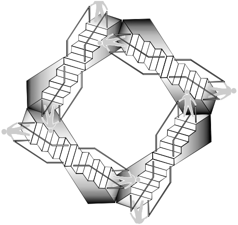
LINEAR EQUALIZATION

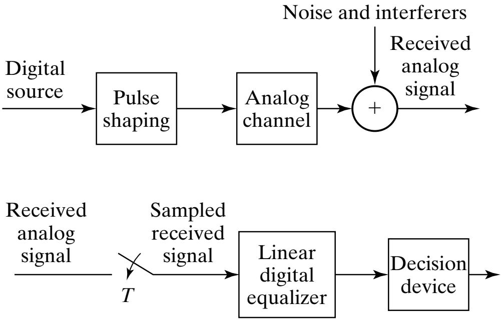
- ⋆ Multipath and Other Interference
- ⋆ Trained Least-Squares Linear Equalization
- ⋆ Trained Adaptive Least-Mean-Square Equalization
- ⋆ Blind Adaptive Decision-Directed Equalization
- ⋆ Blind Adaptive Dispersion Minimizing Equalization



adaptive components

Multipath and Other Interference

- Assume up and down conversion and carrier and clock recovery (including matched filtering and downsampling) all executed transparently.
- Impairment of interest is multipath interference (linear filtering by analog channel and receiver front-end preceding equalizer) and other additive interference (broadband noise and narrowband interferers).



Multipath ... Interference (cont'd)

- FIR channel model:

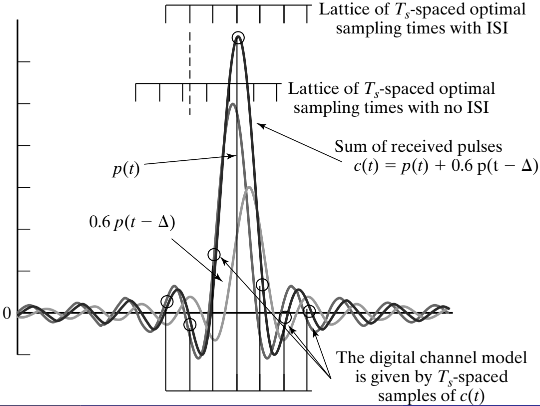
$$y(kT) = a_1 u(kT) + a_2 u((k-1)T)$$

$$+ \ldots + a_n u((k-n)T) + \eta(kT)$$

  where $\eta(kT)$ is sample of other interference.
- Order n of discrete-time FIR channel model dependent on physical delay spread of channel.
- For 4 μsec delay spread by "physical" channel:
  - $T = 0.04$ μsec $\rightarrow$ 25 Msymbols/sec $\rightarrow n = 100$
  - $T = 0.4$ μsec $\rightarrow$ 2.5 Msymbols/sec $\rightarrow n = 10$
  - $T = 4$ μsec $\rightarrow$ 0.25 Msymbols/sec $\rightarrow n = 1$

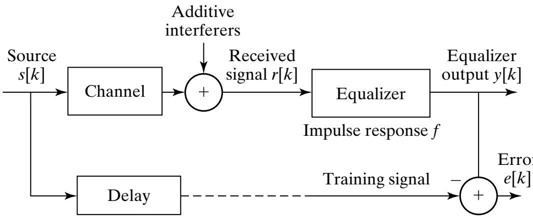
Multipath ... Interference (cont'd)

- Multipath FIR model coefficients depend on actual baud-timing choice of clock recovery algorithm, which need not match timing in non-ISI situation.
- Example: Two-ray analog channel $c(t) = p(t) + 0.6p(t - \Delta)$ with $\Delta = 0.7T$

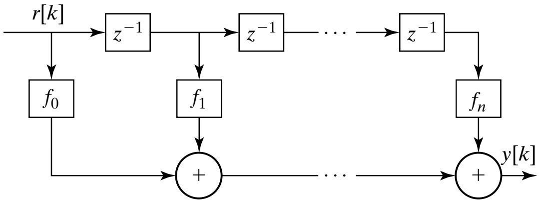


Trained Least-Squares Linear Equalization

- *Objective*: Choose impulse response f of equalizer so $y[k] \approx s[k-\delta]$ (so $e \approx 0$) for some δ.



