Digital signal processing

Chapter 4. Random signal processing

Zhenyu Huang

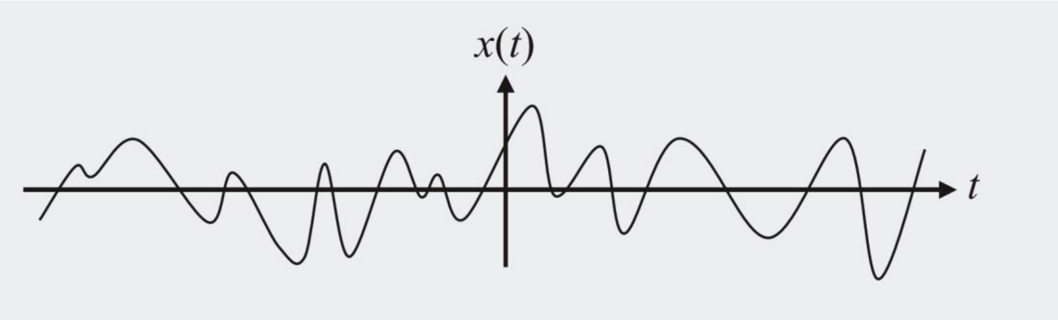
bighuang@sjtu.edu.cn

Study Points

1. Statistics of random process (2h)
2. Correlation (2h)
3. Spectrum density (2h)
4. Transmission (2h)
5. Case study 2 (2h)：System modelling and Identification
6. Adaptive filter (2h)
7. Course review (2h)

4.1 Random process

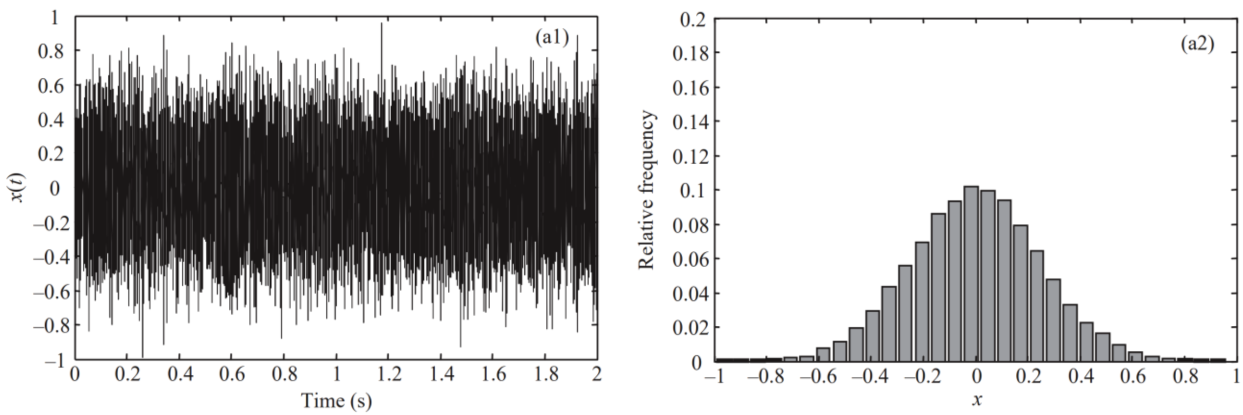
- A **random** (or stochastic, uncertain) process includes non-deterministic signals, and cannot be predicted exactly. We need some basic ideas of probability and statistics concepts to describe random processes.
- And the signal processing of the random process can de generally divided into the time-domain and frequency -domain methods, e.g. statistic parameters, correlation, power (energy) spectral density, transmission, transfer function, etc.



4.1 Stationary random process

- The random process is said to be **stationary or ergodic process** if the probability distributions obtained for the **ensemble** do **not** depend on absolute time. The term, *stationary*, refers to that all the averages are **independent** of absolute time and, specifically, that the mean, mean square, root of mean square (RMS), variance and standard deviation are independent of time altogether.

集合所有的统计平均值都不随时间变化外，从任一样本求得的时间平均值也与集合的统计平均值相同

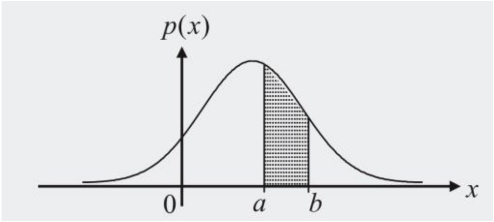


- Since all engineering random processes must have a beginning and ending, they cannot be truly stationary, but for practical purposes it is often adequate to assume that a process is stationary for the majority of its lifetime, or that it can be divided into several separate periods each of which is approximately stationary.

4.1 Probability density function

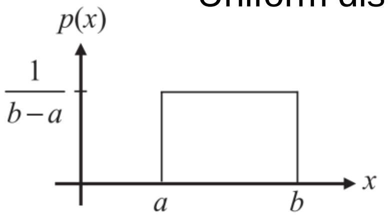
- Probability density function (continuous)

$$p(x) = \lim_{\delta x \to 0} \frac{P\left[x < X \le x + \delta x\right]}{\delta x}$$



- Well-known distributions

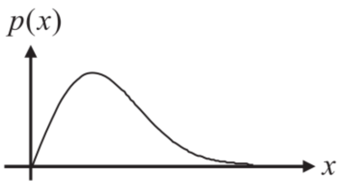
Uniform distribution



Mean value: $\mu_x = \dfrac{a+b}{2}$

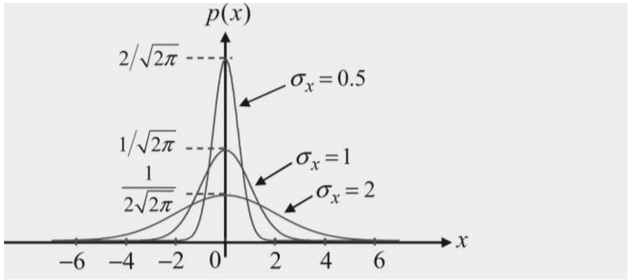
Variance: $\sigma_x^2 = \dfrac{(b-a)^2}{12}$

Rayleigh distribution



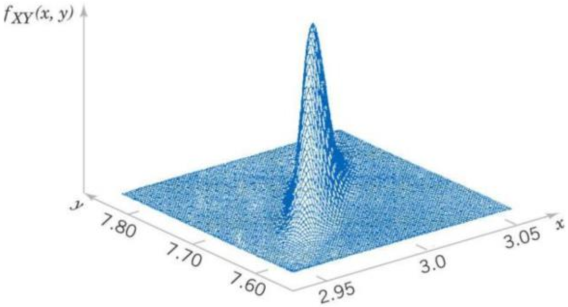
Normal distribution

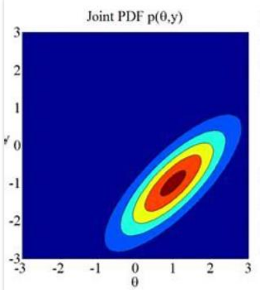
$$p(x) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-(x-\mu_x)^2/2\sigma_x^2}$$



- Joint probability density function

$$p(x, y) = \lim_{\substack{\delta x \to 0 \\ \delta y \to 0}} \frac{P[x < X \le x + \delta x \cap y < Y \le y + \delta y]}{\delta x \delta y}$$





4.1 Statistical parameters of a random process

- Mean, average, expectation, first moment

$$E[X] = \int_{-\infty}^{\infty} x p(x) dx$$

- Mean square, second moment, root of mean square (RMS)

$$E[X^2] = \int_{-\infty}^{\infty} x^2 p(x) dx$$

$$\text{RMS} = \sqrt{E[X^2]}$$

- Variance, standard deviation

$$\text{Var}(X) = \sigma_x^2 = E[(X - \mu_x)^2] = \int_{-\infty}^{\infty} (x - \mu_x)^2 p(x) dx$$

Standard deviation σ_x

- The third (Skewness) and fourth moments are useful in considerations of processes that are non-Gaussian distribution. Following is the *k*th moment:

$$M_k' = E[X^k] = \int_{-\infty}^{\infty} x^k p(x) dx$$

4.1 Statistical parameters of two random variables

- Mean of two variables X and Y (General Second moment)

$$E[XY] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xyp(x, y)dxdy$$

$$\boxed{E[X^2] = \int_{-\infty}^{\infty} x^2 p(x)dx}$$

- Covariance

$$\mathrm{Cov}(X, Y) = \sigma_{xy} = E[(X-\mu_x)(Y-\mu_y)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\mu_x)(y-\mu_y)p(x, y)dxdy$$

$$\boxed{\mathrm{Var}(X) = \sigma_x^2 = E[(X-\mu_x)^2] = \int_{-\infty}^{\infty} (x-\mu_x)^2 p(x)dx}$$

- *E*[*XY*] is called the correlation between *X* and *Y*, and Cov(*X*, *Y*) is called the covariance between X and Y . They are related by

$$\mathrm{Cov}(X, Y) = E[XY] - \mu_x\mu_y$$

- If the means of *X* and *Y* are zeros, the correlation is the same as the covariance.

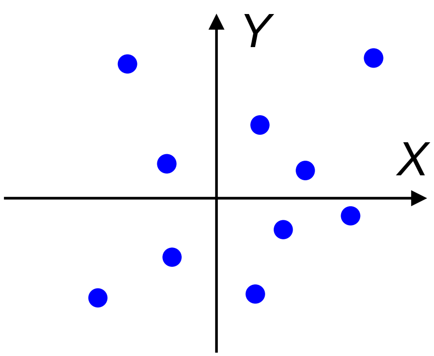
4.1 Correlation of two random variables

- Covariance: $\mathrm{Cov}(X, Y) = E[X\,Y] - \mu_x \mu_y$

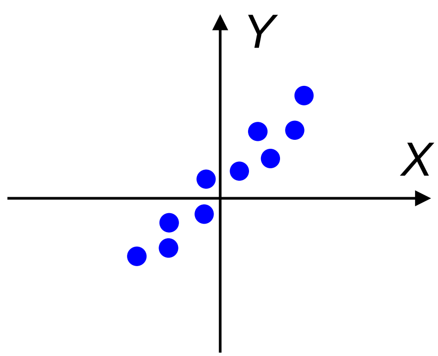
- The **linear** correlation can be evaluated by the **correlation coefficient** (Karl Pearson), which means Cov(x,y) is scaled by $\sigma_x \sigma_y$

$$\rho_{xy} = \frac{\mathrm{Cov}(X,Y)}{\sigma_x \sigma_y} = \frac{E[XY] - \mu_x \mu_y}{\sigma_x \sigma_y}$$

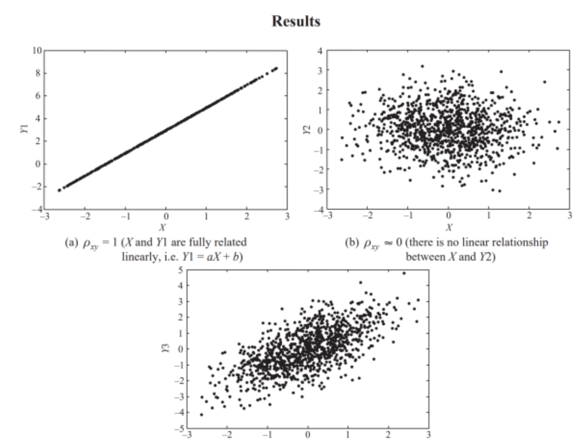
- Suppose that each pair of values is represented by a point on a graph of *y* against *x*. The correlated pair exhibits **linearly** correlated pattern.



