Channel Model in Magnetic Recording Channel (MRC)

Luoming Zhang

January 14, 2010

Description: Simulate the error pattern in the Longitude Magnetic Recording Channel

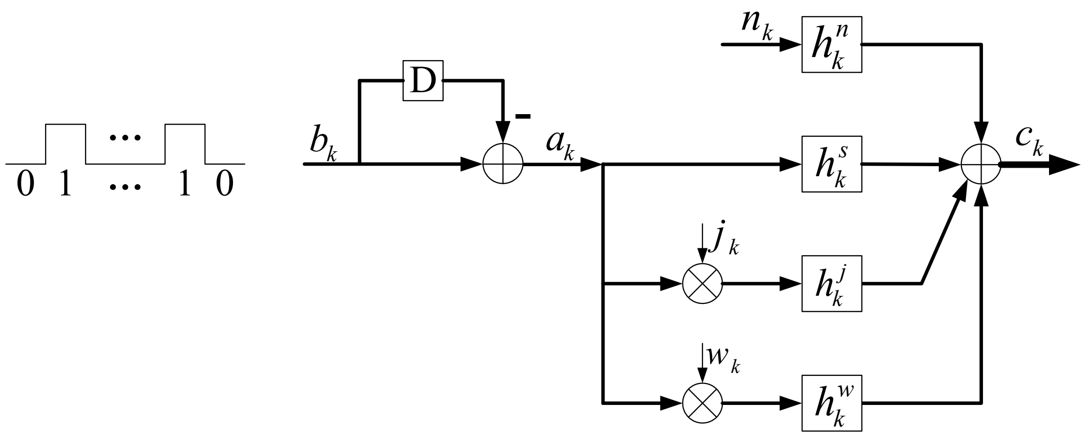

Figure 1: Discrete Channel Model for MRC

- **Detailed Description**
  As shown in Figure **??**, the data bits vector b_k sequentially passed the channel model so that we could get the simulated readback data vector c_k . The error pattern mainly is generated from 4 parts, the transition response h_k^s, the first-order position jitter response h_k^j, the first-order width jitter response h_k^w and the white noise response h_k^n. These four different responses are all in discrete form, which means that each response is discretized into corresponding vectors. In a word, by adding some sequences into the original data vector b_k, we could get the dicretized readback signal c_k.

Given the parameters SNRdB, recording density D_s, fraction of noise power due to media noise α, percentage of media noise power due to width variation λ, and the power of the transition E_t, we could get the variance of the white noise α_n, the variance of position jitter noise α_j, the variance of width variation α_w and also series of the corresponding response h_k^n, h_k^j, h_k^w and series of the transition response h_k^s. The following is the detailed equations.

$$N_\alpha = E_t 10^{-SNRdB/10} \tag{1}$$

$$N_0 = (1-\alpha)N_\alpha \qquad M = \alpha N_\alpha \tag{2}$$

$$M_j = (1-\lambda)M \qquad M_w = \lambda M \tag{3}$$

$$\alpha_n = \sqrt{\frac{N_0}{2}} \quad \alpha_j = \sqrt{\frac{M_j}{4E_t}} \quad \alpha_w = \sqrt{\frac{M_w}{4E_t}} \tag{4}$$

$$h_k^s = E_t \frac{D_s^2}{D_s^2 + k^2} \tag{5}$$

$$h_k^j = -E_t \frac{2D_s^3 k}{(D_s^2 + k^2)^2} \tag{6}$$

$$h_k^w = -E_t D_s^2 \frac{D_s^2 - k^2}{(D_s^2 + k^2)^2} \tag{7}$$

$$h_k^n = \sqrt{\frac{E_t D_s}{2\pi} \tanh\left(\frac{D_s \pi}{2}\right)} \frac{k + \frac{D_s}{2}}{(\frac{D_s}{2})^2 + k^2} \tag{8}$$

And usually, E_t is normalized to 1.

- **Specified**
  The sequences of the white noise, position jitter and width jitter are normally distributed pseudorandom sequences, which could be generated using gsl (http://www.gnu.org/software/gsl/).