Technical Note No.
WS-N-2021.006

Welkin Sciences

Adaptive T/N Equalizers for Quadrature Amplitude Modulation

Date: 11 November 2021

Prepared by: Dr. David K. Milovich Welkin Sciences, Colorado Springs, CO

Security Classification: Unclassified (U)

1 A Matched T/N Transversal Filter

1.1 Problem Statement

- Let $\mathbf{d}$ be an integer-indexed sequence of complex-valued data symbols d_ζ and assume that each symbol is a random variable with mean zero, unit variance, and no correlation with other symbols in the stream. In other words, assume that $\langle d_\zeta \rangle = 0$ and $\left\langle d_\zeta d_\eta^* \right\rangle = \delta_{\zeta,\eta}$ where $\langle \bullet \rangle$ is the expectation operator and δ is the Kronecker δ. (For example, each d_ζ could be a QPSK-modulated symbol in $\{(\pm 1 \pm j)/\sqrt{2}\}$.)
- Fixing $N \geq 1$ and sampling N times per symbol, define $z_l = d_\zeta$ where $\zeta = \lfloor l/N \rfloor$.
- Assume the sample sequence $\mathbf{z}$ is convolved with a known time-invariant FIR filter $\mathbf{g}$ to produce a sequence $\mathbf{w}$ as follows.
$$w_l = \sum_{m \in \mathcal{G}} g_m z_{l-m} \tag{1}$$
Here $\mathcal{G}$ is some (known) finite interval of integers.
- Add white Gaussian noise $\mathbf{y}$ to $\mathbf{w}$ to produce $\mathbf{r}$. Assume that $\mathbf{y}$ and $\mathbf{z}$ are uncorrelated. Because $\mathbf{y}$ is white, $\langle y_l y_m^* \rangle = N_0 \delta_{l,m}$ for a known $N_0 > 0$.
- Fix a finite interval of integers $\mathcal{F}$.
- Given a hypothetical FIR filter $\mathbf{f} = (f_n : n \in \mathcal{F})$, define an associated sequence $\hat{\mathbf{d}}$ of recovered symbols by
$$\hat{d}_\zeta = \sum_{n \in \mathcal{F}} f_n r_{N\zeta - n}. \tag{2}$$
- **Problem.** Find an $\mathbf{f}$ with least mean square error MSE $= \left\langle \left| \hat{d}_\zeta - d_\zeta \right|^2 \right\rangle$.

The above filter $\mathbf{f}$ is called a *transversal* filter in the literature.

As we shall see, with do not need to know $\mathbf{g}$ *per se*. It suffices to know $\mathbf{g}$ convolved with a unit pulse of length N, that is, $g'_m = g_m + g_{m-1} + \cdots + g_{m-N+1}$. Moreover, it is feasible to measure $\mathbf{g}'$ accurately without changing our T/N modulation. For example, if we send a unit impulse data stream $d_\zeta = \delta_{\zeta,0}$, then the received signal $\mathbf{r}$ is $\mathbf{g}' + \mathbf{y}$. Continuing the example, if we transmit a data stream of widely spaced repeated unit impulses, then we use the received signal to compute a least squares estimate of $\mathbf{g}'$ and N_0.

1.2 Direct Solution

Because the MSE is a nonnegative polynomial function of the real and imaginary parts of $\mathbf{f}$, there is indeed at least one optimal $\mathbf{f}$. Moreover, any optimal $\mathbf{f}$ will be a stationary point of the MSE function. A filter $\mathbf{f}$ is such a stationary point if and only if we have

$$0 = \frac{\partial}{\partial f_\nu^*} \left\langle (\hat{d}_\zeta - d_\zeta)(\hat{d}_\zeta - d_\zeta)^* \right\rangle \tag{3}$$

for all $\nu \in \mathcal{F}$. The above derivative is the Wirtinger derivative with respect to the complex conjugate f_ν^* of f_ν. Recursively expanding the definition of $\hat{d}_\zeta$ and then differentiating with respect to f_ν^* (but treating f_ν as a constant), we reduce (3) to

$$0 = \left\langle \left[\sum_{n \in \mathcal{F}} f_n \left(y_{N\zeta - n} + \sum_{m \in \mathcal{G}} g_m z_{N\zeta - n - m} \right) - d_\zeta \right] \left[y_{N\zeta - \nu} + \sum_{p \in \mathcal{G}} g_p z_{N\zeta - \nu - p} \right]^* \right\rangle. \tag{4}$$

The above equation simplifies as follows, first using the assumptions that $\mathbf{y}$ and $\mathbf{z}$ are uncorrelated and that $\mathbf{y}$ is δ-correlated, second using the assumption that $\mathbf{d}$ is δ-correlated, and third introducing new abbreviations.

$$0 = N_0 f_\nu + \sum_{n\in\mathcal{F}} f_n \sum_{m,p\in\mathcal{G}} g_m g_p^* \left\langle z_{N\zeta-n-m} z^*_{N\zeta-\nu-p} \right\rangle \quad (5)$$

$$- \sum_{p\in\mathcal{G}} g_p^* \left\langle z_{N\zeta-\nu-p} d_\zeta \right\rangle \quad (6)$$

$$= N_0 f_\nu + \sum_{n\in\mathcal{F}} f_n \sum \{g_m g_p^* : m,p \in \mathcal{G} \text{ and } \lceil (n+m)/N \rceil = \lceil (\nu+p)/N \rceil \} \quad (7)$$

$$- \sum \{g_p^* : p \in \mathcal{G} \cap (-N-\nu, -\nu]\} \quad (8)$$

$$= N_0 f_\nu + \sum_{n\in\mathcal{F}} f_n R_{\nu,n} \quad (9)$$

$$- \gamma_\nu \quad (10)$$

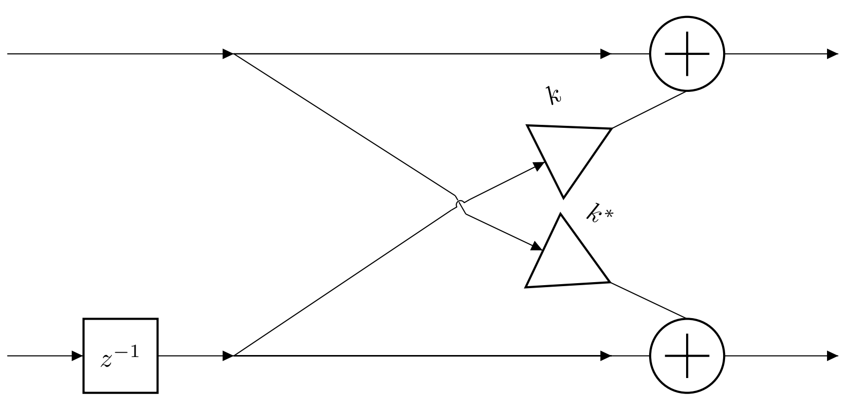
In matrix notation, we have $(N_0\mathbf{I} + \mathbf{R})\mathbf{f} = \boldsymbol{\gamma}$. Observe that $\mathbf{R}$ is Hermitian and positive definite. Therefore, $\mathbf{f} = (N_0\mathbf{I}+\mathbf{R})^{-1}\boldsymbol{\gamma}$ is the unique optimum. Also observe that $R_{\nu,n} = R_{\nu+N,n+N}$. Therefore, if the filter length $|\mathcal{F}|$ is a multiple of N, then $(N_0\mathbf{I}+\mathbf{R})\mathbf{f} = \boldsymbol{\gamma}$ is a block Toeplitz system and the block Levinson algorithm[1] will solve for $\mathbf{f}$ much more efficiently than an arbitrary Hermitian system can be solved. We will describe how to do this in Subsection 1.3.

In general, $\mathbf{f}$ can always be made more accurate by expanding $\mathcal{F}$. But we see from the definition of $\boldsymbol{\gamma}$ that the largest accuracy gains come from expanding the overlap of $\mathcal{F}$ and $-\mathcal{G}$. (Here $-\mathcal{G} = \{-m : m \in \mathcal{G}\}$.)

For our above matched filter and for all other T/N filters we discuss in this section and Section 2, we do not actually need exact knowledge of $\mathbf{g}$. Rather, it suffices to know $\boldsymbol{\gamma}$. In particular, our above optimal $\mathbf{f}$ is a function of $\boldsymbol{\gamma}$ because $R_{\nu,n} = \sum_k \gamma_{\nu+Nk}\gamma^*_{n+Nk}$. Moreover, as discussed in Subsection 1.1, we could measure each $\gamma_\nu = g^*_{-\nu} + \cdots + g^*_{-\nu-N+1}$ by including in $\mathbf{d}$ a training sequence of widely spaced unit impulses.

1.3 A Lattice Solution

In Subsection 1.2, the matched transversal filter $\mathbf{f}$ is directly optimized for a fixed interval of delay indices $\mathcal{F}$. In this subsection we describe an incremental implementation of the same optimal solution that starts with an initial $\mathcal{F} = \mathcal{F}_0$ of length N and uses the block Levinson algorithm to repeatedly append or prepend N elements to $\mathcal{F}$. In the $T/1$ setting, this incremental approach is called a *lattice filter* by some authors because its block diagram includes lattice structures of the form below.[2]



In the on-line setting, where $\mathbf{f}$ must track a slowly varying $\mathbf{g}$, (a T/N analog of) a lattice filter is more promising than the direct approach of Subsection 1.2. Why? Consider the increments $\hat{d}^{(0)}_\zeta$, $\hat{d}^{(1)}_\zeta - \hat{d}^{(0)}_\zeta$, $\hat{d}^{(2)}_\zeta - \hat{d}^{(1)}_\zeta$, $\hat{d}^{(3)}_\zeta - \hat{d}^{(2)}_\zeta$, ... of a recovered symbol that correspond to incrementally extending $\mathcal{F}$ by N elements at a time from $\mathcal{F}_0$ to $\mathcal{F}_1$ to $\mathcal{F}_2$ to $\mathcal{F}_3$ to ... If each $\hat{d}^{(n)}_\zeta$ is nearly optimal, then all the increments are nearly uncorrelated with each other, which greatly accelerates adaptive optimization of these increments.[2] Moreover, these increments can be efficiently computed as explained below.

We begin our lattice solution by fixing some notation. For simplicity, we will assume $\min(\mathcal{F})$ is a multiple of N. Let $\mathbf{f}^{[\alpha,\beta]}$ denote the optimal $\mathbf{f}$ for $\mathcal{F} = [N\alpha, N\beta + N)$ and let $\boldsymbol{\gamma}^{[\alpha,\beta]} = \boldsymbol{\gamma}$ for $\mathcal{F} = [N\alpha, N\beta + N)$:

$$(\boldsymbol{\gamma}^{[\alpha,\beta]})^T = \begin{bmatrix}\gamma_{N\alpha} & \cdots & \gamma_{N\beta+N-1}\end{bmatrix}, \qquad \mathbf{f}^{[\alpha,\beta]} = (N_0\mathbf{I} + \mathbf{R})^{-1}\boldsymbol{\gamma}^{[\alpha,\beta]}. \tag{11}$$

Let $T_{p,q} = N_0\delta_{p,q} + R_{p,q}$ for all integers p, q and divide $\mathbf{T}$ into $N \times N$ blocks each equal to $\mathbf{S}_n$ for some integer n where $(\mathbf{S}_n)_{p,q} = T_{p-1,Nn+q-1}$ for $p, q \in [1, N]$. This is possible because $\mathbf{R}$ has the block Toeplitz property $R_{p,q} = R_{N+p,N+q}$ for all integers p, q. For each $n \geq 0$, let $\mathbf{T}_n$ denote the $(Nn + N) \times (Nn + N)$ matrix given by

$$\mathbf{T}_n = \begin{bmatrix} \mathbf{S}_0 & \mathbf{S}_1 & \mathbf{S}_2 & \cdots & \mathbf{S}_n \\ \mathbf{S}_{-1} & \mathbf{S}_0 & \mathbf{S}_1 & \cdots & \mathbf{S}_{n-1} \\ \mathbf{S}_{-2} & \mathbf{S}_{-1} & \mathbf{S}_0 & \cdots & \mathbf{S}_{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{S}_{-n} & \mathbf{S}_{1-n} & \mathbf{S}_{2-n} & \cdots & \mathbf{S}_0 \end{bmatrix}. \tag{12}$$

Note that if $\mathcal{F} = [N\alpha, N\beta + N)$, then $N_0\mathbf{I} + \mathbf{R} = \mathbf{T}_{\beta-\alpha}$. Also note that $\mathbf{S}_{-n} = \mathbf{S}_n^H$ because $\mathbf{R}$ is Hermitian.

Using the above notation, the optimal matched filter for an initial $\mathcal{F}_0 = [N\alpha, N\alpha + N)$ is given by $\mathbf{f}^{[\alpha,\alpha]} = \mathbf{S}_0^{-1}\boldsymbol{\gamma}^{[\alpha,\alpha]}$. (A natural choice for this α is $\lfloor -m/N \rfloor$ where m maximizes $|g_m|$.) Next, we will use the block Levinson algorithm to compute incremental improvements of the form

$$\mathbf{f}^{[\alpha-1,\beta]} - \begin{bmatrix}\mathbf{0} \\ \mathbf{f}^{[\alpha,\beta]}\end{bmatrix} \qquad \text{or} \qquad \mathbf{f}^{[\alpha,\beta+1]} - \begin{bmatrix}\mathbf{f}^{[\alpha,\beta]} \\ \mathbf{0}\end{bmatrix} \tag{13}$$

where $\mathbf{0}$ is $N \times 1$. We will need one of the $(Nn + N) \times N$ matrix solutions $\mathbf{A}_n$ and $\mathbf{B}_n$ of

$$\mathbf{T}_n\mathbf{A}_n = \begin{bmatrix}\mathbf{I} \\ \mathbf{0}\end{bmatrix}, \qquad \mathbf{T}_n\mathbf{B}_n = \begin{bmatrix}\mathbf{0} \\ \mathbf{I}\end{bmatrix} \tag{14}$$

where $\mathbf{I}$ is $N \times N$, $\mathbf{0}$ is $Nn \times N$, and $n = \beta + 1 - \alpha$. To obtain $\mathbf{A}_{\beta+1-\alpha}$ or $\mathbf{B}_{\beta+1-\alpha}$, we start with $\mathbf{A}_0 = \mathbf{B}_0 = \mathbf{S}_0^{-1}$. Then, for each $n \leq \beta - \alpha$, we compute $\mathbf{A}_{n+1}$ and $\mathbf{B}_{n+1}$ from $\mathbf{A}_n$ and $\mathbf{B}_n$ as follows. We divide $\mathbf{A}_n$ and $\mathbf{B}_n$ into $N \times N$ blocks:

$$\mathbf{A}_n^T = \begin{bmatrix}\mathbf{A}_{n,0}^T & \mathbf{A}_{n,1}^T & \cdots & \mathbf{A}_{n,n}^T\end{bmatrix}, \qquad \mathbf{B}_n^T = \begin{bmatrix}\mathbf{B}_{n,0}^T & \mathbf{B}_{n,1}^T & \cdots & \mathbf{B}_{n,n}^T\end{bmatrix}. \tag{15}$$

Then we compute the $N \times N$ matrices

$$\mathbf{C}_n = \sum_{m=0}^{n} \mathbf{S}_{m+1}^H \mathbf{A}_{n,m}, \qquad \mathbf{D}_n = (\mathbf{I} - \mathbf{C}_n^H\mathbf{C}_n)^{-1}, \qquad \text{and} \qquad \mathbf{E}_n = (\mathbf{I} - \mathbf{C}_n\mathbf{C}_n^H)^{-1}. \tag{16}$$

Finally,

$$\mathbf{A}_{n+1} = \left(\begin{bmatrix}\mathbf{A}_n \\ \mathbf{0}\end{bmatrix} - \begin{bmatrix}\mathbf{0} \\ \mathbf{B}_n\end{bmatrix}\mathbf{C}_n\right)\mathbf{D}_n \qquad \text{and} \qquad \mathbf{B}_{n+1} = \left(\begin{bmatrix}\mathbf{0} \\ \mathbf{B}_n\end{bmatrix} - \begin{bmatrix}\mathbf{A}_n \\ \mathbf{0}\end{bmatrix}\mathbf{C}_n^H\right)\mathbf{E}_n \tag{17}$$

where $\mathbf{0}$ is $N \times N$. Actually, we can use the symmetry $\mathbf{B}_{\nu,\mu} = \mathbf{A}_{\nu,\nu-\mu}^H$ to compute $\mathbf{B}_{n+1}$ without computing $\mathbf{E}_n$. However, we will need $\mathbf{E}_n$ later.

Given $\mathbf{f}^{[\alpha,\beta]}$ and $\mathbf{A}_{\beta+1-\alpha}$ or $\mathbf{B}_{\beta+1-\alpha}$ for some $[\alpha, \beta]$, we compute the increment (13) as follows. Divide $\mathbf{f}^{[\alpha,\beta]}$ and $\boldsymbol{\gamma}^{[\alpha,\beta]}$ into blocks of $N \times 1$ column vectors $\mathbf{f}_n^{[\alpha,\beta]}$ and $\boldsymbol{\gamma}_n^{[\alpha,\beta]}$ for $n \in [0, \beta - \alpha]$:

$$(\mathbf{f}^{[\alpha,\beta]})^T = \begin{bmatrix}(\mathbf{f}_0^{[\alpha,\beta]})^T & \cdots & (\mathbf{f}_{\beta-\alpha}^{[\alpha,\beta]})^T\end{bmatrix}, \qquad (\boldsymbol{\gamma}^{[\alpha,\beta]})^T = \begin{bmatrix}(\boldsymbol{\gamma}_0^{[\alpha,\beta]})^T & \cdots & (\boldsymbol{\gamma}_{\beta-\alpha}^{[\alpha,\beta]})^T\end{bmatrix}. \tag{18}$$

Then we compute the $N \times 1$ column vector

$$\boldsymbol{\sigma}^{[\alpha,\beta]} = \boldsymbol{\gamma}^{[\alpha-1,\alpha-1]} - \sum_{m=0}^{\beta-\alpha} \mathbf{S}_{m+1}\mathbf{f}_m^{[\alpha,\beta]} \qquad \text{or} \qquad \boldsymbol{\tau}^{[\alpha,\beta]} = \boldsymbol{\gamma}^{[\beta+1,\beta+1]} - \sum_{m=0}^{\beta-\alpha} \mathbf{S}_{\beta+1-\alpha-m}^H \mathbf{f}_m^{[\alpha,\beta]}. \tag{19}$$

Finally, we compute

$$\mathbf{f}^{[\alpha-1,\beta]} - \begin{bmatrix}\mathbf{0}\\ \mathbf{f}^{[\alpha,\beta]}\end{bmatrix} = \mathbf{A}_{\beta+1-\alpha}\boldsymbol{\sigma}^{[\alpha,\beta]} \qquad \text{or} \qquad \mathbf{f}^{[\alpha,\beta+1]} - \begin{bmatrix}\mathbf{f}^{[\alpha,\beta]}\\ \mathbf{0}\end{bmatrix} = \mathbf{B}_{\beta+1-\alpha}\boldsymbol{\tau}^{[\alpha,\beta]}. \tag{20}$$

Next we describe how to efficiently compute the recovered symbol increments using an iteration of lattice-like steps. Letting $\hat{d}_\zeta^{[\alpha,\beta]}$ denote the recovered symbol (2) for $\mathcal{F} = [N\alpha, N\beta + N)$ and letting $\boldsymbol{\rho}_\zeta^{[\alpha,\beta]}$ denote the row vector of corrupted samples

$$\begin{bmatrix} r_{N(\zeta-\alpha)} & r_{N(\zeta-\alpha)-1} & \cdots & r_{N(\zeta-\beta)-N+1}\end{bmatrix}, \tag{21}$$

we have $\hat{d}_\zeta^{[\alpha,\beta]} = \boldsymbol{\rho}_\zeta^{[\alpha,\beta]}\mathbf{f}^{[\alpha,\beta]}$. In particular, for an initial $\mathcal{F}$ of the form $[N\alpha, N\alpha+N)$, we compute $\hat{d}_\zeta^{[\alpha,\alpha]} = \boldsymbol{\rho}_\zeta^{[\alpha,\alpha]}\mathbf{f}^{[\alpha,\alpha]}$. To compute an increment $\hat{d}_\zeta^{[\alpha-1,\beta]} - \hat{d}_\zeta^{[\alpha,\beta]}$ or $\hat{d}_\zeta^{[\alpha,\beta+1]} - \hat{d}_\zeta^{[\alpha,\beta]}$, we left-multiply (20) by $\boldsymbol{\rho}_\zeta^{[\alpha,\beta]}$ to obtain

$$\hat{d}_\zeta^{[\alpha-1,\beta]} - \hat{d}_\zeta^{[\alpha,\beta]} = \boldsymbol{\varphi}_\zeta^{[\alpha-1,\beta]}\boldsymbol{\sigma}^{[\alpha,\beta]} \qquad \text{or} \qquad \hat{d}_\zeta^{[\alpha,\beta+1]} - \hat{d}_\zeta^{[\alpha,\beta]} = \boldsymbol{\psi}_\zeta^{[\alpha,\beta+1]}\boldsymbol{\tau}^{[\alpha,\beta]} \tag{22}$$

where $\boldsymbol{\varphi}_\eta^{[\mu,\nu]} = \boldsymbol{\rho}_\eta^{[\mu,\nu]}\mathbf{A}_{\nu-\mu}$ and $\boldsymbol{\psi}_\eta^{[\mu,\nu]} = \boldsymbol{\rho}_\eta^{[\mu,\nu]}\mathbf{B}_{\nu-\mu}$. To efficiently compute the relevant $1 \times N$ row vectors $\boldsymbol{\varphi}_\eta^{[\mu,\nu]}$ and $\boldsymbol{\psi}_\eta^{[\mu,\nu]}$, we start with a boundary value of the form $\boldsymbol{\varphi}_\zeta^{[\alpha,\alpha]} = \boldsymbol{\psi}_\zeta^{[\alpha,\alpha]} = \boldsymbol{\rho}_\zeta^{[\alpha,\alpha]}\mathbf{S}_0^{-1}$ and then iterate lattice-like computations each of the form

$$\begin{bmatrix}\boldsymbol{\varphi}_\zeta^{[\alpha-1,\beta]} & \boldsymbol{\psi}_\zeta^{[\alpha-1,\beta]}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\varphi}_{\zeta+1}^{[\alpha,\beta]} & \boldsymbol{\psi}_\zeta^{[\alpha,\beta]}\end{bmatrix}\begin{bmatrix}\mathbf{D}_n & -\mathbf{C}_n^H\mathbf{E}_n\\ -\mathbf{C}_n\mathbf{D}_n & \mathbf{E}_n\end{bmatrix} \tag{23}$$

or

$$\begin{bmatrix}\boldsymbol{\varphi}_\zeta^{[\alpha,\beta+1]} & \boldsymbol{\psi}_\zeta^{[\alpha,\beta+1]}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\varphi}_\zeta^{[\alpha,\beta]} & \boldsymbol{\psi}_{\zeta-1}^{[\alpha,\beta]}\end{bmatrix}\begin{bmatrix}\mathbf{D}_n & -\mathbf{C}_n^H\mathbf{E}_n\\ -\mathbf{C}_n\mathbf{D}_n & \mathbf{E}_n\end{bmatrix} \tag{24}$$

where $n = \beta - \alpha$. The iterative formulas (23) and (24) are consequences of (17).

Finally, summing an initial recovered symbol $\hat{d}_\zeta^{(0)}$ of the form $\hat{d}_\zeta^{[\alpha,\alpha]}$ and increments $\hat{d}_\zeta^{(n)} - \hat{d}_\zeta^{(n-1)}$ each of the form $\hat{d}_\zeta^{[\alpha-1,\beta]} - \hat{d}_\zeta^{[\alpha,\beta]}$ or $\hat{d}_\zeta^{[\alpha,\beta+1]} - \hat{d}_\zeta^{[\alpha,\beta]}$ for all n in some interval $[1, D)$, our lattice solution outputs $\hat{d}_\zeta = \hat{d}_\zeta^{(0)} + \sum_{n=1}^{D-1}\left(\hat{d}_\zeta^{(n)} - \hat{d}_\zeta^{(n-1)}\right)$.

2 Decision Feedback Equalization

2.1 Problem Statement

Now we add a decision feedback section to the matched filter of Section 1. From the output of the hypothetical FIR filter $\mathbf{f}$, which we will now call the *forward* filter, we will subtract the output of a *feedback* FIR filter $\mathbf{b} = (b_\eta : \eta \in \mathcal{B})$ where $\mathcal{B}$ is a finite interval of integers. Unlike $\mathbf{f}$, which is typically not causal, we require that $\mathbf{b}$ be strictly causal, by which we mean $\min(\mathcal{B}) \geq 1$. The input to the feedback filter will be $\check{\mathbf{d}}$ where each $\check{d}_\zeta$ is a hard decision estimate of d_ζ from $\hat{d}_\zeta$. Thus, our *decision feedback equalizer* computes $\hat{\mathbf{d}}$ as follows.

$$\hat{d}_\zeta = \sum_{n\in\mathcal{F}} f_n r_{N\zeta-n} - \sum_{\eta\in\mathcal{B}} b_\eta \check{d}_{\zeta-\eta} \tag{25}$$

We will not try to find an optimal $(\mathbf{f}, \mathbf{b})$ for (25) because of the analytic difficulties introduced by discontinuous hard decisions. Instead, following Monsen[4], we will optimize an idealized feedback equalizer in which the true symbols $\mathbf{d}$ are the input to the feedback filter:

$$\hat{d}_\zeta = \sum_{n\in\mathcal{F}} f_n r_{N\zeta-n} - \sum_{\eta\in\mathcal{B}} b_\eta d_{\zeta-\eta}. \tag{26}$$

However, if the error rate for the hard decisions $\check{\mathbf{d}}$ is small, then an optimal $(\mathbf{f}, \mathbf{b})$ for our idealized equalizer (26) will be nearly optimal for our decision feedback equalizer (25).

2.2 Direct Solution

Using (26) to define $\hat{d}_\zeta$, we search for a stationary point of the MSE function like we did in Subsection 1.2. This time, we obtain the system of linear equations

$$0 = \frac{\partial \mathrm{MSE}}{\partial f_\nu^*} = N_0 f_\nu + \sum_{n\in\mathcal{F}} f_n R_{\nu,n} - \sum_{\eta\in\mathcal{B}} b_\eta \gamma_{\nu-N\eta} - \gamma_\nu \tag{27}$$

$$0 = \frac{\partial \mathrm{MSE}}{\partial b_\theta^*} = - \sum_{n\in\mathcal{F}} f_n \gamma^*_{n-N\theta} + b_\theta \tag{28}$$

where ν ranges over $\mathcal{F}$ and θ ranges over $\mathcal{B}$. Letting $\Gamma_{\nu,\theta} = \gamma_{\nu-N\theta}$, the block matrix expression of the above system of equation is

$$\begin{bmatrix} N_0\mathbf{I}+\mathbf{R} & -\mathbf{\Gamma} \\ -\mathbf{\Gamma}^H & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\gamma} \\ \mathbf{0} \end{bmatrix}. \tag{29}$$

Like in Subsection 1.2, we have a Hermitian matrix. And again, for a given $\mathbf{g}$, this matrix is invertible except possibly at finitely many values of N_0. Thus, there is almost always a unique optimum given by $\mathbf{f} = (N_0\mathbf{I} + \mathbf{R} - \mathbf{\Gamma}\mathbf{\Gamma}^H)^{-1}\boldsymbol{\gamma}$ and $\mathbf{b} = \mathbf{\Gamma}^H\mathbf{f}$.

By examining (28), we see that, for a given $\mathcal{F}$, symbol recovery accuracy can be increased by decreasing $\min(\mathcal{B})$ (but no lower than 1) and by increasing $\max(\mathcal{B})$ within the interval $[\min(\mathcal{B}), |\mathcal{F}|/N)$. But increasing $\max(\mathcal{B})$ to $\lceil |\mathcal{F}|/N \rceil$ or beyond merely zero-pads $\mathbf{b}$.

Unlike in Subsection 1.2, the system of equations (29) is not block Toeplitz in general. However, if $|\mathcal{F}|$ is a multiple of $N|\mathcal{B}|$, then we can reorder the rows and columns of (29) to make it block Toeplitz and, hence, amenable to the block Levinson algorithm. We describe how to do this in Subsection 2.3.

2.3 A Lattice Solution

As discussed in Subsection 1.3, a lattice-like implementation of a matched filter that computes each recovered symbol as a sum of nearly uncorrelated components is more promising in the on-line setting than is a direct implementation as a transversal filter. This is still true after adding a decision feedback section. In this subsection, we describe a lattice-like implementation of the optimal filter identified in Subsection 2.2.

To make the block Levinson algorithm applicable, $|\mathcal{F}|$ must be a multiple of $N|\mathcal{B}|$. If $|\mathcal{F}| = Nc|\mathcal{B}|$, then the block size is $(Nc+1)\times(Nc+1)$. We will only describe how to implement case $c=1$ because this case is simplest and because, for a given $\mathcal{F}$, choosing $\mathcal{B}$ to make $c=1$ produces higher symbol recovery accuracy than other choices of c. So, we reorder the rows and columns of (29) into $(N+1)\times(N+1)$ blocks as follows. Let $D = |\mathcal{B}|$, $\iota = \min(\mathcal{B})$, and $\varsigma = \min(\mathcal{F})$. (Presumably, $\iota = 1$ is desired.) For all $n \in [0, D)$, regroup the coordinates of column vectors $\begin{bmatrix}\mathbf{f} & \mathbf{b}\end{bmatrix}^T$ and $\begin{bmatrix}\boldsymbol{\gamma} & \mathbf{0}\end{bmatrix}^T$ of (29) into blocks of $(N+1)\times 1$ column vectors

$$\tilde{\mathbf{f}}_n^T = \begin{bmatrix} f_{\varsigma+Nn} & \cdots & f_{\varsigma+Nn+N-1} & b_{\iota+n} \end{bmatrix}, \tag{30}$$

$$\tilde{\boldsymbol{\gamma}}_n^T = \begin{bmatrix} \gamma_{\varsigma+Nn} & \cdots & \gamma_{\varsigma+Nn+N-1} & 0 \end{bmatrix} \tag{31}$$

and define the $(N+1)(n+1)\times(N+1)(n+1)$ Hermitian block Toeplitz matrix

$$\mathbf{T}_n = \begin{bmatrix} \mathbf{S}_0 & \cdots & \mathbf{S}_n \\ \vdots & & \vdots \\ \mathbf{S}_{-n} & \cdots & \mathbf{S}_0 \end{bmatrix} \tag{32}$$

by

$$(\mathbf{S}_{\beta-\alpha})_{p+1,q+1} = (N_0\mathbf{I}+\mathbf{R})_{\varsigma+N\alpha+p,\varsigma+N\beta+q} = (N_0\mathbf{I}+\mathbf{R})_{\varsigma+p,\varsigma+N(\beta-\alpha)+q} \tag{33}$$

$$(\mathbf{S}_{\beta-\alpha})_{p,N+1} = -\Gamma_{\varsigma+N\alpha+p,\iota+\beta} = -\gamma_{\varsigma+N(\alpha-\beta-\iota)+p} \tag{34}$$

$$(\mathbf{S}_{\beta-\alpha})_{N+1,q} = -(\mathbf{\Gamma}^H)_{\iota+\alpha,\varsigma+N\beta+q} = -\gamma^*_{\varsigma+N(\beta-\alpha-\iota)+q} \tag{35}$$

$$(\mathbf{S}_{\beta-\alpha})_{N+1,N+1} = 1 \tag{36}$$

for all $\alpha, \beta \in [0, n]$ and $p, q \in [0, N)$. For each $[\alpha, \beta] \subset [0, D)$, define

$$(\boldsymbol{\gamma}^{[\alpha,\beta]})^T = \begin{bmatrix} \tilde{\boldsymbol{\gamma}}_\alpha^T & \cdots & \tilde{\boldsymbol{\gamma}}_\beta^T \end{bmatrix}, \qquad \mathbf{f}^{[\alpha,\beta]} = \mathbf{T}_{\beta-\alpha}^{-1} \boldsymbol{\gamma}^{[\alpha,\beta]}. \tag{37}$$

In this new notation, the optimal $(\mathbf{f}, \mathbf{b})$ of Subsection 2.2 is given by the appropriately permuted coordinates of $\mathbf{f}^{[0,D-1]}$. The corresponding recovered symbol $\hat{d}_\zeta$ is $\boldsymbol{\rho}_\zeta^{[0,D-1]} \mathbf{f}^{[0,D-1]}$ where

$$\boldsymbol{\rho}_\zeta^{[\alpha,\alpha]} = \begin{bmatrix} r_{N(\zeta-\alpha)-\varsigma} & \cdots & r_{N(\zeta-\alpha)-\varsigma-N+1} & \check{d}_{\zeta-\alpha-\iota} \end{bmatrix}, \tag{38}$$

$$\boldsymbol{\rho}_\zeta^{[\alpha,\beta]} = \begin{bmatrix} \boldsymbol{\rho}_\zeta^{[\alpha,\alpha]} & \cdots & \boldsymbol{\rho}_\zeta^{[\beta,\beta]} \end{bmatrix}. \tag{39}$$

To efficiently compute $\mathbf{f}^{[0,D-1]}$ using the block Levinson algorithm, first compute $\mathbf{f}^{[\alpha,\alpha]} = \mathbf{S}_0^{-1} \boldsymbol{\gamma}^{[\alpha,\alpha]}$ for some α and then compute $\mathbf{f}^{[0,D-1]}$ iteratively. In each iteration, use the value of some $\mathbf{f}^{[\alpha,\beta]}$ to compute $\mathbf{f}^{[\alpha-1,\beta]}$ or $\mathbf{f}^{[\alpha,\beta+1]}$ exactly as in Subsection 1.3 except use $N+1$ in place of N and use the definitions of this subsection for $\mathbf{T}_n$, $\mathbf{S}_n$, $\boldsymbol{\gamma}^{[\alpha,\beta]}$, and $\mathbf{f}^{[\alpha,\beta]}$.

To efficiently compute a recovered symbol $\hat{d}_\zeta$ as a sum of nearly uncorrelated differences between adjacent terms in a sequence of improving estimates, first compute $\hat{d}_\zeta^{[\alpha,\alpha]} = \boldsymbol{\rho}_\zeta^{[\alpha,\alpha]} \mathbf{f}^{[\alpha,\alpha]}$ for some α. To compute an increment $\hat{d}_\zeta^{[\alpha-1,\beta]} - \hat{d}_\zeta^{[\alpha,\beta]}$ or $\hat{d}_\zeta^{[\alpha,\beta+1]} - \hat{d}_\zeta^{[\alpha,\beta]}$, proceed exactly as in Subsection 1.3 except use $N+1$ in place of N and use the definitions of this subsection for $\mathbf{T}_n$, $\mathbf{S}_n$, $\boldsymbol{\gamma}^{[\alpha,\beta]}$, $\mathbf{f}^{[\alpha,\beta]}$, and $\boldsymbol{\rho}_\zeta^{[\alpha,\beta]}$. Compute $D-1$ such increments to attain $\hat{d}_\zeta = \hat{d}_\zeta^{[0,D-1]}$.

3 An RLS Adaptive Transversal Filter

In contrast to Sections 1 and 2, where an unknown transmitted signal $\mathbf{z}$ was convolved with a known FIR filter $\mathbf{g}$ before adding white noise of known power, we will now treat the transmitted signal as known and the channel as unknown. The known transmitted signal should be interpreted as a training sequence that our filter uses to adapt to the unknown channel. Formally, our adaptive filter is a sequence of transversal filters converging towards a matched filter. We terminate the sequence of transversal filters when the training sequence is exhausted.

3.1 Problem Statement

- Let $\mathbf{d}$ and $\mathbf{z}$ be as in Subsection 1.1
- Let $\mathbf{r}$ denote the received discrete signal samples.
- Fix a finite interval of integers $\mathcal{F}$ and assume $r_l = 0$ for all $l \leq -\min(\mathcal{F})$.
- Given integers ζ, η and a hypothetical FIR filter $\mathbf{f}_\zeta = (f_{\zeta,i} : i \in \mathcal{F})$, define the recovered symbol $\hat{\mathbf{d}}_{\eta,\zeta}$ by

$$\hat{d}_{\eta,\zeta} = \sum_{i \in \mathcal{F}} f_{\zeta,i} r_{N\eta-i}. \tag{40}$$

  That is, $\hat{d}_{\eta,\zeta}$ is the estimate of d_η using $\mathbf{f}_\zeta$.
- Fix a forgetting factor $\lambda \in (0, 1]$ and a regularization factor $\delta \in (0, \infty)$.
- **Problem.** Find a sequence $\mathbf{f}_0, \mathbf{f}_1, \mathbf{f}_2, \ldots$ that minimizes the exponentially weighted and regularized objective function

$$E_\zeta = \sum_{\eta=0}^{\zeta} \lambda^{\zeta-\eta} \left| \hat{d}_{\eta,\zeta} - d_\eta \right|^2 + \delta \lambda^\zeta \sum_{i \in \mathcal{F}} |f_{\zeta,i}|^2 \tag{41}$$

  for each $\zeta \geq 0$.

3.2 RLS Solution

To minimize E_ζ, we require the Wirtinger derivatives be zero:

$$0 = \frac{\partial E_\zeta}{\partial f^*_{\zeta,i}} = \sum_{\eta=0}^{\zeta} \lambda^{\zeta-\eta}(\hat{d}_{\eta,\zeta} - d_\eta) r^*_{N\eta-i} + \delta \lambda^\zeta f_{\zeta,i}. \tag{42}$$

We restate (42) as $\mathbf{v}_\zeta = \mathbf{\Xi}_\zeta^{-1}\mathbf{f}_\zeta$ using the following vector notation.

$$\mathbf{r}_\eta = (r_{N\eta-i})_{i\in\mathcal{F}} \qquad \mathbf{v}_\zeta = \sum_{\eta=0}^{\zeta} \lambda^{\zeta-\eta} d_\eta \mathbf{r}^*_\eta \qquad \mathbf{\Xi}_\zeta^{-1} = \delta\lambda^\zeta \mathbf{I} + \sum_{\eta=0}^{\zeta} \lambda^{\zeta-\eta}\mathbf{r}^*_\eta \mathbf{r}^T_\eta \tag{43}$$

We now follow the general outline of [3, Ch. 9] for the RLS (recursive least squares) algorithm for computing $\mathbf{f}_\zeta = \mathbf{\Xi}_\zeta \mathbf{v}_\zeta$ using recursion with respect to ζ. By assumption, $\mathbf{r}_0 = \mathbf{0}$. Therefore, our solutions begins with $\mathbf{\Xi}_0 = \mathbf{I}/\delta$ and $\mathbf{f}_0 = \mathbf{v}_0 = \mathbf{0}$. From d_ζ, $\mathbf{r}_\zeta$, $\mathbf{\Xi}_{\zeta-1}$, and $\mathbf{f}_{\zeta-1}$ we compute $\mathbf{\Xi}_\zeta$ and $\mathbf{f}_\zeta$ as follows.

$$\boldsymbol{\xi}_\zeta = \mathbf{\Xi}_{\zeta-1}\mathbf{r}^*_\zeta \qquad \mathbf{\Xi}_\zeta = (\mathbf{I} - \mathbf{k}_\zeta \mathbf{r}^T_\zeta)\mathbf{\Xi}_{\zeta-1}/\lambda \tag{44}$$

$$\mathbf{k}_\zeta = \boldsymbol{\xi}_\zeta/(\lambda + \mathbf{r}^T_\zeta \boldsymbol{\xi}_\zeta) \qquad \mathbf{f}_\zeta = \mathbf{f}_{\zeta-1} + (d_\zeta - \mathbf{r}^T_\zeta \mathbf{f}_{\zeta-1})\mathbf{k}_\zeta \tag{45}$$

Fundamentally, the above computation exploits a "matrix inversion lemma" to cheaply update an inverse matrix $\mathbf{A}^{-1}$ to the inverse $(\mathbf{A} + \mathbf{x}\mathbf{x}^H)^{-1}$ of a rank-one update of $\mathbf{A}$.

4 An RLS Adaptive Decision Feedback Equalizer

As Section 2 modified Section 1, so this section modifies Section 3. We modify (40) by fixing a finite interval $\mathcal{B}$ of positive integers and defining the estimate $\hat{d}_{\eta,\zeta}$ of symbol d_η using forward filter $\mathbf{f}_\zeta$ and feedback filter $\mathbf{b}_\zeta = (b_{\zeta,\theta} : \theta \in \mathcal{B})$ to be

$$\hat{d}_{\eta,\zeta} = \sum_{i\in\mathcal{F}} f_{\zeta,i} r_{N\eta-i} - \sum_{\theta\in\mathcal{B}} b_{\zeta,\theta}\check{d}_{\eta-\theta}. \tag{46}$$

Here, $\check{d}_{\eta-\theta}$ is a hard decision estimate of $d_{\eta-\theta}$ from $\hat{d}_{\eta-\theta,\eta-\theta}$. We redefine the objective function (41) as follows.

$$E_\zeta = \sum_{\eta=0}^{\zeta} \lambda^{\zeta-\eta} \left|\hat{d}_{\eta,\zeta} - d_\eta\right|^2 + \delta\lambda^\zeta \left(\sum_{i\in\mathcal{F}} |f_{\zeta,i}|^2 + \sum_{\theta\in\mathcal{B}} |b_{\zeta,\theta}|^2\right) \tag{47}$$

Unlike Section 2, we do not need to approximate $\check{d}_{\eta-\theta}$ as $d_{\eta-\theta}$. Using the vector notation

$$\boldsymbol{\chi}_\zeta = \begin{bmatrix}\mathbf{f}_\zeta \\ \mathbf{b}_\zeta\end{bmatrix}, \qquad \mathbf{d}_\eta = (\check{d}_{\eta-\theta})_{\theta\in\mathcal{B}}, \qquad \mathbf{s}_\eta = \begin{bmatrix}\mathbf{r}_\eta \\ -\mathbf{d}_\eta\end{bmatrix}, \tag{48}$$

the RLS solution for $\boldsymbol{\chi}_\zeta$ is very similar to the solution of Subsection 3.2. Assuming $r_l = 0$ for all $l \leq -\min(\mathcal{F})$ and defining $\check{d}_\eta = 0$ for all $\eta < 0$, we start with $\mathbf{\Xi}_0 = \mathbf{I}/\delta$ and $\boldsymbol{\chi}_0 = \mathbf{0}$ and then iterate as follows.

$$\boldsymbol{\xi}_\zeta = \mathbf{\Xi}_{\zeta-1}\mathbf{s}^*_\zeta \qquad \mathbf{\Xi}_\zeta = (\mathbf{I} - \mathbf{k}_\zeta \mathbf{s}^T_\zeta)\mathbf{\Xi}_{\zeta-1}/\lambda \tag{49}$$

$$\mathbf{k}_\zeta = \boldsymbol{\xi}_\zeta/(\lambda + \mathbf{s}^T_\zeta \boldsymbol{\xi}_\zeta) \qquad \boldsymbol{\chi}_\zeta = \boldsymbol{\chi}_{\zeta-1} + (d_\zeta - \mathbf{s}^T_\zeta \boldsymbol{\chi}_{\zeta-1})\mathbf{k}_\zeta \tag{50}$$

5 RLS Adaptive Lattice Equalizers

In this section, we will modify the lattice equalizers of Subsection 1.3 and 2.3 to be adaptive. Like in Sections 3 and 4, a known signal will be transmitted over an unknown channel. We will use the matrices (16) to perform the lattice-like computation (24). We will use the RLS algorithm to compute these matrices instead of the block Levinson algorithm because we do not know the autocorrelation $\mathbf{R}$ of the channel (see

Subsection 1.2). However, by choosing a partially lagged objective function, the RLS update step for the matrices will still be much cheaper than the RLS update steps of Sections 3 and 4.

Fix integers D, ι, ς with $D, \iota \geq 1$ and let $\mathcal{F} = [\varsigma, \varsigma + ND)$ and $\mathcal{B} = [\iota, \iota + D)$. Our adaptive lattice equalizer will be a lattice implementation of an adapting sequence of forward filters $\mathbf{f}_\zeta = (f_{\zeta,l} : l \in \mathcal{F})$ and, optionally, an adapting sequence of feedback filters $\mathbf{b}_\zeta = (b_{\zeta,\theta} : \theta \in \mathcal{B})$. Here, $\zeta = 0, 1, 2, \ldots$ increases until the training sequence is exhausted. For each ζ, we will compute a sequence $\hat{d}_\zeta^{(0)}, \hat{d}_\zeta^{(1)}, \ldots, \hat{d}_\zeta^{(D-1)}$ of estimates of d_ζ. Here, $\hat{d}_\zeta^{(p)}$ corresponds to the segment $(f_{\zeta,l} : 0 \leq l - \varsigma < Np)$ of $\mathbf{f}_\zeta$ and, optionally, the segment $(b_{\zeta,\theta} : 0 \leq \theta - \iota < p)$ of $\mathbf{b}_\zeta$. We will choose our objective function to encourage each $\hat{d}_\zeta^{(p)}$ to converge towards an optimal estimate of d_ζ as ζ increases.

In analogy with (22) and (24), we compute the pth-degree estimate $\hat{d}_\zeta^{(p)}$ of d_ζ from:

- received signal samples $r_{N\zeta-\varsigma-l}$ for $0 \leq l < N(p+1)$,
- optionally, hard decision estimates $\check{d}_{\zeta-\theta}$ of $d_{\zeta-\theta}$ from $\hat{d}_{\zeta-\iota}^{(D-1)}$ for $0 \leq \theta \leq p$,
- $N \times N$ or $(N+1) \times (N+1)$ matrices $\mathbf{C}_{m,\zeta-p+m}$ for $0 \leq m \leq p$, and
- $N \times 1$ or $(N+1) \times 1$ column vectors $\boldsymbol{\tau}_{\zeta-p+m}^{(m)}$ for $0 \leq m \leq p$,

using the following procedure.

$$\boldsymbol{\rho}_\eta = \begin{bmatrix} r_{N\eta-\varsigma} & \cdots & r_{N\eta-\varsigma-N+1} \end{bmatrix} \text{ or } \begin{bmatrix} r_{N\eta-\varsigma} & \cdots & r_{N\eta-\varsigma-N+1} & \check{d}_{\eta-\iota} \end{bmatrix} \tag{51}$$

$$\boldsymbol{\varphi}_{\eta,\zeta}^{(0)} = \boldsymbol{\psi}_{\eta,\zeta}^{(0)} = \boldsymbol{\rho}_\eta \mathbf{C}_{0,\zeta}^H \tag{52}$$

$$\begin{bmatrix} \mathbf{D}_{p,\zeta} & \mathbf{E}_{p,\zeta} \end{bmatrix} = \begin{bmatrix} (\mathbf{I} - \mathbf{C}_{p,\zeta}^H \mathbf{C}_{p,\zeta})^{-1} & (\mathbf{I} - \mathbf{C}_{p,\zeta} \mathbf{C}_{p,\zeta}^H)^{-1} \end{bmatrix} \tag{53}$$

$$\begin{bmatrix} \boldsymbol{\varphi}_{\eta,\zeta}^{(p)} & \boldsymbol{\psi}_{\eta,\zeta}^{(p)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi}_{\eta,\zeta-1}^{(p-1)} & \boldsymbol{\psi}_{\eta-1,\zeta-1}^{(p-1)} \end{bmatrix} \begin{bmatrix} \mathbf{D}_{p,\zeta} & -\mathbf{C}_{p,\zeta}^H \mathbf{E}_{p,\zeta} \\ -\mathbf{C}_{p,\zeta} \mathbf{D}_{p,\zeta} & \mathbf{E}_{p,\zeta} \end{bmatrix} \tag{54}$$

$$\hat{d}_{\eta,\zeta}^{(0)} = \boldsymbol{\psi}_{\eta,\zeta}^{(0)} \boldsymbol{\tau}_\zeta^{(0)} \qquad \hat{d}_{\eta,\zeta}^{(p)} = \hat{d}_{\eta,\zeta-1}^{(p-1)} + \boldsymbol{\psi}_{\eta,\zeta}^{(p)} \boldsymbol{\tau}_\zeta^{(p)} \qquad \hat{d}_\zeta^{(p)} = \hat{d}_{\zeta,\zeta}^{(p)} \tag{55}$$

To define an objective function amenable to the RLS algorithm and to make the RLS update step highly parallelizable, we use two partially lagged estimates of $\hat{d}_{\eta,\zeta}^{(p)}$ as follows.

$$\hat{d}_{\eta,\zeta,1}^{(0)} = \boldsymbol{\psi}_{\eta,\zeta}^{(0)} \boldsymbol{\tau}_{\eta-1}^{(0)} \qquad \hat{d}_{\eta,\zeta,2}^{(0)} = \boldsymbol{\psi}_{\eta,\eta-1}^{(0)} \boldsymbol{\tau}_\zeta^{(0)} \tag{56}$$

$$\hat{d}_{\eta,\zeta,1}^{(p)} = \hat{d}_{\eta,\eta-1}^{(p-1)} + \left(\boldsymbol{\psi}_{\eta-1,\eta-1}^{(p-1)} - \boldsymbol{\varphi}_{\eta,\eta-1}^{(p-1)} \mathbf{C}_{p,\zeta}^H \right) \mathbf{E}_{p,\eta-1} \boldsymbol{\tau}_{\eta-1}^{(p)} \qquad \hat{d}_{\eta,\zeta,2}^{(p)} = \hat{d}_{\eta,\eta-1}^{(p-1)} + \boldsymbol{\psi}_{\eta,\eta-1}^{(p)} \boldsymbol{\tau}_\zeta^{(p)} \tag{57}$$

$$E_{\eta,\zeta} = \sum_{p=0}^{D-1} \sum_{i=1}^{2} \left| \hat{d}_{\eta,\zeta,i}^{(p)} - d_\eta \right|^2 \qquad E_\zeta = \sum_{\eta=0}^{\zeta} \lambda^{\zeta-\eta} E_{\eta,\zeta} + \delta \lambda^\zeta \left| \boldsymbol{\Lambda}_\zeta \right|^2 \tag{58}$$

Here, $\lambda \in (0,1]$ is a fixed forgetting factor, $\delta \in (0, \infty)$ is a fixed regularization factor, and $\boldsymbol{\Lambda}_\zeta$ is a vector that collects the coordinates of $\mathbf{C}_{p,\zeta}$ and $\boldsymbol{\tau}_\zeta^{(p)}$ for all $p \in [0, D)$.

The problem of finding $(\mathbf{C}_{p,\zeta}, \boldsymbol{\tau}_\zeta^{(p)} : 0 \leq p < D)$ that minimize E_ζ reduces to $2D$ vector equations that can be solved independently. To concisely express these equations, we write $\boldsymbol{\theta}_{p,\eta}$ for $\left(\boldsymbol{\psi}_{\eta,\eta-1}^{(p)}\right)^H$ and we introduce the $N^2 \times 1$ or $(N+1)^2 \times 1$ column vectors $\mathbf{c}_{p,\zeta}$ and $\boldsymbol{\omega}_{p,\eta}$ where

$$(\mathbf{C}_{p,\zeta})_{s,t} = (\mathbf{c}_{p,\zeta})_{Ns+t-N} \text{ or } (\mathbf{c}_{p,\zeta})_{(N+1)s+t-N-1} \tag{59}$$

$$-\left(\mathbf{E}_{p,\eta-1} \boldsymbol{\tau}_{\eta-1}^{(p)} \right)_s \left(\boldsymbol{\varphi}_{\eta,\eta-1}^{(p-1)} \right)_t = (\boldsymbol{\omega}_{p,\eta})_{Ns+t-N} \text{ or } (\boldsymbol{\omega}_{p,\eta})_{(N+1)s+t-N-1} \tag{60}$$

for all s, t in $[1, N]$ or $[1, N+1]$. We also introduce the scalars

$$d'_\eta = \left(d_\eta - \hat{d}_{\eta,\eta-1}^{(p-1)} - \boldsymbol{\psi}_{\eta-1,\eta-1}^{(p-1)} \mathbf{E}_{p,\eta-1} \boldsymbol{\tau}_{\eta-1}^{(p)} \right)^* \qquad d''_\eta = d_\eta - \hat{d}_{\eta,\eta-1}^{(p-1)} \tag{61}$$

where $p \in [0, D)$, $\hat{d}^{(-1)}_{\eta,\eta-1} = 0$, and $\boldsymbol{\psi}^{(-1)}_{\eta-1,\eta-1} = \mathbf{0}$. The $2D$ independently solvable vector equations are

$$\sum_{\eta=0}^{\zeta} \lambda^{\zeta-\eta} d'_\eta \boldsymbol{\omega}_{p,\eta} = \left(\delta\lambda^\zeta \mathbf{I} + \sum_{\eta=0}^{\zeta} \lambda^{\zeta-\eta} \boldsymbol{\omega}_{p,\eta} \boldsymbol{\omega}^H_{p,\eta} \right) \mathbf{c}_{p,\zeta} \tag{62}$$

$$\sum_{\eta=0}^{\zeta} \lambda^{\zeta-\eta} d''_\eta \boldsymbol{\theta}_{p,\eta} = \left(\delta\lambda^\zeta \mathbf{I} + \sum_{\eta=0}^{\zeta} \lambda^{\zeta-\eta} \boldsymbol{\theta}_{p,\eta} \boldsymbol{\theta}^H_{p,\eta} \right) \boldsymbol{\tau}^{(p)}_\zeta. \tag{63}$$

Assuming $r_l = 0$ for all $l \le -\varsigma$ and, optionally, defining $\check{d}_\eta = 0$ for all $\eta < 0$, we apply the RLS algorithm to each system, starting with $\boldsymbol{\Xi}'_{p,0} = \mathbf{I}/\delta$, $\boldsymbol{\Xi}''_{p,0} = \mathbf{I}/\delta$, $\mathbf{c}_{p,0} = \mathbf{0}$, and $\boldsymbol{\tau}^{(p)}_0 = \mathbf{0}$, then iterating as follows.

$$\boldsymbol{\xi}'_{p,\zeta} = \boldsymbol{\Xi}'_{p,\zeta-1} \boldsymbol{\omega}_{p,\zeta} \qquad \boldsymbol{\Xi}'_{p,\zeta} = (\mathbf{I} - \mathbf{k}'_{p,\zeta} \boldsymbol{\omega}^H_{p,\zeta}) \boldsymbol{\Xi}'_{p,\zeta-1}/\lambda \tag{64}$$

$$\boldsymbol{\xi}''_{p,\zeta} = \boldsymbol{\Xi}''_{p,\zeta-1} \boldsymbol{\theta}_{p,\zeta} \qquad \boldsymbol{\Xi}''_{p,\zeta} = (\mathbf{I} - \mathbf{k}''_{p,\zeta} \boldsymbol{\theta}^H_{p,\zeta}) \boldsymbol{\Xi}''_{p,\zeta-1}/\lambda \tag{65}$$

$$\mathbf{k}'_{p,\zeta} = \boldsymbol{\xi}'_{p,\zeta}/(\lambda + \boldsymbol{\omega}^H_{p,\zeta} \boldsymbol{\xi}'_{p,\zeta}) \qquad \mathbf{c}_{p,\zeta} = \mathbf{c}_{p,\zeta-1} + (d'_{p,\zeta} - \boldsymbol{\omega}^H_{p,\zeta} \mathbf{c}_{p,\zeta-1}) \mathbf{k}'_{p,\zeta} \tag{66}$$

$$\mathbf{k}''_{p,\zeta} = \boldsymbol{\xi}''_{p,\zeta}/(\lambda + \boldsymbol{\theta}^H_{p,\zeta} \boldsymbol{\xi}''_{p,\zeta}) \qquad \boldsymbol{\tau}^{(p)}_\zeta = \boldsymbol{\tau}^{(p)}_{\zeta-1} + (d''_{p,\zeta} - \boldsymbol{\theta}^H_{p,\zeta} \boldsymbol{\tau}^{(p)}_{\zeta-1}) \mathbf{k}''_{p,\zeta} \tag{67}$$

6 Equalizers for SIMO Channels

All the equalizers described in the previous sections naturally generalize to exploit spatial diversity in the case of $N_A > 1$ receiving antennas.

For the adaptive RLS algorithms of Sections 3, 4, and 5, the generalization is automatic. We simply reinterpret the case of $N_A N$ samples per symbol as N samples per antenna per symbol. Concretely, in Sections 3, 4, and 5 we replace N with $N_A N$ throughout and, for each symbol time index ζ, antenna index $a \in \mathcal{A} = \{0, \ldots, N_A - 1\}$, and intra-symbol time index $t \in \{0, \ldots, N-1\}$, we interpret $r_{(N\zeta+t)N_A+a}$ as the sample obtained from antenna a at sample time index $N\zeta + t$.

The matched filter of Section 1 also admits a multi-receiver generalization. Proceeding with slightly more care than in the previous paragraph, the received discrete signal r_l of Subsection 1.1 is replaced by N_A received discrete signals

$$r_{(l,a)} = \sum_{m \in \mathcal{G}} g_{(m,a)} z_{l-m} + y_{(l,a)} \tag{68}$$

where $\mathbf{z}$ is as in Subsection 1.1, $\mathbf{g}$ is known, $\mathbf{y}$ and $\mathbf{z}$ are uncorrelated, and $\left\langle y^*_{(k,a)} y_{(l,b)} \right\rangle = \varepsilon_a \delta_{k,l} \delta_{a,b}$ where $\boldsymbol{\varepsilon}$ is known. (In particular, each antenna's additive noise is uncorrelated with the other antennas and has a known power ε_a.) The transveral filter $\mathbf{f}$ of Subsection 1.1 becomes also a linear combiner; data symbols are recovered using

$$\hat{d}_\zeta = \sum_{(n,a) \in \mathcal{F} \times \mathcal{A}} f_{(n,a)} r_{(N\zeta - n, a)}. \tag{69}$$

Again, the problem is to find a coefficient vector $\mathbf{f}$ minimizing $\left\langle \left| \hat{d}_\zeta - d_\zeta \right|^2 \right\rangle$. In analogy with Subsection 1.2, we define

$$\gamma_{(\nu,a)} = \sum \{ g^*_{(p,a)} : p \in \mathcal{G} \cap (-N-\nu, -\nu] \}; \tag{70}$$

$$R_{(\nu,\alpha),(n,a)} = \sum_k \gamma^*_{(\nu+Nk,\alpha)} \gamma_{(n+Nk,a)} \tag{71}$$

$$= \sum \{ g^*_{(\mu,\alpha)} g_{(m,a)} : \mu, m \in \mathcal{G} \text{ and } \lceil (\nu+\mu)/N \rceil = \lceil (n+m)/N \rceil \}. \tag{72}$$

In matrix notation, the MMSE solution is

$$\mathbf{f} = (\mathbf{I} \otimes \mathrm{Diag}\,(\boldsymbol{\varepsilon}) + \mathbf{R})^{-1} \boldsymbol{\gamma} \tag{73}$$

where $\mathbf{I}$ is the $\mathcal{F} \times \mathcal{F}$ identity matrix, $\mathrm{Diag}\,(\boldsymbol{\varepsilon})$ is the diagonal matrix with diagonal $\boldsymbol{\varepsilon}$, and $\mathbf{I} \otimes \mathrm{Diag}\,(\boldsymbol{\varepsilon})$ is the Kronecker product.

To obtain a multi-receiver analog of Section 2, first replace (25) with

$$\hat{d}_\zeta = \sum_{(n,a)\in\mathcal{F}\times\mathcal{A}} f_{(n,a)} r_{(N\zeta-n,a)} - \sum_{\eta\in\mathcal{B}} b_\eta \check{d}_{\zeta-\eta}. \tag{74}$$

Proceeding as in Subsections 2.1 and 2.2, the direct solution is

$$\mathbf{f} = (\mathbf{I}\otimes \mathrm{Diag}\,(\boldsymbol{\varepsilon}) + \mathbf{R} - \boldsymbol{\Gamma}\boldsymbol{\Gamma}^H)^{-1}\boldsymbol{\gamma}; \tag{75}$$

$$\mathbf{b} = \boldsymbol{\Gamma}^H \mathbf{f} \tag{76}$$

where $\Gamma_{(\nu,a),\theta} = \gamma_{(\nu-N\theta,a)}$.

To obtain a multi-receiver analog of the lattice solution of Subsection 1.3, replace $N_0\mathbf{I}$ by $\mathbf{I}\otimes \mathrm{Diag}\,(\boldsymbol{\varepsilon})$ and expand each coordinate of every vector and matrix into a block of length N_A. For examples,

$$\boldsymbol{\gamma}^{[\alpha,\beta]} = \begin{bmatrix} \gamma_{(N\alpha,0)} \\ \vdots \\ \gamma_{(N\alpha,N_A-1)} \\ \gamma_{(N\alpha+1,0)} \\ \vdots \\ \gamma_{(N\alpha+1,N_A-1)} \\ \vdots \\ \gamma_{(N\beta+N-1,0)} \\ \vdots \\ \gamma_{(N\beta+N-1,N_A-1)} \end{bmatrix}; \qquad T_{(p,a),(q,b)} = \varepsilon_a \delta_{a,b}\delta_{p,q} + R_{(p,a),(q,b)}. \tag{77}$$

To obtain a multi-receiver analog the lattice solution of Subsection 2.3, again expand each coordinate of every vector and matrix into a block of length N_A, except for coordinates corresponding to the decision feedback.

References

[1] H. Akaike. *Block Toeplitz Matrix Inversion.* SIAM J. Appl. Math. **24(2)** (1973), 234–241.

[2] P.M. Grant and M.J. Rutter. *Application of gradient adaptive lattice filters to channel equalisation.* IEE Proceedings F (Communications, Radar and Signal Processing) **131(5)** (August 1984), 473–479.

[3] S. Haykin. *Adaptive Filter Theory,* 4^{th} ed. Prentice Hall Information and System Sciences Series (2002). ISBN 0-13-090126-1.

[4] P. Monsen. *Feedback Equalization for Fading Dispersive Channels.* IEEE Transactions on Information Theory (January 1971), 56–64.