FAST PARAMETRIC MATRICES FOR LOSSLESS FEEDBACK DELAY NETWORKS

Andrea Coppola

Arturia
Montbonnot-Saint-Martin, France
andrea.coppola@arturia.com

ABSTRACT

This paper presents a framework for designing creative reverbs using parametric orthogonal feedback matrices on Feedback Delay Networks (FDNs) through recursive Kronecker products of 2D rotation and reflection matrices. By parameterizing each 2×2 kernel with a single angle, we construct a family of $2^M \times 2^M$ orthogonal matrices that maintain losslessness while enabling continuous control over network topology. We then exploit their recursive definition to compute the feedback operation with an $O(N \log_2 N)$ divide-and-conquer algorithm that matches the Fast Walsh-Hadamard Transform time complexity while offering parametric flexibility. Strategic manipulation of individual kernel angles enables creative sound design applications, such as stereo cross-coupling, selective freeze, and time-varying modulation for resonance breaking. The framework offers FDN designers an expressive, real-time tunable parametric matrix, balancing computational efficiency with intuitive control for creative reverberation design.

1. INTRODUCTION

Artificial reverberation is a widely used audio effect in music production and sound design. Its applications range from realistic acoustic space emulation to the creative use of unnatural spaces and sonic ambiences as expressive narrative elements [1].

This paper focuses on the parameterization of Feedback Delay Networks (FDNs) for constructing creative reverbs that prioritize controllable sonic textures over physical realism. Such creative reverberators require higher-level feature controls that enable intuitive manipulation of perceptually relevant characteristics, which we achieve through strategic parameterization of the feedback matrix. When designing FDN reverberators, the feedback matrix choice plays a crucial role in determining the diffusion characteristics and overall sonic quality of the reverberations [2]. Furthermore, the ability to efficiently compute the feedback operation is essential for real-time audio processing applications, particularly when parameter modulation is desired [3].

We propose a parametric orthogonal feedback matrix built from recursive Kronecker products of 2×2 rotation and reflection kernels, controlled by M independent angles exposed as real-time parameters. The recursive structure admits an in-place $O(N \log_2 N)$ algorithm for the feedback operation at network size $N = 2^M$, matching the asymptotic efficiency of the Fast Walsh-Hadamard Transform (FWHT) while supporting continuous angle modulation. Specific configurations of these angles induce structurally meaningful partitioning of the network, which we exploit for three creative applications: *stereo cross-coupling* (Section 5.2), where two sub-networks are coupled to control the stereo image; *selective freeze* (Section 5.3), where part of the network is isolated so it can be sustained indefinitely while the rest decays; and *time-varying modulation* (Section 5.4), where the angles are varied continuously to break up fixed-frequency resonances and reduce spectral coloration.[1]

2. RELATED WORK

The use of Kronecker products to construct FDN feedback matrices is not new. Das, Canfield-Dafilou, and Abel [4] analyzed the modal behavior of a Kronecker-parameterized mixing matrix under continuous morphing from identity toward Hadamard, parameterized by a single angle applied uniformly to all kernels, characterizing how the poles move in the complex plane. Das, Abel and Canfield-Dafilou [5, 6] later used Kronecker structure to model coupled rooms via grouped FDNs. In both cases the Kronecker form serves a fixed structural choice tailored to a specific analytical or modeling purpose. The present work treats each of the M kernel angles as an independent real-time control, unlocking specific structural effects on the network, and exploits the resulting matrix family to build MIMO configurations such as stereo FDNs with controllable cross-coupling, network isolation for selective freeze, and time-variation for resonance breaking.

Recent work has framed feedback-matrix selection as a numerical optimization problem. Dal Santo et al. [7] optimize FDN parameters for smooth, colorless reverberation, constructing the feedback matrix via the matrix exponential of a skew-symmetric matrix derived from the upper triangular part of a parameter matrix, while Hörngren [8] treats a Kronecker-parameterized family as the search space for offline optimization. Both approaches fix the resulting parameters once the solver terminates against a chosen objective. The present work instead uses the kernel angles to expose and exploit specific couplings between network subsections, enabling creative effects rather than optimizing toward a single objective.

Time-varying feedback matrices have been formally studied by Schlecht and Habets [9, 10], who showed that continuous modulation of the feedback matrix can spread the system poles and reduce spectral coloration, while also highlighting the practical difficulty of preserving orthogonality under modulation. Das et al. [4] likewise observed mode movement in the complex plane under continuous variation of a Kronecker-parameterized mixing matrix. In our framework, continuous angle modulation is an intrinsic property of the parameterization: orthogonality is preserved by

[1] Audio examples accompanying this paper are available at https://andrea-coppola-arturia.github.io/fastparametricmatrices/.

construction at every value of every angle, with no orthogonalization step required. We therefore verify in Section 5 that the pole-spreading and coloration-reducing effects reported in [10, 4] carry over to our parametric matrices. A range of orthogonal feedback-matrix constructions has been catalogued together with their per-sample computational costs in [11]. Each entry in their table corresponds to a fixed point in the design space of orthogonal matrices. The Kronecker family proposed here instead occupies a region of this design space that prior FDN constructions do not, combining three properties:

1. **Fast:** The feedback operation costs $O(N \log_2 N)$, matching the Hadamard matrix under the FWHT. Yet unlike the fixed Hadamard matrix, every kernel angle remains a free parameter.
2. **Continuously tunable:** Those angles can be modulated at audio rate, and changing one updates a single 2×2 kernel rather than rebuilding the full matrix.
3. **Structurally expressive:** Setting or sweeping individual angles partitions the network into meaningful subnetworks, exposing the partitioning as a continuous real-time control for creative effects such as stereo cross-coupling, selective freeze, and modulation.

