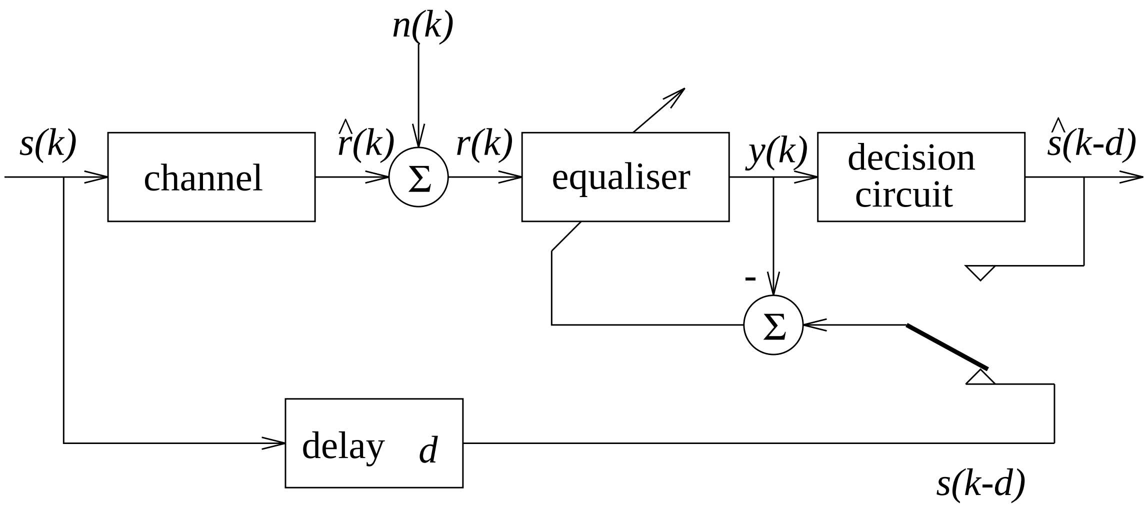
Revision of Lecture Nineteen

- Previous lecture focuses on generic **structure** of adaptive equalisation with two adaptive operation modes, concentrating on class of symbol-decision equalisers, including linear equaliser and decision feedback equaliser



- Classical design based on **minimum mean square error** criterion and novel design based on **minimum bit error rate** criterion have been discussed in details

  Adaptive implementations of these two designs have been developed based on LMS and LBER algorithms, respectively

- This lecture we turn to equalisation based on sequence estimation principle, namely, **maximum likelihood sequence estimation**, and **blind equalisation** techniques

Maximum Likelihood Sequence Estimation

- Recall the digital baseband channel model

$$r(k) = c_0 s(k) + c_1 s(k-1) + \cdots + c_{n_c} s(k - n_c) + n(k)$$

- We can view the **channel** as a "convolutional **encoder**" that convolves the data $\{s(k)\}_{k=1}^{K}$ with a set of channel coefficients $\{c_i\}_{i=0}^{n_c}$
- At the receiver, we try to recover the transmitted data sequence $\{s(k)\}_{k=1}^{K}$, i.e. to provide an estimated data sequence $\{\hat{s}(k)\}_{k=1}^{K}$
- The same **MLSE principle**, as in convolutional decoding, can be applied
- Formally this is formulated as the optimisation: given the received samples $\{r(k)\}_{k=1}^{K}$ find a sequence $\{\hat{s}(k)\}_{k=1}^{K}$ that minimises:

$$\mathcal{M} = \sum_{k=1}^{K} \left| r(k) - \sum_{i=0}^{n_c} c_i \hat{s}(k-i) \right|^2$$

  and the **Viterbi algorithm** is actually used to do it
- The MLSE is the (near) true optimal solution for equalisation in terms of symbol error rate, assuming $n(k)$ is an AWGN

  But it becomes computationally prohibitive for long channel length n_c and large symbol size N

