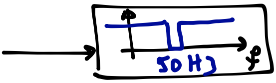
بسمه تعالی

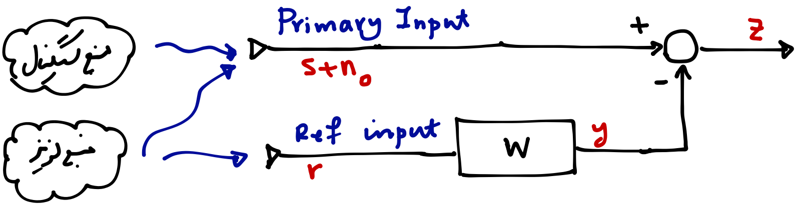
درس فیلتر وفقی

جلسه اول: ۹۶/۶/۲۶

— مسأله حذف هوم $50\,Hz$ برق شهر: ← مقاله Widrow 1975

روش کلاسیک ←





— توجه: r و n_0 یکی نیستند ولی منشأ یکسانی دارند، در نتیجه تابع تبدیلی وجود دارد که r را به n_0 تبدیل کند.

$$z = s + n_0 - y$$

— W باید طوری حساب شود که z تا حد امکان شبیه s شود، لیکن $E\{(z-s)^2\}$ حداقل شود. — مشکل در این است که s را نمی‌دانیم.

<u>قضیه</u>: حداقل کردن انرژی سیگنال $z-s$ معادل است با حداقل کردن انرژی سیگنال z.

<u>اثبات</u>:

$$z = s + n_0 - y \Rightarrow z^2 = s^2 + (n_0-y)^2 + 2s(n_0-y) \xrightarrow{\text{ناهمبستگی سیگنال و نویز}}$$

$$\Rightarrow E\{z^2\} = \underbrace{E\{s^2\}}_{\text{مستقل از فیلتر } W} + E\{(\underbrace{n_0-y}_{z-s})^2\}$$

بنابراین حداقل کردن $E\{z^2\}$ معادل است با حداقل کردن $E\{(z-s)^2\}$.

← صورت مسئله: فیلتر w را طوری تعیین کنید که $E\{z^2\}$ حداکثر شود.

← فرض: فیلتر w، FIR باشد:

$$y(n) = w_0\, r(n) + w_1\, r(n-1) + \cdots + w_{M-1}\, r(n-M+1) \Rightarrow y(n) = \underline{w}^T \cdot \underline{r}(n)$$

$$\underbrace{\underline{w} \triangleq \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{bmatrix}}_{M\times 1} \qquad \underbrace{\underline{r}(n)}_{M\times 1} = \begin{bmatrix} r(n) \\ r(n-1) \\ \vdots \\ r(n-M+1) \end{bmatrix}$$

← هدف: تعیین بردار $\underline{w}$ طوری که تابع $f(\underline{w}) \triangleq E\{z^2(n)\}$ حداکثر شود.

روش Steepest Descent (S.D.) برای حداکثر کردن یک تابع چند متغیره $f(\underline{x})$:

$$f(x_1, \dots, x_n) \rightarrow \underline{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \rightarrow f(\underline{x})$$

~~نکته~~ قضیه: گرادیان تابع در هر نقطه جهتی را نشان می‌دهد که تابع در آن جهت سریع‌ترین افزایش را دارد. ← جهت steepest ascent

منفی گرادیان « « « « « « « « « کاهش « « ← جهت steepest descent

← اثبات:

$$f(\underline{x}+\underline{\delta}) - f(\underline{x}) \quad \text{،} \quad \|\underline{\delta}\| \triangleq \text{ثابت و کوچک}$$

$$f(\underline{x}+\underline{\delta}) - f(\underline{x}) \simeq \frac{\partial f}{\partial x_1}\delta_1 + \frac{\partial f}{\partial x_2}\delta_2 + \cdots + \frac{\partial f}{\partial x_n}\delta_n = \langle \underline{\delta}, \underline{\nabla} f(\underline{x}) \rangle$$

$$= \underline{\delta}^T \cdot \underline{\nabla} f(\underline{x})$$

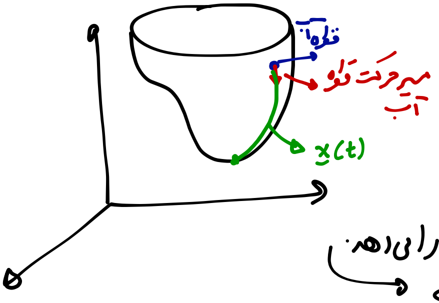
← که در آن:

$$\underline{\nabla} f(\underline{x}) \triangleq \begin{bmatrix} \partial f / \partial x_1 \\ \vdots \\ \partial f / \partial x_n \end{bmatrix}$$

$$\Rightarrow f(\underline{x}+\underline{\delta}) - f(\underline{x}) = \langle \underline{\delta}, \nabla f(\underline{x})\rangle = \underbrace{\|\underline{\delta}\|}_{\text{ثابت}} \cdot \underbrace{\|\nabla f(\underline{x})\|}_{\text{مستقل از } \theta} \cdot \cos\theta$$

$\theta = 0 \rightarrow$ بیشترین افزایش.