3. BACKGROUND

3.1. MIMO Feedback Delay Network

Given input and output vectors defined respectively as $\mathbf{x}(n) \in \mathbb{R}^{N_{\text{in}}}$, $\mathbf{y}(n) \in \mathbb{R}^{N_{\text{out}}}$, we define a Feedback Delay Network of size N with the following finite difference state space equation [12]:

$$\begin{aligned} \mathbf{y}(n) &= \mathbf{C}\mathbf{s}(n) + \mathbf{D}\mathbf{x}(n), \\ \mathbf{s}(n+\mathbf{m}) &= \mathbf{A}\mathbf{s}(n) + \mathbf{B}\mathbf{x}(n), \end{aligned} \tag{1}$$

where $\mathbf{B} \in \mathbb{R}^{N \times N_{\text{in}}}$ represents the *input gain* matrix, $\mathbf{C} \in \mathbb{R}^{N_{\text{out}} \times N}$ the *output gain* matrix, $\mathbf{A} \in \mathbb{R}^{N \times N}$ the *feedback matrix* and $\mathbf{D} \in \mathbb{R}^{N_{\text{out}} \times N_{\text{in}}}$ is the *direct gain* matrix. The FDN size N represents the number of delay lines whose delay in samples is given by the vector $\mathbf{m} = [m_1, \ldots, m_N]$, and whose outputs are denoted by the vector $\mathbf{s}(n+\mathbf{m}) = [s_1(n+m_1), \ldots, s_N(n+m_N)]$. A block diagram of the MIMO FDN structure is depicted in Figure 1.

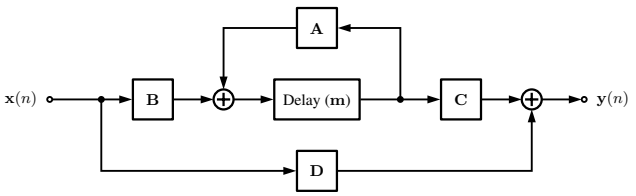

Figure 1: MIMO Feedback Delay Network.

The transfer function matrix of this structure is given by the following equation:

$$\mathbf{H}(z) = \mathbf{C}(\mathbf{D}_{\mathbf{m}}(z^{-1}) - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \tag{2}$$

where $\mathbf{D}_{\mathbf{m}}(z^{-1}) = \text{diag}([z^{-m_1}, \ldots, z^{-m_N}])$ is the diagonal $N \times N$ delay matrix [12]. The poles of the transfer function are given by the roots of the generalized characteristic polynomial $p_{\mathbf{A},\mathbf{m}}(z)$ defined as:

$$p_{\mathbf{A},\mathbf{m}}(z) = \det[\mathbf{D}_{\mathbf{m}}(z^{-1}) - \mathbf{A}]. \tag{3}$$

An FDN is said to be lossless (therefore energy conserving) if its $p_{\mathbf{A},\mathbf{m}}(z)$ has only unimodular roots [13].

3.2. Unilossless Feedback Matrix

When designing creative artificial reverbs, it is desirable to choose a feedback matrix that provides homogeneous, frequency-independent decay across all delay lines [14]. Starting from this neutral baseline allows us to later craft musically meaningful frequency-dependent decays by introducing absorption coefficients and attenuation filters in the feedback loop. This provides us with independent control over the decay profile without affecting the underlying mixing structure.

A feedback matrix $\mathbf{A}$ is said to be *unilossless* if, for any choice of delay vector $\mathbf{m} \in \mathbb{N}^N$, the FDN is lossless, and therefore $p_{\mathbf{A},\mathbf{m}}(z)$ has only unimodular roots. This property ensures energy conservation in the absence of absorption, regardless of the specific delay line lengths chosen. More formally, a matrix $\mathbf{A}$ is unilossless if and only if either:

(I) $\mathbf{A}$ is reducible and all irreducible components are unilossless, or

(II) there exists a non-singular diagonal matrix $\mathbf{E}$ with $\mathbf{A}\mathbf{E}\mathbf{A}^H = \mathbf{E}$ [13].

A specific subset of unilossless feedback matrices is given by complex-valued *unitary* matrices $\mathbf{U} \in \mathbb{C}^{N \times N}$, for which the following property holds:

$$\mathbf{U}\mathbf{U}^H = \mathbf{I}_N \tag{4}$$

where $(\cdot)^H$ denotes the Hermitian conjugate (complex-conjugate transpose), and $\mathbf{I}_N$ denotes the $N \times N$ identity matrix. Any unitary matrix satisfies the unilossless condition (II) when $\mathbf{E} = \mathbf{I}_N$, which guarantees unimodular roots of $p_{\mathbf{A},\mathbf{m}}(z)$ independent of the delay configuration [14].

In this paper, we focus exclusively on real-valued unitary matrices, known as orthogonal matrices. This restriction allows us to construct feedback matrices using only real-valued elements, which is more useful for practical audio processing applications.

To achieve controllable decay, we factorize the feedback matrix $\mathbf{A}$ as a product of an orthogonal matrix $\mathbf{U}$ and a diagonal absorption matrix $\mathbf{\Gamma}$:

$$\mathbf{A} = \mathbf{U}\mathbf{\Gamma} \tag{5}$$

where $\mathbf{\Gamma} = \text{diag}(\gamma_1, \ldots, \gamma_N) \in \mathbb{R}^{N \times N}$ is a diagonal matrix whose entry $\gamma_i \leq 1$ represents the absorption coefficient applied to delay line i.

For a desired reverberation time T_{60}, we define each absorption coefficient as:

$$\gamma_i(T_{60}) = 10^{-\frac{3m_i}{T_{60} f_s}} \tag{6}$$

with m_i the delay length in samples for the i-th delay line and f_s the sample rate [15, 16]. This equation ensures that each delay line contributes proportionally to the overall decay time based on its length.

3.3. Kronecker Product

The Kronecker product provides a systematic way to construct larger matrices from smaller building blocks [17]. Given a matrix

$\mathbf{V} \in \mathbb{R}^{m\times n}$ and a matrix $\mathbf{W} \in \mathbb{R}^{p\times q}$, their Kronecker product $\mathbf{V} \otimes \mathbf{W} \in \mathbb{R}^{mp\times nq}$ is defined as:

$$\mathbf{V} \otimes \mathbf{W} = \begin{bmatrix} (\mathbf{V})_{11}\mathbf{W} & \cdots & (\mathbf{V})_{1n}\mathbf{W} \\ \vdots & \ddots & \vdots \\ (\mathbf{V})_{m1}\mathbf{W} & \cdots & (\mathbf{V})_{mn}\mathbf{W} \end{bmatrix}, \tag{7}$$

where $(\mathbf{V})_{ij}$ denotes the (i, j)-th entry of $\mathbf{V}$.