Channel as an Encoder

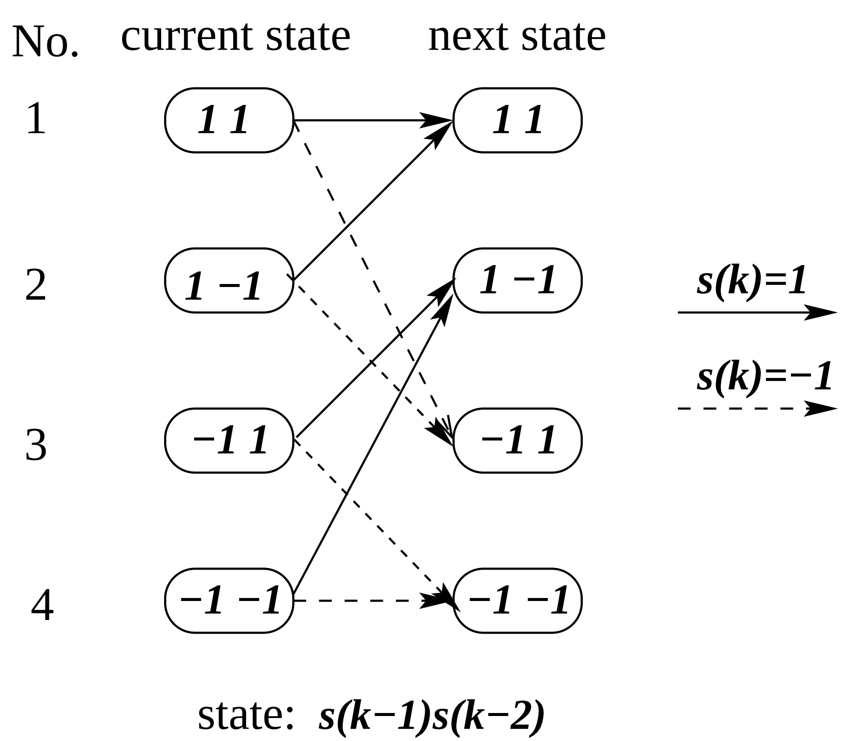
- Example: $r(k) = \bar{r}(k) + n(k) = c_0 s(k) + c_1 s(k-1) + c_2 s(k-2)$ with BPSK, i.e. $s(k) \in \{\pm 1\}$

  "**State transition**" diagram:

  State of encoder: $(s(k-1)\ s(k-2))$

  Output $\bar{r}(k)$ depends on the state and the "input" $s(k)$

  Similar to a convolutional encoder, and number of states: 2^{n_c}

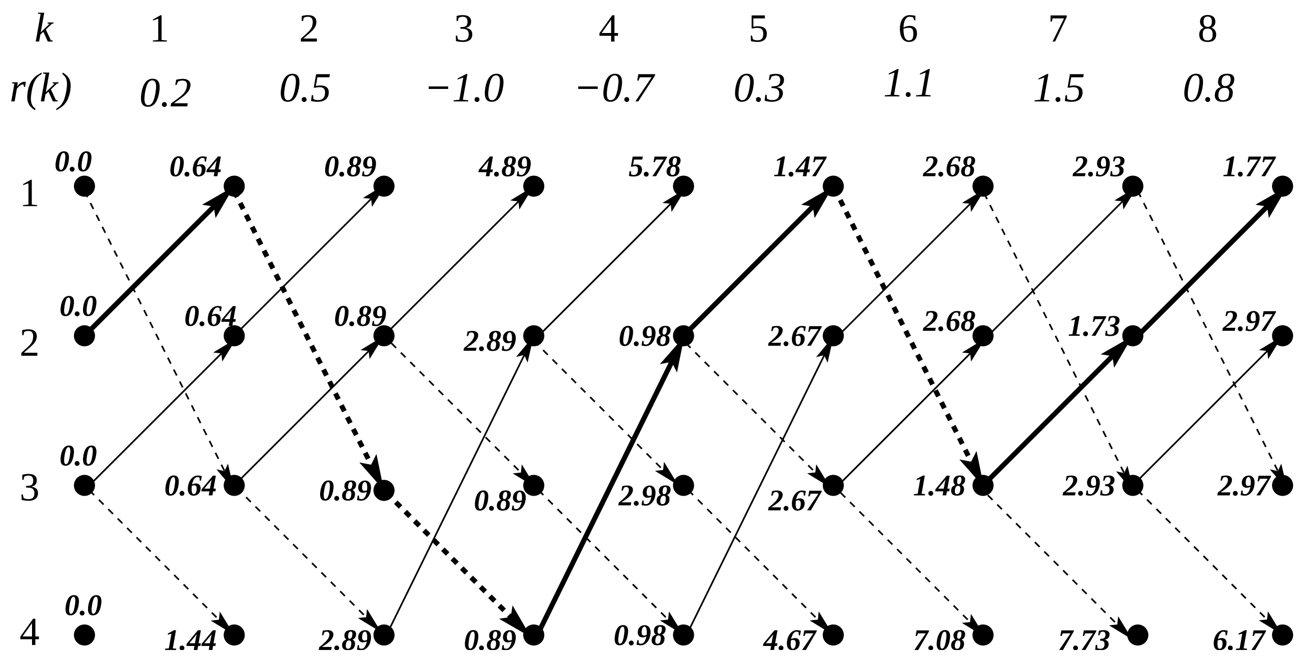


- **Viterbi algorithm** can be used for "decoding", as in a convolutional decoder, and all the Viterbi algorithm rules apply, with the **branch metric** defined as

$$\left(r(k) - \sum_{i=0}^{n_c} c_i s(k-i) \right)^2$$

MLSE with Viterbi algorithm

- Previous example with $c_0 = c_1 = c_2 = 1$, and the received samples $r(1), \cdots, r(8) = 0.2, 0.5, -1.0, -0.7, 0.3, 1.1, 1.5, 0.8$

