Revision of Lecture 4

- We have discussed all basic components of MODEM
  - Pulse shaping Tx/Rx filter pair
  - Modulator/demodulator
  - Bits $\overset{map}{\leftrightarrow}$ symbols
- Discussions assume ideal channel, and for dispersive channel results no longer valid
- This ISI must be compensated

MODEM components

pulse shaping Tx/Rx filter pair

modulator/demodulator

bits $\overset{map}{\leftrightarrow}$ symbols

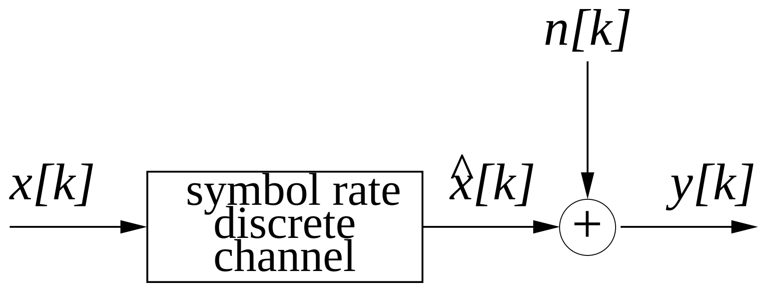
equalisation (distorting channel)

The problem: the combined impulse response of Tx filter, channel and Rx filter will not have desired property of regular zero crossings at symbol spacing

This lecture: **equalisation**

Discrete Channel Model

- Recall slide **15**, examine the combined channel model between $x[k]$ and $\hat{x}[k]$:



- If physical transmission channel is ideal, $y[k]$ is a noise corrupted delayed $x[k]$:

$$y[k] = x[k - k_d] + n[k]$$

- If physical channel is **dispersive** (note ISI):

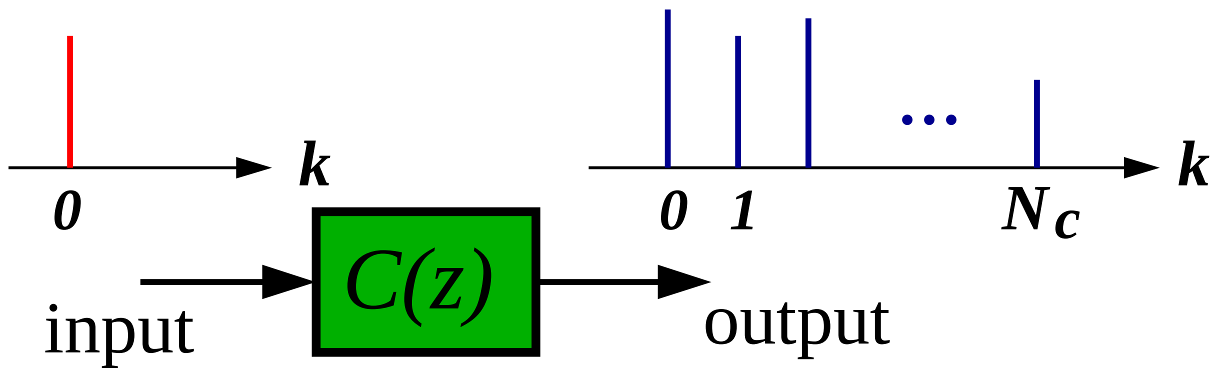
$$y[k] = \sum_{i=0}^{N_c} c_i \cdot x[k - i] + n[k]$$

$\{c_i\}$ are the channel impulse response (CIR) taps, and N_c the length of CIR

Channel Impulse Response

- Continuous-time signal/system $\rightarrow$ Fourier transform
- Discrete-time signal/system $\rightarrow$ z-transform
- Discrete channel $\{c_0, c_1, \cdots, c_{N_c}\}$

