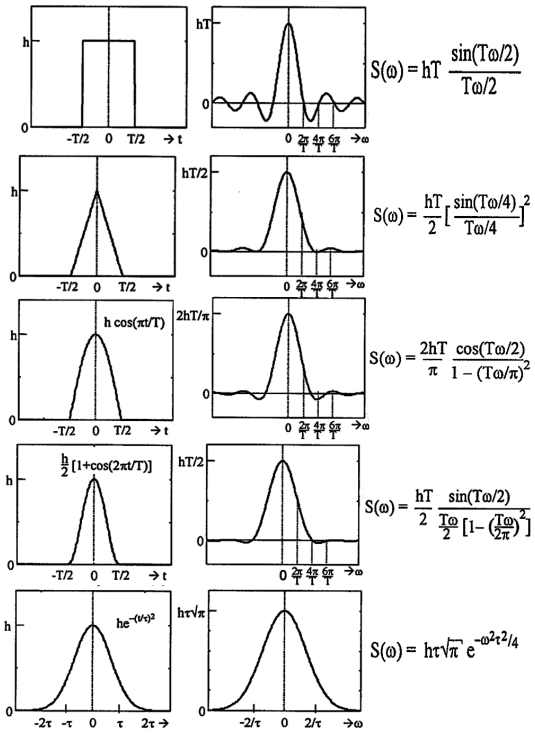
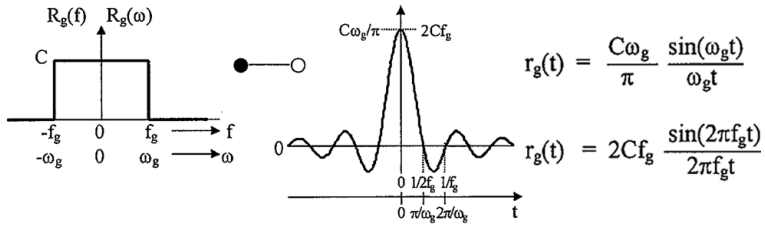


f(t), t ≥ 0	F(s)	f(n), n ≥ 0	F(z)
δ(t)	1	δ(n) = 1,0,0,0,...	1
δ(t−t$_o$), t$_o$>0	e$^{-st_o}$	δ(n−m), m>0	z^{-m}
1 [ε(t)]	$\frac{1}{s}$	1 [ε(n)]	$\frac{1}{1-z^{-1}}$
t	$\frac{1}{s^2}$	nT	$\frac{T\,z^{-1}}{(1-z^{-1})^2}$
e^{-at}	$\frac{1}{s+a}$	e^{-anT}	$\frac{1}{1-e^{-aT}z^{-1}}$
$\frac{e^{-at}-e^{-bt}}{b-a}$	$\frac{1}{(s+a)(s+b)}$	$\frac{e^{-anT}-e^{-bnT}}{b-a}$	$\frac{1}{b-a}\left(\frac{1}{1-e^{-aT}z^{-1}}-\frac{1}{1-e^{-bT}z^{-1}}\right)$
$\frac{1-e^{-at}}{a}$	$\frac{1}{s(s+a)}$	$\frac{1-e^{-anT}}{a}$	$\frac{(1-e^{-aT})z^{-1}}{a(1-z^{-1})(1-e^{-aT}z^{-1})}$
t e^{-at}	$\frac{1}{(s+a)^2}$	nT e^{-anT}	$\frac{T\,e^{-aT}z^{-1}}{(1-e^{-aT}z^{-1})^2}$
cos bt	$\frac{s}{s^2+b^2}$	cos bnT	$\frac{1-(\cos bT)z^{-1}}{1-2(\cos bT)z^{-1}+z^{-2}}$
sin bt	$\frac{b}{s^2+b^2}$	sin bnT	$\frac{(\sin bT)z^{-1}}{1-2(\cos bT)z^{-1}+z^{-2}}$
1 − cos bt	$\frac{b^2}{s(s^2+b^2)}$	1 − cos bnT	$\frac{1}{1-z^{-1}}-\frac{1-(\cos bT)z^{-1}}{1-2(\cos bT)z^{-1}+z^{-2}}$
e^{-at} cos bt	$\frac{s+a}{(s+a)^2+b^2}$	e^{-anT} cos bnT	$\frac{1-e^{-aT}(\cos bT)z^{-1}}{1-2e^{-aT}(\cos bT)z^{-1}+e^{-2aT}z^{-2}}$
e^{-at} sin bt	$\frac{b}{(s+a)^2+b^2}$	e^{-anT} sin bnT	$\frac{e^{-aT}(\sin bT)z^{-1}}{1-2e^{-aT}(\cos bT)z^{-1}+e^{-2aT}z^{-2}}$
e^{-at} [cos bt − (a/b)sin bt]	$\frac{s}{(s+a)^2+b^2}$	e^{-anT} [cos bnT − (a/b)sin bnT]	$\frac{1-e^{-aT}[\cos bT+(a/b)\sin bT]z^{-1}}{1-2e^{-aT}(\cos bT)z^{-1}+e^{-2aT}z^{-2}}$



Matrix inversion

$$A^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{\det(A)} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = \frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

$$A^{-1} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}^{-1} = \frac{1}{\det(A)} \begin{pmatrix} ei-fh & ch-bi & bf-ce \\ fg-di & ai-cg & cd-af \\ dh-eg & bg-ah & ae-bd \end{pmatrix}$$

Trigonometry

$$e^{j\varphi} = \cos(\varphi) + j \cdot \sin(\varphi)$$
$$e^{-j\varphi} = \cos(\varphi) - j \cdot \sin(\varphi)$$
$$\cos(x) = \frac{e^{jx}+e^{-jx}}{2}$$
$$\sin(x) = \frac{e^{jx}-e^{-jx}}{2j}$$
$$\cos^2(x) = \frac{1}{2} + \frac{\cos(2x)}{2}$$
$$\sin^2(x) = \frac{1}{2} - \frac{\cos(2x)}{2}$$
$$\sin(2x) = 2\sin(x) \cdot \cos(x)$$
$$\sin^2(x) + \cos^2(x) = 1$$

Determinants

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

$$\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + bfg + cgh - ceg - fha - ibd$$

Designation	Time domain		Continuous frequency domain		Discrete frequency domain	
	Continuous	Discrete	Fourier	Laplace	Discrete FT	Z Transform
Linearity	$\alpha \cdot f(t) + \beta \cdot g(t)$	$\alpha \cdot f(n) + \beta \cdot g(n)$	$\alpha \cdot F(\omega) + \beta \cdot G(\omega)$	$\alpha \cdot F(s) + \beta \cdot G(s)$	$\alpha \cdot F(n) + \beta \cdot G(n)$	$\alpha \cdot F(z) + \beta \cdot G(z)$
Similarity / Time scaling or reflection about the Y-axis	$f(\alpha t)$	$f(-n)$	$\frac{1}{\lvert\alpha\rvert} F\left(\frac{\omega}{\alpha}\right)$	$\frac{1}{\alpha} F\left(\frac{s}{\alpha}\right)$	$F(-n)$	$F(z^{-1})$
Shift in the time domain	$f(t \pm t_0)$	$f(n \pm n_0)$	$e^{\pm j\omega t_0} F(\omega)$	$F(s) e^{\pm t_0 s}$	$e^{\pm j \frac{n}{N} 2\pi n_0} F(n)$	$z^{\pm n_0} F(z)$
Shift in the frequency domain	$f(t) e^{\mp \alpha t}$	$f(n) e^{\mp j \frac{n}{N} 2\pi n_0}$	$F(\omega \pm \alpha)$	$F(s \pm \alpha)$	$F(n \pm n_0)$	$F(z \pm n_0)$
Convolution in the time domain	$f(t) * g(t)$	$f(n) * g(n)$	$F(\omega) \cdot G(\omega)$	$F(s) \cdot G(s)$	$F(n) \cdot G(n)$	$F(z) \cdot G(z)$
Convolution in the frequency domain	$f(t) \cdot g(t)$	$f(n) \cdot g(n)$	$\frac{1}{2\pi} F(\omega) * G(\omega)$	$\frac{1}{2\pi} F(s) * G(s)$	$\frac{1}{N} F(n) * G(n)$	$\frac{1}{N} F(z) * G(z)$
Derivatives in the time domain / difference formation	$\frac{\partial^n f(t)}{\partial t^n}$	$\Delta^k f(n)$	$(j\omega)^n F(\omega)$	$s^n F(s)$ $-$ $s^{n-1} f(0+)$ $-$ $s^{n-2} \frac{\partial f(0+)}{\partial t}$ $-$ $\ldots - s^0 \frac{\partial^{n-1} f(0+)}{\partial t^{n-1}}$		$(1 - z^{-1})^k F(z)$
Derivative in the frequency domain	$(-t)^k \cdot f(t)$	$nf(n)$	$j^k \frac{-\partial^k F(\omega)}{\partial \omega^k}$	$\frac{\partial^k F(s)}{\partial s^k}$		$-z \frac{\partial F(z)}{\partial z}$
Integration / Summation	$\int_{-\infty}^{t} f(\tau) d\tau$	$\sum_{n=0}^{k} f(n)$	$\frac{F(\omega)}{j\omega}$ $+$ $F(0)\pi\delta(\omega)$	$\frac{F(s)}{s}$		$\frac{1}{1-z^{-1}} F(z)$
Initial value	$\lim_{t \to 0} f(t)$	$f(0)$		$\lim_{s \to \infty} sF(s)$		$\lim_{z \to \infty} F(z)$
Final value	$\lim_{t \to \infty} f(t)$	$\lim_{n \to \infty} f(n)$		$\lim_{s \to 0} sF(s)$		$\lim_{z \to 1} (1 - z^{-1}) F(z))$
Special	Bessel's theorem $\int_{-\infty}^{\infty} f(t) g^*(t) dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) G^*(\omega) d\omega$			Parseval's theorem $W = \int_{-\infty}^{\infty} \lvert f(t)\rvert^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} \lvert F(\omega)\rvert^2 d\omega$		

Quadratic Equation:

$a \cdot x^2 + b \cdot x + c = 0$

$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

Fourier Transform:

$\delta(t) \circ\!\!-\!\!\bullet\ 1$

$1 \circ\!\!-\!\!\bullet\ 2\pi\delta(\omega)$

$\sigma(t) \circ\!\!-\!\!\bullet\ \frac{1}{j\omega} + \pi\delta(\omega)$

$\mathrm{sgn}(t) \circ\!\!-\!\!\bullet\ \frac{2}{j\omega}$

$e^{\pm j\omega_0 t} \circ\!\!-\!\!\bullet\ 2\pi\delta(\omega \mp \omega_0)$

$\sin(\omega_0 t) \circ\!\!-\!\!\bullet\ j\pi(\delta(\omega + \omega_0) - \delta(\omega - \omega_0))$

$\cos(\omega_0 t) \circ\!\!-\!\!\bullet\ \pi(\delta(\omega + \omega_0) + \delta(\omega - \omega_0))$

1 Probability Theory

1.1 Combinatorics

Type of selection or composition of k from n elements	Number of possibilities	
	without repetitions ($k \leq n$)	with repetitions ($k \leq n$)
Permutations	$P_n = n!(n = k)$	$P_n^{(k)} = \frac{n!}{k!}$
Combinations	$C_n^{(k)} = \binom{n}{k}$	$C_n^{(k)} = \binom{n+k-1}{k}$
Variations	$V_n^{(k)} = k!\binom{n}{k}$	$V_n^{(k)} = n^k$

$\binom{n}{k}$ with TR: `nCr(n,k)` En
`Kombinat(n,k)` De

- **Permutations**: Given are n different objects. Then there are $n!$ different orders in which these objects can be arranged.
  e.g.: $x, y, z;\quad x, z, y;\quad z, y, x; \ldots$
- **Combination**: Given are n different objects. Then there are $\binom{n}{k}$ ways to select k objects from them, if the order does not matter.
  e.g.: How many different ways are there to select 6 numbers out of 49 in the lottery?
- **Variation** is called a selection of k elements from n different elements considering the order

1.2 Probability

Range of values: $0 \leq P(A) \leq 1$

Certain event: $P(\Omega) = 1$

Impossible event: $P(\emptyset) = 0$

Calculation rules

Complementary event: $P(\bar{A}) = P(\Omega \setminus A) = 1 - P(A)$

Difference of the events A and B: $P(A \setminus B) = P(A) - P(A \cap B)$

Union of two events: $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

1.3 Laplace Events

In a finite probability space Ω, all elementary events have the same probability.

$$P(A) = \frac{|A|}{|\Omega|}$$

1.4 Independent Events

Independent events A and B are present when:

$P(A \mid B) = P(A)$ and $P(B \mid A) = P(B)$

is satisfied. For them, it applies

$P(A \cap B) = P(A)P(B)$

The fact that A has occurred has no influence on the probability of B.