$\rho \to 0$, uncorrelated



$\rho \to \pm 1$, correlated



(a) $\rho_{xy} = 1$ (X and $Y1$ are fully related linearly, i.e. $Y1 = aX + b$)

(b) $\rho_{xy} \approx 0$ (there is no linear relationship between X and $Y2$)

(c) $\rho_{xy} = 0.6529$ (not fully related, but obviously there is some degree of linear relationship)

$$\rho_{xy} = \begin{cases} \pm 1, & \text{perfect correlation in the linear way} \\ 0, & \text{not correlated in the linear way, maybe dependent in nonlinear ways} \end{cases}$$

4.1 Computation of a random process in the digital form

- Suppose we have a set of data ($x1$, $x2$,..., x_N) collected from N measurements of a random variable X. Then the estimation of the mean is

$$\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$$

- For the estimations of the variance σ^2 and covariance σ^2_{xy} are

$$s_x^2 = \frac{1}{N-1} \sum_{n=1}^{N} (x_n - \bar{x})^2$$

$$s_{xy} = \frac{1}{N-1} \sum_{n=1}^{N} (x_n - \bar{x})(y_n - \bar{y})$$

N.B. The estimator with the divisor N usually underestimates the variance. Thus the divisor $N - 1$ is more frequently used. This gives an unbiased sample variance. Their differences are usually insignificant if N is ‘large enough’

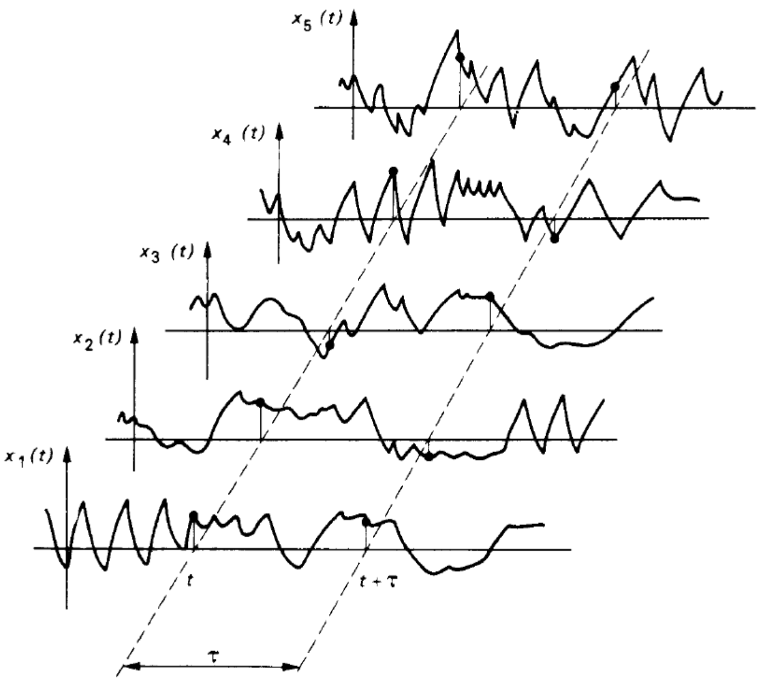
4.2 Autocorrelation

- The **autocorrelation** function for a random process $x(t)$ is defined as the average value of the product $x(t)x(t+\tau)$. The process is sampled at time t and then again at time $t+\tau$, and the average value of the product $E[x(t)x(t+\tau)]$, calculated for the ensemble over T.

Records of a time section T

- **Autocorrelation function**:

$$R_x(\tau) \equiv E[x(t)x(t+\tau)] = \frac{1}{T}\int_0^T x(t)x(t+\tau)\mathrm{d}t$$



4.2 Autocorrelation properties

- Correlation coefficient of x and $x(t+\tau)$

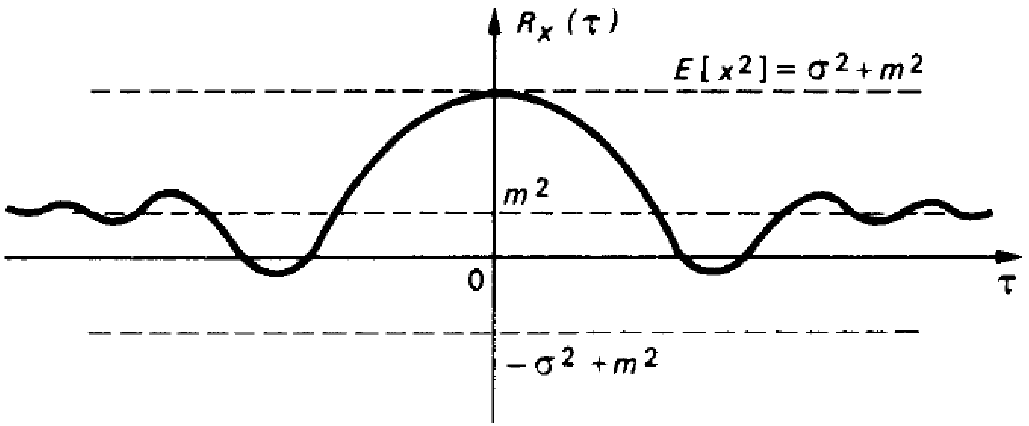
$$\rho = \frac{E[\{x(t)-m\}\{x(t+\tau)-m)\}]}{\sigma^2} = \frac{E[x(t)x(t+\tau)] - mE[x(t+\tau)] - mE[x(t)] + m^2}{\sigma^2}$$

$$= \frac{R_x(\tau) - m^2}{\sigma^2}$$

- $-1 \le \rho \le 1 \quad \rightarrow \qquad -\sigma^2 + m^2 \le R_x(\tau) \le \sigma^2 + m^2$

- $\rho = \begin{cases} \pm 1, & \text{perfect correlation} \\ 0, & \text{no correlation} \end{cases} \qquad \begin{array}{l} R_x(\tau = 0) = E[x^2] = \sigma^2 + m^2 \\ R_x(\tau = \infty) = m^2 \end{array}$

- $R_x(\tau) \equiv E[x(t)x(t+\tau)] = E[x(t)x(t-\tau)] \equiv R_x(-\tau)$



4.2 Cross-correlation

The cross-correlation functions between two different stationary random functions of time $x(t)$ and $y(t)$ are defined as

$$R_{xy}(\tau) \equiv E[x(t)y(t+\tau)]$$

$$R_{yx}(\tau) \equiv E[y(t)x(t+\tau)]$$

Because the processes are stationary, it follows that

$$R_{xy}(\tau) = E[x(t-\tau)y(t)] = R_{yx}(-\tau)$$

$$R_{yx}(\tau) = E[y(t-\tau)x(t)] = R_{xy}(-\tau)$$

But in general R_{xy} and R_{yx} are not the same, so they are not even in τ.

4.2 Cross-correlation

Define cross-correlation coefficient

$$\rho_{xy} = \frac{E[\{x(t)-m_x\}\{y(t+\tau)-m_y)\}}{\sigma_x\sigma_y}$$

$$= \frac{E[x(t)y(t+\tau)]-m_xE[y(t+\tau)]-m_yE[x(t)]+m_xm_y}{\sigma_x\sigma_y} =$$

$$= \frac{R_{xy}(\tau)-m_xm_y}{\sigma_x\sigma_y}$$

The correlation function can be written as

$$R_{xy}(\tau) = \sigma_x\sigma_y\rho_{xy} + m_xm_y$$

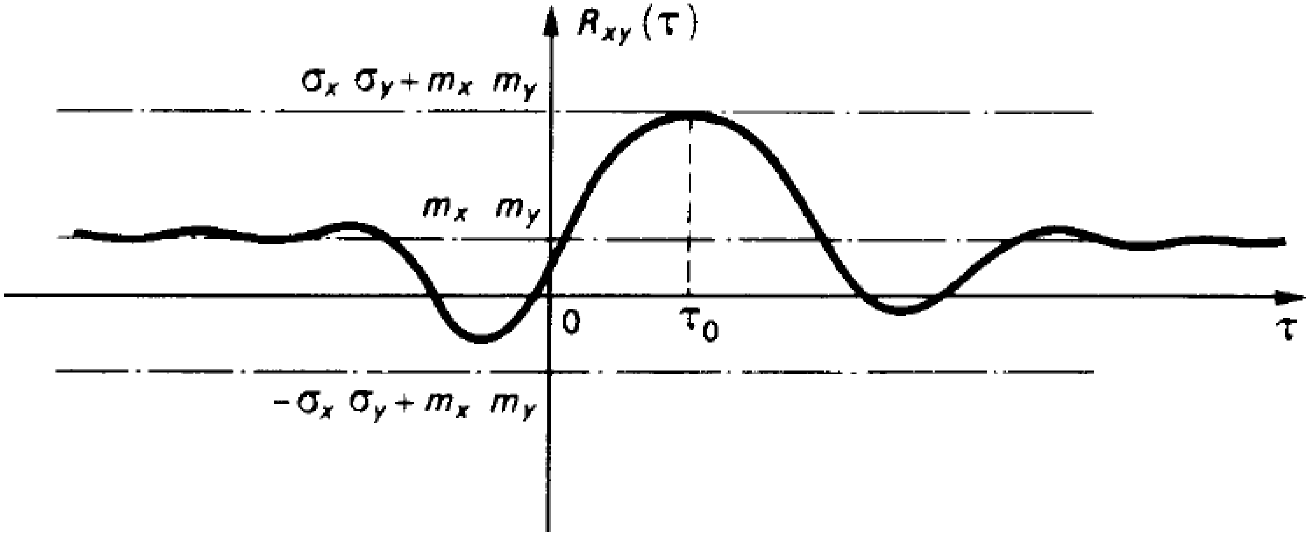
$$R_{yx}(\tau) = \sigma_y\sigma_x\rho_{yx} + m_ym_x$$

As the cross-correlation coefficient is not more than the unit, there are

$$-\sigma_x\sigma_y + m_xm_y \le R_{xy}(\tau),\ R_{yx}(\tau) \le \sigma_x\sigma_y + m_xm_y$$

$$R_{xy}(\tau \to \pm\infty) \to m_xm_y,\ R_{yx}(\tau)(\tau \to \pm\infty) \to m_ym_x$$

4.2 Cross-correlation



$$R_{xy}(\tau) = E[x(t-\tau)y(t)] \neq R_{xy}(-\tau)$$

$$-\sigma_x\sigma_y + m_x m_y \leq R_{xy}(\tau) \leq \sigma_x\sigma_y + m_x m_y$$

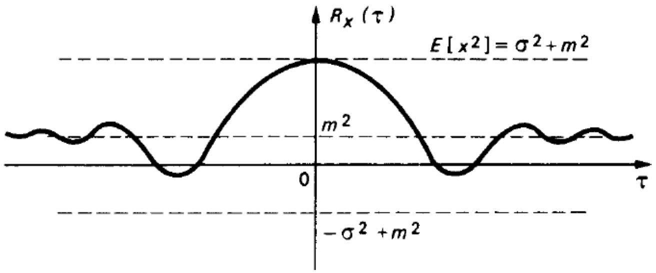
$$R_{xy}(\tau \to \pm\infty) \to m_x m_y$$

4.3 Power Spectral density

- The mean square value of a stationary random process x is given by the area under a graph of spectral density $S_x(\omega)$ against ω.
- Wiener-Khinchin theorem (1930) states that the power spectral density associated with a wide-sense stationary random process is equal to the Fourier transform of the autocorrelation function associated with that process

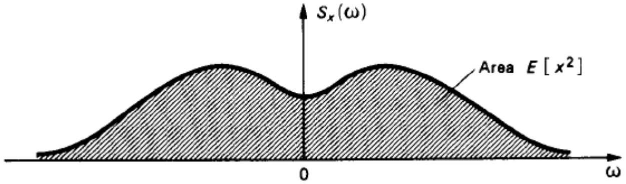
  - **Autocorrelation function**:

$$\boxed{\begin{aligned} R_x(\tau) &\equiv E[x(t)x(t+\tau)] \\ &= \frac{1}{T}\int_0^T x(t)x(t+\tau)\mathrm{d}t \end{aligned}}$$



As R_x is an even function, the spectral density is a real even function:

$$S_x(\omega) = \int_{-\infty}^{\infty} R_x(\tau) e^{-j\omega\tau}\mathrm{d}\tau = \int_{-\infty}^{\infty} R_x(\tau)\cos(\omega\tau)\mathrm{d}\tau$$



$$\boxed{R_x(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_x(\omega) e^{j\omega\tau}\mathrm{d}\omega} \Rightarrow R_x(\tau = 0) = E[x^2] = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_x(\omega)\mathrm{d}\omega = \int_{-\infty}^{\infty} S_x(f)\mathrm{d}f$$

4.3 Cross-spectral density

Define the cross-spectral density as

$$S_{xy}(\omega) = \int_{-\infty}^{\infty} R_{xy}(\tau) e^{-j\omega\tau} \mathrm{d}\tau$$

$$R_{xy}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{xy}(\omega) e^{j\omega\tau} \mathrm{d}\omega$$

$$S_{yx}(\omega) = \int_{-\infty}^{\infty} R_{yx}(\tau) e^{-j\omega\tau} \mathrm{d}\tau$$

$$R_{yx}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{yx}(\omega) e^{j\omega\tau} \mathrm{d}\omega$$

As R_{xy} is an uneven function, the cross-spectral density is a **complex** function.