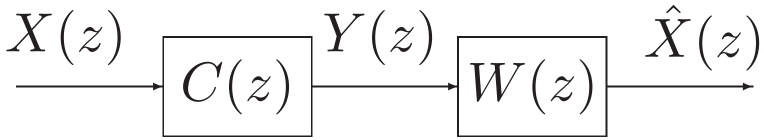
$$C(z) = \sum_{i=0}^{N_c} c_i z^{-i}$$



- We assume real signal/system but results can be extended to complex signal/system (as in QAM system)

Equalisation — I

- If the channel has severe amplitude and phase distortion, equalisation is required:



- The system $C(z)$ is the z-transform of the discrete baseband channel model (including Tx and Rx filters, modulation, physical transmission channel, demodulation, and sampling)

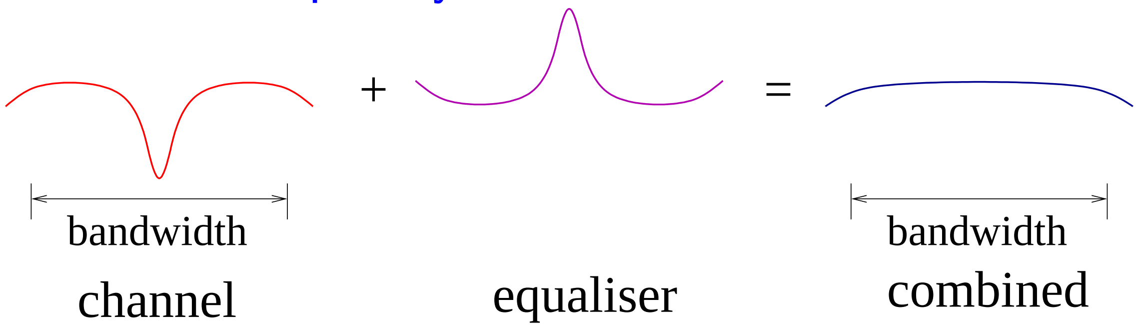
- We want to find an equalisation filter $W(z)$ such that the recovered symbols $\hat{X}(z)$ are only delayed versions of the transmitted signal, $\hat{X}(z) = z^{-k_d} \cdot X(z)$

- The optimal solution for the noise-free case is (zero-forcing equalisation):

$$W(z) \cdot C(z) \; = \; z^{-k_d} \qquad \text{or} \quad W(z) \; = \; z^{-k_d} \cdot C^{-1}(z)$$

Equalisation — II

- Equaliser: make the combined channel/equaliser a Nyquist system again — zero-forcing equalisation will completely remove ISI:



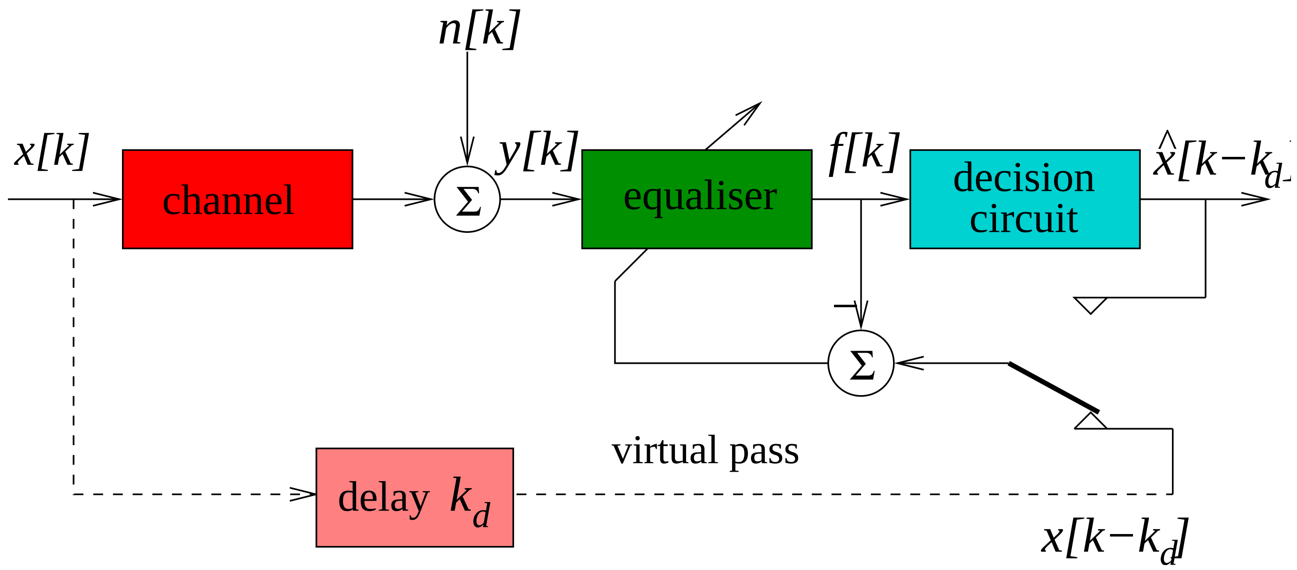
- But the noise is amplified by the equaliser; in a high noisy condition, zero-forcing equalisation may enhance the noise to an unacceptable level $(N(z) \cdot C^{-1}(z))$