- *Equalizer Output*: $y[k] = \sum_{j=0}^{n} f_j r[k-j]$



Trained ... Equalization (cont'd)

- Write equalizer output for $k = n + 1$ as inner product

$$y[n+1] = [r[n+1],\ r[n], \ldots,\ r[1]] \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}$$

- Similarly, for $k = n + 2$

$$y[n+2] = [r[n+2],\ r[n+1], \ldots,\ r[2]] \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}$$

Trained ... Equalization (cont'd)

Concatenating these equations for $k = n + 1$ to p

$$\begin{bmatrix} y[n+1] \\ y[n+2] \\ y[n+3] \\ \vdots \\ y[p] \end{bmatrix} = \begin{bmatrix} r[n+1] & r[n] & \dots & r[1] \\ r[n+2] & r[n+1] & \dots & r[2] \\ r[n+3] & r[n+2] & \dots & r[3] \\ \vdots & \vdots & & \vdots \\ r[p] & r[p-1] & \dots & r[p-n] \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}$$

or with appropriate definitions

$$Y = RF$$

where R with its diagonal stripes of repeated values is a Toeplitz matrix.

Trained ... Equalization (cont'd)

Delayed source recovery error:

$$e[k] = s[k-\delta] - y[k]$$

Delayed source vector:

$$S = \begin{bmatrix} s[n+1-\delta] \\ s[n+2-\delta] \\ s[n+3-\delta] \\ \vdots \\ s[p-\delta] \end{bmatrix}$$

Error vector:

$$E = \begin{bmatrix} e[n+1] \\ e[n+2] \\ e[n+3] \\ \vdots \\ e[p] \end{bmatrix}$$

$$= S - Y = S - RF$$

Trained ... Equalization (cont'd)

Average squared delayed source recovery error:

$$\bar{J} = \left(\frac{1}{p-n}\right) \sum_{i=n+1}^{p} e^2[i]$$

Summed squared error:

$$\begin{aligned} J &= \textstyle\sum_{i=n+1}^{p} e^2[i] \\ &= E^T E \\ &= (S - RF)^T (S - RF) \\ &= S^T S - (RF)^T S - S^T RF + (RF)^T RF \end{aligned}$$

Because J is a scalar, $(RF)^T S$ and $S^T RF$ are scalars and

$$(RF)^T S = ((RF)^T S)^T = S^T ((RF)^T)^T = S^T RF$$

so

$$J = S^T S - 2S^T RF + (RF)^T RF$$

Trained ... Equalization (cont'd)

- Define

$$\begin{aligned}\Psi &\triangleq [F-(R^TR)^{-1}R^TS]^T(R^TR)\cdot[F-(R^TR)^{-1}R^TS] \\ &= F^T(R^TR)F - S^TRF - F^TR^TS + S^TR(R^TR)^{-1}R^TS\end{aligned}$$

- Rewrite J as

$$\begin{aligned}J &= \Psi + S^TS - S^TR(R^TR)^{-1}R^TS \\ &= \Psi + S^T[I - R(R^TR)^{-1}R^T]S\end{aligned}$$

- Because the term $S^T[I-R(R^TR)^{-1}R^T]S$ is not a function of F, the minimum of J by choice of F occurs at the F that minimizes Ψ, i.e.

$$F^* = (R^TR)^{-1}R^TS$$

assuming $(R^TR)^{-1}$ exists.

Trained ... Equalization (cont'd)

- The remaining term in J when $F = F^*$ is the minimum achievable (summed squared delayed source recovery error) cost for the associated δ
$$J_{\min} = S^T[I - R(R^T R)^{-1} R^T]S$$
- *Example (using* `LSequalizer`*)*: Indicating importance of appropriate delay δ selection
  - Source: binary (± 1)
  - T-spaced channel impulse response: $\{0.5,\ 1,\ -0.6\}$ for $k = 0, 1, 2$
  - Equalizer length: $n + 1 = 4$
  - Data record length: $p = 1000$
  - Additive interferers: none

Trained ... Equalization (cont'd)

- *Example (cont'd)*
  - Results:

δ	J_{min}	F^*
0	832	$\{0.33,\ 0.027,\ 0.070,\ 0.01\}$
1	134	$\{0.66,\ 0.36,\ 0.16,\ 0.08\}$
2	30	$\{-0.28,\ 0.65,\ 0.30,\ 0.14\}$
3	45	$\{0.1,\ -0.27,\ 0.64,\ 0.3\}$

  - Smallest J_{min} for $\delta = 2$
  - All δ except $\delta = 0$ result in open eye and no decision errors.