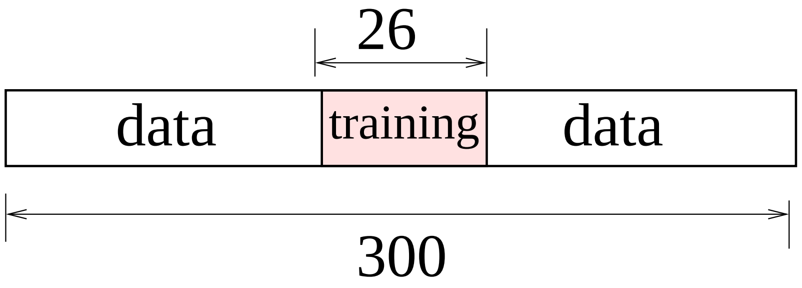


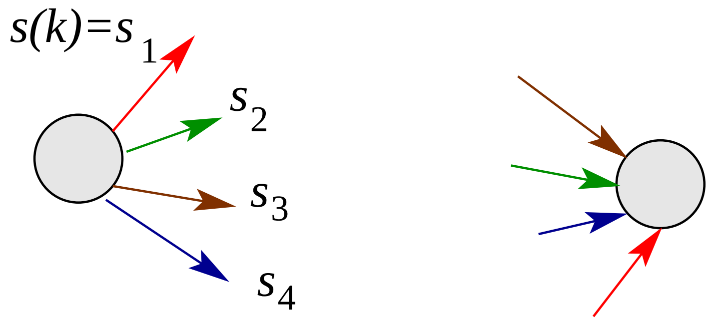
The detected data $s(1), \cdots, s(8) = 1, -1, -1, 1, 1, -1, 1, 1$

- The sequence length K should be sufficiently long ($> 5n_c$), and for adaptive implementation, use the LMS/RLS to identify the channel $\{\widehat{c}_i\}_{i=0}^{n_c}$

GSM Example

- GSM: frame contains approximately 300 data samples with middle 26 as training symbols
- Modulation is 2 bits per symbol: $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$, and worst case channel has 6 taps, i.e. $n_c = 5$





- Thus, number of states is $4^5 = 1024$, and each state has 4 outgoing branches, and 4 incoming branches
- 26 specifically designed training symbols (correlation matrix is diagonal) in each frame are used to estimate channel taps using LS estimate
- Viterbi algorithm is then used to detect the data symbols of each frame
- Digital signal processor is sufficiently powerful to implement Viterbi algorithm for this 4-QAM scheme with 1024 states
- For longer channel length and/or higher-order QAM scheme, computational complexity of Viterbi algorithm become excessively high

Blind Equalisation

- In blind equalisation, there is no training, an equaliser has to estimate the transmitted symbols and/or channel based only on the received samples $r(k)$

- There are three classes of blind equalisation algorithms
  - **Joint data and channel estimation**: e.g. using blind or super trellis search techniques. This produces the best results but can be computationally prohibitive
  - **Higher-order statistics based methods**: to identify the channel using $r(k)$ only, 2nd order statistic is insufficient as it is phase blind. Higher-order statistics based methods can overcome this problem. This approach produces very good results but computational cost can be very expensive
  - **Bussgang-type adaptive FIR filters**: optimise some non-MSE type cost functions using stochastic gradient, computationally very simple

- We will discuss the 3rd class. Since there is no desired response $s(k-d)$ for the adaptive filter, one has to "invent" some substitute $\rightarrow$ the resulting non-MSE cost functions generally have local minima, and this often causes problems

