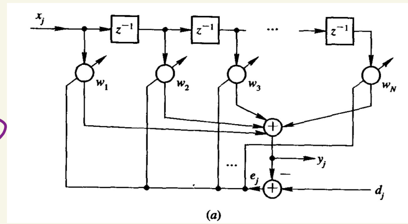
第四章 自适应滤波

1. 概念.

自适应滤波：根据信号环境特性 用递归算法(自适应算法) 自动调整滤波器的脉冲响应，使自适应滤波器输出尽可能接近其期望输出。

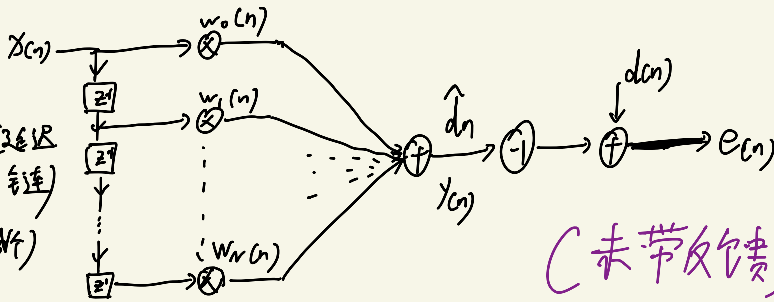
在平稳环境中.迭代次数充分大时.自适应算法给出的结果在某种统计意义上收敛于最佳维纳解. 在非平稳环境中.自适应算法具有一定跟踪输入数据统计特性(慢)变化的能力.

完整结构



(a)

2. 自适应滤波器的结构.



(未带反馈)

(只能算是一个普通滤波器)

(权参数可变)(通过误差来调整)

$$X(n)=\begin{pmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-N+1) \end{pmatrix}$$

输入向量

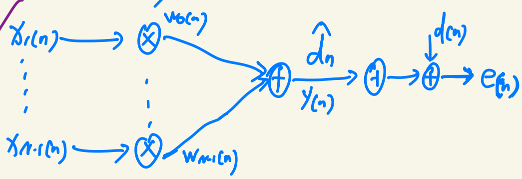
$$X(n)=\begin{pmatrix} x_1(n) \\ x_2(n) \\ \vdots \\ x_{N-1}(n) \end{pmatrix}$$

$$W_n=\begin{pmatrix} w_0(n) \\ w_1(n) \\ \vdots \\ w_{N-1}(n) \end{pmatrix}$$ 权向量

(N-1个输入)

(一个输入, N-1个延迟器)

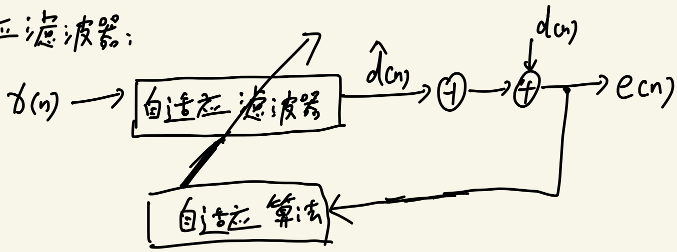
两种不同的输入方式.



输出:

(标量) $y(n) = W^T(n)X(n) = X^T(n)\cdot W(n)$ (内积)

自适应滤波器:



完整带反馈的简略框图.

自适应算法:
- 随机梯度算法 (LMS算法) (误差统计平均最小)
- 最小二乘算法 (RLS算法) (误差时间平均最小)

自适应滤波的应用:

1. 系统辨识

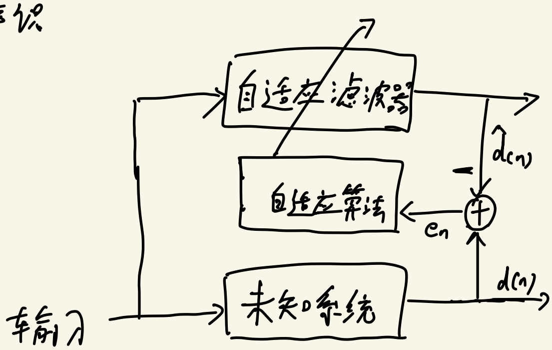
$x(n) \to \boxed{h(m)} \to y(n)$

(白噪声)