An important property for our application is that the Kronecker product of two orthogonal matrices is itself orthogonal [17], enabling recursive construction of higher-dimensional orthogonal matrices from smaller components.

3.4. Parametric 2D Orthogonal Matrices

To construct parametric orthogonal matrices using Kronecker products, we require smaller building blocks that offer continuous parametric control. The most elementary building blocks of this kind are 2×2 orthogonal matrices.

Any orthogonal matrix in $\mathbb{R}^{2\times 2}$ represents either a rotation or a reflection in two-dimensional space [18]. These transformations can be parameterized by a single angle θ, providing a compact way to describe the entire family of 2D orthogonal matrices.

A rotation matrix in two dimensions is given by:

$$\mathrm{Rot}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{8}$$

which rotates vectors counterclockwise by an angle θ.

A reflection matrix in two dimensions is given by:

$$\hat{\mathrm{Ref}}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix}, \tag{9}$$

which reflects vectors across a line oriented at angle $\theta/2$ from the horizontal axis.[2]

When setting the angle $\theta = 0$, the rotation matrix reduces to the identity matrix

$$\mathrm{Rot}(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

while the reflection matrix becomes a simple axis reflection

$$\hat{\mathrm{Ref}}(0) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

These parameterizations will serve as the elementary building blocks for constructing higher-dimensional feedback matrices as detailed in Section 4.

4. PARAMETRIC FEEDBACK MATRIX

Building on the previous section, we now present a method for constructing $2^M \times 2^M$ orthogonal feedback matrices via recursive Kronecker products. This recursive structure enables an $O(N \log_2 N)$ algorithm for the feedback operation, compared to $O(N^2)$ for direct matrix multiplication.

[2]While reflection matrices are traditionally denoted as $\begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix}$, we adopt a version with a half angle of reflection to maintain a consistent parameterization throughout the framework.

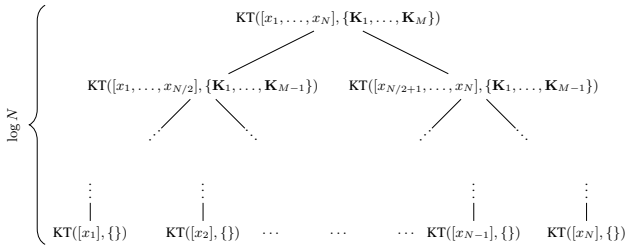

Figure 2: Recursive call tree for the Recursive Kronecker Transform algorithm, KT in the diagram.

4.1. Kronecker Feedback Matrix

Similarly to [4], starting from M orthogonal matrices $\mathbf{K}_i \in \mathbb{R}^{2\times 2}$ for $i = 1, \ldots, M$, referred to as *kernels*, we define the Kronecker Feedback Matrix $\mathbf{\Psi}_M \in \mathbb{R}^{2^M\times 2^M}$ as:

$$\mathbf{\Psi}_M = \mathbf{K}_M \otimes \mathbf{K}_{M-1} \otimes \ldots \otimes \mathbf{K}_1. \tag{10}$$

This matrix can also be defined recursively as:

$$\begin{aligned} \mathbf{\Psi}_0 &= \mathbf{I}_1, \\ \mathbf{\Psi}_M &= \mathbf{K}_M \otimes \mathbf{\Psi}_{M-1}, \end{aligned} \tag{11}$$

where $\mathbf{I}_1$ represents the scalar identity as the base case of recursion. The orthogonality of $\mathbf{\Psi}_M$ follows directly from the property that Kronecker products of orthogonal matrices preserve orthogonality [17]. When we choose the kernels $\mathbf{K}_i$ as the rotation or reflection matrices defined in Section 3.4, each kernel contributes a single angle parameter θ_i, resulting in an M-parameter family of $2^M \times 2^M$ orthogonal matrices.

We will explore various choices of these parameters in Section 5 to achieve different network topologies.

4.2. Fast Matrix-Vector Multiplication

Standard matrix-vector multiplication for an $N \times N$ matrix requires $O(N^2)$ operations. Since this operation must be performed per sample in the feedback loop, the computational cost can be significant.

The recursive structure of the Kronecker Feedback Matrix can be exploited to derive a divide-and-conquer algorithm [19] that computes the matrix-vector product in $O(N \log_2 N)$ time instead, enabling efficient real-time processing. To do this, we define a recursive algorithm that performs matrix-vector multiplication exploiting the matrix composition of equation (11) in Algorithm 1.

Figure 2 illustrates the recursive structure for an input of size $N = 2^M$. Each call halves the vector, giving $\log_2(N)$ levels of recursion. Each level processes every sample exactly once, giving $O(N)$ operations per level. Combined, this yields a total complexity of $O(N \log_2 N)$.

While the recursive formulation clearly illustrates the relationship to the Kronecker product structure and facilitates complexity analysis, it is less suitable for real-time audio processing. Recursive implementations incur function call overhead and stack management costs that can impact performance. Additionally, the recursive version creates temporary array copies at each level. Since any recursive algorithm admits an equivalent iterative form [19], we propose the in-place, iterative implementation in Algorithm 2, which avoids these overheads.