1.5 Conditional Probability

The probability of the occurrence of event A given that event B has already occurred.

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \underbrace{\frac{P(A) \cdot P(B)}{P(B)} = P(A)}_{\text{only if independent}}$$

1.6 Bayes' Theorem

$$P(B \mid A) = P(A \mid B) \cdot \frac{P(B)}{P(A)}$$

1.7 Total Probability

$$P(A) = \sum_{i=1}^{N} P(A \mid G_i) \cdot P(G_i)$$

1.8 Probability Distribution

1.8.1 Distribution Function

discrete	**continuous**
$P(X \leq x) = F(x) = \sum_{k=-\infty}^{x} p_k$	$P(X \leq x) = F(x) = \int_{-\infty}^{x} \varphi(\tilde{x})d\tilde{x}$
$P(X > x) = 1 - P(X \leq x)$	$P(X > x) = 1 - P(X \leq x)$
$P(\alpha_1 \leq X \leq \alpha_2) = F(\alpha_2) - F(\alpha_1) = \sum_{k=\alpha_1}^{\alpha_2} p_k$	$P(\alpha_1 \leq X \leq \alpha_2) = F(\alpha_2) - F(\alpha_1) = \int_{\alpha_1}^{\alpha_2} \varphi(\tilde{x})d\tilde{x}$
$F_{x_1,x_2}(\alpha_1, \alpha_2) = P(\alpha_1 \leq X_1, \alpha_2 \leq X_2)$	$F_{x_1,x_2}(\alpha_1, \alpha_2) = P(\alpha_1 \leq X_1, \alpha_2 \leq X_2)$

Properties

$\boxed{\mathbb{D}(F) = \mathbb{R}}$ $\boxed{\mathbb{W}(F) \in [0,1]}$ $\boxed{F(-\infty) = 0}$ $\boxed{F(\infty) = 1}$ $\boxed{F(x) \text{ is monotonically increasing}}$

1.8.2 Probability Density

$\varphi(x) = F'(x)$ Density function or probability density

$\varphi_{x_1,x_2}(\alpha_1, \alpha_2) = \frac{\delta^2}{\delta_{x_1}\delta_{x_2}} F_{x_1,x_2}(\alpha_1, \alpha_2)$ Density function or probability density with multiple variables

At jump points of F(x):
$\varphi(x)$ = Dirac with the weight of the jump height

1.8.3 Calculation Rules for φ and F

Given: X, Y random variables
φ_x, φ_y known

Distribution Function:	Density:
$F_{x+a}(x) = F_x(x-a)$	$\varphi_{x+a}(x) = \varphi_x(x-a)$
$F_{\lambda x}(x) = F_x(\frac{x}{\lambda})$	$\varphi_{\lambda x}(x) = \varphi_x(\frac{x}{\lambda})\frac{1}{\lambda}$
$F_{x+y}(x) = F_x * \varphi_y(y) = F_y * \varphi_x(x)$	$\varphi_{x+y}(x) = \varphi_x * \varphi_y(x)$
$F_{\sqrt{x}}(x) = F_x(x^2)$	$\varphi_{\sqrt{x}}(x) = 2x\varphi_x(x^2)$
$F_{x^2}(x) = F_x(\sqrt{x})$	$\varphi_{x^2}(x) = \frac{1}{2}x^{-\frac{1}{2}}\varphi_x(\sqrt{x})$

Algorithm **Example**

1. Apply definition of F: $F_{\lambda x}(x) = P(\underbrace{\lambda X \leq x}_{*})$
2. Reformulate condition *: $P(X \leq \frac{x}{\lambda}) = F_x(\frac{x}{\lambda})$
3. for density: $\frac{d}{dx}$

$\varphi_{\lambda x}(x) = \frac{d}{dx}F_{\lambda x}(x) = \frac{d}{dx}F_x(\frac{x}{\lambda}) = \varphi_x(\frac{x}{\lambda})\frac{1}{\lambda}$

1.8.4 Expected Value

Let X be a function on Ω, and suppose Ω can be divided into finitely many events, on which $X(\omega)$ is constant, A_i, then the expected value of X

$Expectedvalue = \sum Value \cdot Probability$

$$E(X) = \sum_{i=0}^{n} \underbrace{a_i}_{\text{Value}} \cdot \underbrace{P(X = a_i)}_{\text{Prob.}} = \int_{-\infty}^{\infty} \alpha \cdot \varphi_x(x)d\alpha$$

$$E(y) = E(g(x)) = \int_{-\infty}^{\infty} g(\alpha) \cdot \varphi_x(x)d\alpha \quad \text{for } y = g(x)$$

Calculation Rules

$E(X+Y) = E(X) + E(Y)$
$E(\lambda X + \mu) = \lambda \cdot E(X) + \mu \quad \lambda, \mu \in \mathbb{R}$
$E(XY) = E(X) \cdot E(Y)$ if X,Y are independent

1.8.5 Variance

$var(x) = \sigma^2 = E[(X - E(X))^2] = E(X^2) - E(X)^2$
$= \int_{\infty}^{\infty} (\alpha - E(x))^2 \varphi_x(\alpha)d\alpha$

Calculation Rules

$var(\lambda X) = \lambda^2 var(X) \qquad \lambda, \mu \in \mathbb{R}$
$var(X_1 + X_2 + \ldots + X_n) \neq var(nX)$

$$var(X+Y) = \begin{cases} var(X) + var(Y) & \text{(X,Y independent)} \\ var(X) + var(Y) + 2 \cdot cov(X,Y) & \text{(X,Y dependent)} \end{cases}$$

$var(XY) = var(Y)var(X) + var(Y)E(X)^2 + var(X)E(Y)^2$

1.8.6 Correlation

$r_{xy} = E(XY^*)$

1.8.7 Covariance

$c_{xy} = cov(X,Y) = E((x - E(X))(y - E(Y)))$
$= E(XY^*) - E(X)E(Y) = \underbrace{0}_{\text{if X,Y are independent}} = \underbrace{r_{xy}}_{\text{if X,Y are mean-free}}$

1.8.8 Correlation Coefficient

$\rho_{xy} = \frac{E((X-m_x)\cdot(Y-m_y)^*)}{\rho_x \rho_y}$
$|\rho_{xy} \leq 1|$
Independent
Two random variables are statistically independent if the probability density function is separable:
$\varphi_{xy}(\alpha,\beta) = \varphi_x(\alpha)\varphi_y(\beta) \Longrightarrow$ also $E(XY) = E(X)E(Y)$
If the variables are additionally uncorrelated, it holds that: $Var(X+Y) = Var(X) + Var(Y)$

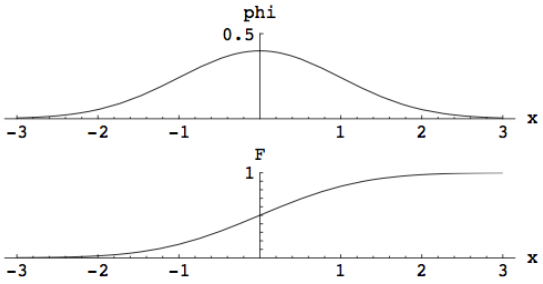
1.8.9 Moment Generating Function

The moment generating function is a representation of all moments of a distribution. $\Phi(t) = E(e^{tX}) = \int_{-\infty}^{\infty} e^{tx} \cdot \varphi(x)dx$ or
$= \sum_{i=1}^{\infty} e^{tx_i} \cdot P(X = x_i)$
$\Phi(t)$ is called the moment generating function because:
$\frac{d^n}{dt^n}\Phi(t) = \int_{-\infty}^{\infty} x^n \cdot e^{tx} \cdot \varphi(x)dx$ or $= \sum_{i=1}^{\infty} x_i^n \cdot e^{tx_i} \cdot P(X = x_i)$
for $\frac{d^n}{dt^n}\Phi(0) = \int_{-\infty}^{\infty} x^n \cdot \varphi(x)dx$ or $= \sum_{i=1}^{\infty} x_i^n \cdot P(X = x_i) = E(X^n) = \mu_n$

If $\Phi_x(t) = \Phi_y(t)$ then $F_x(x) = F_y(x)$. This means a probability distribution can be uniquely associated with a moment generating function.

1.8.10 Various Probability Density Functions

Gaussian Distribution/ Normal Distribution
Many small, independent random variables accumulate to form a normally distributed random variable.
$\varphi(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} = N(\mu;\sigma^2)$
$F(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \int_{-\infty}^{x} e^{-\frac{(\bar{x}-\mu)^2}{2\sigma^2}}$



Density function (above) and distribution function (below) of the normal distribution.

Standardization
Expected Value: $E(X) = \mu$ (=0 in Standard normal dist.)
Variance : $var(X) = \sigma^2$ (=1 in Standard normal dist.)

$x = \frac{X - \mu}{\sigma}$ x from table

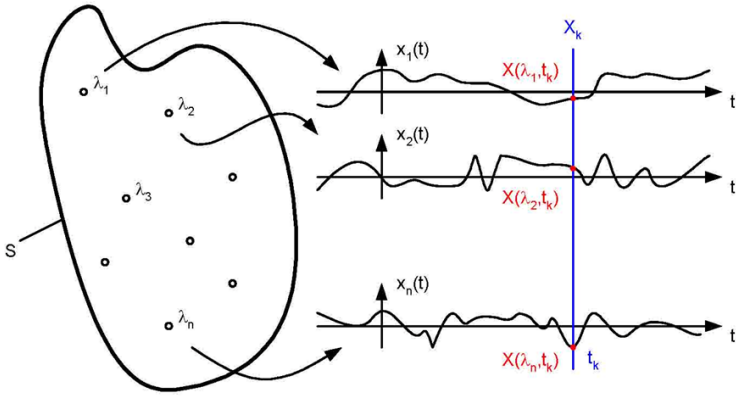
68% of the values lie within the interval $[\mu - \sigma, \mu + \sigma]$, 95% in $[\mu - 2\sigma, \mu + 2\sigma]$, 99.7% in $[\mu - 3\sigma, \mu + 3\sigma]$

Normal Distribution with Multiple Variables
$\varphi_{x,y}(\alpha,\beta) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho_{xy}^2}} e^{-\frac{1}{2(1-\rho_{xy})^2}+\frac{(\alpha-m_x)^2}{\sigma_x^2}-2\rho_{xy}\frac{(\alpha-m_x)\beta-m_y)^2}{\sigma_x\sigma_y}+\frac{(\beta-m_y)^2}{\sigma_y^2}}$
$\varphi_x(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{C}_x|^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_x)^T\mathbf{C}_x^{-1}(\mathbf{x}-\mathbf{m}_x)}$ with
$\mathbf{m}_x = [E(x_1), E(x_2), \ldots, E(x_n)]^T$ and covariance matrix $\mathbf{C}_x$; $c_{ij} = E((x_i - E(x_i)(x_j - E(x_j))))$
Calculation Rule for the Normal Distribution with Multiple Variables

- if $z = ax + by$ then
  $m_z = am_x + bm_y$
  $\sigma_z^2 = a^2\sigma_x^2 + b^2\sigma_y^2 + 2ab\sigma_x\sigma_y$

- if x and y are uncorrelated, then
  $f_{x,y}(\alpha,\beta) = \varphi_x(\alpha)\varphi_y(\beta)$ meaning x and y are also statistically independent.
- The optimal nonlinear estimator for the values μ and σ is the same as the linear estimator.

