مقدمه

عبارت فیلتر معمولاً به دستگاهی، سخت افزاری یا نرم افزاری، اطلاق می شود که برای بازیابی اطلاعات مفید در یک سیگنال نویزی به کار می رود. نویز یک سیگنال ناخواسته است که اطلاعات موردنظر ما را تحت تأثیر قرار می دهد و در اثر شرایط متفاوتی تولید می شود. به عنوان مثال سیگنال ممکن است توسط یک سنسور در محیطی نویزی خوانده شود یا شاید سیگنال در طول انتقال در کانال مخابراتی دچار اختلال گردد.

فیلتر به طور کلی سه کاربر دارد:

۱-فیلتر کردن[1]:

بازیابی سیگنال با دقت خواسته شده در زمان t با توجه به اطلاعات موجود در زمان t

۲-یکنواخت ساختن[2]:

در این کاربرد اطلاعات مورد نظر با دقت خواسته شده در زمان t وجود ندارد ولی به کمک داده هایی که در زمان های بعد از t بدست می آید، سیگنال مورد نظر بازیابی می شود. به همین دلیل برای یکنواخت ساختن باید از تأخیر استفاده کرد.

۳-پیش بینی[3]:

در این مورد هدف بدست آوردن سیگنال در زمان $t+\tau$ در آینده $(\tau > 0)$، بوسیله اطلاعات موجود در زمان t می باشد.

فیلترها را می توان به دو دسته تقسیم بندی نمود:

---

[1] -Filtering
[2] -Smoothing
[3] -Prediction

-خطی[۴]

-غیرخطی

یک فیلتر را خطی می نامند هرگاه خروجی آن تابعی خطی از ورودی باشد. در رهیافت آماری برای فیلتر خطی، ما به پارامترهای آماری، مانند میانگین و یا تابع همبستگی[۵]، سیگنال و نویز احتیاج داریم. یک راه کاربردی برای بهبود فیلتر کردن، حداقل نمودن مقدار میانگین مربع خطایی[۶] که از کم کردن پاسخ مورد نظر و خروجی فیلتر بدست می آید، می باشد. برای ورودی های ساکن[۷]، راه حل مناسب فیلتر Wiener می باشد. در این حالت منحنی MSE برحسب پارامترهای قابل تنظیم فیلتر سطح اجرایی خطا[۸] نامیده می شود. نقطه حداقل در این نمودار، ضرایب بهینه را مشخص می کند.

فیلتر Wiener در مواقعی که سیگنال یا نویز غیرساکن[۹] می باشند، غیرقابل استفاده است. در این شرایط فیلتر بهینه متغیر با زمان فرض می شود که از معروف ترین این نمونه می توان به فیلتر Kalman اشاره کرد.

تئوری فیلترهای وفقی مانند Wiener یا Kalman، در حوزه پیوسته همچون گسسته بحث شده اند ولی در عمل بدلیل حضور کامپیوتر و پردازشگرهای دیجیتال[۱۰] در حوزه گسسته کارایی بیشتری دارند. در فیلترهای وفقی، معمولاً از یک فیلتر دیجیتال به همراه

[4] -Linear
[5] -Carrelation Function
[6] -Mean Square Error
[7] -Stationary
[8] -Error Performance Surface
[9] -Non Stationary
[10] -DSP

یک الگوریتم وفقی استفاده می شود که ضرایب[۱۱] فیلتر دیجیتال توسط الگوریتم موجود تعیین می شود.

در زیر چند کاربرد فیلترهای وفقی را نام می بریم:

۱-در مهندسی پزشکی و دستگاه هایی مانند MRI، EEG و ECG

۲-مخابرات دیجیتال

۳-حذف اکو در تلفن[۱۲]

۴-سیستم رادار[۱۳]

۵-سیستم هدایت[۱۴]

این پایان نامه مشتمل بر چهار فصل می باشد. در فصل اول در باره فیلترهای دیجیتال بحث های مختصر و پایه ای شده و خواننده را برای درک مفهوم فیلتر وفقی آماده می سازد. فصل دوم به دو بخش تقسیم شده است. در بخش اول ریاضیات مورد نیاز برای فیلتر وفقی آورده شده است و در بخش دوم به معرفی فیلتر وفقی پرداخته شده و در باره انواع الگوریتم های آن بحث شده است. فصل سوم راجع به قابلیت های نرم افزار تخصصی MATLAB در زمینه فیلترکردن و فیلترهای وفقی می باشد. و در فصل آخر تعدادی از کاربردهای[۱۵] فیلترهای وفقی را مرور می کنیم.

[11] -Weights
[12] -echo cancellation
[13] -radar system
[14] -Navigation System
[15] -Application

فصل اول

فیلترها

۱-۱)اصولاً فیلتر به دستگاه یا وسیله ای گفته می شود که برای جدا کردن ۱ باند فرکانسی از باندهای دیگر و یا حذف نویز یا سیگنالهای مزاحم استفاده می شود.

فیلترها به طور عمده ۲ کاربرد دارند

۱-**جداسازی یا تفکیک سیگنال**[۱۶]: زمانی استفاده می شود که سیگنال با استفاده از نویز – تداخل و سیگنالهای دیگر آلوده شود مثال: اندازه گیری فعالیت الکتریکی قلب کودک[۱۷] در زمان بودن در رحم که سیگنال خام یا اصلی در اثر صدای ضربان قلب مادر یا تنفس او خراب می شود لذا بایستی سیگنال اصلی از بقیه سیگنالها تفکیک شود.

۲-**بازیابی سیگنال**[۱۸]: وقتی که یک سیگنال در مسیر خاصی مشوش یا خراب[۱۹] شود. مثال:۱ ضبط ساده که از وسایل آماتور ساخته شده است ممکن است فیلتر شود تا صدای بهتری را نمایش دهد.

۱-۲ هر فیلتر دارای ۳ پاسخ اصلی است

۱) پاسخ پله ۲) پاسخ ضربه ۳) پاسخ فرکانسی

هر ۳ این پاسخ ها دارای اطلاعات یکسان ولی در فرمتهای مختلف می باشند.

۱-۳) ۲ روش در طراحی فیلترها (عموماً دیجیتال) وجود دارد

[16] -Signal Sepration
[17] -EKG
[18] -restoration
[19] -Corrupt

۱) روش کانولوشن[۲۰] سیگنال ورودی با پاسخ ضربه فیلتر دیجیتال (روش کانولوشن)

۲) روش طراحی فیلتر دیجیتال با روش بازگشتی[۲۱] (روش بازگشتی)

به فیلترهایی که به روش کانولوشن طراحی می شوند اصطلاحاً فیلتر FIR[۲۲] یا فیلترهای دارای پاسخ ضربه محدود می گویند و به فیلترهایی که به روش بازگشتی طراحی می شوند اصطلاحاً فیلتر IIR[۲۳] یا فیلترهای دارای پاسخ ضربه نامحدود گویند.

۴-۱)پارامترهای حوزه زمان و فرکانس در فیلترها:

۱-۴-۱)پارامترهای حوزه زمان

۱-زمان رشد و نمو: مدت زمانی است که طول می کشد تا پاسخ پله فیلتر از ٪۱۰ به ٪۹۰ مقدار نهایی برسد هرچه زمان فوق کمتر باشد سرعت فیلتر بیشتر است.

۲-بالازدگی[۲۴] : مقدار بالازدگی در پاسخ پله را گویند که معیاری از پایداری[۲۵] سیستم است هرچه بالازدگی کمتر باشد سیستم پایدارتر است.

۲-۴-۱) پارامتر حوزه فرکانسی:

[20] -Convolution
[21] -recursion
[22] -Finite Impulse response
[23] -Infinite Impule response
[24] -Overshoot
[25]– rise time

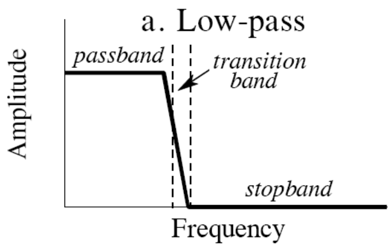
۱-باند عبور[26] ۲-باند قطع[27] ۳-باند گذر[28]

باند عبور: باندی که در آن باند فیلتر سیگنالها را عبور می دهد.

باند قطع: باندی که در آن باند فیلتر سیگنالها را عبور نمی دهد.

باند گذر: باندی که بین باند عبور و باند قطع است.

مطابق شکل (۱-۱)



باند عبور $\leftarrow 0 < w < w_1$

باند قطع $\leftarrow w_2 < w < \infty$

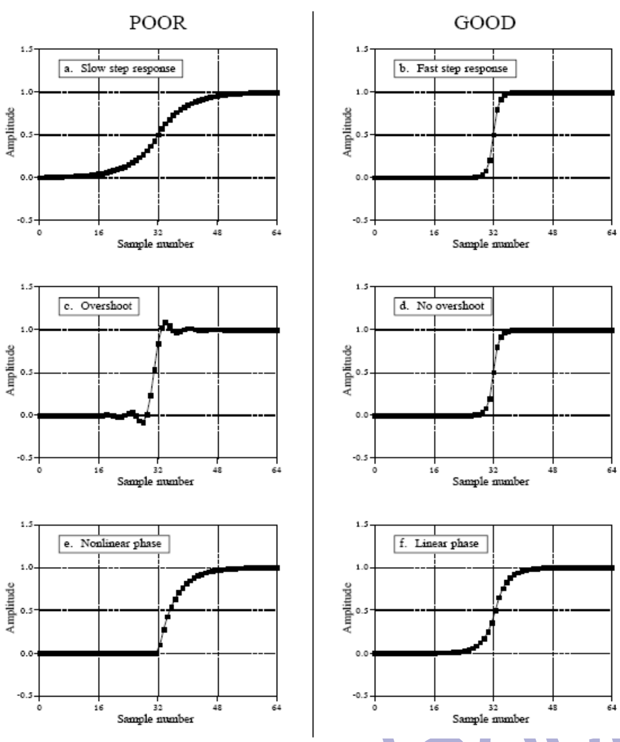
۴-شیب[29]: شیب یا تندی در حوزه فرکانس را می نامند هرچه شیب فوق بیشتر باشد ناحیه گذر نازکتر و در نتیجه فیلتر به فیلتر ایده آل نزدیکتر است.

۵-تضعیف ناحیه قطع: هرچه تضعیف در ناحیه قطع بیشتر باشد فیلتر بهتر طراحی شده است (رنج طراحی: db ۱۰۰-۸۰)

---

[26] -Pass band
[27] -Stop band
[28] -Transition band
[29] -roll-off

شکل ۲-۱



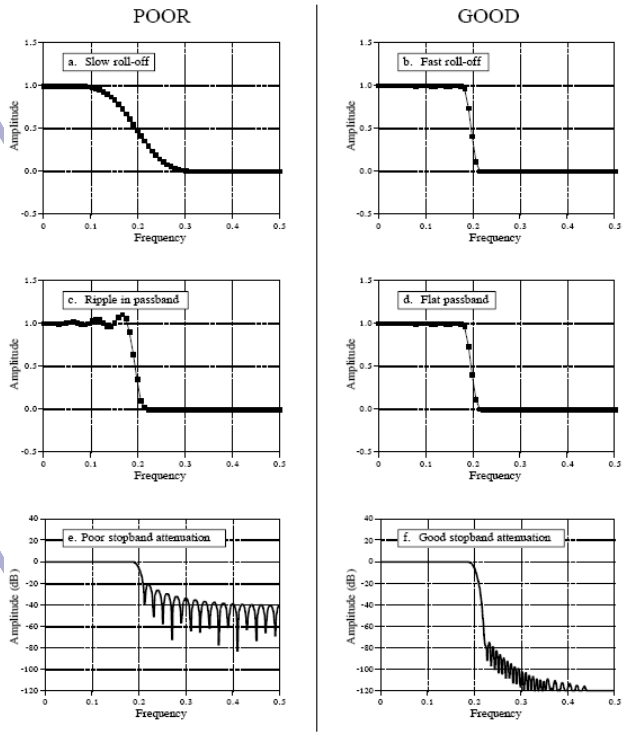
پاسخ پله برای

اندازه گیری پارامترهای حوزه زمان استفاده می شود ۳ پارامتر مهم در پاسخ پله عبارتند از:

۱-زمان نمو ۲- overshoot ۳-خطی یا غیرخطی بودن فاز

در شکل های زیر پاسخ های فرکانسی یک فیلتر پایین گذر نمایش داده شده است ۳ پارامتر مهم در پاسخ فرکانس عبارتند از ۱-roll-off ۲-ریپل باند عبور ۳- تضعیف در ناحیه قطع

شکل ۳-۱



۲-۱-روش برای

تبدیل فیلتر پایین گذر به بالاگذر وجود دارد

۲-۱-۲-معکوس طیفی[۳۰]: روش فوق شامل ۲ مرحله است

۱)تغییر علامت دادن تمام نمونه ها (در حوزه زمان)

۲) افزودن عدد (۱) به نمونه واقع در مرکز (حوزه زمان)

۲) تغییر در حوزه فرکانس: معکوس کردن طیف سیگنال از بالا به پایین[۳۱] (حوزه فرکانس)

