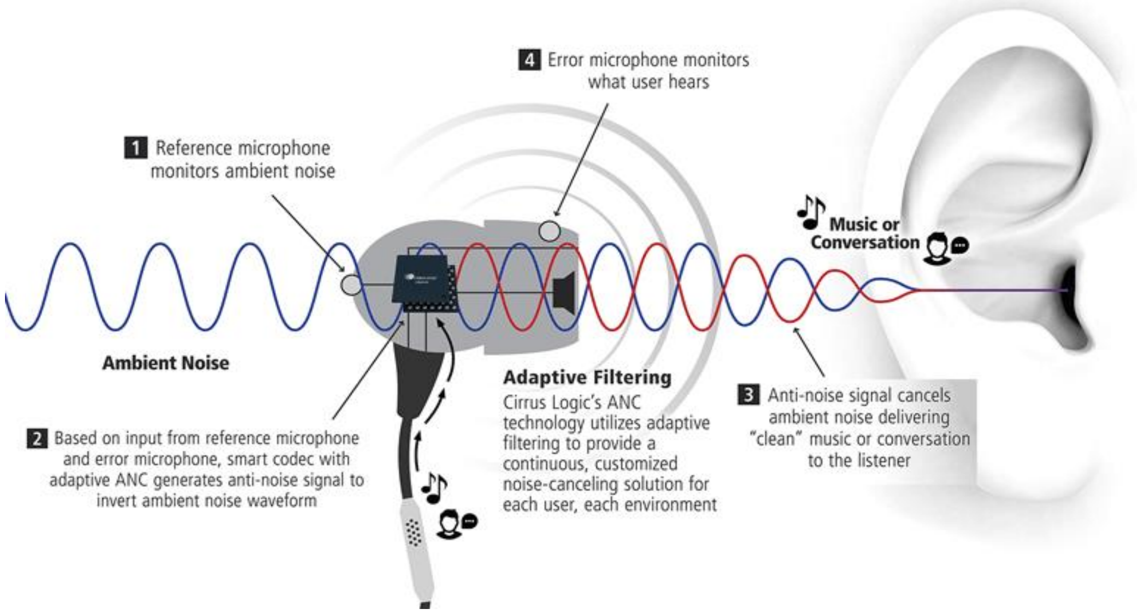
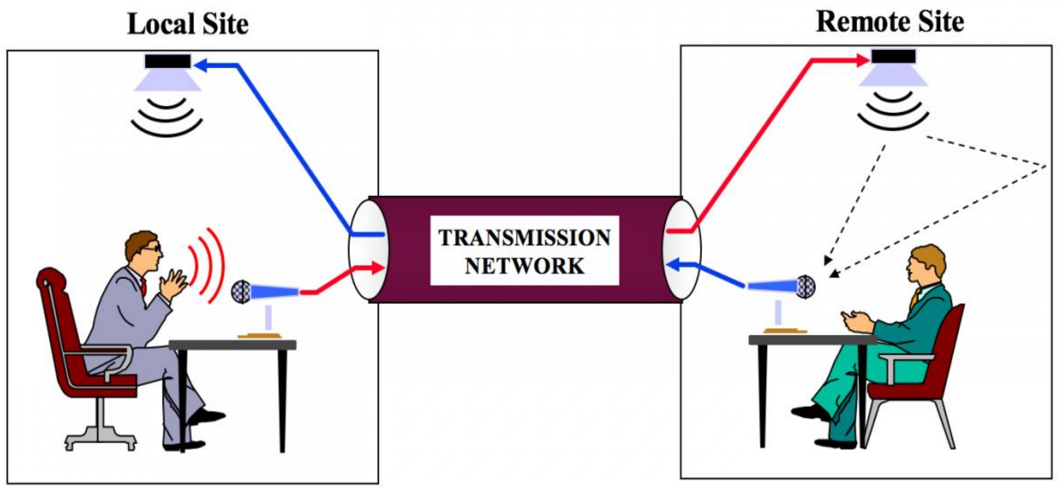