Trained ... Equalization (cont'd)

Another Example:

- Equalizer: $y[k] = f_0 r[k] + f_1 r[k-1]$
- Received signal data set:

$$\{r[k]\} = \{r[1],\ r[2],\ r[3],\ r[4],\ r[5]\}$$

- Source signal data set:

$$\{s[k]\} = \{s[1],\ s[2],\ s[3],\ s[4],\ s[5]\}$$

- Zero-delay objective: $y[k] \sim s[k]$. The largest collection of equations available from dataset is

$$\begin{bmatrix} s[2] \\ s[3] \\ s[4] \\ s[5] \end{bmatrix} \sim \begin{bmatrix} r[2] & r[1] \\ r[3] & r[2] \\ r[4] & r[3] \\ r[5] & r[4] \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \end{bmatrix}$$

Trained ... Equalization (cont'd)

Another Example (cont'd):

- $\delta = 1$ objective: $y[k] \sim s[k-1]$. The largest collection of equations available from dataset is

$$\begin{bmatrix} s[1] \\ s[2] \\ s[3] \\ s[4] \end{bmatrix} \sim \begin{bmatrix} r[2] & r[1] \\ r[3] & r[2] \\ r[4] & r[3] \\ r[5] & r[4] \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \end{bmatrix}$$

- $\delta = 2$ objective: $y[k] \sim s[k-2]$. The largest collection of equations available from dataset is

$$\begin{bmatrix} s[1] \\ s[2] \\ s[3] \end{bmatrix} \sim \begin{bmatrix} r[3] & r[2] \\ r[4] & r[3] \\ r[5] & r[4] \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \end{bmatrix}$$

Trained ... Equalization (cont'd)

Another Example (cont'd):
Largest common set of equations for testing delays from 0 to $n+1$:

$$\begin{bmatrix} s[3] & s[2] & s[1] \\ s[4] & s[3] & s[2] \\ s[5] & s[4] & s[3] \end{bmatrix} \sim \begin{bmatrix} r[3] & r[2] \\ r[4] & r[3] \\ r[5] & r[4] \end{bmatrix} \begin{bmatrix} f_{00} & f_{01} & f_{02} \\ f_{10} & f_{11} & f_{12} \end{bmatrix}$$

where f_{ij} corresponds to i for index of delay associated with coefficient in equalizer FIR and j for desired delay in channel-equalizer combination.

Trained ... Equalization (cont'd)

Another Example (cont'd):
All together now... $\bar{S} \sim \bar{R}\bar{F}$ with

$$\bar{S} = \begin{bmatrix} s[\alpha+1] & s[\alpha] & \ldots & s[1] \\ s[\alpha+2] & s[\alpha+1] & \ldots & s[2] \\ \vdots & \vdots & & \vdots \\ s[p] & s[p-1] & \ldots & s[p-\alpha] \end{bmatrix}$$

$$\bar{R} = \begin{bmatrix} r[\alpha+1] & r[\alpha] & \ldots & r[\alpha-n+1] \\ r[\alpha+2] & r[\alpha+1] & \ldots & r[\alpha-n+2] \\ \vdots & \vdots & & \vdots \\ r[p] & r[p-1] & \ldots & r[p-n] \end{bmatrix}$$

$$\bar{F} = \begin{bmatrix} f_{00} & f_{01} & \cdots & f_{0\alpha} \\ f_{10} & f_{11} & \cdots & f_{1\alpha} \\ \vdots & \vdots & & \vdots \\ f_{n0} & f_{n1} & \ldots & f_{n\alpha} \end{bmatrix}$$