(۳-۱-۲) معکوس یا عکس طیفی[۳۲]: روش فوق نیز شامل ۲ مرحله است

[30] -spectral-inversion
[31] -flipped top-for-bottom
[32] -spectral-reversal

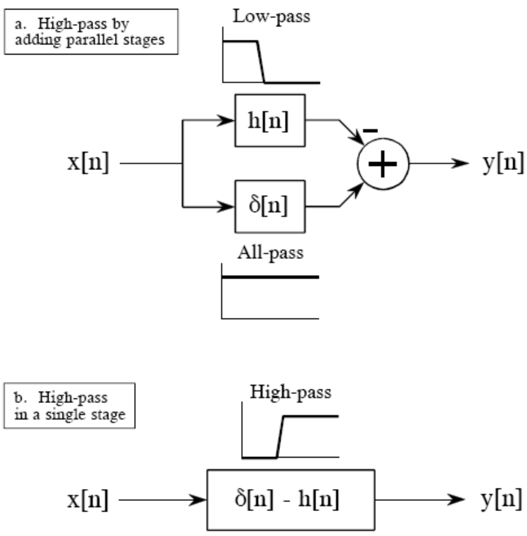
۱-تغییر علامت دادن هر نمونه: ضرب کردن فیلتر کرنل با ۱ موج سینوس با فرکانس 0/5 (حوزه زمان)

۲-در حوزه فرکانس نیز معادل است: معکوس کردن طیف از چپ به راست[۳۳]

شمای تبدیل فیلترها به یکدیگر

شکل۴-۱-بیانگر معکوس طیفی است در شکل اول این بخش نشان داده شده است سیگنال ورودی برای

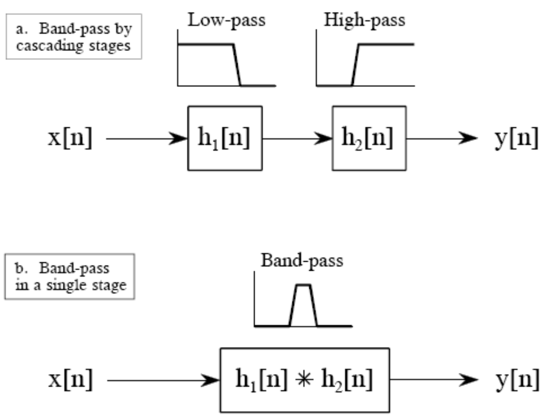
(تبدیل فیلتر پایین گذر به بالاگذر)



[33] -flipped left-for-rig

۲سیستم موازی درخواست شده و ۲ سیستم موازی دارای پاسخ ضربه h[n] و δ[n]می باشند همانطور که در شکل دو نمایش داده شده است سیستم ترکیبی دارای پاسخ ضربه h[n] و δ[n]است این به این معنی است که پاسخ فرکانس سیستم ترکیب شده عکس پاسخ فرکانسی h[n] است.

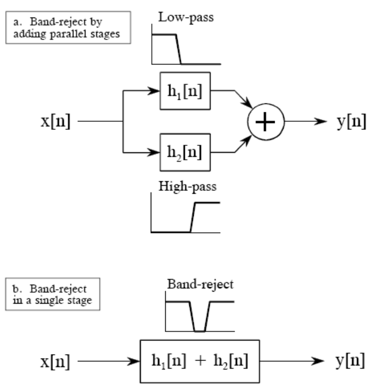
(ساخت فیلتر میان گذر از پایین گذر و بالاگذر): این شکل طراحی ۱ فیلتر میان گذر است که در شکل نمایش داده شده است.



۱ فیلتر میان

گذر با کسکد کردن 2 فیلتر بالاگذر و پایین گذر ساخته می شود این شکل می تواند در ۱ مرحله نمایش داده شود پاسخ ضربه سیستم کل مساوی است با کانولوشن پاسخ ضربه ۲ سیستم پایین گذر و بالاگذر

شکل ۶-۱ طراحی فیلتر میان گذر: همانطور که در شکل نمایش داده شده است ۱ فیلتر میان نگذر از ترکیب موازی ۱ فیلتر پایین گذر و بالاگذر به دست می آید شکل دوم همین بخش نشان می دهد که در نتیجه پاسخ ضربه سیستم کل برابر است با جمع پاسخ ضربه سیستم پایین گذر و بالاگذر



۳-۱) طبقه بندی فیلترها[۳۴]

۱-۳-۱) فیلترهای قابل ساخت و طراحی با روش بازگشتی (ضرایب بازگشتی)

۱-فیلتر تک قطبی[۳۵]

۲-فیلتر چیپشف[۳۶]

۳-فیلتر ایریتیو[۳۷]

[34] -Filter classification
[35] -Single Pole
[36] -chebyshev

۳-۳-۱) فیلترهای قابل ساخت، طراحی با روش کانولوشن (پاسخ ضربه)

۱-فیلتر میانگیر[۳۸]

۲-فیلتر پنجره ای سینک[۳۹]

۴-۱)فیلتر میانگیر[۴۰]

همانطور که از نام فوق پیداست با میانگین گرفتن تعدادی از نمونه های سیگنال ورودی برای تولید سیگنال خروجی بدست می آید.

و طبق معادله[۴۱] روبرو محاسبه می شود.

$$y[i]=\frac{1}{M}\sum_{j=0}^{M-1}x[i+j]$$

که به x[] سیگنال ورودی و y[] سیگنال خروجی می گویند.

به عنوان مثال اگر M=۵ باشد $y[80]=\frac{x[80]+x[81]+x[82]+x[83]+x[84]}{5}$

و همچنین می توان کرانهای سری فوق را تغییر داد

$$j=0\rightarrow M-1\Rightarrow j=-(\frac{M-1}{2})\rightarrow(\frac{M-1}{2})$$

و در نتیجه فرمول ۱ به شکل روبرو می آید.

$$y[i]=\frac{1}{M}\sum_{j=-(\frac{M-1}{2})}^{j=M-\frac{1}{2}}x[i+j]$$

---

37 -Irative
38 -Moving-Average
39 -Windowed-Sinc
40 -Moving average filter
41 -equation

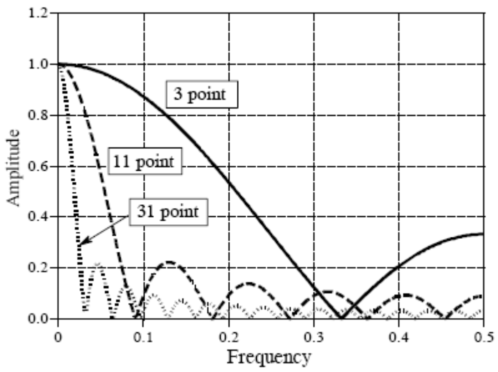
۱-۴-۱-مزایای فیلتر میانگیر:

این فیلترباعث کاهش نویز رندوم می شود (خوب) ولی باعث کاهش تیزی گوشه ها می گردد.

مقدار کاهش دامنه نویز مساوی است با ریشه دوم شماره نمونه ها $(\sqrt{M})$

۲-۴-۱-پاسخ فرکانس فیلتر میانگیر:

شکل ۷-۱



شکل: این شکل نشان دهنده پاسخ فرکانسی فیلتر میانگیر است $H(f)=\frac{\mathrm{Sin}(\pi fm)}{M\,\mathrm{Sin}(\pi f)}$

ویژگیهای فیلتر خاص: فیلتر میانگیر ۱ فیلتر پایین گذر نامطلوب با roll-off کند و تضعیف ناحیه قطع نامطلوب می باشد. دارای شیب آرام در حوزه فرکانس و تضعیف ناحیه قطع نامطلوب باشد همچنین فیلتر فوق نمی تواند ۱ باند فرکانسی را از باندهای دیگر جدا نماید فیلتر فوق در حوزه زمان ۱ فیلتر نرم کننده است ولی در حوزه فرکانس فیلتر پایین گذر بد است.

۳-۴-۱-فیلتر پنجره ای[42]

فیلتر پنجره ای ۱ فیلتر پایین گذر ایده ال است باند عبور کاملاً صاف است و تضعیف در ناحیه قطع نامحدود می باشد.

$$h(i)=\frac{\mathrm{Din}(۲\pi fci)}{\pi.i}$$

کانولوشن سیگنال ورودی با فیلتر کرنل ۱ فیلتر پایین گذر کامل را ایجاد می کند مشکل اینجاست که تابع ادامه می یابد. بدون خاصیت میراشوندگی. برای حل این مشکل ما ۲ کار انجام می دهیم. آنرا تا M+۱ نمونه قطع می نماییم. تمام نمونه ها صفر می شوند بعد سیگنال را به سمت راست شیفت می دهیم و نمونه های منفی را حذف می نماییم در نتیجه پاسخ فرکانسی از حالت ایده آل خارج شده و ریپل هایی در باند عبور قطع مشاهده می شود

۱-۵ فیلتر های پنجره ای معروف :

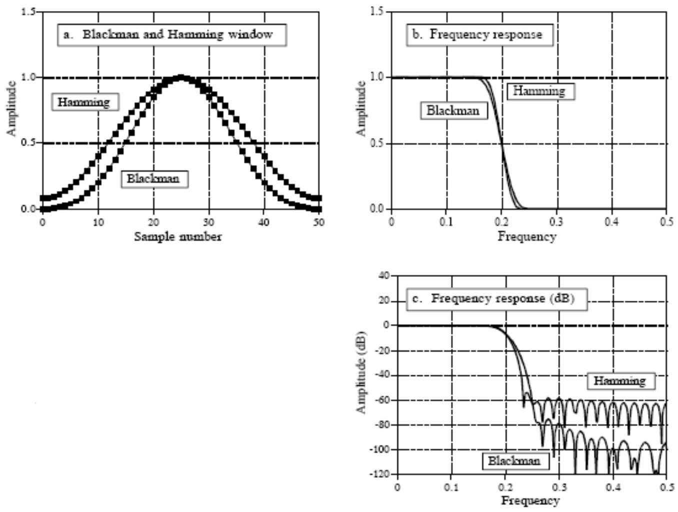
[42] -Windowed-Sinc

فیلتر همینگ با معادله روبرو: $w[n]=۰/۵۴-۰/۴۶\mathrm{Cos}\left(\frac{۲\pi n}{m}\right)$

فیلتر بلک من با معادله روبرو: $w[n]=۰/۴۲-۰/۵\mathrm{Cos}\left(\frac{۲\pi n}{m}\right)+۰/۸\mathrm{Cos}\left(\frac{۴\pi n}{m}\right)$

این شکل بیانگر مشخصات فیلترهای Black-man و Hamming می باشد که مشخصات آنها در شکل ۸-۱ نمایش داده شده است همانطوریکه در شکل می بینید فیلتر Hamming ۲۰٪ از لحاظ roll-off سریعتر از Black-man است گرچه فیلتر Black-man تضعیف ناحیه قطع بهتری نسبت به Hamming دارد.

شکل ۸-۱



توضیح

حاتی در ارتباط با تبدیل Z و فیلتر بازگشتی

۵-۱) تبدیل لاپلاس[۴۳] ۱ سیگنال از رابطه زیر محاسبه می شود.

۱) $X(s) = \int_{-\infty}^{\infty} x(t)e^{-ST} dT$

که به ترتیب نمایش سیگنال در حوزه Z و S می باشد.

و می دانیم که S یک کمیت مختلط[۴۴] است.

۲) $S = \sigma + jw$

با جایگذاری ۲ در ۱ خواهیم داشت

$$x(\sigma,\omega) = \int_{-\infty}^{\infty} x(t)e^{-jwt} e^{-\sigma t}dt$$

این عبارت نشان می دهد که هر نقطه در صحنه s با ۲ پارامتر σ و w نمایش داده می شود که به σ قسمت حقیقی[۴۵] و به ω (قسمت موهومی)[۴۶] می گویند برای پیدا کردن (قسمت حقیقی) کافی است سیگنال حوزه زمان را در ۱ موج کسینوسی با فرکانس ω ضرب کنیم و دامنه آن مطابق با σ کاهش می یابد برای محاسبه ω نیز از روش مشابه بالا استفاده می شود.

تبدیل لاپلاس می تواند در ۳ قدم به تبدیل z تبدیل شود تبدیل سیگنال از پیوسته به گسسته: این کار با جابجایی و تعویض متغیر t با n امکان پذیر است و دیگری تبدیل انتگرال به سیگما

$$x(\sigma,\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-\sigma n}e^{-jwn}$$

---
43 -Laplace Transform
44 -Complex
45 -Real-Part
46 -Imaginary-Part

توجه کنید $x(\sigma,\omega)$ یک کمیت پیوسته است نه گسسته

دومین قدم بازنویسی ترم و جمله مجهول می باشد

$$y[n]=e^{-\sigma n}\xrightarrow{\sigma=\mathrm{Lnr}}y[n]=r^{-n}$$

بعد فرمول تبدیل Z به صورت زیر بازنویسی می شود

$$x(z)=\sum_{n=-\infty}^{\infty}x[n]z^{-n}$$

۱-۵-۱) تفاوتهای تبدیل لاپلاس و Z:

هر نقطه در صفحه S با ۲ پارامتر σ (متغیر کاهش در راستای محور افقی) و W (متغیر فرکانس در راستای محور عمودی) مشخص می شود.

ولی در تبدیل Z هر نقطه روی ۱ دایره با (r) خاص (فاصله از مبدأ صفحه Z) و فاز ϕ یا زاویه نسبت به محور حقیقی سنجیده می شود.

