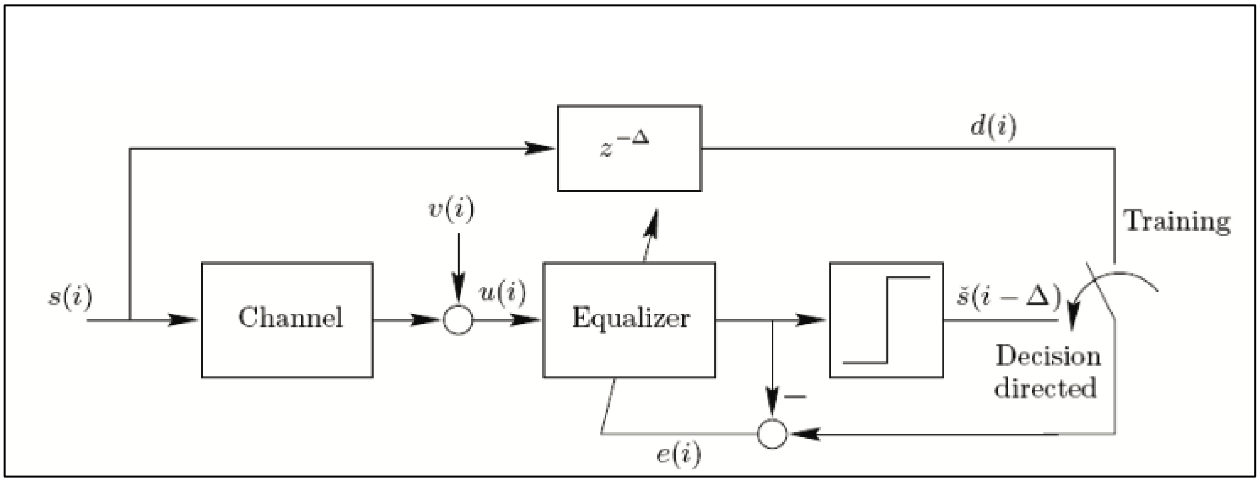
Adaptive Channel Equalization

In this project we examine the design of ***adaptive*** equalizers. We consider the channel:
$C(z) = 0.5 + 1.2z^{-1} + 1.5z^{-2} - z^{-3}$
and proceed to design an adaptive linear equalizer for it. The equalizer structure is shown below. Symbols $\{s(i)\}$ are transmitted through the channel and corrupted by additive complex-valued white noise $\{v(i)\}$. The received signal $\{u(i)\}$ is processed by the FIR equalizer to generate estimates $\{\hat{s}(i-\Delta)\}$, which are fed into a decision device. The equalizer possesses two modes of operation: a training mode during which a delayed replica of the input sequence is used as a reference sequence, and a decision-directed mode during which the output of the decision-device replaces the reference sequence. The input sequence $\{s(i)\}$ is chosen from a quadrature-amplitude modulation (QAM) constellation (e.g., 4-QAM, 16-QAM, 64-QAM, and 256-QAM)



An adaptive linear equalizer operating in two modes: training mode and decision-directed mode

a. Write a program that trains the adaptive filter with 500 symbols from a QPSK constellation, followed by decision-directed operation during 5000 symbols from a 16-QAM constellation. Choose the noise variance σ_v^2 in order to enforce an SNR level of 30 dB at the input of the equalizer. Note that symbols chosen from QAM constellations do not have unit variance. For this reason, the noise variance needs to be adjusted properly for different QAM orders in order to enforce the desired SNR level. Choose $\Delta = 15$ and equalizer length $L = 35$. Use ε-NLMS to train the equalizer with step-size $\mu = 0.4$ and $\varepsilon = 10^{-6}$. Plot the scatter diagrams of $\{s(i), u(i), \hat{s}(i-\Delta)\}$.

b. For the same setting as part (a), plot and compare the scatter diagrams that would result at the output of the equalizer if training is performed only for 150, 300, and 500 iterations. Repeat the simulations using LMS with $\mu = 0.001$.

c. Now assume the transmitted data are generated from a 256-QAM constellation rather than a 16-QAM constellation. Plot the scatter diagrams of the output of the equalizer, Then trained with ε-NLMS using 500 training symbols.

d. Generate symbol-error-rate (SER) curves versus signal-to-noise ratio (SNR) at the input of the equalizer for (4, 16, 64, 256)-QAM data. Let the SNR vary between 5 dB and 30 dB in increments of 1 dB.

e. Continue with SNR at 30 dB. Design a decision-feedback equalizer with L=10 feedforward taps and Q = 2 feedback taps. Use Δ = 7 and plot the resulting scatter diagram of the output of the equalizer. Repeat for L = 20, Q = 2 and Δ = 10. In both cases, choose the transmitted data from a 64-QAM constellation.

f. Generate SER curves versus SNR at the input of DFE for (4, 16, 64, 256)-QAM data. Let the SNR vary between 5 dB and 30 dB. Compare the performance of the DFE with that of the linear equalizer of part (d).

g. Load the file channel, which contains the impulse response sequence of a more challenging channel with spectral nulls. Set the SNR level at the input of equalizer to 40 dB and select a linear equalizer structure with 55 taps. Set also the delay at $\Delta = 30$. Train the equalizer using ϵ-NLMS for 2000 iterations before switching to decision-directed operation. Plot the resulting scatter diagram of the output of the equalizer. Now train it again using RLS for 100 iterations before switching to decision-directed operation, and plot the resulting scatter diagram. Compare both diagrams.