From the relationship $R_{xy}(\tau) = R_{yx}(-\tau)$

$$\begin{aligned} S_{xy}(\omega) &= \int_{-\infty}^{\infty} R_{xy}(\tau) e^{-j\omega\tau} \mathrm{d}\tau \\ &= \int_{-\infty}^{\infty} R_{yx}(-\tau) e^{-j\omega\tau} \mathrm{d}\tau \\ &= -\int_{\infty}^{-\infty} R_{yx}(\tau) e^{j\omega\tau} \mathrm{d}\tau = \int_{-\infty}^{\infty} R_{yx}(\tau) e^{j\omega\tau} \mathrm{d}\tau \\ &= \int_{-\infty}^{\infty} R_{yx}(\tau) \big(\cos(\omega\tau) + j\sin(\omega\tau)\big) \mathrm{d}\tau \end{aligned}$$

$$\begin{aligned} S_{yx}(\omega) &= \int_{-\infty}^{\infty} R_{yx}(\tau) e^{-j\omega\tau} \mathrm{d}\tau \\ &= \int_{-\infty}^{\infty} R_{yx}(\tau) \big(\cos(\omega\tau) - j\sin(\omega\tau)\big) \mathrm{d}\tau \end{aligned}$$

$$\boxed{S_{xy}(\omega) = S_{yx}^{*}(\omega)}$$

4.3 Units on power spectral density

The definition of PSD sets a radian frequency scale from -∞ to +∞.

- For practical applications, the frequencies expressed in Hz instead of rad/s.
- Only the positive frequencies from 0 to +∞ are represented, called one-side spectrum.
- The units of $S_x(f)$ are mean square / unit of frequency and a more complete name for $S_x(f)$ is the mean square spectral density. For example: a sound signal in Pa, and the units of the spectral density is Pa^2/Hz

$$E[x^2] = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_x(\omega)d\omega$$

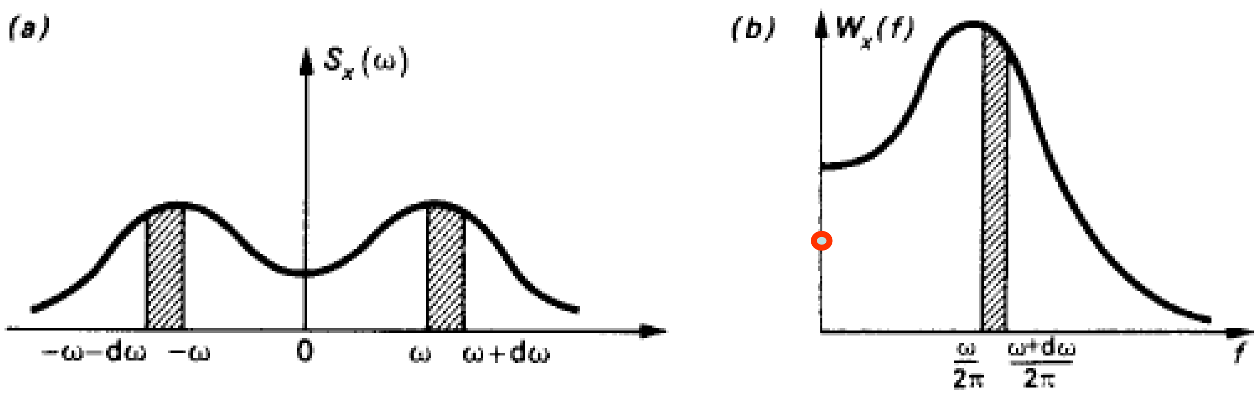
$$\Rightarrow$$

$$E[x^2] = \int_0^{\infty} 2S_x(f)df = \int_0^{\infty} W_x(f)df$$

$$W_x(f) = 2Sx(f), f > 0$$

$$W_x(f) = Sx(f), f = 0$$

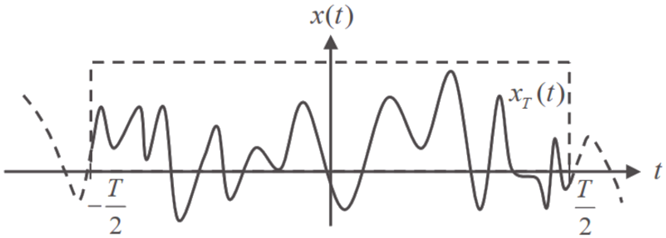
$$W_x(f) = 0, f < 0$$



4.3 Power spectral density Estimation

- The power spectral density function (PSD) states that the average power of the random process (variance) is decomposed in the frequency domain.
- Nonparametric methods: (1) Filter bands, (2) Indirect method (spectral density of autocorrelation function), (3) Direct methods (FFT)

$$x_T(t) = x(t) \quad |t| < T/2$$
$$= 0 \quad \text{otherwise}$$



Averaged power

$$\frac{1}{T}\int_{-T/2}^{T/2} x_T^2(t)dt$$

Parseval's relation of FT

$$\int_{-\infty}^{\infty} x_T^2(t)dt = \int_{-\infty}^{\infty} |X_T(f)|^2 df$$

$$\frac{1}{T}\int_{-T/2}^{T/2} x_T^2(t)dt = \frac{1}{T}\int_{-\infty}^{\infty} |X_T(f)|^2 df = \int_{-\infty}^{\infty} \frac{1}{T} |X_T(f)|^2 df$$

Raw PSD

$$\hat{S}_{xx}(f) = \frac{1}{T} |X_T(f)|^2$$

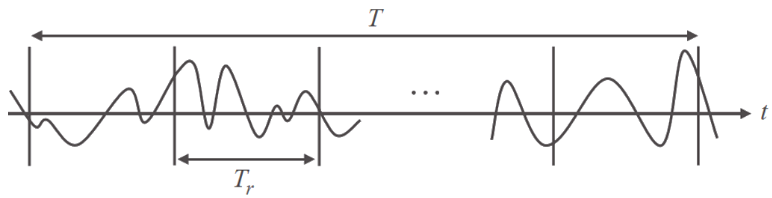
4.3 PSD estimation with segment average

- Note that, increase of T to infinite, raw PSD does not converge.
- Averaging the raw PSD to remove the erratic behavior.

$$E\left[\lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} x_T^2(t)dt\right] = E\left[\int_{-\infty}^{\infty}\lim_{T\to\infty}\frac{|X_T(f)|^2}{T}df\right]$$

$$E(x^2) = \int_{-\infty}^{\infty} S_{xx}(f)df = E[\int_{-\infty}^{\infty}\hat{S}_{xx}(f)df]$$

- Welch’s method



$$\frac{\mathrm{Var}\left(\tilde{S}_{xx}(f)\right)}{S_{xx}^2(f)} \approx \frac{1}{q}$$

$$\frac{\sigma\left(\tilde{S}_{xx}(f)\right)}{S_{xx}(f)} \approx \frac{1}{\sqrt{q}}$$

$$\hat{S}_{xx_i}(f) = \frac{1}{T_r}\left|X_{T_{ri}}(f)\right|^2 \quad \text{for } i = 1, 2, \ldots, q$$

For rectangle window, no overlap between window, giving $1/q = T_r/T$, and the physical resolution $B = 1/T_r$

$$\frac{\mathrm{Var}\left(\tilde{S}_{xx}(f)\right)}{S_{xx}^2(f)} \approx \frac{1}{BT}$$

For particular time section T, the larger B (lower resolution) means shorter section and more number of sections, which leads to small variation of PSD estimation.

To decrease the leakage, choose a smooth window instead of the rectangle window.

4.3 CSD estimation

- Cross-spectral density function CSD: Generalizing the Wiener-Khinchin theorem, CSD is defined as

$$S_{xy}(f) = \int_{-\infty}^{\infty} R_{xy}(\tau) e^{-j2\pi f\tau} d\tau$$

with inverse

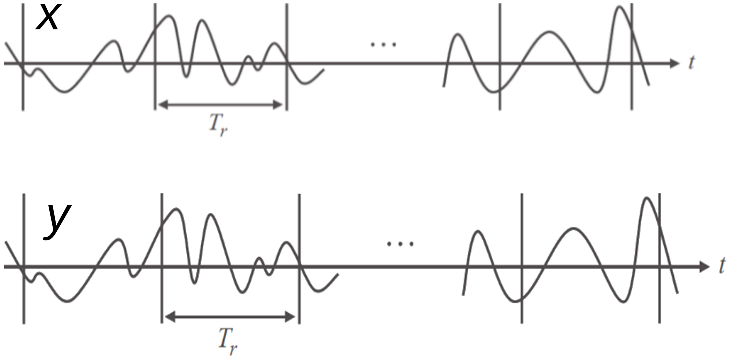
$$R_{xy}(\tau) = \int_{-\infty}^{\infty} S_{xy}(f) e^{j2\pi f\tau} df$$

This can be interpreted as the frequency domain equivalent of the cross-correlation function. That is, $|S_{xy}(f)|$ is the *cross-amplitude spectrum* and it shows whether frequency components in one signal are 'associated' with large or small amplitude at the same frequency in the other signal, i.e. it is the measure of association of amplitude in x and y at frequency f; $\arg S_{xy}(f)$ is the *phase spectrum* and this shows whether frequency components in one signal 'lag' or 'lead' the components at the same frequency in the other signal, i.e. it shows lags/leads (or phase difference) between x and y at frequency f.

- Alternatively, CSD can be defined as the expectation of the multiply of Fourier transforms of truncated functions $x_T(t)$ and $y_T(t)$. Generally, CSD is complex function.

$$S_{xy}(f) = \lim_{T\to\infty} \frac{E[X_T^*(f) Y_T(f)]}{T}$$

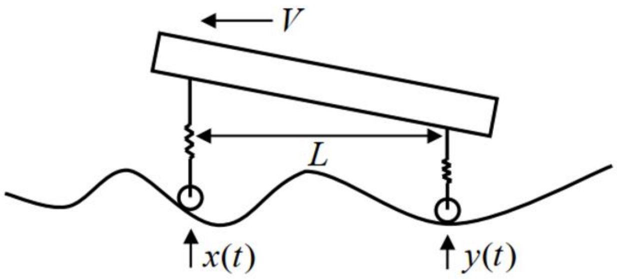
$$S_{xy}(f) = \left|S_{xy}(f)\right| e^{j \arg S_{xy}(f)}$$



- CSD shows whether frequency components in one signal are associated with the other signal in amplitudes, and the phase lag or lead between two signals.

4.3 CSD phase

- An example of the phase relationship in CSD



$y = x(t-\Delta)$, Δ=L/V

$$R_{xy}(\tau) = \frac{1}{T}\int_T x(t)y(t+\tau)dt$$

$$= \frac{1}{T}\int_T x(t)x(t-\Delta+\tau)dt$$

$$= R_{xx}(\tau-\Delta)$$

$$S_{xy}(f) = \int_{-\infty}^{\infty} R_{xx}(\tau-\Delta)e^{-j2\pi f\tau}d\tau = \int_{-\infty}^{\infty} R_{xx}(u)e^{-j2\pi f(u+\Delta)}du$$

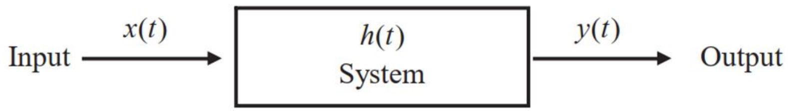
$$= e^{-j2\pi f\Delta}\int_{-\infty}^{\infty} R_{xx}(u)e^{-j2\pi fu}du$$

$$= e^{-j2\pi f\Delta}S_{xx}(f)$$

This shows that the frequency component f in the signal $y(t)$ lags that component in $x(t)$ by phase angle $2\pi f\Delta$. This is obvious from simple considerations: for example, if $x(t) = A\cos(\omega t)$ then $y(t) = A\cos[\omega(t-\Delta)] = A\cos(\omega t - \omega\Delta)$, i.e. the lag angle is $\omega\Delta = 2\pi f\Delta$.

