Design of Discrete-time Matrix All-Pass Filters Using Subspace Nevanlinna Pick Interpolation

A thesis submitted in partial fulfillment of
the requirements for the degree of

Bachelor & Master of Technology

by

Agulla Surya Bharath
(Roll No. 17D070055)

Under the guidance of
Guide
Prof. Kumar Appaiah
Co-Guide
Prof. Debasattam Pal

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY–BOMBAY
2022

Thesis Approval

The thesis entitled

Design of Discrete-time Matrix All-Pass Filters Using Subspace Nevanlinna Pick Interpolation

by

Agulla Surya Bharath
(Roll No. 17D070055)

is approved for the degree of

Bachelor & Master of Technology

Digital Signature
Madhu Nagraj Belur (i03137)
30-Jun-22 02:18:42 PM

Examiner

Digital Signature
V Rajbabu (i07164)
30-Jun-22 12:45:09 PM

Examiner

Digital Signature
Kumar Appaiah (i14170)
30-Jun-22 11:51:23 AM

Guide

Digital Signature
Debasattam Pal (i14087)
30-Jun-22 01:46:47 PM

Co-Guide

Digital Signature
Madhu Nagraj Belur (i03137)
30-Jun-22 02:18:52 PM

Chairman

Date: i03137 30-Jun-22 02:19:05 PM June 2022

Place: i03137 30-Jun-22 02:19:19 PM Mumbai

Declaration

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

Date: June 30, 2022

Name: Agulla Surya Bharath

Place: IIT Bombay, Mumbai

Roll No: 17D070055

Abstract

Unitary matrix-valued functions of frequency can be viewed as matrix all-pass systems, since they preserve the norm of the input vector signals. Typically, such systems are represented and analyzed using their unitary-matrix valued frequency domain characteristics, although obtaining rational realizations for matrix all-pass systems enables compact representations and efficient implementations. However, an approach to obtain matrix all-pass filters that satisfy phase constraints at certain frequencies was hitherto unknown. In this paper, we present an interpolation strategy to obtain a rational matrix-valued transfer function from frequency domain constraints for discrete-time matrix all-pass systems. Using an extension of the Subspace Nevanlinna Pick Interpolation Problem (SNIP), we design a construction for discrete-time matrix all-pass systems that satisfies the desired phase characteristics. A key innovation that enables this is the extension of the SNIP to the boundary case to obtain efficient time-domain implementations of matrix all-pass filters as matrix linear constant coefficient difference equations, facilitated by a rational (realizable) transfer function matrix. We also show that the derivative of matrix phase constraints, related to the notion of group delay at the interpolating points, can be optimized to control the all-pass transfer matrices at the unspecified frequencies. Simulations show that the proposed technique for unitary matrix filter design performs as well as traditional DFT based interpolation approaches, such as Geodesic interpolation. Moreover, the proposed approach yields a more compact representation of the filter as a rational transfer function matrix, and thus, enables lower complexity implementations in applications such as MIMO wireless communication systems.

Keywords: Digital filters, filtering theory, IIR filters.

Contents

Chapter 1

Introduction

Filtering signals is among the most fundamental operations in signal processing. In general, filtering scalar signals is well understood, and there is mature theory that discusses filter design and implementation for both analog and digital scalar filters. However, with the increased interest in multiple-input multiple-output systems in several allied areas, the concept of filtering vector signals has gained importance. Designing precise filters for vector signals (wherein the signal at each time instant is a real or complex vector) under various constraints is also interesting from the point of view of several practical applications, although there has not been much work in the past in this direction. In this paper, we focus on the design of discrete-time "matrix" all-pass filters, that transform a vector signal's phase while ensuring that their norm is not altered for all frequencies. In particular, unlike the standard practice of using frequency domain transform techniques for filtering, we present an interpolation based filter design technique that produces matrix all-pass filters for practical realizability. This idea has several applications, such as combined left and right audio signals in case of stereo audio as well as for feedback in control and communication systems etc. As an example to show the effectiveness of the proposed techniques, we consider the MIMO precoders for wireless communication systems which employ orthogonal frequency division multiplexing (OFDM). These precoders can be accurately and efficiently realized using time domain techniques, as opposed to the traditionally used approaches [1, 2].

Matrix filtering with a norm preservation constraint is typically accomplished using frequency domain techniques [3, 4]. In particular, when the matrix all-pass filter has an efficient linear constant coefficient difference equation (LCCDE) realization, the filter realized using frequency domain techniques will be inaccurate, and will also result in less efficient realizations. To address this, we present an interpolation based matrix all-pass filter design technique that results in a realizable filter (that can be implemented in the time domain as an LCCDE) while satisfying the frequency domain constraints. Our approach extends the classical Subspace Nevanlinna Pick Interpolation (SNIP) method [5] that is well-known in the context of control systems to the "boundary" case to obtain matrix filters that satisfy some prior constraints, while ensuring that the Fourier transform of its system function is a unitary matrix at all frequencies, thus obtaining norm-preserving (matrix all-pass) filters.

The classical Nevanlinna interpolation problem has its roots in the problem of synthesis of dynamical systems as passive electrical networks (see [6]). This, as well as all the subsequent extensions of it, however, considers only the situation wherein the interpolating frequencies and the prescribed values of the desired transfer function are strictly within the respective critical regions. For example, in the scalar version of the SNIP dealt with in [5], the polar plot of the transfer function must lie strictly within the unit disk, and the frequencies that are given lie on the open right-half of the complex plane. It is important to note that the solution of the classical SNIP crucially depends on these strictness assumptions. In this paper, we push the SNIP to its boundary: we deal with the case wherein the desired transfer function's polar plot is on the unit disk (i.e., all-pass), and the frequencies, too, are given on the boundary (the unit circle because we consider discrete time systems).

Our key contributions in this project are as follows:

- We present an approach to realize a discrete-time matrix all-pass filter, when given a feasible set of frequency responses (unitary matrices) and group delay matrices for a finite set of frequencies $\{\omega_i\}$. Specifically, our solution yields a rational matrix z-transform for the required all-pass filter that satisfies all the given frequency domain conditions, and its transfer function matrix is unitary valued for all $\omega \in (-\pi, \pi]$. This can be viewed as a generalization of the Blaschke interpolation based approach that is specific to *scalar* all-pass filter design [7, 8] to the *matrix* case.

- We obtain this *matrix* all-pass filter by extending the SNIP technique to the boundary case. Specifically, since the Pick matrix in the case of standard SNIP [5] becomes ill-defined when we demand a unitary valued solution, we provide a modified approach using the modified Pick (Schwarz-Pick) matrix to generalize the SNIP filter realization to the boundary case in discrete-time setting.

- We also present an optimization based approach that tunes the slopes of the matrix phase response at specific frequencies to obtain realizable filters with desirable characteristics.

- We demonstrate a methodology to derive the lattice structure realization of a given discrete-time matrix all-pass filter for compact implementation.

1.1 Thesis outline

- Chapter-2 outlines the importance of matrix all-pass filter design problem statement

- Chapter-3 briefly describes the classical Subspace Nevanlinna-pick Interpolation Problem (SNIP)

- Chapter-4 describes the discrete-time matrix all-pass filter design problem and its solution

- Chapter-5 contains the simulations results and interpretations for some practical purposes

- Chapter-6.1 discusses the methods to quantize coefficients of matrix all-pass filters and presents the performance plots of quantized filter.

- Chapter-7 provides the empirical results of the rates achieved by the MIMO-OFDM system with limited feedback and matrix all-pass filter based precoder interpolation

- Chapter-8 presents a left coprime factorized solution to matrix all-pass filter problem

- Chapter-9 introduces the lattice structure realization of matrix all-pass filters

- Chapter-10 provides some concluding remarks and discusses future directions

- Appendix-A contains some relevant proofs related to discrete-time lossless systems.

Notation: Unless otherwise specified, bold capital symbols refer to matrices, bold small-case symbols correspond to vectors, $\mathbf{I}_m$ refers to an $m \times m$ identity matrix and $\mathbf{0}_{m\times m}$ refers to an $m \times m$ all-zero matrix. We also use the abbreviations CT for continuous-time and DT for discrete-time. For any matrix $\mathbf{A}$, $\mathbf{A}^*$ is the conjugate transpose of $\mathbf{A}$. For any matrix $\mathbf{A}$, $\mathbf{A}^H$ is the conjugate transpose of $\mathbf{A}$.

Chapter 2

Motivation

To motivate this problem, we first pose the *scalar* all-pass filter problem: if the frequency response of a discrete-time all-pass filter is given for certain (finitely many) frequencies, how can we obtain an all-pass filter that satisfies these constraints? In general, there exist an infinite number of filters that satisfy these constraints. Recent work has shown that, if the group delays are also known at the given frequencies, then a realizable all-pass filter can be obtained as a Blaschke product [7]. However, the Blaschke product based approach is only suited to the solution of the discrete-time *scalar* all-pass filter design problem, and a direct extension of the same approach to the case of matrix all-pass filters is not known.

Time-domain realizations of all-pass filters can be advantageous in several scenarios. Specifically, in the context of feedback based MIMO-OFDM communication systems, current approaches largely employ feedback of precoders for certain frequencies, and interpolation of the precoders across the remaining frequencies [9, 10]. This becomes error prone if sufficient parameters are not fed back [11]. We show that parameterizing the all-pass filters using their time-domain coefficients could lead to reduction of computational complexities. In addition, a time-domain realization of the precoder could also simplify the OFDM structure.

Discrete-time all-pass filters are used for phase compensation in various applications. A scalar all-pass filter can be used to correct phase distortions in scalar signals. To the best of our knowledge, this concept is yet to be extended to vector signals (MIMO systems), wherein the input and output are complex vector signals, and the transformation filter is a matrix valued all-pass filter (unitary). These matrix all-pass filters are commonly encountered in several situations, such as MIMO-OFDM systems, wherein the data symbols are precoded with unitary matrices at the transmitter, if the transmitter possesses some channel state information (CSI). In MIMO-OFDM systems, unitary matrix precoding is typically performed using the right singular vectors (or related unitary matrices) that are obtained from the singular value decomposition (SVD) of the channel matrix for every subcarrier (frequency band) [12]. Multiplying with a unitary matrix in the frequency domain is norm preserving, and thus, it can be thought of as a matrix all-pass filtering operation. In practice, having norm preserving matrix filters is important in order to satisfy the power constraint at the transmitter. Since CSI is typically not available at all subcarriers, interpolation is required to infer CSI at intermediate subcarriers. However, interpolating the precoding matrices in the space of unitary matrices requires operations over the group structure of unitary matrices, which do not form a vector space. Performing this operation using the coefficients of an appropriate discrete-time matrix all-pass filter with a standard LCCDE implementation would obviate the need for frequency domain interpolation.