Algorithm 1 Recursive Kronecker Transform

function RECURSIVEKT($\mathbf{x}, \{\mathbf{K}_1, \ldots, \mathbf{K}_M\}$)
  // Size of input vector and Kernels
  $n \leftarrow |\mathbf{x}|$

  // Base case: single element
  if $n = 1$ **then**
    return $\mathbf{x}$
  end if

  // Split input in two
  $m \leftarrow n/2$
  $\mathbf{x}_a \leftarrow \mathbf{x}[1 : m]$
  $\mathbf{x}_b \leftarrow \mathbf{x}[m + 1 : n]$

  // Current level matrix
  $\mathbf{K} \leftarrow \mathbf{K}_M$

  // Apply matrix multiplication
  $\mathbf{y}_a \leftarrow (\mathbf{K})_{11} \cdot \mathbf{x}_a + (\mathbf{K})_{12} \cdot \mathbf{x}_b$
  $\mathbf{y}_b \leftarrow (\mathbf{K})_{21} \cdot \mathbf{x}_a + (\mathbf{K})_{22} \cdot \mathbf{x}_b$

  // Recursion step
  $\mathbf{y}_a \leftarrow$ RECURSIVEKT$(\mathbf{y}_a, \{\mathbf{K}_1, \ldots, \mathbf{K}_{M-1}\})$
  $\mathbf{y}_b \leftarrow$ RECURSIVEKT$(\mathbf{y}_b, \{\mathbf{K}_1, \ldots, \mathbf{K}_{M-1}\})$

  // Concatenate results and return
  return $[\mathbf{y}_a\,;\,\mathbf{y}_b]$
end function

The iterative algorithm cycles through the $M = \log_2(N)$ recursive levels with the outer loop. Then, for each level, it splits the input vector into the corresponding number of subdivisions (matching the number of recursive calls at that level) with the middle loop. Finally, the innermost loop applies the kernel to each pair of samples within the subdivision in-place, avoiding temporary arrays and function call overheads. This results in a more efficient implementation suitable for real-time audio processing. Each of the $\log_2 N$ levels processes every sample at constant cost, yielding the $O(N \log_2 N)$ total complexity. Moreover, since the matrix itself is never explicitly constructed, the algorithm requires only $O(N)$ memory for the input/output vector plus $O(\log_2 N)$ memory for storing the M kernel matrices, compared to $O(N^2)$ for storing the full feedback matrix. This translates also into reduced cache usage and improved performance for large networks.

To validate this complexity in an implementation-specific use case, we present a benchmark of time per operation comparing the proposed algorithm against a naive matrix-vector multiplication implemented with nested loops and against Eigen. Each reported value is the mean computation time per operation over 5×10^6 timed iterations, preceded by 5×10^5 warm-up iterations. Table 1 reports the results for a fixed feedback matrix.

The advantage grows further when the kernel angles vary over time, an essential condition for the parametric applications presented in Section 5. Table 2 reports the corresponding cost when one angle is updated at every sample. A matrix-multiplication approach must rebuild the feedback matrix whenever any θ_i changes, while the proposed algorithm only refreshes the affected 2×2 kernel.

Algorithm 2 In-Place Iterative Kronecker Transform

function ITERATIVEKT($\mathbf{x}, \{\mathbf{K}_1, \ldots, \mathbf{K}_M\}$)
  // Size of input vector
  $n \leftarrow |\mathbf{x}|$

  // For each level
  for $c = 1$ **to** M **do**
    // Current level matrix
    $\mathbf{K} \leftarrow \mathbf{K_c}$

    // Number of elements in sub-division
    $h \leftarrow 2^{c-1}$

    // Number of subdivisions
    $m \leftarrow n/(2h)$

    // For each subdivision
    for $i = 1$ **to** m **do**
      beg $\leftarrow (i - 1) \cdot 2h + 1$

      // For each couple of elements in sub-division
      for $j = 0$ **to** $h - 1$ **do**

        // Get elements in couples
        $a \leftarrow x_{\text{beg}+j}$
        $b \leftarrow x_{\text{beg}+j+h}$

        // Apply matrix multiplication
        $x_{\text{beg}+j} \leftarrow (\mathbf{K})_{11} \cdot a + (\mathbf{K})_{12} \cdot b$
        $x_{\text{beg}+j+h} \leftarrow (\mathbf{K})_{21} \cdot a + (\mathbf{K})_{22} \cdot b$
      end for
    end for
  end for
end function

The Kronecker structure gives us parametric control through the kernel angles and $O(N \log_2 N)$ efficiency through the divide-and-conquer algorithm. This combination enables real-time exploration of different FDN topologies.

5. NETWORK PARAMETRIZATION

Having introduced the framework for Kronecker Feedback Matrices and their efficient computation, we now explore the range of behaviors achievable through different kernel parameterizations. By choosing specific kernel types and values for their angles θ_i, it is possible to obtain well-known structures such as Hadamard matrices, enable novel features like stereo field control and independent network freezing, or continuously morph the matrix to break resonances, as discussed in the following.

5.1. Hadamard

In FDN design, the Hadamard matrix is a widely adopted feedback matrix due to its properties of orthogonality, diffusion, and computational efficiency [20]. A Hadamard matrix of order 2 is defined as:

$$\mathbf{H}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{12}$$

Table 1: Average time per operation (ns) for applying a fixed $N \times N$ orthogonal feedback matrix. Naive: matrix-vector multiplication with nested loops; Eigen: Eigen 5.0.1; Proposed: Algorithm 2. C++17 implementation compiled with MSVC, running on an Intel Core Ultra 9 185H.

	Time (ns)			Speedup	
N	Naive	Eigen	Proposed	N/P	E/P
4	4.3	2.5	5.4	0.80×	0.46×
8	13.4	15.7	12.8	1.05×	1.23×
16	54.9	27.1	26.9	2.04×	1.01×
32	233.2	48.5	31.0	7.54×	1.57×
64	1662.5	151.2	58.4	28.47×	2.59×

Table 2: Average time per operation (ns) with one kernel angle updated at every sample. The two matrix-multiplication baselines (Naive and Eigen) must rebuild the dense $N \times N$ matrix from the $\log_2 N$ kernel angles at each step while the Proposed algorithm refreshes only the modified 2×2 kernel. Setup as in Table 1.