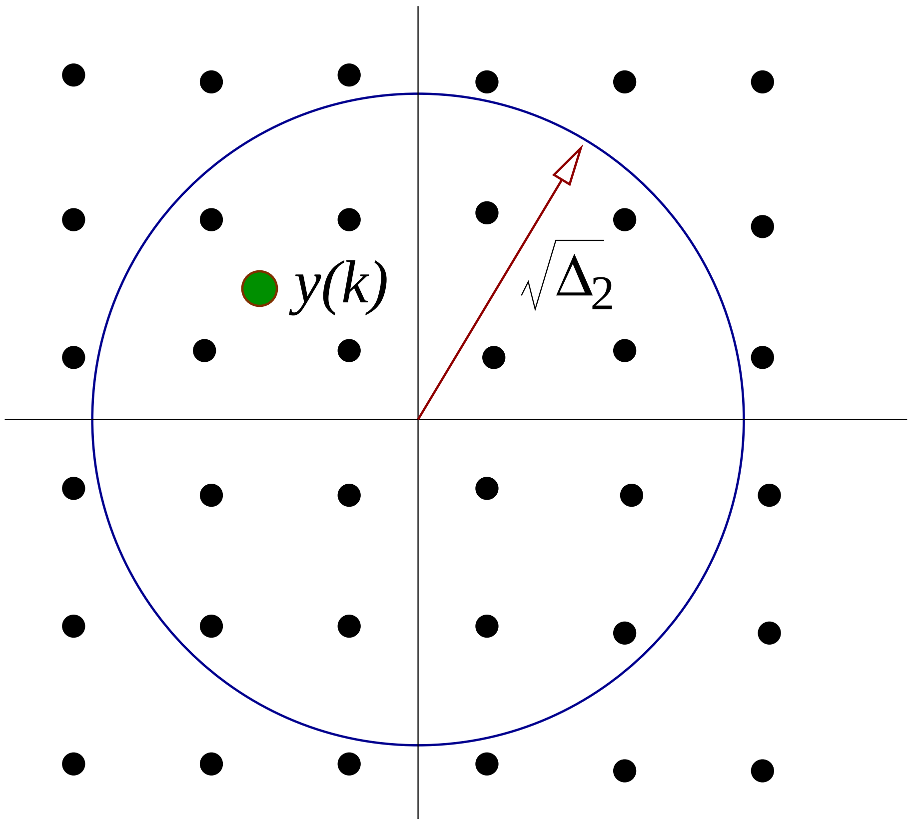
Constant Modulus Algorithm

- Consider generic QAM case with notations:
  - channel taps $c_i = c_{R,i} + \mathrm{j}c_{I,i}$
  - received signals $r(k) = r_R(k) + \mathrm{j}r_I(k)$
  - QAM symbols $s(k) = s_R(k) + \mathrm{j}s_I(k)$
  - equaliser weights $w_i = w_{R,i} + \mathrm{j}w_{I,i}$
- Define constant $\Delta_2 = E[|s(k)|^4]/E[|s(k)|^2]$, and consider adaptive filter or blind equaliser:

$$y(k) = \mathbf{w}^H \mathbf{r}(k)$$

with $\mathbf{w} = [w_0 \; w_1 \cdots w_M]^T$ and $\mathbf{r}(k) = [r(k) \; r(k-1) \cdots r(k-M)]^T$

- Note that QAM symbols do not fall on the constant modulus circle of radius $\sqrt{\Delta_2}$



- However, by penalising equaliser output $y(k)$ which deviates from this circle, the correct symbol constellation can be restored
  - This idea of CMA in fact exploits higher-order statistics of Δ_2
- The CMA, which is the most popular blind equaliser for high-order QAM signalling, has very simple computational requirements similar to those of the LMS

CMA (continue)

- The CMA can be viewed to adjust $\mathbf{w}$ by minimising the non-convex cost function

$$\bar{J}_{\mathrm{CMA}}(\mathbf{w}) = E[(|y(k)|^2 - \Delta_2)^2]$$

using a stochastic gradient method, i.e. actually through minimising $(|y(k)|^2 - \Delta_2)^2$

- At sample k, given $y(k) = \mathbf{w}^H(k)\mathbf{r}(k)$, the equaliser weights are updated using:

$$\left.\begin{array}{l} \epsilon(k) = y(k)(\Delta_2 - |y(k)|^2) \\ \mathbf{w}(k+1) = \mathbf{w}(k) + \mu\epsilon^*(k)\mathbf{r}(k) \end{array}\right\}$$

where μ is a very small positive adaptive gain and $\epsilon^*(k)$ is the conjugate of $\epsilon(k)$

- Compare this with the LMS, where $\epsilon(k) = s(k-d) - y(k)$
- There are many solutions $\mathbf{w}_s$ that minimise the cost function $\bar{J}_{\mathrm{CMA}}(\mathbf{w})$. One of them, $\mathbf{w}_{\mathrm{opt}}$, restores the correct signal constellation and is corresponding to the MMSE solution
- The weight vectors that minimise $\bar{J}_{\mathrm{CMA}}(\mathbf{w})$ are thus

$$\mathbf{w}_s = \exp(j\phi)\mathbf{w}_{\mathrm{opt}}, \ \ 0 \le \phi < 2\pi$$

- This undesired phase shift cannot be resolved by the CMA (all blind equalisers suffer more or less a similar problem), and must be eliminated by other means, e.g. using differential encoding