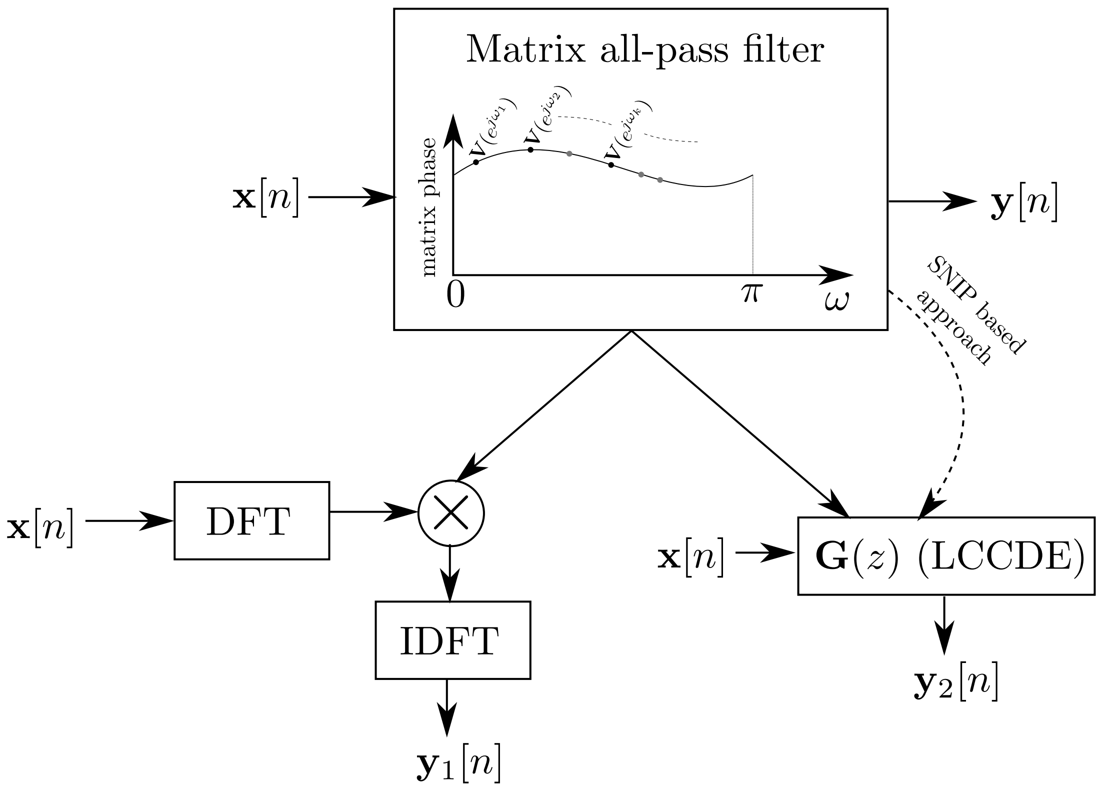
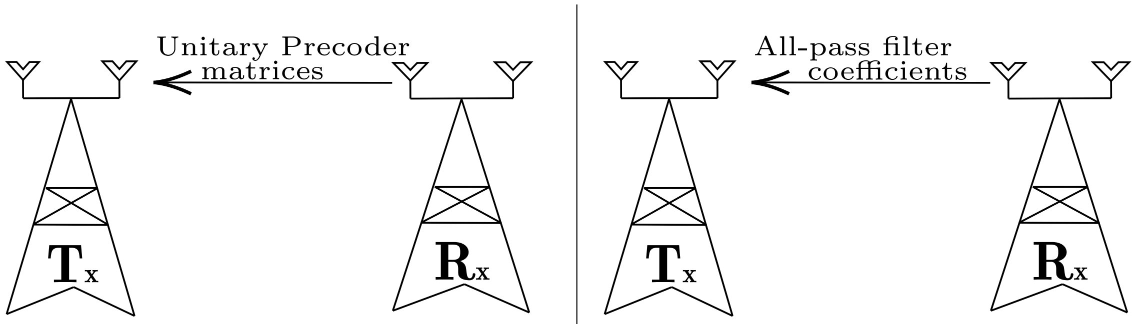


Figure 2.1: A schematic comparing the DFT based filter implementations with time-domain (LCCDE) based implementations.

The block diagram shown in Figure 2.1 depicts an unknown matrix all pass filter with efficient LCCDE representation, with $\mathbf{x}[n]$ as input and $\mathbf{y}[n]$ as output. Our goal is to construct a system that mimics the unknown matrix all-pass filter. One approach is to use DFT based techniques, wherein output $\mathbf{y}_1[n]$ is produced for the input $\mathbf{x}[n]$. Another method is to follow a matrix filter design technique and construct a rational matrix all pass filter with LCCDE representation, and this yields output $\mathbf{y}_2[n]$ for the input $\mathbf{x}[n]$. In this situation only the latter method is accurate.

It is evident from the block diagram Figure 2.1 that when the matrix all-pass filter has an efficient LCCDE realization, the filter realized using frequency domain techniques may be inaccurate, and may also result in less efficient realizations. An advantage of using the SNIP based LCCDE filter design is that the computational complexity can be reduced with a rational (realizable) all-pass filter method that has a compact representation, being more amenable to time-domain LCCDE implementations (details of the same are discussed further in the chapter 5).



(a) Feedback for precoder interpolation in typical DFT based methods

(b) Feedback for precoder interpolation in SNIP-based All-pass filter design method

Figure 2.2: Difference between classical frequency domain interpolation methods and our all-pass filter design method

Having a LCCDE form for the matrix all-pass filter allows us to just transmit the coefficients of LCCDE as feedback in MIMO-OFDM systems, for precoder interpolation. This also makes our method different from the typical frequency domain (DFT) based precoder interpolation techniques, as shown in Figure2.2.

In the context of control theory, realizing LCCDE matrix filters that satisfy frequency constraints (such as bounded $\mathbf{H}_\infty$ norm and prescribed matrix values at a given collection of finitely many complex numbers) is generally accomplished using the Subspace

Nevanlinna Pick Interpolation (SNIP) technique [5], although this approach was primarily designed for the systems, that have non-unitary frequency domain transfer functions. The requirement of matrix all-pass filter challenges us to extend the SNIP to the *boundary* case, wherein the solution is unitary valued for all frequencies (ω), and thus, SNIP is not directly applicable in our case. We, thus, present a modified SNIP that accommodates the boundary case to address our needs.

Example: Consider the following data set , $\mathbb{D} = \{(\omega_1 = 0, \mathbf{A}_1 = \mathbf{I}_2, \mathbf{\Gamma}_1 = 4\mathbf{I}_2), (\omega_2 = 1, \mathbf{A}_2 = -\mathbf{I}_2, \mathbf{\Gamma}_2 = 2\mathbf{I}_2)\}$. All-pass filter that exactly matches at these two data points has the following matrix transfer function:

$$\begin{aligned} \mathbf{G}(z) &:= \mathbf{N}(z)\mathbf{D}(z)^{-1} \\ \text{where, } \mathbf{N}(z) &= (0.8 + j0.76)\mathbf{I}_2 z^2 - (1.88 + j1.03)\mathbf{I}_2 z + (3.46 + j1.56)\mathbf{I}_2 \\ \text{and } \mathbf{D}(z) &= (3.19 + j2.06)\mathbf{I}_2 z^2 - (1.88 + j1.03)\mathbf{I}_2 z + (1.08 + j0.26)\mathbf{I}_2 \end{aligned}$$

This is a realizable (rational) matrix all-pass filter that satisfies the given phase and group delay constraints at $\omega \in \{0, 1\}$ and can be implemented as an LCCDE.

In the subsequent chapters, we first discuss the SNIP approach for continuous-time filter interpolation along with its limitations in the context of the unitary matrix valued (all-pass) constraint (boundary SNIP interpolation) due to the ill-defined Pick matrix. We then present our modified SNIP that addresses this issue and prove that we obtain a filter that satisfies the required conditions.

Chapter 3

Subspace Nevanlinna-pick Interpolation Problem

We now briefly outline the SNIP approach to perform filter interpolation that is commonly used in the context of continuous-time control systems to obtain filters that satisfy frequency domain constraints [5]. We restrict our consideration to square matrices, since our eventual focus would be on square unitary matrix valued frequency responses.

A *contractive subspace* ($\mathscr{V}_i \subset \mathbb{C}^{2m}$) is a subspace that satisfies the following property:

$$\left\{ \mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix} \in \mathscr{V}_i, \text{ with } \mathbf{v}_1, \mathbf{v}_2 \in \mathbb{C}^m \text{ and } \mathbf{v} \neq \mathbf{0} \right\} \Rightarrow \{||\mathbf{v}_1||_2 > ||\mathbf{v}_2||_2\}.$$

We consider N distinct points λ_i in the open right-half complex plane, given together with N subspaces $\mathscr{V}_i \subset \mathbb{C}^{2m}$, where $1 \leq i \leq N$.

The statement of the classical SNIP is as follows: Given the N pairs $(\lambda_i,\ \mathscr{V}_i)$, find an $m \times m$ polynomial matrix $\mathbf{U}$ and a non-singular $m \times m$ polynomial matrix $\mathbf{Y}$ such that

- $\mathbf{U}$ and $\mathbf{Y}$ are left coprime;
- $(\mathbf{U}(\lambda_i) \quad -\mathbf{Y}(\lambda_i))\mathbf{v} = 0 \quad \forall \quad \mathbf{v} \in \mathscr{V}_i, 1 \leq i \leq N$;
- and $||\mathbf{Y}^{-1}\mathbf{U}||_{\mathbf{H}_\infty} < 1$.

We assume that a full column rank matrix $\mathbf{V}_i \in \mathbb{C}^{2m \times m}$ is given such that $\mathrm{Im}(\mathbf{V}_i) = \mathscr{V}_i$ for every $i \in \{1, 2, \ldots N\}$. The Pick matrix $\mathbf{P}$ for the given data, for all $j, k \in \{1, 2, \ldots N\}$, is defined as

$$\begin{aligned} &\mathbf{P}[(k-1)m+1 : km, (j-1)m+1 : jm] := \left(\frac{\mathbf{V}_k{}^*\mathbf{J}\mathbf{V}_j}{\lambda_k{}^* + \lambda_j}\right) \\ &\text{where } \mathbf{J} := \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_{m\times m} \\ \mathbf{0}_{m\times m} & -\mathbf{I}_m \end{bmatrix}. \end{aligned} \tag{3.1}$$

Here, $\mathbf{J}$ is called a *signature matrix*, and $\mathbf{P}[a : b, c : d]$ is a submatrix of $\mathbf{P}$ obtained from contiguous rows, ranging from the ath row to the bth row, and contiguous columns ranging from the cth column to the dth column.

Proposition 1. *There exists a solution to the Subspace Nevanlinna Interpolation Problem (SNIP) discussed above if and only if the Hermitian matrix* $\mathbf{P}$ *is positive definite.*

This result is established in Theorem 4.1 of [5].

One key disadvantage with the SNIP is that the *contractive subspace* structure prevents it from being applicable to the case of matrix all-pass filter design (as detailed in the following chapters). In Chapter 4, we adapt the SNIP framework to design matrix all-pass filters.

Chapter 4

Modified SNIP for DT Matrix All-Pass Filters

In this chapter, we present an adaptation of SNIP to the discrete-time all-pass filter design case, wherein the filter constraints are presented for a sampled system. We remark that this case is directly applicable to several signal processing applications, such as stereo audio processing, MIMO-OFDM precoding etc.