$$\Phi_{xy}(m) = \Phi_{xx}(m) * h(m) = \sigma_x^2\delta(m) * h(m) = \sigma_x^2 h(m)$$

$$h(m) = \frac{1}{\sigma_x^2}\cdot\Phi_{xy}(m)$$

2. 自适应辨识

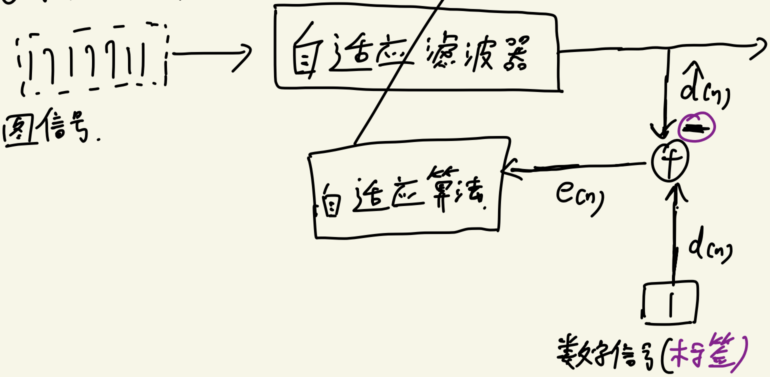


通过足够多的迭代,使误差极小,就可认为两个输出相等,即两个系统近似相等

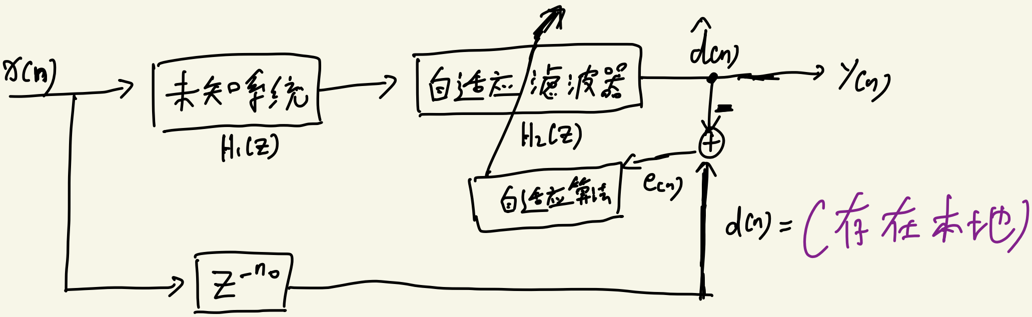
应用
- ① 手写体识别
- ② 逆滤波

自适应辨识的应用:

① 手写体识别



② 逆滤波.



理想信道: $H(z)=1$, $h(n)=\delta(n)$; 因为有延迟, 所以只要求 (理想)

(要求输出=输入, $Y(z)=H(z)X(z)$) $H(z)=z^{-n_0}$, $h(n)=\delta(n-n_0)$

即 $y(n)=x(n)$ 即 $y(n)=x(n-n_0)$

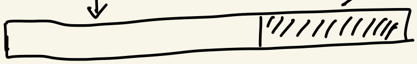
实际:

$H_1(z)=z^{-n_0}$, 做不到

加上 $H_2(z)$ 使得 $H_1(z)\cdot H_2(z)=z^{-n_0}$

待发送的实际数据

训练数据，可存储在接收端



训练结束时，得到训练好的权向量

即使 $H_1(z)\cdot H_2(z)\approx z^{-n_0}$

↓

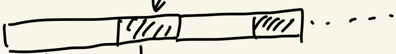
$H_1(z)\xrightarrow{\text{突变}}\tilde{H}_1(z)$，导致 $\tilde{H}_1(z)\cdot H_2(z)$ 偏离 z^{-n_0}

↓

重新训练权向量，使 $\tilde{H}_1(z)\cdot\tilde{H}_2(z)\approx z^{-n_0}$

重复迭代

训练数据



有用信息

辅助识别（波负信道）

算法收敛越快，训练数据长度越短，信道使用率越高。

改进：盲信号处理，不用训练数据。

（可提高信道效率，但是精度不如上述方法高）

自适应算法

1、最速下降法(LMS算法的基础)

$$J(w)=E(e^2(n))=\sigma_d^2-2w^Tp+w^TRw$$

(目标代价函数)

$$w(n+1)=w(n)+\Delta w=?$$

$$J(w+\Delta w)=J(w)+\sum_{i=0}^{N-1}\frac{\partial J}{\partial w_i}\Delta w_i+\frac{1}{2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\Delta w_i\frac{\partial^2 J}{\partial w_i\partial w_j}\Delta w_j+\cdots$$

$$=J(w)+(\Delta w)^T\nabla J(w)+\frac{1}{2}(\Delta w)^T\nabla^2 J(w)(\Delta w)+\cdots$$