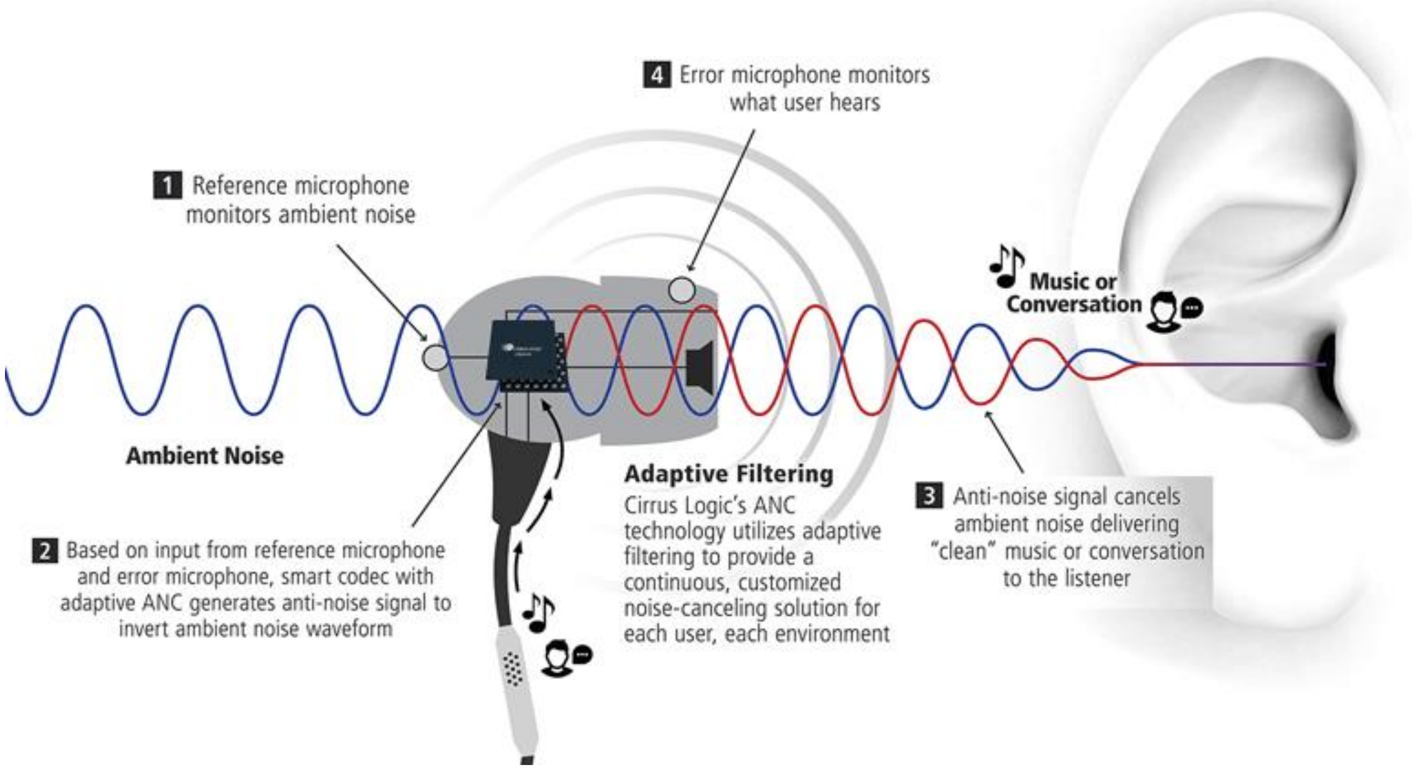


Adaptive Filters And Noise Cancelling

400102222 امیررضا ولائی

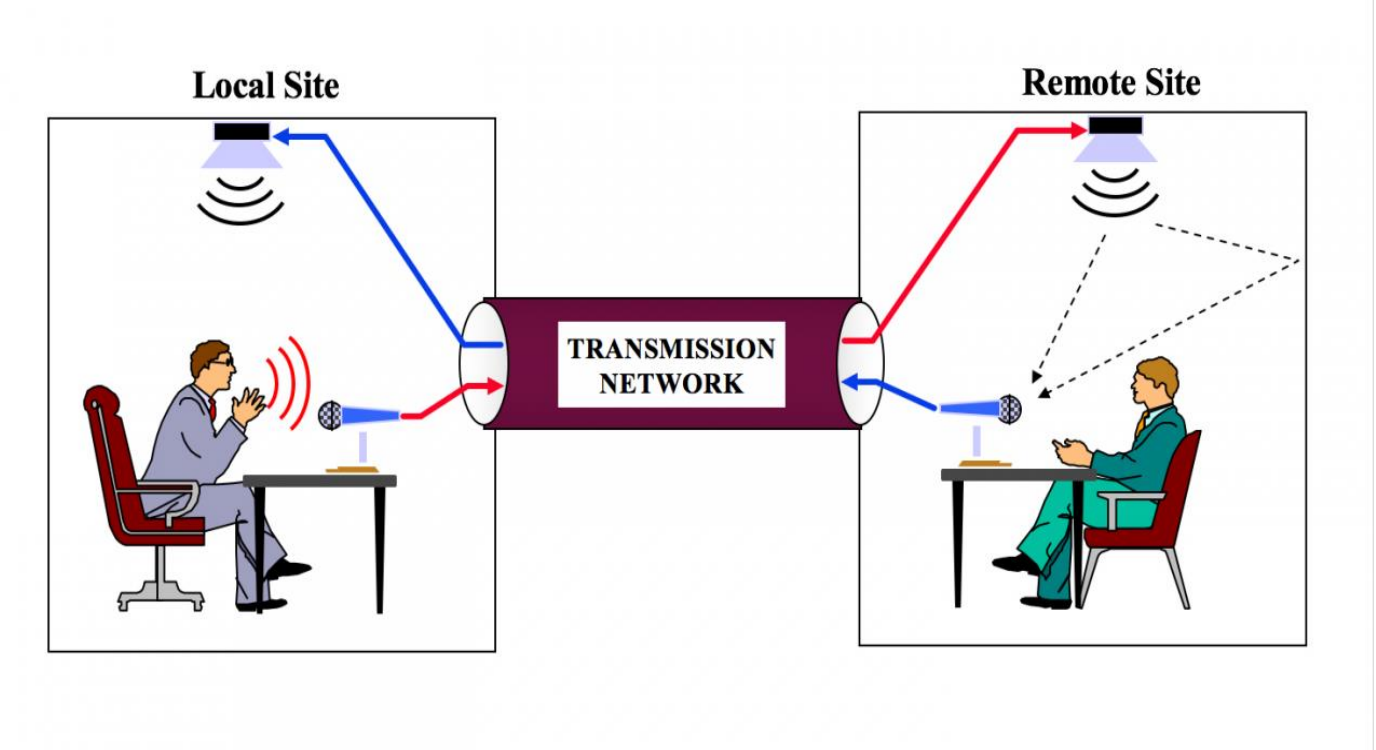
400101937 احسان مریخی

Active noise control (ANC)



یکی از کاربرد های مهم فیلتر های وفقی، ANC یا کنترل نویز فعال است. در این تکنولوژی، با استفاده از میکروفون‌های داخلی درون هدفون، صدای محیط شناسایی شده و با ایجاد موج‌های صوتی با فاز معکوس، سعی می‌شود تا صدای محیط به حداقل رسیده و کیفیت صدای درون هدفون بهبود یابد.