4.4 LTI system response to Random

- Convolution relationship of single input and single output in LTI system



$$y(t) = x(t)*h(t) = \int_{t_0}^{t} h(t-t_1)x(t_1)dt_1$$

- It is more useful to develop relationship between the random inputs and outputs, represented with mean, correlation, and spectral density functions (power and cross).
- Mean of the random input and output

$$\mu_y = E\left[y(t)\right] = E\left[\int_0^{\infty} h(\tau)x(t-\tau)d\tau\right]$$

$$\int_0^{\infty} h(\tau)E\left[x(t-\tau)\right]d\tau = \int_0^{\infty} h(\tau)\mu_x d\tau = \mu_x \int_0^{\infty} h(\tau)d\tau$$

So it follows that

$$\mu_y = \mu_x \int_0^{\infty} h(\tau)d\tau$$

4.4 Auto-correlation relation

- Auto-correlation of the random input and output

$$R_{yy}(\tau) = E[y(t)y(t+\tau)] = E\left[\int_0^\infty\int_0^\infty h(\tau_1)x(t-\tau_1)h(\tau_2)x(t+\tau-\tau_2)d\tau_1 d\tau_2\right]$$

$$= \int_0^\infty\int_0^\infty h(\tau_1)h(\tau_2)E\left[x(t-\tau_1)x(t+\tau-\tau_2)\right]d\tau_1 d\tau_2$$

Thus,

$$R_{yy}(\tau) = \int_0^\infty\int_0^\infty h(\tau_1)h(\tau_2)R_{xx}(\tau+\tau_1-\tau_2)d\tau_1 d\tau_2$$

Taking Fourier transform: only magnitude relationship between the input and output

$$S_{yy}(f) = \int_{-\infty}^\infty R_{yy}(\tau)e^{-j2\pi f\tau}d\tau$$

$$= \int_0^\infty h(\tau_1)e^{j2\pi f\tau_1}d\tau_1 \int_0^\infty h(\tau_2)e^{-j2\pi f\tau_2}d\tau_2 \int_{-\infty}^\infty R_{xx}(\tau+\tau_1-\tau_2)e^{-j2\pi f(\tau+\tau_1-\tau_2)}d\tau$$

Let $\tau+\tau_1-\tau_2 = u$ in the last integral to yield

$$S_{yy}(f) = |H(f)|^2 S_{xx}(f)$$

4.4 Cross-correlation relation

- Cross-correlation of the random input and output

$$R_{xy}(\tau) = E[x(t)y(t+\tau)] = E\left[\int_0^\infty x(t)h(\tau_1)x(t+\tau-\tau_1)d\tau_1\right]$$

$$= \int_0^\infty h(\tau_1)E[x(t)x(t+\tau-\tau_1)]d\tau_1$$

$$R_{xy}(\tau) = \int_0 h(\tau_1)R_{xx}(\tau-\tau_1)d\tau_1$$

Taking the Fourier transform: contain information of phase between the input and output

$$S_{xy}(f) = \int_{-\infty}^{\infty} R_{xy}(\tau)e^{-j2\pi f\tau}d\tau = \int_0^\infty h(\tau_1)e^{-j2\pi f\tau_1}d\tau_1 \int_{-\infty}^{\infty} R_{xx}(\tau-\tau_1)e^{-j2\pi f(\tau-\tau_1)}d\tau$$

thus

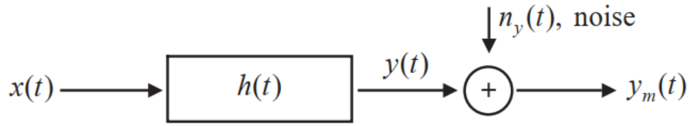
$$S_{xy}(f) = H(f)S_{xx}(f)$$

by forming the ratio $S_{xy}(f)/S_{xx}(f)$ to give $H(f)$

4.4 Noise in system

- Consider the noise in the input and output. For example, the output with measurement noise $y_m(t) = y(t) + n_y(t)$

$$\gamma^2_{xy_m}(f) = \frac{\left|S_{xy_m}(f)\right|^2}{S_{xx}(f)S_{y_m y_m}(f)}$$



Assume $x(t)$ is linearly with $y(t)$, and both are uncorrelated with the noise $n_y(t)$

where $S_{y_m y_m}(f) = S_{yy}(f) + S_{n_y n_y}(f) = |H(f)|^2 S_{xx}(f) + S_{n_y n_y}(f)$. Using the standard input–output relationship, i.e.

$$S_{xy_m}(f) = S_{xy}(f) + S_{xn_y}(f) = S_{xy}(f) = H(f)S_{xx}(f)$$

the coherence function becomes

$$\gamma^2_{xy_m}(f) = \frac{|H(f)|^2 S^2_{xx}(f)}{S_{xx}(f)\left[|H(f)|^2 S_{xx}(f) + S_{n_y n_y}(f)\right]} = \frac{1}{1 + \dfrac{S_{n_y n_y}(f)}{|H(f)|^2 S_{xx}(f)}}$$

$$\boxed{\gamma^2_{xy_m}(f) = \frac{1}{1 + \dfrac{S_{n_y n_y}(f)}{S_{yy}(f)}} = \frac{S_{yy}(f)}{S_{yy}(f) + S_{n_y n_y}(f)} = \frac{S_{yy}(f)}{S_{y_m y_m}(f)}}$$

4.4 Transmission

- Estimate the frequency response function of LTI system $H(f)$ using the random input and output signals. Note: the input and output signals may be noise contaminated.

$$S_{xy}(f)/S_{xx}(f) \neq S_{x_m y_m}(f)/S_{x_m x_m}(f)$$

Consider no input noise and the output noise n_y, use H_1 instead of H_2

$$H_1(f) = \frac{S_{xy_m}(f)}{S_{xx}(f)} = \frac{S_{xy}(f) + S_{xn_y}(f)}{S_{xx}(f)} = \frac{S_{xy}(f)}{S_{xx}(f)} = H(f)$$

$$H_2(f) = \frac{S_{y_m y_m}(f)}{S_{y_m x}(f)} = \frac{S_{yy}(f) + S_{n_y n_y}(f)}{S_{yx}(f)} = H(f)\left(1 + \frac{S_{n_y n_y}(f)}{S_{yy}(f)}\right)$$

Consider both the input and output noise, use

$$H_T(f) = \frac{S_{y_m y_m}(f) - S_{x_m x_m}(f) + \sqrt{\left[S_{x_m x_m}(f) - S_{y_m y_m}(f)\right]^2 + 4\left|S_{x_m y_m}(f)\right|^2}}{2S_{y_m x_m}(f)}$$

4.4 Transmission estimate function in Matlab

- Cross-correlation of the random input and output

$$H(f) = S_{xy}(f)/S_{xx}(f) \qquad R_{xy}(\tau) = E[x(t)y(t+\tau)]$$

Transfer Function

The relationship between the input x and output y is modeled by the linear, time-invariant *transfer function* `txy`. In the frequency domain, $Y(f) = H(f)X(f)$.

- For a single-input/single-output system, the H_1 estimate of the transfer function is given by

$$H_1(f) = \frac{P_{yx}(f)}{P_{xx}(f)},$$

where P_{yx} is the cross power spectral density of x and y, and P_{xx} is the power spectral density of x. This estimate assumes that the noise is not correlated with the system input.

Cross Power Spectral Density

The cross power spectral density is the distribution of power per unit frequency and is defined as

$$P_{xy}(\omega) = \sum_{m=-\infty}^{\infty} R_{xy}(m)e^{-j\omega m}.$$

The cross-correlation sequence is defined as

$$R_{xy}(m) = E\{x_{n+m}y_n^*\} = E\{x_n y_{n-m}^*\},$$

where x_n and y_n are jointly stationary random processes, $-\infty < n < \infty$, $-\infty < n < \infty$, and $E\{\cdot\}$ is the expected value operator.

Algorithms

`cpsd` uses Welch’ s averaged, modified periodogram method of spectral estimation.

4.4 Coherence

- Coherence as a measure of the degree of linear association between the input and output. Defined as

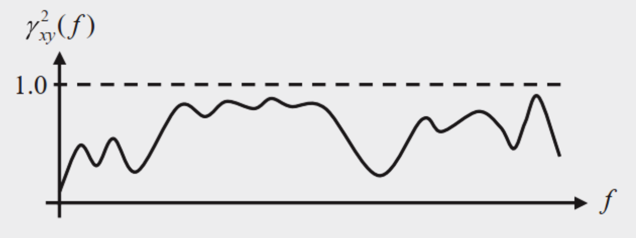
$$\gamma_{xy}^2(f) = \frac{|S_{xy}(f)|^2}{S_{xx}(f)S_{yy}(f)}$$

Linear: $S_{xy}(f) = H(f)S_{xx}(f)$ $S_{yy}(f) = |H(f)|^2 S_{xx}(f)$ $\gamma_{xy}^2(f) = \frac{|H(f)|^2 S_{xx}^2}{S_{xx}(f)\,|H(f)|^2 S_{xx}(f)} = 1$

- The input and output are linearly related. If S_{xy} = 0, then input and output are non-linear totaly. If the coherence function is greater than zero but less than one,

$$0 \le \gamma_{xy}^2(f) \le 1$$

the input and output are partially linearly related. Possible reasons in practice:



1. *Noise* may be present in the measurements of either or both $x(t)$ and $y(t)$.
2. $x(t)$ and $y(t)$ are *not only linearly related* (e.g. they may also be related nonlinearly).
3. $y(t)$ is an output due not only to input $x(t)$ but also to *other inputs*.

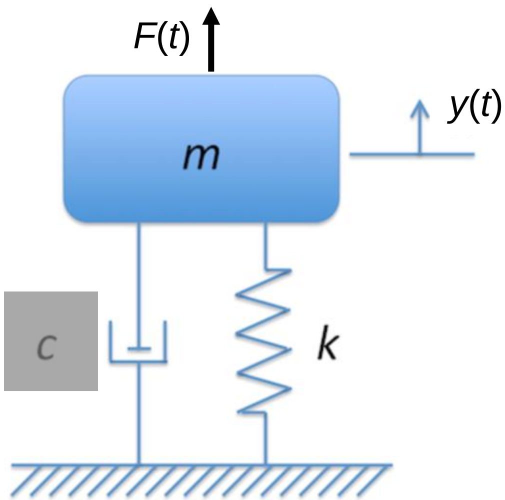
4.5 System modelling and identification

- System identification is a methodology for building mathematical models of dynamic system based on the input and output signals of the system.
  - ✓ In a dynamic system, the values of the output signals depend on both the instantaneous values of the input signals and also on the past behavior of the system.
- System identification is an iterative process: identify model parameters of different model structures based on the input and output experimental data, until find the optimized model which best describes the dynamics of the system, finally validates the identified model with independent inputs and outputs.
  - ✓ A model is a mathematical relationship between the input and output variables of the system.
  - ✓ How to get the model structure: First-principle or Knowledge-driven modelling (White box), Data-driven modelling (Black box), and mixed modelling (grey box).

4.5 Knowledge-driven modelling

- Select the model structure is the most difficult problem in the system identification
- You want a specific structure derived from the first principles.
- If you do not know values of its model parameters. Estimate the values of its parameters from data. This approach is known as grey-box modeling.

Knowledge-driven modelling



Here, *m* is the mass, *k* is the stiffness constant of the spring, and *c* is the damping coefficient. The solution to this differential equation lets you determine the displacement of the mass *y*(*t*), as a function of external force *F*(*t*) at any time *t* for known values of constant *m*, *c*, and *k*. The greyed parameter *C* is required to be estimated from data.