1.9 Random Processes



In a **random process**, each **outcome $\boldsymbol{\lambda}$** from the **outcome space $\boldsymbol{S}$** is assigned a **deterministic function $\boldsymbol{x(\lambda, t)}$** .
Random processes describe a deterministic time function triggered by the outcome of a random experiment. e.g., $sin(\omega t + X)$ where X is a random variable assigned from the event space at the beginning.
Temporally **randomly occurring functions** can also be seen as **deterministic functions interpreted** in such a way that the observer never knows which specific function $x_\lambda(t)$ is present.

For comparison: In the case of random variables, each elementary event is assigned a number.

1.9.1 Statistical Averages (Ensemble Averages)

The statistical averages are a function of time t, as they are averages over the ensemble (entire outcome space). Here, all deterministic functions are averaged at a specific time t.

Expected Value:	$\mu_x(t) = E[X(t)] = \int_{-\infty}^{+\infty} x \cdot f_x(x;t)\,dx$ or $= \frac{1}{N}\sum_{i=1}^{N} x_i(n)$
Expected Value of a Deterministic Signal:	$\mu_f(t) = E(f(t)) = f(t)$
Variance:	$\sigma_x^2(t) = E((x(t) - \mu_x(t))^2) = c_x(t_1, t_1)$

1.9.2 Stationarity

A stationary process does not change its statistical properties over time.

Strict Sense Stationarity (SSS)
In a strictly stationary process, the n-dimensional probability densities remain constant over time. That is, the **statistical properties** and thus the probability densities are the **same at all times**.

$$f_x(x_1, x_2, \ldots, x_n; t_1, t_2, \ldots, t_n) = f_x(x_1, x_2, \ldots, x_n; t_1 + c, t_2 + c, \ldots, t_n + c) \qquad \forall (c, n \in \mathbb{R})$$

Wide Sense Stationarity (WSS) - Second-order Stationarity
In a weakly stationary process, the statistical properties are **not the same at every moment in time**, however, they are **not dependent on an absolute** point in time, but rather on the **difference** (τ) **between two points in time** (t_1, t_2).

$$f_x(x_1, x_2; t_1, t_2) = f_x(x_1, x_2; t_1 + c, t_2 + c) \qquad \forall (c \in \mathbb{R})$$

Mean:	$E[X(t)] = \mu_x = \text{const.}$	remains constant over time
Quadratic Mean:	$E[X^2(t)] = r_x(0)$	
Autocorrelation:	$r_{xx}(t_1, t_2) = r_x(\tau)$	only **dependent** on the **time difference** ($\tau = t_2 - t_1$) and **not directly** on the **time** t
Autocovariance:	$c_{xx}(t_1, t_2) = r_x(\tau) - \mu_x^2 = c_{xx}(\tau)$	

A random process is always a WSS process as long as the expectation is constant and the autocorrelation function is only a function of τ, i.e., both statistical parameters are independent with respect to a temporal shift. Every strictly stationary process is also weakly stationary, but not vice versa.

1.9.3 Time Averages (Time Means)

Here, the respective deterministic functions (sample functions) are averaged over time. If the time average is calculated over the entire random process, then the time averages are **random variables**, i.e., the following two expressions are dependent on which function is used (hence the index $_i$).

Mean: $\overline{x_i} = \langle x_i(t) \rangle = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} x_i(t)\, dt$

Autocorrelation: $\overline{r}_{x_i x_i}(\tau) = \langle x_i(t) \cdot x_i(t+\tau) \rangle = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} x_i(t) \cdot x_i(t+\tau)\, dt$

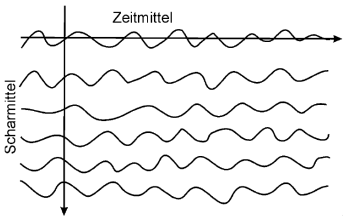
If the **process is stationary**, the following applies:

Mean: $E[\overline{x}] = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} E[x(t)]\, dt = \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} \mu_x\, dt = \mu_x$

Autocorrelation: $E[\overline{r}_{xx}(\tau)] = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} E[x(t)x(t+\tau)]\, dt = \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} r_{xx}(\tau)\, dt = r_{xx}(\tau)$

1.9.4 Ergodicity

A stationary process is also ergodic if the **time averages** correspond to the **statistical averages**.



Only in ergodic processes it necessarily holds:

$E[X(t)] = \overline{x} = \langle x(t) \rangle$	DC-Level
$E[X(t)]^2 = (\overline{x})^2 = \langle x(t) \rangle^2$	DC-Power
$E[X^2(t)] = r_{xx}(0) = \overline{x^2} = \langle x^2(t) \rangle$	Total Power
$\sigma_x^2(t) = \langle x^2(t) \rangle - \langle x(t) \rangle^2$	AC-Power
$\sigma_x(t) = \overline{\sigma}_x$	RMS-Level (Effective value) of the AC signal

1.9.5 Comparison Stationary - Ergodic

Swarm of mosquitoes:
Ergodic: If all mosquitoes fly together in a swarm, then averaged over time (Time average), each mosquito flies as fast as the entire swarm on average (Ensemble average), otherwise, the swarm could not stay together.
Stationary: If a mosquito is sick and cannot keep up with the swarm, it flies alone and, above all, slower. Thus, its average speed (Time average) is not equal to that of the swarm (Ensemble average), so it arrives later at the destination.
Neither: If the mosquitoes fly slower after the start, the average speed of the swarm (Ensemble average) is not constant.
School grades:
Ergodic: All students would have to have the same report card grade (Time average) and moreover, this grade would also have to correspond to the class average (Ensemble average) of the individual exams.
Stationary: The class average (Ensemble average) is the same for every exam, but there are students of varying strengths with different report card grades (Time average).
Neither: The class average (Ensemble average) is always different.
Thermal noise:
This is **ergodic** at a constant temperature.

1.9.6 Correlations and Power Spectra

Formulas in this section apply to **stationary** processes.

Autocorrelation: $r_x(t_1,t_2) = r_x(t_1-t_2) = r_x(\tau) = r_{xx}(\tau) = E[X(t)X(t+\tau)]$

$|r_{xx}(\tau)| \leq r_{xx}(0) = E[X^2(t)]$ $\quad r_{xx}(-\tau) = r_{xx}(\tau)$ (even, Fourier transform becomes real)

Cross-correlation: $r_{xy}(\tau) = r_{xy}(\tau) = E[X(t)Y(t+\tau)]$ $\quad r_{xy}(-\tau) = r_{yx}(\tau)$ (Order of indices!)

$|r_{xy}(\tau)| \leq \frac{1}{2}\left[r_{xx}(0) + r_{yy}(0)\right]$ $\quad |r_{xy}(\tau)| \leq \sqrt{r_{xx}(0) r_{yy}(0)}$

Autocovariance : $c_{xx}(t_1,t_2) = E\left[(X(t_1)-\mu_x(t_1))\cdot(X(t_2)-\mu_x(t_2))\right] = r_{xx}(t_1,t_2) - \mu_x(t_1)\cdot\mu_x(t_2) =$

$c_x(\tau) = c_{xx}(\tau) = E[(X(t)-E[X(t)])\cdot(X(t+\tau)-E[X(t+\tau)])] = r_{xx}(\tau) - \mu_x^2$

Cross-covariance: $c_{xy}(t_1,t_2) = E\left[(X(t_1)-\mu_x(t_1))\cdot(Y(t_2)-\mu_y(t_2))^*\right] = r_{xy}(t_1,t_2) - \mu_x(t_1)\cdot\mu_y(t_2) =$

$c_{xy}(\tau) = E[(X(t)-E[X(t)])\cdot(Y(t+\tau)-E[Y(t+\tau)])] = r_{xy}(\tau) - \mu_x\mu_y$

Random processes are considered **uncorrelated** if $c_{xy}(\tau) = 0$

Spectral Power

The autocorrelation function $r_{xx}(\tau)$ and power spectral density $P_{xx}(\omega)$ form a Fourier-**transform pair**.
The power spectral density can be interpreted as the **average power per frequency band** and is defined as follows:

$$P_{xx}(\omega) = E\left[\lim_{T\to\infty}\frac{1}{T}\cdot|X(\omega)|^2\right] = \int_{-\infty}^{+\infty} r_{xx}(\tau)\cdot e^{-j\omega\tau}\,d\tau \qquad \bullet\!\!-\!\!\circ \qquad r_{xx}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} P_{xx}(\omega)\cdot e^{j\omega\tau}\,d\omega$$

or:

$$P_{xx}(z) = \sum_{k=-\infty}^{\infty} r_{xx}(k)z^{-k} \qquad \bullet\!\!-\!\!\circ \qquad r_{xx}(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} P_{xx}(e^{jw})e^{jkw}dw \qquad r_{xx}(0) = \frac{1}{2\pi}\int_{-\pi}^{\pi} P_{xx}(e^{jw})dw = \lim_{z\to 0} P_{xx}(z)$$

$P_{xx}(\omega)$ is purely real, symmetric ($P_{xx}(z) = P_{xx}^*(\frac{1}{z^*})$) and ≥ 0.
Cross-correlations ($r_{yx}(\tau), r_{xy}(\tau)$) and cross-spectral densities ($P_{yx}(\tau), P_{xy}(\tau)$) form a Fourier-transform pair.

$$r_{yx}(\tau) \circ\!\!-\!\!\bullet\ P_{yx}(\omega) \qquad\qquad r_{xy}(\tau) \circ\!\!-\!\!\bullet\ P_{xy}(\omega)$$

Since autocorrelations are definitely non-causal (defined also for negative time values), the bilateral z-transform must be applied.
The eigenvalues of the autocorrelation matrix have the following limitation:

$$\min_{\omega} P_{xx}(z) \leq \lambda_i \leq \max_{\omega} P_{xx}(z)$$

1.9.7 Autocorrelation Examples

White Noise	$r_{xx}(n) = \frac{N_0}{2}\delta(n) \circ\!\!-\!\!\bullet\ P_{xx}(z) = \frac{N_0}{2}$;	all z
First-Order Process	$r_{xx}(n) = P_s\cdot a^{\lvert n\rvert} \circ\!\!-\!\!\bullet\ P_{xx}(z) = P_s\cdot\frac{1-a^2}{(1-az^{-1})(1-az)}$	for $a < \lvert z\rvert < \frac{1}{a}$
	$r_{xx}(n) = a^n(u(n)-u(n-N)) \circ\!\!-\!\!\bullet\ P_{xx}(z) = \frac{1-a^Nz^{-N}}{1-az^{-1}}$	for $\lvert z\rvert > 0$
	$r_{xx}(n) = a^n\cdot u(n) \circ\!\!-\!\!\bullet\ P_{xx}(z) = \frac{1}{1-az^{-1}}$	for $\lvert z\rvert > a$
	$r_{xx}(n) = -a^n\cdot u(-n-1) \circ\!\!-\!\!\bullet\ P_{xx}(z) = \frac{1}{1-az^{-1}}$	for $\lvert z\rvert < a$
Binary Data Signal	$r_{xx}(0) = P_s = A_1^2p_1 + A_0^2p_0$ $r_{ss}(\lvert n\rvert < T)$ = linear transition from $r_{ss}(0)$ to $r_{ss}(\lvert n\rvert = T)$ $r_{xx}(\lvert n\rvert \geq T) = P_s = A_1^2p_1^2 + A_0A_1p_0p_1 + A_1A_0p_1p_0 + A_0^2p_0^2$	
Superimposed Signals $g_\pm(t) = s(t) \pm f(t)$	$r_{gg\pm}(n) = r_{ss}(n) + r_{sf}(n) + r_{sf}(-n) + r_{ff}(n)$ $\circ\!\!-\!\!\bullet\ P_{g+}(\omega) = P_s(\omega) + P_f(\omega) \pm 2\mathrm{Re}\{P_{sf}(\omega)\}$	