Echo suppression and cancellation



یک کاربرد دیگر از فیلتر های وفقی، حذف اکو از سیگنال است. چالش اصلی برای یک لغو کننده اکو، تعیین ماهیت فیلتری است که باید روی سیگنال انتهای دور اعمال شود، به طوری که شبیه سیگنال پایان نزدیک حاصل شود. این فیلتر در اصل مدلی از بلندگو، میکروفون و ویژگی های صوتی اتاق است. حذف کننده های اکو باید تطبیق پذیر باشند زیرا ویژگی های بلندگو و میکروفون نزدیک به آن معمولاً از قبل مشخص نیست.

سیگنال های تک تن و سیگنال های ویژه ها

برای سیگنال های تک تن پیوسته داریم:

$$x(t) = Ae^{j(\omega t+\phi)}$$

اگر فرض کنیم پاسخ ضربه سیستم $h(t)$ باشد، داریم:

$$y(t) = x(t) * h(t) \to y(t) = \sum_{\theta=-\infty}^{\infty} Ae^{j(\omega(t-\theta)+\phi)}h(\theta) = Ae^{j(\omega t+\phi)} \sum_{\theta=-\infty}^{\infty} Ae^{-j\omega\theta}h(\theta)$$

$$y(t) = \lambda x(t)$$

برای سیگنال های تک تن گسسته داریم:

$$x[n] = Ae^{j(\omega n+\phi)}$$

اگر فرض کنیم پاسخ ضربه سیستم $h_{[t]}$ باشد و $x_{[n]}$ متناوب باشد، آنگاه

$$y[n] = x[n] * h[n] \to y[n] = \sum_{\theta=-\infty}^{\infty} Ae^{j(\omega(n-\theta)+\phi)}h[\theta] = Ae^{j(\omega n+\phi)} \sum_{\theta=-\infty}^{\infty} Ae^{-j\omega\theta}h[\theta]$$

$$y[n] = \lambda x[n]$$

بنابراین نتیجه میگیریم به صورت کلی سیگنال های تک تن سیگنال ویژه های سیستم هستند؛ به صورتی که اگر وارد سیستم شوند، خروجی ضریبی از ورودی خواهد بود.