	Time (ns)			Speedup	
N	Naive	Eigen	Proposed	N/P	E/P
4	31.7	35.0	15.2	2.08×	2.30×
8	75.4	82.1	19.5	3.87×	4.21×
16	240.5	219.6	33.2	7.25×	6.62×
32	914.8	722.9	37.8	24.19×	19.12×
64	4417.3	2862.5	69.5	63.60×	41.21×

while higher order Hadamard matrices $\mathbf{H}_{2^M}$ of order 2^M can be recursively defined as:

$$\mathbf{H}_{2^M} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{H}_{2^{M-1}} & \mathbf{H}_{2^{M-1}} \\ \mathbf{H}_{2^{M-1}} & -\mathbf{H}_{2^{M-1}} \end{bmatrix} = \mathbf{H}_2 \otimes \mathbf{H}_{2^{M-1}}. \quad (13)$$

As rightly noted in [4], it can be shown that the Hadamard matrix of order 2 is a special case of (9) when angle $\theta = \frac{\pi}{4}$:

$$\hat{\mathrm{Ref}}\left(\frac{\pi}{4}\right) = \begin{bmatrix} \cos\frac{\pi}{4} & \sin\frac{\pi}{4} \\ \sin\frac{\pi}{4} & -\cos\frac{\pi}{4} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (14)$$

Setting all kernels to $\mathbf{K}_i = \hat{\mathrm{Ref}}\left(\frac{\pi}{4}\right)$ for $i = 1, \ldots, M$ yields $\mathbf{\Psi}_M = \mathbf{H}_{2^M}$, the Hadamard matrix of order 2^M.

Alternatively, a rotation matrix with angle $\theta = \frac{\pi}{4}$:

$$\mathrm{Rot}\left(\frac{\pi}{4}\right) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad (15)$$

also provides equal-magnitude entries and thus uniform energy distribution. Using $\mathrm{Rot}\left(\frac{\pi}{4}\right)$ as the Kronecker kernel $\mathbf{K}_i$ yields an orthogonal feedback matrix with equal-energy mixing properties. Our divide-and-conquer algorithm computes the Hadamard matrix-vector product with the same $O(N \log_2 N)$ complexity as the Fast Walsh-Hadamard Transform (FWHT) [21]. While the FWHT can be further optimized by avoiding multiplications when factoring out the scalar normalization $\frac{1}{\sqrt{2}}$, our parametric framework offers the ability to continuously vary the kernel angles away from $\theta_i = \frac{\pi}{4}$ while maintaining orthogonality. This enables smooth morphing between the Hadamard structure and other topologies, as demonstrated in the following subsections.

5.2. Stereo Configuration via Parametric Decoupling

By strategically choosing the kernel parameters, we can partition the Kronecker Feedback Matrix into a block structure that enables independent processing of stereo channels with controllable cross-feedback. Similarly to Das, Abel and Canfield-Dafilou [6, 5], who used Kronecker products to model coupled acoustic spaces, we exploit this block-partitioned configuration for MIMO stereo FDNs. Consider an FDN of size $N = 2^M$ where we split the delay lines into two groups of size $N/2$ each. When a parametric angle is set to $\theta_i = 0$, the corresponding kernel becomes a diagonal orthogonal matrix. More specifically, when the angle θ_M of the outermost kernel $\mathbf{K}_M$ is set to 0, the feedback matrix $\mathbf{\Psi}_M$ assumes a block-diagonal structure:

$$\mathbf{\Psi}_M = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0}_{N/2} \\ \mathbf{0}_{N/2} & \mathbf{A}_2 \end{bmatrix}, \quad (16)$$

where $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{N/2 \times N/2}$ are independent feedback matrices for each partition, corresponding to the first and second halves of the delay lines, and $\mathbf{0}_{N/2}$ denotes an $N/2 \times N/2$ zero matrix. As the angle θ_M is varied away from zero, off-diagonal coupling blocks emerge:

$$\mathbf{\Psi}_M = \begin{bmatrix} \mathbf{A}_1 & \mathbf{F}_1 \\ \mathbf{F}_2 & \mathbf{A}_2 \end{bmatrix}, \quad (17)$$

where $\mathbf{F}_1, \mathbf{F}_2 \in \mathbb{R}^{N/2 \times N/2}$ represent cross-feedback matrices that couple the two partitions. When the angle θ_M reaches $\frac{\pi}{2}$, the matrix becomes an anti-diagonal block matrix:

$$\mathbf{\Psi}_M = \begin{bmatrix} \mathbf{0}_{N/2} & \mathbf{F}_1 \\ \mathbf{F}_2 & \mathbf{0}_{N/2} \end{bmatrix}. \quad (18)$$

For stereo processing, we configure the input and output gain matrices to route left and right channels to their respective partitions. We define the stereo input vector $\mathbf{x}(n) = [x_L(n), x_R(n)]^T$ and output vector $\mathbf{y}(n) = [y_L(n), y_R(n)]^T$. The input gain matrix is structured as:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{0}_{N/2 \times 1} \\ \mathbf{0}_{N/2 \times 1} & \mathbf{B}_2 \end{bmatrix}, \quad (19)$$

where $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{R}^{N/2 \times 1}$ distribute the left and right inputs to the first and second halves of the delay lines, respectively, with zero blocks preventing cross-channel mixing. Similarly, the output gain matrix assumes the following form:

$$\mathbf{C}^T = \begin{bmatrix} \mathbf{C}_1 & \mathbf{0}_{N/2 \times 1} \\ \mathbf{0}_{N/2 \times 1} & \mathbf{C}_2 \end{bmatrix}, \quad (20)$$

where $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{R}^{N/2 \times 1}$ extract the outputs for left and right channels respectively.

Figure 3 illustrates this stereo configuration with two coupled FDN partitions. The kernel angle θ_M thus becomes a continuous parameter controlling the cross-coupling strength between channels. At $\theta_M = 0$, the two networks are fully decoupled ($\mathbf{F}_1 = \mathbf{F}_2 = \mathbf{0}_{N/2}$), producing independent channel processing with no cross-feedback. As θ_M increases, cross-coupling between channels gradually increases through the emergence of non-zero $\mathbf{F}_1$ and $\mathbf{F}_2$ matrices. At $\theta_M = \pi/4$, the matrix approaches full coupling, behaving as a single unified FDN. This parametric approach enables smooth, artifact-free transitions between stereo imaging modes.