Differential model

$$m\frac{d^2y}{dt^2} + c\frac{dy}{dt} + ky(t) = F(t)$$

State-space model

$$\frac{dx}{dt} = Ax(t) + BF(t)$$

$$y(t) = Cx(t)$$

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & \frac{1}{m} \end{bmatrix}$$

$$C = [1 \quad 0]$$

Discrete-time model

$$y(t) + a_1 y(t - T_s) + a_2 y(t - 2T_s) = bF(t - T_s)$$

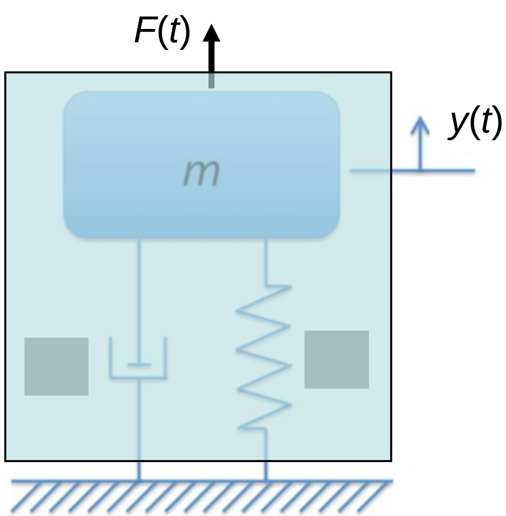
Transfer function model

$$G(s) = \frac{Y(s)}{F(s)} = \frac{1}{(ms^2 + cs + k)}$$

4.5 Data-driven modelling

- You want a model that is able to reproduce your measured data and is as simple as possible. Black-box modeling is useful when your primary interest is fitting the data regardless of a particular mathematical structure of the model.
- The model structures vary in complexity depending on the flexibility to account for the dynamics of the system. Choose one of these structures, ARX, ARMA, K-pole-zero, and compute its parameters to fit the measured response data.
- Black-box modeling is usually a trial-and-error process. Typically, start with the simple linear model structure (ARX) with low order and progress to more complex structures, e.g., ARMX, high order, nonlinear.

Data-driven model



A continuous transfer function is a ratio of polynomials in s-domain

$$G(s) = \frac{(b_0 + b_1 s + b_2 s^2 + \ldots)}{(1 + f_1 s + f_2 s^2 + \ldots)} \longrightarrow G(s) = \frac{Y(s)}{F(s)} = \frac{1}{(ms^2 + cs + k)}$$

A discrete-time transfer function is a ratio of polynomials in z-domain

$$G(z^{-1}) = \frac{bz^{-1}}{(1 + f_1 z^{-1} + f_2 z^{-2})}$$

4.5 Input and output data

- System identification requires the data capture the important dynamics of the system. Good experimental design ensures the measured data have sufficient accuracy and duration to capture the dynamics of the model.
- We assume there are a number of inputs constituting the excitation and a number of outputs constituting the response.
- In general, the experiment must
  - ✓ Excite the system dynamics adequately.
  - ✓ Measure data long enough to capture the important time constants.
  - ✓ Set up a data acquisition system that has a good signal-to-noise ratio.
  - ✓ Measure data at appropriate sampling intervals or frequency resolution.
  - ✓ Check the correlation between the input and output signals.

4.5 System identification of data-driven models

- MA (Moving Average) system is considered as an FIR filter. The coefficients of MA system can be determined by the impulse response sequence.
- ARMA (Auto-Regressive and Moving Average) system is considered as an IIR filter. Using the curve fitting algorithm identify the coefficients of the system.
- Matlab provides *invfreqz* and *invfreqs* for the identification of the discrete-time and continuous-time systems, respectively.

Frequency-Domain Based Modeling

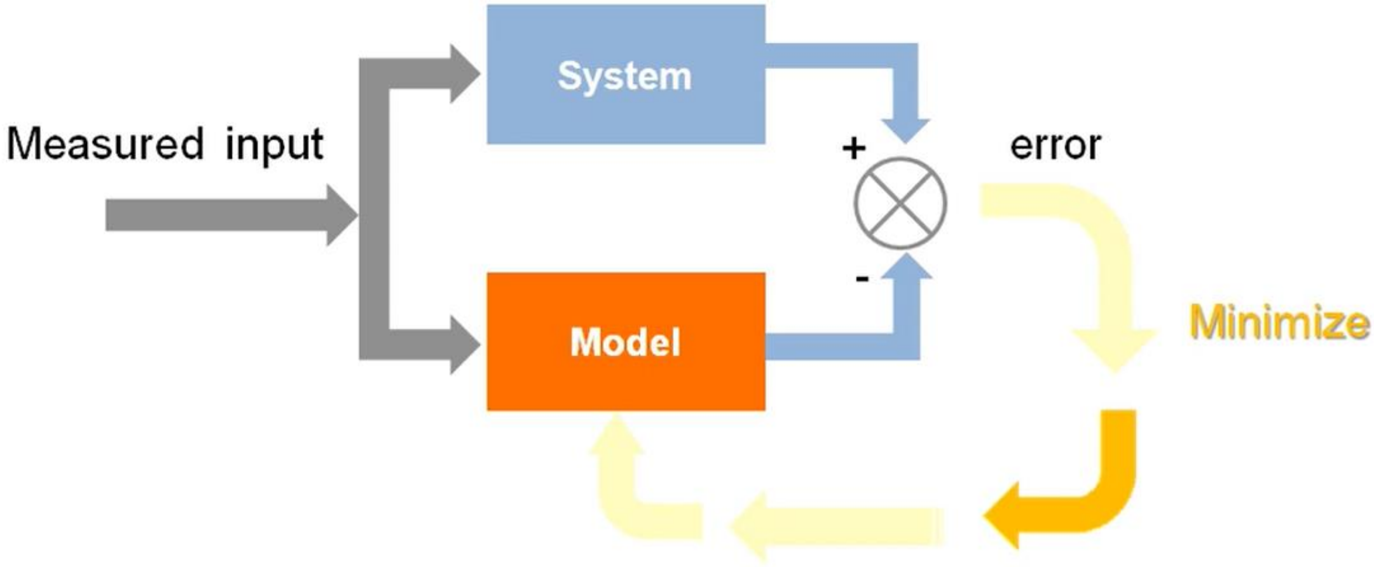
The `invfreqs` and `invfreqz` functions implement the inverse operations of `freqs` and `freqz`; they find an analog or digital transfer function of a specified order that matches a given complex frequency response. Though the following examples demonstrate `invfreqz`, the discussion also applies to `invfreqs`.

To recover the original filter coefficients from the frequency response of a simple digital filter:

```
[b,a] = butter(4,0.4)         % Design Butterworth lowpass
[h,w] = freqz(b,a,64);        % Compute frequency response
[b4,a4] = invfreqz(h,w,4,4)   % Model: n = 4, m = 4
```

4.5 Identification process

- For a particular model structure, the core feature of the iterative process is to estimate the model by tuning the model parameters, make the model output is close to measured output.



4.6 Case study —— Background

Discontinuity detection in long structures should have

- Long detection distance: hundreds of meters
- Large detection area: whole cross-section or a large zone
- High enough detection sensitivity

All these characteristics belong to the guided wave method.

Pipe leakage

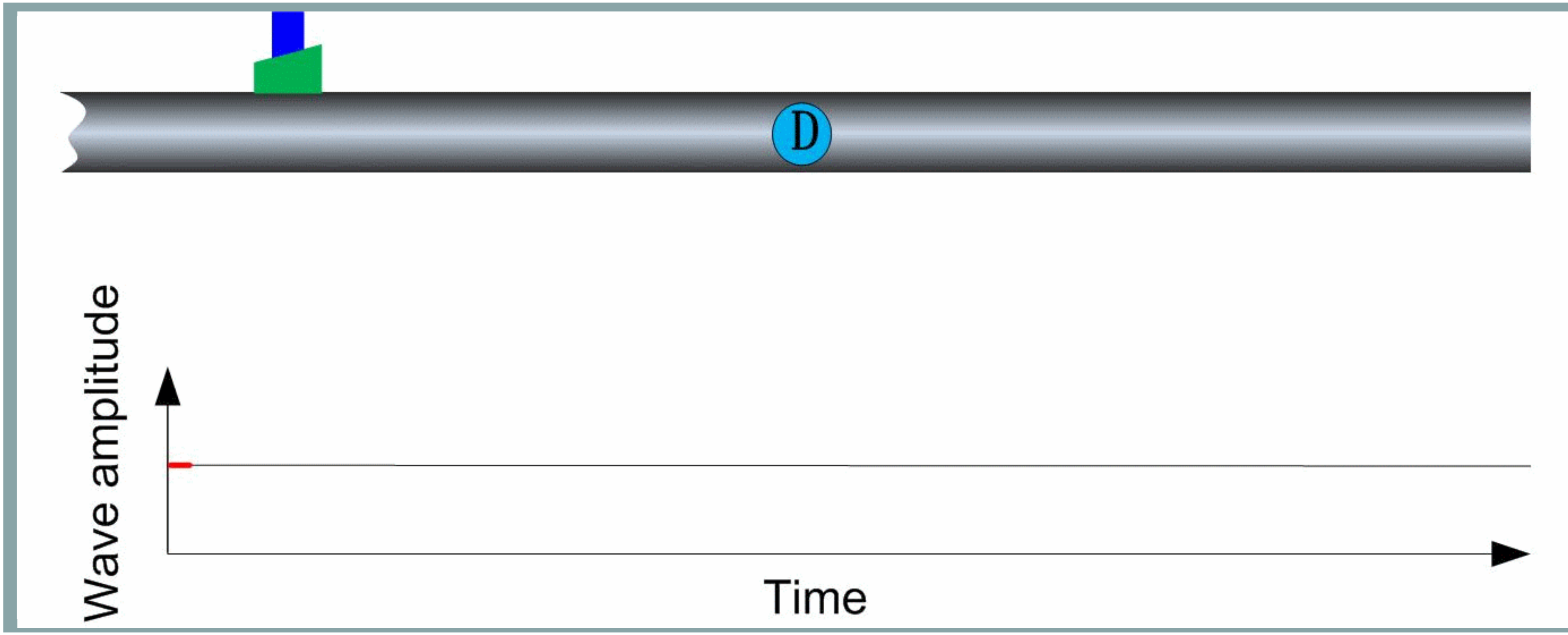
Cable broken

Rail crack

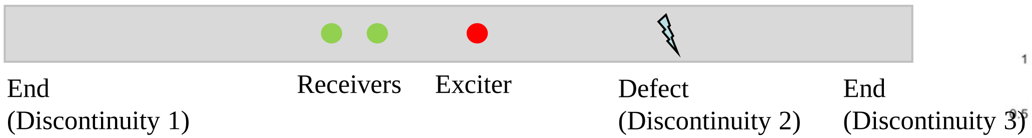
4.6 Working principle

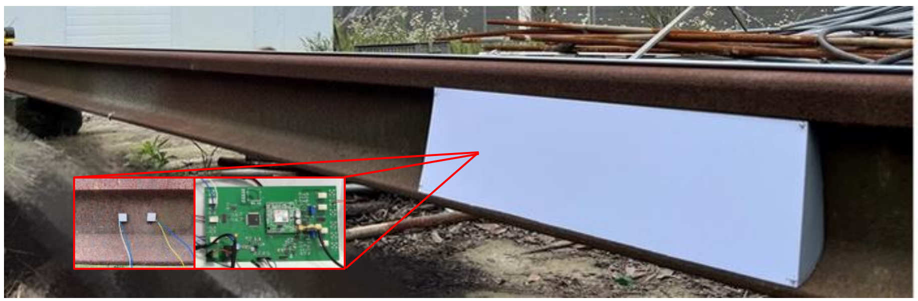
Classic guided wave method

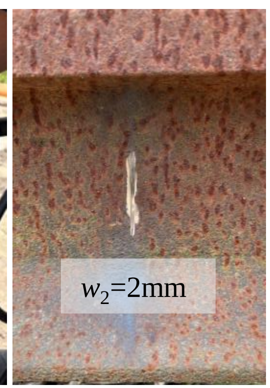
- narrow band pulse with low dispersive property