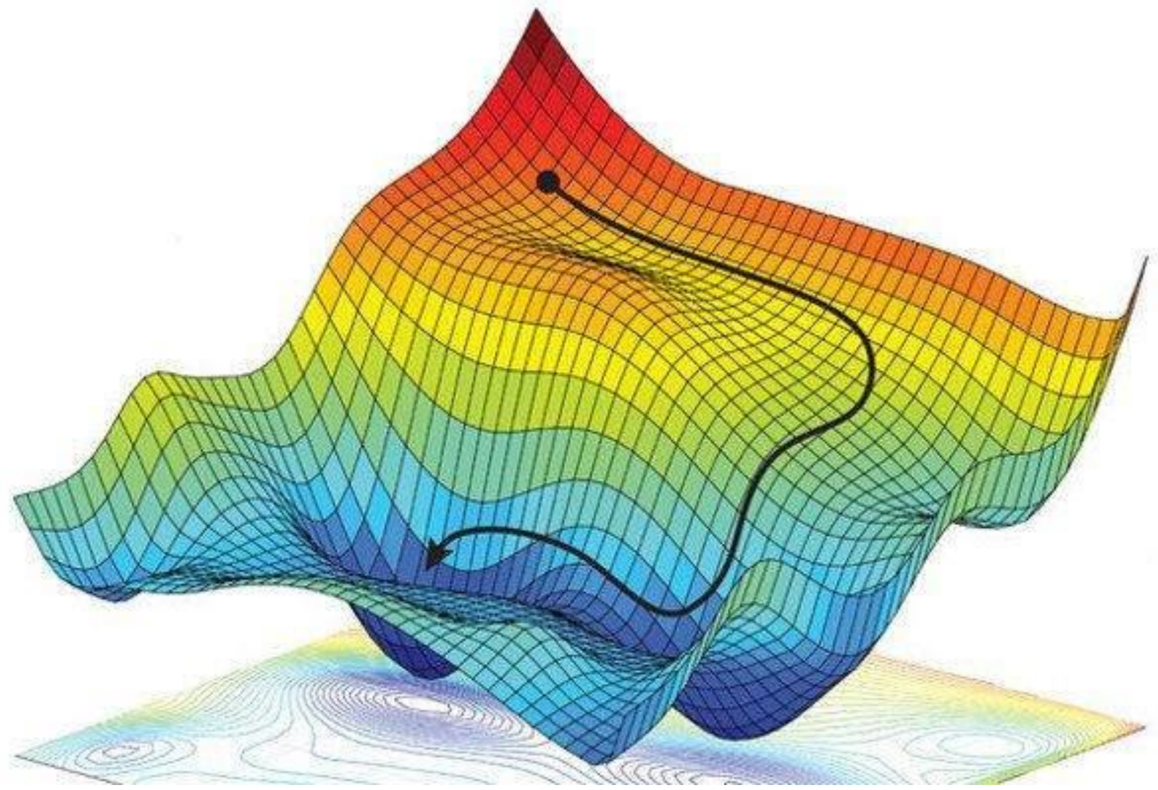
ویژگی سیگنال های ویژه

می دانیم که $e^{j\omega}$ ها سیگنال ویژه های یک سیستم LSI هستند. می توانیم $\cos(2\pi 50n)$ و $\sin(2\pi 50n)$ را به صورت زیر بنویسیم:

$$\cos(2\pi 50n) = \frac{e^{j2\pi 50n} + e^{-j2\pi 50n}}{2}$$
$$\sin(2\pi 50n) = \frac{e^{j2\pi 50n} - e^{-j2\pi 50n}}{2j}$$

در نتیجه می توانیم بگوییم که $\cos(2\pi 50n)$ و $\sin(2\pi 50n)$ نیز سیگنال های ویژه هستند و اگر ورودی ترکیب خطی از آن ها باشد، بنا به اصل *Super position* خروجی نیز ترکیب خطی از خروجی های آن ها خواهد بود و تنها با دو ضریب می‌توانیم خروجی را با استفاده از پایه های $\sin(2\pi 50n)$ و $\cos(2\pi 50n)$ بنویسیم.

Gradient Descent



Gradient Descent

الگوریتم (GD) یا Gradient descent یکی از الگوریتم های مهم در مسئله یافتن پاسخ بهینه سازی برای توابع Convex هست. هرچند تعمیم هایی از این الگوریتم برای توابع غیر محدب نیز استفاده می‌شود که در آن مسائل بجای یک پاسخ بهینه برای هر نقطه شروع، ممکن است به نقاط بهینه مختلفی دست پیدا کنیم.
یک تابع در صورتی محدب یا Convex می‌باشد که رابطه ریاضی زیر برقرار باشد:

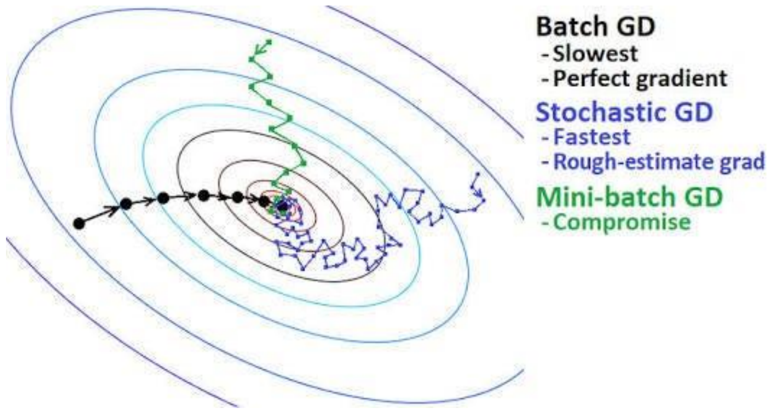
$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y) \quad \forall x, y \in \mathbb{R}^n, \alpha \in [0,1]$$

یا به زبان ساده، اگر برای هر دو نقطه x, y در تابع f، نقطه میانی z وجود داشته باشد که مقدار تابع در آن نقطه کمتر از میانگین مقدار تابع در x, y باشد، آنگاه تابع f محدب است.