1.9.8 Numerical Calculation

Numerical Calculation	
Autocorrelation	$\hat{r}_{xx}(n) = \frac{1}{N}\sum_{m=0}^{N-1} s(m)s[(m+n)] = \frac{1}{N}\sum_{m=0}^{N-1} s[(m-n)]s(m) \circ\!\!-\!\!\bullet\ \hat{P}_{xx}(z)$
Cross-correlation	$\hat{r}_{xy}(n) = \frac{1}{N}\sum_{m=0}^{N-1} s(m)f[(m+n)] = \frac{1}{N}\sum_{m=0}^{N-1} s[(m-n)]f(m) \circ\!\!-\!\!\bullet\ \hat{P}_{xy}(z)$
Power Spectral Density	$\hat{P}_{xx}(z) = \frac{1}{N}\lvert S_T(z)\rvert^2 \bullet\!\!-\!\!\circ\ \hat{r}_{xx}(n)$ with $S_T(z) \bullet\!\!-\!\!\circ\ s(n)$
Cross Power Spectral Density	$\hat{P}_{xy}(z) = \frac{1}{N}S_T^*(z)F_T(z) \bullet\!\!-\!\!\circ\ \hat{r}_{xy}(n)$ with $F_T(z) \bullet\!\!-\!\!\circ\ f(n)$

1.9.9 Transmission of Random Processes through LTI Systems

A random process is transmitted through an LTI system. $Y(t) = L[X(t)] \Rightarrow Y(t) = h(t) * X(t)$

	General	WSS Process
Mean	$\mu_y(t) = h(t) * \mu_x(t)$	$\mu_y = H(0)\mu_x$
Autocorrelation*	$r_{yy}(t_1,t_2) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} h(\alpha)h(\beta)r_{xx}(t_1-\alpha,t_2-\beta)\,d\alpha\,d\beta$	$r_{yy}(\tau) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} h(\alpha)h(\beta)r_{xx}(\tau+\alpha-\beta)\,d\alpha\,d\beta$
Spectral Power		$P_{yy}(\omega) = H^*(\omega)H(\omega)P_{xx}(\omega) = \lvert H(\omega)\rvert^2 P_{xx}(\omega)$

* = It is much simpler to calculate the autocorrelation from the spectral power (transform pair) - instead of from this dreadful integral.
A WSS process at the input also produces a WSS process at the output.

1.9.10 Filtering of Random Processes

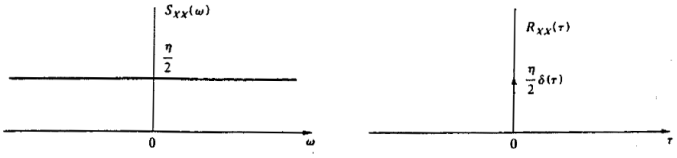
The output autocorrelation $r_y(k)$ is

$r_y(k) = r_x(k) * h(k) * h^*(-k)$ or $\boxed{P_y(z) = P_x(z)\cdot H(z)\cdot H^*(\frac{1}{z^*}) \quad P_y(e^{j\omega}) = P_x(e^{j\omega})\cdot\lvert H(e^{j\omega})\rvert^2}$

1.9.11 White Noise

$P_{xx}(\omega) = \frac{\eta}{2} \qquad r_{xx}(\tau) = \frac{\eta}{2}\cdot\delta(\tau) = \sigma_x^2\cdot\delta(\tau)$

Example: thermal noise of resistors
Decreases in practice in the **Tera-Hz range**!



1.9.12 Colored Noise Signals

Term (German)	Term (English)	Power Spectrum	Note
Rosa Rauschen	Pink Noise	$P_{xx}(\omega) = c\cdot\frac{1}{\omega}$	Test signal for audio engineering, due to constant power per octave
Braunes/Rotes Rauschen	Brown/Red Noise	$P_{xx}(\omega) = c\cdot\frac{1}{\omega^2}$	
Blaues Rauschen	Blue Noise	$P_{xx}(\omega) = c\cdot\omega$	
Violettes Rauschen	Purple/Violet Noise	$P_{xx}(\omega) = c\cdot\omega^2$	

1.10 Estimation Hayes p 72

1.10.1 Bias

The bias is the difference between the real value Θ and the estimate $\hat{\Theta}$: $B = \Theta - E(\hat{\Theta}_N)$. When $B = 0$, the estimator is *unbiased*. When the estimator is biased but the bias goes to zero as the number of observations N goes to infinity, then the estimator is *asymptotically unbiased* ($\lim_{N\to\infty} E(\hat{\Theta}_N) = \Theta$). If the bias stays $B \neq 0$, the estimator is *biased*.

1.10.2 Consistency

If $\lim_{N\to\infty} Var(\hat{\Theta}_N) = 0$, the estimator is consistent. $\hat{\Theta}_N$ is said to *converge to* Θ *with probability one*. Here, a *consistent* estimator if it is asymptotically unbiased and has a variance that goes to zero as N goes to infinity.

2 Signal Modeling Hayes p 129

Objective: Calculate the coefficients for a filter whose impulse response matches as closely as possible with a given signal ($x[n]$ - with length N).

$$y(n) = \sum_{k=0}^{q} b(k)x(n-k) - \sum_{k=1}^{p} a(k)y(n-k)$$

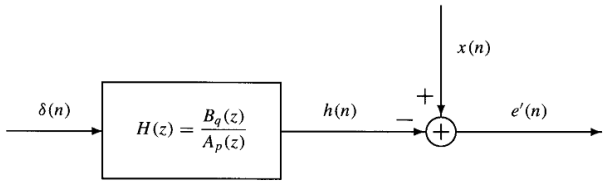
$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_q z^{-q}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_p z^{-p}}$$

Method	exact match	minimal error
LS	$[0]$	$[1, N-1]$
Padé	$[0, q+p+1]$	-
Prony	$[0, q]$	$[q+1, N-1]$
Shanks	$[0]$	$[1, N-1]$

q: number of zeros
p: number of poles
N: length of signal $x(n)$

2.1 Least Squares Method Hayes p 131

The coefficients **A and B** are estimated using the least squares error method. Since solving the estimation requires a $p+q$ equation system, this method is very rarely applied.



$$\frac{\partial \sum_{n=0}^{\infty} |e'(n)|^2}{\partial a_p^*(k)} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left[X(e^{j\omega}) - \frac{B_q(e^{j\omega})}{A_p(e^{j\omega})}\right] \frac{B_q^*(e^{j\omega})}{A_p^*(e^{j\omega})^2} e^{jk\omega} d\omega = 0$$

$$\frac{\partial \sum_{n=0}^{\infty} |e'(n)|^2}{\partial b_q^*(k)} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left[X(e^{j\omega}) - \frac{B_q(e^{j\omega})}{A_p(e^{j\omega})}\right] \frac{e^{jk\omega}}{A_p^*(e^{j\omega})} d\omega = 0$$

2.1.1 Example FIR Equalizer

With a FIR first order (2 coefficients) a given sequence (e.g. $[0,2,1,0]$) should be filtered so that output is a Dirac $[0,1,0,0]$.
$\underbrace{\begin{bmatrix} x[0] & 0 \\ x[1] & x[0] \\ 0 & x[1] \end{bmatrix}}_{\boldsymbol{A}} \cdot \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$. To solve the equations the pseudo inverse has to be used: $\boldsymbol{A}^\dagger = \left(\boldsymbol{A}^T \cdot \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \Rightarrow \boldsymbol{A}^\dagger \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$

2.2 Padé Approximation

The resulting impulse response **matches exactly** with the given signal $x[n]$ in the interval $[0, q+p+1]$, afterwards, the filter is not necessarily stable.

1. Calculate a-coefficients / poles: $\boldsymbol{X}_q \cdot a_p = -x_{q+1} \Longrightarrow a_p = -\boldsymbol{X}_q^{-1} \cdot x_{q+1}$ (Matlab: $a_t = -X_t \backslash x$)

$$\underbrace{\begin{bmatrix} x[q] & x[q-1] & \dots & x[q-p+1] \\ x[q+1] & x[q] & \dots & x[q-p+2] \\ \vdots & \vdots & \ddots & \vdots \\ x[q+p-1] & x[q+p-2] & \dots & x[q] \end{bmatrix}}_{\boldsymbol{X}_q} \cdot \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix}}_{a_p} = \underbrace{\begin{bmatrix} -x[q+1] \\ -x[q+2] \\ \vdots \\ -x[q+p] \end{bmatrix}}_{x_{q+1}} \Longleftrightarrow \begin{bmatrix} x[q+1] & x[q] & \dots & x[q+1-p] \\ x[q+2] & x[q+1] & \dots & x[q+2-p] \\ \vdots & \vdots & \ddots & \vdots \\ x[q+p] & x[q+p-1] & \dots & x[q] \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

If A is singular, the assumption that $a_0 = 1$ was incorrect. $\Longrightarrow$ Set $a_0 = 0$. $\Longrightarrow \boldsymbol{X}_q \cdot \boldsymbol{a}_p = \boldsymbol{0}$

2. Calculate b-coefficients / zeros: $\boldsymbol{X}_0 \cdot a_p = b_q$

$$\begin{matrix} & & & {\scriptstyle (0\dots p)} & \\ n=0 \\ n=1 \\ \vdots \\ n=q \end{matrix} \underbrace{\begin{bmatrix} x[0] & 0 & \dots & 0 \\ x[1] & x[0] & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ x[q] & x[q-1] & \dots & x[q-p] \end{bmatrix}}_{\boldsymbol{X}_0} \cdot \underbrace{\begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix}}_{a_p} = \underbrace{\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_q \end{bmatrix}}_{b_q}$$

2.3 Prony's Method Hayes p 144

The resulting impulse response **matches exactly** with the given signal $x[n]$ in the interval $[0, q]$. In the interval $[q+1, N-1]$, the values are approximated with the **least squares error** (overdetermined system of equations).

1. Calculate autocorrelation: $r_x(k,l) = \sum_{n=q+1}^{\infty} x(n-l)x^*(n-k); \qquad k,l \geq 0$

2. Calculate a-coefficients: $\boldsymbol{R}_x \cdot a_p = -r_x \Longrightarrow a_p = -\boldsymbol{R}_x^{-1} \cdot r_x$ (Matlab: $a_p = -R_x \backslash r_x$)
   The minimal error is: $\epsilon_{p,q} = r_x(0,0) + \sum_{k=1}^{p} a_p(k) r_x(0,k)$

$$\underbrace{\begin{bmatrix} r_x[1,1] & r_x[1,2] & \dots & r_x[1,p] \\ r_x[2,1] & r_x[2,2] & \dots & r_x[2,p] \\ \vdots & \vdots & \ddots & \vdots \\ r_x[p,1] & r_x[p,2] & \dots & r_x[p,p] \end{bmatrix}}_{\boldsymbol{R}_x} \cdot \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix}}_{a_p} = -\underbrace{\begin{bmatrix} r_x[1,0] \\ \vdots \\ r_x[p,0] \end{bmatrix}}_{r_x} \Longrightarrow \begin{bmatrix} r_x[0,0] & r_x[0,1] & \dots & r_x[0,p] \\ \hline r_x[1,0] & r_x[1,1] & \dots & r_x[1,p] \\ r_x[2,0] & r_x[2,1] & \dots & r_x[2,p] \\ \vdots & \vdots & \ddots & \vdots \\ r_x[p,0] & r_x[p,1] & \dots & r_x[p,p] \end{bmatrix} \cdot \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \epsilon_{p,q} \\ \hline 0 \\ \vdots \\ 0 \end{bmatrix}$$

   For all-pole models, the a-coefficients can be directly calculated from the b system of equations. For this, all b-coefficients except for b_0 must be set to 0.

3. Calculate b-coefficients as in **Padé**: $\boldsymbol{X}_0 \cdot a_p = b_q$

2.4 Shanks' Method Hayes p 154

The B-coefficients are estimated given the A's in such a way that the resulting impulse response in the interval $[0, N-1]$ with the **least squares error** (overdetermined system of equations) approximates the given signal $x[n]$. The only improvement would be if the A-coefficients could also include all signal values (Least-square).