Complex Variable Derivative

- Complex-valued variable derivative is defined as

$$\frac{\partial J(\mathbf{w})}{\partial w_i} = \frac{1}{2}\left(\frac{\partial J}{\partial w_{R,i}} + \mathrm{j}\frac{\partial J}{\partial w_{I,i}}\right)$$

- Note that $y(k) = w_0^* r(k) + \cdots + w_M^* r(k-M)$ and

$$J(\mathbf{w}) = \frac{1}{2}\left(|y(k)|^2 - \Delta_2\right)^2 = \frac{1}{2}\left(y(k)y^*(k) - \Delta_2\right)^2$$

- Hence we have

$$\frac{\partial J}{\partial w_i} = \frac{1}{2}\cdot 2\left(y(k)y^*(k) - \Delta_2\right)\frac{\partial y(k)y^*(k)}{\partial w_i} = \left(|y(k)|^2 - \Delta_2\right)\left(\frac{\partial y(k)}{\partial w_i}y^*(k) + y(k)\frac{\partial y^*(k)}{\partial w_i}\right)$$

- Note

$$\frac{\partial y(k)}{\partial w_i} = \frac{1}{2}\left(\frac{\partial y(k)}{\partial w_{R,i}} + \mathrm{j}\frac{\partial y(k)}{\partial w_{I,i}}\right)$$

$$\frac{\partial y(k)}{\partial w_{R,i}} = r(k-i) \quad \text{and} \quad \frac{\partial y(k)}{\partial w_{I,i}} = -\mathrm{j}r(k-i)$$

- This leads to

$$\frac{\partial y(k)}{\partial w_i} = r(k-i)$$

Complex Variable Derivative (continue)

- Note

$$\frac{\partial y^*(k)}{\partial w_i} = \frac{1}{2}\left(\frac{\partial y^*(k)}{\partial w_{R,i}} + \mathrm{j}\frac{\partial y^*(k)}{\partial w_{I,i}}\right)$$

$$\frac{\partial y^*(k)}{\partial w_{R,i}} = r^*(k-i) \quad \text{and} \quad \frac{\partial y^*(k)}{\partial w_{I,i}} = \mathrm{j}r^*(k-i)$$

- This leads to

$$\frac{\partial y^*(k)}{\partial w_i} = 0$$

- Therefore, the gradient

$$\nabla J(\mathbf{w}) = \left[\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}\right]^T = y^*(k)(|y(k)|^2 - \Delta_2)\mathbf{r}(k)$$

- Using

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\left(-\nabla J(\mathbf{w}(k))\right)$$

- leads to

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu y^*(k)(\Delta_2 - |y(k)|^2)\mathbf{r}(k) = \mathbf{w}(k) + \mu\epsilon^*(k)\mathbf{r}(k)$$

where $\epsilon(k) = y(k)(\Delta_2 - |y(k)|^2)$

Concurrent CMA and Decision Directed

- The steady-state MSE of the CMA equaliser may not be sufficiently small to obtain an adequate performance (BER)
- A solution is to switch to a decision directed adaptation using the LMS, and this should significantly reduce the steady-state MSE
- However, the decision-directed LMS only works if the MSE is already low enough and this may not be achievable by the CMA
- How to automatically switch to decision directed LMS and how to know when can be switched? $\longrightarrow$ the concurrent CMA and DD algorithm
- The equaliser is divided into two parallel sub-equalisers:

$$\mathbf{w} = \mathbf{w}_c + \mathbf{w}_d$$

- The CMA sub-equaliser $\mathbf{w}_c$ is designed, as previously, to minimise the CMA cost function $\bar{J}_{\text{CMA}}(\mathbf{w}_c)$
- The concurrent decision-directed equaliser $\mathbf{w}_d$ is designed to minimise the decision based MSE

$$\bar{J}_{DD}(\mathbf{w}_d) = \frac{1}{2} E[|\mathcal{Q}[y(k)] - y(k)|^2]$$

where $\mathcal{Q}[y(k)]$ denotes the quantised equaliser output or equaliser hard decision

- Define an indicator function: $\delta(x) = 1$ if $x = 0 + \text{j}0$ and $\delta(x) = 0$ if $x \neq 0 + \text{j}0$