- Design of equaliser is a **trade off** between eliminating ISI and not enhancing noise too much

- Also the channel can be time-varying, hence *adaptive* equalisation is needed

Adaptive Equalisation — I

- The generic framework of **adaptive** equalisation:



- *Training mode*: Tx transmits a prefixed sequence known to Rx. The equaliser uses the locally generated symbols $x[k]$ as the desired response to adapt the equaliser

- *Decision-directed mode*: the equaliser assumes the decisions $\hat{x}[k-k_d]$ are correct and uses them to substitute for $x[k-k_d]$ as the desired response

Adaptive Equalisation — II

- Fixed (time-invariant) channel: equalisation is done once during link set up, e.g. digital telephone, during dial up, a prefixed training sequence is sent, and equalisation is performed (before you realized)

- Time-varying channel: equalisation must be performed periodically, e.g. GSM mobile phone, middle of each Tx frame contains 26 training symbols

data	training	data

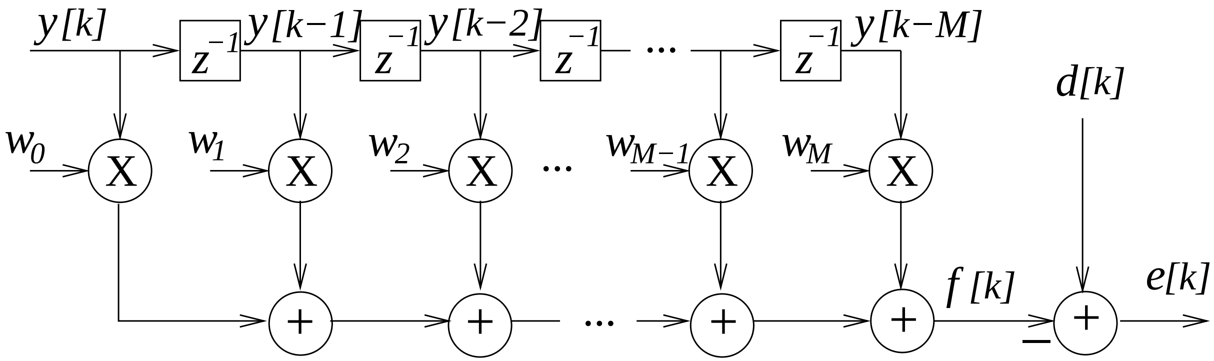
Frame structure

- Blind equalisation: perform equalisation based on Rx signal $\{y[k]\}$ without access to training symbols $\{x[k]\}$, e.g. multipoint network, digital TV, etc