1. Calculate a-coefficients as in **Prony**: $a_p = -\boldsymbol{R}_x^{-1} \cdot r_x$ (Matlab: $a_p = -R_x \backslash r_x$)
2. Calculate impulse response $g[n]$ of the purely recursive filter ($g[n] = \delta(n) - \sum_{k=1}^{p} a_p(k) g(n-k)$), or using inverse z-transform of $H(z)$.
3. Calculate autocorrelation: $r_g(k) = \sum_{n=0}^{\infty} g(n) g^*(n-k)$; or inverse z-transform of $|H(z)|^2$
4. Calculate cross-correlation: $r_{xg}(k) = \sum_{n=0}^{\infty} x(n) g^*(n-k)$
5. Calculate b-coefficients (for stationary processes): $b_q = R_g^{-1} \cdot r_{xg}$ (Matlab: $b_q = R_g \backslash r_{xg}$)
   for

$$\begin{matrix} \\ n=0 \\ n=1 \\ \vdots \\ n = N_{ha} + q \end{matrix} \overbrace{\begin{bmatrix} r_g(0) & r_g(1) & \dots & r_g(q) \\ r_g(1) & r_g(0) & \dots & r_g(q-1) \\ \vdots & \vdots & \ddots & \vdots \\ r_g(q) & r_g(q-1) & \dots & r_g(0) \end{bmatrix}}^{(0 \dots q)} \underbrace{\cdot \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_q \end{bmatrix}}_{b_q} = \underbrace{\begin{bmatrix} r_{xg}(0) \\ r_{xg}(1) \\ \vdots \\ r_{xg}(q) \end{bmatrix}}_{r_{xg}}$$

2.5 All-Pole Modeling Hayes p 160

Corresponds to: 2.9.2

1. Calculate autocorrelation: $r_x(k) = \sum_{n=0}^{\infty} x(n) x^*(n-k); \qquad k \geq 0$
2. Calculate a-coefficients: $\boldsymbol{R}_x \cdot a_p = -r_x \Longrightarrow a_p = -\boldsymbol{R}_x^{-1} \cdot r_x$ (Matlab: $a_p = -R_x \backslash r_x$)
   The minimal error is: $\epsilon_p = r_x(0) + \sum_{k=1}^{p} a_p(k) r_x^*(k)$

$$\underbrace{\begin{bmatrix} r_x(0) & r_x^*(1) & \dots & r_x^*(p-1) \\ r_x(1) & r_x(0) & \dots & r_x^*(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(p-1) & r_x(p-2) & \dots & r_x(0) \end{bmatrix}}_{\boldsymbol{R}_x} \cdot \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix}}_{a_p} = -\underbrace{\begin{bmatrix} r_x(1) \\ \vdots \\ r_x(p) \end{bmatrix}}_{r_x}$$

3. $b(0) = \sqrt{\epsilon_p}$

2.6 Linear Prediction Hayes p 165 (Exercise 4.13)

$$\hat{x}(n+n_0) = -\sum_{k=1}^{p} a_p(k)x(n-k)$$

$$\underbrace{\begin{bmatrix} r_x[1,1] & r_x[1,2] & \dots & r_x[1,p] \\ r_x[2,1] & r_x[2,2] & \dots & r_x[2,p] \\ \vdots & \vdots & \ddots & \vdots \\ r_x[p,1] & r_x[p,2] & \dots & r_x[p,p] \end{bmatrix}}_{\boldsymbol{R}_x} \cdot \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix}}_{a_p} = \underbrace{\begin{bmatrix} r_x[1,-n_0] \\ \vdots \\ r_x[p,-n_0] \end{bmatrix}}_{r_x}$$

2.7 FIR Inverse Filter Hayes p 166

$$G(z)H(z) \approx 1 \qquad g(n) * h(n) \approx \delta(n) \qquad r_g(k) = \sum_{n=0}^{\infty} g(n)g^*(n-k) \qquad \text{Delay: } n_0$$

$$\underbrace{\begin{bmatrix} r_g(0) & r_g(1) & \dots & r_g(N-1) \\ r_g(1) & r_g(0) & \dots & r_g(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_g(N-1) & r_g(N-2) & \dots & r_g(0) \end{bmatrix}}_{\boldsymbol{R}_g} \cdot \underbrace{\begin{bmatrix} h_N(0) \\ \vdots \\ h_N(N-1) \end{bmatrix}}_{h_N} = -\underbrace{\begin{bmatrix} g(n_0) \\ g(n_0-1) \\ \vdots \\ g(0) \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{r_{gd}}$$

2.8 Approximation with limited data using All-Pole Filter

2.8.1 Autocorrelation Method Hayes p 178

Given the autocorrelations, see Chapter 2.9.2.
Assumption: The data $x[n]$ is zero outside the definition range ($n = [0..N]$): $x[n] = 0$. This forces the filter into a stable state.

1. Calculate a-coefficients (using Pseudoinverse / generalized Inverse): $a_p = -\left(\boldsymbol{X}_p^T \cdot \boldsymbol{X}_p\right)^{-1} \cdot \boldsymbol{X}_p^T \cdot x_1$ (Matlab: $a_p = -X_p \backslash x_1$)

$$\begin{array}{c} \\ n=q+1 \\ n=q+2 \\ \vdots \\ n=N \\ n=N+1 \\ \vdots \\ N+p \end{array} \underbrace{\left[\begin{array}{cccc} x[0] & 0 & \dots & 0 \\ x[1] & x[0] & \dots & 0 \\ x[2] & x[1] & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \hline x[p-1] & x[p-2] & \dots & x[0] \\ x[p] & x[p-1] & \dots & x[1] \\ \vdots & \vdots & \ddots & \vdots \\ x[N-1] & x[N-2] & \dots & x[N-p] \\ \hline x[N] & x[N-1] & \dots & x[N-p+1] \\ 0 & x[N] & \dots & x[N+2-p] \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & x[N] \end{array}\right]}_{\boldsymbol{X}_p=(N+p \times p)Matrix} \cdot \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix}}_{a_p} = \underbrace{\begin{bmatrix} -x[1] \\ -x[2] \\ \vdots \\ -x[N] \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{x_1}$$

2.8.2 Covariance Method Hayes p 183

No assumptions are made, only the existing data is used. A disadvantage is that the filter can become unstable. Another disadvantage is that A is not a Toeplitz matrix, so solving it is not as straightforward as with the Autocorrelation Method. However, the signal is reconstructed more accurately. For a data series corresponding to an impulse response, the filter is

perfectly reconstructed. With the disadvantage that unstable poles are also implemented. $a_p = -\left(\boldsymbol{X}_p^T \cdot \boldsymbol{X}_p\right)^{-1} \cdot \boldsymbol{X}_p^T \cdot x_p$ (Matlab: $a_p = -X_p \backslash x_p$)

$$\underbrace{\begin{bmatrix} x[p-1] & x[p-2] & \dots & x[0] \\ x[p] & x[p-1] & \dots & x[1] \\ \vdots & \vdots & \ddots & \vdots \\ x[N-1] & x[N-2] & \dots & x[N-p] \end{bmatrix}}_{\boldsymbol{X}_p} \cdot \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix}}_{a_p} = \underbrace{\begin{bmatrix} -x[p] \\ -x[p+1] \\ \vdots \\ -x[N] \end{bmatrix}}_{x_p}$$

(Note: These equations are a subset of the equations of the Autocorrelation Method.)

2.9 Stochastic Models

In stochastic models, generally the same models are used as in the deterministic world. Instead of the Dirac pulse, white noise is used at the input of the filter, and the autocorrelation is formed using the expected value. $r_x(k) = E\{x(n) \cdot x^*(n-k)\}$ Instead of the cross-correlation r_{gx}, the convolution of $b_q(k)$ and $h^*(-k)$ is used.
$c_q(k) = b_q(k) * h^*(-k) = \sum_{l=0}^{q-k} b_q(l+k) h^*(l)$

2.9.1 Autoregressive Moving Average Models, ARMA Hayes p 189

An ARMA(p,q) process is white noise that is filtered through an LTI system $H(z)$.

To determine the coefficients of $H(z) = \frac{B_q(z)}{A_p(z)} = \frac{\sum_{k=0}^{q} b_q(k) z^{-k}}{1 + \sum_{k=1}^{p} a_p(k) z^{-k}}$, the **extended Yule-Walker equations** are solved:

$$\underbrace{\begin{bmatrix} r_x[0] & r_x^*[1] & \dots & r_x^*[p] \\ r_x[1] & r_x[0] & \dots & r_x^*[p-1] \\ \vdots & \vdots & \ddots & \vdots \\ r_x[q] & r_x[q-1] & \dots & r_x[q-p] \\ r_x[q+1] & r_x[q] & \dots & r_x[q-p+1] \\ \vdots & \vdots & \ddots & \vdots \\ r_x[L] & r_x[L-1] & \dots & r_x[L-p] \end{bmatrix}}_{R_x} \cdot \underbrace{\begin{bmatrix} 1 \\ a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix}}_{a_p} = \begin{bmatrix} c_q[0] \\ c_q[1] \\ \vdots \\ c_q[q] \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

1. Calculate the A- coefficients: $R_q \cdot a_p = -r_{q+1}$

$$\underbrace{\begin{bmatrix} r_x[q] & r_x[q-1] & \dots & r_x[q-p+1] \\ r_x[q+1] & r_x[q] & \dots & r_x[q-p+2] \\ \vdots & \vdots & \ddots & \vdots \\ r_x[q+p-1] & r_x[q+p-2] & \dots & r_x[q] \end{bmatrix}}_{R_q} \cdot \underbrace{\begin{bmatrix} a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix}}_{a_p} = \underbrace{\begin{bmatrix} -r_x[q+1] \\ -r_x[q+2] \\ \vdots \\ -r_x[q+p] \end{bmatrix}}_{r_{q+1}}$$

2. To calculate the B- coefficients, one must go via the C- coefficients: $R_x \cdot a_p = c_q$

$$\underbrace{\begin{bmatrix} r_x[0] & r_x^*[1] & \dots & r_x^*[p] \\ r_x[1] & r_x[0] & \dots & r_x^*[p-1] \\ \vdots & \vdots & \ddots & \vdots \\ r_x[q] & r_x[q+1] & \dots & r_x[0] \end{bmatrix}}_{R_x} \cdot \underbrace{\begin{bmatrix} 1 \\ a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix}}_{a_p} = \underbrace{\begin{bmatrix} c_q[0] \\ c_q[1] \\ \vdots \\ c_q[q] \end{bmatrix}}_{c_q}$$

3. Write down $[C_q(z)]_+$: $[C_q(z)]_+ = \sum_{k=0}^{q} c_q(k) z^{-k}$
4. Calculate $[P_y(z)]_+ = \left[[C_q(z)]_+ \cdot A_p^*(1/z^*)\right]_+$ e.g. $[C_1(z)]_+ = \frac{45}{2} - 6z^{-1}$, $A_p^*(1/z^*) = 1 - 0.5z$; $[C_1(z)]_+ \cdot A_p^*(1/z^*) = -\frac{45}{4}z + \frac{51}{2} - 6z^{-1} \Rightarrow [P_y(z)]_+ = \frac{51}{2} - 6z^{-1}$
5. Make $[P_y(z)]_+$ symmetric. e.g.: $P_y(z) = -6z + \frac{51}{2} - 6z^{-1}$

6. Factorize $P_y(z) = B_q(z)B_q^*(1/z^*)$ so $B_p(z)$ can be used:
   e.g. $B_1(z) \cdot B_1^*(1/z^*) = \underbrace{2\sqrt{6}(1 - \frac{1}{4}z^{-1})}_{B_q(z)} \cdot \underbrace{2\sqrt{6}(1 - \frac{1}{4}z^{1})}_{B_q^*(1/z^*)}$
7. $H(z) = \frac{B_q(z)}{A_p(z)}$ e.g.: $H(z) = \frac{2\sqrt{6}(1-\frac{1}{4}z^{-1})}{A_p(z)}$

2.9.2 Autoregressive Models - AR Hayes p 194

In the AR(p) model, the ARMA(p,q=0) method is used, but only with poles. $H(z) = \frac{b(0)}{1+\sum_{k=1}^{p} a_p(k)z^{-k}}$.