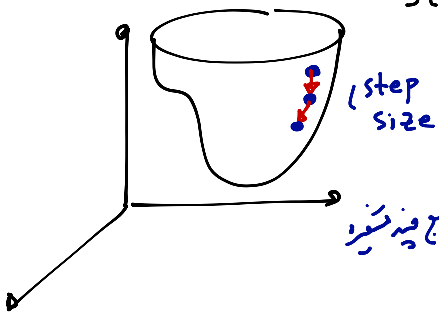
$\theta = 180 \rightarrow$ " کاهش.



مسیر حرکت قطره آب:

$$-\nabla f(\underline{x}(t)) = \frac{d}{dt}\underline{x}(t)$$

حل این معادله دیفرانسیل مسیر حرکت قطره آب را می‌دهد → steepest descent path



تقریب گسسته از مسیر حرکت قطره آب:

(step size) یک عدد مثبت کوچک

$$\boxed{\underline{x}_{k+1} = \underline{x}_k - \mu \nabla f(\underline{x}_k)}$$

الگوریتم S.D. برای حداقل کردن یک تابع چند متغیره

استفاده از الگوریتم بالا در مسئله خودمان:

$$f(\underline{w}) = E\{z^2(n)\} = E\{[x(n)-y(n)]^2\} = E\{[x(n)-\underline{w}^T\underline{r}(n)]^2\}$$

$$= E\{[x(n) - w_0 r(n) - w_1 r(n-1) - \dots - w_{M-1} r(n-M+1)]^2\}$$

همان داخل کروشه بالا

$$\frac{\partial f}{\partial w_i} = -2E\{[\underbrace{-//-}_{z(n)\ \text{همان}}]\, r(n-i)\} = -2E\{z(n)\, r(n-i)\}$$

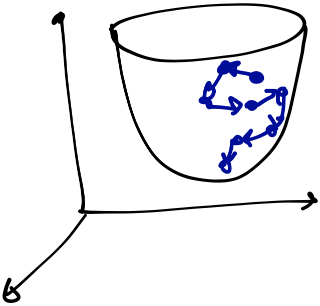
دو trick دیگر:

① هر iteration از S.D. را با آمدن یک نمونه جدید سیگنال بزنیم ($k=n$ می‌گیریم)

اندیس iteration — اندیس نمونه‌های زمانی سیگنال

② از علامت $E\{\ \}$ صرف‌نظر کنیم و به جای آن μ خیلی کوچک بگیریم ← در طول زمان به نوعی متوسط‌گیری می‌کند.

در واقع:



⟸ الگوریتم نهایی برای حذف نویز:

با رسیدن هر ورودی جدید (هر $x(n)$ و $r(n)$ جدید).

$$y(n) = \underline{w}_n^T \underline{r}(n) \longrightarrow \text{تخمین نویز}$$

$$z(n) = x(n) - y(n) \longrightarrow \text{سیگنال خروجی}$$

$$\underline{w}_{n+1} = \underline{w}_n + \mu\, z(n)\, \underline{r}(n) \longrightarrow \text{آپدیت فیلتر}$$

ضریب ۲ را در μ مستهلک کرده‌ایم.

نکاتی که از اجرا الگوریتم بالا دیده شد:

misadjustment

① $\mu \uparrow$ ⟸ سرعت همگرایی $\uparrow$ ولی خطای نهایی $\uparrow$

② در این روش برخلاف روش کلاسیک خود فیلتر را طراحی نمی‌کنیم بلکه الگوریتمی را که فیلتر باید خودش را در حین کار طراحی کند، طراحی می‌کنیم.

③ اگر نویز تغییر کند (مثلاً 50Hz شود 60Hz) بازهم سیستم ما کار می‌کند. ولی notch filter نه.

④ اگر محیط تغییر دائم (50Hz مرتباً نوسان داشته باشد) سیستم ما آن را تعقیب می‌کند.

⑤ اگرچه فیلتر خطی برای $\underline{w}$ گذاشتیم، سیستم در کل خطی نیست (پاسخ به مجموع دو ورودی لزوماً مجموع پاسخ به آن دو ورودی نمی‌شود). سیستم TI هم نیست.

البته فیلترهای وفقی نه خطی هستند نه تغییرناپذیر با زمان.

اصطلاح: ما AF را خطی می‌گوییم اگر فیلتر آن (در راستای w) خطی باشد.

⑥ تغییرات ممکن است خودش باعث ناپایداری سیستم شود.

برخی مشخصات برای مقایسه الگوریتم‌های مختلف فیلترهای وفقی:

1. Rate of convergence
2. Misadjustment : بعد از همگرایی چقدر با مقدار ایده‌آل فاصله داریم. (فیلتر وینر)
3. Tracking : قابلیت و سرعت تعقیب کردن تغییرات محیط
4. Robustness : اغتشاشات کوچک به خطاهای کمی کوچک منجر شوند.
5. Computational Requirements
   - حجم محاسبات
   - مقدار حافظه مورد نیاز
   - پیچیدگی الگوریتم
6. Numerical properties
   - Stability
   - Accuracy