(列向量)^T

其中 $\nabla J(w)=-2p+2Rw$ (求导)，$\nabla^2 J(w)=2R$ (二阶导)，R 通常正定，所以 $(\Delta w)^T\nabla^2 J(w)(\Delta w)>0$

若要减小代价函数，即使 $(\Delta w)^T\nabla J(w)<0$

$(J(w+\Delta w)<J(w))$

$\because (\Delta w)^T\cdot\nabla J(w)=\|\Delta w\|\,\|\nabla J(w)\|\cdot\cos\theta$

$\therefore$ 当 $\cos\theta=-1$ 时，$(\Delta w)^T\cdot\nabla J(w)$ 为负值，且最小。

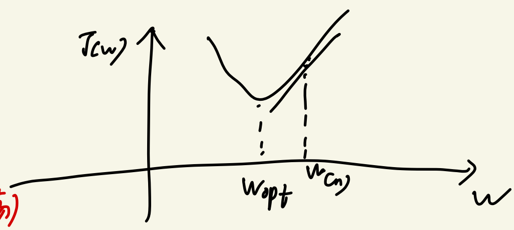
此时 $\boxed{\Delta w=-\nabla J(w)=\boxed{2p-2Rw}}$，$J(w+\Delta w)$ 的减小量最大。

$(\theta=\pi)$

$\therefore$ 负的梯度方向是最速下降方式。

$$w(n+1)=w(n)+\Delta w=w(n)+\mu[-\nabla J(w)]$$

$\mu>0$，称为步长参数。(控制算法收敛速度)



μ 变大，迭代速度变快，但是 μ 过大会导致发散(来回振荡)

μ (可以控制算法的稳定性和收敛速度)

确定一个 μ 的范围，使算法收敛 $(\lim_{n\to\infty}w(n)=w_{opt})$???

$$E(e^2) = J(w) = \sigma_d^2 - 2w^T P + w^T R w$$

对w求导

$$\nabla J(W(n)) = 2R\,W(n) - 2P$$

$$W(n+1) = W(n) + 2\mu[P - R\,W(n)]$$

$$= (I - 2\mu R)\,W(n) + 2\mu P$$

$$(2\mu R\,W_{opt})$$

$$W_{opt} = R^{-1}P$$

$$\Downarrow$$

$$P = R\,W_{opt}$$

坐标平移:

$$\tilde{W}(n) = W_n - W_{opt}$$

→0　　最优

$$\tilde{W}(n+1) = (I-2\mu R)\,W(n) + 2\mu R\,W_{opt} - W_{opt}$$

$$= (I-2\mu R)\,W(n) - (I-2\mu R)\,W_{opt}$$

$$= (I-2\mu R)(W(n) - W_{opt})$$

$$= (I-2\mu R)\,\tilde{W}(n)$$

$$\vdots$$

$$\tilde{W}(n+1) = (I-2\mu R)^n\,\tilde{W}(0)$$

要求: $n\to\infty$, $\tilde{W}(n+1) \longrightarrow \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$

① $(I-2\mu R)$为1维的时候，只需让其小于1即可，n次方后趋向于0

② $(I-2\mu R)$为2维矩阵时，因其内部数据太多，无法观察，

③ $(I-2\mu R)$转换为对角矩阵，只需考虑对角线元素即可。

核心要对角化

坐标旋转: $V(n) = Q^T\,\tilde{W}(n) = Q^{-1}\,\tilde{W}(n)$　　$(R = Q\Lambda Q^{-1})$

$$\tilde{W}(n) = Q\,V(n)$$

$$Q\,V(n+1) = (I - 2\mu R)\,\tilde{W}(n) = (I-2\mu R)\,Q\,V(n)$$

$$Q^{-1}Q\,V(n+1) = Q^{-1}(I-2\mu R)\,Q\,V(n)$$

$$V(n+1) = Q^{-1}(I-2\mu R)\,Q\,V(n)$$

$$V(n+1) = Q^{-1}(I - 2\mu R)Q\,V(n)$$
$$= (I - 2\mu Q^{-1}RQ)\,V(n)$$
$$= (I - 2\mu\Lambda)\,V(n)$$
$$\vdots$$
$$V(n) = (I - 2\mu\Lambda)^n V(0)$$