$$\begin{bmatrix} r_x(0) & r_x^*(1) & r_x^*(2) & \cdots & r_x^*(p-1) \\ r_x(1) & r_x(0) & r_x^*(1) & \cdots & r_x^*(p-2) \\ r_x(2) & r_x(1) & r_x(0) & \cdots & r_x^*(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_x(p-1) & r_x(p-2) & r_x(p-3) & \cdots & r_x(0) \end{bmatrix} \cdot \begin{bmatrix} a_p(1) \\ a_p(2) \\ a_p(3) \\ \vdots \\ a_p(p) \end{bmatrix} = \begin{bmatrix} -r_x(1) \\ -r_x(2) \\ -r_x(3) \\ \vdots \\ -r_x(p) \end{bmatrix}$$

With $|b(0)|^2 = r_x(0) + \sum_{k=1}^{p} a_p(k)r_x(k)$

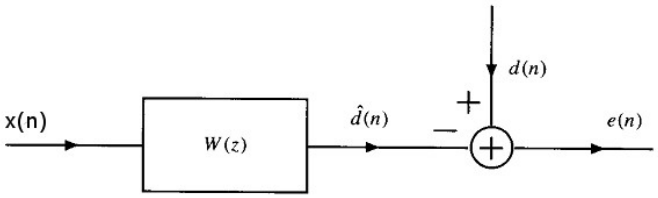
2.9.3 Moving Average Models - MA Hayes p 195

A MA process is a special form of an ARMA(p=0,q) process, which only has zeros.
$P_x(z) = \sum_{k=-q}^{q} r_x(k)z^{-k} = B_q(z)B_q^*(1/z^*)$

1. Spectrally factorize $P_x(z)$ into: $P_x(z) = \sigma_0^2 \cdot Q(z) \cdot Q^*(1/z^*) = \underbrace{\sigma_0^2}_{|b_q(0)|^2} \underbrace{\prod_{k=1}^{q}(1 - \beta_k z^{-1})}_{B_q(z)} \cdot \underbrace{\prod_{k=1}^{q}(1 - \beta_k^* z)}_{B_q^*(1/z^*)}$
2. Expand $B_q(z) \Rightarrow$ minimum phase filter (all zeros are inside the unit circle)
3. Or, expand $B_q^*(1/z^*)$ to get a maximum phase filter (all zeros are outside the unit circle)

3 Optimum Filters

An optimum filter minimizes the mean-square error $\xi = E\left(|e(n)|^2\right)$ $x(n) = d(n) + v(n)$ $e(n) = d(n) - \tilde{d}(n)$



The minimal error is **always orthogonal** to the data: $Ee(n)x^*(n-k) = 0 \; \forall \; k \in [0, p-1]$

3.1 Wiener FIR Filter Hayes p 337

Wiener-Hopf equations: $\sum_{l=0}^{p-1} w(l) r_x(k-l) = r_{dx}(k);\ \forall\ k \in [0, p-1]$

$$\underbrace{\begin{bmatrix} r_x[0] & r_x^*[1] & \dots & r_x^*[p-1] \\ r_x[1] & r_x[0] & \dots & r_x^*[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ r_x[p-1] & r_x[p-2] & \dots & r_x[0] \end{bmatrix}}_{R_x} \cdot \underbrace{\begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(p-1) \end{bmatrix}}_{w} = \underbrace{\begin{bmatrix} r_{dx}[0] \\ r_{dx}[1] \\ \vdots \\ r_{dx}[p-1] \end{bmatrix}}_{r_{dx}}$$

$$R_x \cdot w = r_{dx}$$

Correlations: $r_x(k) = E\{x(n)x^*(n-k)\}$

$r_{dx}(k) = E\{d(n)x^*(n-k)\}$

Minimum Error: $\xi_{min} = r_d(0) - \sum_{l=0}^{l} w(l) r_{dx}^*(l)$

3.1.1 Filtering Hayes p 339

The input signal is $x(n) = d(n) + v(n)$. Because the noise and data signals are uncorrollated and the noise has zero mean, the Wiener-Hopf equation simplifies as follows:

$r_x(k) = r_d(k) + r_v(k)$

$r_{dx}(k) = r_d(k) \quad \rightarrow [R_d + R_v] \cdot w = r_d$

with $R_v = \begin{bmatrix} \sigma_v^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_v^2 \end{bmatrix}$

3.1.2 Linear Prediction FIR Hayes p 342

$$\hat{x}(n+\alpha) = \sum_{k=0}^{p-1} w(k)[x(n-k) + v(n-k)]$$

$$\hookrightarrow \begin{bmatrix} r_y[0] & r_y^*[1] & \dots & r_y^*[p-1] \\ r_y[1] & r_y[0] & \dots & r_y^*[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ r_y[p-1] & r_y[p-2] & \dots & r_y[0] \end{bmatrix} \cdot \begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(p-1) \end{bmatrix} = \begin{bmatrix} r_x[\alpha] \\ r_x[\alpha+1] \\ \vdots \\ r_x[\alpha+p-1] \end{bmatrix}$$

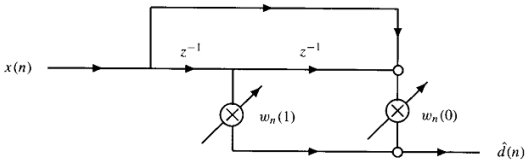
α are the steps to predict. Noise: $r_y(k) = r_x(k) + r_v(k)$

3.1.3 Linear Phase Filter Hayes p 17, exercise 9.16

All frequency are delayed same.

For **FIR**: the coefficients are symmetric $(w_n(0)z^1 + w_n(1)z^0 + w_n(0)z^{-1})$ or with a causal filter: $w_n(0)z^0 + w_n(1)z^{-1} + w_n(0)z^{-2}$

$$\begin{bmatrix} 2r_x[0] + 2r_x[2] & 2r_x[1] \\ 2r_x[1] & r_x[0] \end{bmatrix} \cdot \begin{bmatrix} w_n(0) \\ w_n(1) \end{bmatrix} = \underbrace{\begin{bmatrix} r_x[\alpha] + r_x[\alpha+2] \\ r_x[\alpha+1] \end{bmatrix}}_{\text{for linear prediction}} = \underbrace{\begin{bmatrix} = r_{dx}[0] + r_{dx}[2] \\ r_{dx}[1] \end{bmatrix}}_{\text{for optimal filter}}$$



How a linear phase FIR filter 2 order is implemented in hardware.

3.2 Wiener IIR Filter

3.2.1 Noncausal Hayes p 353

Wiener-Hopf equations: $H(e^{jw}) = \frac{P_{dx}(e^{jw})}{P_x(e^{jw})} \Rightarrow \frac{P_d(e^{jw})}{P_d(e^{jw}) + P_v(e^{jw})} = \frac{SNR(e^{j\omega})}{SNR(e^{j\omega})+1}$ (often called **the** Wiener Filter)

Correlations: $r_x(k) = E\{x(n)x^*(n-k)\}$

$r_{dx}(k) = E\{d(n)x^*(n-k)\}$

Minimum Error: $\xi_{min} = r_d(0) - \sum_{l=-\infty}^{\infty} h(l) r_{dx}^*(l) = \frac{1}{2\pi} \int_{-\pi}^{\pi} [P_d(e^{jw}) - H(e^{jw}) P_{dx}^*(e^{jw})] dw$

$= r_d(0) - \frac{1}{2\pi} \int_{-\pi}^{\pi} [H(e^{jw}) P_{dx}^*(e^{jw})] dw$

3.2.2 Causal Hayes p 358

Spectral Factorization: $P_x(z) = \sigma_0^2 Q(z) Q^*(z^{-1})$

System function: $H(z) = \frac{1}{\sigma_0^2 Q(z)} [\frac{P_{dx}(z)}{Q^*(z^{-1})}]_+$

Minimum Error: $\xi_{min} = r_d(0) - \sum_{l=0}^{\infty} h(l) r_{dx}^*(l) = \frac{1}{2\pi} \int_{-\pi}^{\pi} [P_d(e^{jw}) - H(e^{jw}) P_{dx}^*(e^{jw})] dw$

Example Hayes p 362

3.2.3 Linear Prediction IIR Hayes p 365

System function: $r_{dx}(k) = r_x(k+\alpha) \qquad \alpha =$ steps to predict.

$H(z) = \frac{1}{Q(z)} [z^\alpha Q(z)]_+ \qquad \rightarrow$ monic, one step $H(z) = z[1 - \frac{1}{Q(z)}]$

Minimum Error: $\xi_{min} = r_d(0) - \sum_{l=0}^{\infty} h(l) r_{dx}^*(l) = \frac{1}{2\pi} \int_{-\pi}^{\pi} [P_d(e^{jw}) - H(e^{jw}) P_{dx}^*(e^{jw})] dw \rightarrow$ monic, one step $\xi_{min} = \sigma_0^2$

3.2.4 Wiener Deconvolution Hayes p 369

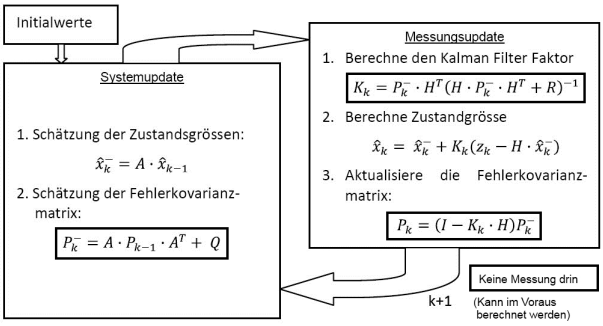
Deconvolutate a signal $x(n) = d(n) * g(n) + w(n)$ (not $x(n) = d(n) + g(n)$) it's not easy. At the best, the signal could be reconstruct as: $\hat{D}(e^{\omega}) = D(e^{j\omega}) + \frac{W(e^{j\omega})}{G(e^{j\omega})} = D(e^{j\omega}) + V(e^{j\omega})$ $V(e^{j\omega})$ isn't anymore white noise. Espacialy if $G(e^{j\omega})$ becomes small the noise will be amplified.
System function: $H(z) = \frac{1}{G(z)} \left[\frac{P_d(z)}{P_d(z) + P_w(z)/|G(z)|^2} \right]$

3.3 Discrete Kalman Filter

The Kalman Filter is a Best Linear Unbiased Estimator (BLUE).
Basic Idea The Kalman Filter estimates the Kalman Gain based on the least squares error. Specifically, the Kalman Filter defines measurement values using state variables and a noise signal.

3.3.1 Prerequisites

- A physical model/system development (for the standard version, even a linear model)
- Measurement values, sensor fusion possible (multiple measurements for one system state)
- For standard Kalman: linear relationship between measurement values and state variables



- $\hat{x}_{k-1}$ and P_{k-1} require initialization values
- P_K and K_K can be pre-calculated offline if the system does not change

3.3.2 Function

- A System development matrix: Physical model
- K Kalman Gain: Weights internal estimation and the measurements of individual sensors (0=Only internal estimation;1=only measurement)
- H Measurement matrix: Relationship between measurement and state variables
- P (Error covariance matrix) Estimate of the error. The larger it is, the more the current measurement is considered. In the normal Kalman Filter, this converges over time.
- z: Measurement values (also from several similar sensors possible)
- x: State variables
- Initial value: System values arbitrary; Error covariance matrix chosen very large, so that initially only the measurement is considered
- Q Standard deviation of the system (system noise)
- R Standard deviation of the sensors (measurement noise)
- The ratio of Q to R is responsible for the filter setting.
  $\frac{Q}{R}$ large → trusts the measurement data more
  $\frac{Q}{R}$ small → trusts the system properties more

3.3.3 Kalman Algorithm

Idea: Best estimation when process and observer equation as well as some initial conditions are given. $\hat{\mathbf{x}}(n|n)$ is the prediction, $\mathbf{P}$ is the error covariance matrix, $\mathbf{K}$ is the Kalman gain. The index $\hat{\mathbf{x}}(a|b)$ a stands for the iteration number (n = now) and

b is which input data has to be taken ($n-1$ = data up to the last iteration).