Trained ... Equalization (cont'd)

Another Example (cont'd):

- In our example, $n = 1$, $\alpha = 2$, and $p = 5$
- Summed squared delayed source recovery error minimized for delays from zero to α by columns of

$$\bar{F}^* = (\bar{R}^T \bar{R})^{-1} \bar{R}^T \bar{S}$$

- Minimum cost for a particular delay δ associated with $(\delta + 1)$th (or ℓth) column of $\bar{F}^*$:

$$J_{\min,\ell} = \bar{S}_\ell^T [I - \bar{R}(\bar{R}^T \bar{R})^{-1} \bar{R}^T] \bar{S}_\ell$$

  where $\bar{S}_\ell^*$ is ℓth columns of $\bar{S}^*$.
- Matrix with diagonal as minimum costs for various delays:

$$\Phi = \bar{S}^T [I - \bar{R}(\bar{R}^T \bar{R})^{-1} \bar{R}^T] \bar{S}$$

Trained ... Equalization (cont'd)

The steps of the linear FIR equalizer design strategy are:

1. Select the order n for the FIR equalizer.
2. Select maximum of candidate delays α $(> n)$.
3. Utilize set of p training signal samples $\{s[1],\ s[2],\ ...,\ s[p]\}$ with $p > n + \alpha$.
4. Obtain corresponding set of p received signal samples $\{r[1],\ r[2],\ ...,\ r[p]\}$.
5. Compose $\bar{S}$.
6. Compose $\bar{R}$.
7. Check if $\bar{R}^T\bar{R}$ has poor conditioning induced by any (near) zero eigenvalues.
8. Compute $\bar{F}^*$.
9. Compute $\Phi = \bar{S}^T[\bar{S} - \bar{R}\bar{F}^*]$.

Trained ... Equalization (cont'd)

Equalizer design strategy (cont'd):

10. Find the minimum value on the diagonal of Φ. This index is $\delta + 1$. The associated diagonal element of Φ is the minimum achievable summed squared delayed source recovery error $\sum_i e^2[i]$ over the available data record.
11. Extract the $(\delta + 1)$th column of the previously computed $\bar{F}^*$. This is the impulse response of the optimum equalizer.
12. Test the design. Test it on synthetic data, and then on measured data (if available). If inadequate, repeat design, perhaps increasing n or twiddling some other designer-selected quantity.

Trained ... Equalization (cont'd)

Complex Signals:

- For modulations such as QAM, the signals (and parameters) are effectively complex valued.
- For a complex error $e[k] = e_R[k] + je_I[k]$ where $j = \sqrt{-1}$, consider $e[k]e^*[k]$ where $*$ superscript indicates complex conjugation.
- The cost

$$\begin{aligned} e[k]e^*[k] &= e_R^2[k] - je_R[k]e_I[k] \\ &+je_R[k]e_I[k] - j^2e_I^2[k] \\ &= e_R^2[k] + e_I^2[k] \end{aligned}$$

  is desirably nonnegative.
- Optimal equalizer to minimize $\sum_k e[k]e^*[k]$ is

$$F^* = (R^H R)^{-1} R^H S$$

  where superscript H denotes transposition and complex conjugation.

Trained ... Equalization (cont'd)

Fractionally-Spaced Equalizer:

- For an equalizer with an input sampled M times per symbol period, we wish to minimize the square of e only at the the baud times, i.e. every Mth sample (with synchronized sampler).
- Thus, only every Mth e in E matters, and the underlying equations of interest are the rows of $E = S - RF$ left after removing all but every Mth one.
- The remaining matrix equation is solved, which can admit a perfect solution if the row-decimated R has been reduced to a square matrix.