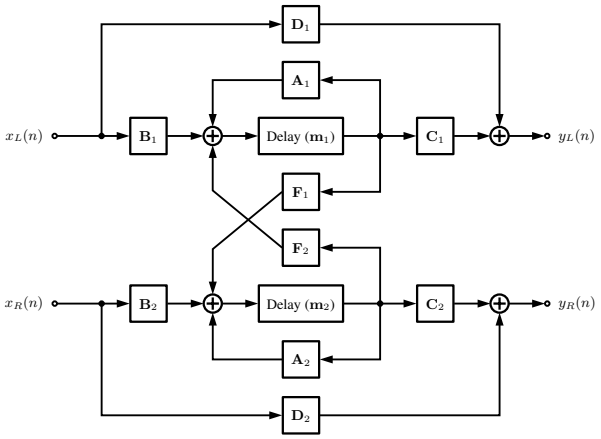

Figure 3: Stereo Configuration of FDN.

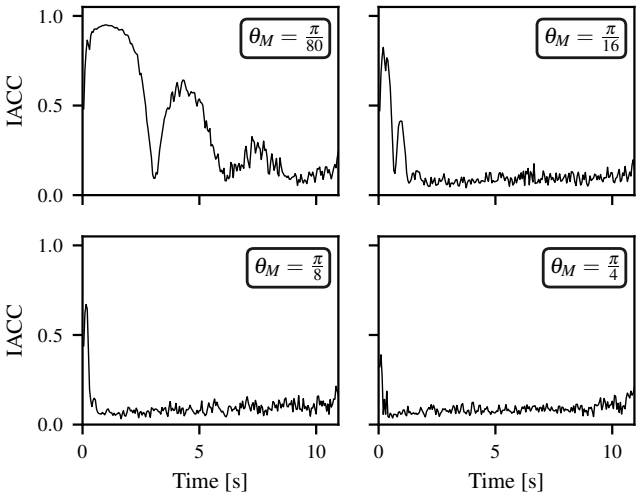

Figure 4: Sliding-window IACC (window length 100 ms, hop 50 ms) for a stereo FDN (32 channels, rotation kernels, $T_{60} = 10\,\mathrm{s}$) at four values of θ_M, which controls the cross-coupling from fully decoupled ($\theta_M = 0$) to maximally coupled ($\theta_M = \pi/4$). The impulse response is generated by feeding a unit impulse to the left input only. At $\theta_M = 0$ the right channel remains silent, so the IACC is undefined and that case is omitted.

To quantify the inter-channel correlation, we use the interaural cross-correlation coefficient (IACC) [22]. Over a window $[t, t+T]$, the normalized cross-correlation of the left and right impulse responses h_L, h_R is

$$\rho_t(\tau) = \frac{\int_t^{t+T} h_L(s)\, h_R(s+\tau)\,\mathrm{d}s}{\sqrt{\int_t^{t+T} h_L^2(s)\,\mathrm{d}s \int_t^{t+T} h_R^2(s)\,\mathrm{d}s}}, \tag{21}$$

and the IACC coefficient is its peak magnitude over the $\pm 1\,\mathrm{ms}$ lag range, $\mathrm{IACC}(t) = \max_{|\tau| \le 1\,\mathrm{ms}} |\rho_t(\tau)|$. As the impulse responses are discrete, the integrals are evaluated as the corresponding windowed sums over the sampled signals. Sliding the window yields the time-varying coefficient, which reveals how the stereo image evolves as the response decays.

As we can observe in the example plots of Figure 4, the IACC coefficient between the left and right channels varies with θ_M, demonstrating the ability to control stereo image and spatial characteristics through this coupling mechanism. Of particular interest is the case of light cross-coupling, which creates a rich spatial effect where the impulse bounces from one channel to the other over time, as observable from both the IACC plot at low coupling values and from the audio examples in the supplementary materials. By contrast, increasing the coupling gradually unifies the two channels in a single network, creating a wide and decorrelated stereo image, as shown in the plots and demonstrated in the provided audio examples. This is consistent with the fact that the network becomes a single unified FDN with high diffusion properties as θ_M approaches $\pi/4$. Together, the audio examples demonstrate the coupling sweep from $\theta_M = 0$ (fully decoupled) to $\theta_M = \pi/4$ (unified FDN).

5.3. Network Decoupling and Selective Freeze

The stereo configuration presented in the previous section demonstrates decoupling through $\theta_M = 0$, which partitions the network into two contiguous halves of size $N/2$ each. However, this is not the only form of decoupling available in the framework. Setting any kernel angle $\theta_i = 0$ always creates a partition into two independent blocks of size $N/2$ each, but the grouping pattern depends on which kernel position is chosen. For instance, setting $\theta_M = 0$ creates two contiguous halves, while choosing lower kernel indices produces increasingly fine-grained interleaving patterns.

Of particular interest is the case where $\theta_1 = 0$. As can be observed in Figure 5, this choice affects the finest level of the recursive Kronecker product structure, creating an interleaved pattern that partitions the state vector into even-indexed and odd-indexed delay lines. Each subvector evolves independently with no cross-coupling between the two groups. When delay lengths are chosen in ascending order, the even-odd partitioning distributes the delay range across both subnetworks. This ensures that both partitions maintain balanced echo density characteristics, unlike other types of partitioning which would separate short and long delays into distinct groups with different temporal behaviors.

One application of this even-odd decoupling is enabling selective *freeze* operations on independent subsets of delay lines. A *freeze* operation in an FDN is achieved by setting the input gain matrix to zero ($\mathbf{B} = \mathbf{0}_{N \times N_{\text{in}}}$) and the absorption coefficients to unity ($\mathbf{\Gamma} = \mathbf{I}_N$), making the network lossless and input-free, thereby sustaining the current state indefinitely. By decoupling the network into even and odd delay lines, we can apply a *freeze* operation independently to each subset by interleaving zeroes in $\mathbf{B}$ and ones in $\mathbf{\Gamma}$. This allows us to freeze the even-indexed lines while allowing the odd-indexed lines to continue processing new input and vice versa. This decoupling is essential since it allows us to render lossless and input-free only a portion of the network. If we interleave the inputs in $\mathbf{B}$ and the absorption coefficients in $\mathbf{\Gamma}$ without decoupling, the marginally stable section of the network would be coupled with the rest of the system, causing buildup of energy in the frozen section from the input coming through the cross-coupling paths. This provides fine-grained control over the reverb tail, enabling creative effects such as sustaining specific parts of the network while others decay naturally or freezing two independent parts of the input to create layered textures. The kernel angle θ_1 enables smooth transitions between coupled and decoupled states, varying from entirely decoupled with independent even/odd subsystems when $\theta_1 = 0$ to fully coupled with

$\theta_1 = \pi/4$. This smooth interpolation avoids abrupt discontinuities in the feedback matrix that would otherwise introduce audible artifacts when transitioning between freeze modes. The supplementary materials demonstrate independent freeze operations on even and odd partitions and illustrate as well a creative use case in which two parallel freezes capture short pulses from an evolving loop to assemble sustained pad sounds.

	1	2	3	4	5	6	7	8
1	+0.65	0	-0.65	0	-0.27	0	+0.27	0
2	0	+0.65	0	-0.65	0	-0.27	0	+0.27
3	+0.65	0	+0.65	0	-0.27	0	-0.27	0
4	0	+0.65	0	+0.65	0	-0.27	0	-0.27
5	+0.27	0	-0.27	0	+0.65	0	-0.65	0
6	0	+0.27	0	-0.27	0	+0.65	0	-0.65
7	+0.27	0	+0.27	0	+0.65	0	+0.65	0
8	0	+0.27	0	+0.27	0	+0.65	0	+0.65

Figure 5: Feedback matrix $\mathbf{\Psi}_3$ for $N = 8$ with rotation kernels at $\theta_1 = 0$, $\theta_2 = \pi/4$, and $\theta_3 = \pi/8$. Setting $\theta_1 = 0$ decouples the network into even and odd-indexed delay lines, while $\theta_3 = \pi/8$ retains a partial stereo coupling between the two halves, showing that the even/odd freeze and stereo cross-coupling configurations are compatible and can be combined within a single matrix. The remaining angle θ_2 is left free for time-varying modulation.

5.4. Modulation

Beyond these configurations, the parametric nature of the kernel angles enables dynamic modulation of the feedback matrix. As observed in [4], continuous variation of a kernel angle causes the FDN modes to move in the complex plane. We exploit this behavior by varying one or more kernel angles θ_i over time, preventing the buildup of resonances at fixed frequencies and increasing the perceptual liveliness of the reverberation tail [14]. To analyze the effect of modulation on pole distribution, we replicate the histogram of [10] as shown in Figure 6. Without modulation, poles remain fixed and produce sharp peaks in the histogram, corresponding to resonant frequencies that cause spectral coloration. As modulation is introduced, the pole angles start to move, spreading the histogram toward a more uniform distribution. The standard deviation σ of the histogram bins serves as a measure of spectral flatness: lower values indicate a more uniform pole density, corresponding to reduced coloration [10].

Unlike delay line modulation, which introduces chorus-like artifacts, feedback matrix modulation affects only the routing coefficients between delay lines. This operation alters the spectral content of the impulse response while introducing less pitch variation. The modulation frequency and amplitude thus provide sound designers with additional dimensions of control over the reverb character without the characteristics associated with traditional delay line modulation.

The modulation amplitude determines the extent of pole movement: smaller amplitudes yield subtle animation of the reverb tail, while larger amplitudes approaching π produce more pronounced effects. In this work, we use amplitudes in the range of $2\,\%$ to $15\,\%$ of π, combined with modulation rates between 0.1 Hz

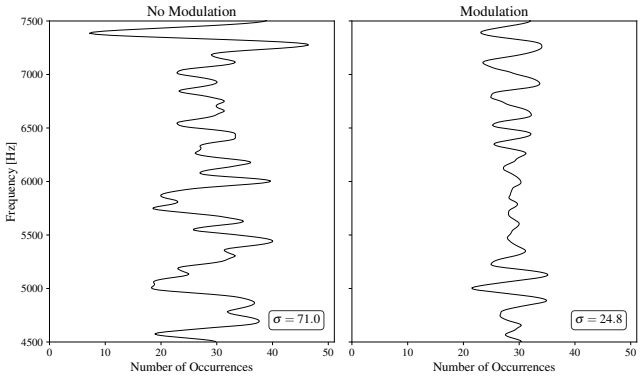

Figure 6: Pole frequency histograms for an 8×8 Kronecker FDN with rotation kernels and delay lengths $\mathbf{m} = [53, 61, 71, 79, 89, 101, 113, 127]$ samples at $f_s = 48\,\text{kHz}$. The middle kernel angle θ_2 is modulated with amplitude π using a triangle wave at 1 Hz. Reflection kernels exhibit similar behavior.

to 1.0 Hz. The examples in the supplementary materials audition the modulation at the no-modulation baseline, at a subtle setting within this range, and at an aggressive upper bound of π at 2 Hz.

6. CONCLUSIONS AND FUTURE WORK

This paper presents a creative parametric framework for designing lossless Feedback Delay Networks using orthogonal feedback matrices constructed through recursive Kronecker products of 2D rotation and reflection matrices. By parameterizing each of the M kernels with a single angle θ_i, we construct a family of $2^M \times 2^M$ orthogonal matrices that can be efficiently computed using a divide-and-conquer algorithm with $O(N \log_2 N)$ complexity for inputs of size $N = 2^M$, matching the computational efficiency of the Fast Walsh-Hadamard Transform while offering continuous parametric control.

The framework unifies several design objectives within a single mathematical structure. Setting all kernel angles to $\theta_i = \pi/4$ recovers the widely-used Hadamard matrices. Beyond this baseline, the parametric nature enables novel topological control: strategically varying individual kernel angles allows for stereo field manipulation with adjustable cross-coupling strength, selective network decoupling, and time-varying modulation to break resonances. Crucially, all transitions between these configurations maintain orthogonality, avoiding discontinuities that would otherwise introduce audible artifacts. These behaviors are continuously controllable through the kernel angle parameters, providing real-time access to a variety of network topologies from fully decoupled channels to complete cross-feedback.

While this framework does not span the entire space of orthogonal feedback matrices, the configurations it exposes proved sufficient for the three creative uses demonstrated above. These applications represent only a fraction of the possible configurations: other combinations of kernel types and angles across recursive levels may prove useful for different creative purposes. For instance, while this paper focuses on the stereo case for simplicity, this approach generalizes naturally to multi-channel configurations, such as a quadraphonic setup, by partitioning the feedback matrix into a larger number of blocks using two contiguous angles. Moreover, the framework is not limited to two-dimensional ker-

nels: any family of parametric orthogonal matrices could be used as building blocks, potentially offering different types of control over the feedback matrix structure. It would be especially interesting to investigate the use of other types of parametric kernels in conjunction with the Kronecker structure to achieve different topological properties and control. Additionally, if the kernel design admits a fast computation scheme, it would be interesting to see if it is possible to derive an efficient algorithm combining the fast kernel computation with the divide-and-conquer approach of Algorithm 1 and Algorithm 2, potentially enabling real-time control over more complex feedback matrix structures. Finally, another interesting development would be to perform a perceptual evaluation of the different topologies enabled by the framework, to better understand the relationship between rotation and reflection kernels, different angle parameters, network structure, and perceptual characteristics of the resulting reverberation.

The Kronecker Feedback Matrix framework offers FDN designers a unified approach to parametric reverb design that combines computational efficiency with expressive control over network topology, enabling real-time exploration of creative artificial reverberation effects while maintaining the orthogonality of the feedback matrix.

7. ACKNOWLEDGMENTS

I want to express my gratitude to the anonymous reviewers for their insightful comments and suggestions that helped to greatly improve the clarity and depth of this paper. Many thanks to Kevin Arcas, Rasmus Kürstein, Fanny Roche and Yann Bourdin for their careful review and constructive comments. Also thanks to the whole DSP team at Arturia for their support and feedback.

8. REFERENCES

[1] W. G. Gardner, "Reverberation algorithms," in *Applications of digital signal processing to audio and acoustics*. Springer, 1998, pp. 85–131.

[2] J. Heldmann and S. J. Schlecht, "The role of modal excitation in colorless reverberation," in *Proceedings of the 24th International Conference on Digital Audio Effects (DAFx-21)*, Vienna, Austria, 2021.

[3] D. Rocchesso, "Maximally diffusive yet efficient feedback delay networks for artificial reverberation," *Signal Processing Letters, IEEE*, vol. 4, p. 252–255, 1997.

[4] O. Das, E. K. Canfield-Dafilou, and J. S. Abel, "On the behavior of delay network reverberator modes," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. New Paltz, NY, USA: IEEE, Oct. 2019, p. 50–54.

[5] O. Das and J. S. Abel, "Grouped feedback delay networks for modeling of coupled spaces," *Journal of the Audio Engineering Society*, vol. 69, no. 7/8, p. 486–496, Nov. 2021.

[6] O. Das, J. S. Abel, and E. K. Canfield-Dafilou, "Delay network architectures for room and coupled space modeling," in *Proc. Int. Conf. Digit. Audio Effects*, 2020, pp. 234–241.

[7] G. D. Santo, K. Prawda, S. J. Schlecht, and V. Välimäki, "Efficient optimization of feedback delay networks for smooth reverberation," 2024. [Online]. Available: https://arxiv.org/abs/2402.11216

[8] J. Hörngren, "Optimization approaches for feedback delay networks," Master's thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2021.

[9] S. J. Schlecht and E. A. P. Habets, "Practical Considerations of Time-Varying Feedback Delay Networks," in *Journal of the Audio Engineering Society*, no. 9255, AES. Paper 9255; AES Convention 138; May 2015, may 2015.

[10] S. J. Schlecht and E. P. Habets, "Time-varying feedback matrices in feedback delay networks and their application in artificial reverberation," *The Journal of the Acoustical Society of America*, 2015.

[11] S. J. Schlecht, "FDNTB: The Feedback Delay Network Toolbox," in *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx-20)*, 2020.

[12] ——, "Allpass feedback delay networks," *IEEE Transactions on Signal Processing*, vol. 69, p. 1028–1038, 2021.

[13] S. J. Schlecht and E. A. P. Habets, "On lossless feedback delay networks," *IEEE Transactions on Signal Processing*, vol. 65, no. 6, p. 1554–1564, 2017.

[14] S. Schlecht, "Feedback delay networks in artificial reverberation and reverberation enhancement," Ph.D. dissertation, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2018.

[15] J. O. Smith, *Physical Audio Signal Processing*. CCRMA, Stanford University, 2010, online book. [Online]. Available: https://ccrma.stanford.edu/~jos/pasp/

[16] S. J. Schlecht and E. A. Habets, "Accurate reverberation time control in feedback delay networks," in *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17)*, Edinburgh, UK, 2017.

[17] H. Zhang and F. Ding, "On the Kronecker products and their applications," *Journal of Applied Mathematics*, 2013.

[18] B. C. Hall, "An elementary introduction to groups and representations," 2000. [Online]. Available: https://arxiv.org/abs/math-ph/0005032

[19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, 2001.

[20] H. Bai, G. Richard, and L. Daudet, "Late reverberation synthesis: From radiance transfer to feedback delay networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, p. 2260–2271, 2015.

[21] K. G. Beauchamp, *Applications of Walsh and Related Functions, with an Introduction to Sequency Theory*. Academic Press, 1984.

[22] International Organization for Standardization, "ISO 3382-1:2009 Acoustics, Measurement of room acoustic parameters, Part 1: Performance spaces," Standard, International Organization for Standardization, Geneva, CH, 2009. [Online]. Available: https://www.iso.org/standard/40979.html