Note: We refer to square matrices $\mathbf{X}$ and $\mathbf{Y}$ of size $k \times k$ as *unitarily similar* if there exists an unitary $k \times k$ matrix $\mathbf{T}$ such that $\mathbf{Y} = \mathbf{T}^*\mathbf{X}\mathbf{T}$.

4.1 Problem Statement

The discrete-time matrix all-pass filter design problem can be stated as follows: given data set $\mathbb{D} = \{(\omega_i, \mathbf{A}_i, \mathbf{\Gamma}_i | i = 1, ..., n\}$, where $\omega_i \in (-\pi, \pi], \mathbf{A}_i \in \mathbb{C}^{m\times m}$ is unitary, and $\mathbf{\Gamma}_i \in \mathbb{C}^{m\times m}$ is a positive definite Hermitian matrix ($\mathbf{\Gamma}_i = \mathbf{\Gamma}_i^*$), we wish to obtain a rational transfer function matrix $\mathbf{G}(z) \in \mathbb{C}(z)^{m\times m}$ that satisfies the following conditions

$$\mathbf{G}(e^{j\omega_i}) = \mathbf{A}_i \quad \forall \quad i = 1, ..., n, \tag{4.1a}$$

$$\mathbf{G}^*(e^{j\omega})\mathbf{G}(e^{j\omega}) = \mathbf{I}_m \quad \forall \quad \omega \in (-\pi, \pi], \tag{4.1b}$$

$$\begin{aligned} &\text{if } \mathbf{F}_i = j\mathbf{G}^*(e^{j\omega_i})\frac{d\mathbf{G}(e^{j\omega_i})}{d\omega} \text{ then} \\ &\mathbf{F}_i \text{ and } \mathbf{\Gamma}_i \text{ are unitarily similar matrices} \quad \forall \quad i = 1, ..., n \end{aligned} \tag{4.1c}$$

The condition described by equation (4.1b) provides us an all-pass filter transfer function, that also matches the desired (unitary) frequency responses $\mathbf{A}_i$ at the frequencies ω_i as constrained by equation (4.1a). Specifying *only* these two constraints leads to infinite possible all-pass filters (as in the scalar case [7]). Therefore, to restrict the set of possible solutions, we further specify $\mathbf{\Gamma}_i$ that are positive definite matrices and constrain the matrices $\mathbf{F}_i$ at ω_i (equation (4.1c)). The matrices $\mathbf{F}_i$ correspond to the derivative of the matrix-valued phase and thus, are referred to as *group delay matrices* [13]. The group delay matrices $\{\mathbf{F}_i\}$ are Hermitian $\forall i \in \{1, 2, ..n\}$. This is a direct consequence of the unitary constraint imposed in (4.1b), and can be verified by differentiating (4.1b) with respect to ω:

$$\begin{aligned} \mathbf{G}^*(e^{j\omega})\frac{d\mathbf{G}(e^{j\omega})}{d\omega} + \frac{d\mathbf{G}^*(e^{j\omega})}{d\omega}\mathbf{G}(e^{j\omega}) &= \mathbf{0}_{m\times m} \\ \Rightarrow j\mathbf{G}^*(e^{j\omega_i})\frac{d\mathbf{G}(e^{j\omega_i})}{d\omega} &= -j\frac{d\mathbf{G}^*(e^{j\omega_i})}{d\omega}\mathbf{G}(e^{j\omega_i}). \end{aligned}$$

It is evident from the above problem statement that the Subspace Nevanlinna Interpolation Problem (SNIP) [5] is very similar to our problem, but the SNIP does not require any slope or group delay constraints (like (4.1c) in our problem statement). Moreover, in the SNIP, the norm of the output vector is strictly smaller than the norm of the input vector, whereas in our case, they are equal.

Thus, our current all-pass filter design problem can be considered as an extension of the SNIP problem to the *boundary* case, wherein the norm of the output vector is exactly equal to the norm of the input vector. This requirement prevents the SNIP from being directly applicable to our problem, and motivates the formulation of a modified version that can be used for boundary problems as well.

4.2 Formulating the modified Pick matrix

To enable a step-wise solution to the matrix all-pass interpolation problem, we suitably modify the data set $\mathbb{D}$ to facilitate an inductive solution in subsequent steps. The data set $\mathbb{D}$ is altered by replacing $\mathbf{A}_i \in \mathbb{C}^{m\times m}$ with $\mathbf{B}_i \in \mathbb{C}^{2m\times m}$ that satisfy

$$\operatorname{span}_{\mathbb{C}}\mathbf{B}_i = \operatorname{span}_{\mathbb{C}}\begin{bmatrix}\mathbf{I}_m & \mathbf{A}_i^T\end{bmatrix}^T \quad \forall \quad i = 1, ..., n.$$

Now, in order to retain the unitary nature of $\mathbf{A}_i$, we define *neutral* $\mathbf{B}_i$ that satisfy

$$\mathbf{B}_i^* \mathbf{J} \mathbf{B}_i = \mathbf{0}_{m\times m} \qquad \text{where} \quad \mathbf{J} = \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_{m\times m} \\ \mathbf{0}_{m\times m} & -\mathbf{I}_m \end{bmatrix}. \tag{4.2}$$

The modified form of our data set is given by:
$\mathbb{D} = \{(\omega_i, \mathbf{B}_i, \mathbf{\Gamma}_i | i = 1, ..., n\}$, where $\omega_i \in (-\pi, \pi]$, $\mathbf{B}_i \in \mathbb{C}^{2m\times m}$ is neutral, $\mathbf{\Gamma}_i = \mathbf{\Gamma}_i^* \in \mathbb{C}^{m\times m}$ is positive definite.

We now set out to obtain the rational transfer function matrix $\mathbf{G}(z) \in \mathbb{C}(z)^{m\times m}$ that satisfies the constraint equation (4.1), with $\mathbf{A}_i$ replaced with $\mathbf{B}_{i,2}\ \mathbf{B}_{i,1}^{-1}$, where,

$$\mathbf{B}_i = \begin{bmatrix} \mathbf{B}_{i,1}^T & \mathbf{B}_{i,2}^T \end{bmatrix}^T \text{ and } \mathbf{B}_{i,1}, \mathbf{B}_{i,2} \in \mathbb{C}^{m\times m} \quad \forall i \in \{1, 2, \ldots n\}.$$

We cannot use the the classical SNIP solution construction from [5] directly by substituting $\mathbf{B}_i$ and $e^{j\omega_i}$ in place of $\mathbf{V}_i$ and λ_i respectively in the definition of the Pick matrix block (3.1) and matrix $\mathbf{J}$ defined in (4.2). This is because doing so results in a $\frac{0}{0}$ form along the diagonal blocks of the Pick matrix due to unitary nature of matrix $\mathbf{A}_i$:

$$\mathbf{P}[(i-1)m + 1 : im, (k-1)m + 1 : km] = \frac{\mathbf{B}_i{}^* \mathbf{J} \mathbf{B}_i}{1 - e^{j(\omega_i - \omega_i)}} \tag{4.3}$$

$$= \frac{\mathbf{I}_m - \mathbf{A}_i{}^* \mathbf{A}_i}{1 - 1} = \frac{\mathbf{0}_{m\times m}}{0}. \tag{4.4}$$

Therefore, we modify the Pick matrix to replace the the ill-defined matrices forming the diagonal blocks, and establish a set of *necessary and sufficient conditions* on the modified Pick matrix for solvability of the DT matrix all-pass filter design problem.

The modified $nm \times nm$ Pick matrix $\mathbf{P}$ is defined as follows:

$$\mathbf{P}[(i-1)m + 1 : im, (k-1)m + 1 : km] := \begin{cases} \mathbf{\Gamma}_i \quad , & \text{if } i = k \\ \frac{\mathbf{B}_i^* \mathbf{J} \mathbf{B}_k}{1 - e^{j(\omega_i - \omega_k)}}, & \text{otherwise} \end{cases} \tag{4.5}$$

for all $i, k \in \{1, .., n\}$. We now state a theorem that guarantees a solution for our problem using this Pick matrix.

Theorem 1. *Given the data set* $\mathbb{D} = \{(\omega_i, \mathbf{A}_i, \mathbf{\Gamma}_i)| i = 1, ..., n\}$ *with* ω_i *distinct, a rational transfer function matrix* $\mathbf{G}(z) \in \mathbb{C}(z)^{m\times m}$ *that satisfies* (4.1a), (4.1b) *and* (4.1c) *exists* if and only if *the Pick matrix defined in* (4.5) *is positive definite.*

Proof. Explicit construction of a solution is specified in Section 4.3 to prove the *if* part. To prove the *only if* part, we show that, for a given data set $\mathbb{D} = \{(\omega_i, \mathbf{A}_i, \mathbf{\Gamma}_i) | i = 1, ..., n\}$ if we have $\mathbf{G}(z) \in \mathbb{C}(z)^{m \times m}$ that satisfies (4.1a), (4.1b) and (4.1c) then the Pick matrix for $\mathbb{D}$ defined in (4.5) is positive definite.

Since our interpolation problem concerns itself with the discrete-time case, both the input ($\mathbf{x}[n]$) and output ($\mathbf{y}[n]$) to the matrix all-pass filter are discrete-time vector signals. For simplicity, we represent $\mathbf{x}[n]$ as $\mathbf{x}$ and $\mathbf{y}[n]$ as $\mathbf{y}$, suppressing the time index. In the language of the theory of dissipative systems, the all-pass transfer function represents a *lossless* DT dynamical system with the *supply rate* defined as $\mathbf{x}^*\mathbf{x} - \mathbf{y}^*\mathbf{y}$. From the fundamental theorem of dissipative systems, it follows that lossless dissipative systems satisfy the following equation [14–16]. That is,

$$\textit{supply rate}[n] = \textit{storage function}[n+1] - \textit{storage function}[n] \tag{4.6}$$

where $\{\textit{supply rate}[n]\}$ is the supply rate value computed at time index n and similarly $\{\textit{storage function}[n]\}$ is the storage function value computed at time index n.

Therefore, since the *storage function* is a positive definite function of state variables [16], using the concepts of discrete-time *lossless* dissipative dynamical systems explained in [15], we observe that

$$\sum_{n=-\infty}^{0} (\mathbf{x}^*\mathbf{x} - \mathbf{y}^*\mathbf{y}) \geq 0 \tag{4.7}$$

Now, consider $\mathbf{x} = \mathbf{v}e^{(\epsilon_i + j\omega_i)n}$, where $\mathbf{v}$ is an arbitrary vector in $\mathbb{C}^m$, ϵ_i is an arbitrary small positive real number, and ω_i is the i-th interpolation frequency. The input to the system is $\mathbf{x}$, and hence, the output is $\mathbf{y} = \mathbf{G}\mathbf{x}$. Thus, we get $\mathbf{y} = \mathbf{G}(e^{\epsilon_i + j\omega_i})\mathbf{v}e^{(\epsilon_i + j\omega_i)n}$. Let $\mathbf{G}_i := \mathbf{G}(e^{\epsilon_i + j\omega_i})$. Now, substituting $\mathbf{x}, \mathbf{y}$ in (4.7), we get

$$\begin{aligned} \sum_{n=-\infty}^{0} (\mathbf{x}^*\mathbf{x} - \mathbf{y}^*\mathbf{y}) &= \mathbf{v}^*(\mathbf{I}_m - \mathbf{G}_i^*\mathbf{G}_i)\mathbf{v} \sum_{n=-\infty}^{0} (e^{2\epsilon_i n}) \\ &= \frac{\mathbf{v}^*(\mathbf{I}_m - \mathbf{G}_i^*\mathbf{G}_i)\mathbf{v}}{1 - e^{-2\epsilon_i}} \geq 0. \end{aligned}$$

Taking limit $\epsilon_i \to 0^+$ in the above equation, we get $\lim_{\epsilon_i \to 0^+} \frac{\mathbf{v}^*(\mathbf{I}_m - \mathbf{G}_i^*\mathbf{G}_i)\mathbf{v}}{1 - e^{-2\epsilon_i}} \geq \mathbf{0}$.