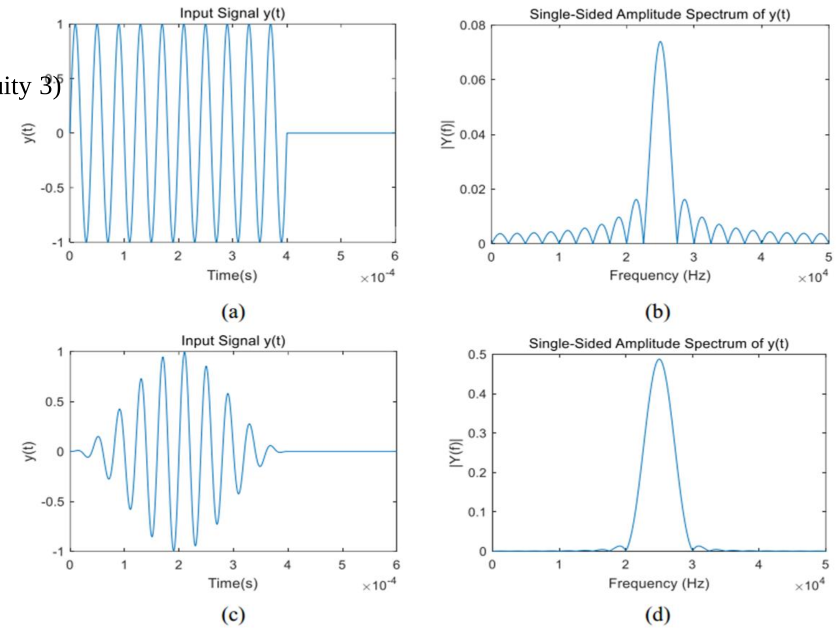
4.6 Experimental setup



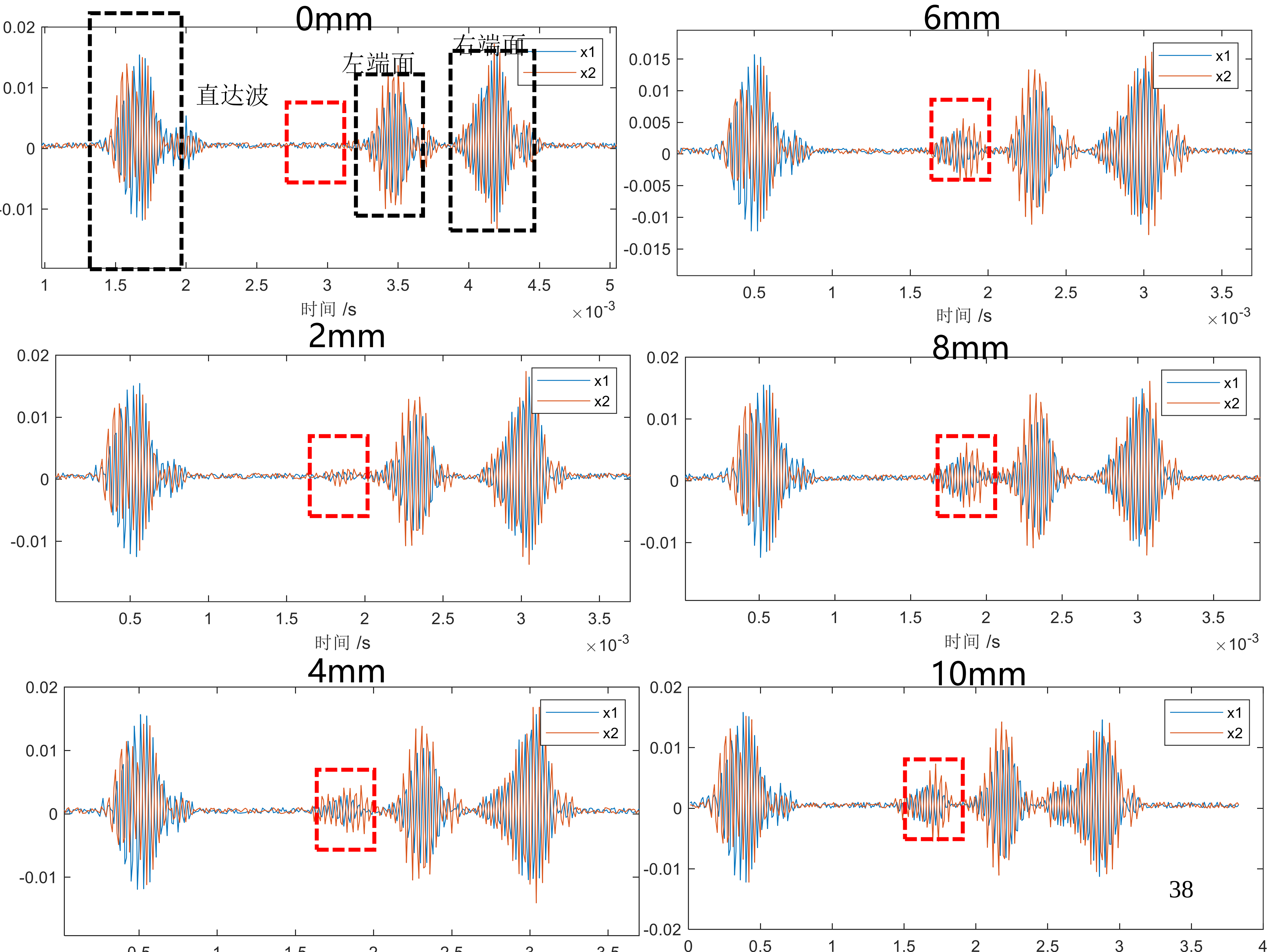




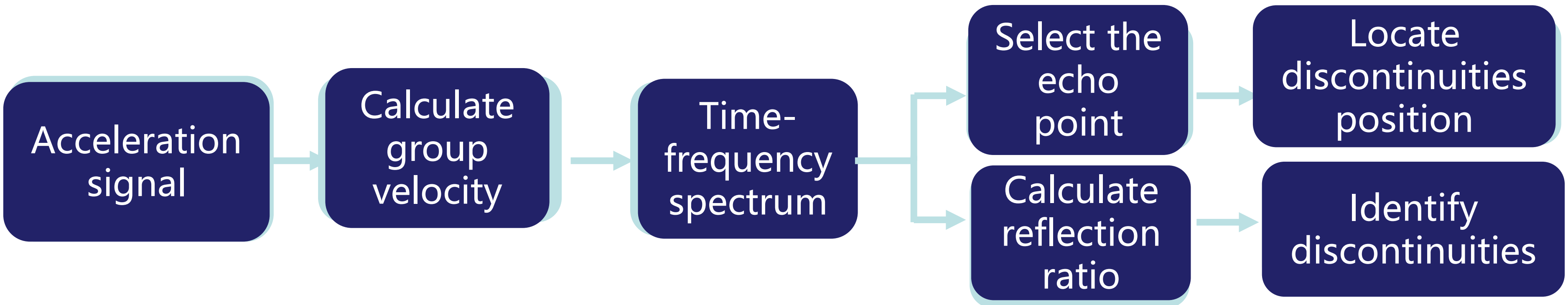
Exciter Signal

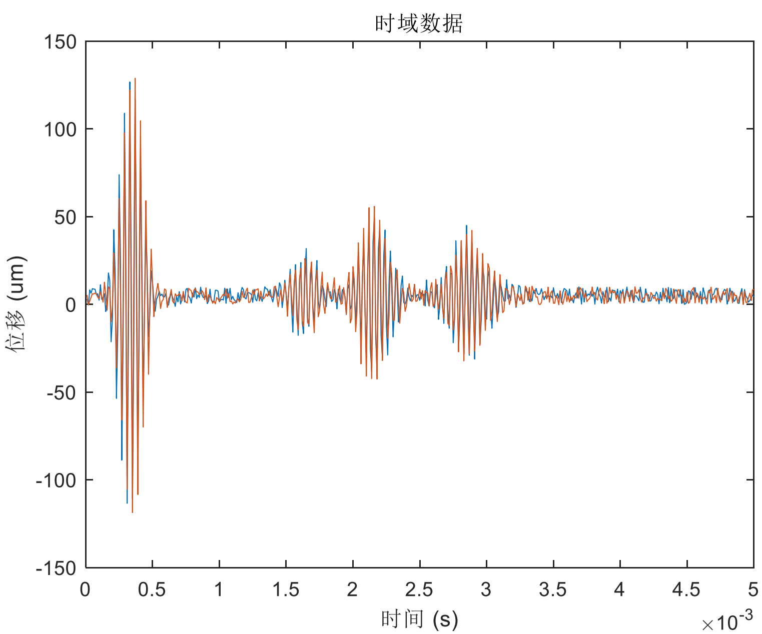


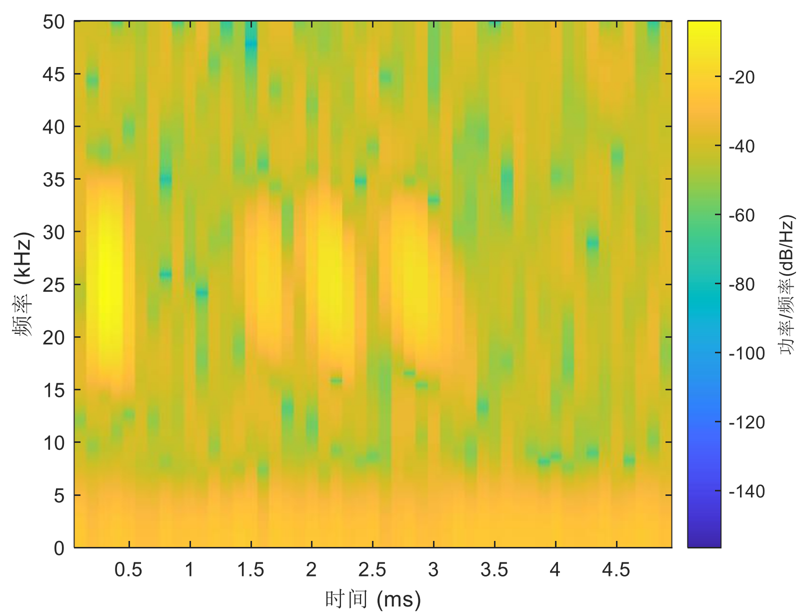
$$s(t) = \begin{cases} \frac{1}{2}\left(1 - \cos\left(\frac{2\pi t}{H}\right)\right)\sin(2\pi f_c t), 0 < t < \frac{N}{f_c} \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad , else \end{cases}$$


0mm
直达波
左端面
右端面
x1
x2
时间 /s
×10-3
6mm
2mm
8mm
4mm
10mm

4.6 Signal processing

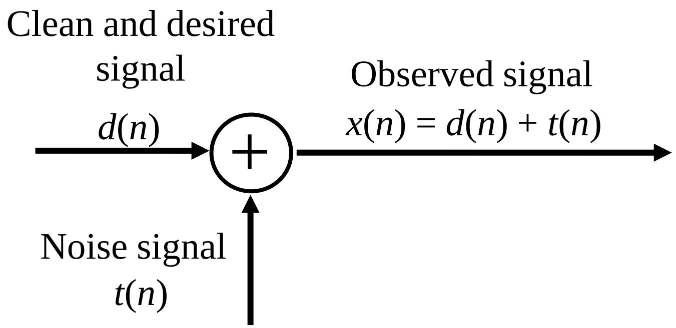




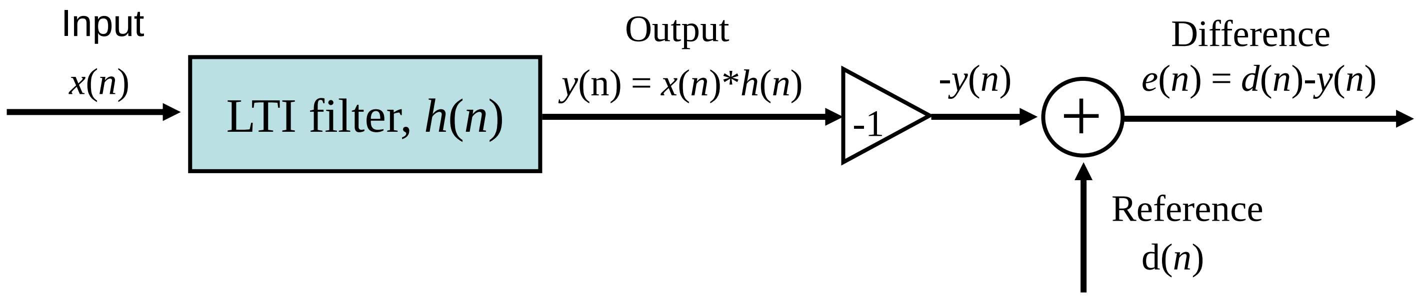


4.7 Wiener filter

- The problem is stated as follows: the stationary zero-mean desired signal $d(n)$ is mixed with the uncorrelated noise signal $t(n)$, and $d(n)$ we can estimate from the observed stationary signal $x(n)$