State equation: $\mathbf{x}(n) = \mathbf{A}(n-1)\mathbf{x}(n-1) + \mathbf{w}(n)$ $\quad \mathbf{Q}_w$ = noise covariance matrix $\underbrace{= \sigma_w^2}_{\text{when one dimensional}}$

Observer equation: $\mathbf{y}(n) = \mathbf{C}(n)\mathbf{x}(n) + \mathbf{v}(n)$ $\quad \mathbf{Q}_V$ = noise covariance matrix $\underbrace{= \sigma_v^2}_{\text{when one dimensional}}$

Initial condition:
$$\mathbf{x}(0|0) = E\{\mathbf{x}(0)\} \quad \text{might be zero}$$
$$\mathbf{P}(0|0) = E\{\mathbf{x}(0)\mathbf{x}^{\mathrm{T}}(0)\} \quad \text{should be high}$$

Prediction:
$$\hat{\mathbf{x}}(n|n-1) = \mathbf{A}(n-1)\hat{\mathbf{x}}(n-1|n-1) \tag{1}$$
$$\mathbf{P}(n|n-1) = \mathbf{A}(n-1)\mathbf{P}(n-1|n-1)\mathbf{A}^{\mathrm{T}}(n-1) + \mathbf{Q}_w(n) \tag{2}$$

Update:
$$\mathbf{K}(n) = \mathbf{P}(n|n-1)\mathbf{C}^{\mathrm{T}}(n)\left[\mathbf{C}(n)\mathbf{P}(n|n-1)\mathbf{C}^{\mathrm{T}}(n) + \mathbf{Q}_v(n)\right]^{-1} \tag{3}$$
$$\hat{\mathbf{x}}(n|n) = \hat{\mathbf{x}}(n|n-1) + \mathbf{K}(n)\left[\mathbf{y}(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n|n-1)\right] \tag{4}$$
$$\mathbf{P}(n|n) = \left[\mathbf{I} - \mathbf{K}(n)\mathbf{C}(n)\right]\mathbf{P}(n|n-1) \tag{5}$$
continue with prediction step (6)

The steady-state is reached when $\mathbf{P}(n|n) = \mathbf{P}(n-1|n-1)$ and $\mathbf{K}(n) = \mathbf{P}(n|n)$.

3.3.4 Simplified Principle of the Kalman Filter

Given two inaccurate sensors T_1 and T_2 (uncorrelated), measuring the same signal with known standard deviations σ_1 and σ_2. The best estimator $\hat{T}$ is calculated as follows:

$$\hat{T} = \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2} \cdot T_2 + \frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2} \cdot T_1$$

Given two correlated signals T_1 and T_2 with known σ_1, σ_2, and the correlation coefficient ρ. The best estimator $\hat{T}$ is calculated as follows:

$$\hat{T} = k_1 \cdot T_1 + k_2 \cdot T_2$$
$$k_1 = \frac{\sigma_2^2 - \rho\sigma_1\sigma_2}{\sigma_1^2 - 2\rho\sigma_1\sigma_2 + \sigma_2^2}$$
$$k_2 = 1 - k_1 = \frac{\sigma_1^2 - \rho\sigma_1\sigma_2}{\sigma_1^2 - 2\rho\sigma_1\sigma_2 + \sigma_2^2}$$

3.3.5 Kalman Modelling of Known System Behaviour

When a system has known behaviour, the state space representation can be used to find the Kalman matrices. The system is interpreted as a measurement with input $w(n)$ (measurement noise), output $y(n)$ and an added process noise $v(n)$ that is directly added to $y(n)$.

1.: General transfer function:	$H(z) = \frac{Y(z)}{W(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_q z^{-q}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_p z^{-p}}$ Warning: The z-Transform of a known model does not always yield the correct difference equation! For example if a pole is on the unit circle e.g $y(n) = (-1)^n \circ\!\!-\!\!\bullet \frac{1}{1+z^{-1}} \bullet\!\!-\!\!\circ y(n) \neq 1 - y(n-1)$. →What you need to do is to find the difference equation! The z-Transform might be one way to do that, but it doesn't always work.
2.: Difference equation:	$y(n) = v(n) + b_0 w(n) + \ldots + b_q w(n-q) - a_1 y(n-1) + \ldots - a_q y(n-q)$
3.: States:	$x(n) = w(n) - a_1 x(n-1) - a_2 x(n-2) - \ldots - a_p x(n-p)$ $x(n-1) = x(n-1)$ $\vdots = \vdots$ $x(n - max(p,q) = x(n - max(p,q))$

	For more information see DigSig2 cheat sheet section 1.2 Canonical Form. (Flow diagram and states)
4.: State space representation:	$A = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_{p-1} & -a_p \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} w(n) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ $C = \begin{bmatrix} b_0 & b_1 & \dots & b_q \end{bmatrix}, D = v(n)$

4 Spectrum Estimation

The problem of estimation a spectrum is, that there is never an infinity number of samples. Therefore, the autocorrelation is always multiplied with a window.

4.1 Nonparametric Method - Periodogram Hayes p 393

autocorrelation: $\hat{r}_x(k) = \frac{1}{N} \sum_{n=0}^{N-1-k} x(n+k)x^*(n)$ $k = 0, 1, \dots, N-1$

power spectrum: $\hat{P}_{per}(e^{jw}) = \frac{1}{N} X_N(e^{jw}) X_N^*(e^{jw}) = \frac{1}{N} \left| X_N(e^{jw}) \right|^2 = \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n) e^{-jn\omega} \right|^2$

computing: The easiest way to compute the Periodogram is to compute the FFT, take the absolute values and square it

$x_N(n) \xrightarrow{DFT} X_N(e^{jw}) \Rightarrow \frac{1}{N} |X_N(k)|^2 = \hat{P}_{per}(e^{j2\pi k/N})$

Bias: $E\left\{ \hat{P}_{per}(e^{jw}) \right\} = \frac{1}{2\pi} P_x(e^{jw}) * W_B(e^{jw})$ $W_B \rightarrow$ Bartlet window

Resolution: $Res[\hat{P}_{per}(e^{jw})] = \Delta w = 0.89 \frac{2\pi}{N}$ Resolution = 3dB BW $(\Delta\omega)_{3dB}$

The Problem of the Periodogram is that the variance doesn't get smaller and it's far **too big!!!**.

Variance $Var\left\{ \hat{P}_{per}^{(i)}(e^{j\omega}) \right\} \approx P_x^2(e^{j\omega})$

Because the samples are multiplied with a rectangle window, a dirac in the original power spectrum becomes a sinc with a first sidelobe just 13 dB lower than the mainlobe.

The Periodogram is okay to be used when a high frequency resolution is required but should not be used to estimate the spectrum accurately.

4.2 Nonparametric Method - Modified Periodogram Hayes p 408

To have a better sidelobe ratio, different windows are used to multiplied the given data.

autocorrelation: $\hat{r}_x(k) = \frac{1}{N} \sum_{n=0}^{N-1-k} x(n+k)x^*(n)$ $k = 0, 1, \dots, N-1$

power spectrum: $\hat{P}_M(e^{jw}) = \frac{1}{NU} \left| \sum_{n=\infty}^{\infty} w(n)x(n)e^{-jn\omega} \right|^2$

scaling Because the window itself hasn't normaly a power of 1, the power spectrum has to be scaled by factor U:

$U = \frac{1}{N} \sum_{n=0}^{N-1} |w(n)|^2$

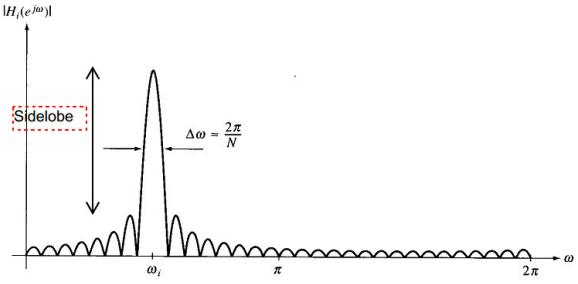
Bias: $E\left\{ \hat{P}_M(e^{jw}) \right\} = \frac{1}{2\pi NU} P_x(e^{jw}) * |W(e^{jw})|^2$

Resolution : Window dependent

The problem of variance isn't solved.

Variance $Var\left\{ \hat{P}_M^{(i)}(e^{j\omega}) \right\} \approx P_x^2(e^{j\omega})$

Window	Sidelobe Level (dB)	3dB BW $(\Delta\omega)_{3dB}$
Rectangular	-13	0.89 $(2\pi/$ N)
Bartlett (triangle)	-27	1.28 $(2\pi/$ N)
Hann	-32	1.44 $(2\pi/$ N)
Hamming	-43	1.30 $(2\pi/$ N)
Backman	-58	1.68 $(2\pi/$ N)



4.3 Nonparametric Method - Bartlett's Method Hayes p 412

The idea of Bartlett was to take K non-overlapping sub-sequences (lenght L) of the given data. A Periodogram (rectangular window) has to be calculated for every sub-sequence and then averaged. $N = K \cdot L$

power spectrum: $\hat{P}_B(e^{jw}) = E\left\{\frac{1}{L}\left|X_L(e^{jw})\right|^2\right\} = \frac{1}{N}\sum_{i=0}^{K-1}\left|\sum_{n=0}^{L-1} x(n+iL)e^{-jn\omega}\right|^2$

Bias: $E\left\{\hat{P}_B(e^{jw})\right\} = \frac{1}{2\pi}P_x(e^{jw}) * W_B(e^{jw})$ W → Bartlet window

Resolution: $Res\left[\hat{P}_B(e^{jw})\right] = \Delta w = 0.89 \cdot K \cdot \frac{2\pi}{N}$ Resolution = 3dB BW $(\Delta\omega)_{3dB}$

Variance $Var\left\{\hat{P}_B(e^{j\omega})\right\} = \frac{1}{K}Var\left\{\hat{P}^{(i)}_{per}(e^{j\omega})\right\} \approx \frac{1}{K}P_x^2(e^{j\omega})$

So the problem of a bad and inconsistent variance could be eliminated. But the resolution is K times worse.

4.4 Nonparametric Method - Welch's Method Hayes p 415

Welch' method is a modification of the Bartlett's method. It uses overlapping of the sub-sequences. When maintaining $L =$ section length, $K =$ number of sections, $D =$ overlap, the formula $\boxed{N = L + D(K-1)}$ can be established. This means, that if $D = L$ there is no overlap and $D = L/2$ means 50% overlap.