Now, applying L'Hospital's rule yields

$$\begin{aligned}\lim_{\epsilon_i \to 0^+} \frac{\mathbf{v}^*(\mathbf{I}_m - \mathbf{G}_i^*\mathbf{G}_i)\mathbf{v}}{1-(e^{-2\epsilon_i})} &= \lim_{\epsilon_i \to 0^+} \frac{\frac{d}{d\epsilon_i}[\mathbf{v}^*(\mathbf{I}_m - \mathbf{G}_i^*\mathbf{G}_i)\mathbf{v}]}{2} \\ &= \lim_{\epsilon_i \to 0^+} -\frac{\mathbf{v}^*\frac{d}{d\epsilon_i}[\mathbf{G}_i^*\mathbf{G}_i)]\mathbf{v}}{2} \\ &= \lim_{\epsilon_i \to 0^+} -\frac{\mathbf{v}^*[\frac{d}{d\epsilon_i}(\mathbf{G}_i^*)\mathbf{G}_i + \mathbf{G}_i^*\frac{d}{d\epsilon_i}(\mathbf{G}_i)]\mathbf{v}}{2}.\end{aligned}$$

We know that, the following equalities hold,

$$\frac{d}{d\epsilon_i}\mathbf{G}_i = -j\frac{d}{d\omega_i}\mathbf{G}_i \text{ and } \frac{d}{d\epsilon_i}(\mathbf{G}_i^*) = j\frac{d}{d\omega_i}(\mathbf{G}_i^*).$$

So,

$$\begin{aligned}&\lim_{\epsilon_i \to 0^+} \frac{\mathbf{v}^*[\frac{d}{d\epsilon_i}(\mathbf{G}_i^*)\mathbf{G}_i + \mathbf{G}_i^*\frac{d}{d\epsilon_i}\mathbf{G}_i]\mathbf{v}}{2} \\ &\qquad = j\mathbf{v}^*[\mathbf{G}^*(e^{j\omega_i})\frac{d\mathbf{G}(e^{j\omega_i})}{d\omega}]\mathbf{v} = \mathbf{v}^*\mathbf{F}_i\mathbf{v} \geq 0.\end{aligned}$$

Since $\mathbf{F}_i$ and $\mathbf{\Gamma}_i$ are Hermitian and unitarily similar matrices, we obtain

$$\mathbf{v}^*\mathbf{\Gamma}_i\mathbf{v} \geq 0, \quad \forall \mathbf{v} \in \mathbb{C}^m \text{ and } \forall i \in \{1, 2, \dots n\}. \tag{4.8}$$

It is evident that equality holds in (4.8) only if $\mathbf{v} = \mathbf{0}_{m\times 1}$, because if $\mathbf{v}$ is non-zero, then the corresponding $\mathbf{y}$ cannot be zero, because $\mathbf{G}(e^{j\omega})$ is unitary for all $\omega \in (-\pi, \pi]$ and $\mathbf{y} = \mathbf{G}(e^{\epsilon_i + j\omega_i})\mathbf{v}e^{(\epsilon_i+j\omega_i)n}$.

Next, consider $\mathbf{x} = \sum_{i=1}^{n} \mathbf{v}_i e^{(\epsilon_i+j\omega_i)n}$, where $\mathbf{v}_i$ are arbitrary vectors in $\mathbb{C}^m$. Correspondingly, $\mathbf{y} = \sum_{i=1}^{n} \mathbf{G}_i\mathbf{v}_i e^{(\epsilon_i+j\omega_i)n}$. By a calculation similar to the one above, we get

$$\sum_{n=-\infty}^{0} (\mathbf{x}^*\mathbf{x} - \mathbf{y}^*\mathbf{y}) = \begin{bmatrix}\mathbf{v}_1^* & \mathbf{v}_2^* & \dots & \mathbf{v}_n^*\end{bmatrix} \mathbf{T} \begin{bmatrix}\mathbf{v}_1^T & \mathbf{v}_2^T & \dots & \mathbf{v}_n^T\end{bmatrix}^T \tag{4.9}$$

where, $\mathbf{T}$ is a $nm \times nm$ matrix. The ith $m \times m$ diagonal block of $\mathbf{T}$ is $[\mathbf{T}]_{ii} = \mathbf{\Gamma}_i$ as derived in (4.8). The off-diagonal (i, k)th $m \times m$ block entries of $\mathbf{T}$ are specified as follows:

$$\begin{aligned}[\mathbf{T}]_{ik} &= \lim_{\epsilon_i,\epsilon_k \to 0^+} \sum_{n=-\infty}^{0} (\mathbf{I}_m - \mathbf{G}_i^*\mathbf{G}_k)e^{(\epsilon_k+\epsilon_i+j(\omega_k-\omega_i))n} \\ &= \frac{(\mathbf{I}_m - \mathbf{G}_i^*\mathbf{G}_k)}{1-e^{j(\omega_i-\omega_k)}} = \frac{(\mathbf{I}_m - \mathbf{A}_i^*\mathbf{A}_k)}{1-e^{j(\omega_i-\omega_k)}}.\end{aligned} \tag{4.10}$$

Thus, we see that $\mathbf{T}$ is, in fact, the Pick matrix $\mathbf{P}$ that we have defined in (4.5). In addition, from (4.7) and (4.9), we have

$$\begin{bmatrix}\mathbf{v}_1^* & \mathbf{v}_2^* & \dots & \mathbf{v}_n^*\end{bmatrix} \mathbf{P} \begin{bmatrix}\mathbf{v}_1^T & \mathbf{v}_2^T & \dots & \mathbf{v}_n^T\end{bmatrix}^T \geq \mathbf{0}_{m\times m} \quad \forall \mathbf{v}_i$$

where equality holds if and only if all $\mathbf{v}_i$s are $\mathbf{0}_{m\times 1}$ vectors. Thus, the Pick matrix $\mathbf{P}$ defined for data set $\mathbb{D}$ is positive definite. $\square$

4.3 Induction based solution construction

As mentioned earlier in Theorem 1, if the Pick matrix (defined in (4.5)) for the given data set $\mathbb{D}$ is positive definite, then a matrix all-pass filter that satisfies the constraints (4.1a), (4.1b) and (4.1c) can be constructed. (Proof of the *if* part of the Theorem 1).

We construct a solution to the problem discussed above using induction on the number of data points n as follows. First, we define

$$\mathbf{B}_i = \begin{bmatrix}\mathbf{I}_m & \mathbf{A}_i^T\end{bmatrix}^T, \qquad i = 1, 2, .., n.$$

Base Step: $n = 1, \mathbb{D} = (\omega_1, \mathbf{B}_1, \mathbf{\Gamma}_1)$.

Let, $z = e^{j\omega}$ and $z_1 = e^{j\omega_1}$. We first construct the following matrix polynomials. Define,

$$\begin{aligned}\mathbf{N}(z) &:= (z - z_1)\mathbf{I}_m + \frac{\mathbf{A}_1\mathbf{\Gamma}_1^{-1}(\mathbf{I}_m - \mathbf{A}_1^*)z_1}{(1 + z_1)}(1 + z)\\ \mathbf{D}(z) &:= (z - z_1)\mathbf{I}_m + \frac{\mathbf{\Gamma}_1^{-1}(\mathbf{I}_m - \mathbf{A}_1^*)z_1}{(1 + z_1)}(1 + z).\end{aligned} \tag{4.11}$$

Since, $\mathbf{\Gamma}_1$ is positive definite, $\mathbf{N}(z)$ and $\mathbf{D}(z)$ are well-defined. Correspondingly, we define $\mathbf{G}(z) := \mathbf{N}(z)\mathbf{D}(z)^{-1}$. We now present a brief verification that confirms that the base step of this induction indeed satisfies the required conditions. To this end, we note that $\mathbf{N}(z_1)\mathbf{D}(z_1)^{-1} = \mathbf{A}_1$, thus satisfying (4.1a). We know that,

$$\begin{gathered}\mathbf{G}^*(e^{j\omega})\mathbf{G}(e^{j\omega}) = \mathbf{I}_m\\ \Updownarrow\\ \mathbf{N}^*(e^{j\omega})\mathbf{N}(e^{j\omega}) = \mathbf{D}^*(e^{j\omega})\mathbf{D}(e^{j\omega}) \quad \forall\omega \in (-\pi, \pi]\end{gathered} \tag{4.12}$$

which can be easily verified by performing simple polynomial multiplications, thus satisfying (4.1b). Next, we see that (4.1c) can be verified as follows:

$$\mathbf{G}(z)\mathbf{D}(z) := \mathbf{N}(z) \Rightarrow \frac{d\mathbf{G}(z_1)}{d\omega}\mathbf{D}(z_1) + \mathbf{G}(z_1)\frac{d\mathbf{D}(z_1)}{d\omega} = \frac{d\mathbf{N}(z_1)}{d\omega}$$

$$\Rightarrow j\mathbf{G}^*(z_1)\frac{d\mathbf{G}(z_1)}{d\omega}\mathbf{D}(z_1) = j\mathbf{G}^*(z_1)\frac{d\mathbf{N}(z_1)}{d\omega} - j\frac{d\mathbf{D}(z_1)}{d\omega}$$

$$\Rightarrow \left(j\mathbf{G}^*(z_1)\frac{d\mathbf{G}(z_1)}{d\omega} \right) \left(\mathbf{\Gamma}_1^{-1}(\mathbf{I}_m - \mathbf{A}_1^*)z_1 \right) = (\mathbf{I}_m - \mathbf{A}_1^*)z_1.$$

Therefore, we have $j\mathbf{G}^*(e^{j\omega_1})\frac{d\mathbf{G}(e^{j\omega_1})}{d\omega} = \mathbf{\Gamma}_1$, and thus, $\mathbf{G}(z)$ satisfies the constraint mentioned in (4.1c). Thus, the base step is verified.