۱-۵-۲)آنالیز سیستمهای بازگشتی: سیستمهای بازگشتی با فرمول

$$y[n]=a_0x[n]+a_1x[n-1]+a_2x[n-2]+\ldots b_1y[n-1]+b_2y[n-2]+\ldots$$

که در آن []x به عنوان ورودی و []y به عنوان خروجی می باشد و a و b ضرایب بازگشتی نامیده می شوند.

برای بدست آوردن تابع فیلتر کافی است تابع تبدیل سیستم را بدست بیاوریم.

$$H(z)=\frac{Y(z)}{X(z)}$$

$$H(z)=\frac{a_0+a_1z^{-1}+a_2z^{-2}+\ldots}{1-b_1z^{-1}-b_2z^{-2}-b_3z^{-3}+\ldots}$$

۱-۶-۱)تفاوت فیلترهای آنالوگ و دیجیتال.

فرضیه را با مثال شروع می کنیم ما می خواهیم ۱ فیلتر چبیشف (۶ قطب) را با ریپل حدود ٪۶ بسازیم.[۴۷]

بر طبق داده ها می دانیم که این فیلتر با سه عدد OPAMP، ۱۲ عدد ترانزیستور، ۶ عدد خازن ساخته می شود در دید دیجیتال فیلتر پنجره ای سینک[۴۸] آماده برای رقابت می باشد سیگنال آنالوگ بایستی با فرکانس ۱۰KHz نمونه برداری شده و فرکانس قطع آن نیز در مقیاس ۰/۱ تنظیم گردد طول فیلتر پنجره ای ۱۲۹ نمونه می باشد و کاهش شیب از ٪۹۰ به ٪۱۰ شبیه فیلتر آنالوگ صورت گیرد

مقایسه ۲ فیلتر فوق:

۱-فیلتر آنالوگ در باند عبور دارای (٪۶) ریپل است ولی فیلتر دیجیتال کاملاً صاف است مقدار صافی باند عبور در فیلتر آنالوگ بستگی به درستی و طراحی دقیق مقاومتها و خازنها می باشد.

۲-در فیلتر دیجیتال پاسخ پله حالت متقارن دارد (بین ۲ بخش بالایی و پایینی) و در نتیجه دارای فاز خطی[۴۹] است و در فیلتر آنالوگ برعکس علاوه بر داشتن فاز غیرخطی دارای پاسخ های متقارن نیز نمی باشد و همچنین فیلترهای آنالوگ در حدود ٪۲۰

47 -Flatness
48 -windowed_ sinc
49 -Linearphase

اورشوت در پاسخ های خود و فیلترهای دیجیتال در حدود ۱۰٪ اورشوت دارند فیلترهای آنالوگ سریع و فیلترهای دیجیتال[50] نرم و قابل استفاده ترند.

۲-۶-۱) تفاوت میان فیلتر پنجره ای سینک و چبیشف:

هر ۲ فیلتر دارای ویژگی زیر هستند که ۱ باند فرکانسی را از باندهای دیگر جدا می سازد فیلتر پنجره ها سینک ۱ فیلتر FIR است که از طریق کانولوشن طراحی می شود و دیگری چبیشف است که ۱ فیلتر IIR است که از طریق ضرایب بازگشتی طراحی می شود.

فیلتر IIR فوق حدود ۰/۵٪ ریپل در باند عبور دارد در حالتی که فیلتر پنجره ای سینک صاف می باشد و فیلتر پنجره ای سینک دارای تضعیف بهتری در ناحیه قطع خود می باشد.

۳-۶-۱)-روش Over-Lap Add Method:

وقتی بخواهیم ۱ سیگنال طولانی را فیلتر کنیم روش های متعددی وجود دارد یکی از آن روشها روش Over-Lap Add Method است روش Over-Lap-add-method برروی این تئوری بنا شده که سیگنالی که باید فیلتر شود به بخش های مختلف تقسیم می کنیم پس هر کدام از این قسمت ها را تجزیه و تحلیل کرده و فیلتر می کنیم بعد از فیلتر کردن تمام سیگنال ها آنها را با هم جمع می نماییم و هر بخش با کانولوشن پاسخ ضربه با فیلتر اصلی خروجی اش محاسبه می شود.

[50] -Digital

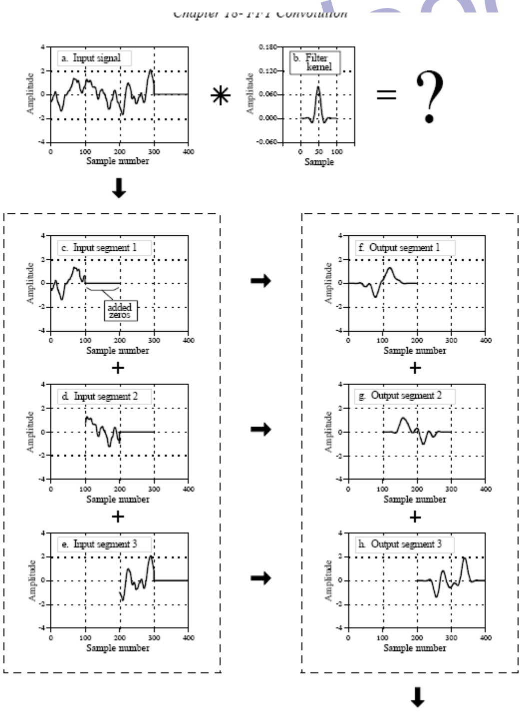


FIGURE 18-1
The overlap-add method. The goal is to convolve the input signal, (a), with the filter kernel, (b). This is done by breaking the input signal into a number of segments, such as (c), (d) and (e), each padded with enough zeros to allow for the expansion during the convolution. Convolving each of the input segments with the filter kernel produces the output segments, (f), (g), and (h). The output signal, (i), is then found by adding the overlapping output segments.

شکل ۹-۱ روش

Over-Lap-add Method = هدف کانولوشن سیگنال ورودی با پاسخ ضربه فیلتر اصلی باشد این روش با تجزیه سیگنال ورودی به تعدادی بخش حاصل می شود و سپس هر بخش خروجی اش از طریق کانولوشن آن بخش با پاسخ ضربه فیلتر اصلی بدست می آید (لازم به یادآوری است در آخر هر بخش دامنه از ۱ نمونه ای به بعد صفر می شود (مقدار قابل توجه)

تاخیر فاز: وقتی سیگنال ورودی حاوی فرکانس های متفاوت باشد آنگاه سیگنال خروجی در هرکدام از فرکانس ها به نسبت $Z_p(w_\circ) = \frac{-\theta(w_\circ)}{w_\circ}$ تاخیر پیدا می کند که این مطلب باعث می شود که سیگنال خروجی شباهتی به سیگنال ورودی نداشته باشد.

تأخیر گروهی: معیار خطی بودن فاز است نسبت به فرکانس و نمایش گر میزان تاخیر زمانی شکل موج ورودی و خروجی آنالوگ می باشد.

فصل دوم

فیلترهای وفقی

۲-۱-سیگنال های گسسته تصادفی

۲-۱-۱-مقدمه:

سیگنال هایی که از الگوی پریودیک تبعیت می کنند و یا توسط روابط ریاضی بیان می شوند را قطعی[۵۱] می نامند. در مقابل سیگنال هایی وجود دارند که مقادیر آنها را نمی توان با قطعیت پیش بینی کرد، به این دسته سیگنال های تصادفی[۵۲] می گوییم. به طور کلی سیگنال های قطعی برای آزمایش سیستم ها به کار می روند، به عنوان مثال پیدا کردن پاسخ فرکانس یک فیلتر، در حالی که سیگنال های تصادفی هر جا تبادل اطلاعات است، یافت می شوند. در پردازش سیگنال های دیجیتال[۵۳] فهمیدن توابع تصادفی مقدمه ای برای شناخت هر دوی اطلاعات و نویز موجود در سیگنال می باشد.

۲-۱-۲-میانگین[۵۴]، میانگین مربع[۵۵] و واریانس[۵۶]:

میانگین یک سیگنال گسسته با مجموع حاصلضرب اندازه هر نمونه در احتمال وقوع آن بدست می آید. به این رابطه بعضی اوقات مقدار dc نیز گفته می شود و «E» بیانگر امید ریاضی می باشد.

---
51 -deterministic
52 -random signals
53 -Sigital Signal Processing
54 -Mean
55 -Mean Square
56 -Variance

$$m_x = \overline{x[n]} = E\{x[n]\} = \sum_{c=-\infty}^{+\infty} C.P_{x[n]=c}$$

پارامتر مهم بعدی میانگین مربع می باشد که به آن توان متوسط نیز می گویند و طبق رابطه زیر محاسبه می شود.

$$\overline{x^{2}[n]} = E\{x^{2}[n]\} = \sum_{c=-\infty}^{+\infty} C.^{2}P_{x[n]=c}$$

و پارامتر سوم در شناخت سیگنال واریانس بوده که بیانگر میزان نوسانات حول میانگین می باشد.

$$\delta^{2} = E\{(x_{[n]} - m_{(x)})^{2}\}$$

از آنجائیکه میانگین مربع برابر با توان کل متوسط و میانگین نشان دهنده توان dc می باشد، می توان به نوعی واریانس را میزان توان ac دانست.

۲-۱-۳-میانگین زمانی و میانگین Ensemble

ما تصور می کردیم که احتمالات مربوط به مقادیر گسسته سیگنال تصادفی موجود بوده یا محاسبه می گردند. سپس این احتمالات در بدست آوردن میانگین و واریانس مفید واقع می شدند. حال می خواهیم تفاوت بین میانگین زمانی و میانگین ensemble را مشخص کنیم.

پردازش یک سیگنال آماری به تعداد نامتناهی متغیرهای آماری ختم شده، که سیگنال $x[n]$، $-\infty < n < \infty$، تنها یکی از متغیرهای مورد بحث می باشد. همه روابط بدست آمده از این پردازش یک ensemble نامتناهی نامیده می شود و توابع آماری و امید ریاضی توصیف شده در بخش ۲-۱-۲ از این قبیل می باشند.

در پردازش سیگنال دیجیتال ما کار با قسمت های خاصی از سیگنال را نسبت به حالت ensemble ترجیح می دهیم. هر نمونه بیانگر مقدار یکی از متغیرهای در حال پردازش می باشد و ما باید بین مشخصات کلی و اطلاعات موجود در یک بازه زمانی ارتباط برقرار کنیم. این پروسه بوسیله میانگین زمانی بوجود می آید و روابط آن به صورت زیر است.

$$\overline{x[n]} = E\{x[n]\} = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x[n]$$

$$\overline{x[n]} = E\{x[n]\} = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x^2[n]$$

$$\delta^2 = E\left\{(x[n] - \overline{x[n]})^2\right\} = \lim_{N\to\infty} \frac{1}{2Na1} \sum_{n=-N}^{N} (x[n] - \overline{x[n]})^2$$

حدهای به کار رفته در روابط بالا فقط در صورتی موجود می باشند که $x[n]$ دارای میانگین محدود باشد.

یک سیگنال را ساکن[57] می نامند هرگاه روابط آماری آن مستقل از زمان باشد و همچنین یک سیگنال را Wide Sense Stationary می نامند هرگاه هم میانگین و هم تابع همبستگی آن مستقل از زمان باشد.

۲-۱-۴-تابع خود همبستگی[58]

مقادیر توزیع یک سیگنال تصادفی، مشخصات آماری پراکندگی دامنه را و همچنین احتمال وجود یک نمونه را بیان می کند. ولی متأسفانه در باره اینکه این احتمالات نمونه ها به هم مرتبط هستند یا نه، اطلاعاتی نمی دهد. به هر حال می توان مشخصات ساختار حوزه زمان یک سیگنال را بوسیله تابع خود همبستگی بدست آورد. تابع خود همبستگی

[57] -Stationary
[58] -Auto Correlation Function

(ACF) را میانگین حاصلضرب سیگنال x[n] با شیفت یافته خود در حوزه زمان تعریف می کنند. این تابع یک معیار ارزشمند سیگنال آماری می باشد که وابستگی بین مقادیر x[n] را در زمان های مختلف بیان می دارد و به طور کل ساختار حوزه زمان را خلاصه می کند. این تابع طبق رابطه زیر بیان می شود.

$$\phi_{xx}[m] = E\{x[n]x[n+m]\} = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x[n]x[n+m]$$

تابع دیگر که به ACF شبیه می باشد تابع Auto Covariance می باشد:

$$\gamma_{xx}[m] = E\{(x[n]-\overline{x[n]})\ (x[n+m]-\overline{x[n]}\ \} = \mathrm{Lim}_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} (x[n]-\overline{x[n]})\ (x[n+m]-\overline{x[n]})$$

از روابط بالا این نکته برداشت می شود که در صورت صفر بودن میانگین سیگنالی، آنگاه دو تابع فوق برای آن برابرند. مقدار مرکزی ACF برابر است با میانگین مربع آن سیگنال و به نوعی بیانگر توان کلی آن می باشد و همیشه بیشترین دامه را دارد. یک استفاده کاربردی این تابع شناسایی یک سیگنال پریودیک در حضور نویز می باشد.