Power spectrum: $\hat{P}_W(e^{jw}) = \frac{1}{KLU}\sum_{i=0}^{K-1}\left|\sum_{n=0}^{L-1} w(n)\cdot x(n+iD)e^{-jn\omega}\right|^2$

$U = \frac{1}{L}\sum_{n=0}^{L-1}|w(n)|^2 =$ equalization factor for the given window

Bias: $E\left\{\hat{P}_W(e^{jw})\right\} = \frac{1}{2\pi LU}P_x(e^{jw}) * |W(e^{jw})|^2$

Resolution: Window dependent

Variance $Var\left\{\hat{P}_W(e^{j\omega})\right\} \approx \frac{9}{16}\frac{L}{N}P_x^2(e^{j\omega}) = \frac{9}{16}Var\left\{\hat{P}_B(e^{j\omega})\right\}$ with Barlett window and 50% overlap

4.5 Nonparametric Method - Blackman-Tukey Hayes p 420

The idea of Blackman Tukey is to minimalize the effect of the of the autocorrelation samples with a *big lag*, because with finite data record the variance of $\hat{r_x}(k)$ will be large. This is because these samples just have some few samples that can be averaged. Therefore, the autocorrelation is weighted with a window (like Hann, Bartlett or Hamming).
The window length M has to be much smaller than the length of the autocorrelation N: $M \ll N \Rightarrow M < \frac{1}{5}N$

Power spectrum: $\hat{P}_{BT}(e^{jw}) = \sum_{k=-M}^{M}\hat{r}_x(k)w(k)e^{-jk\omega}$ $\hat{P}_{BT}(z) = \sum_{k=-M}^{M}\hat{r}_x(k)w(k)z^{-k}$

Bias: $E\left\{\hat{P}_{BT}(e^{jw})\right\} = \frac{1}{2\pi}P_x(e^{jw}) * W(e^{jw})$ W → choosen window

Resolution: Winow dependent

Variance $Var\left\{\hat{P}_{BT}(e^{j\omega})\right\} \approx P_x^2(e^{j\omega})\frac{1}{N}\sum_{k=-M}^{M}w^2(k)$

4.6 Performance Comparison Hayes p 424

To compare different methods the following criteria can be used:

Variability:	$\nu = \frac{Var\{\hat{P}_x(e^{j\omega})\}}{E^2\{\hat{P}_x(e^{j\omega})\}} = \frac{\sigma^2}{\mu^2}$	it's also called normalized variance (smaller is better)
resolution	$\Delta\omega$	smaller is better
figure of merit	$M = \nu \cdot \Delta\omega$	smaller is better

	Variability ν	Resolution $\Delta\omega$	Figure of Merit M
Periodogram	1	$0.89\frac{2\pi}{N}$	$0.89\frac{2\pi}{N}$
Bartlett	$\frac{1}{K}$	$0.89 \cdot K \cdot \frac{2\pi}{N}$	$0.89\frac{2\pi}{N}$
Welch	$\frac{9}{8 \cdot K}$	$1.28\frac{2\pi}{L}$	$0.72\frac{2\pi}{N}$
Blackman-Tukey (Barlett)	$\frac{2 \cdot M}{3 \cdot N}$	$0.64\frac{2\pi}{M}$	$0.43\frac{2\pi}{N}$

4.7 Parametric Method - Minimum Variance Method Hayes p 426

Power spectrum: $\hat{P}_{MV}(e^{jw}) = \frac{p+1}{e^H R_x^{-1} e}$ R_x = Autocorrelation Matrix $e = [1, e^{j\omega}, \ldots, e^{jp\omega}]^T$

4.8 Parametric Method - Maximum Entropy Method Hayes p 433

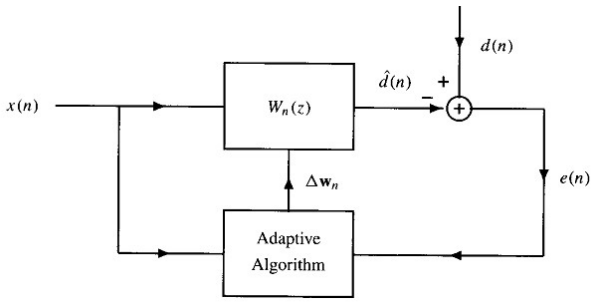
The idea is to use a infinite autocorrelation. For that we have to interpolate the left (übrigen) samples. The maximum entropy method (MEM) use a all pole interpolation. Because of this, the MEM is really good **if and only if** a spectrum of an **all-pole model** is given. The coefficients $a_p(k)$ are calculated using the autoregressive model method in section 2.9.2.

Power spectrum: $\hat{P}_{MEM}(e^{jw}) = \frac{|b(0)|^2}{\left|1+\sum_{k=1}^{p} a_p(k)e^{-jk\omega}\right|^2} = \frac{|b(0)|^2}{|e^H a_p|^2}$ $a_p = [1, a_p(1), \ldots, a_p(k)]^T$ $e = [1, e^{j\omega}, \ldots, e^{jp\omega}]^T$

$$\epsilon_p = |b(0)|^2 = r_x(0) + \sum_{k=1}^{p} a_p(k) r_x^*(k)$$

5 Adaptive Filtering

5.1 Adaptive FIR Filters Hayes p 497



Goal: The filter coefficients w_n should be adapted with recursive modification Δw_n itself.

application area:

- System identification: estimate $H[k], h[n]$
- Equalize transmission canal with known sequence: GSM, ...
- Adaptive noise cancelling (in ear plugs)
- Echo cancelling on phone calls

The MSE (mean square error) $\xi(w_0, w_1, \ldots, w_n) = E\left(|e(i)|^2\right) = E\left(\left[d(i) - \sum_{k=0}^{N} w_k x(i-k)\right]^2\right)$ should be minimizied. The solution for stationary process is the Wiener-Hopf equations:

$$R_x \cdot w = r_{dx}$$

If the process isn't stationary, the filter coefficients have to be updated every step for the optimal result: $w_{n+1} = w_n + \Delta w_n$
The adaptive filter should

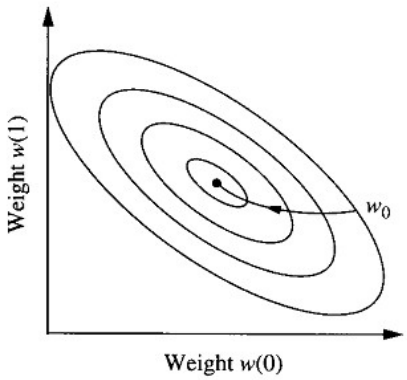
- converge **to the Wiener-Hopf equations** if the environment is stationary. $\lim_{n\to\infty} w_n = R_x^{-1} \cdot r_{dx}$
- be able to compute the update coefficients Δw_n, **without knowledge** of R_x and r_{dx}
- be able to adapt to the **changing statistics** and track the solution, if the environment is nonstationary

5.1.1 Steepest Descent Adaptive Filter Hayes p 499

To solve the Wiener-Hopf equations, in every step the autocorrelation matrix has to be inverted and with the cross-correlation multiplied. To reduce this calculations, the coefficient are updated into optimal direction with a small step.

$$w_{n+1} = w_n - \mu \cdot \frac{\delta \xi}{\delta w_n} = w_n - \mu \cdot \bigtriangledown \xi(n) \Rightarrow \boxed{w_{n+1} = w_n + \mu \cdot E\left\{e(n)x^*(n)\right\}}$$

$$\boxed{w_{n+1} = w_n - \mu \cdot [R_x w_n - r_{dx}]}$$



$$0 < \mu < \frac{2}{\lambda_{max}}; \lambda_{max} = \text{maximum eigenvalue of autocorrelation matrix } R_x$$

To compute the $E\left\{e(n)x^*(n)\right\}$, a averaging over the last L sample has to be done.

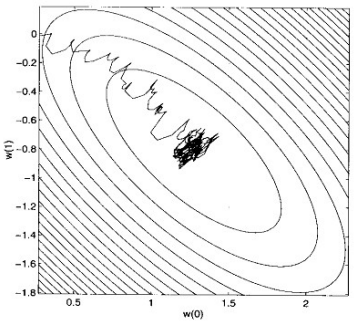
$$E\left\{e(n)x^*(n)\right\} = \frac{1}{L}\sum_{l=0}^{L-1} e(n-l)x^*(n-l)$$

How fast converge the solution/filter: $\tau = \max\{\tau_k\} \approx \frac{1}{\mu \lambda_{min}}$

5.1.2 LMS Algorithm Hayes p 505

The LMS algorithm use the same idea as the steepest descent, but just choose the last sample (L=1).

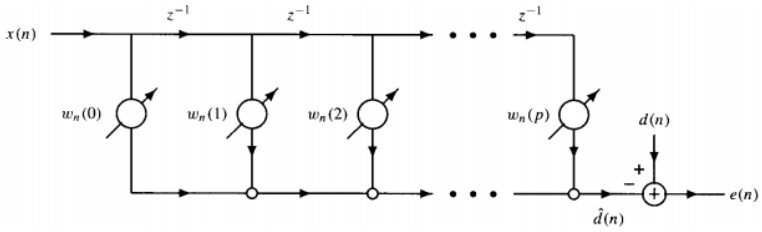
$$\boxed{w_{n+1} = w_n + \mu \cdot e(n) \cdot x^*(n)}$$



The LMS just converges in the mean, if $0 < \mu < \frac{2}{(p+1)E\left(|x(n)|^2\right)} = \frac{2}{(p+1)r_x(0)} = \frac{2}{(p+1)\cdot\frac{1}{N}\sum_{k=0}^{N-1}|x(n-k)|^2}$. The filter coefficients converge to $\hat{d}(n) = d(n)$ or $e(n) = 0$. That means that filter coefficients always will move, but in the mean they are correct.

Parameters: p = Filter order
Init: $w_0 = 0$
Computation:

1. calculate filter output: $\hat{d}(n) = w_n^T \cdot x(n)$
2. calculate error: $e(n) = d(n) - \hat{d}(n)$
3. update filter coefficients: $w_{n+1} = w_n + \mu e(n) x^*(n)$



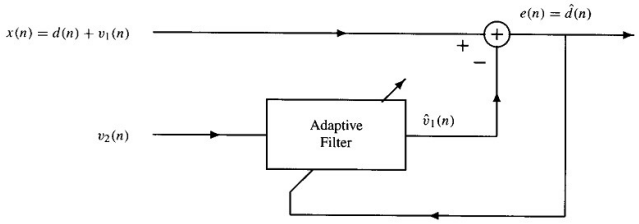
5.1.3 Normalized LMS (NLMS) Hayes p 514

The NLMS is most command used, because it's easy to program, fast to compute and convergence quite fast.

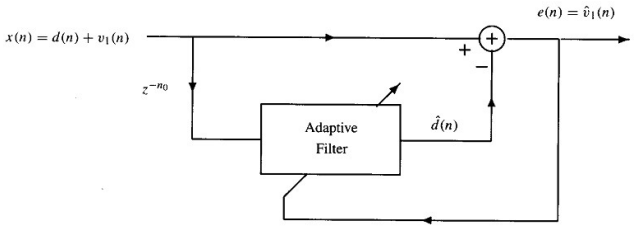
The idea is to adapt also the stepsize $\mu(n) = \frac{\beta}{\|x(n)\|^2} \Rightarrow \boxed{w_{n+1} = w_n + \beta\frac{x^*(n)}{\epsilon + \|x(n)\|^2}e(n)}$

- $\|x(n)\|^2$ is the squared sum over p+1 samples $\rightarrow \sum_{k=0}^{p}|x(n-k)|^2$, recursively: $\|x(n+1)\|^2 = \|x(n)\| + |x(n+1)|^2 - |x(n-p)|^2$
- ϵ is used for small $\|x(n)\|$ (numerical stability)
- β is a normalized step size with $0 < \beta < 2$

5.1.4 Applications - Noise Cancellation Hayes p 517



Input signal $x(n)$ with noise (e.g. car noise). The second input measures only the noise (e.g. separate mic). Therefore, v_1 and v_2 are correlated. The filter now guesses the first noise so that the error signal gives the wanted signal $d(n)$.



Input signal $x(n)$ with noise (e.g. car noise). The signal $d(n)$ has to be narrowband and $v_1(n)$ has to be broadband. (d has to be a "slow" and v_1 has to be a "fast" changing signal) The filter predicts now $d(n)$ and the error signal is the noise v_1.

5.2 Adaptive Recursive Filters (won't be tested) Hayes p 534

Requires in comparison to the LMS algorithm $20M$ only $2M$ iterations until it converges. RLS uses all past information for the calculation of the correction; LMS uses only current information.
$\mathbf{s}(n) = [s(n), s(n-1), \ldots, s(n-N+1)]^T$
Forgetting factor ($\mu = 1$), Initialization factor ($\delta = 1$)

1. Initialize: $n = 1; \mathbf{P}(0) = \gamma \cdot \mathbf{I}; \hat{\mathbf{c}}(0) = 0$
2. Gain vector: $\mathbf{k}(n) = \frac{\mathbf{P}(n-1)\mathbf{s}(n)}{1+\mathbf{s}^T(n)\mathbf{P}(n-1)\mathbf{s}(n)}$
3. True estimation error: $\eta(n) = r(n) - \mathbf{s}^T(n)\hat{\mathbf{c}}(n-1)$
4. Update coefficients: $\hat{\mathbf{c}}(n) = \hat{\mathbf{c}}(n-1) + \mathbf{k}(n)\eta(n)$
5. Error correlation matrix: $\mathbf{P}(n) = \frac{1}{\mu}[\mathbf{P}(n-1) - \mathbf{k}(n)\mathbf{u}^T(n)\mathbf{P}(n-1)]$
6. $n++$ and back to step 2