  Note training causes extra bandwidth, thus blind equalisation is attractive but is more difficult

Linear Equaliser

- The setup of a generic linear equaliser with filter coefficients w_i:



- The aim of the equaliser w_i is to produce an output $f[k]$:

$$f[k] = \sum_{i=0}^{M} w_i \cdot y[k-i]$$

as close as possible to the desired signal $d[k]$:

$$d[k] = \begin{cases} x[k-k_d], & \text{training} \\ \hat{x}[k-k_d], & \text{decision directed} \end{cases}$$

Mean Square Error

- The formulation of error signal $e[k]$:

$$e[k] = d[k] - f[k] = d[k] - \sum_{i=0}^{M} w_i \cdot y[k-i] = d[k] - \mathbf{w}^T \cdot \mathbf{y}_k$$

with definitions $\mathbf{w} = [w_0 \; w_1 \cdots w_M]^T$ and $\mathbf{y}_k = [y[k] \; y[k-1] \cdots y[k-M]]^T$

- The **mean square error** formulation:

$$\begin{aligned} \mathcal{E}\{e^2[k]\} &= \mathcal{E}\{(d[k] - \mathbf{w}^T \cdot \mathbf{y}_k)^2\} \\ &= \mathcal{E}\{d^2[k]\} - 2\mathbf{w}^T \cdot \mathcal{E}\{d[k] \cdot \mathbf{y}_k\} + \mathbf{w}^T \cdot \mathcal{E}\{\mathbf{y}_k \cdot \mathbf{y}_k^T\} \cdot \mathbf{w} \\ &= \sigma_d^2 - 2\mathbf{w}^T \cdot \mathbf{p} + \mathbf{w}^T \cdot \mathbf{R} \cdot \mathbf{w} \end{aligned}$$

with the autocorrelation matrix $\mathbf{R} = \mathcal{E}\{\mathbf{y}_k \cdot \mathbf{y}_k^T\}$ and the cross-correlation vector $\mathbf{p} = \mathcal{E}\{d[k] \cdot \mathbf{y}_k\}$; note: as $\mathbf{w}^T \cdot \mathbf{y}_k$ is a scalar, $\mathbf{w}^T \cdot \mathbf{y}_k = (\mathbf{w}^T \cdot \mathbf{y}_k)^T = \mathbf{y}_k^T \cdot \mathbf{w}$

Minimum Mean Square Error

- A standard optimisation procedure to achieve the minimum MSE:

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{E}\{e^2[k]\} = \mathbf{0}$$

- Inserting the previous expression for the MSE:

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{E}\{e^2[k]\} = \mathcal{E}\{-2e[k]\mathbf{y}_k\} = -2\mathbf{p} + 2\mathbf{R} \cdot \mathbf{w} = \mathbf{0}$$

- If $\mathbf{R}$ is invertible, then the optimum filter coefficients $\mathbf{w}_{\text{opt}}$ are given by the above Wiener-Hopf equation as:

$$\mathbf{w}_{\text{opt}} = \mathbf{R}^{-1} \cdot \mathbf{p}$$

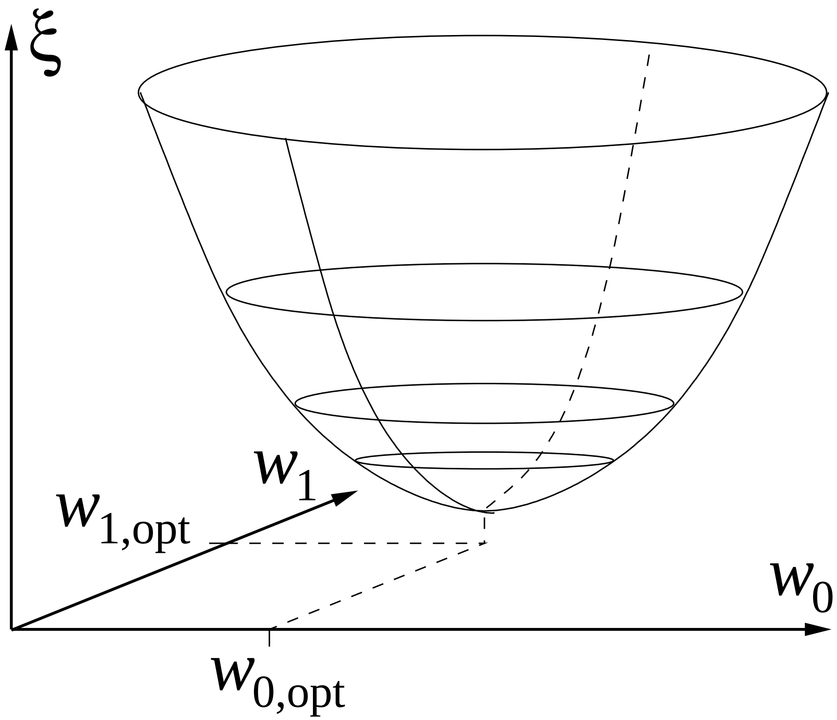
- The **MMSE solution** $\mathbf{w}_{\text{opt}}$ is unique and is also called the **Wiener solution**

- The **minimum MSE** (MMSE) value is $\mathcal{E}\{e^2[k]\}\,|_{\mathbf{w}_{\text{opt}}} = \sigma_d^2 \; - \; \mathbf{p}^T \mathbf{R}^{-1} \mathbf{p}$

Mean Square Error Surface

- Being quadratic in the filter coefficients $\mathbf{w}$, the surface of the MSE $\xi = \mathcal{E}\{e^2[k]\}$ is a hyperparabola in $(M+1)+1$ dimensional space

- Example for $(M+1) = 2$ coefficients:



Autocorrelation Matrix & Cross-correlation Vector

- Given the autocorrelation sequence $r_{yy}[\tau]$ and the cross-correlation sequence $r_{yd}[\tau]$:

$$r_{yy}[\tau] = \sum_{k=-\infty}^{\infty} y[k] \cdot y[k-\tau] \qquad r_{yd}[\tau] = \sum_{k=-\infty}^{\infty} d[k] \cdot y[k-\tau]$$

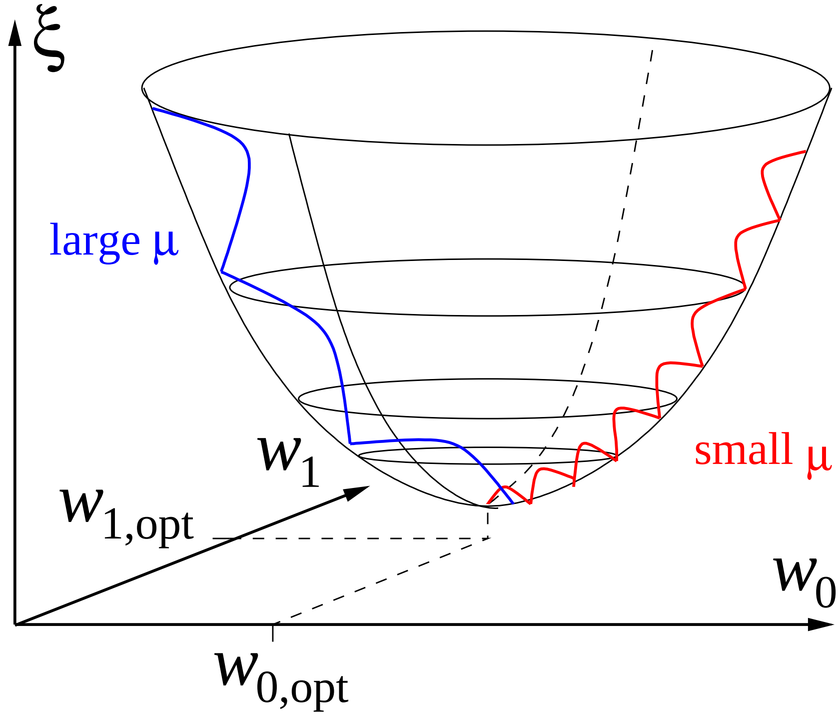
- The autocorrelation matrix $\mathbf{R}$ and cross-correlation vector $\mathbf{p}$ can be written as:

$$\mathbf{R} = \begin{bmatrix} r_{yy}[0] & r_{yy}[1] & \cdot & r_{yy}[M] \\ r_{yy}[1] & r_{yy}[0] & \cdot & r_{yy}[M-1] \\ \vdots & & \ddots & \vdots \\ r_{yy}[M] & r_{yy}[M-1] & \cdots & r_{yy}[0] \end{bmatrix} \qquad \mathbf{p} = \begin{bmatrix} r_{yd}[0] \\ r_{yd}[1] \\ \vdots \\ r_{yd}[M] \end{bmatrix}$$

- The Wiener-Hopf solution requires estimation of $\mathbf{R}$ and $\mathbf{p}$, and a matrix inversion (numerically costly and potentially unstable)

Iterative Solution

- Consider the MSE surface $\longrightarrow$



- The solution can be sought iteratively by moving $\mathbf{w}$ in the direction of the negative gradient:

$$\mathbf{w}_{n+1} = \mathbf{w}_n \; + \; \mu \cdot (-\nabla \xi_n)$$

- μ is the step size; $\nabla \xi_n = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}_n$

- No more inversion of $\mathbf{R}$ required, but statistics still need to be estimated

- Communication systems have fast symbol rates, and channels are often time-varying; adaptation is preferred on a sample-by-sample base, as this can track a time-varying channel better and has simple computational requirements $\longrightarrow$ stochastic gradient approach

Least Mean Square Algorithm

- Rather than using the mean square error, using an instantaneous squared error $e^2[k]$ leads to an instantaneous (stochastic) gradient:

$$\hat{\nabla}\xi_k = \frac{\partial}{\partial \mathbf{w}} e^2[k] = -2e[k] \cdot \mathbf{y}_k$$

- LMS algorithm — during the k-th symbol (sample) period, it does

1. filter output:

$$f[k] = \mathbf{w}_k^T \cdot \mathbf{y}_k = \sum_{i=0}^{M} w_{i,k} \cdot y[k-i]$$

2. estimation error:

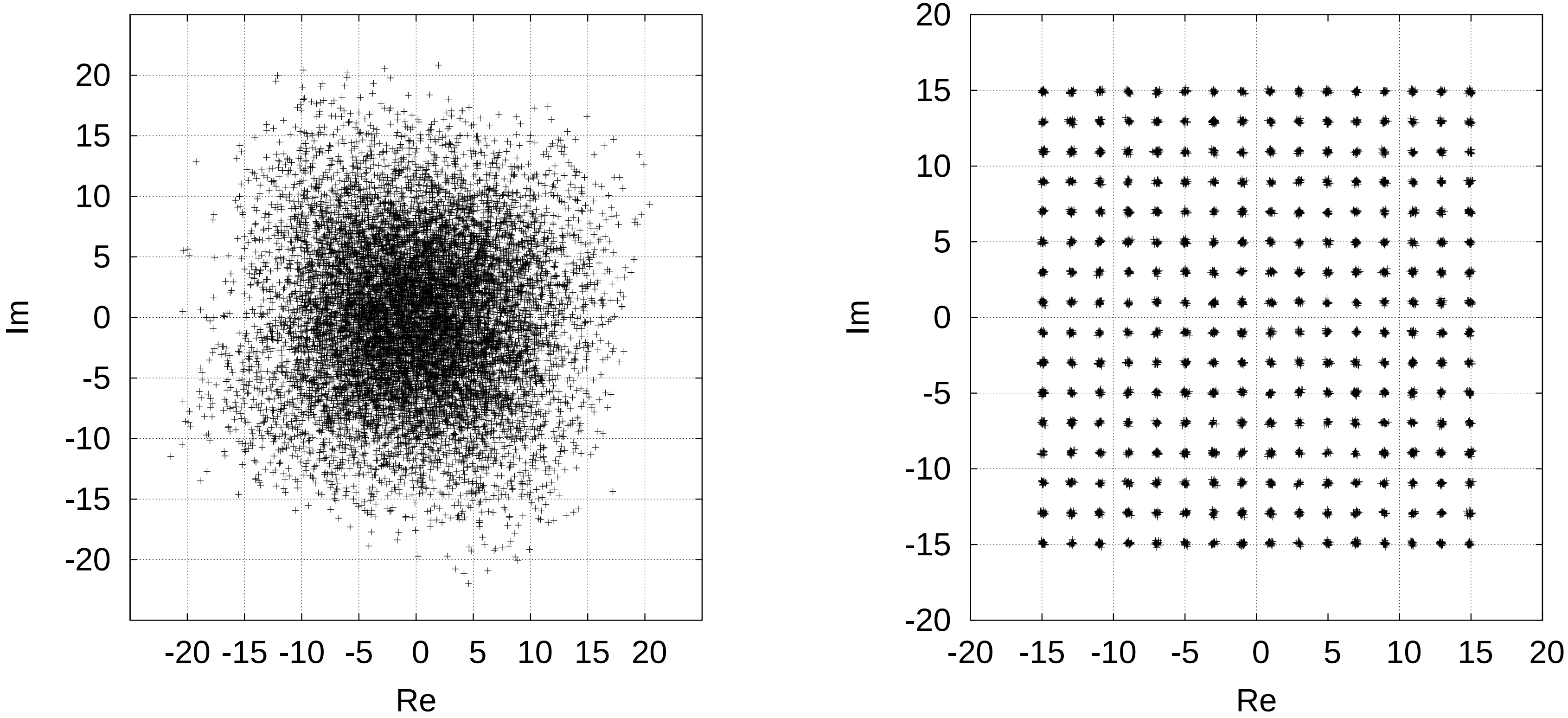
$$e[k] = d[k] - f[k]$$

3. weight adaptation:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu \cdot \mathbf{y}_k \cdot e[k]$$

Blind Equalisation Example

Constellations of a blind equaliser's input $y[k]$ and output $\hat{x}[k]$ (after convergence)



Blind equaliser used is called constant modulus algorithm aided soft decision directed scheme

Optimal Equalisation: Sequence Estimation

- In channel coding part, you'll learn convolutional coding: optimal decoding can be done using Viterbi algorithm

- Channel can be viewed as "convolutional codec", Viterbi algorithm used for "decoding", i.e. estimate transmitted symbol sequence $\{x[k]\}$

- For Example, GSM (QPSK modulation), training symbols used to estimate the channel $\{c_0, c_1, ..., c_6\}$ using for example LMS algorithm

  Viterbi algorithm then used to "decode" Tx symbol sequence $\{x[k]\}$

  Thus in GSM mobile phone hand set there are two Viterbi algorithms, one for channel coding, the other for equalisation

- Sequence estimation is optimal (truly minimum symbol error rate) but may be too complex for high-order modulation schemes and long channels

Summary

- Discrete channel model in the presence of channel amplitude and phase distortion
- Equaliser tries to make the combined channel/equaliser a Nyquist system
- Design of equaliser is a trade off between eliminating ISI and not enhancing noise too much
- Adaptive equalisation structure: training mode and decision directed mode
- Linear equaliser (filter), the MMSE solution, and an iterative algorithm
- Least mean square algorithm
- Equalisation as sequence estimation