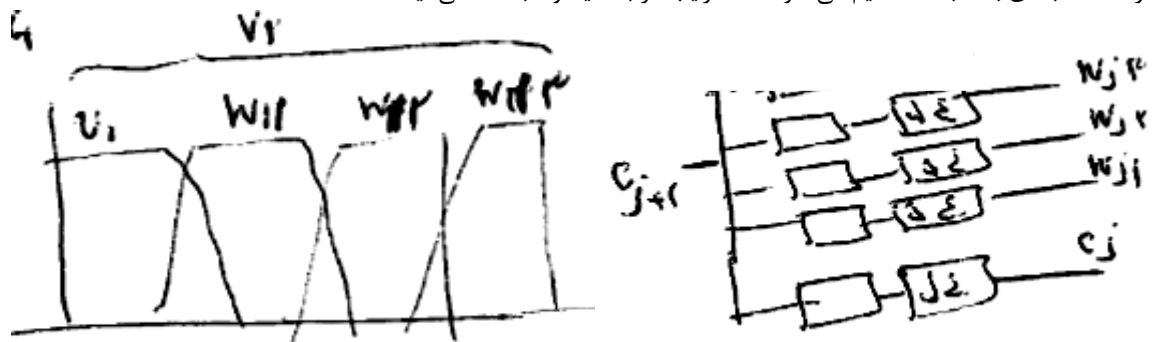
- حالت M بانندی: در بحثهای قبلی از حالت ۲ بانندی استفاده شد. در حالت M -band روابط اینگونه تغییر می کند.

$$\varphi(t) = \sum_n h(n) \sqrt{M} \varphi(Mt - n) \quad \sum_n h(n) = \sqrt{M} \quad \sum_n |h(n)|^2 = 1$$

در حالت ۲ بانندی طیف سیگنال از صفر تا π مطابق شکل به ۲ بانک مساوی تقسیم می گردد. این تقسیم بندی با نصف شدن $V1$ و همینطور متوالیا ادامه می یابد. به این ترتیب است که رزولوشن در فرکانسهای بالا کم و در فرکانسهای پایین زیاد می گردد.



ولی در حالت ۴ بانندی به ۴ بانک تقسیم می شود. که ضرایب هر بانند اینگونه بدست می آید



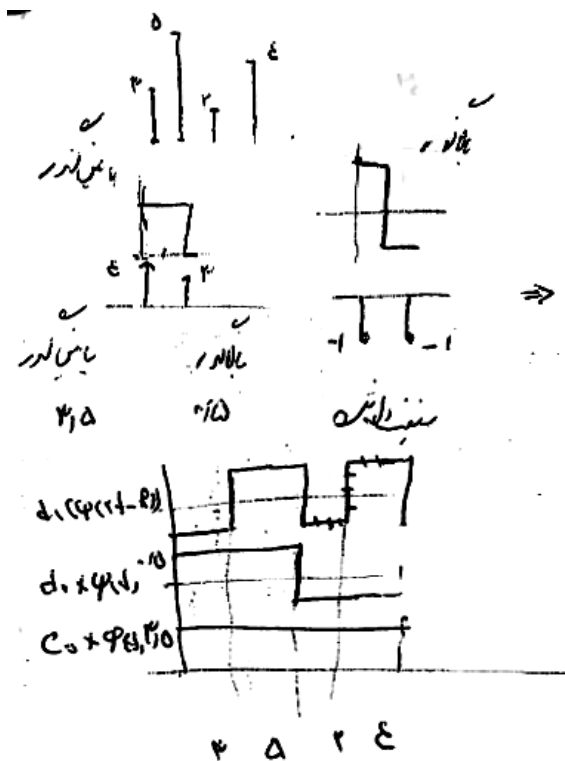
• دو متعامدی Biorthogonality

در حالت متعامد طول Ψ , ϕ باید برابر باشد و طول زوج باشد. با *biorthogonal* امکان داشتن ϕ و Ψ متقارن وجود دارد. در این حالت رابطه پارسوال دیگر بقرار نیست

• ویولت پیوسته

$$F(a,b) = \int f(t) \psi\left(\frac{t-a}{b}\right) dt \quad f(t) = \int F(a,b) \psi\left(\frac{t-a}{b}\right) da db$$

مثالی:



فیلترهای Haar

| ψ | ϕ | ψ | ϕ | C_1 |
|----------|----------|----------|----------|-------|
| ψ | ϕ | ψ | ϕ | d_1 |
| ψ_0 | ϕ_0 | ψ_0 | ϕ_0 | d_0 |
| ψ_0 | ϕ_0 | ψ_0 | ϕ_0 | d_0 |