Concurrent CMA and DD (continue)

- At sample k, given $y(k) = \mathbf{w}_c^H(k)\mathbf{r}(k) + \mathbf{w}_d^H(k)\mathbf{r}(k)$, the CMA algorithm adapts $\mathbf{w}_c$ with adaptive gain μ_c
- The DD algorithm follows after the CMA adaptation with adaptive gain μ_d using

$$\mathbf{w}_d(k+1) = \mathbf{w}_d(k) + \mu_d \cdot \delta(\mathcal{Q}[\tilde{y}(k)] - \mathcal{Q}[y(k)]) \cdot (\mathcal{Q}[y(k)] - y(k))^* \mathbf{r}(k)$$

  where $\tilde{y}(k) = \mathbf{w}_c^H(k+1)\mathbf{r}(k) + \mathbf{w}_d^H(k)\mathbf{r}(k)$ is the equaliser output after the CMA adaptation
- Note that $(\mathcal{Q}[y(k)] - y(k))^* \mathbf{r}(k)$ is corresponding to the decision-directed adaptation
  - It only takes place if the equaliser's decisions before and after the CMA adaptation are the same, i.e. $\mathcal{Q}[\tilde{y}(k)] - \mathcal{Q}[y(k)] = 0 + \mathrm{j}0$
- This ensures that the CMA adaptation is probably a right one, and a DD adaptation can follow
- To reduce error propagation $\longrightarrow$ we have developed alternative soft DD
  - If equalisation has been achieved, posteriori PDF of $y(k)$ is approximately

$$p(\mathbf{w}, y(k)) \approx \sum_{q=1}^{Q} \sum_{l=1}^{Q} \frac{p_{ql}}{2\pi\rho} \exp\left(-\frac{|y(k) - s_{ql}|^2}{2\rho} \right)$$

  p_{ql} are priori probabilities of symbol points s_{ql} and we have $M = Q^2$-QAM

Concurrent CMA and SDD

- A local approximation of this posteriori PDF is

$$\widehat{p}(\mathbf{w}, y(k)) \approx \sum_{p=2i-1}^{2i} \sum_{q=2l-1}^{2l} \frac{1}{8\pi\rho} \exp\left(-\frac{|y(k)-s_{pq}|^2}{2\rho}\right)$$

with $S_{i,l} = \{s_{pq},\ p = 2i-1, 2i, q = 2l-1, 2l\}$

- SDD designed to maximise

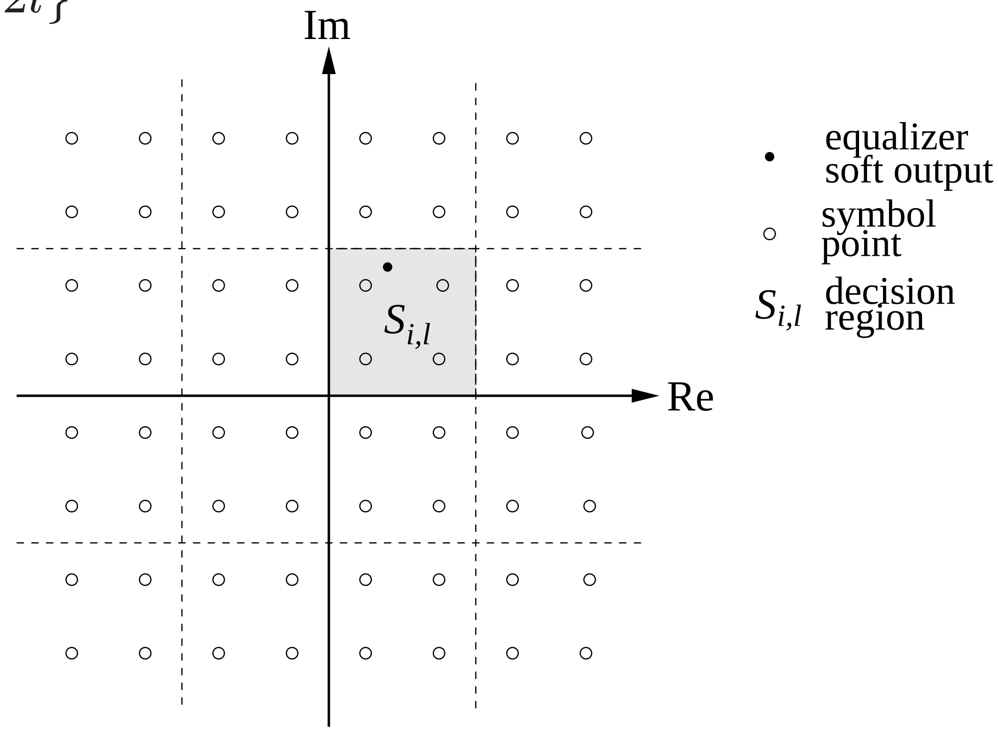
$$\bar{J}_{\text{LMAP}}(\mathbf{w}) = \mathsf{E}[J_{\text{LMAP}}(\mathbf{w}, y(k))]$$

by adjusting $\mathbf{w}_d$ where

$$J_{\text{LMAP}}(\mathbf{w}, y(k)) = \rho \log\left(\widehat{p}(\mathbf{w}, y(k))\right)$$