Trained Adaptive Least-Mean-Square (LMS) Equalization

We choose to minimize

$$\text{avg}\{e^2[k]\} = \frac{1}{N}\sum_{k=k_0}^{k_0+N-1} e^2[k]$$

with $e[k] = s[k-\delta] - \sum_{i=0}^{n} f_i r[k-i]$ using a gradient descent scheme

$$f_i[k+1] = f_i[k] - \bar{\mu}\frac{\partial(\text{avg}\{e^2[k]\})}{\partial f_i}\Big|_{f=f[k]}$$

With differentiation and average approximately commutable (see App. G)

$$f_i[k+1] \approx f_i[k] - \bar{\mu}\cdot\text{avg}\left\{\frac{\partial e^2[k]}{\partial f_i}\Big|_{f=f[k]}\right\}$$

Dropping the "outer" average produces LMS

$$f_i[k+1] = f_i[k] - 2\bar{\mu}\left(e[k]\frac{\partial e[k]}{\partial f_i}\right)\Big|_{f=f[k]}$$

$$= f_i[k] + \mu(s[k-\delta] - y[k])r[k-i]$$

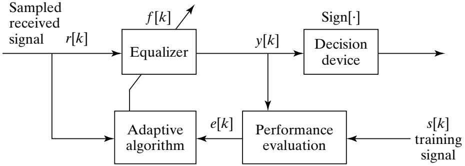
with $y[k] = \sum_{j=0}^{n} f_j[k]r[k-j]$.

Trained Adaptive Least-Mean-Square (LMS) Equalization (cont'd)

With the definition of the FIR equalizer output

$$y[k] = \sum_{j=0}^{n} f_j[k] r[k-j]$$

in



the trained approximate gradient descent adaptation algorithm LMS for the linear equalizer is

$$f_i[k+1] = f_i[k] + \mu(s[k-\delta] - y[k]) r[k-i]$$

Blind Adaptive Decision-Directed Equalization

We choose to minimize

$$\text{avg}\{(\mathrm{Q}(\sum_{j=0}^{n} f_j r[k-j]) - \sum_{j=0}^{n} f_j r[k-j])^2\}$$

$$= \frac{1}{N} \sum_{k=k_0}^{k_0+N-1} (\mathrm{Q}(\sum_{j=0}^{n} f_j r[k-j]) - \sum_{j=0}^{n} f_j r[k-j])^2$$

using a gradient descent scheme

$$f_i[k+1] = f_i[k] - \bar{\mu} \frac{\partial}{\partial f_i} \left(\text{avg}\{(\mathrm{Q}(\sum_{j=0}^{n} f_j r[k-j]) - \sum_{j=0}^{n} f_j r[k-j])^2\} \right) \Big|_{f=f[k]}$$

Blind Adaptive Decision-Directed Equalization (cont'd)

Commute average and partial derivative, drop "outer" average, and presume $\partial(\mathrm{Q}(\sum_{j=0}^{n} f_j r[k-j]))/\partial f_i = 0$ to produce

$$f_i[k+1] = f_i[k] - 2\bar{\mu}\{(\mathrm{Q}(\sum_{j=0}^{n} f_j r[k-j])$$

$$- \sum_{j=0}^{n} f_j r[k-j]) \frac{\partial(-\sum_{j=0}^{n} f_j r[k-j])}{\partial f_i} \}|_{f=f[k]}$$

$$= f_i[k] - 2\bar{\mu} \left(\mathrm{Q}(\sum_{j=0}^{n} f_j[k] r[k-j]) \right.$$

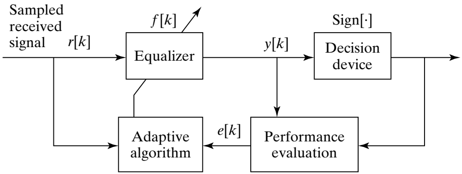
$$\left. - \sum_{j=0}^{n} f_j[k] r[k-j] \right) (-r[k-i])$$

Blind ... Equalization (cont'd)

With the definition of

$$y[k] = \sum_{j=0}^{n} f_j[k] r[k-j]$$

in



the decision-directed approximate gradient descent adaptation algorithm for the linear FIR equalizer is

$$f_i[k] = f_i[k] + \mu(\mathrm{Q}(y[k]) - y[k]) r[k-i]$$

- Relative to trained adaptation via LMS, the decision device output just replaces the training signal.