Inductive Step: We assume that our problem is solvable for $n-1$ points, and use this to prove that the problem is solvable for n points. First we suitably modify the given data set of $n-1$ points. To this end, we define the following $2m \times 2m$ matrix,

$$\mathbf{H}(z) = \begin{bmatrix} 2(z-z_1)\mathbf{I}_m - (z+1)(z_1+1)\mathbf{\Gamma}_1^{-1} & (z+1)(z_1+1)\mathbf{\Gamma}_1^{-1}\mathbf{A}_1^* \\ -(z+1)(z_1+1)\mathbf{A}_1\mathbf{\Gamma}_1^{-1} & 2(z-z_1)\mathbf{I}_m + (z+1)(z_1+1)\mathbf{A}_1\mathbf{\Gamma}_1^{-1}\mathbf{A}_1^* \end{bmatrix}.$$

Now, for each $i \in \{1, 2, \ldots n-1\}$, we define a modified data set as follows:

$$\begin{aligned} \widehat{\mathbf{B}}_i &= \mathbf{H}(e^{j\omega_{i+1}})\begin{bmatrix} \mathbf{I}_m & \mathbf{A}_{i+1}^T \end{bmatrix}^T \\ \widehat{\mathbf{\Gamma}}_i &= \mathbf{\Gamma}_{i+1} - \frac{(\mathbf{I} - \mathbf{A}_{i+1}^*\mathbf{A}_1)\mathbf{\Gamma_1}^{-1}(\mathbf{I} - \mathbf{A}_1^*\mathbf{A}_{i+1})}{(1-e^{j(\omega_{i+1}-\omega_1)})(1-e^{j(\omega_1-\omega_{i+1})})}. \end{aligned} \tag{4.13}$$

Both of the above modifications result in new data sets that also satisfy the conditions outlined in the section 4.2, since $\widehat{\mathbf{B}}_i$s are neutral and $\widehat{\mathbf{\Gamma}}_i$s are still Hermitian. Thus, the modified data set is $\widehat{\mathbb{D}} := \{(\omega_{i+1}, \widehat{\mathbf{B}}_i, \widehat{\mathbf{\Gamma}}_i) | \ i = 1, ..., n-1\}$.

It is important to note that the modifications in (8.3) are performed so that the Pick matrix of the new data set ($\widehat{\mathbb{D}}$) is the Schur complement of the Pick matrix of the original input data set ($\mathbb{D}$) with respect to $\mathbf{\Gamma}_1$ [17]. Thus, using the Schur complement property, we can argue that if the original Pick matrix for the data set $\mathbb{D}$ is positive definite, then the new Pick matrix for data set $\widehat{\mathbb{D}}$ will also be positive definite.

Using the original induction assumption, we know that we can solve for $\widehat{\mathbf{N}}(z), \widehat{\mathbf{D}}(z) \in \mathbb{C}^{m\times m}$ such that $\widehat{\mathbf{G}}(z) = \widehat{\mathbf{N}}(z)\widehat{\mathbf{D}}(z)^{-1}$ satisfies the matrix all-pass filter design problem for the new data set $\widehat{\mathbb{D}}$ containing $n-1$ points. Therefore, for $z_i := e^{j\omega_i}$, the following holds

$$\widehat{\mathbf{B}}_i^* \begin{bmatrix} \widehat{\mathbf{D}}(z_{i+1}) \\ -\widehat{\mathbf{N}}(z_{i+1}) \end{bmatrix} = 0 \quad \forall \quad i = 1, \ldots, N-1$$

$$\widehat{\mathbf{B}}_i = \mathbf{H}(z_{i+1})\mathbf{B}_{i+1} \Rightarrow \mathbf{B}_{i+1}^*\mathbf{H}(z_{i+1})^* \begin{bmatrix} \widehat{\mathbf{D}}(z_{i+1}) \\ -\widehat{\mathbf{N}}(z_{i+1}) \end{bmatrix} = 0.$$

We know that $\mathbf{B}_1^*\mathbf{H}(z_1)^* = \mathbf{0}_{m\times 2m}$. Therefore, $\forall \quad i = 1,\ldots,N$, we have

$$\begin{bmatrix} \mathbf{D}(z) \\ -\mathbf{N}(z) \end{bmatrix} := \mathbf{H}(z)^* \begin{bmatrix} \widehat{\mathbf{D}}(z) \\ -\widehat{\mathbf{N}}(z) \end{bmatrix} \Rightarrow \mathbf{N}(z_i)\mathbf{D}(z_i)^{-1} = \mathbf{A}_i. \tag{4.14}$$

We now verify the unitary nature of $\mathbf{N}(e^{j\omega})\mathbf{D}(e^{j\omega})^{-1}$ as follows:

$$\begin{aligned}
\mathbf{N}^*(z)\mathbf{N}(z) - \mathbf{D}^*(z)\mathbf{D}(z) &= \begin{bmatrix} \mathbf{N}^*(z) & -\mathbf{D}^*(z) \end{bmatrix} \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_{m\times m} \\ \mathbf{0}_{m\times m} & -\mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{N}(z) \\ -\mathbf{D}(z) \end{bmatrix} \\
&= \begin{bmatrix} \widehat{\mathbf{N}}^*(z) & -\widehat{\mathbf{D}}^*(z) \end{bmatrix} [\mathbf{H}(z)^*]^* \mathbf{J} [\mathbf{H}(z)]^* \begin{bmatrix} \widehat{\mathbf{N}}(z) \\ -\widehat{\mathbf{D}}(z) \end{bmatrix} \\
&= \frac{-4(z_1 - z)^2}{z_1 z} \begin{bmatrix} \widehat{\mathbf{N}}^*(z) & -\widehat{\mathbf{D}}^*(z) \end{bmatrix} \mathbf{J} \begin{bmatrix} \widehat{\mathbf{N}}(z) \\ -\widehat{\mathbf{D}}(z) \end{bmatrix} \\
&= \frac{-4(e^{j\omega_1} - e^{j\omega})^2}{e^{j(\omega_1+\omega)}} \left(\widehat{\mathbf{N}}^*(e^{j\omega})\widehat{\mathbf{N}}(e^{j\omega}) - \widehat{\mathbf{D}}^*(e^{j\omega})\widehat{\mathbf{D}}(e^{j\omega}) \right) \\
&= 0.
\end{aligned} \tag{4.15}$$

The last equality can be inferred from the original induction assumption that $\widehat{\mathbf{N}}(e^{j\omega})\widehat{\mathbf{D}}(e^{j\omega})^{-1}$ satisfies the matrix all-pass filter constraints. Therefore, $\{\mathbf{N}^*(e^{j\omega})\mathbf{N}(e^{j\omega}) - \mathbf{D}^*(e^{j\omega})\mathbf{D}(e^{j\omega}) = 0\} \quad \forall \omega \in (-\pi, \pi]$, which implies $\mathbf{N}(e^{j\omega})\mathbf{D}(e^{j\omega})^{-1}$ is unitary for all $\omega \in (-\pi, \pi]$. Thus, we expand the expression of the solution for the original data set of n points as follows (from equation (8.4)):

$$\begin{aligned}
\mathbf{N}(z) &:= [2(z - z_1)\mathbf{I} - (z+1)(z_1+1)\mathbf{A}_1\mathbf{\Gamma}_1^{-1}\mathbf{A}_1^*]\widehat{\mathbf{N}}(z) \\
&\qquad + [(z+1)(z_1+1)\mathbf{A}_1\mathbf{\Gamma}_1^{-1}]\widehat{\mathbf{D}}(z) \\
\mathbf{D}(z) &:= [2(z - z_1)\mathbf{I} + (z+1)(z_1+1)\mathbf{\Gamma}_1^{-1}]\widehat{\mathbf{D}}(z) \\
&\qquad - [(z+1)(z_1+1)\mathbf{\Gamma}_1^{-1}\mathbf{A}_1^*]\widehat{\mathbf{N}}(z)
\end{aligned}$$

$$\mathbf{G}(z) := \mathbf{N}(z)\mathbf{D}(z)^{-1}.$$

This $\mathbf{G}(z)$ satisfies the conditions in (4.1a), (4.1b), and (4.1c). This completes the mathematical induction steps.

Note: For a more intuitive understanding, we refer to $\mathbf{\Gamma}_i$s as ‘group delay matrices’ in the sequel, while noting that they are unitarily similar to the original group delay matrices $\mathbf{F}_i$ specified in (4.1c).

The group delay matrices ($\mathbf{\Gamma}_i$s) may not always be available at the interpolating points in the given data set. In such situations we can obtain suitable $\mathbf{\Gamma}_i$s through an optimization process described in Section 4.4.

4.4 Optimizing group delay matrices at interpolating points

Theorem 1 indicates that, if the Pick matrix defined in (4.5) is positive definite, then an infinite number of solutions exist that satisfy (4.1c). These solutions are determined by specifying $\mathbf{\Gamma}_i \quad \forall i \in \{1, 2, \ldots n\}$.

It is desirable to minimize the trace of $\mathbf{\Gamma}_i$s at the interpolating points to obtain a useful filter [7]. The constraints on the trace can be expressed as linear matrix inequalities (LMI), and semi-definite programming (SDP) can be used to solve it efficiently. The optimization technique is to minimize the sum of the diagonal elements in the Pick matrix, subject to the constraint that the modified Pick matrix defined in Section 4.2 is positive definite (for the selected $\mathbf{\Gamma}_i$s).

$$
\begin{aligned}
\text{minimize} \quad & \text{Trace}(\mathbf{P}) \\
\text{subject to} \quad & \mathbf{P} \succ \mathbf{0}, \quad \mathbf{P}_{ii} = \mathbf{\Gamma}_i \succ \mathbf{0} \text{ and} \\
& \mathbf{P}_{ik} = \frac{\mathbf{B}_i^* \mathbf{J} \mathbf{B}_k}{1 - e^{j(\omega_i - \omega_k)}}, \quad i \neq k \\
& \text{for all } i, k \in \{1, 2, \ldots, n\}
\end{aligned}
$$

where, $\mathbf{P}_{ik} = \mathbf{P}[(i-1)m + 1 : im, (k-1)m + 1 : km]$. The key intuition behind the formulation of this optimization is that minimizing the sum of the trace of the group delay matrices yields a filter that has smaller phase variations, and thus have better phase characteristics across frequencies without abrupt variation. Simulations confirm that the above optimization problem yields effective realizable matrix all-pass filters.

Chapter 5

Simulation and Discussion

To evaluate the effectiveness of the proposed matrix all-pass filter design technique, we present a simulation based on MIMO-OFDM systems that uses an N_{FFT} sized IDFT on a wireless Rayleigh fading frequency selective channel. The OFDM case is particularly interesting, since the channel within each FFT sub-band (referred to as *subcarrier*) can be assumed to be frequency flat (i.e., flat fading). Thus, assuming that the channel in the k-th subcarrier is modeled as the matrix $\mathbf{H}[k] \in \mathbb{C}^{m\times m}$, this matrix can be decomposed using the SVD as $\mathbf{H}[k] = \mathbf{U}[k]\mathbf{\Sigma}[k]\mathbf{V}^*[k]$, where $\mathbf{V}[k]$ is typically used to *precode* (pre-multiply) the data at the transmitter to enable channel parallelization. In other words, the symbol vector at every subcarrier is premultiplied by $\mathbf{V}[k]$, $k = 0, 1, 2, \ldots N_{\text{FFT}}$. However, since the channel is typically estimated at the receiver, the receiver must feedback $\mathbf{V}[k]$ for all subcarriers to the transmitter. This can lead to a significant overhead in the feedback requirements. In contrast, the technique from Chapter 4 can offer two key advantages:

- By viewing $\mathbf{V}[k]$ as samples of $\mathbf{V}(e^{j\omega})$, $\omega \in (-\pi, \pi]$, we can use the technique from Chapter 4 to obtain a rational transfer function that represents the precoding operation. Thus, we need to feedback only the coefficients of the transfer function.