- Specifically

$$\mathbf{w}_d(k+1) = \mathbf{w}_d(k) + \mu_d \frac{\partial J_{\text{LMAP}}(\mathbf{w}(k), y(k))}{\partial \mathbf{w}_d}$$



Concurrent CMA and SDD (continue)

- Note that:

$$\frac{\partial J_{\text{LMAP}}(\mathbf{w}, y(k))}{\partial \mathbf{w}_d} = \frac{\sum_{p=2i-1}^{2i} \sum_{q=2l-1}^{2l} \exp\left(-\frac{|y(k)-s_{pq}|^2}{2\rho}\right)(s_{pq} - y(k))^*}{\sum_{p=2i-1}^{2i} \sum_{q=2l-1}^{2l} \exp\left(-\frac{|y(k)-s_{pq}|^2}{2\rho}\right)}\mathbf{r}(k)$$

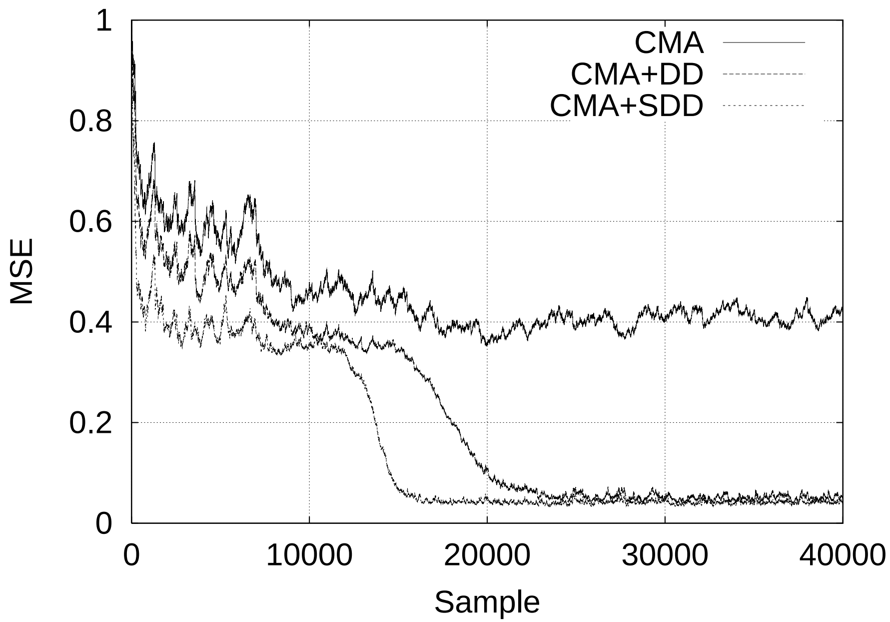
  – Soft decision: rather than committed to a single hard decision $\mathcal{Q}[y(k)]$ as the DD scheme does, alternative decisions are also considered in a local region $S_{i,l}$ that includes $\mathcal{Q}[y(k)]$
  – Each tentative decision is weighted by an exponential term $\exp(\bullet)$ which is a function of the distance between equaliser soft output $y(k)$ and the tentative decision s_{pq}
  – μ_d can be larger and $\rho < 1$ and not too small