Blind Adaptive Dispersion-Minimizing Equalization

We choose to minimize

$$\text{avg}\{(1-(\sum_{j=0}^{n} f_j r[k-j])^2)^2\} = \frac{1}{N}\sum_{k=k_0}^{k_0+N-1}(1-(\sum_{j=0}^{n} f_j r[k-j])^2)^2$$

using a gradient descent scheme

$$f_i[k+1] = f_i[k] - \bar{\mu}\frac{\partial\left(\text{avg}\{(1-(\sum_{j=0}^{n} f_j r[k-j])^2)^2\}\right)}{\partial f_i}|_{f=f[k]}$$

Commuting average and differentiation and dropping "outer" average produces

$$f_i[k+1] = f_i[k] + 2\bar{\mu}\{(1-(\sum_{j=0}^{n} f_j r[k-j])^2)$$

$$\cdot\frac{\partial(\sum_{j=0}^{n} f_j r[k-j])^2}{\partial f_i}\}|_{f=f[k]}$$

Blind ... Equalization (cont'd)

Evaluating derivative produces

$$f_i[k+1] = f_i[k] + \mu(1 - (\sum_{j=0}^{n} f_j[k]r[k-j])^2) \cdot (\sum_{j=0}^{n} f_j[k]r[k-j])r[k-i]$$

where

$$\sum_{j=0}^{n} f_j[k]r[k-j] = y[k]$$

so

$$f_i[k+1] = f_i[k] + \mu(1 - y^2[k])y[k]r[k-i]$$

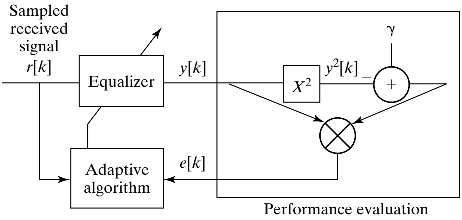
In comparison to LMS the prediction error $s[k-\delta] - y[k]$ has been effectively replaced by $(1 - y^2[k])y[k]$.

Blind ... Equalization (cont'd)

With the definition of

$$y[k] = \sum_{j=0}^{n} f_j[k] r[k-j]$$

in



the dispersion-minimizing approximate gradient descent adaptation algorithm for the linear FIR equalizer is

$$f_i[k+1] = f_i[k] + \mu(1 - y^2[k]) y[k] r[k-i]$$

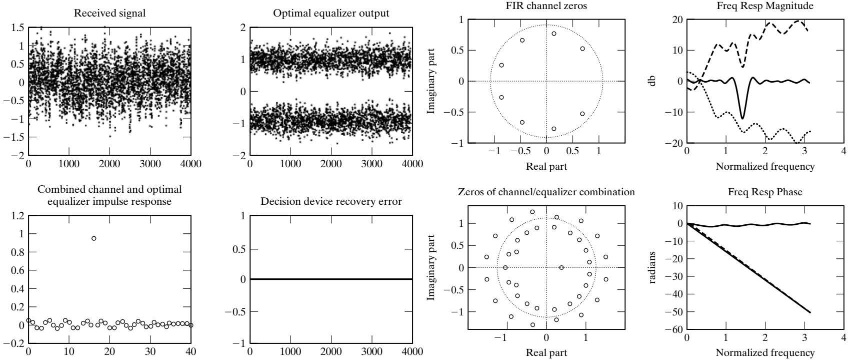
- The adaptive scheme is labelled as blind (rather than trained) due to the creation of the correction term without a training signal.

Example (using dae)

- Source: binary (± 1)
- Channel:
  - Zero: $\{1\ .9\ .81\ .73\ .64\ .55\ .46\ .37\ .28\}/4.138$
  - One: $\{1\ 1\ 1\ 0.2\ \text{-}0.4\ 2\ \text{-}1\}/8.2$
  - Two: $\{\text{-}0.2\ .1\ .3\ 1\ 1.2\ .4\ \text{-}.3\ \text{-}.2\ .3\ .1\ \text{-}.1\}/2.98$
- Sinusoidal interferer frequency: 1.4 radians/sample
- Some broadband noise present
- Equalizer length: 33