۲-۱-۵-سیگنال و نویز

یکی از مهم ترین بحث ها در پردازش سیگنال دیجیتال بازیابی اطلاعات مفید از نویز ناخواسته می باشد. در این قسمت تابع ACF یک سیگنال ترکیب شده با نویز را مشاهده می کنیم.

$$\begin{aligned}
&y[n] = s[n] + q[n] \\
&\phi_{yy}[n] = E\{(s[n]+q[n])(s[n+m)+q(n+m)\} \\
&\qquad = E\{s[n].s[n+m]\} + E\{q[n].q[n+m]\} + 2E\{s[n].q[n.m]\} \\
&\qquad = \phi_{ss}[m] + \phi_{qq}[m] + 2E\{s[n]+q[n+m]\}
\end{aligned}$$

سیگنال $x[n]$ و نویز $q[n]$ کاملاً ناهمبسته می باشند یا به بیان دیگر احتمال هر نمونه $q[n]$ مستقل از نمونه های $s[n]$ است. پس آخرین جمله در رابطه بالا برابر صفر می باشد.

$$E\{s[n].q[n+m]\} = E\{s[n]\}E\{q[n+m]\} = 0$$

دلیل صفر شدن عبارت بالا برابر صفر بودن میانگین نویز است. بنابراین ACF سیگنال با نویزی که میانگین آن صفر است به صورت زیر است:

$$\phi_{yy}[m] = \phi_{ss}[m] + \phi_{qq}[m]$$

۲-۱-۶-تابع همبستگی متقابل[59]

ACF برای شناخت ساختار سیگنال در حوزه زمان به کار می رود. تابع همبستگی متقابل (CCF) تابعی می باشد که به جای مقایسه سیگنال با شیفت یافته خود، دو سیگنال متفاوت را مقایسه می کند. CCF دو سیگنال $x[n]$ و $y[n]$ و کواریانس متقابل در قسمت میانگین زمانی به صورت زیر تعریف می شود.

$$\phi_{xy}[m] = E\{x[n]y[n+m]\} = \lim_{N\to\infty}\frac{1}{2N+1}\sum_{n=-N}^{N}x[n]y[n+m]$$

$$\gamma_{xy}[m] = E\{(x[n]-\overline{x[n]})(y[n+m]-\overline{y[n]})\} = \lim_{N\to\infty}\frac{1}{2N+1}\sum_{n=-N}^{N}(x[n]-\overline{x[n]})(y[n+m]-\overline{y[n]})$$

هر دوی این توابع مرتبه دوم می باشند.[60] CCF اجزای فرکانسی موجود بین دو سیگنال را منعکس می کند و همچنین اطلاعات ارزشمندی را از قبیل فازهای منسوب به فرکانس

[59] -Cross Correlation Function
[60] -Second Order

های مشترک[۶۱] را در بر دارد. مهم ترین کاربرد (CCF) در مقایسه با دو سیگنال مشابه که فقط از نظر زمانی متفاوت می باشند، است. به عبارتی سیگنال هدف[۶۲] را در سیگنال دیگر می یابد و زمان وقوع آن را محاسبه می کند. در این کاربرد همه نمونه های CCF برابر صفر است مگر آن نمونه هایی که دو سیگنال مشابه به هم می رسند. این نکته در سیستم رادار و سنجش فاصله کاربرد دارد.

۲-۲-فیلترهای وفقی

۲-۲-۱-مقدمه

خصوصیت اصلی فیلتر وفقی متغیر با زمان بودن ضرایب آن و تنظیم خودکار مشخصات فیلتر می باشد. یک فیلتر وفقی معمولاً از ساختار فیلتر FIR، به همراه الگوریتم وفقی که به طور پیوسته ضرایب فیلتر را تغییر می دهد، بهره می برد. اکثر الگوریتم های وفقی برگرفته شده از فیلتر Wiener می باشند، بنابراین در ابتدا مروری بر نحوه کارکرد این فیلتر داریم.

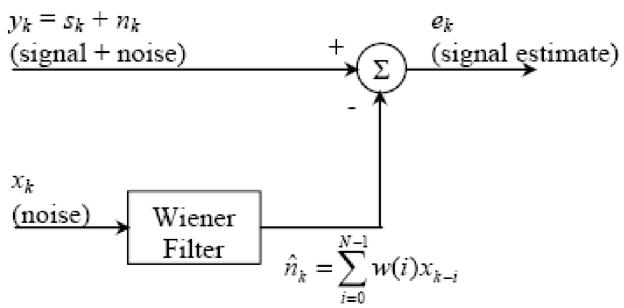
۲-۲-۲-تئوری فیلتر Wiener

نقطه شروع در روابط وفقی، تعریف مشخص فیلتر بهینه می باشد. فیلتر Wiener رایج ترین نوع مورد استفاده می باشد.

61 -phase related common frequencies
62 -target

در شکل زیر عملکرد آن مشخص شده است.

شکل 1-2



The Wiener filter configuration.

Kامین

نمونه سیگنال y ، y_k، شامل سیگنال اصلی s_k و نویز n_k می باشد، که نویز n_k خود با x_k همبسته است. فیلتر wiener به کمک یک فیلتر دیجیتال (در این مثال فیلتر FIR مرتبه N) نویز را تخمین می زند.

$$e_k = y_k - \hat{n}_k = y_k - \sum_{i=0}^{N-1} w(i).x_{k-i}$$

w(i)، iامین ضریب فیلتر می باشد، از آنجاییکه با سیگنال گسسته کار می کنیم می توانیم روابط را به صورت ماتریسی بیان کنیم:

$$X_k = \begin{bmatrix} x_k \\ x_{k-1} \\ \cdot \\ \cdot \\ \cdot \\ x_{k-(N-1)} \end{bmatrix} \qquad W = \begin{bmatrix} w_{(0)} \\ w_{(1)} \\ \cdot \\ \cdot \\ \cdot \\ w_{(N-1)} \end{bmatrix}$$

$$e_k = y_k - W^T X_k = y_k - X_k^T W$$

مربع خطای لحظه ای سیگنال به صورت زیر است:

$$e_k^2 = y_k^2 - 2W^T (y_k X_k) + W^T X_k X_k^T W$$

میانگین مربع خطا همان امید ریاضی عبارت بالا می باشد:

$$\varepsilon = MSE = E\{e_k^2\} = E\{y_k^2\} - 2W^T E\{y_k X_k\} + W^T E\{X_k X_k^T\} W$$

اگر $E\{X_k X_k^T\}$ را با R_{xx} و $E\{y_k X_k\}$ را با R_{yx} جایگزین کنیم، رابطه بالا به صورت زیر ساده می شود:

$$MSE = E\{y_k^2\} - 2W^T R_{yx} + W^T R_{xx} W$$

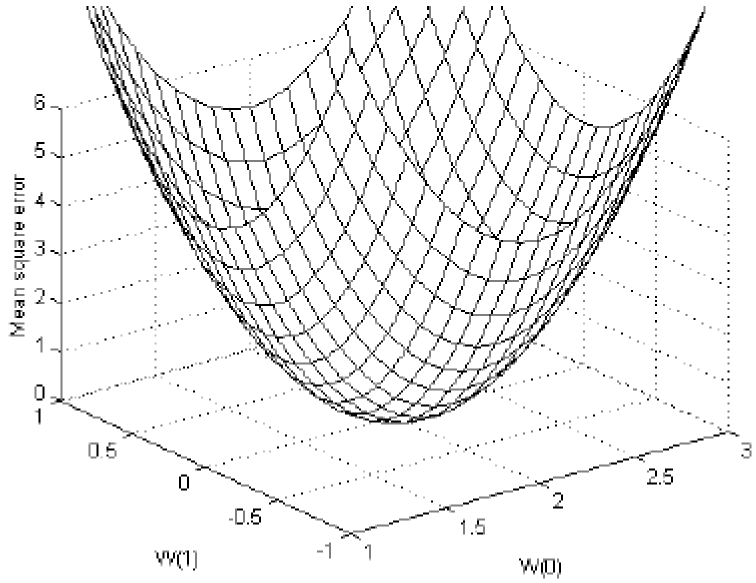
از رابطه بالا می توان فهمید که MSE یک معادله درجه دوم بر حسب ضرایب فیلتر W می باشد.

۲-۲-۳-سطح اجرا[63]

قسمتی از تابع MSE دو بعدی در زیر به نمایش آمده است. محور عمودی مقدار MSE و محورهای افقی ضرایب فیلتر می باشند. تابع درجه دوم خطا برای بدست آوردن ضرایب بهینه W_{opt} مفید می باشد.

[63] -Performance Surface

شکل ۲-۲



A two-dimensional quadratic performance surface.

بسیاری از پروسه های وقتی دنبال نقطه حداقل بردار وزن[۶۴] می گردند و برای این کار از گرادیان استفاده می کنند. گرادیان (∇) MSE بوسیله رابطه زیر بدست می آید.

$$\nabla = \frac{\partial(MSE)}{\partial W} = \begin{bmatrix} \frac{\partial MSE}{\partial W(0)} \\ \frac{\partial MSE}{\partial W(1)} \\ \frac{\partial MSE}{\partial W(N-1)} \end{bmatrix}$$

در قسمت قبل دیدیم که MSE امید ریاضی مجذور خطا می باشد. پس داریم:

$$\nabla = E\left\{\frac{\partial e^2 k}{\partial W}\right\} = E\left\{2e_k \frac{\partial e_k}{\partial W}\right\} = 2E\left\{(y_k - X_k^T W)\partial\!\!\diagup_{\partial W}(y_k - X_k^T W)\right\}$$

$$= -2E\left\{(y_k - X_k^T W)X_k\right\} = -2E\{X_k y_k\} + 2E\left\{X_k X_k^T\right\}W = -2R_{yx} + 2R_{xx}W$$

[64] -Weights Vector

وقتی ضرایب فیلتر مقدار حداقل را برمی گیزیند که مشتق آن برابر صفر شود.

$$0 = -2R_{yx} + 2R_{xx}W_{opt} \Rightarrow W_{opt} = R_{xx}^{-1}R_{yx}$$

این رابطه به Wiener-Hopf معروف است و فیلتر بدست آمده از رابطه بالا همان فیلتر Wiener می باشد. البته در عمل از این رابطه استفاده نمی کنند، مشکل این است که محاسبه ماتریس معکوس $N \times N$ R_{xx}^{-1} برای هر نمونه بسیار پیچیده است.

یک روش دیگر برای این کار، الگوریتم steepest descent می باشد. در این روش وزن ها به صورت بازگشتی از رابطه زیر محاسبه می شوند:

$$W_{p+1} = W_p - \mu \nabla_p$$

W_p، بردار وزن است بعد از p بار تکرار و ∇_p بردار گرادیان است که بعد از جایگزینی W_p محاسبه شده و μ یک ثابت است که سایز پله[65] و پایداری و سرعت همگرایی را تنظیم می کند.

۲-۲-۴-الگوریتم LMS

الگوریتم Least Mean Square بدلیل سادگی شهرت زیادی دارد. این الگوریتم از روش steepest descent استفاده می کند ولی بدلیل محاسبه رابطه بازگشتی فقط یک بار در هر نمونه و همچنین محاسبه تقریبی گرادیان ($\hat{\nabla}_k$) ساده تر شده است.

[65] -Step Size

$$\hat{\nabla}_k = \begin{bmatrix} \frac{\partial e^2 k}{\partial w(0)} \\ \frac{\partial e^2 k}{\partial w(1)} \\ \frac{\partial e^2 k}{\partial w(N-1)} \end{bmatrix} = 2e_k \begin{bmatrix} \frac{\partial ek}{\partial w(0)} \\ \frac{\partial ek}{\partial w(1)} \\ \frac{\partial ek}{\partial w(N-1)} \end{bmatrix} = -2e_k X_k$$

با جایگذاری عبارت بالا در رابطه با رابطه گشتی داریم:

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

برای پیاده سازی الگوریتم LMS مراحل زیر باید طی شوند.

۱-مقداردهی اولیه به ضرایب فیلتر

۲-در هر نمونه مراحل زیر را اجرا می کنیم.

۲-a محاسبه خروجی فیلتر: $\hat{n}_k = \sum_{i=0}^{N-1} w_k(i) x_{k-i}$

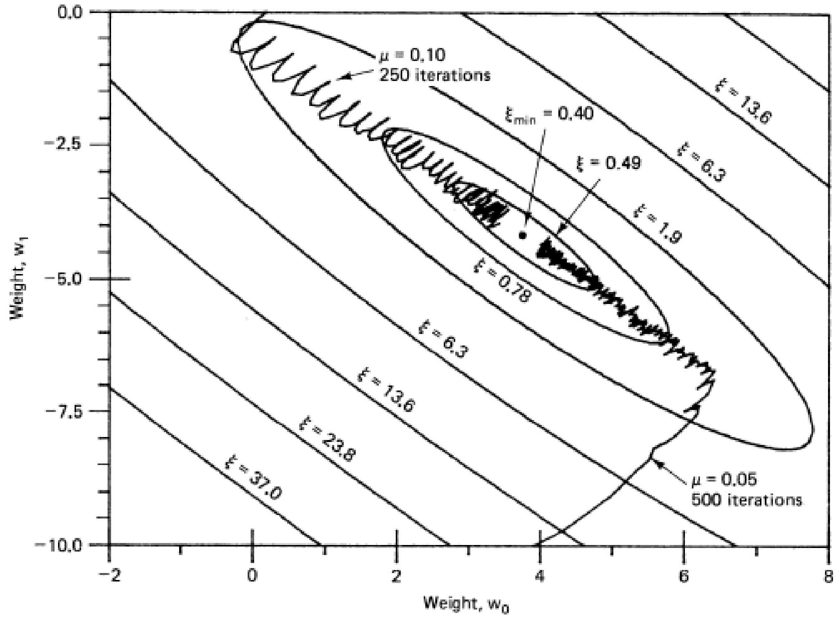
۲-b محاسبه تخمین خطا: $e_k = y_k - \hat{n}_k$

۲-c بدست آوردن ضرایب جدید فیلتر $w_{k+1}(i) = w_r(i) + 2\mu e_k x_k$

۲-۲-۵-خصوصیات همگرایی

نحوه اجرای الگوریتم LMS در شکل زیر آمده است. منحنی های MSE (ε) در شکل نشان داده شده اند و ضرایب بهینه برای این مثال خاص $\omega(0) = 3/784$ و $\omega(1) = -4/178$ می باشد. مقادیر منحنی طبق جدول زیر بدست آمده است.