می‌دانیم که گرادیان یک تابع در هر نقطه، جهت بیشینه را نشان می‌دهد. بنابراین اگر در هر نقطه، به جای حرکت در جهت گرادیان، در جهت عکس گرادیان حرکت کنیم، مقدار تابع کمتر می‌شود. این روش به روش کاهش گرادیان یا Gradient Descent معروف است.

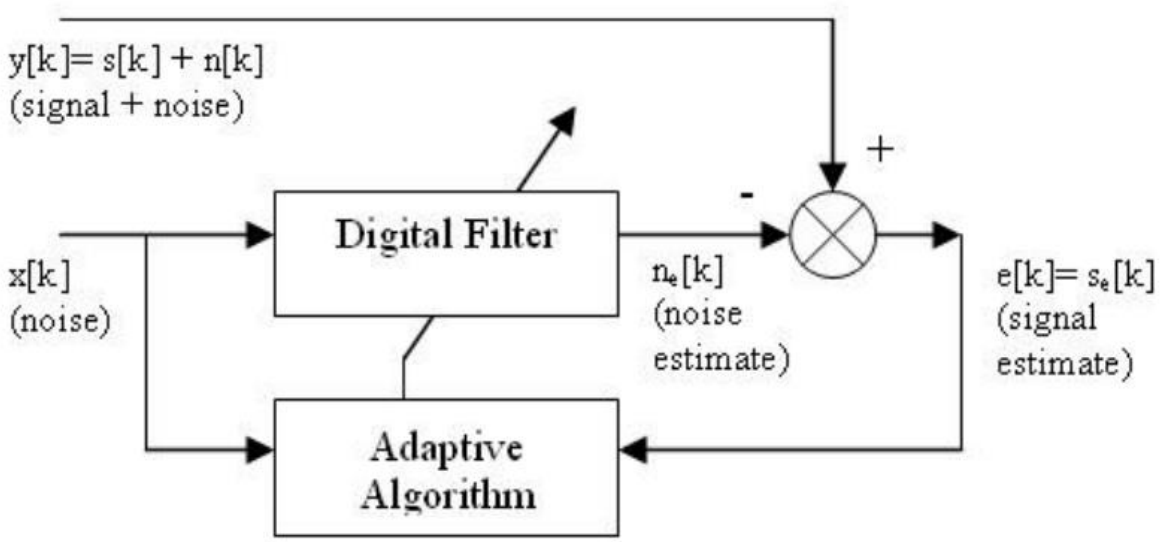
Stochastic Gradient Descent

در این الگوریتم، در هر iteration، یک نمونه از داده ها را انتخاب می‌کنیم و برای آن نمونه، گرادیان را محاسبه می‌کنیم و در جهت عکس آن حرکت می‌کنیم. این الگوریتم برای مسائلی که داده ها بسیار زیاد هستند و یا محاسبه گرادیان برای تمام داده ها بسیار زمان بر است، استفاده می‌شود.
همچنین می‌توانیم به جای انتخاب تمام data set، بخش کوچکی از آن را انتخاب کنیم که به این الگوریتم نیز می‌گویند mini-batch gradient descent.