- A compact rational transfer function would yield a simple system of linear constant coefficient difference equations (LCCDE) that can be implemented in the time domain, leading to a more compact representation. We can thereby compute the precoder matrix by simple matrix addition and multiplication operations, unlike the matrix exponential and logarithm operations that are used for geodesic interpolation. More importantly, these alternate approaches do not result in realizable filters, thereby making them suitable for only frequency domain processing.

To emphasize these points, consider the precoding problem in a frequency selective MIMO channel for a wireless system, wherein the matrix precoding function $\mathbf{V}(e^{j\omega})$ has a compact representation in terms of ω. For several such randomly generated channels, we compare the performance of our interpolation technique and the traditional geodesic interpolation [18]. We implemented the above discussed discrete-time matrix all-pass filter design method on a 2×2 MIMO system, with an input data set of 6 elements. That is, the unitary precoding filter and the corresponding group delay matrices are known at six frequencies in $(-\pi, \pi]$. The data set of unitary matrices was generated from a wireless MIMO Rayleigh fading channel ($\mathbf{H}(e^{j\omega})$), and taking the singular value decomposition as follows:

$$\mathbf{H}(e^{j\omega}) = \mathbf{U}(e^{j\omega})\mathbf{\Sigma}(e^{j\omega})\mathbf{V}^*(e^{j\omega}),$$

where $\mathbf{U}(e^{j\omega})$ and $\mathbf{V}(e^{j\omega})$ are unitary. The channels follow the the ITU Vehicular A power delay profile [19]. The unitary matrices $\{\mathbf{V}(e^{j\omega})\}$ that correspond to the right singular vectors of the channel are evaluated at ($\omega \in \{-0.99\pi, -\frac{3\pi}{5}, -\frac{\pi}{5}, \frac{\pi}{5}, \frac{3\pi}{5}, 0.99\pi\}$), both for the modified SNIP approach as well as the frequency domain geodesic approach. At frequencies other than those in data set, we interpolate the unitary matrices using the respective techniques. To quantify the accuracy of interpolation method, we plot the error between the interpolated unitary matrix $\{\hat{\mathbf{V}}(e^{j\omega})\}$ and the unitary matrix$\{\mathbf{V}(e^{j\omega})\}$ that is realized from the channel matrix $\{\mathbf{H}(e^{j\omega})\}$ for all ω. We remark here that the frequency domain approach does not yield a realizable rational matrix all-pass filter, while our proposed approach is guaranteed to do so.

Figure 8.1 and Figure 8.2 present the error as measured both using the Frobenius norm, and Flag Distance in [20] as performance metrics.

The Flag Distance gives a measure of distance between two matrices which are considered equivalent upon multiplication by a diagonal unitary matrices. The SVD is not unique for a matrix $\mathbf{H}(e^{j\omega})$, since multiplying the right and left singular vectors by diagonal unitary matrices results in equivalent precoders [21]. The channel capacity can be achieved on each subcarrier by precoding with any equivalent right-singular matrix extracted from channel matrix ($\mathbf{H}(e^{j\omega})$) using the SVD [22] (due to the Flag manifold structure).

The error values in the plots are averaged over 1000 channel iterations. We can observe from the simulations that the output of the discrete-time matrix all-pass filter exactly matches the unitary matrix in data set at the frequencies in data set (referred to as interpolating points), thereby confirming that the realizable filter satisfies the specified frequency domain constraints.

While the performance of our discrete-time matrix all-pass filter design technique based on SNIP (without optimizing group delays) is comparable to geodesic interpolation, it is evident that there is still a gap in performance when compared to the geodesic approach. This can be attributed to the fact that there may exist other choices of group delay matrices $\{\mathbf{\Gamma}_i\}$ that are unitarily similar matrices (see (4.1c)) which are not considered. We address this limitation using the optimization based approach to filter realization described in Section 4.4. The group delay ($\mathbf{\Gamma}_i$) matrices are obtained using optimization, and then the matrix all-pass filter is constructed using the technique presented in 4.3.

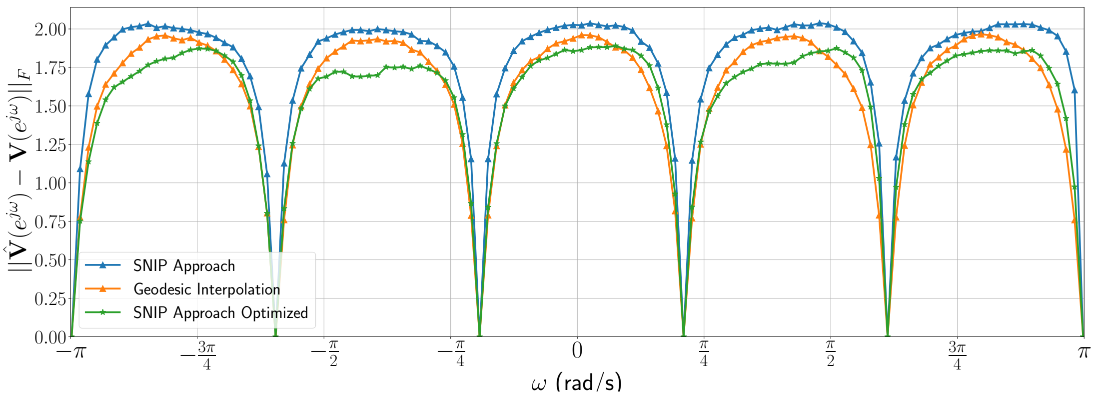


Figure 5.1: Frequency domain error measured for matrix all-pass filters constructed from samples for a 2×2 MIMO system, with Frobenius norm as error measure.

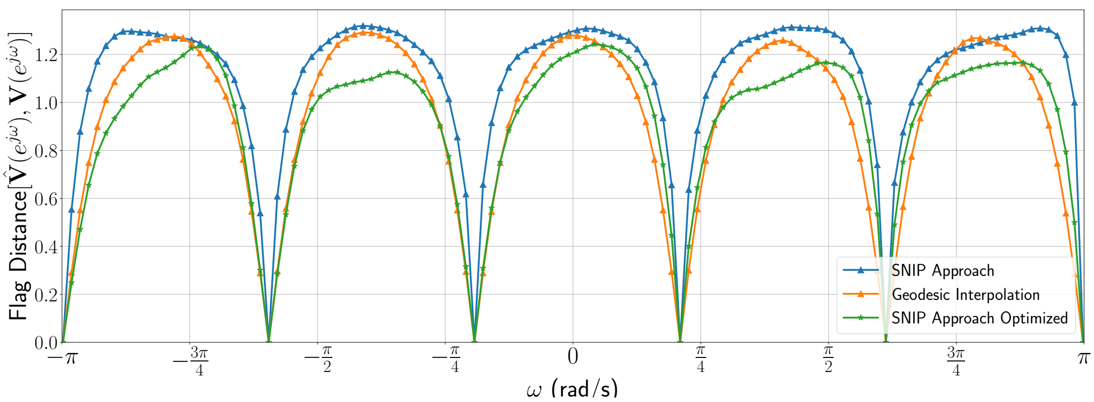


Figure 5.2: Frequency domain error measured for matrix all-pass filters constructed from samples for a 2×2 MIMO system, with Flag Distance as error measure.

It is evident from Figure 5.1 and Figure 5.2 that optimally choosing the $\mathbf{\Gamma}_i$ (group delay) matrices results in better performance. Thus, by constraining the Pick matrix to be positive definite and optimizing $\mathbf{\Gamma}_i$ matrices, we are able to achieve significantly improved interpolation, even while obtaining realizable (rational) matrix all-pass filters. Therefore, by using SNIP based matrix all-pass filtering technique, not only do we have computationally efficient LCCDE realizations, but we are also able to obtain performance comparable to a frequency-domain only technique like geodesic interpolation (which does not yield LCCDEs).

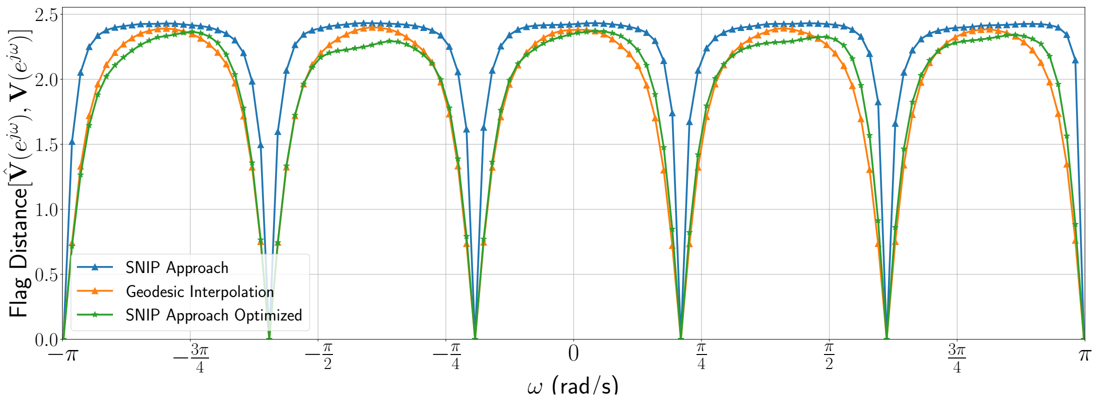


Figure 5.3: Frequency domain error measured for matrix all-pass filters constructed from samples for a 4×4 MIMO system, with Flag Distance as error measure.

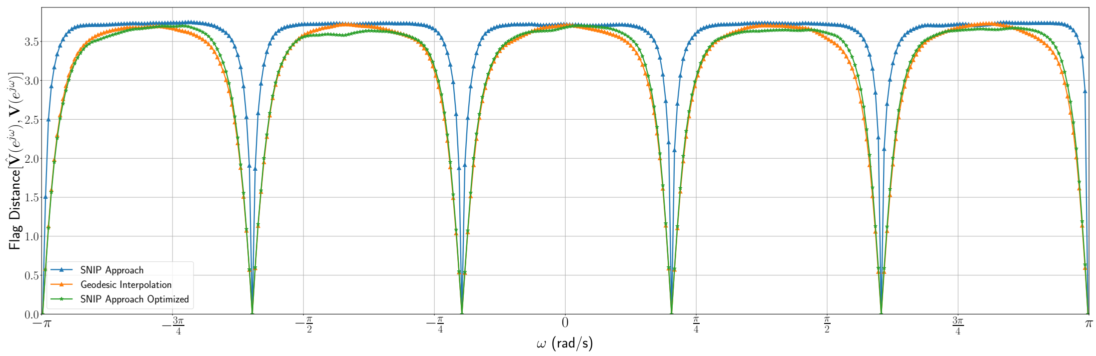


Figure 5.4: Frequency domain error measured for matrix all-pass filters constructed from samples for a 8×8 MIMO system, with Flag Distance as error measure.