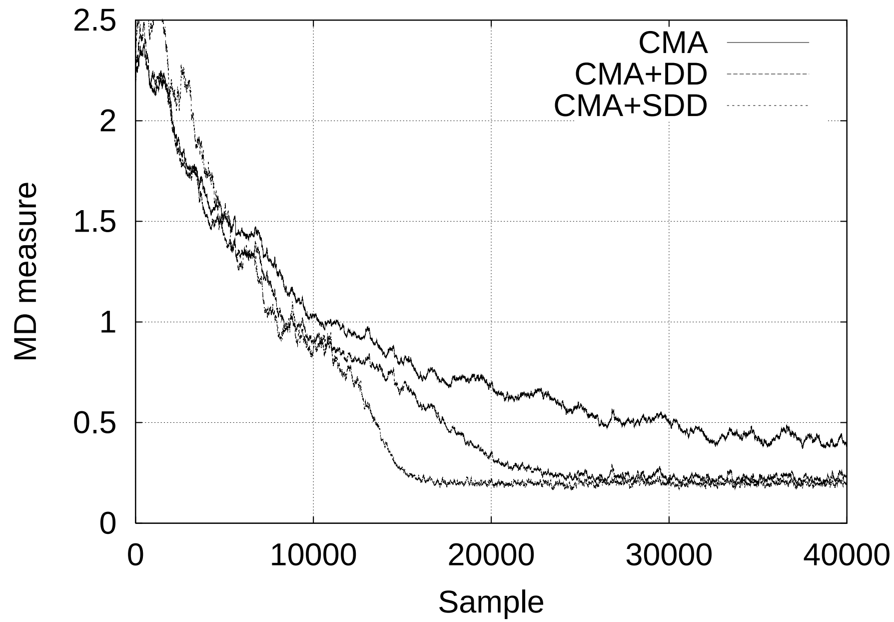
- **Example**: consists of a 22-tap channel and a 23-tap equaliser with 64-QAM and SNR$= 40$ dB
  – In simulation, an estimated MSE based on a separate block of data and the maximum distortion measure defined by

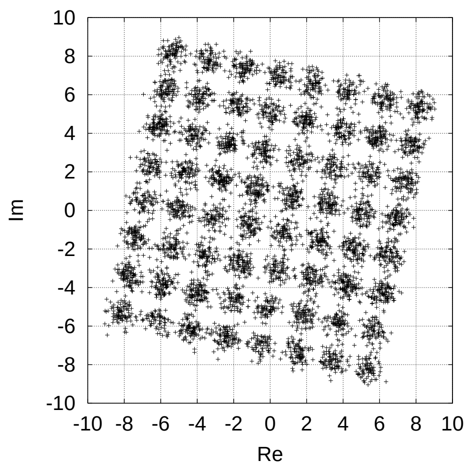
$$\text{MD} = \frac{\sum_{i=0}^{n_{\text{tot}}} |f_i| - |f_{i_{\max}}|}{|f_{i_{\max}}|}$$

  are used to assess convergence rate, where $\{f_i\}_{i=0}^{n_{\text{tot}}}$ is the combined impulse response of the channel and equaliser, $n_{\text{tot}} = n_c + M$, and $f_{i_{\max}} = \max\{f_i,\ 0 \le i \le n_{\text{tot}}\}$
  – Perfect equalisation corresponds to $f_{i_{\max}} = 1 + \text{j}0$ and $f_i = 0 + \text{j}0$ for $\le i \le n_{\text{tot}}$ and $i \ne i_{\max}$

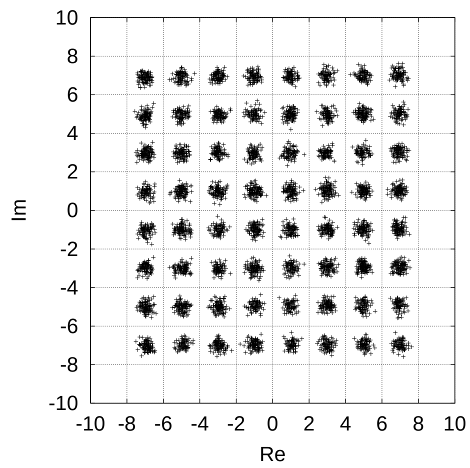
Simulation Results



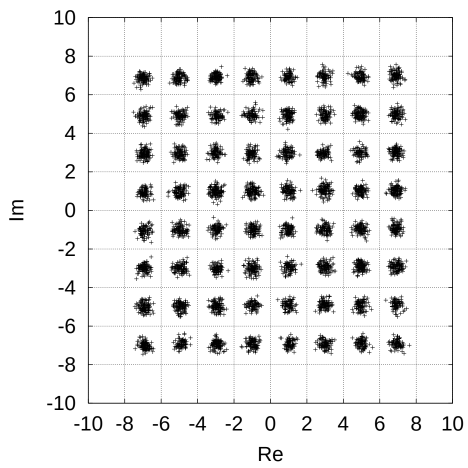




(a) CMA



(b) CMA+DD



(c) CMA+SDD

Summary

- Maximum likelihood sequence estimation using Viterbi algorithm: optimal equalisation performance but expensive
- Blind equalisation: three classes
- Low complexity blind equalisers for high-order QAM: the CMA, the concurrent CMA+DD, and the concurrent CMA+SDD