要求 $n \to \infty$, $V(n) \to \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$

$$\begin{bmatrix} V_0(n) \\ V_1(n) \\ \vdots \\ V_{N-1}(n) \end{bmatrix} = \begin{bmatrix} (1-2\mu\lambda_0)^n & & & \\ & (1-2\mu\lambda_1)^n & & \\ & & \ddots & \\ & & & (1-2\mu\lambda_{N-1})^n \end{bmatrix} \begin{bmatrix} V_0(0) \\ V_1(0) \\ \vdots \\ V_{N-1}(0) \end{bmatrix}$$

在主坐标系下，变量不再耦合.

k 与 $k-1$ …. 无关.
解耦

$$V_k(n) = (1-2\mu\lambda_k)^n V_k(0)$$
$$k = 0, 1 \cdots N-1$$

收敛要求：$|1-2\mu\lambda_k| < 1$

$$-1 < (1-2\mu\lambda_k) < 1$$
$$-2 < -2\mu\lambda_k < 0$$
$$1 > \mu\lambda_k > 0$$
$$0 < \mu < \frac{1}{\lambda_k}, \quad k = 0 \cdots N-1$$

满足所有 λ：$0 < \mu < \dfrac{1}{\lambda_{max}}$, 此时算法收敛.

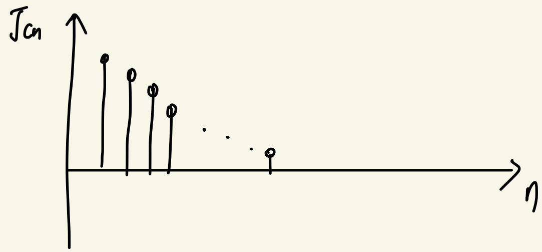
此时, $\lim\limits_{n\to\infty} V(n) = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$

即 $\lim\limits_{n\to\infty} Q^T(W(n) - W_{opt}) = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \Longrightarrow \lim\limits_{n\to\infty} W(n) = W_{opt}$

$$J(n) = J_{min} + \sum_{k=0}^{N-1} \lambda_k (v_k(n))^2$$
$$= J_{min} + \sum_{k=0}^{N-1} \lambda_k (1-2\mu\lambda_k)^{2n} v_k^2(0)$$

指数曲线的加权和

曲线 $J(n) \sim n$ 称为学习曲线



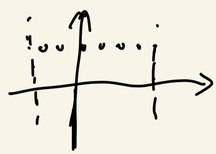
$$J(n) \le \left[\frac{cond(R)-1}{cond(R)+1}\right]^2 J(n-1)$$

$cond(R)$ 是相关矩阵 R 的条件数.

$cond(R) = \frac{\lambda_{max}}{\lambda_{min}} \ge 1$. 越小越好.

越小收敛越快.

当 $R = \begin{bmatrix} \ddots & & 0 \\ & \ddots & \\ 0 & & \ddots \end{bmatrix}$



∴ $x(n)$ 是白噪声 $\lambda_{max} = \lambda_{min}$, 收敛最快

λ_{max} 对应的特征向量 q_{max} 下标 和 λ_{min} 对应的特征向量 q_{min} 下标 (与值无关)

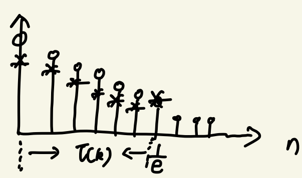
给出了均方误差的最大 (λ_{max}) 和最小 (λ_{min}) 曲线方向.

等高线(有相同的均方误差)越扁, 收敛速度越慢.

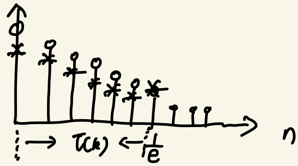
越像圆, 收敛速度越快.