شکل ۳-۲



Performance surface contours and weight value tracks for the LMS [Widrow 85].

	نقطه شروع $\omega(۰)$ و $\omega(۱)$	مقدار μ	تعداد تکرار
منحنی بالایی	۰ ۰	۰/۱	۲۵۰
منحنی پایینی	۴ -۱۰	۰/۰۵	۵۰۰

به نظر می رسد منحنی که مقدار μ بزرگتر دارد، بدلیل اینکه مقدار ضرایب در هر تکرار بزرگتر است، بیشتر نوسان می کند، ولی با مقدار تکرار کمتر به همان فاصله ایده آل از MSE_{min} رسیده است.

انتخاب μ بسیار مهم است زیرا تنظیم کننده پایداری و سرعت همگرایی می باشد. اگر μ خیلی کوچک باشد، ممکن است پروسه خیلی طولانی شود و اگر خیلی بزرگ باشد، امکان ناپایداری وجود دارد. پس محدوده ای که برای μ وجود دارد به صورت زیر است:

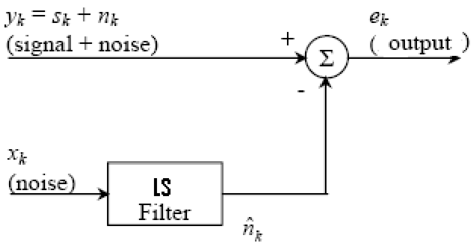
$$0 < \mu < 1/\lambda_{max}$$

که λ_{max} بزرگترین مقدار ویژه[66] ماتریس کواریانس سیگنال ورودی است.

۲-۲-۶-الگوریتم RLS[67]

به دلیل محدودیت های موجود در LMS از قبیل حجم زیاد محاسبات، الگوریتم دیگری بنام RLS در بعضی موارد به LMS ترجیح داده می شود. این الگوریتم از ساختار زیر پیروی می کند.

شکل ۲-۴



سیگنال y_k، پاسخ به

ورودی $x_{k(i)}$، $i = ۱,۲,...,n$ می باشد و طبق این معادله به هم مربوط می شوند.

$$y_k = \sum_{i=۰}^{n-۱} w(i) x_k(i) + e_k$$

که e_k بیانگر میزان خطا و $w(i)$ تعیین کننده میزان تأثیر iامین نمونه ورودی در سیگنال y_k می باشد. مسئله در روش LS این است که $x_k(i)$ و y_kرا داریم و $w(0)$ تا

66 -eigenvalue
67 -Recursive Least Square

w(n-1) را می خواهیم. تخمین های مناسب ضرایب فیلتر w(i) از رابطه زیر بدست می آیند:

$$W_m = \left[X_m^T X_m\right]^{-1} X_m^T Y_m$$

که

$$Y_m = \begin{bmatrix} y_0 \\ y_1 \\ . \\ . \\ . \\ y_{m-1} \end{bmatrix} \quad X_m = \begin{bmatrix} X^T(0) \\ X^T(1) \\ . \\ . \\ . \\ X^T(m-1) \end{bmatrix} \quad W_m = \begin{bmatrix} w(0) \\ w(1) \\ . \\ . \\ . \\ w(n-1) \end{bmatrix}$$

و $k = 0,1,\dots,m-1$

و $X^T(k) = \left[x_{k(0)} \quad x_k(1)\dots\dots x_k(n-1)\right]$

پسوند m در روابط بالا نشان دهنده این است که ماتریس بالا از همه m داده خود استفاده می کنند. محاسبه رابطه بالا بسیار وقت گیر بوده و برای زمان – حقیقی[68] و یا فیلتر کردن online مناسب نمی باشد. در عمل وقتی با داده های پیوسته کار می کنیم، روش های بازگشتی ترجیح داده می شوند. با الگوریتم RLS تخمین W_m می تواند با هر ورودی جدید محاسبه شود و نیازی به حساب کردن مکرر ماتریس معکوس نیست.

الگوریتم RLS از داده گذاری نمایی برای از بین بردن تدریجی مقدارهای قبلی W_m استفاده می کنند.

[68] -Real-time

$$W_k = W_{k-1} + G_k e_k$$

$$P_k = \frac{1}{\gamma}[P_{k-1} - G_k X^T(k)\, P_{k-1}]$$

$$G_k = \frac{P_{k-1} X(k)}{\alpha_k}$$

$$e_k = y_k - X^T(k) W_{k-1} \quad , \quad \alpha_k = \gamma + X^T(k) P_{k-1} X_k$$

نقش P_k در روش بازگشتی محاسبه ماتریس معکوس $\left[X_k^T X_k\right]^{-1}$ می باشد. K بر این نکته دلالت می کند که مقادیر در هر نمونه بدست می آیند. γ به ضریب فراموش شده[69] معروف است و معمولاً بین ۰/۹۸ تا ۱ انتخاب می شود.

اشکال الگوریتم بالا این است که اگر سیگنال ورودی برای مدت زمان طولانی صفر بماند آنگاه طبق روابط بالا داریم:

$$\lim_{K\to\infty} P_k = \lim_{K\to\infty} \left(\frac{P_{k-1}}{\gamma_{k-1}}\right)$$

در این حالت P_k به صورت نمایی اضافه شده و باعث ناپایداری سیستم می شود.

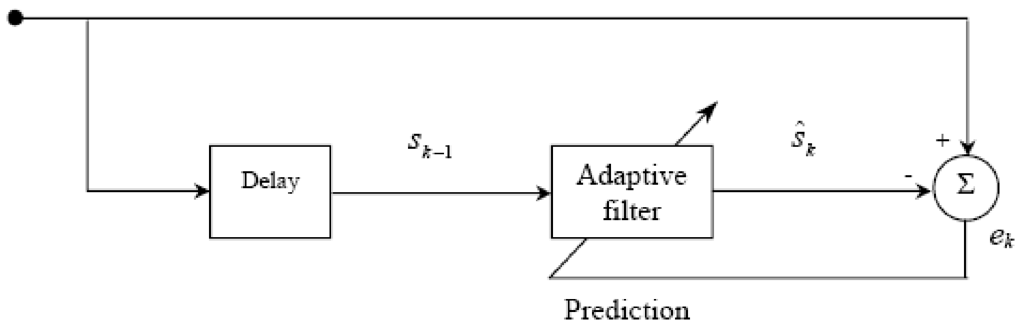
۲-۲-۷ مدل های مختلف کاربرد فیلترهای وفقی

۱-پیش بینی خطی[70]

شاید ساده ترین مدل استفاده فیلتر وفقی همین حالت می باشد که در شکل زیر آمده است. سیگنال خواسته شده همان ورودی S می باشد و تأخیر داده شده S به فیلتر وفقی فرستاده می شود که باید مقدار فعلی S را پیش بینی کند تا مقدار خطا e به صفر برسد. این مدل معمولاً در رمزگذاری سیگنال[71] کاربرد دارد.

[69] -Forgetting Factor
[70] -Liner Prediction
[71] -Signal encoding

شکل ۵-۲

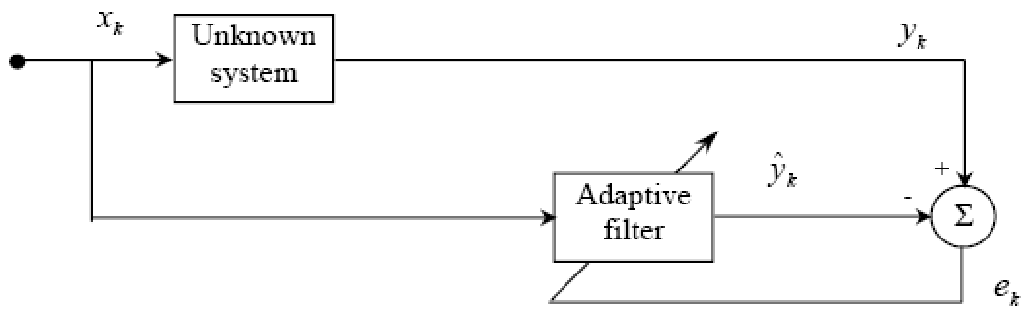


۲-مدلسازی سیستم به صورت مستقیم[۷۲]

در این حالت (شکل زیر) سیگنال x_k هم برای فیلتر وفقی و هم برای سیستم نامعلوم[۷۳] فرستاده می شود. برای کم کردن مقدار e_k پردازشگر وفقی سعی در شبیه سازی مشخصات سیستم را دارد. این مدل در حذف اکو و نویز کاربرد دارد.

شکل ۶-۲

---
72 -Direct System modelling
73 -Unknown System



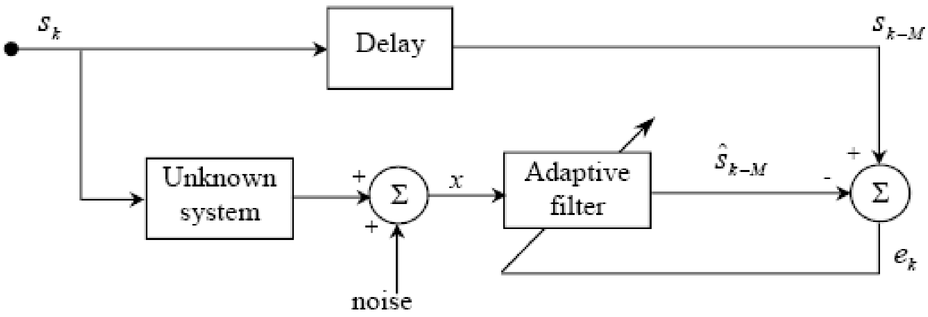
Direct system modelling

۳-مدلسازی سیستم به صورت معکوس[۷۴]

مطابق شکل زیر، این مدل تأخیر یافته سیگنال S_k را بازسازی می کند که قرار بوده است با سیستم نامعلوم دچار تغییر شود. کاربرد این مدل در از بین بردن اثر کانال مخابراتی می باشد.

[74] -Inverse System Modelling

شکل ۷-۲



Inverse system modelling

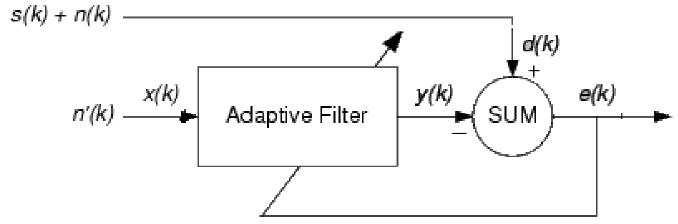
۴-

حذف نویز وفقی[۷۵]

در این مدل پرکاربرد فیلتر وفقی نویز تصادفی را از روی سیگنال اصلی برمی دارد. در شکل زیر توجه داشته باشید که هدف یکسان سازی سیگنال e_k و s_k می باشد و مثل روش های قبلی e_k صفر نمی شود.

شکل ۸-۲

[75] -Apaptive noise Cancelling



Using an Adaptive Filter to Remove Noise from an Unknown System

فصل سوم

کار با MATLAB

۳-۱-مقدمه:

امروزه نرم افزار MATLAB به دلیل قابلیت های گوناگون در زمینه های مختلف، یکی از محبوب ترین نرم افزارها بین مهندسین می باشد. این نرم افزار بین دانشجویان و مهندسین برق از جایگاه بخصوصی برخوردار است، زیرا به کمک MATLAB می توان پروژه های مختلف کنترلی، مخابراتی و ... را بدون هیچ هزینه سخت افزاری شبیه سازی نمود. امکانات جدیدی که MATLAB در نسخه های جدید خود راائه کرده، از قبیل ساختن خروجی های COE برای Xilinx و C header file و Code composer این نرم افزار را به قوی ترین در زمینه پردازش سیگنال تبدیل کرده است.

همچنین MATLAB طبق انتظار در زمینه طراحی و پیاده سازی انواع فیلترهای دیجیتال، FIR و IIR، بدون نیاز به محاسبات و مدارات پیچیده، توابعی گوناگونی را در اختیار کاربران گذاشته است. برای این کار MATLAB دو محیط را به کاربر پیشنهاد می کند. یکی استفاده از توابع متعدد برای انواع فیلترها در محیط Command Windows می باشد. هر فیلتر اعم از چپی شف، باتر و رث و ... دارای توابع خاص خود بوده و به راحتی می توان مشخصات آنها را تنظیم نمود. راه دیگر Filter design Toolbox می باشد. در این محیط به سهولت می توان نوع فیلتر مورد نظر را انتخاب کرد و با توجه به نیاز مشخصات آن را تنظیم نمود. در ادامه با هر یک از این دو محیط و

قابلیت های آنها بیشتر آشنا خواهیم شد. در پایان برنامه ای برای نشان دادن توانایی MATLAB در زمینه فیلترهای وفقی آورده ایم.