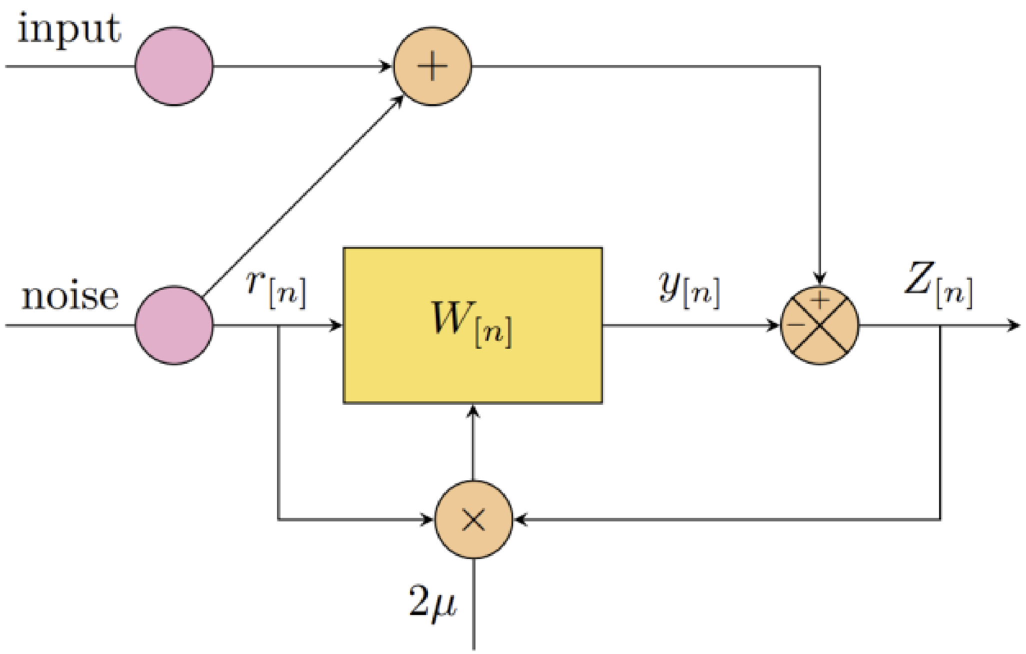


بلوک دیاگرام فیلتر وفقی برای فیلتر نویز برق شهری

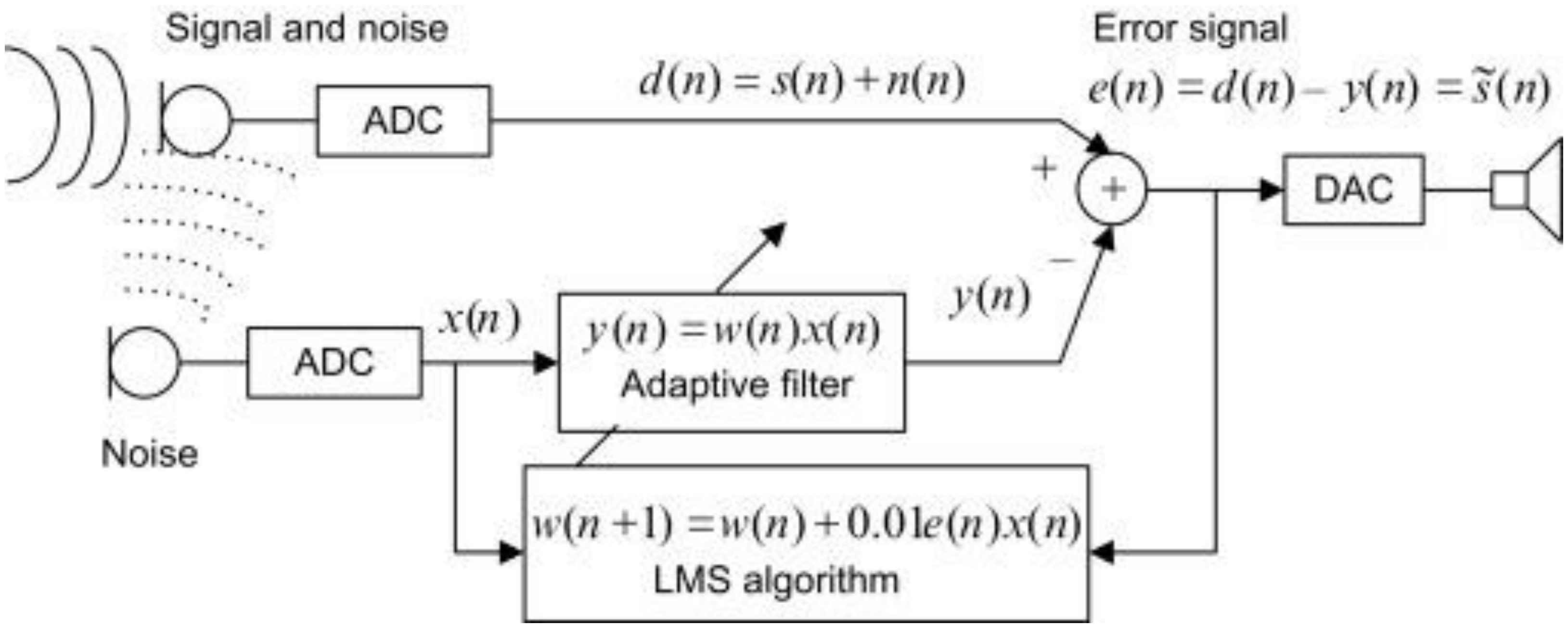
بلوک دیاگرام فیلترهای وفقی با کاربرد حذف نویز به صورت کلی به شکل زیر است:



اگر بخواهیم الگوریتم حذف نویز را به شکل بلوک دیاگرام رسم کنیم، بلوک دیاگرام کلی فیلتر وفقی به شکل زیر در می‌آید:



گزارش پروژه فیلتر های وفقی با هدف حذف نویز



حذف نویز با فیلتر وفقی و الگوریتم LMS

مقدار $\mathbb{E}(e(n))$

مقدار $\mathbb{E}(e(n))$ با توجه به بلوک دیاگرام موجود برابر سیگنال ورودی بدون نویز است. یعنی:

$$\mathbb{E}(e(n)) = \mathbb{E}(d(n) - y(n)) = \mathbb{E}(d(n)) - \mathbb{E}(y(n))$$

به عبارتی، انتظار داریم در صورت همگرایی الگوریتم GD، سیگنال $\mathbb{E}(e(n))$ به سیگنال ورودی بدون محتوی نویز هم‌گرا شود.

۶ حذف نویز سینوسی 50 هرتز از وویس ریکورد شده با روش LMS

توضیح کلی الگوریتم پیاده سازی شده : در اینجا به جای استفاده از اختلاف سیگنال بدون نویز و سیگنال تخمین به عنوان تابع خطا از خود سیگنال تخمین به عنوان خطا استفاده میکنیم. هدف مینیمم کردن این تابع خطا می‌باشد که از الگوریتم LMS کمک میگیریم تا به این هدف برسیم. در این الگوریتم یه تابع وزن داریم که در هر مرحله نویز رفرنس را آپدیت میکند تا شبیه به نویز روی سیگنال اصلی شود. در هر مرحله تابع وزن خودش را با کمک تابع خطا آپدیت میکند تا به حالت اپتیمال برسد.

تاثیر مقدار μ روی همگرایی و دقت : با انتخاب بزرگتر مقدار μ سرعت همگرایی بیشتر میشود اما دقت آن کمتر میشود. دلیل این اتفاق این است که تابع وزن سریعتر میتواند به مقدار اپتیمال نزدیک شود اما وقتی به آن نزدیک میشود به خاطر بزرگ بودن μ نمیتواند خود را به اپتیمال ترین حالت ممکن برساند. همچنین اگر μ بسیار کوچک انتخاب شود دفت همگرایی زیاد میشود اما سرعت همگرایی کم میشود.