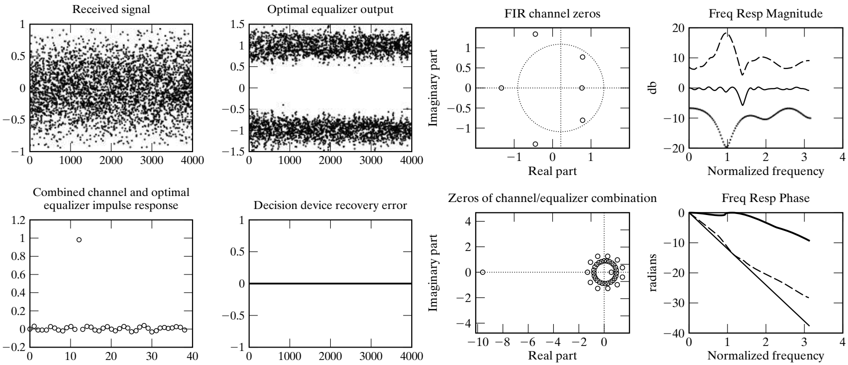
Example (cont'd)

Trained LS for channel zero:



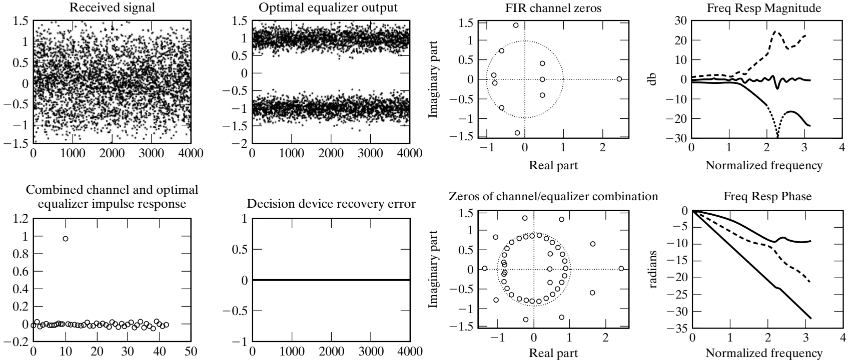
Example (cont'd)

Trained LS for channel one:



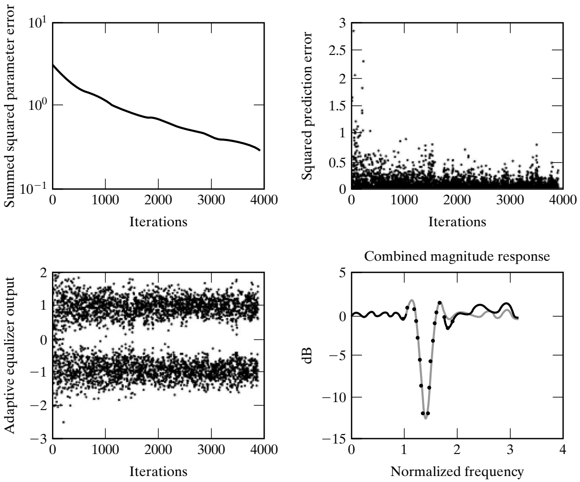
Example (cont'd)

Trained LS for channel two:



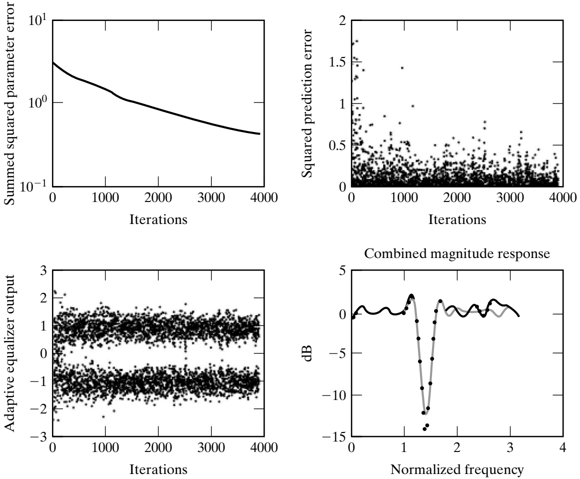
Example (cont'd)

Trained LMS for channel zero:



Example (cont'd)

Decision-directed for channel zero:



Example (cont'd)

Dispersion minimization for channel zero: