EE 5654 - Digital Communications Spring 2005

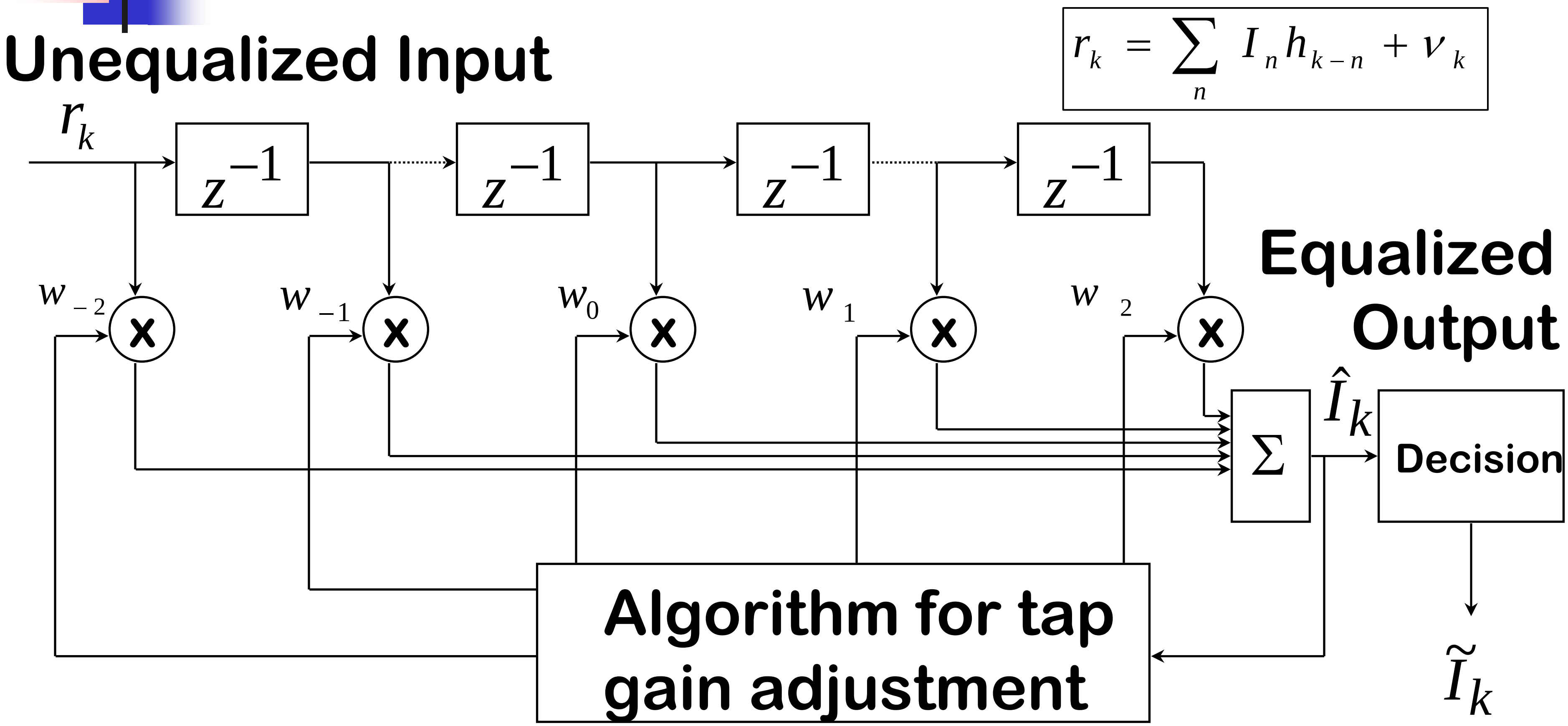
Lecture #16 - Adaptive Algorithms for Equalization

Instructor: R. Michael Buehrer

Introduction

- Previously we examined equalizers which can improve performance in ISI channels
- In time varying channels, the optimal equalizer coefficients will very with time and thus must be constantly estimated and updated
- This update technique is called an *adaptive algorithm*
- The algorithm used depends both on the minimization criterion and the update approach

Linear Transversal Filter



Linear Transversal Filter Equalizer

- Complex samples of the received signal input into the equalizer: r_k
- Each element z^{-1} represents a one sample delay
- There are *2K+1* taps on the equalizer where the ISI produces *K* samples of overlap in the signal
- The equalizer has $2K+1$ complex coefficients w_k
- Equalized output samples: $\hat{I}_k = \sum_{j=-K}^{K} w_j r_{k-j}$
- Decisions $\tilde{I}_k$ are made based on $\hat{I}_k$

Distortion Criteria

- Choice of tap coefficients depends on choice of decision criteria.
- Peak distortion criteria leads to "zero-forcing equalizer"
  - Robert Lucky of Bell Labs was famous for his work on this
- Mean Squared Error (MSE) criteria is much more widely used:

$$E\left[\left(I_k - \hat{I}_k\right)^2\right]$$

Non-adaptive MMSE Solution

- We wish to minimize: $$E\left[\left|I_k - \sum_{j=-K}^{K} w_j r_{k-j}\right|^2\right]$$
- In matrix form, it can be shown that the weight vector **w** = [w_{-K} w_{-K+1} ...w_{K-1} w_K]^T which minimizes the MSE is:

$$\mathbf{w} = \mathbf{R}_{rr}^{-1}\mathbf{d}$$

- where $\mathbf{R}_{rr}$ is the covariance matrix of the input data

$$R_{rr} = E\left[\mathbf{r}\mathbf{r}^T\right]$$

- and **d** is the correlation between the received signal and the information sequence (*i.e.*, the received channel vector)

$$\mathbf{d} = E\left[\mathbf{r}_k I_k\right] = \mathbf{h}$$

Note that the vectors are defined over the $2K$+1 time samples

More on the Non-adaptive Solution

- The coefficients **h** represent the channel and are as defined last time (i.e., the ISI and channel of the desired signal after filtering).
- The vector **w** represents the equalizer coefficients
- The 2K+1 X 2K+1 matrix $\mathbf{R}_{rr}$ represents the covariance matrix of the received signal samples
- The vector **d** represents the correlation between the input signal and the data
- The direct solution requires estimation of $\mathbf{R}_{rr}$ and **d** followed by inversion of $\mathbf{R}_{rr}$ and multiplication by **d**. This approach to the MMSE solution is sometimes called Direct Matrix Inversion or DMI

Problems with Non-adaptive Solution

- Requires direct computation of a full matrix inversion with every update.
- Explicitly makes use of channel parameters which must be estimated.
- Combined, this results in a very computationally intense approach, especially in a time varying environment

Solution: Real-time Adaptive Algorithms

- Least Mean Square (LMS)
  - Gradient search technique
- Recursive Least Squares (RLS)
  - Estimates matrix inverse

Least Mean Squares (LMS) Algorithm

- We begin with an initial guess of the solution $\mathbf{w}_0$:
- We form a sequence of estimates: $\mathbf{w}_0$, $\mathbf{w}_1$, $\mathbf{w}_2$, of the solution to the equation: $\mathbf{R}_{\mathbf{rr}}\mathbf{w} = \mathbf{d}$
- The estimates of solution are found from the steepest descent method
- Estimates are updated iteratively: $\mathbf{w}_{k+1} = \mathbf{w}_k - \Delta\, G_k$
- Δ is a small constant to control the step size of the update
- G_k is the gradient vector of the error (i.e., the derivative of the MSE wrt the coefficients)

$$\mathbf{G}_k = \mathbf{R}_{rr}\mathbf{w}_k - \mathbf{d} = -E\left[e_k \mathbf{r}_k^*\right]$$

$$e_k = I_k - \hat{I}_k$$

Gradient Vector in the LMS Algorithm

- Calculation of the gradient vector requires knowledge of the covariance matrix $\mathbf{R}_{rr}$ and correlation vector $\mathbf{d}$.
- In practice, we use the actual samples as the unbiased estimate of the true gradient.
- This is a stochastic version of the method of steepest descent
- **The LMS Algorithm:**

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \Delta e_k \mathbf{r}_k^*$$

Unbiased estimate of the gradient

Interpretation of LMS Algorithm

- Coefficients are adjusted in small steps to reduce the error in the response of the equalizer to the current set of input samples.
- Coefficients can be updated after each new sample.
- Note that this algorithm implicitly makes use of channel parameters, but only makes use of the received signal samples which we can directly measure.
- Error calculation requires knowledge of the information symbols I_k. which are in general unknown. Thus, we must send a training sequence periodically. (Can use *decision directed* mode in between training.)

The Role of Step Size in LMS Algorithm

- Larger Δ means faster convergence to optimum
- Need $\Delta < 2/\lambda_{\max}$ to guarantee convergence of algorithm, where $\lambda_{\max}$ is the largest eigenvalue of the covariance matrix $\mathbf{R}_{rr}$.
- Larger Δ also means that the algorithm will be less able to finely converge to the optimal solution at the end - sometimes called "Excess MSE"
- Too small a choice for Δ may prevent the LMS algorithm from adapting to a rapidly-varying channel

A Key Weakness of the LMS Algorithm

- Clearly, a good choice of Δ must be made to avoid too slow initial convergence or poor final convergence and instability.
- The same step size Δ must be used for the update of all coefficients, even though some may be more sensitive to instability and others may be more important to update rapidly
- Solution: Recursive Least Squares Algorithm

Recursive Least Squares (RLS) Algorithm

- The RLS algorithm implements the optimal solution

$$\mathbf{w} = \mathbf{R}_{rr}^{-1}\mathbf{d}$$

by iteratively estimating $\mathbf{R}_{\text{rr}}$ and $\mathbf{d}$ and by using the matrix inversion lemma

- Stochastic estimate of R_{rr} :

$$\widehat{\mathbf{R}}_k = \alpha \widehat{\mathbf{R}}_{k-1} + \mathbf{r}_k \mathbf{r}_k^{\dagger}$$

- Stochastic inverse using matrix inversion lemma:

$$\widehat{\mathbf{R}}_k^{-1} = \frac{1}{\alpha}\left\{ \widehat{\mathbf{R}}_{k-1}^{-1} - \frac{\widehat{\mathbf{R}}_{k-1}^{-1}\mathbf{r}_k \mathbf{r}_k^{\dagger}\widehat{\mathbf{R}}_{k-1}^{-1}}{\alpha + \mathbf{r}_k^{\dagger}\widehat{\mathbf{R}}_{k-1}^{-1}\mathbf{r}_k} \right\}$$

Recursive Least Squares (RLS) Algorithm

- We then update the weights according to:

$$
\begin{aligned}
\mathbf{w}_{k+1} &= \mathbf{R}_{k+1}^{-1}\mathbf{d}_{k+1} \\
&= \frac{1}{\alpha}\left[\mathbf{R}_k^{-1} - \mathbf{K}_{k+1}\mathbf{r}_k^{\dagger}\mathbf{R}_k^{-1}\right]\left[\alpha\mathbf{d}_k + I_k\mathbf{r}_k^{*}\right] \\
&= \mathbf{w}_k + \mathbf{K}_{k+1}\left[I_k - \mathbf{r}_k^{\dagger}\mathbf{w}_k\right] \\
&= \mathbf{w}_k + \mathbf{K}_{k+1}e_k
\end{aligned}
$$

- Where

$$
\mathbf{K}_{k+1} = \frac{\mathbf{R}_k^{-1}\mathbf{r}_k}{\alpha + \mathbf{r}_k\mathbf{R}_k^{-1}\mathbf{r}_k^{\dagger}}
$$

Recursive Least Squares (RLS) Algorithm (cont.)

- Coefficients are updated according to:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mathbf{K}_{k+1} e_k$$

- In this formulation, $\mathbf{K}_k$ is called the Kalman gain vector. It is a computed according to:

$$\mathbf{K}_{k+1} = \frac{\mathbf{P}_k \mathbf{r}_k}{\alpha + \mathbf{r}_k \mathbf{P}_k \mathbf{r}_k^{\dagger}}$$

- Where $\mathbf{P}_k$ is the recursively updated inverse of the correlation matrix

$$\mathbf{P}_{k+1} = \frac{1}{\alpha}\left[\mathbf{P}_k - \mathbf{K}_{k+1} \mathbf{r}_k \mathbf{P}_k\right]$$

Recursive Least Squares (RLS) Algorithm

- The inverse of the cross-correlation matrix is computed recursively as follows:

$$\mathbf{P}_{k+1} = \frac{1}{\alpha}\left[\mathbf{P}_k - \mathbf{K}_{k+1}\mathbf{r}_k\mathbf{P}_k\right]$$

- This recursive method of computing the matrix inversion is the heart of the RLS algorithm.
- Instead of estimating the correlation matrix and inverting it, we estimate the inverse directly
- The coefficient $0 < \alpha < 1$ is a scalar which we will discuss in a moment.
- An RLS algorithm also exists for the DFE structure

Interpretations of the RLS Algorithm

- Rather than minimizing the *average* mean-square error like the LMS, the RLS algorithm is minimizing the actual squared error for the given data sequence.
- The RLS algorithm is an implementation of a more general technique known as Kalman Filtering.
- The Kalman gain replaces the fixed step size in updating the equalizer coefficients.

The Role of the Weighting Factor

- The weighting factor $0 < \alpha < 1$ determines how quickly older data is discarded from the updated estimate.
- Large α means that older data included in estimate for a longer period of time.
- Small α means that older data is discounted quickly.

RLS vs. LMS Algorithms

- The RLS algorithm converges faster than the LMS algorithm.
- The RLS algorithm is more computationally complex than the LMS algorithm
  - Note that the recursive nature of the RLS algorithm is still simpler than a full matrix inversion operation.
  - Some additional simplifications of the RLS computations are possible.
  - Note that the RLS algorithm operates directly on signal samples.

Trained vs. Blind Equalization

- Note that computation of the error $\varepsilon_k = I_k - \hat{I}_k$ is required for implementation of algorithms.
- Many systems use initial training sequences for which I_k is known to both transmitter and receiver
  - Example: GSM transmits periodic training sequences as part of each signal frame
- After the algorithms begins to converge, we can switch to "decision-directed" mode: $\varepsilon_k = \tilde{I}_k - \hat{I}_k$
- Recently, there has been interest in "Blind Equalization", which requires no training at all

References on Blind Equalization

- Y. Sato, "A method of self-recovering equalization for multilevel amplitude modulation systems," *IEEE Trans. Comm.*, pp. 679-682, June 1975.
- D. N. Godard, "Self-recovering equalization and data tracking for two dimensional data communication systems," *IEEE Trans. Comm.*, pp. 1867-1875, Nov. 1980.
- J. Bellini, "Bussgang techniques for blind equalization," *IEEE GLOBECOM '86*, pp. 46.1.1-46.1.7.

Summary of Adaptive Algorithms

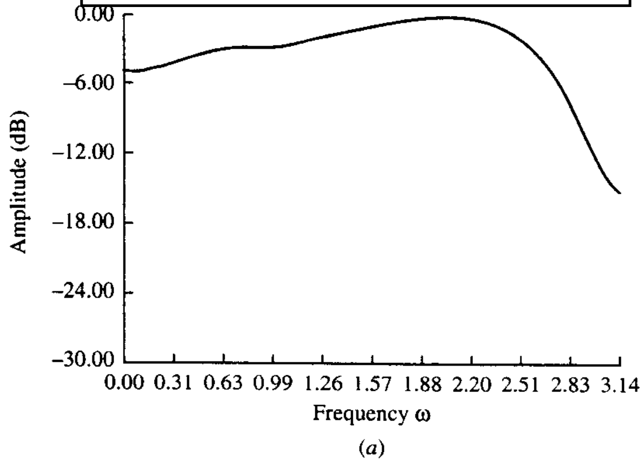
- Two adaptive algorithms were presented to adapt equalizer coefficients.
- The LMS algorithm is less complex
- The RLS algorithm converges faster
- User-specified parameters determine the speed of convergence for each of these algorithms.

Performance of Equalizers

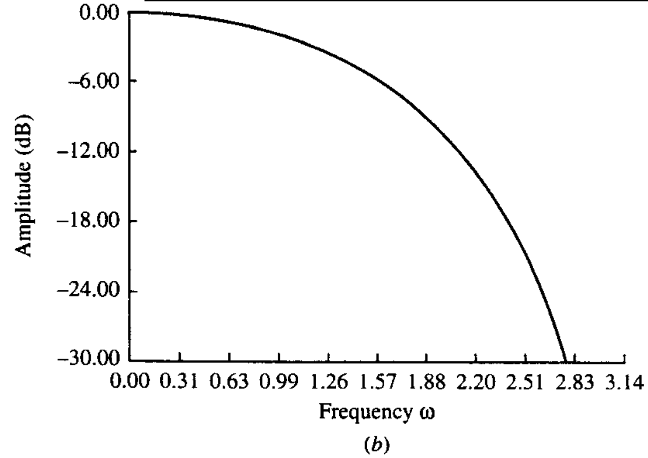
- We are interested in:
  - Steady State Performance
    - Performance After Convergence
    - Largely dependent on structure (provided algorithm converges)
  - Transient Performance
    - Rate of Convergence
    - Dependent on Adaptive Algorithm

Three Frequency Selective Channels

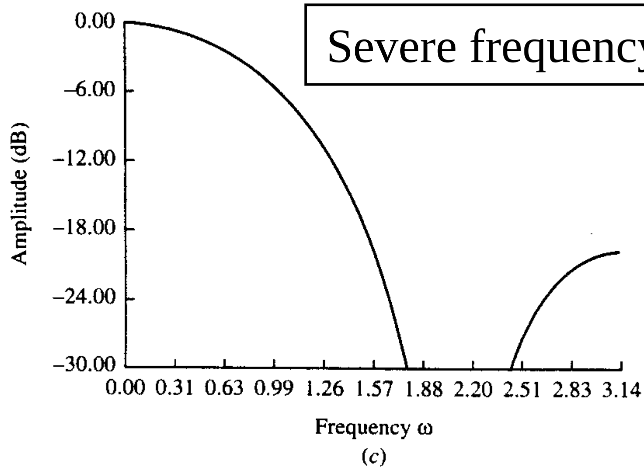
Mild frequency selectivity



Moderate frequency selectivity



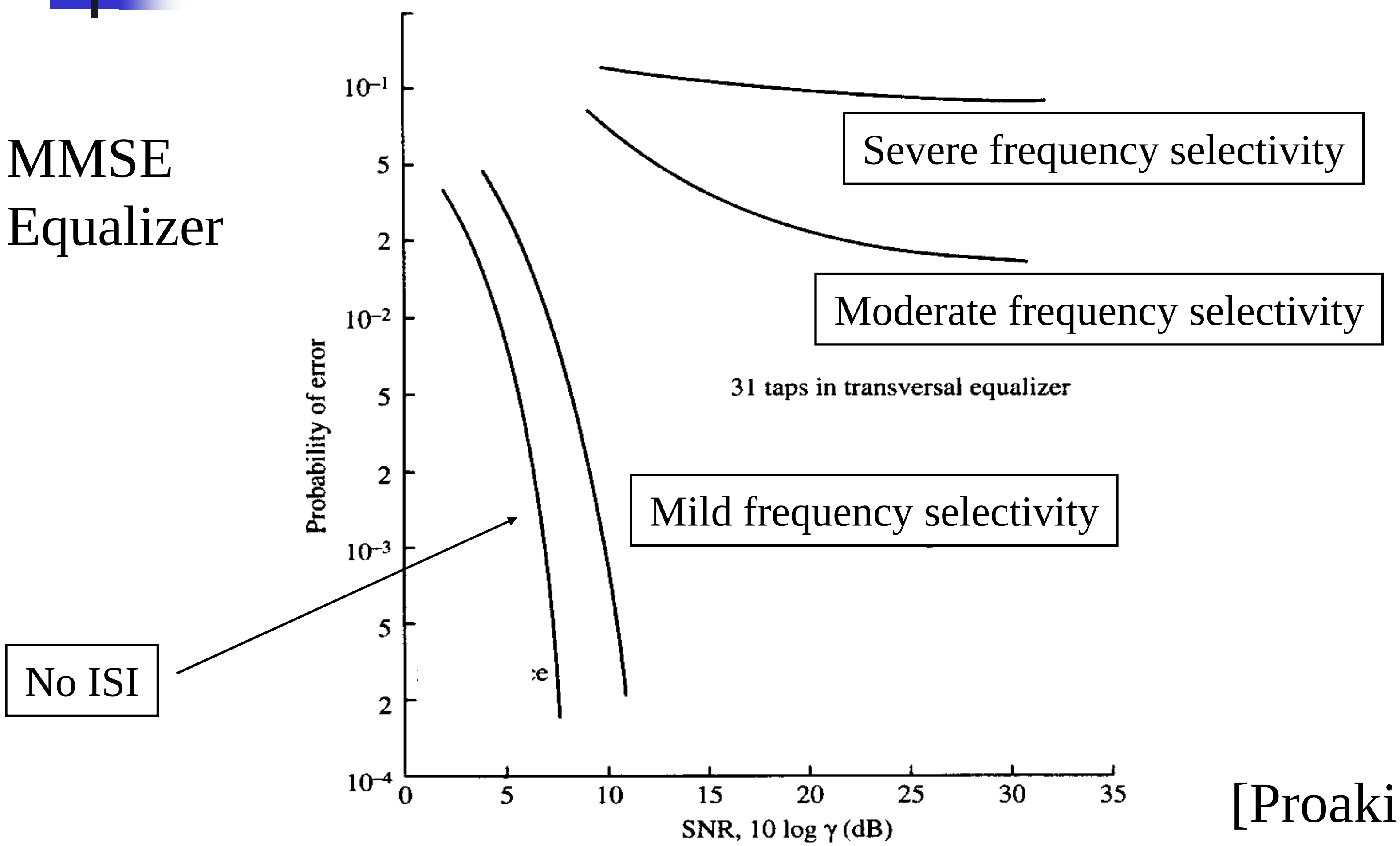
Severe frequency selectivity



[Proakis 2001]

Performance of Linear Transversal Filter Equalizer

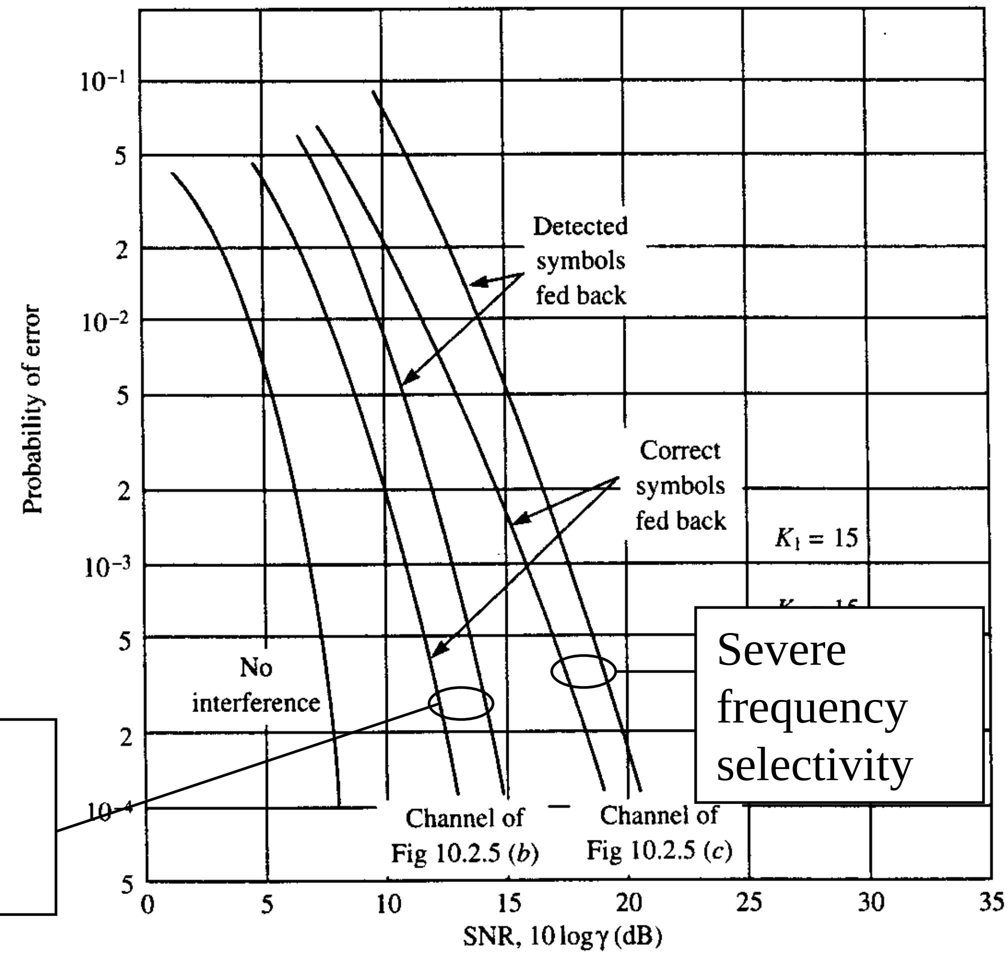
MMSE Equalizer



[Proakis 2001]

Performance of DFE with and without Error Propagation

Performance of decision-feedback equalizer with and without error propagation.



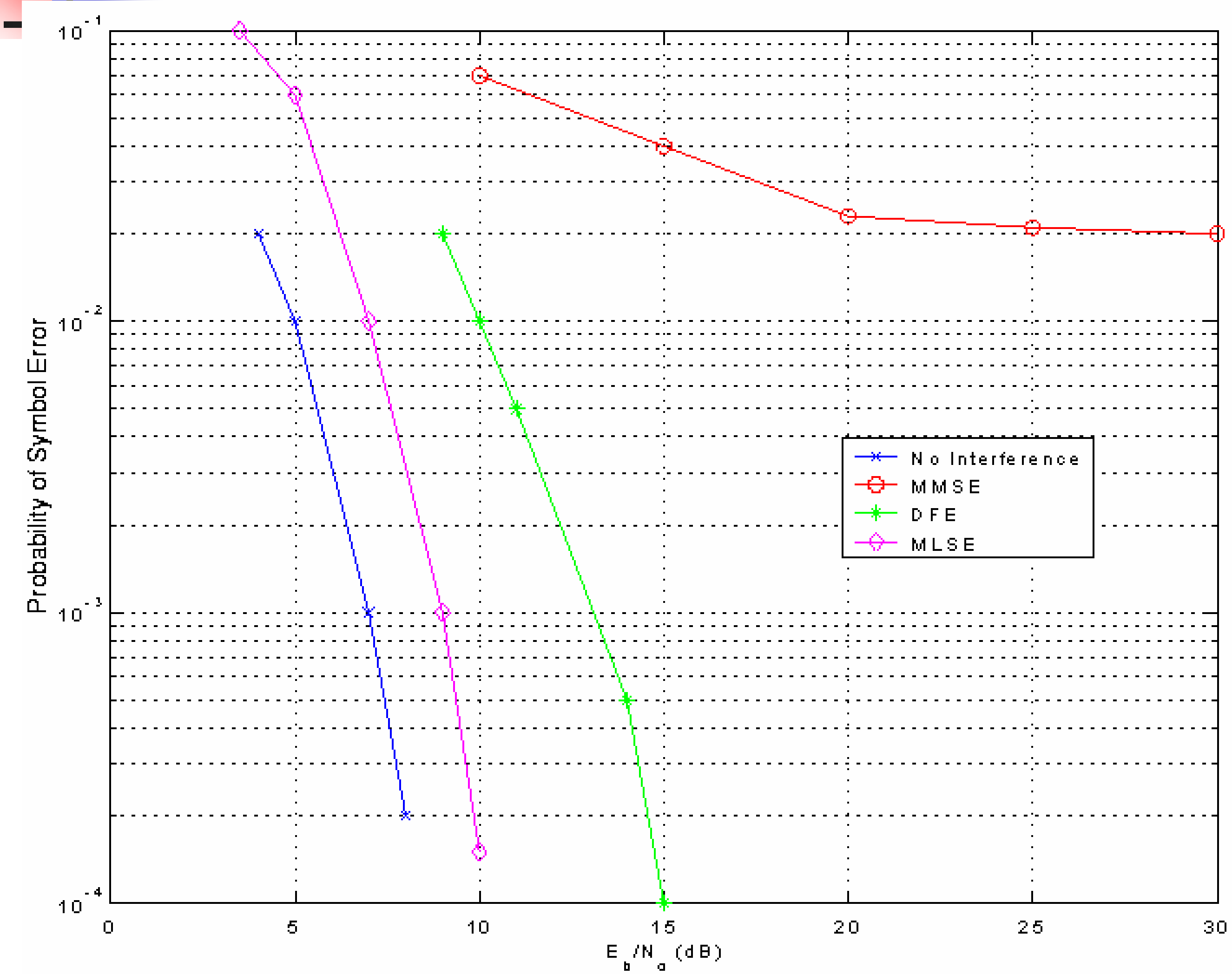
Approximately 2dB penalty due to error propagation

Severe frequency selectivity

Moderate frequency selectivity

[Proakis 2001]

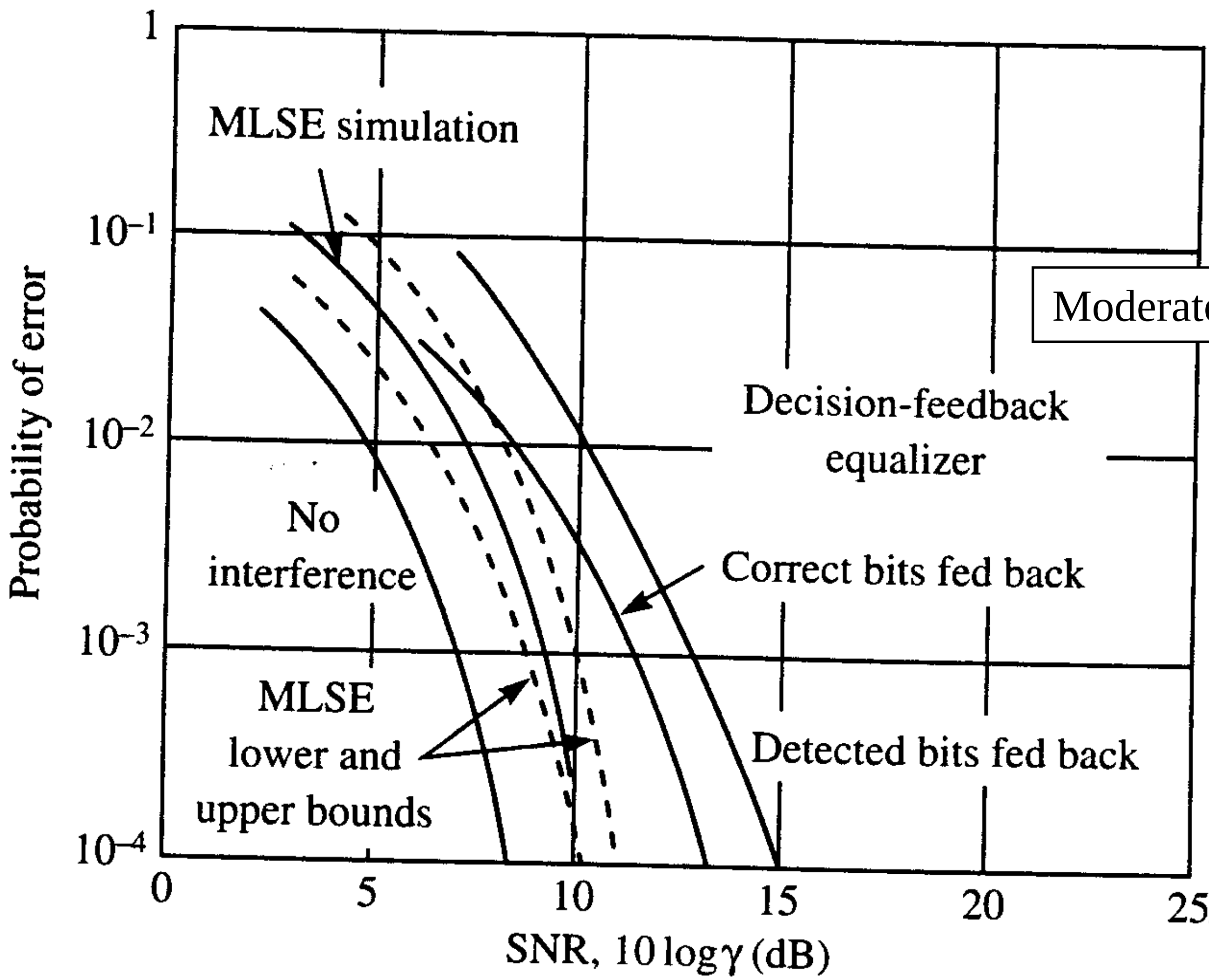
Performance of different Equalizers



Moderate Frequency Selectivity

DFE can provide substantial gains

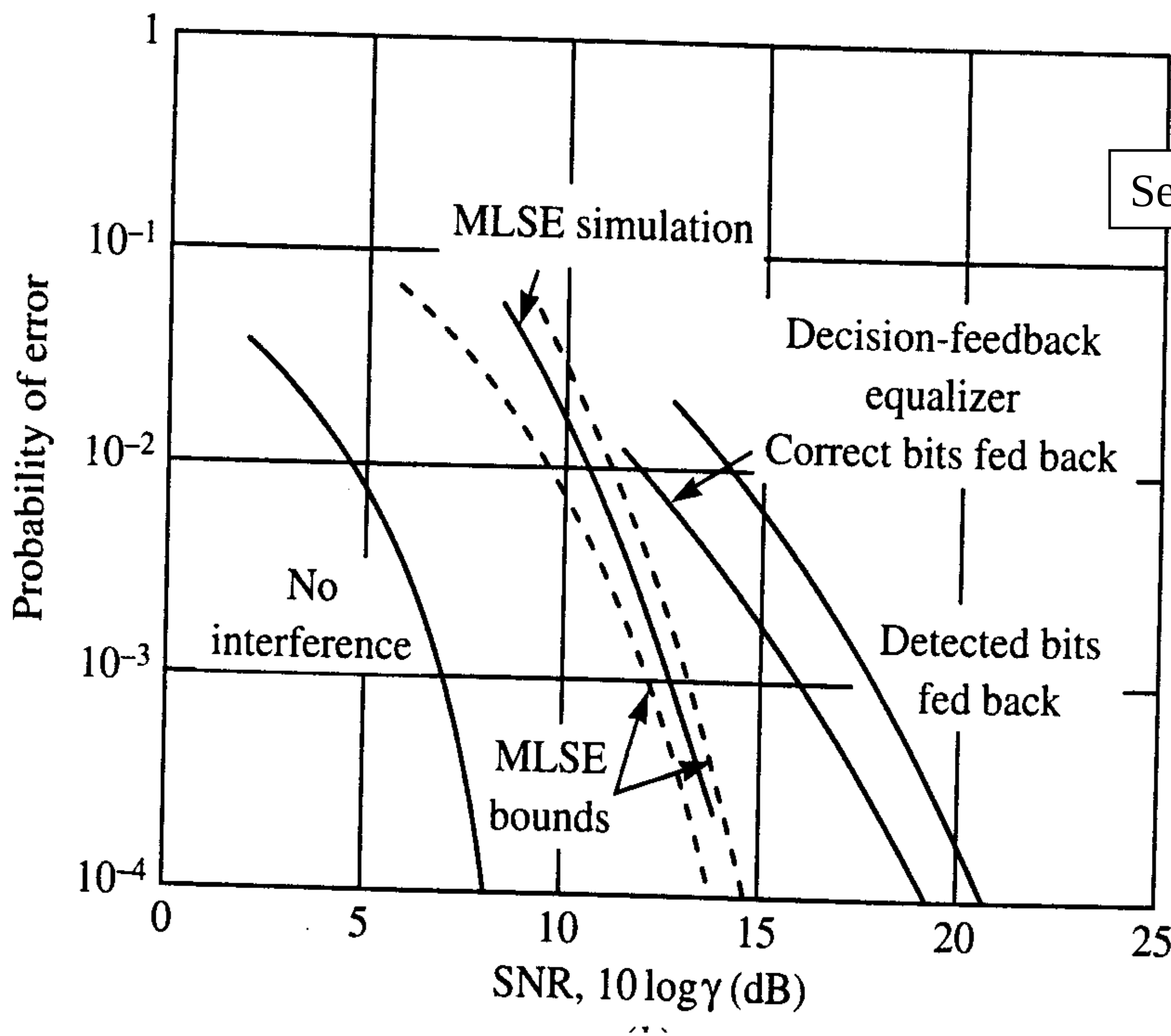
Performance of Decision Feedback and MLSE Structures



Moderate frequency selectivity

[Proakis 2001]

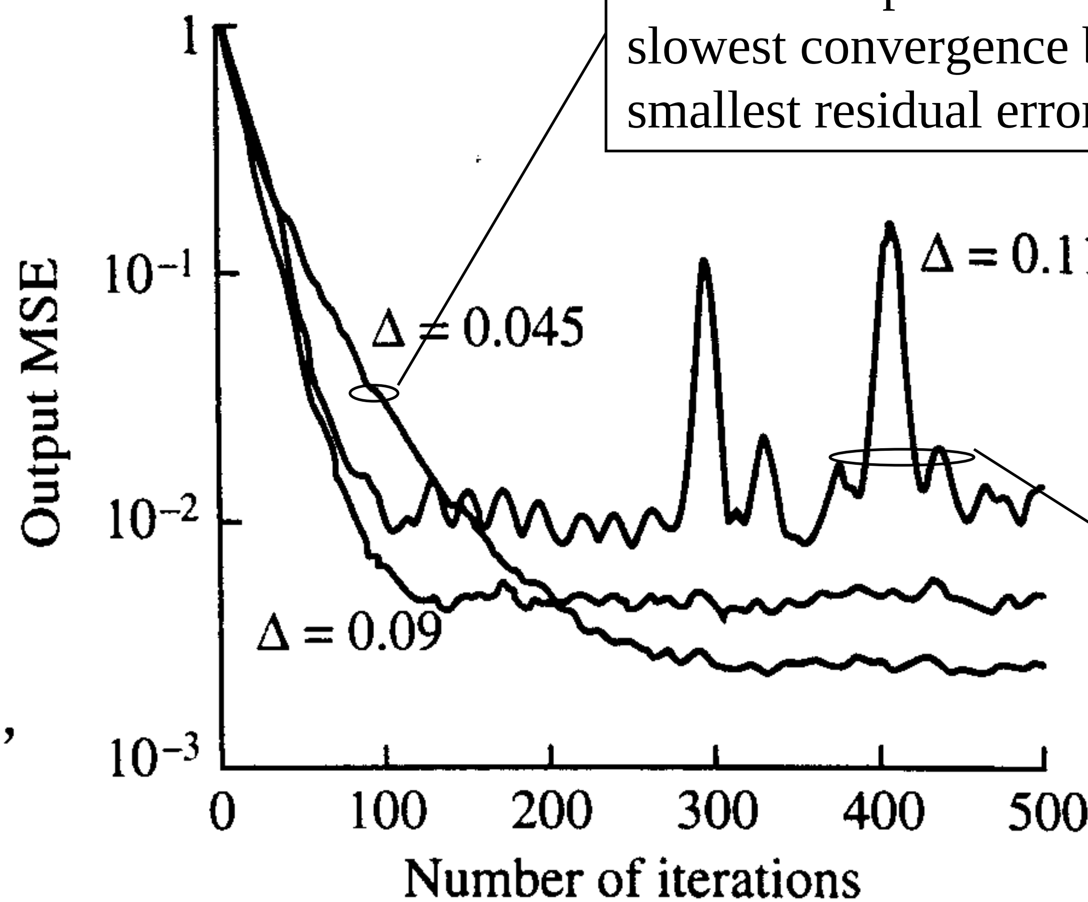
Performance of Decision Feedback and MLSE Structures



Severe frequency selectivity

[Proakis 2001]

Effect of Step Size on LMS Algorithm

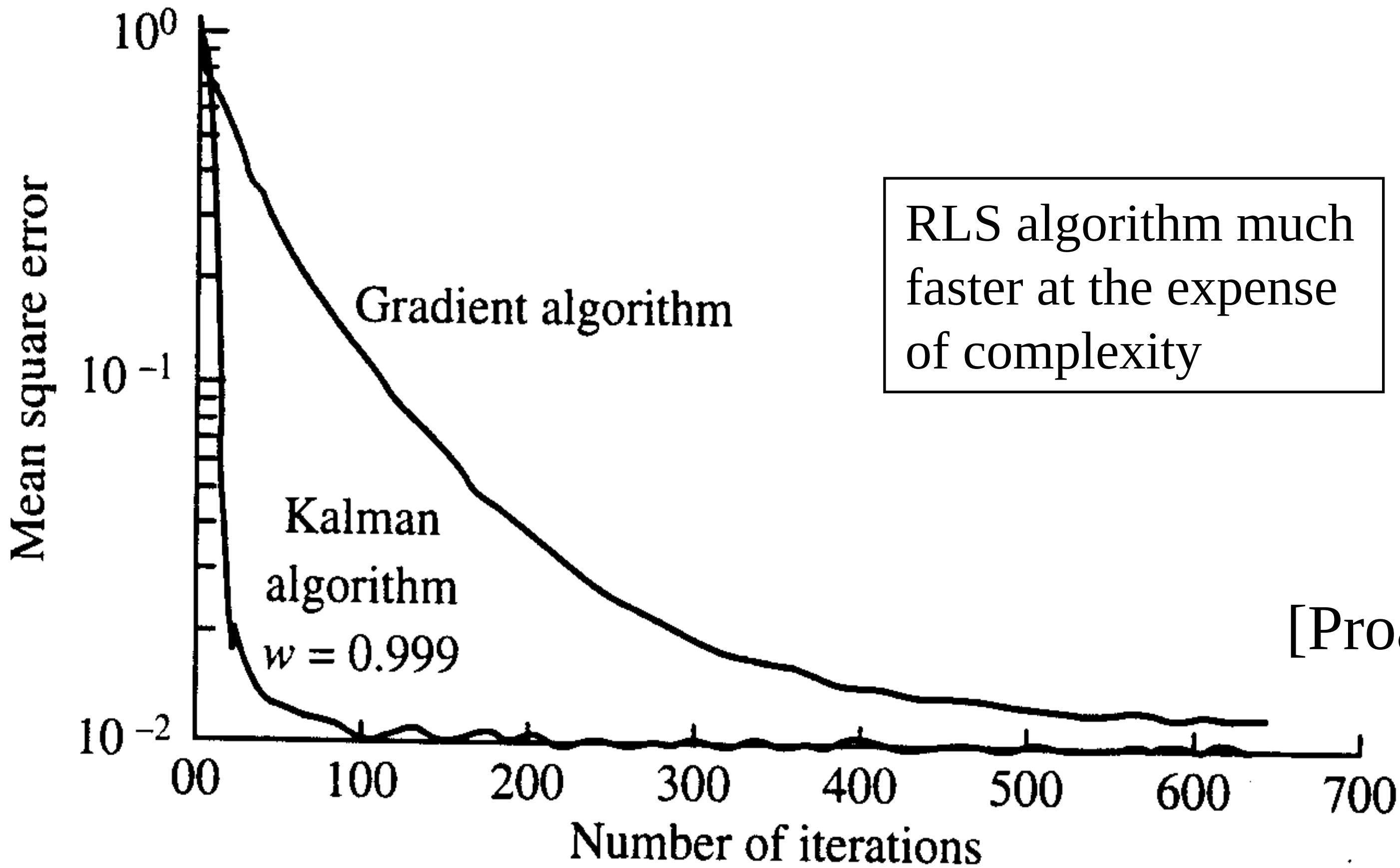


Smallest step size results in slowest convergence but smallest residual error

Large step size results in fast convergence but high residual error and undesirable instability

[Proakis 2001]

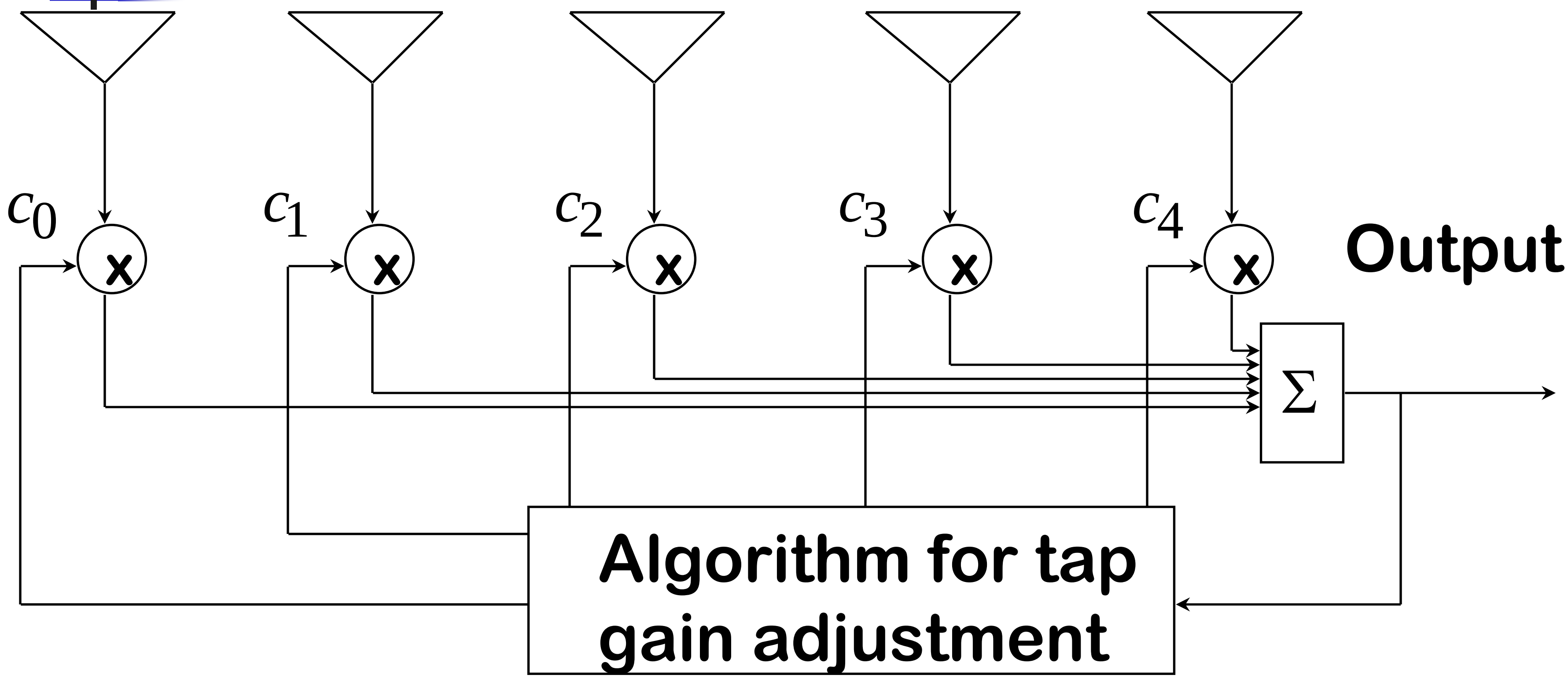
Convergence Rate of LMS (Gradient) and RLS (Kalman) Algorithms



RLS algorithm much faster at the expense of complexity

[Proakis 2001]

Adaptive Antennas

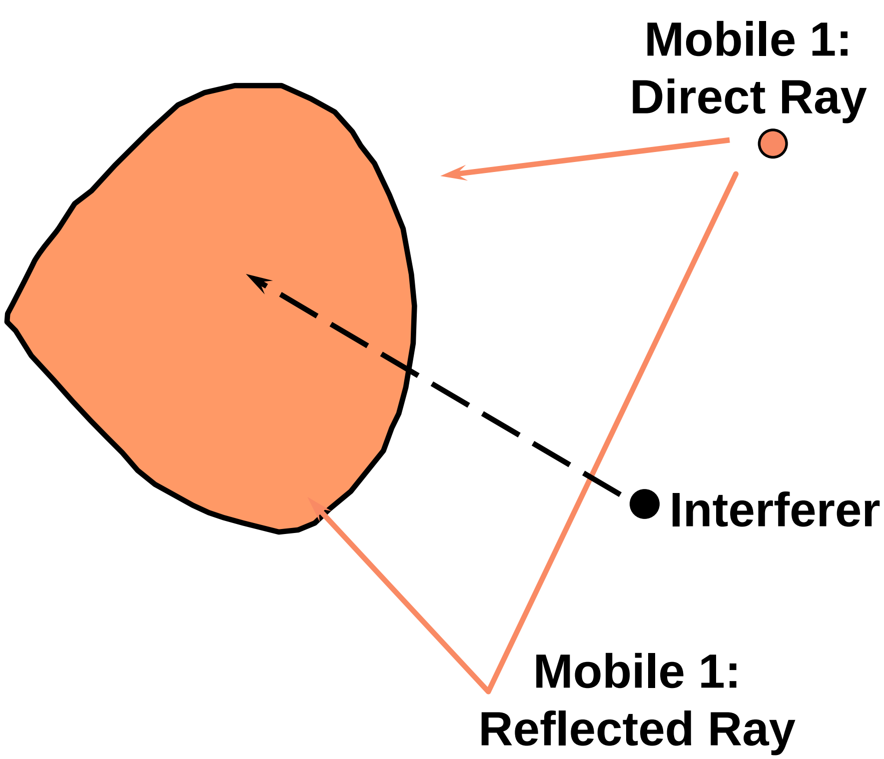


Analogy Between Adaptive Antennas and Equalizers

- Antenna arrays are formed by combining the signals from a number of antenna elements (frequently spaced one half wavelength apart).
- Coefficients which weight coefficients form a “spatial filter”
- Goal is usually to minimize MSE in resulting signal
- Adaptive algorithms such as LMS or RLS can be used to steer the resulting antenna beam
- Interference is analogous to ISI. We are now interested in the correlation across antennas rather than in time.

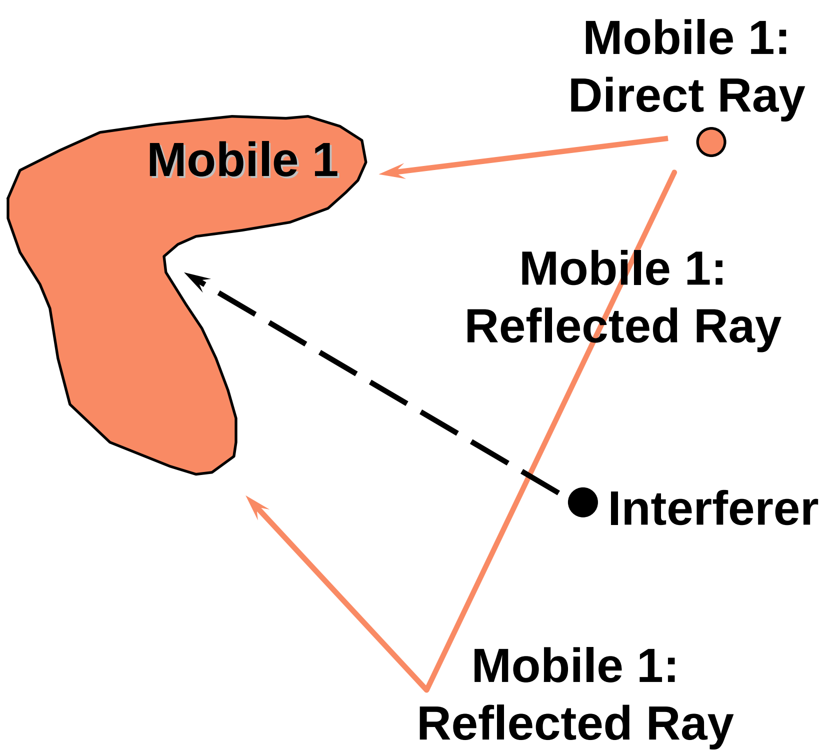
Adaptive Antenna Illustration

Conventional Receiver



Antenna spatial gain patterns are fixed.

Adaptive Antenna



Adaptively "Weight" and Combine Multiple Antenna Elements To Optimize Performance

Adaptive Antenna Array Beam-forming

Multiple patterns can be created at base station for multiple users.

- **4 signals positioned in a 120^0 sector.**
- **6 element linear array (λ/2 spacing).**
- **Optimal adaptive beamforming (MMSE)for each user.**