Figure 5.3 and Figure 5.4 depicts the corresponding performance for a 4×4 system and a 8×8 system respectively, this confirms that the optimization based construction is effective for higher sizes as well and the performance compares favourably with the geodesic technique in spite of the realizability requirement.

In terms of complexity, using the SNIP approach would incur complexity in evaluating

$\mathbf{N}(z)$, $\mathbf{D}(z)$ (*from their coefficients*) and finally evaluating $\mathbf{G}(e^{j\omega}) = \mathbf{N}(e^{j\omega})\mathbf{D}(e^{j\omega})^{-1}$, while the geodesic based approach typically requires the computation of the matrix exponential for each subcarrier, thereby needing many more compuations than the SNIP approach. To practically compare the computational complexity of the geodesic as well as the SNIP based approach of evaluating the frequency response at the intermediate frequencies, we compare the time taken and memory used to construct a precoder from the given data set. The comparisons between the the geodesic interpolation and SNIP based matrix all-pass filter technique are listed in the Table 5.1, 6.1. These values are computed on a standard Google Colab GPU free tier, consisting of a 2 core Intel Xeon 2.2 GHz CPU with 13 GB RAM [23].

$m \times m$ **MIMO**	**SNIP Appproach (in ms)**	**Geodesic (in ms)**
$m = 2$	[HTML]FFFFFF**0.5665**	[HTML]FFFFFF**4.8178**
$m = 3$	[HTML]FFFFFF**0.5917**	[HTML]FFFFFF**5.7858**
$m = 4$	[HTML]FFFFFF**0.6006**	[HTML]FFFFFF**6.0138**
$m = 5$	[HTML]FFFFFF**0.6189**	[HTML]FFFFFF**6.0593**
$m = 6$	[HTML]FFFFFF**0.6542**	[HTML]FFFFFF**6.143**
$m = 7$	[HTML]FFFFFF**0.8176**	[HTML]FFFFFF**6.9861**

Table 5.1: Average time taken to construct a precoder matrix from the feedback data-set

$m \times m$ **MIMO**	**SNIP Appproach (in KB)**	**Geodesic (in KB)**
$m = 2$	[HTML]FFFFFF**3.3971**	[HTML]FFFFFF**6.4022**
$m = 3$	[HTML]FFFFFF**4.4769**	[HTML]FFFFFF**7.0285**
$m = 4$	[HTML]FFFFFF**5.7518**	[HTML]FFFFFF**7.6951**
$m = 5$	[HTML]FFFFFF**7.0732**	[HTML]FFFFFF**8.9847**
$m = 6$	[HTML]FFFFFF**8.6007**	[HTML]FFFFFF**10.7569**
$m = 7$	[HTML]FFFFFF**9.7012**	[HTML]FFFFFF**11.2269**

Table 5.2: Average peak size used to construct a precoder matrix from the feedback data-set

It is evident from the simulation results that, SNIP based approach takes much less time and consumes less memory than geodesic interpolation to construct a precoder (unitary matrix).

Chapter 6

Quantization of Matrix All-Pass Filter

We follow the methods described in [10], to quantize the unitary matrix space and also to perform the geodesic interpolation. Geodesic interpolation is the linear interpolation on the Grassmann manifold. By representing the precoder matrices (that are received through feedback) as points in the Grassmann manifold, we reformulate the problem as that of finding a curve (known as geodesic) in this manifold, which when appropriately sampled gives the precoding matrices for all the subcarriers.

The use of matrix all-pass filters for the interpolation of precoder matrices through limited feedback mandates for the quantization of matrix all-pass filter. In case of the previously discussed MIMO-OFDM system, the quantization can be carried out in two ways.

1. Construct the exact filter at the receiver with an unquantized data set and the quantize the coefficients of filter, feedback the quantized coefficients to the transmitter.
2. Quantize the input unitary matrix data set, so that transmitter can perform SNIP and interpolate the remaining precoders

6.1 Quantizing the coefficients of filter

The key challenges involved in quantization of matrix all-pass filter coefficients:

- The all-pass nature of the filter has to be retained even after quantization of the coefficients.

  If $\mathbf{G}(z) = \mathbf{N}(z)\mathbf{D}(z)^{-1}$ is all-pass, then the filter with transfer matrix $\mathbf{G}_Q(z)$

(obtained by quantizing coefficients of $\mathbf{G}(z)$) should also be all-pass.

- The space consisting of coefficients of matrix all-pass filters is not a vector space. Hence distance measures has to be chosen carefully.

6.1.1 Generating coefficient space

In order to be all-pass, the coefficients of filter should satisfy the condition described in equation (4.12). Solving the system equations in coefficients of $\mathbf{N}(z)$ and $\mathbf{D}(z)$ is very difficult to solve.
So, instead we use the boundary SNIP algorithm described in section 4.3. We construct a large number of data-sets containing randomly generated unitary matrices at certain frequencies. Using these data-sets as input to the boundary SNIP algorithm, we generate the corresponding matrix all-pass filters in large number.
Example of an element in the coefficient space: Consider the set of filters generated form the data-sets of size 4 each. In this case, $\mathbf{N}(z)$ and $\mathbf{D}(z)$ has 5 coefficients each.

$$\mathbf{N}(z) = \mathbf{N}_0 + \mathbf{N}_1 z + \mathbf{N}_2 z^2 + \mathbf{N}_3 z^3 + \mathbf{N}_4 z^4$$
$$\mathbf{D}(z) = \mathbf{D}_0 + \mathbf{D}_1 z + \mathbf{D}_2 z^2 + \mathbf{D}_3 z^3 + \mathbf{D}_4 z^4$$

The element in coefficient space will be of the form $\{\mathbf{N}_0, \mathbf{N}_1, \mathbf{N}_2, \mathbf{N}_3, \mathbf{N}_4, \mathbf{D}_0, \mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3, \mathbf{D}_4\}$

6.1.2 Quantization algorithm

Given the coefficient space, we would want to split the space into multiple clusters and then define a center to each cluster and thereby construct a code-book with these centers.

In order to do so, we need a good distance measure between the two points in the given space (i.e distance between two sets of coefficients).

Let two matrix all-pass filters be $\mathbf{G}_1(z)$ and $\mathbf{G}_2(z)$, such that

$$\mathbf{G}_1(z) = (\mathbf{N}_0^1 + \mathbf{N}_1^1 z + ... + \mathbf{N}_n^1 z^n)(\mathbf{D}_0^1 + \mathbf{D}_1^1 z + ... + \mathbf{D}_n^1 z^n)^{-1}$$
$$\mathbf{G}_2(z) = (\mathbf{N}_0^2 + \mathbf{N}_1^2 z + ... + \mathbf{N}_n^2 z^n)(\mathbf{D}_0^2 + \mathbf{D}_1^2 z + ... + \mathbf{D}_n^2 z^n)^{-1}$$

Distance Measure 1: Flatten each coefficient (of size $m \times m$ into 1D vector($m^2 \times 1$), then concatenate all the $2n + 2$ flattened coefficient vectors into a single dimension vector (of size $(2n+2)m^2 \times 1)$. Use Frobenius norm as distance measure between these coefficient flattened vectors of two filters.

Distance Measure 2: Consider the frequencies $\{e^{j\omega_1}, e^{j\omega_2}, e^{j\omega_3}, ..., e^{j\omega_K}\}$, where K is a fixed large number.

$$dist(\mathbf{G}_1(z), \mathbf{G}_2(z)) = \sum_{i=1}^{K} ||\mathbf{G}_1(e^{j\omega_i}) - \mathbf{G}_2(e^{j\omega_i})||_F^2 \tag{6.1}$$

Distance Measure 3: This is useful to rcenter the cluster. This measure relies on the linear approximation of logarithm of unitary matrix.

Consider the frequencies $\{e^{j\omega_1}, e^{j\omega_2}, e^{j\omega_3}, ..., e^{j\omega_K}\}$, where K is a fixed large number. Consider the cluster of size l.

$$\mathbf{S}_i := \sum_{j=1}^{K} \log\left(\mathbf{N}_i(z_j)\mathbf{D}_i^{-1}(z_j)\right), \quad \text{where} \quad i \in \{1, 2, ..., l\}$$
$$\mathbf{S} := \frac{\left(\sum_{i=1}^{l} \mathbf{S}_i\right)}{l}$$

Then the goal is to find $q \in \{1, 2, ..., l\}$ such that $||\mathbf{S} - \mathbf{S}_q||_F$ is minimum. So that, the coefficients of $\mathbf{N}_q(z)\mathbf{D}_q^{-1}(z)$ will be assigned as the centre of given cluster.

This would in particular helpful for the MIMO-OFDM systems case, as it uses the distance measure in the grassmann manifold [24].

Now, after choosing an appropriate distance measure between two sets of coefficients, we choose the size of desired codebook to be produced through quantization. Say we choose codeword length to be b-bits in the codebook then the size of codebook is $k = 2^b$.

Thereafter implement the following standard steps in K-Means clustering algorithm:

1. Initialize the k-centers for clusters
2. Assign each point in the coefficient space to the cluster with closest (in terms chosen *distance measure*) center.
3. Recompute the center of each cluster by choosing a point in cluster which has least sum of distances(in terms chosen *distance measure*) to all other points in the cluster.

4. Repeat the steps 2 and 3

6.1.3 Hermitian Mirror-Image property of coefficients

We know that, for a given scalar all-pass filter, we can *always* come up with equivalent transfer function such that the coefficients of numerator polynomial are just the mirrored conjugate of the coefficients of the denominator polynomial. The numerator and denominator polynomials form a Hermitian mirror-image pair as shown in equation (6.2)

$$G(z) = \frac{N(z)}{D(z)} = \frac{a_0 + a_1 z + a_2 z^2}{a_2^* + a_1^* z + a_0^* z^2} \tag{6.2}$$

But, there is no such property for matrix all-pass filters. In fact, we show that transfer matrices ($\mathbf{G}(z) = \mathbf{N}(z)\mathbf{D}(z)^{-1}$ with such mirrored hermitian structure relation between coefficients of $\mathbf{N}(z)$ and $\mathbf{D}(z)$ may not always be all-pass.

Consider the transfer matrix of the following form

$$\mathbf{G}(z) = \mathbf{N}(z)\mathbf{D}(z)^{-1}, \quad \text{where} \tag{6.3a}$$

$$\mathbf{N}(z) = \mathbf{A}_0 + \mathbf{A}_1 z + \mathbf{A}_2 z^2 \tag{6.3b}$$

$$\mathbf{D}(z) = \mathbf{A}_2^* + \mathbf{A}_1^* z + \mathbf{A}_0^* z^2 \tag{6.3c}$$

We know, from equation (4.12) that, If $\mathbf{G}(z)$ is all-pass then

$$\mathbf{N}^*(e^{j\omega})\mathbf{N}(e^{j\omega}) = \mathbf{D}^*(e^{j\omega})\mathbf{D}(e^{j\omega}) \quad \forall\omega \in (-\pi, \pi]$$

But, from equation (6.3), we observe that $[\mathbf{N}^*(e^{j\omega})] = e^{-2j\omega}\mathbf{D}(e^{j\omega})$. Therefore

$$\begin{aligned}\mathbf{N}^*(e^{j\omega})\mathbf{N}(e^{j\omega}) &= \mathbf{D}(e^{j\omega})\mathbf{D}^*(e^{j\omega}) \quad \forall\omega \in (-\pi, \pi] \\ &\neq \mathbf{D}^*(e^{j\omega})\mathbf{D}(e^{j\omega})\end{aligned}$$

Thus, the transfer matrices of the form described in equation (6.3) may not be all-pass always. Hence, we don't get any special benefits of relations between coefficients of $\mathbf{N}(z)$ and $\mathbf{D}(z)$, while performing quantization.