۳-۲-فیلتر در محیط Command Window

در این بخش ابتدا به بیان توابع کلی فیلتر پرداخته، سپس توابع موجود برای فیلتر های خاص مورد بررسی قرار گرفته اند. همانطوریکه در فصل ۱ دیدیم، فیلترهای دیجیتال دو دسته اند، یکی فیلترهایی که با کانولوشن کار می کنند (FIR) و دیگری فیلترهای بازگشتی (IIR) می باشد. در ادامه دستوراتی را برای تحقق فیلترهای دیجیتال مرور می کنیم.

اولین دستور Conv است که این دستور برای تحقق کانولوشن و فیلترهای FIR مفید است

C=Conv(A,B);

B و A سیگنال های ورودی بوده و C مقدار کانولوشن آنها می باشد که طبق رابطه زیر محاسبه می شود.

$$C[n] = A * B = \sum_{k=-\infty}^{\infty} A[k]B[n-k]$$

مثال:

```
x = randn(٥,١);   %        یک متغیر تصادفی
h = [١ ١ ١ ١ ]/٤;   %      یک فیلتر میان گیر
y = Conv(h,x);
```

برای کار با فیلتر بازگشتی، ابتدا باید تابع تبدیل فیلتر را در حوزه Z بررسی کرد. می دانیم که روابط بین ورودی و خروجی سیگنال های گسسته در حوزه زمان با معادلات تفاضلی نشان داده می شوند. این معادلات در حوزه Z پس از مرتب کردن به شکل زیر درمی آید:

$$H(Z)=\frac{Y_{(z)}}{X_{(z)}}=\frac{b(1)+b(2)z^{-1}+\ldots+b(nb+1)z^{-nb}}{a(1)+a(2)z^{-1}+\ldots+a(na+1)z^{-na}}$$

و به $\max\{na,nb\}$، مرتبه فیلتر می گویند.

البته چون برای ما کار با صفر و قطب راحت تر است، MATLAB توابعی دارد که این دو را به هم تبدیل می کند:

$$[Z,P,K]=tf2zp(num,den)$$
$$[num,den]=zp2tf(z,p,k)$$

حال که با طریقه نمایش ضرایب فیلتر آشنا شدیم، معروف ترین تابع MATLAB برای فیلتر کردن را می بینیم:

$$y=Filter(b,a,x);$$

که x ورودی بوده و b و a ضرایب فیلتر می باشند.

همچنین در این تابع می توان شرایط اولیه را تأثیر داد.

$$[y,zf]=filtter(b,a,x,z_i)$$

در مواردی که فیلتر دارای مشخصات فاز غیرخطی می باشد از دستور Filtfilt استفاده می کنند. این دستور با دو بار فیلتر کردن، یکی به صورت مستقیم و دیگری معکوس، پارامتر group delay را خطی می سازد. البته هر وقت بخواهید می توانید با دستور

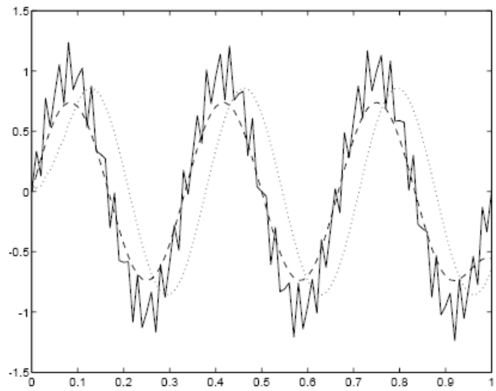
[g,d,w]=grpdelay(b,a,n) مشخصات group delay را مشاهده کنید. در مثال زیر تفاوت filtfilt را با filter می بینیم.

مثال ۲:

شکل ۱-۳

```
Fs = 100;
t = 0:1/Fs:1;
x = sin(2*pi*t*3)+.25*sin(2*pi*t*40);

b = ones(1,10)/10;          % 10 point averag
y = filtfilt(b,1,x);        % non-causal filt
yy = filter(b,1,x);         % normal filterin
plot(t,x,t,y,'--',t,yy,':')
```



همانطوریکه در شکل مشخص است خروجی تابع Filter حدود ۵ نمونه نسبت به سیگنال اصلی تأخیر دارد.

در ادامه چندین دستور برای پیاده سازی فیلترهای FIR و IIR آمده است.

در این توابع می توان پارامترهای مختلف فیلتر مانند ریپل باند عبور[76]، تضعیف باند قطع[77] و پهنای باند[78] و ... را تعیین کرد و در موارد دقیق تر حتی می توان مرتبه فیلتر را به حداقل رساند.

جدول زیر روش های مختلف فیلتر کردن و توابع موجود برای پیاده سازی آنها را بیان می کند.

[76] -Pass band ripple
[77] -Stop band attenuation
[78] -transition band

T1

Method	Description	Functions
Analog Prototyping	Using the poles and zeros of a classical lowpass prototype filter in the continuous (Laplace) domain, obtain a digital filter through frequency transformation and filter discretization.	Complete design functions: besself, butter, cheby1, cheby2, ellip Order estimation functions: buttord, cheb1ord, cheb2ord, ellipord Lowpass analog prototype functions: besselap, buttap, cheb1ap, cheb2ap, ellipap Frequency transformation functions: lp2bp, lp2bs, lp2hp, lp2lp Filter discretization functions: bilinear, impinvar
Direct Design	Design digital filter directly in the discrete domain by approximating a piecewise linear magnitude response.	yulewalk
Parametric Modeling*	Find a digital filter that approximates a prescribed time or frequency domain response.	Time-domain modeling functions: lpc, prony, stmcb Frequency-domain modeling functions: invfreqs, invfreqz
Generalized Butterworth Design	Design lowpass Butterworth filters with more zeros than poles.	maxflat

۳-۲-۱-توابع فیلترهای IIR

این توابع، همان فیلترهای آنالوگ IIR را به صورت دیجیتال پیاده سازی می کند. شما به راحتی می توانید فیلتری پایین گذر، بالاگذر، میان گذر و میان گذر را با هر مرتبه ای طراحی نمایید.

T2

Filter Type	Design Function

Butterworth	`[b,a] = butter(n,Wn,options)`
	`[z,p,k] = butter(n,Wn,options)`
	`[A,B,C,D] = butter(n,Wn,options)`

Chebyshev type I	`[b,a] = cheby1(n,Rp,Wn,options)`
	`[z,p,k] = cheby1(n,Rp,Wn,options)`
	`[A,B,C,D] = cheby1(n,Rp,Wn,options)`

Chebyshev type II	`[b,a] = cheby2(n,Rs,Wn,options)`
	`[z,p,k] = cheby2(n,Rs,Wn,options)`
	`[A,B,C,D] = cheby2(n,Rs,Wn,options)`

Elliptic	`[b,a] = ellip(n,Rp,Rs,Wn,options)`
	`[z,p,k] = ellip(n,Rp,Rs,Wn,options)`
	`[A,B,C,D] = ellip(n,Rp,Rs,Wn,options)`

Bessel (analog only)	`[b,a] = besself(n,Wn,options)`
	`[z,p,k] = besself(n,Wn,options)`
	`[A,B,C,D] = besself(n,Wn,options)`

به طور پیش فرض این توابع فیلتری پایین گذر با W_n خواسته شده را در اختیار شما قرار می دهد. برای طراحی فیلتر بالاگذر باید کلمه 'high' را در فهرست پارامترهای آن نوشت و همچنین برای فیلتر میانگذر، W_n را دوتایی وارد کنید و برای میان گذر از کلمه 'stop' استفاده کنید. در مثال های زیر می توانید مطالب بالا را مشاهده کنید.

مثال ۳:

فیلتر باتروث پایین گذر % ;(۰.٤,۰)[b,a] = butter

فیلتر چپی شف نوع اول میان گذر %;([۰.٤ ۰.۷],۱,٤)[b,a] = cheby۱

فیلتر چپی شف نوع دوم بالا گذر % ,('high,۰.۸,٦٥,٦)[b,a] = cheby۲

فیلتر الپتیک میان گذر ;('stop' ,[۰.٤ ۰.۷],٦٥,۱,۳)[b,a] = ellip

۳-۲-۲-طراحی مستقیم فیلتر IIR

برخلاف فیلترهای قبل، روش طراحی مستقیم فقط به چهار حالت، پایین گذر، بالاگذر، میان گذر و میان گذر محدود نمی شود و حتی توانایی ساخت فیلترهای با پاسخ فرکانسی دلخواه را دارد.

تابع yulewalk با مشابه سازی پاسخ فرکانسی مطلوب، فیلتر IIR بازگشتی را طراحی می کند. این تابع با FFT معکوس گرفتن از حوزه فرکانس و حل معاملات yule-walker به کمک تابع خود همبستگی فیلتر مربوط را طراحی می کند.

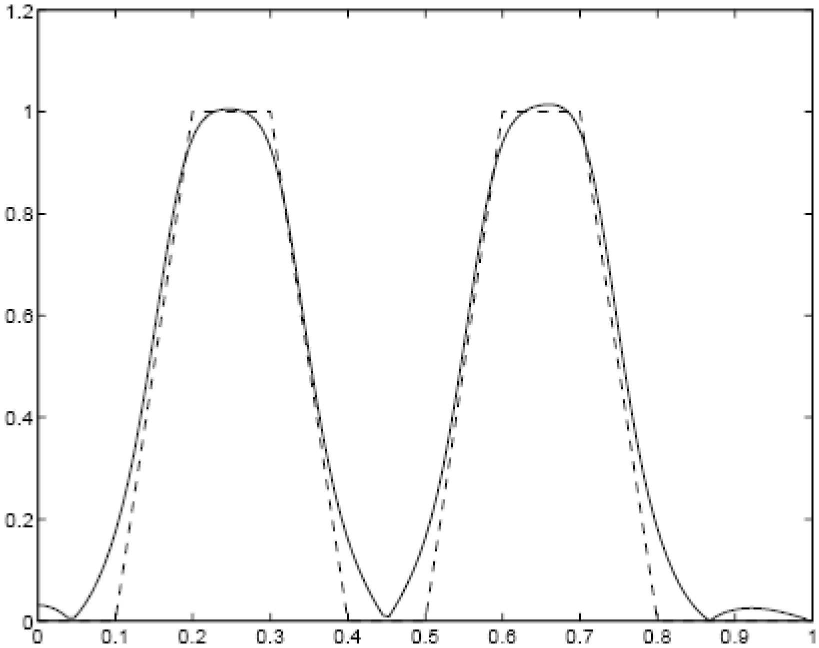
$[b,a] = \mathrm{yulewalk}(n,f,m)$

که این تابع، n مرتبه فیلتر و f و m بردارهایی هستند که مشخصات حوزه فرکانس فیلتر را مشخص می کنند.

مثال ۴: در این مثال یک فیلتر مولتی باند را به کمک تابع yule walk طراحی می کنیم.

شکل ۲-۳

```
m = [0   0   1   1   0   0   1   1   0 0];
f = [0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 1];
[b,a] = yulewalk(10,f,m);
[h,w] = freqz(b,a,128);
plot(f,m,w/pi,abs(h))
```



۳-۲-۳-طراحی فیلتر FIR

فیلترهای با پاسخ ضربه محدود دارای مزیت های زیر می باشند:

-دارای فاز کاملاً خطی هستند.

-همیشه پایدارند.

-روش های طراحی خطی دارند.

-در سخت افزار راحت پیاده سازی می شوند.

و عیب فیلترهای FIR این است که برای رسیدن به پاسخ فرکانسی مطلوب نیاز به مرتبه ی بالاتری نسبت به فیلترهای IIR داریم. روش های طراحی و توابع مربوط به آنها در جدول زیر آمده است.

T3

Method	Description	Functions
Windowing	Apply window to truncated inverse Fourier transform of desired "brickwall" filter	`fir1,fir2,kaiserord`
Multiband with Transition Bands	Equiripple or least squares approach over sub-bands of the frequency range	`firls,remez,remezord`
Constrained Least Squares	Minimize squared integral error over entire frequency range subject to maximum error constraints	`fircls,fircls1`
Arbitrary Response	Arbitrary responses, including nonlinear phase and complex filters	`cremez`
Raised Cosine	Lowpass response with smooth, sinusoidal transition	`firrcos`

به غیر از Cremez، بقیه توابع فیلترهای FIR تنها فیلترهای با فاز خطی طراحی می کنند. ضرایب «taps» این فیلترها از تقارن زوج یا فرد تبعیت می کنند. تأخیر فاز[۷۹] و

[79] -Phase-delay
80- group delay

تأخیر گروهی[۸۰] فیلترهای FIR با فاز خطی برابر می باشند و در تمام فرکانس ها ثابت می باشند. برای یک فیلتر FIR مرتبه n، تأخیر گروهی برابر $n/۲$ خواهد بود. یعنی سیگنال فیلتر شده به اندازه $n/۲$ فاصله زمانی تأخیر دارد ولی شکل موج آن بدون تغییر می ماند.

۳-۳- Filter design & Analysis toolbox (FDATOOL)

جعبه ابزار طراحی و آنالیز فیلتر مجموعه ای از ابزارها می باشد که تکنیک هایی برای طراحی، شبیه سازی و آنالیز فیلتر ارائه می دهند. این جعبه ابزار امکانات پردازش سیگنال را با ساختار فیلتر و روش های طراحی برای کاربردهای زمان حقیقی DSP مختلط مانند فیلترهای وفقی و فیلتر چندسرعته[۸۱]، افزایش می دهد. همچنین، به کمک آن می توان خروجی های VHDL و Verilag را دریافت نمود.

قابلیت ها این جعبه ابزار عبارتند از:

-روش های پیشرفته طراحی فیلتر FIR مانند کمترین مرتبه[۸۲]، مینیمم فاز، تحمیل کردن ریپل[۸۳]، باند متوسط[۸۴]، نایکویست و فاز غیرخطی

-بازسازی کامل و دو کاناله بانک طراحی فیلتر FIR

-روش های پیشرفته طراحی فیلتر IIR شامل دامنه دلخواه، اکولایزرهای تأخیر گروهی و فیلترهای notching، peaking و comb

[80] -group-delay
[81] -Multirate-Filters
[82] -minimum-order
[83] -Constrained ripple
84- half band
[84] -holfband

-آنالیز و پیاده سازی فیلترهای دیجیتال floating point، fixed point

-طراحی و ساخت فیلترهای IIR به فرم مرتبه دوم[۸۵]

-آنالیز نویز Round-off برای پیاده سازی فیلتر

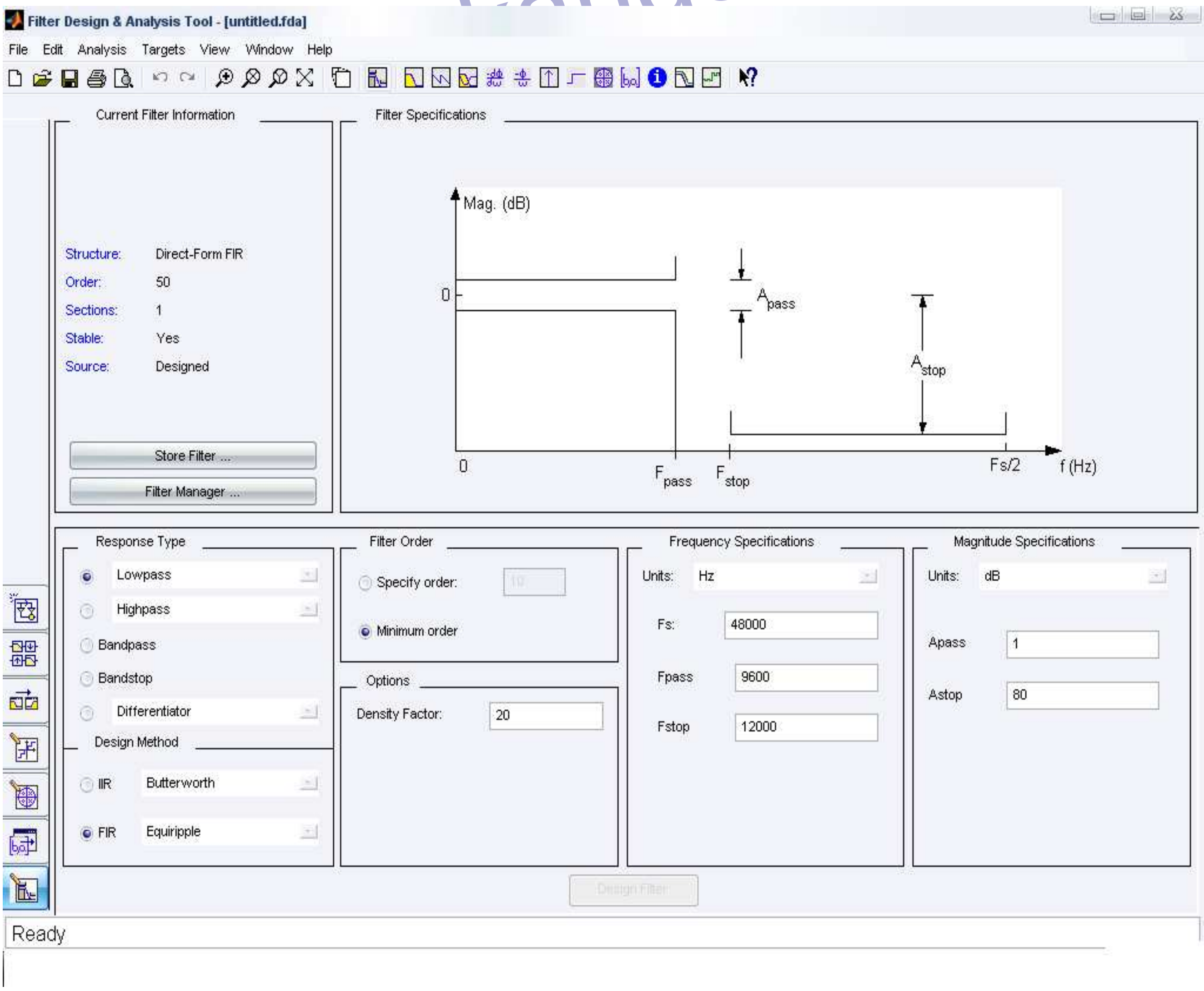
-طراحی، آنالیز و پیاده سازی فیلترهای وفقی شامل الگوریتم های LMS، RLS، Lattice و ...

-تولید کدهای VHDL و Verilog برای فیلترهای Fixed Point

در شکل زیرنمای کلی این جعبه ابزار را می بینید.

شکل ۳-۳

[85] -Second-order-Section

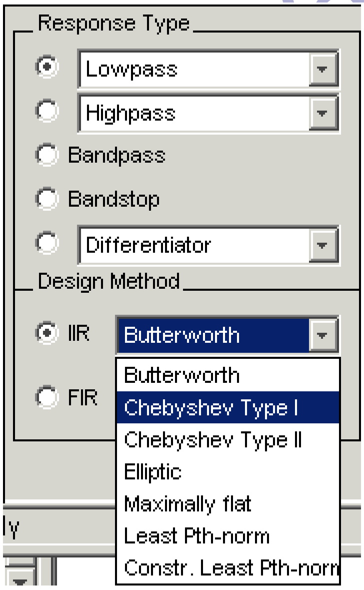


مثال ۵: در این مثال یک فیلتر چبیشف نوع اول رات طراحی و به کمک آن یک سیگنال را فیلتر می کنیم.

برای این کار مراحل زیر را انجام می دهیم:

۱-در قسمت Design Method، از لیست IIR، Chebyshev type1 را انتخاب می کنیم و دکمه Design Filtter را می زنیم.

شکل ۴-۳

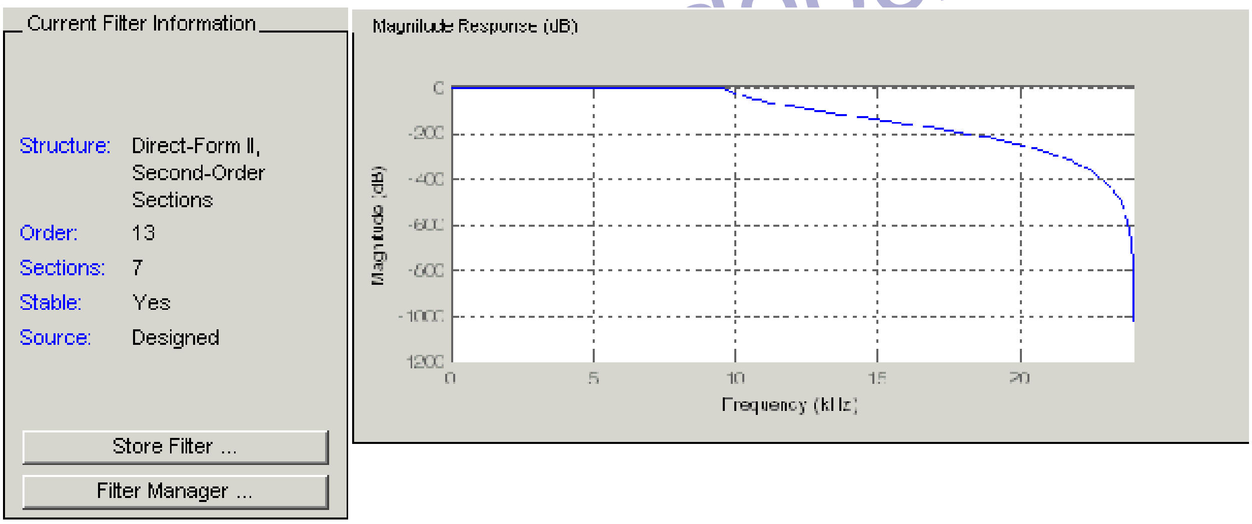


فیلتر ساخته شده یک

فیلتر پایین گذر با دقت مضاعف[86] می باشد که در قسمت پاسخ دامنه[87] رسم شده است. در قسمت اطلاعات فیلتر می توانیم مشخصات طراحی شده را ببینیم. در این مثال خاص مرتبه فیلتر (13) می باشد و ساختار Direct Form II را دارد.

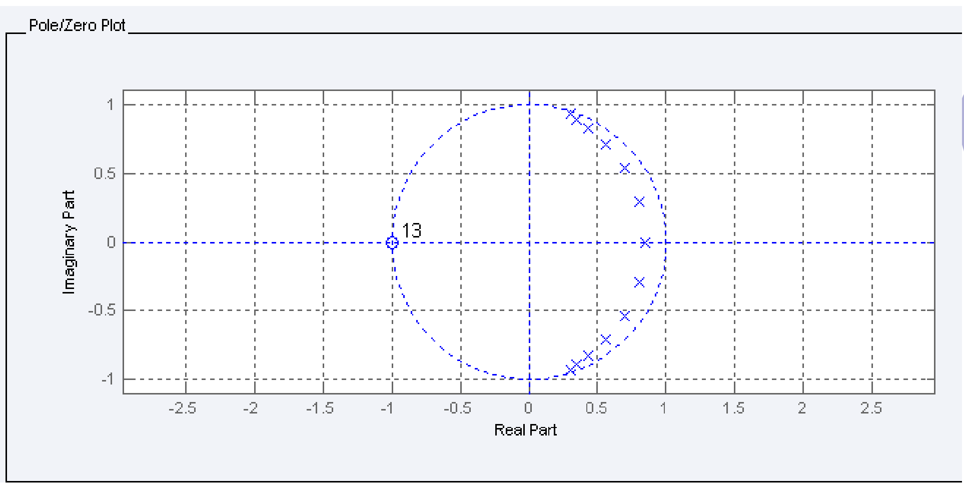
شکل ۵-۳

[86] -duble-precision
[87]-magnitude- response
[87] -magnitude-reponse

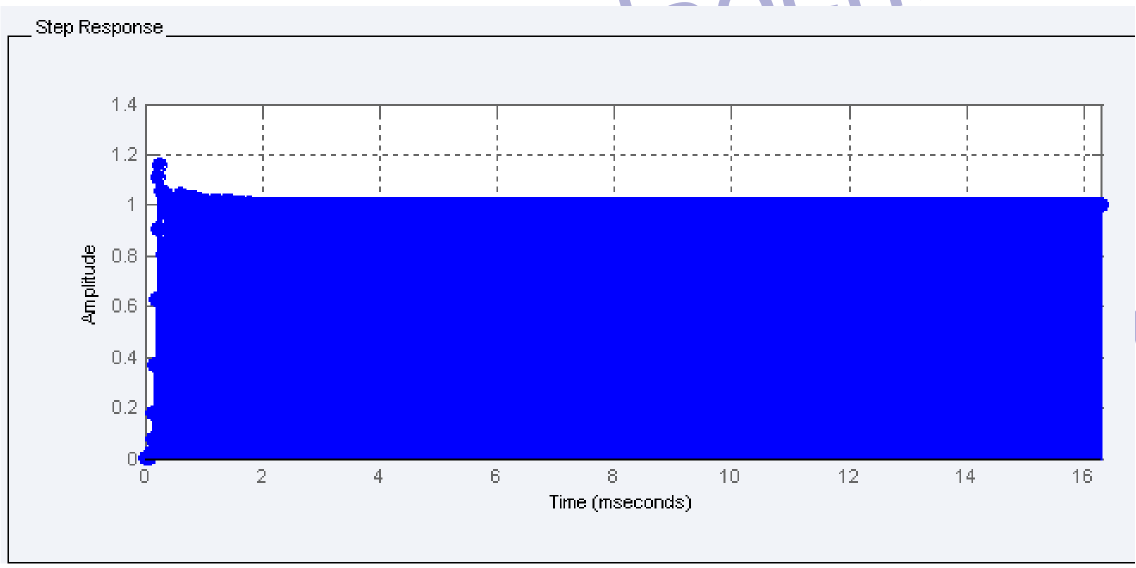


همچنین در این جعبه ابزار می توانیم نمودار صفر و قطب، پاسخ پله، پاسخ ضربه، تأخیر فاز و تأخیر گروهی را مشاهده کنیم.

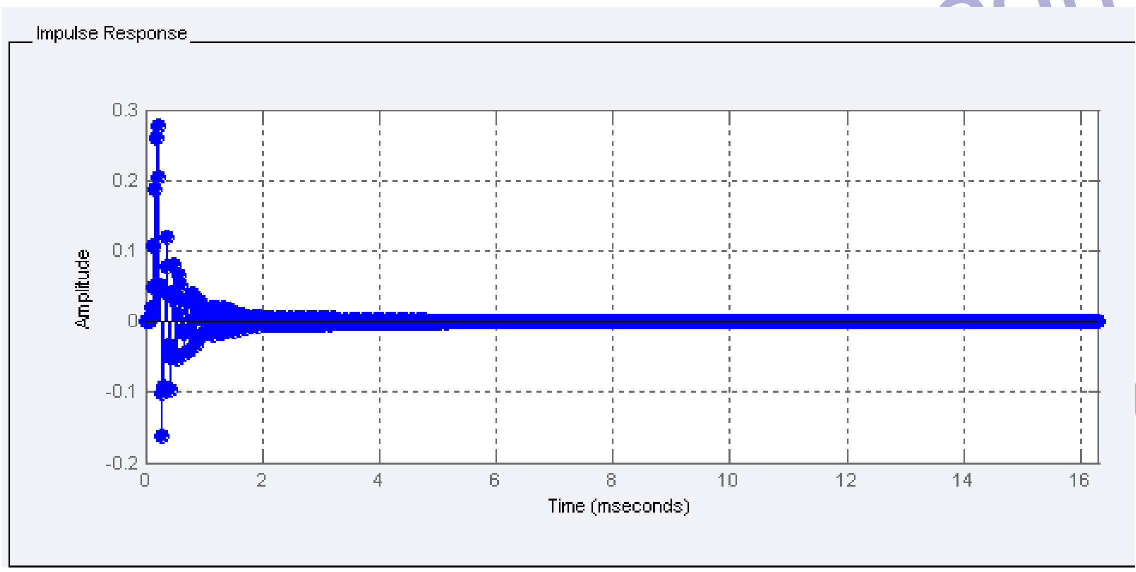
شکل ۶-۳



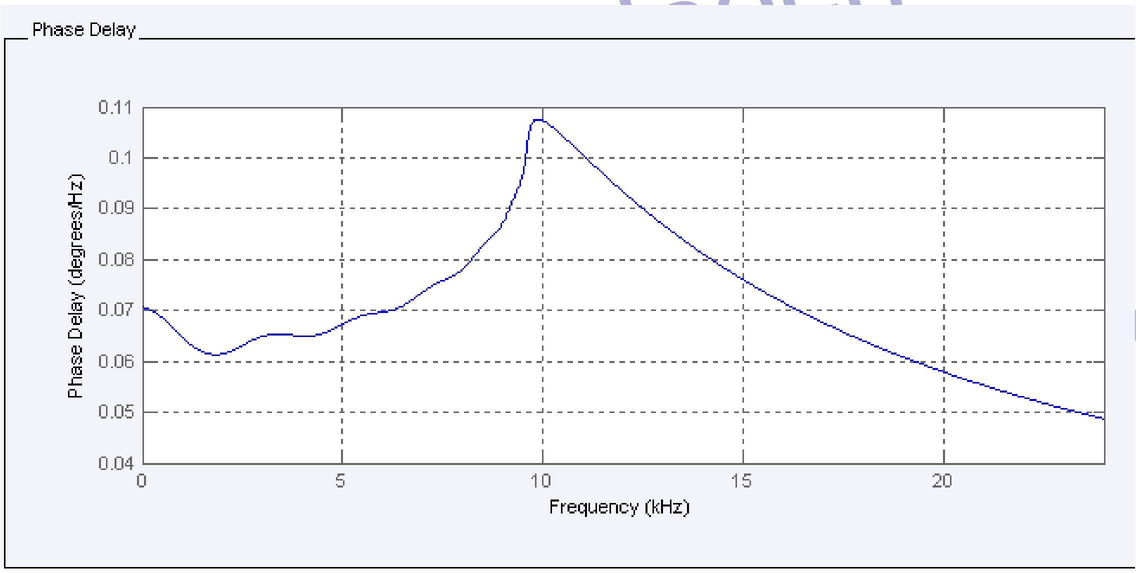
شکل ۳-۷


Step Response
Amplitude
Time (mseconds)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
12
14
16

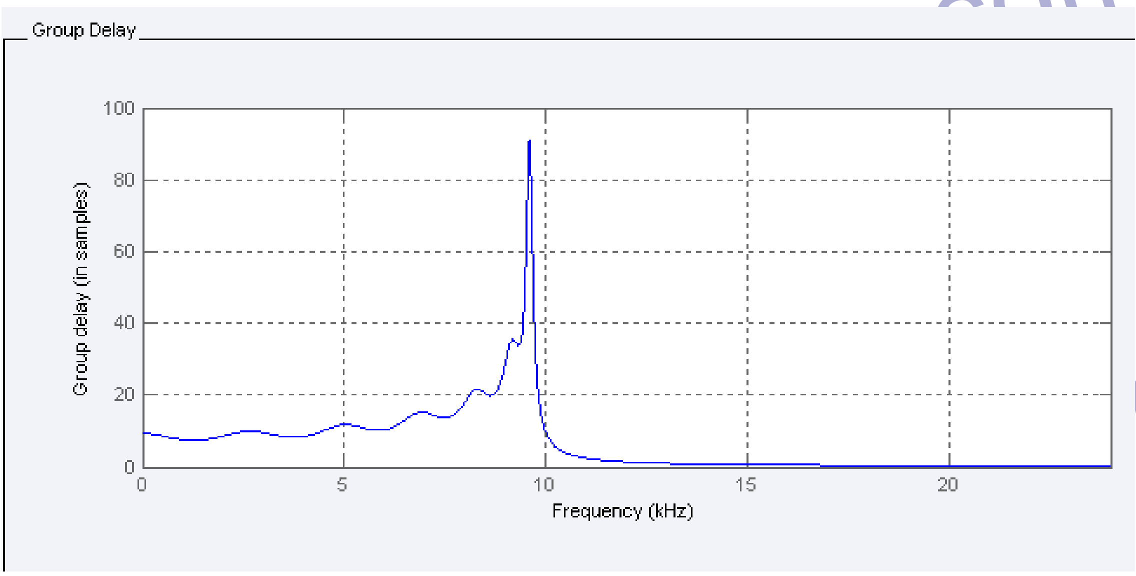
شکل ۸-۳


Impulse Response
Amplitude
Time (mseconds)
0.3
0.2
0.1
0
-0.1
-0.2
0
2
4
6
8
10
12
14
16

شکل ۹-۳



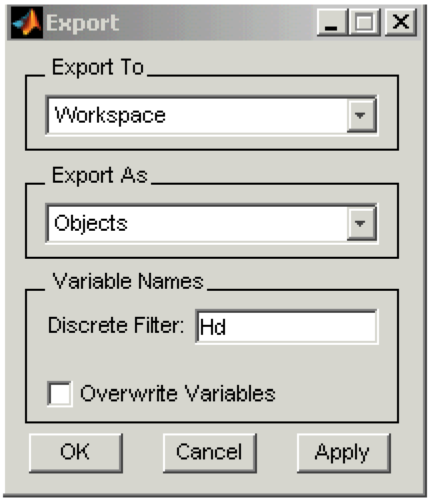
شکل ۱۰-۳



۲-حال می توانیم فیلتر را به محیط work space انتقال داد.

برای اینکار در جعبه ابزار FDA به منوی File → export بروید. بعد از اینکار می توان از فیلتر طراحی شده برای فیلتر کردن سیگنال مورد نظر استفاده کرد.

شکل ۱۱-۳



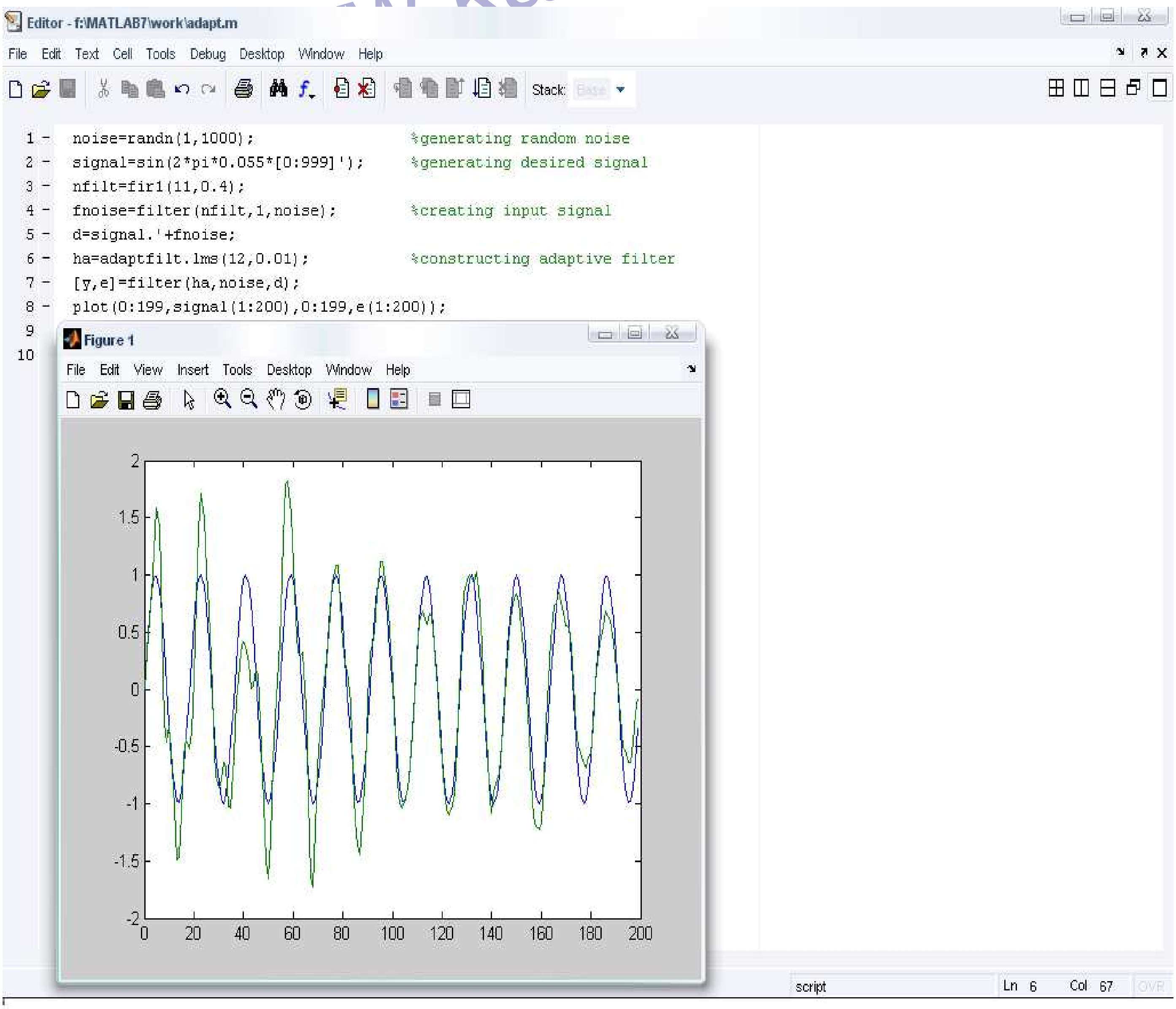
حال ما فیلتر طراحی شده را در متغیر Hd داریم و می توانیم به کمک دستور Filter به هدف خود برسیم.

۳-۴ توانایی های MATLAB در فیلترهای وفقی

در فصل دو در باره فیلترهای وفقی و الگوریتم های مربوط بحث هایی کردیم، در این بخش می خواهیم مطالب گفته شده را در MATLAB پیاده سازی نماییم. در این نرم افزار توابع متعددی برای الگوریتم های متفاوت فیلتر وفقی وجود دارد. توابعی مانند Adapt Filt.Lms، Adapt Filt.Rls و ... برای پیاده سازی الگوریتم ها پیش بینی شده اند. در ادامه در قالب یک مثال کاربرد Adapt Filt.Lms را نشان می دهیم.

مثال ۶: در این مثال ابتدا نویز تصادفی را تولید کرده و با سیگنال سینوسی مورد نظر ترکیب می کنیم. هدف بازسازی سیگنال سینوسی از مجموع نویز و سیگنال است. همانطور که در بخش ۲-۲-۷ دیدیم، ورودی فیلتر وفقی در کاربرد حذف نویز، سیگنالی همبسته با نویز تصادفی می باشد. بنابراین برای ساخت سیگنال مذکور، نویز را از یک فیلتر FIR عبور می دهیم. فیلتر وفقی استفاده شده در این مثال از مرتبه ۱۲ بوده و دارای step-size برابر ۰/۰۱ می باشد.

شکل ۱۲-۳



در مثال بالا ما stepsize را برابر ۰/۰۱ انتخاب کردیم. همانطور که قبلاً دیدیم مقدار μ، پایداری و سرعت همگرایی را تحت تأثیر قرار می داد و ما می توانیم با تغییر آن این قضیه را امتحان کنیم. البته MATLAB برای این مهم تابعی دارد که در زیر به آن آشنا می شویم.

```
Mu=maxstep(halms,d);
```

که halms ساختار فیلتر وفقی است که قرار است سیگنال d را فیلتر کند.

برای اینکه فیلتر همگرایی و پایداری بهتری داشته باشد معمولاً mu بدست آمده از تابع بالا را بر ۳۰ تقسیم می کنند.