

Storage Channel Modelling in Simulink

Ross Wilson, Member, IEEE, Scott O'Brien, Daniel Galaba, and John Leighton

Abstract — *A novel model of the magnetic/optical storage channel is reported. The model is implemented in the Matlab/Simulink block-diagram language, and is self-configuring to arbitrary user-specified partial response targets. All key continuous- and discrete-time channel components are implemented in a configurable architecture. A unified system description which merges the channel model with a behavioral representation of the analog processing electronics and interconnects preceding the channel is described, and results presented. Applications are in design of magnetic tape/disk, and optical storage systems.*

Index Terms — Behavioral model, equalization, identification, partial response, Simulink, storage.

I. INTRODUCTION

CONSUMER products are increasingly dependent on optical and magnetic storage, which today exhibit transfer rates approaching 2 Gb/s. These speeds tax the capabilities of electronics fabricated even in the latest silicon-germanium processes, and require transmission line interconnects between recording transducer, preamplifier, and channel.

Timely and cost-effective system design therefore requires reconfigurable models able simultaneously to predict consequences of such choices as that of partial response (PR) target; and to account for signal dispersion and bandwidth limitations of front-end electronics. Models are also valuable in the experimental phases of system design, as software receivers for signals digitized on a test stand.

System modelling is complicated by the customary use of Spice-based circuit simulators for transistor-level design of front-end electronics, and of behavioral modelling for digital signal processing within the channel itself.

This paper describes a flexible Simulink-based model of the recording system reproduce side. Parameters, including PR target, may be changed by editing a single line in a configuration file. The model is divided into Preamplifier and Channel subsections. Use of identification techniques solves the problem of importing Spice simulation results into a behavioral system simulation.

Simulink is a window-oriented visual block diagram language with an intuitive easy-to-learn graphical interface. Seamless linkage to Matlab brings Matlab's computational power to bear on model setup and postprocessing tasks. A wide variety of built-in functions, and Simulink's ability to handle hybrid discrete/continuous time systems and vector

signals, facilitates modelling of complex recording systems. Simulink-based [2] models can also be used as entry points in 'Chip-in-a-Day' LSI design flows [3] and in targeting programmable logic devices.

Block diagram languages provide visual representation of the algorithms under study, aiding implementation and understanding. However, graphic coding is not the best choice in all cases. Accordingly, the Viterbi detector, shift registers, and recording code decoder blocks of the model are written in C, to increase execution speed. These blocks are provided with wrappers and parameter/help menus, unifying them with the native Simulink environment.

II. CHANNEL MODEL

Fig. 1 shows the block diagram of the channel portion of the model. The signal received from the preamplifier passes through a variable gain amplifier (VGA), part of the gain control loop, to a fourth order continuous-time Butterworth filter (CTF) which provides anti-alias filtering. The CTF incorporates a real right-half-plane zero for high frequency boost, reducing need for boost in the digital equalizer and minimizing quantization noise enhancement.

The CTF output is converted to digital form by quantization and asynchronous sampling at period T_S to form $x[k]$. In this work, T_S is 6% higher than symbol interval T , to assure satisfaction of the sampling criterion despite minor variation of head-medium velocity.

The $x[k]$ are first equalized to the chosen target to obtain $y[k]$, then interpolated under control of the timing loop to yield properly timed symbol rate samples $w[k]$ used by the Viterbi detector, gain/timing regulation loops, and equalizer. The equalizer precedes the interpolator, to reduce timing loop transport delay and enhance stability.

A Viterbi detector performs maximum likelihood sequence detection on $w[k]$ and provides a serial bit stream to the framing and recording code decoding logic. A low latency early decision output $d[k] \in [-1, +1]$ is delivered to the equalizer and gain/timing loops.

The underlying timebase in all discrete-time blocks in Fig. 1 is T_S . Selective enabling of blocks nominally clocking at T assures that *mean* period of this value is maintained in these blocks.

The model supports arbitrary degree K PR [1] targets

$$P(D) = \psi_0 + \psi_1 D + \dots + \psi_K D^K, \quad (1)$$

The authors are with Agere Systems, Inc., 1230 Northland Drive, Mendota Heights, MN, 55120. E-mail rwilson8@agere.com.

$$y[k] = \mathbf{C}[k]^T \mathbf{X}[k]. \quad (3)$$

Adaptation minimizes the MSE, the mean square value of equalization error

$$\varepsilon[k] = \hat{w}[k] - w[k], \quad (4)$$

where $\hat{w}[k]$ denotes the symbol rate reference sequence obtained by convolving $d[k]$ with $p(kT)$,

$$\hat{w}[k] = d[k] \otimes p(kT). \quad (5)$$

Viterbi detector and reference reconstruction latency is compensated in (4) by delaying $w[k]$ to align it with $\hat{w}[k]$. Because equalizer FIR filter and tap weight integrators clock at T_S , symbol rate $\varepsilon[k]$ is resampled at T_S to obtain $\hat{\varepsilon}[k]$,

$$\hat{\varepsilon}[k] = \varepsilon[k] - (\varepsilon[k] - \varepsilon[k-1]) \cdot \frac{T_S}{T} \cdot \lambda[k]. \quad (6)$$

In (6) $\lambda[k]$ is phase indicator (20) and linear interpolation is satisfactory owing to robustness of the LMS adaptation.

Approximating $E\{\hat{\varepsilon}^2\} \approx \hat{\varepsilon}^2$, where $E\{\cdot\}$ indicates expectation and \tilde{w} is w upsampled to T_S , yields the MSE gradient,

$$\frac{\partial}{\partial \mathbf{W}} E\{\hat{\varepsilon}^2\} \approx \frac{\partial}{\partial \mathbf{W}} [\tilde{w}^2 + \mathbf{C}^T \mathbf{X} \mathbf{X}^T \mathbf{C} - 2\tilde{w} \mathbf{C}^T \mathbf{X}] = -2 \cdot \hat{\varepsilon} \cdot \mathbf{X}. \quad (7)$$

The optimum $\mathbf{C}[k]$ corresponding to minimum MSE is found iteratively by steepest descent solution with step size μ and gradient (7),

$$\begin{aligned} \mathbf{C}[k+1] &= \mathbf{C}[k] + 2\mu \cdot \hat{\varepsilon}[k] \cdot \Psi \cdot \mathbf{X}[k] \\ \Rightarrow \mathbf{C}(z) &= \frac{z^{-1}}{1-z^{-1}} \cdot 2\mu \cdot \hat{\varepsilon}(z) \cdot \Psi \cdot \mathbf{X}(z) \end{aligned} \quad (8)$$

Constraint matrix Ψ enforces linear relations between tapweights. A simple first-order constraint is imposed by

$$\Psi = [\psi_{ij}]: \quad \psi_{ij} = \begin{cases} 1: & i = j, c_{i=j} \text{ variable} \\ 1: & i \neq j, c_i + c_j = \text{const.} \\ 0: & \text{otherwise} \end{cases} \quad (9)$$

Specifically, in the present work, $N = 12$. Ψ is chosen to fix c_5 at unity and permit other tap weights to vary arbitrarily subject to the constraint $c_4 + c_6 = \text{const.}$ At preamble frequency this constraint removes two adaptation degrees of freedom. Fig. 3 illustrates equalizer tapweight convergence under conditions outlined in Section IV. Note tracking of the c_4 and c_6 tap weights.

Equalizer implementation in Simulink is depicted in Fig. 2; bold lines signify vector paths.

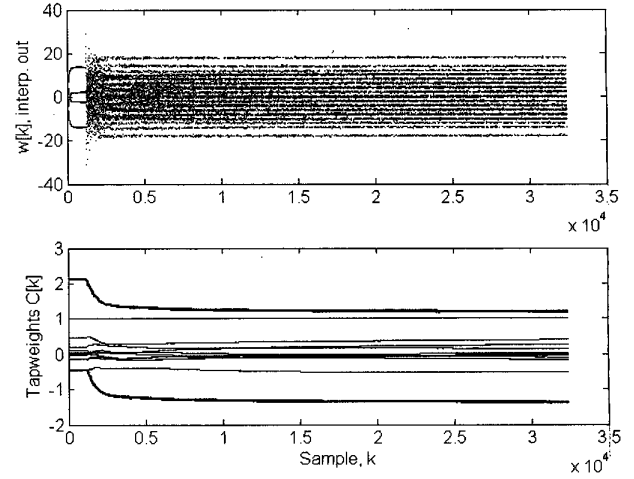


Fig. 3. Adaptive Equalizer Convergence for $P(D) = (1 - D^2) \cdot (5 + 4D + 2D^2)$. Taps 4, 6 bold.

B. Timing and Gain Control Loops

The decision directed timing and gain loops regulate the mean value of their respective error gradients to zero [7].

The amplitude control (gain G_G) loop applies an integral control law to the gain gradient $\gamma[k]$ to achieve zero steady-state error to step changes in input amplitude:

$$\begin{aligned} \Gamma[k] &= G_G \cdot \gamma[k-1] + \Gamma[k-1] \\ \Rightarrow \Gamma(z) &= G_G \cdot \gamma(z) \cdot \frac{z^{-1}}{1-z^{-1}} \end{aligned} \quad (10)$$

$\Gamma[k]$ sets VGA gain, effecting closed loop gain control.

Proportional-integral (gains G_P, G_I) compensation in the timing loop of timing gradient $\tau[k]$ eliminates phase error stemming from static frequency offsets:

$$\begin{aligned} I[k] &= G_I \cdot \tau[k-1] + I[k-1]; \quad P[k] = G_P \cdot \tau[k] \\ T[k] &= P[k] + I[k] \\ \Rightarrow T(z) &= \tau(z) \cdot \left(G_I \frac{z^{-1}}{1-z^{-1}} + G_P \right) \end{aligned} \quad (11)$$

$T[k]$ acts on the timing interpolator to close the timing loop.

Rapid loop acquisition is traded for reduced timing and gain variance during tracking by gear shifting gain coefficients. In (10) and (11), in acquisition mode, $\gamma = \gamma_{ACQ}$ and $\tau = \tau_{ACQ}$. Conversely, in tracking mode, $\gamma = \gamma_{TRACK}$ and $\tau = \tau_{TRACK}$.

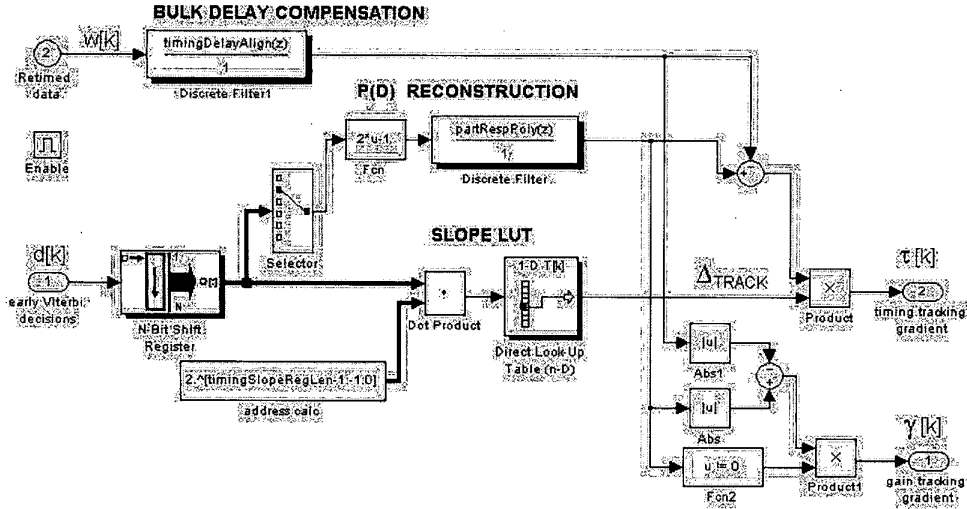


Fig. 4. Timing and gain tracking gradient computation.

Gains G_G , G_P , and G_I are each switched between acquisition- and tracking-mode values.

During gain and timing acquisition at the beginning of a record, presence of a known preamble pattern, $pre[k] \equiv [+1, 0, -1, 0, +1, \dots]$, is assumed. Here, 0 samples furnish timing information with maximum linear range, whereas ± 1 samples drive the gain acquisition process. Because a chosen PR target may not yield $pre[k]$ when convolved with the $4T$ -period $[+1, +1, -1, -1, +1, \dots]$ NRZ preamble, the equalized and retimed preamble sample sequence $w[k]$ is filtered by

$$w_{ACQ}[k] = b_0 w[k] + b_1 w[k-1], \quad (12)$$

prior to use in tracking gradient computation. Coefficients b_0 and b_1 are precalculated to yield $w_{ACQ}[k] = pre[k]$ in the preamble.

Timing (τ_{ACQ}) and gain (γ_{ACQ}) acquisition gradients are then calculated from samples partitioned by a counter into odd and even groups:

$$\tau_{ACQ}[k] = \Delta_{ACQ}[k] \cdot w_{ACQ}[k], \quad k=0, 2, 4, \dots \quad (13a)$$

$$\Delta_{ACQ}[k] \equiv (-1)^{k/2}$$

$$\gamma_{ACQ}[k] = C_{ACQREF} - |w_{ACQ}[k]|, \quad k=1, 3, 5, \dots \quad (13b)$$

In (13a) and (13b), Δ_{ACQ} enforces slope alternation on zero-crossing samples, and C_{ACQREF} is the gain setpoint. Once loop acquisition has been attained, gradients revert to tracking

mode, wherein no assumptions are made regarding $w[k]$ values.

Tracking gain gradient updates [9] are provided by nonzero reference samples according to

$$\gamma_{TRACK}[k] = \begin{cases} |\hat{w}[k]| - |w[k]| & : \hat{w}[k] \neq 0 \\ 0 & : \text{otherwise} \end{cases} \quad (14)$$

Tracking timing gradient is calculated [8, 9] by multiplying $\varepsilon[k]$ (4) by a slope-related term Δ_{TRACK} :

$$\tau_{TRACK}[k] = (\hat{w}[k] - w[k]) \cdot \Delta_{TRACK}[k] \quad (15)$$

To determine Δ_{TRACK} , the continuous-time derivative of pulse response (2) is required,

$$p'(t) = \sum_{k=0}^K \frac{\psi_k}{t-kT} \left[\cos\left(\pi \frac{t-kT}{T}\right) - \text{sinc}\left(\frac{t-kT}{T}\right) \right]. \quad (16)$$

Convolution (16) and $d[k]$ yields the ideal small-signal slopes of $w[k]$ at sample instants and at $\pm T/2$ displacements,

$$\begin{aligned} m_-[k] &= p'(kT - T/2) \otimes d[k] \\ m_0[k] &= p'(kT) \otimes d[k] \\ m_+[k] &= p'(kT + T/2) \otimes d[k] \end{aligned} \quad (17)$$

Amplitude measures associated with $w[k]$ are similarly defined:

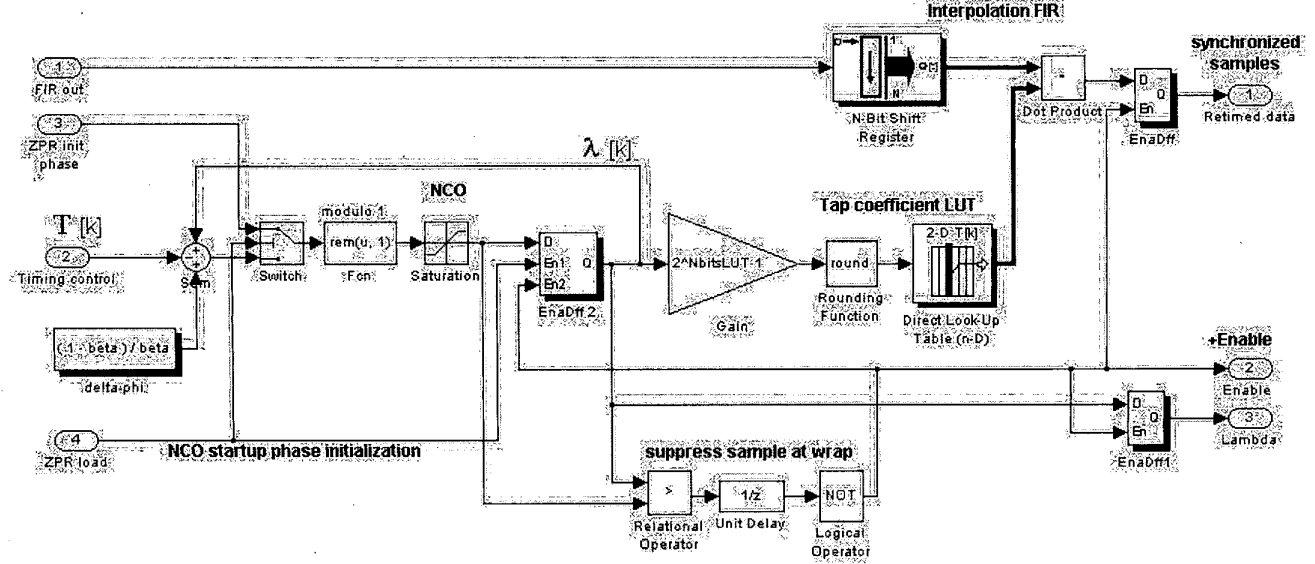


Fig. 5. Timing interpolator and NCO.

$$\mathbf{X} = \begin{bmatrix} p(N_S - (-M_1)) & p(N_S - (-M_1 + 1)) & \dots \\ p((N_S + 1) - (-M_1)) & p((N_S + 1) - (-M_1 + 1)) & \dots \\ \vdots & \vdots & \vdots \\ p(N_E - (-M_1 + 1)) & p(N_E - (-M_1 + 1)) & \dots \\ \dots & p(N_S - (M_2 - 1)) & p(N_S - M_2) \\ \dots & p((N_S + 1) - (M_2 - 1)) & p((N_S + 1) - M_2) \\ \vdots & \vdots & \vdots \\ \dots & p(N_E - (M_2 - 1)) & p(N_E - M_2) \end{bmatrix} \quad (25c)$$

The optimum value of \mathbf{H} is found by differentiating (24) and equating the result to zero:

$$\frac{\partial v}{\partial \mathbf{H}} = \mathbf{0} = -2\mathbf{X}^T \mathbf{P} + 2\mathbf{X}^T \mathbf{X} \mathbf{H} \Rightarrow \mathbf{H} = [\mathbf{X}^T \mathbf{X}]^{-1} \mathbf{X}^T \mathbf{P}. \quad (26)$$

\mathbf{H} is computed offline for each value of λ and all tap weight sets are stored in the interpolator tap weight LUT. At the optimum (26), the mean square fit error is

$$MSFE = \mathbf{P}^T \mathbf{P} - \mathbf{P}^T \mathbf{X} \mathbf{H}. \quad (27)$$

Interpolator length is chosen as the shortest value yielding acceptable MSFE (27). Indicating dependence on λ by writing $h[m, \lambda]$, the interpolator output $w[l]$ for input $y[l]$ is therefore

$$w[l] = \sum_{m=-M_1}^{M_2} y[m-l] \cdot h[m, \lambda]. \quad (28)$$

D. Configurable Viterbi Detector

The Viterbi detector [14] traces a path at symbol rate through an $L = 2^K$ -state trellis under guidance of $w[k]$. Trellis states $S_j[k]$ at symbol instants k are numbered consecutively $j = 0, 1, \dots, L-1$. Each state has two incident branches stemming from prior states $S_p[k-1]$ and $S_q[k-1]$, $p, q \in [0, 1, \dots, L-1]$; and an associated path metric value. Branch weights BW_{pj} , BW_{qj} and underlying $m = \pm 1$ message values are associated with the two branches. The trellis for $P(D) = (1 - D^2) \cdot (5 + 4D + 2D^2)$ appears for example in Fig. 6.

Prior Viterbi detector designs have employed a hard-wired trellis structure. Instead, the trellis is here flexibly defined by precomputed $L \times 2$ prior-state Φ and branch weight \mathbf{B} matrices

$$\Phi = [\phi_{ji}]: \begin{cases} \phi_{j0} = S_p[k], & S_p[k-1] \xrightarrow{m=-1} S_j[k] \\ \phi_{j1} = S_q[k], & S_q[k-1] \xrightarrow{m=+1} S_j[k] \end{cases} \quad (29)$$

$$\mathbf{B} = [\beta_{ji}]: \begin{cases} \beta_{j0} = BW_{pj}, & S_p[k-1] \xrightarrow{BW_{pj}} S_j[k] \\ \beta_{j1} = BW_{qj}, & S_q[k-1] \xrightarrow{BW_{qj}} S_j[k] \end{cases} \quad (30)$$

Add-compare-select operators at each state extend the detector trajectory from one of two candidate previous states S_p , S_q forward to new state S_j through that branch yielding the minimum total path Euclidean distance at S_j :

$$sel_br = \begin{cases} S_p \rightarrow S_j : |w[k]-BW_{pj}|^2 + S_p[k-1] \\ < |w[k]-BW_{qj}|^2 + S_q[k-1] \\ S_q \rightarrow S_j : & otherwise \end{cases} \quad (31)$$

The path metric associated with each current state is then updated with the sum of path metric associated with the selected prior state, and the metric of the chosen branch:

$$S_j[k] = \begin{cases} S_p[k-1] + |w[k]-BW_{pj}|^2 : sel_br = S_p \rightarrow S_j \\ S_q[k-1] + |w[k]-BW_{qj}|^2 : & otherwise \end{cases} \quad (32)$$

A path memory using a simple register-exchange scheme stores results of branch choices. An early output $d[k]$ at shallow decoding depth of 4 reduces transport delay in the gradient and equalizer adaptation computations. Both early and normal path memory outputs originate at the trellis node having lowest-valued path metric.

The pseudo-code in Fig. 7 summarizes the C implementation of the Viterbi detector. Trellis definition utilizes indirect addressing through elements of Φ .

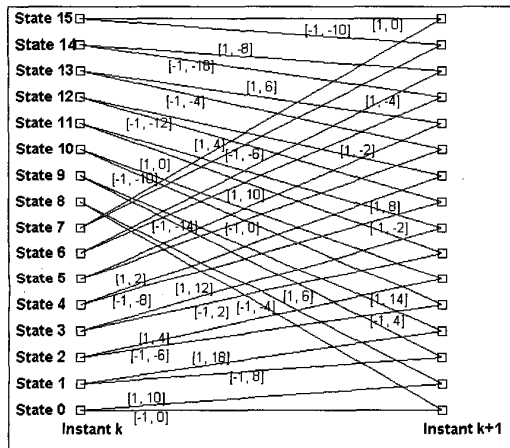


Fig. 6. Viterbi detector trellis diagram for $(1-D^2) \cdot (5+4D+2D^2)$. Branch labelling: [Channel Input Bit (± 1), Branch Metric].

E. Framing and Decoding

The framing and decoding icon in Fig. 1 accepts the serial output of the Viterbi detector, detects the end-of-preamble synchronization character within an expected-time-of-arrival window using a polarity-insensitive correlation scheme; and deserializes framed data into codewords of the fixed-blocklength recording code. After decoding and optional descrambling, message frames are sent to the bit error rate measurement routine via a *To Workspace* block gated at the recording code frame rate.

A. STATE UPDATE

```

for (jk = 0; jk < L; jk++) { // save old states
  oldViterbiStates[jk] = viterbiStates[jk]
  for (kk = 0; kk < normalPathMemoryLength; kk++) {
    oldPmStates[jk, kk] = pmStates[jk, kk]
  }
  for (jk = 0; jk < L; jk++) { // iterate over N states
    bm0 = |w[jk] - B[jk, 0]|^2 // compute branch metrics
    bm1 = |w[jk] - B[jk, 1]|^2
    bm0 = bm0 + oldViterbiStates[Phi[jk, 0]] // ADD
    bm1 = bm1 + oldViterbiStates[Phi[jk, 1]]
    if (bm0 < bm1) { // COMPARE - SELECT
      viterbiStates[jk] = bm0; acsPointer = 0
    } else {
      viterbiStates[jk] = bm1; acsPointer = 1
    }
  }
  if (acsPointer == 0) { // register exchange path memory
    pmStates[jk, 0] = 0;
    for (kk = 1; kk < normalPathMemoryLength; kk++) {
      pmStates[jk, kk] = oldPmStates[Phi[jk, 0], kk - 1]
    }
  } else {
    pmStates[jk, 0] = 1;
    for (kk = 1; kk < normalPathMemoryLength; kk++) {
      pmStates[jk, kk] = oldPmStates[Phi[jk, 1], kk - 1]
    }
  }
}

```

B. DETECTOR OUTPUTS

```

a = minIndex{viterbiStates} // index minimum path metric state
normalViterbiOut = viterbiStates[a, normalPathMemoryLength - 1]
earlyViterbiOut = viterbiStates[a, earlyPathMemoryLength - 1]

```

Fig. 7. Configurable Viterbi detector algorithm.

III. PREAMPLIFIER MODEL

There are two concerns in simulating the cascade of a device-level preamplifier model with the behavioral channel model. First, use of a mixed-mode simulator greatly increases simulation time, as the device-level circuits must be iterated at each timestep. The choice of simulation tools is also greatly narrowed. A second, more significant, problem is that Spice-like simulators generally suppress element internal noise sources when performing time domain simulation.

To overcome these deficiencies, the preamplifier is modelled behaviorally as shown in Fig. 8; all blocks are clocked at T_{OVS} , a submultiple of symbol time T . The playback signal source is furnished by the *From Workspace* block.

In the present work, the playback transducer is a magnetoresistive head which is modelled in Spice as a series resistor and signal voltage source at the preamplifier input. The head, preamplifier, and recording channel are interconnected by lossy, dispersive, coupled transmission lines described by Verilog-A modules [16] imported into the Spice simulator.

Spice small-signal ac simulation yields the complex response relating voltage at the end of the preamplifier-to-channel interconnect, to head internal voltage. Identification [15] of these data leads to a rational approximation STF , a

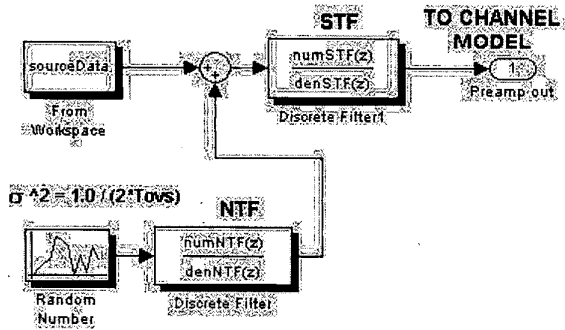


Fig. 8. Preamp model.

behavioral description of the preamplifier signal transfer function.

Similarly, Spice noise analysis furnishes the noise power spectral density (PSD) referred to the head’s internal signal voltage source. Identification of the noise root-PSD result yields *NTF*, the preamplifier noise shaping transfer function.

Identification results for a commercially available preamplifier fabricated in 10 GHz BiCMOS are presented in Fig. 9.

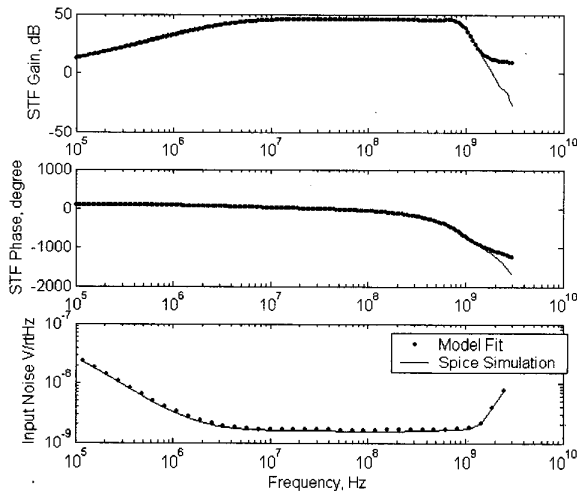


Fig. 9. Identification of STF (order = 8) and NTF (order = 5) for commercially available preamplifier in 10 GHz BiCMOS.

A. Signal Transfer Function Identification

The problem is one of fitting in least squares sense a stable rational *M*-th-order transfer function in $\hat{z} \equiv e^{j\omega T_{OVS}}$,

$$STF(\hat{z}, \mathbf{a}, \mathbf{b}) \equiv \frac{P(\hat{z}, \mathbf{a})}{Q(\hat{z}, \mathbf{b})} = \frac{a_M \hat{z}^{-M} + \dots + a_1 \hat{z}^{-1} + a_0}{b_M \hat{z}^{-M} + \dots + b_1 \hat{z}^{-1} + 1}, \quad (33)$$

to the set of tabulated, complex signal transfer function data points $R(\omega_n)$, $n = 0, 1, \dots, N-1$ produced by the Spice circuit

simulator. The underlying transfer function will be transcendental if distributed interconnects are present. The minimum of the nonlinear cost function, with real weighting function η_n , is sought:

$$\min_{\mathbf{a}, \mathbf{b}} \sum_{n=1}^N \left| \eta_n STF(e^{j\omega_n T_{OVS}}, \mathbf{a}, \mathbf{b}) - \eta_n R(\omega_n) \right|^2. \quad (34)$$

Equation (34) is solved using Levenberg-Marquardt nonlinear optimization, starting from an initial point obtained by solving the linear frequency weighted least squares problem

$$\min_{\mathbf{a}, \mathbf{b}} \sum_{n=1}^N \left| \eta_n P(e^{j\omega_n T_{OVS}}, \mathbf{a}) - \eta_n R(\omega_n) \cdot Q(e^{j\omega_n T_{OVS}}, \mathbf{b}) \right|^2. \quad (35)$$

An alternative identification scheme useful over multi-decade frequency ranges recasts (33) in the continuous frequency variable $j\omega$, solves (34), (35) in an orthogonal basis [17] to improve conditioning, then transforms the result to discrete time.

B. Noise Transfer Function Identification

Spice circuit noise analysis yields a pointwise head signal source referred noise PSD $N_I(\omega_n)$ V^2/Hz , reflecting contributions of all circuit devices, including transducer thermal noise. The behavioral model produces colored noise $N_I(\hat{z})$ approximating the signal-source-referred noise of the actual preamplifier, by filtering discrete-time white noise $N(\hat{z})$ with stable transfer function $NTF(\hat{z})$ [1]:

$$N_I(\hat{z}) = |NTF(\hat{z})|^2 \cdot N(\hat{z}). \quad (36)$$

$NTF(\hat{z})$ is obtained by least squares fit to the Spice noise root-PSD results. For $N(\hat{z}) = 1.0$,

$$|NTF(e^{j\omega_n T_{OVS}})| \approx \sqrt{N_I(\omega_n)}. \quad (37)$$

Variance $\sigma^2 = 1/2T_{OVS}$ properly scales the output of the *Random Number* block in Fig. 9 to the dc to Nyquist band.

Because phase of $NTF(e^{j\omega_n T_{OVS}})$ is immaterial in (37), phase can be chosen freely. Constraining the phase to, say, a minimum-phase criterion, may penalize fit quality. Therefore, a new iterative fitting technique, outlined in Fig. 11, is adopted in which the phase of the fit result on an iteration is propagated as the objective fit phase on the next iteration. Using a least squares methodology similar to that employed to fit $STF(\hat{z})$, this procedure converges rapidly to the phase associated with the desired magnitude response.

```

Given (real)root-PSD  $\sqrt{N_f(\omega_n)}$  signal source referred noise
defined over a set of  $N$  frequency points  $n = 0,1,\dots,N-1$ :

Set  $mag_n = \sqrt{N_f(\omega_n)}$ ;  $\varphi_n = 0$ 
for (iter = 0; iter < L; iter++) {
  1. Compute  $P(\omega_n) = mag_n \cdot e^{j\varphi_n}$ 
  2. Least-squares fit transfer function  $H(e^{j\omega T_{OVS}})$ 
to Re, Im parts of  $P(\omega_n)$ :
 $H(e^{j\omega T_{OVS}}) = fit\{P(\omega_n)\}$ 
  3. Compute phase of  $H(e^{j\omega T_{OVS}})$ :

$$\varphi_n = \text{unwrap} \left\{ \arctan \left( \frac{\text{Im}\{H(e^{j\omega T_{OVS}})\}}{\text{Re}\{H(e^{j\omega T_{OVS}})\}} \right) \right\}$$

}
NTF( $\hat{z}$ ) =  $H(e^{j\omega T_{OVS}})$ 

```

Fig. 10. NTF fitting algorithm. $L = 8$ iterations for results of Fig. 9.

IV. RESULTS

Three longitudinal magnetic recording targets are compared for error rate performance using cascaded preamplifier and channel models: EEPR-IV $(1-D) \cdot (1+D)^3$; MEEPR-IV [4] $(1-D^2) \cdot (5+4D+2D^2)$; and a new high-density 32-state target [5] $(1-D) \cdot (1+D)^2 \cdot (2+D+D^2)$.

The recording code is a proprietary rate 96/98 design and user data bit rate $f_b = 1.0$ Gb/s. The head playback signal is created by $6\times$ upsampling a symbol rate $(1/T)$ NRZ binary sequence $\in [-1,+1]$ comprising $4T$ -period preamble, synchronization mark, and coded pseudo-random data; and convolving the result with dipulse response $r[k] = s(kT_{OVS}) - s(kT_{OVS} - T)$. $s(t) = V_{BP} / [1 + (2t/PW50)^2]$ is the Lorentzian isolated transition response of half-amplitude width $PW50 = D_U / f_b$, base-peak amplitude V_{BP} , normalized recording density $D_U = 3.0$; and $T_{OVS} = T/6$.

Signal-to-noise ratio $SNR \equiv 20 \cdot \log_{10}(V_{BP} / \sigma_{Nyquist})$ dB, where $\sigma_{Nyquist}$ is the dc-Nyquist RMS noise referred to the head playback signal.

Optimum equalizer tap weight difference and CTF cutoff frequency for each target are determined by search for lowest converged equalizer MSE. Fig. 11 shows the pronounced minimum MSE obtained for an MEEPR-IV target at 26 dB SNR.

Bit error rate plots for an infinite bandwidth preamplifier with additive white Gaussian noise are reported in Fig. 12. Error rate performance of the EEPR-IV target at $1e-5$ BER is 1 dB inferior to that of MEEPR-IV.

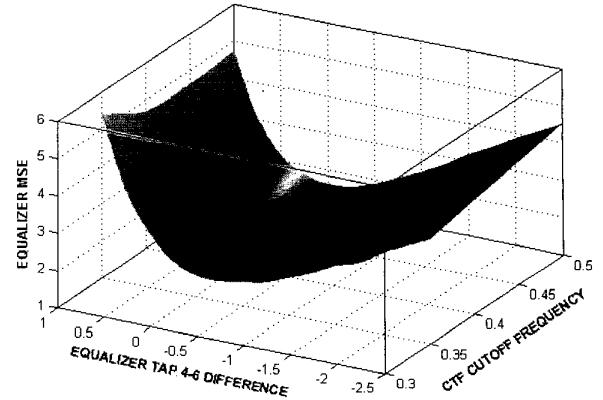


Fig. 11. Equalizer MSE surface plotted against tap 4-6 difference and CTF cutoff, at 26 dB SNR; for commercially available preamplifier; and MEEPR-IV target.

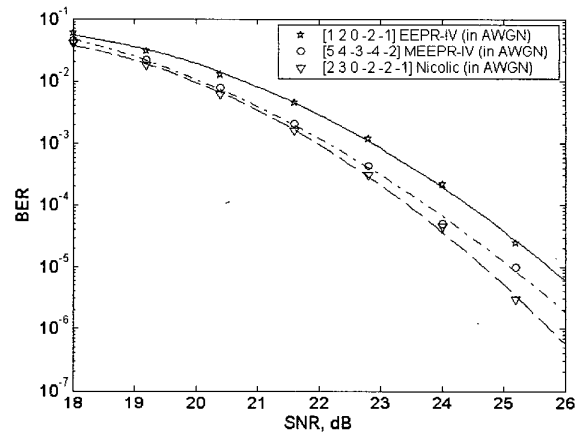


Fig. 12. BER versus channel input SNR for three target responses, in additive white Gaussian noise.

The $(1-D) \cdot (1+D)^2 \cdot (2+D+D^2)$ target secures an additional 0.3 dB improvement over the 16-state MEEPR-IV target at cost of a 32-state Viterbi detector.

For comparison, Fig. 13 shows error rate plots for a system utilizing a commercially available preamplifier having the characteristics shown in Fig. 9. Performance impairment is minimal for this design.

Inclusion of transition noise [18] in the input data record, and consideration of other modulation codes, may modify these findings.

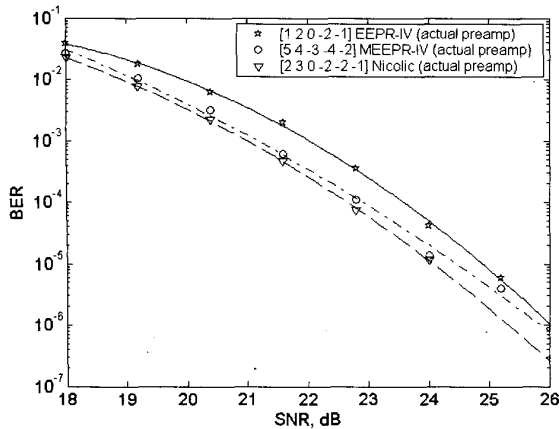


Fig. 13. BER versus channel input SNR for three target responses, for commercially available preamplifier.

V. CONCLUSIONS

A flexible model of a complex hybrid continuous/discrete-time data storage system has been presented. The dynamics of front-end preamplifier electronics described at the device level have been included. Execution speed on a 450 MHz Pentium-III computer is ~ 50 seconds per 4096 byte record transferred. Compilation reduces this time by a factor of 2.5.

The present work is a flexible base for investigation of other receiver architectures, including iterative decoding and noise-whitening post-processing [19, 20]. The described identification methods can be extended using Volterra series approach to model mildly nonlinear front-end circuits.

REFERENCES

- [1] E.A. Lee and D.G. Messerschmitt, *Digital Communication, Second Edition*. Norwell, MA: Kluwer Academic Publishers, 1994.
- [2] *Simulink 5: Design and Simulate Continuous- and Discrete-Time Systems*. Natick, MA: The Mathworks, Inc., 2002.
- [3] W. Davis, N. Zhang, K. Camera, F. Chen, D. Marković, T. Smilkstein, M. Ammer, E. Yeo, S. Augsburg, B. Nicollić, and R. Brodersen, "A design environment for high-throughput, low-power dedicated signal processing systems," *IEEE J. Solid State Circuits*, vol. 35, pp. 420-431, March 2002.
- [4] T. Nishiya, K. Tsukano, T. Hirai, T. Nara, and S. Mita, "Turbo-EEPRML: an EEPR4 channel with an error-correcting post-processor designed for 16/17 rate quasi-MTR code," *Globecom 1998 Conf. Record*, vol. 5, pp. 2706-2711, 1998.
- [5] B. Nicollić, M. Leung, and L. Fu, "Rate 8/9 sliding block distance-enhancing code with stationary detector," *IEEE Trans. Magn.*, vol. 37, pp. 1168-1174, May 2001.
- [6] S. Gopalaswamy and P. McEwen, "Read channel issues in perpendicular magnetic recording," *IEEE Trans. Magn.*, vol. 37, pp. 1929-1931, July 2001.

- [7] H. Meyr, M. Moeneclaey, and S. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing*. New York, NY: Wiley, 1998.
- [8] K. Mueller and M. Müller, "Timing recovery in digital synchronous data receivers," *IEEE Trans. Commun.*, COM-24, pp. 516-531, May 1976.
- [9] J.M.W. Bergmans, *Digital Baseband Transmission and Recording*. Norwell, MA: Kluwer Academic Publishers, 1996.
- [10] F. Gardner, "Interpolation in digital modems—part I: fundamentals," *IEEE Trans. Commun.*, vol. 41, pp. 501-507, March 1993.
- [11] L. Erup, F. Gardner, and R. Harris, "Interpolation in digital modems—part II: implementation and performance," *IEEE Trans. Commun.*, vol. 41, pp. 998-1008, June 1993.
- [12] T. Laakso, V. Valimäki, M. Karjalainen, and U. Laine, "Splitting the unit delay," *IEEE Signal Process. Mag.*, vol. 13, pp. 30-60, Jan. 1996.
- [13] D. Kim, M. Narasimha, and D. Cox, "Design of optimal interpolation filter for symbol timing recovery," *IEEE Trans. Commun.*, vol. 45, pp. 877-884, July 1997.
- [14] G. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268-278, March 1973.
- [15] R. Pintelon, P. Guillaume, Y. Rolain, J. Schoukens, and H. Van Hamme, "Parametric identification of transfer functions in the frequency domain—a survey," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 2245-2260, Nov. 1994.
- [16] R. Wilson, "Automatic coupled-transmission-line model generation," to be submitted to *IEEE Trans. Microwave Theory Tech.*
- [17] J. Johnson and D. Trudnowski, "Frequency-domain transfer-function identification using Chebyshev polynomials," *Proc. 32nd Conf. Decision and Control*, Dec. 1993, pp. 3510-3511.
- [18] J. Caroselli, J. Fitzpatrick, and J.K. Wolf, "A simple model for transition interactions with the microtrack model," *ICC 1998 Conf. Record*, vol. 2, 1998, pp. 942-946.
- [19] R. Cideciyan, J. Coker, E. Eleftheriou, and R. Galbraith, "Noise predictive maximum likelihood detection combined with parity-based post-processing," *IEEE Trans. Magn.*, vol. 37, pp. 714-720, March 2001.
- [20] T. Nishiya, S. Mita, and T. Nara, "Turbo decoding of a low-density generator matrix code for a high-density magnetic recording channel," *Globecom 2000 Conf. Record*, vol. 2, pp. 1882-1886, 2000.



Ross Wilson (M'76) received the B.S.E.E. and M.S.E.E. degrees from the University of California, Berkeley, in 1971 and 1972, respectively. He is currently with Agere Systems, Inc. He has worked in the data storage industry, and as a Consultant, and has been granted ten U.S. patents. His interests include architectural and mixed-signal circuit design and modelling for servo systems, storage, RF, and communications.

Scott O'Brien received the B.S.E.E. degree from the University of Minnesota in 2001. He is currently with Agere Systems, Inc. His interests include high speed analog circuit design and communication systems.

Daniel Galaba received his B.S.C.S. degree from Minnesota State University, Minneapolis, MN and is presently working towards the M.S.S.E. degree at St. Thomas University, St. Paul, MN. He currently is employed at Agere Systems. His work has focused on magnetic recording systems, embedded systems, and motion control.

John Leighton received his B.S.E.E., M.S., and Ph.D. degrees from the University of Minnesota in 1984, 1989, and 1994, respectively.