$J(n)$ 下降的等效时间常数, τ_k 为下降为初始值的 $\frac{1}{e}$ 所用的时间(迭代次数)

即用 $e^{-\frac{n}{\tau_k}}$ 近似 $(1-2\mu\lambda_k)^{2n}$



即用 $e^{-\frac{n}{\tau_k}}$ 近似 $(1-2\mu\lambda_k)^{2n}$



$$(1-2\mu\lambda_k)^2 = e^{-\frac{1}{\tau_k}} \approx \underline{1-\frac{1}{\tau_k}} \qquad (e^x \approx 1+x)$$

当μ很小时，

$$(1-2\mu\lambda_k)^2 \approx \underline{1-2\cdot 2\mu\lambda_k}$$

$$\therefore \tau_k \approx \frac{1}{4\mu\lambda_k}$$

学习曲线是N个指数曲线（对应不同的λ_k）的加权和.

$$\therefore \tau \approx \frac{1}{4\mu\lambda_{min}}$$

$$\because 0<\mu<\frac{1}{\lambda_{max}}$$

$$\therefore \frac{1}{\mu} > \lambda_{max} \qquad \therefore \tau \approx \frac{1}{4\lambda_{min}}\cdot\frac{1}{\mu} > \frac{1}{4}\frac{\lambda_{max}}{\lambda_{min}} = \frac{1}{4}\mathrm{cond}(R)$$

如果相关矩阵R的特征值动态范围很大（R为病态矩阵）.
（条件数很大）

收敛很慢.

$Ax=b$

$x=A^{-1}b$

$A \leftarrow A+\underbrace{\Delta A}_{\text{误差}}$

$x \leftarrow x+\Delta x$

如果 $\|\Delta A\|$ 小, $\|\Delta x\|$ 也小, 良态

$\|\Delta A\|$ 小, $\|\Delta x\|$ 越大, 病态 （蝴蝶效应）

（越大矩阵，越容易出现病态）

$$W(n+1) = W(n) + \mu[-\nabla J(W(n))]$$

$$= W(n) + 2\mu[P - RW(n)]$$ ，n是迭代次数，不是时间，不是一个自适应算法。

仅仅是 $W_{opt} = R^{-1}P$ 的迭代实现。

都要知道统计特性．P．R

① $W_{opt} = R^{-1}P$ ，一次性需要矩阵求逆

② $W(n+1) = W(n) + 2\mu(P - RW(n))$ 　多次迭代，无矩阵求逆．

4.3 最小均方(LMS)算法

$$P = E[x(n)d(n)]$$

$$R = E[x(n)x^T(n)]$$

为了计算简单，用瞬时值代替平均。

梯度的瞬时估计为：

$$\hat{\nabla} J(w) = 2x(n)x^T(n)\hat{w}(n) - 2x(n)d(n)$$

拿掉取平均

对比 $\nabla J(w) = 2Rw(n) - 2P$

$$= 2E[x(n)x^T(n)]w(n) - 2E[x(n)d(n)]$$

$$\hat{\nabla} J(w) = -2x(n)[d(n) - \underbrace{x^T(n)\hat{w}(n)}_{\hat{d}(n) = y(n)}]$$

$$= -2x(n)e(n)$$

LMS:

$$y(n) = \hat{w}^T(n)x(n)$$ N次乘, N-1次加

$$e(n) = d(n) - y(n)$$ 1次加

$$\hat{w}(n+1) = \underset{\text{列向量}}{\hat{w}(n)} + 2\mu \underset{\text{列向量乘子}}{x(n)e(n)}$$ (n是时间，所以是自适应算法)

N+1次乘，N次加。

总：2N+1 次乘，2N 次加 $\boxed{O(N)\ \text{时间复杂度}}$

优：计算量小

缺：收敛慢。

在一定条件下，如果 $0 < \mu < \frac{1}{\lambda_{max}}$，则 LMS 算法均方收敛：

$$\lim_{n\to\infty} E[w(n)] = w_{opt} \qquad \lim_{n\to\infty} E[\|\hat{w}(n) - w_{opt}\|^2] = \text{常数}\ a$$

$\because \lambda_{max} < tr[R] = \lambda_0 + \lambda_1 + \cdots + \lambda_{N-1} \quad (\lambda > 0)$