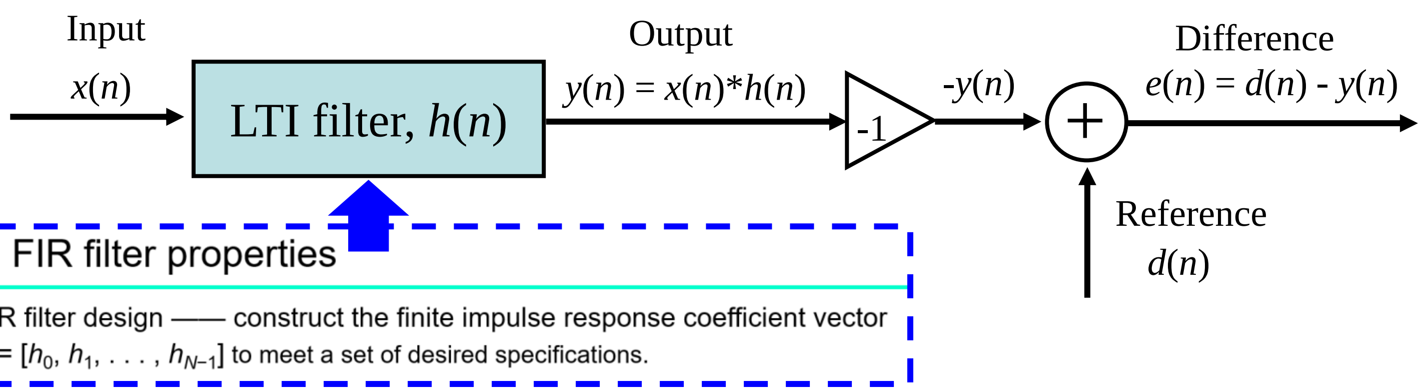


- The goal is to estimate the desired signal by filtering noise from x(n) in the time domain. Norbert Wiener proposed a linear filter design method for this random process, noting Wiener filter. Design $h(n)$ to minimize the error $e(n)$ between the reference and output.



4.7 Wiener filter

- Discrete-time system is easy for the deduction of the Wiener filter. In many cases, the system is continuous, but we only care the output at the discrete time *n*T.
- FIR filter structure (non-recursive) is the simplest form of LTI DT system.



For a causal FIR filter with N-1 order, $y(n) = h(n)*x(n) = \sum_{m=0}^{N-1} h(m)x(n-m)$

- Using the matrix form：

FIR filter vector (N*1)	$\mathbf{h} = [h_0, h_1, \ldots, h_{N-1}]^T$
Input vector at *n*T (N*1),	$\mathbf{Xn} = [x_n, x_{n-1}, \ldots, x_{n-(N-1)}]^T$
Output value at *n*T,	$y_n = \mathbf{h}^T\mathbf{Xn}$

4.7 Wiener filter

- The difference or error function: $e(n) = d(n) - y(n)$

Winner filter adopts the object function of minimizing expectation of the square of error.

i.e., MSE: $E[e^2(n)] = E[d^2(n) - 2d(n)y(n) + y^2(n)]$

Algebra calculations

$y(n) = \mathbf{h}^{\mathrm{T}}\mathbf{Xn}$

$R_{xy}(\tau) \equiv E[x(t)y(t+\tau)]$

$R_x(\tau) \equiv E[x(t)x(t+\tau)]$

$$E[e^2(n)] = \sigma_d^2 - 2\mathbf{h}^{\mathrm{T}}\mathbf{P_n} + \mathbf{h}^{\mathrm{T}}\mathbf{R_n}\mathbf{h},$$

- Cross-correlation vector at nT time between the desired and input signals, (N*1):

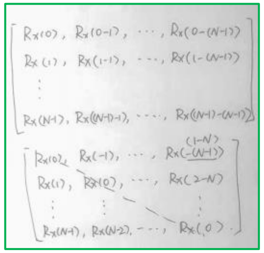
$$\mathbf{P_n} = E[d(n)\mathbf{X_n}] = [E[d(n)x(n)], E[d(n)x(n-1)], \ldots, E[d(n)x(n-(N-1))]]^{\mathrm{T}}$$

$$= [R_{dx}(0), R_{dx}(-1), \ldots, R_{dx}(-(N-1))]^T = [R_{xd}(0), R_{xd}(1), \ldots, R_{xd}(N-1)]^T$$

- Auto-correlation matrix of the input signal at nT, (N*N)

$$\mathbf{R_n} = E[\mathbf{X_n}\mathbf{X_n}^{\mathrm{T}}] = \begin{bmatrix} E[x(n)x(n)] & \cdots & E[x(n)x(n-(N-1))] \\ \vdots & \ddots & \vdots \\ E[x(n-(N-1))x(n)] & \cdots & E[x(n-(N-1))x(n-(N-1))] \end{bmatrix}$$

$$= \begin{bmatrix} R_x(0) & \cdots & R_x(-(N-1)) \\ \vdots & \ddots & \vdots \\ R_x(N-1) & \cdots & R_x(0) \end{bmatrix}$$



4.7 Wiener filter

- $\mathbf{P_n}$ and $\mathbf{R_n}$ are known as the Wiener solution.

$$E[e^2(n)] = \sigma_d^2 - 2\mathbf{h}^T\mathbf{P_n} + \mathbf{h}^T\mathbf{R_n}\mathbf{h},$$

$$\mathbf{P_n} = [R_{xd}(0), R_{xd}(1), \ldots, R_{xd}(N-1)]^T,\ \mathbf{R_n} = E[\mathbf{X_n}\mathbf{X_n}^T] = \begin{bmatrix} R_x(0) & \cdots & R_x(N-1) \\ \vdots & \ddots & \vdots \\ R_x(N-1) & \cdots & R_x(0) \end{bmatrix}$$

- MSE function is actually a **quadratic** function of the filter coefficients, $h_0, h_1, \ldots, h_{N-1}$

$$\varphi(\mathbf{h}) = E[e^2(n)] = \sigma_d^2 - 2\mathbf{h}^T\mathbf{P_n} + \mathbf{h}^T\mathbf{R_n}\mathbf{h}, \text{ where } \mathbf{h} = [h_0, h_1, \ldots, h_{N-1}]^T$$

- The gradient vector at nT, (N*1): $\mathbf{g_n} = \nabla\varphi(\mathbf{h}) = \left[\frac{\partial\varphi}{\partial h_0}, \quad \cdots, \quad \frac{\partial\varphi}{\partial h_{N\_1}}\right]^T$

$$= -2\mathbf{P_n}\frac{\partial \mathbf{h}^T}{\partial \mathrm{h}} + \mathbf{R_n}\frac{\partial \mathbf{h}^T\mathbf{h}}{\partial \mathrm{h}}$$

$$= -2\mathbf{P_n} + 2\mathbf{R_n}\mathbf{h}$$

Matrix calculation

$$\frac{\partial(-2h^TP)}{\partial h} = -2P \cdot \frac{\partial(h^T)}{\partial h} = -2PI = -2P$$

$$\frac{\partial(h^TRh)}{\partial h} = R \cdot \frac{\partial(h^Th)}{\partial h} = 2Rh$$

- $\mathbf{g_n} = 0$ for the minimum of MSE, gives $\mathbf{R_n}\mathbf{h} = \mathbf{P_n}$, Wiener-Hopf equations, or the normal equation, or the Yule-Walker equations.
- $\mathbf{R_n}$ is positive semi-definitive with non-negative eigenvalues. If it is non-singular (has an inverse), the optimized filter $\mathbf{h} = -\mathbf{R_n}^{-1}\mathbf{P_n}$. Further, it is a Toeplitz matrix (constant along the diagonal)

```
>> T=toeplitz(1:6)

T =

     1     2     3     4     5     6
     2     1     2     3     4     5
     3     2     1     2     3     4
     4     3     2     1     2     3
     5     4     3     2     1     2
     6     5     4     3     2     1
```

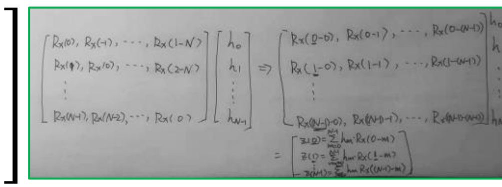
- The Wiener solutions are constant vector and matrix for stationary signals, and **n** can be dropped, giving **P** and **R**.

4.7 Wiener filter transfer function

$$\mathbf{R} = \begin{bmatrix} R_x(0) & \cdots & R_x(-(N-1)) \\ \vdots & \ddots & \vdots \\ R_x(N-1) & \cdots & R_x(0) \end{bmatrix}, \quad \mathbf{P} = [R_{xd}(0), R_{xd}(1), \dots, R_{xd}(N-1)]^T$$

$$\mathbf{Rh} = \begin{bmatrix} R_x(0) & \cdots & R_x(-(N-1)) \\ \vdots & \ddots & \vdots \\ R_x(N-1) & \cdots & R_x(0) \end{bmatrix} \begin{bmatrix} h_0 \\ \vdots \\ h_{N-1} \end{bmatrix} = \begin{bmatrix} \sum_{m=0}^{N-1} h_m R_x(0-m) \\ \vdots \\ \sum_{m=0}^{N-1} h_m R_x((N-1)-m) \end{bmatrix}$$

Assume $z(k) = \sum_{m=0}^{N-1} h_m R_x(k-m) = h(k) * Rx(k)$ $\quad \mathbf{Rh} = \begin{bmatrix} z(0) \\ \vdots \\ z(N-1) \end{bmatrix}$

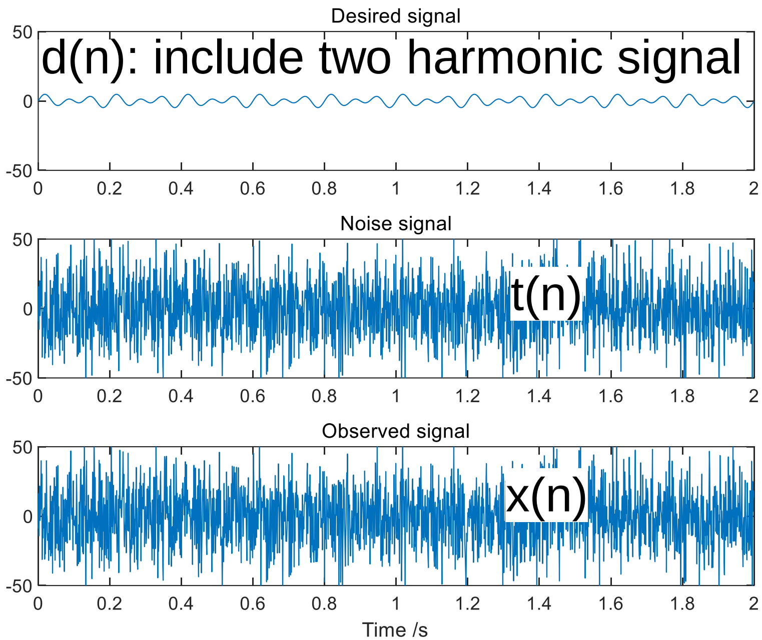


- Wiener-Hopf equation $\mathbf{Rh} = \mathbf{P} \rightarrow \begin{bmatrix} z(0) \\ \vdots \\ z(N-1) \end{bmatrix} = \begin{bmatrix} R_{xd}(0) \\ \vdots \\ R_{xd}(N-1) \end{bmatrix} \rightarrow z(k) = Rxd(k)$

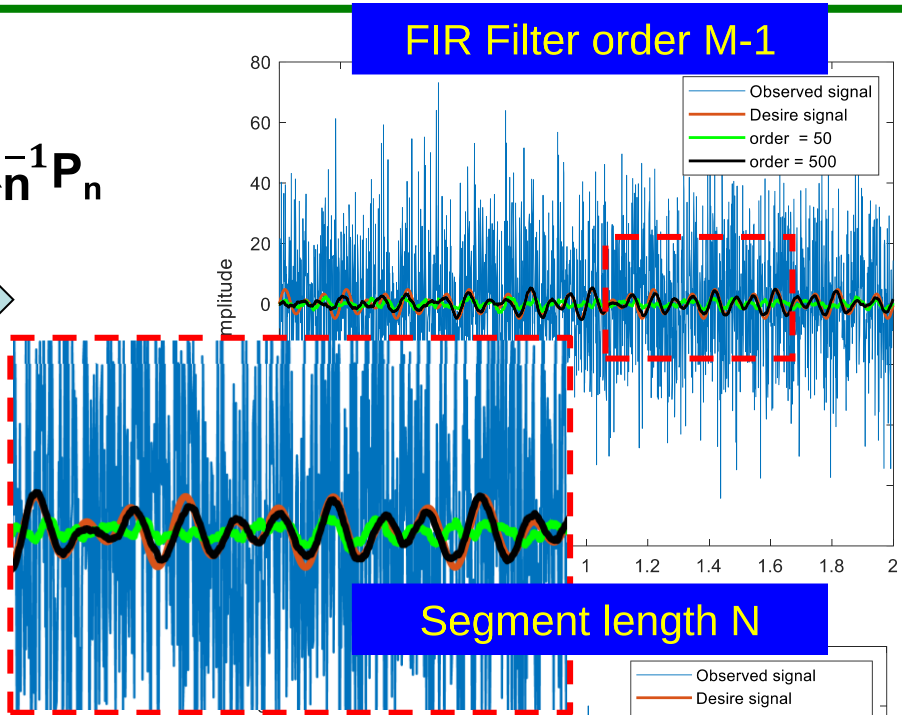
Taking Fourier transform on both sides

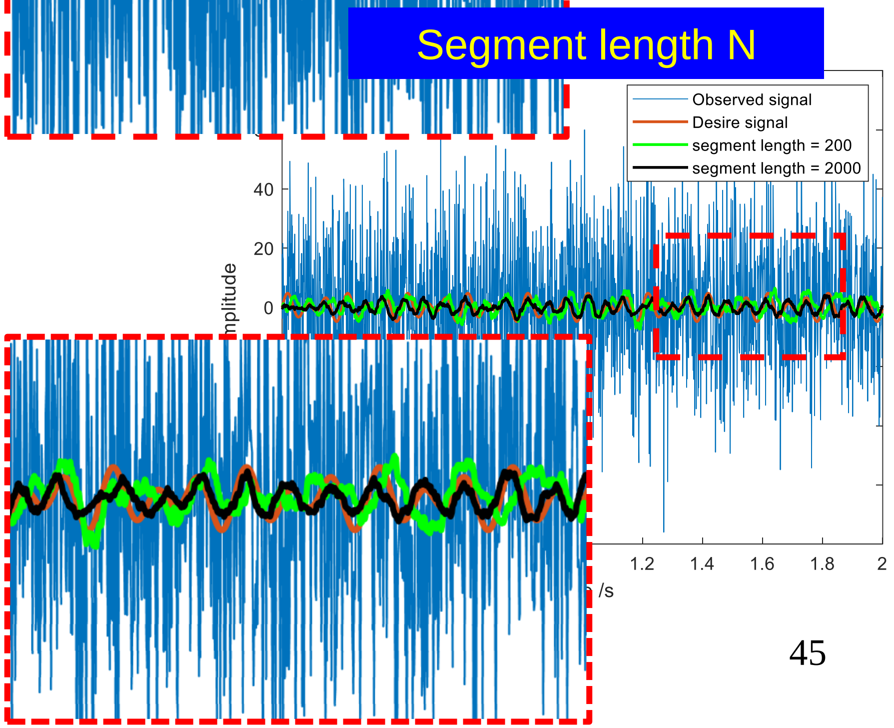
$$H(f)S_x = Sxd \quad \rightarrow \quad H(f) = \frac{S_{xd}}{S_x}$$

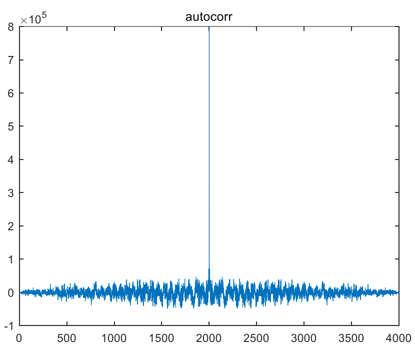
4.7 Wiener filter - Example1



$$\mathbf{h} = -\mathbf{R}_n^{-1}\mathbf{P}_n$$

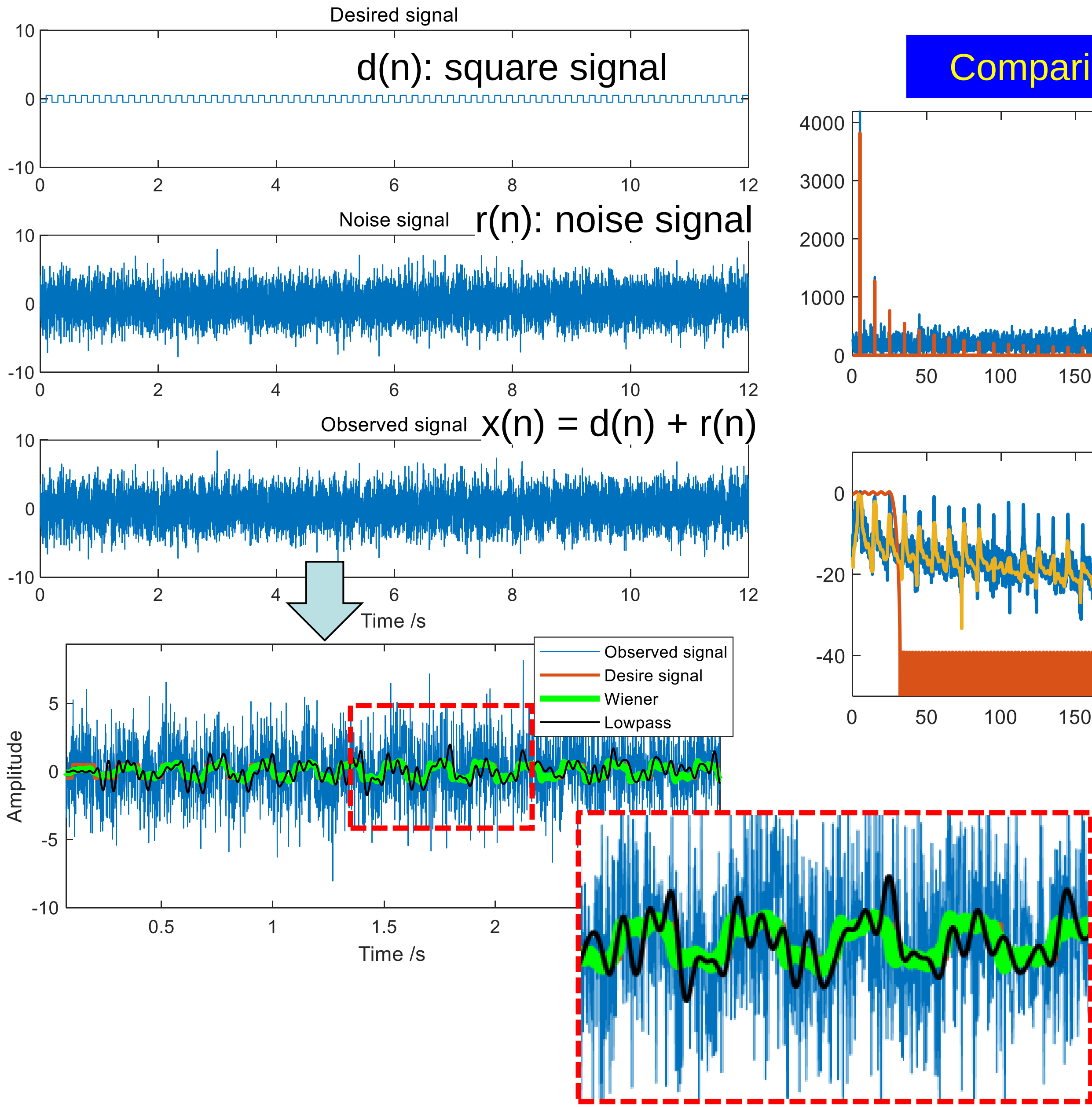




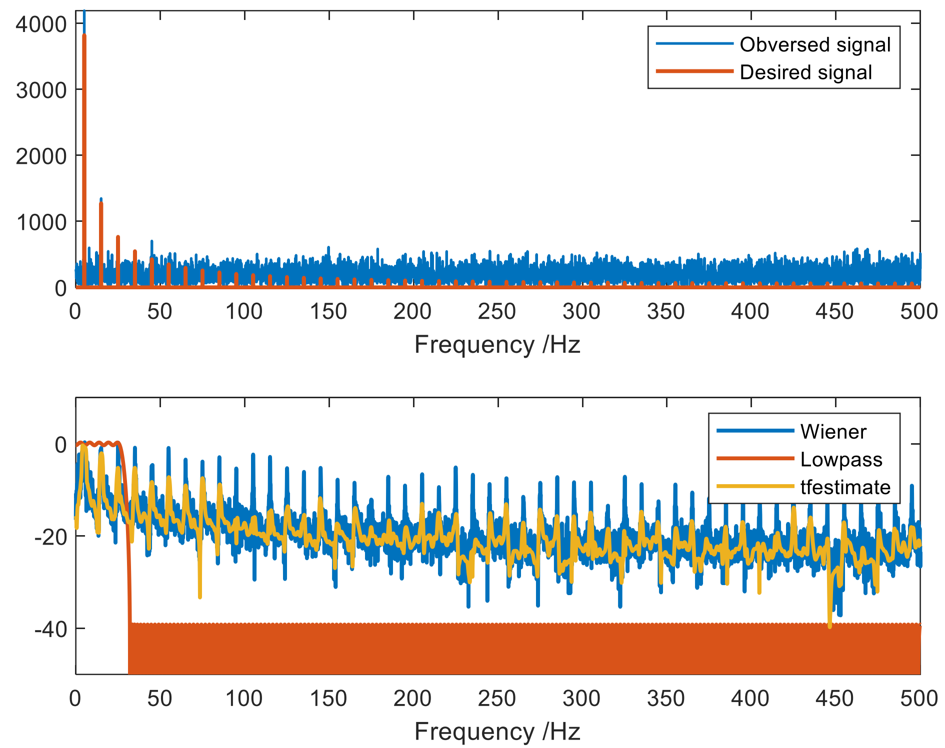



```
%% autocorr matrix
N = 2000;          % segment length
M = 500;        % order
xn_seg = xn(1:N);
dn_seg = dn(1:N);
t_seg = t(1:N);
rx=xcorr(xn_seg);
Rxx=toeplitz(rx(N:N+M))/N;
%% crosscorr matrix
rxs = xcorr(xn_seg,dn_seg);
Rxs = rxs(N:N+M)'/N;
%% H
h = inv(Rxx)*Rxs;
%% filter
yn = filter(h,1,xn);
```

4.7 Wiener filter - Example 2



Comparison in frequency domain



4.7 Adaptive filter

- In many applications, time-variable filters whose characteristics can be varied with time are required. The adaptive filter with coefficients can be adjusted online so as to meet the objective function, e.g., minimize $e(n) = d(n) - y(n)$.
- Wiener filter is a kind adaptive filter with the object function MMSE: minimize $E(e^2)$, and the structure of the adaptive filter can be non-recursive or recursive LTI filter.
- The adaptation algorithms are normally borrowed from some optimization algorithms: Newton algorithm, Quasi-Newton algorithm, Steepest-descent algorithm, see references.