$\therefore$ μ 也可保守地取为

$0 < \mu \le \dfrac{1}{tr[R]}$ ， $tr[R]$ 是输入矢量 $x(n)$ 的输入功率

一定条件：假定 $x(n)$、$x(n+1)$、$x(n-1)\cdots$ 互不相关、$\longrightarrow$ 做不到

$x(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-N+1) \end{bmatrix}$ $\quad x(n+1) = \begin{bmatrix} x(n+1) \\ x(n) \\ \vdots \\ x(n-N+2) \end{bmatrix}$ $\longrightarrow$ $x(n)$、$x(n+1)$ 是相关的

若 $x(n) = \begin{bmatrix} x_0(n) \\ x_1(n) \\ \vdots \\ x_{N-1}(n) \end{bmatrix}$ $\quad x(n+1) = \begin{bmatrix} x_0(n+1) \\ x_1(n+1) \\ \vdots \\ x_{N-1}(n+1) \end{bmatrix}$

此时 $x(n)$、$x(n+1)$ 可能不相关

(白噪声) 要求很苛刻.

两种不同的输入方式.

LMS 算法的均方误差：

$$J(n) = J_{min} + J_{tr}(n) + J_{ex}(\infty)$$

J_{min} 是理论上的最小均方误差.

$J_{tr}(n)$ 是瞬态均方误差. $J_{tr}(n=\infty)=0$.

$J_{ex}(\infty)$ 是稳态超量均方误差，由梯度噪声产生的 (用 $\hat{\nabla} J$ 代替 ∇J)

($\hat{\nabla} J = \nabla J +$ 噪声)

在稳态时 $(n=\infty)$

$$J(\infty) = J_{min} + J_{ex}(\infty) > J_{min}$$

性能变差 $\longrightarrow$ 次最优解.

μ 很小的时候 $J_{ex}(\infty) \approx \mu \cdot tr[R] \cdot J_{min}$

$\therefore$ μ 越小，$J_{ex}(\infty)$ 越小. 但收敛速度变慢.

失调量定义为

$$M = \frac{J_{ex}(\infty)}{J_{min}} \approx \mu\, tr[R]$$

在标准LMS算法中

收敛慢

$$\hat{W}(n+1) = \hat{W}(n) + 2\mu e(n) x(n)$$

改进：① 归一化的LMS算法

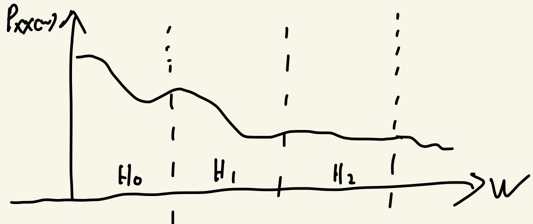
$$\hat{W}(n+1) = \hat{W}(n) + 2\mu(n) e(n) x(n), \quad \mu(n) = \frac{\tilde{\mu}}{\|x(n)\|}, \quad 0<\tilde{\mu}<1$$

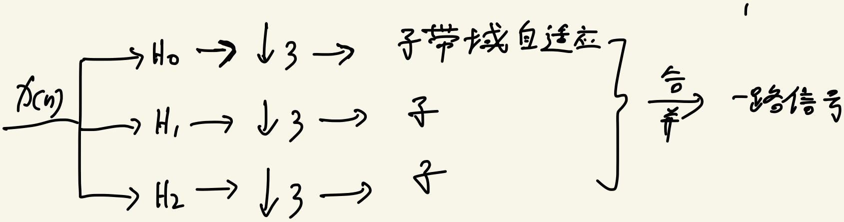
② 对输入矢量 $x(n)$ 作变换，去掉其相关性（白化），减小特征值的动态范围，从而提高收敛速度。

③ 子带域自适应。

可以证明：功率谱的最大值与最小值之比相当于相关矩阵的最大特征值与最小特征值之比。

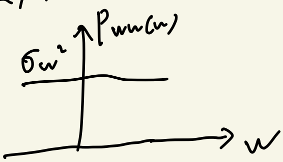
$$\frac{P_{xx}(\omega)_{max}}{P_{xx}(\omega)_{min}} \approx \frac{\lambda_{max}}{\lambda_{min}}$$





频率分割，可减小特征值的波动范围。

白噪声:



$$P_{ww}(w) = \sigma_w^2$$

$$\frac{P_{ww}(w)_{max}}{P_{ww}(w)_{min}} = 1$$

$$\frac{\lambda_{max}}{\lambda_{min}} = 1$$

FFT