6.1.4 Simulations and Discussion

In the process of codebook generation for coefficients of SNIP based matrix all-pass filters, we implemented the K-Means algorithm as described in this chapter.

We have used *distance measure 2* in the step-2 and *distance measure 3* in the step-3 of K-means algorithm. These distance measures are chosen to be apt for faster convergence of algorithm on larger coefficient spaces and for larger data-sets. The performance error plots for these simulations are show in Figure 6.1 and Figure 6.2

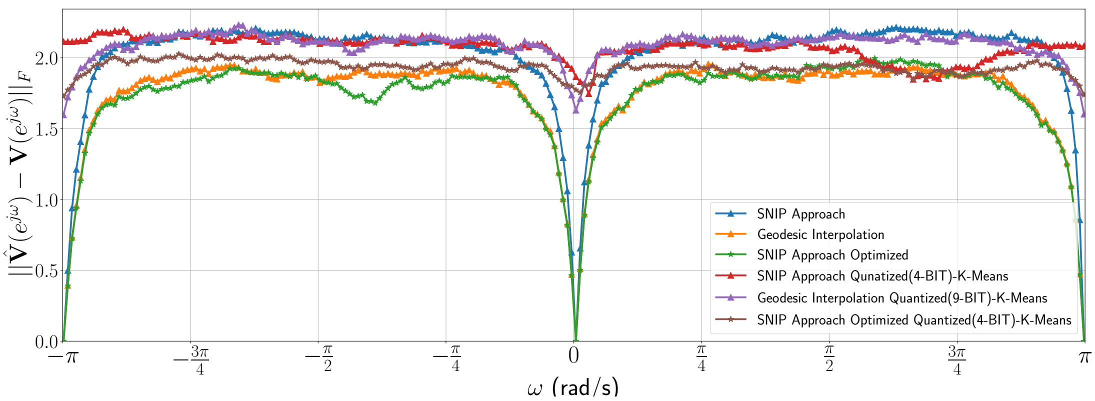


Figure 6.1: Frequency domain error measured for quantized matrix all-pass filters constructed from samples for a 2×2 MIMO system, with Frobenius norm as error measure.

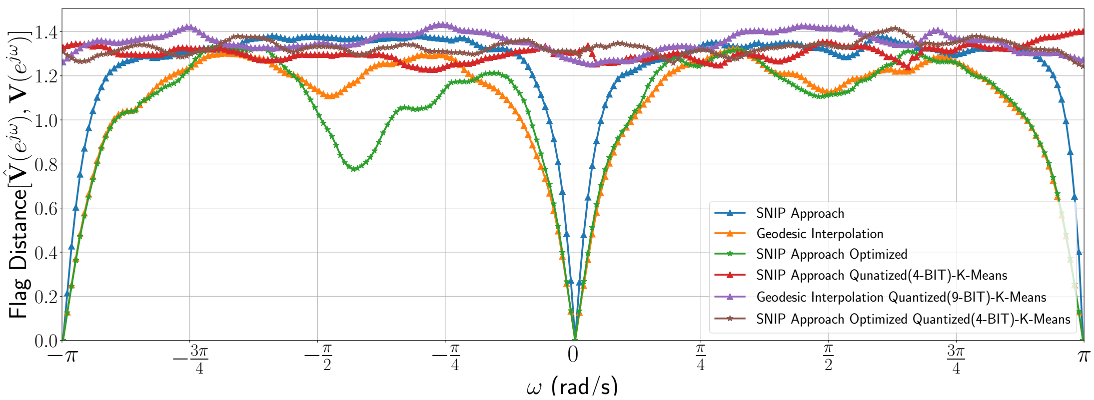


Figure 6.2: Frequency domain error measured for quantized matrix all-pass filters constructed from samples for a 2×2 MIMO system, with Flag distance as error measure.

NOTE: The codebook size used for SNIP based matrix all-pass filtering algorithm is different from the codebook size used for Geodesic interpolation. We know that, feedback information is different in these two methods. Say, the size of data-set used for geodesic interpolation is n_1 and the size of data-set used for matrix filtering be n_2. Therefor, the number of coefficient matrices in matrix filtering method are $2n_2 + 1$.

To this end, in order to allocate equal amount of feedback rate in both cases, the following relation should hold

$$\frac{\text{no. of bits used to quantize unitary matrix in geodesic interpolation}}{\text{no. of bits used to quantize each coefficient matrix}} = \frac{2n_2+1}{n_1}$$

6.2 Quantizing the Unitary Matrix Data-set

In this case we quantize the unitary matrices in the data-set, by using the K-means technique used to quantized the data-set for geodesic interpolation. Therefore, the feedback will be same for both geodesic and SNIP based filtering technique, in this case.

The transmitter performs precoder interpolation by using either construction of SNIP based matrix all-pass filter or geodesic interpolation accordingly.

6.2.1 Quantization and codebook generation

We follow the methods described in [10], to quantize the unitary matrix space just same as the technique used in the geodesic interpolation.

6.2.2 Simulations and Discussion

Size of space of unitary matrices used to quantize = 10^4.

Size of code book = 2^4 (4-bit).

MIMO channel model used for simulation: *ITU Vehicular A*.

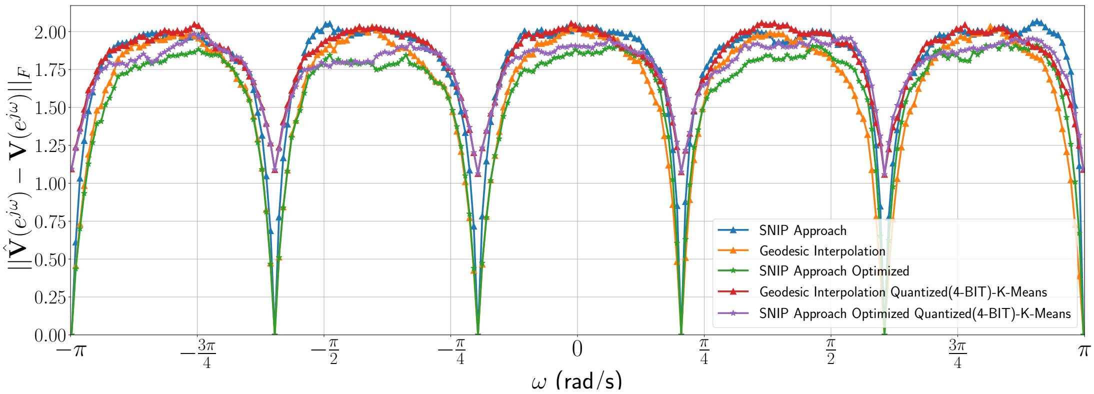


Figure 6.3: Frequency domain error measured for quantized matrix all-pass filters constructed from samples for a 2 × 2 MIMO system, with Frobenius norm as error measure.

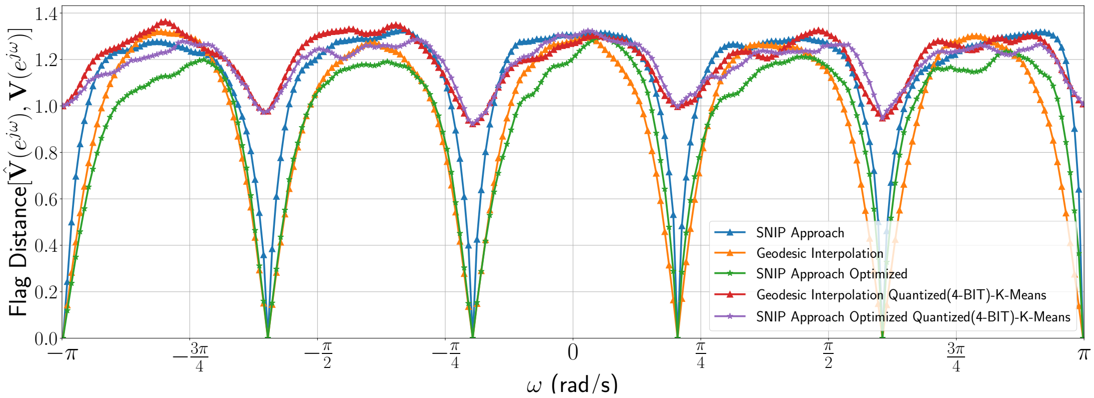


Figure 6.4: Frequency domain error measured for quantized matrix all-pass filters constructed from samples for a 2×2 MIMO system, with Flag distance as error measure.

We observe that the SNIP approach with group delay optimization performs better than geodesic interpolation even after quantization, and the dip in the error plots at the interpolating points can be seen.

It is evident from the simulations that the quantization technique described in section 6.2 performs much better than the quantization method used in section 6.1.

Having said that, it is important to note that the method in section 6.2 requires slightly higher computations at transmitter, when compared to method in section 6.1.

However, if the total number of subcarriers is greater than $\{3+n\}$ (where n is the size of the feedback data-set), then the SNIP based approach will still have huge computational advantage than geodesic interpolation. This happens because, the transmitter needs to perform the SNIP-based filter construction only once to get the coefficients of matrix all-pass filter and thereafter uses them to obtain the remaining precoders easily, whereas in case of geodesic interpolation the transmitter has to perform matrix exponential for every precoder interpolation.

For most of the MIMO systems, the simulations show that

$$r := \left\lfloor \frac{T_{filter}}{T_{geodesic} - T_{SNIP}} \right\rfloor \leq 3 \quad \text{, where}$$

T_{filter} =Average time consumed by transmitter to construct the matrix all-pass filter

$T_{geodesic}$ =Average time required to interpolate a precoder matrix by geodesic interpolation

T_{SNIP} =Average time required to interpolate a precoder by SNIP based filtering

$m \times m$ **MIMO**	T_{filter}**(in ms)**	T_{SNIP}**(in ms)**	$T_{geodesic}$**(in ms)**	r
$m = 2$	[HTML]FFFFFF**12.21**	[HTML]FFFFFF**0.58**	[HTML]FFFFFF**5.75**	[HTML]FFFF
$m = 3$	[HTML]FFFFFF**20.52**	[HTML]FFFFFF**0.59**	[HTML]FFFFFF**7.87**	[HTML]FFFF
$m = 4$	[HTML]FFFFFF**21.93**	[HTML]FFFFFF**0.64**	[HTML]FFFFFF**8.47**	[HTML]FFFF
$m = 5$	[HTML]FFFFFF**23.30**	[HTML]FFFFFF**0.66**	[HTML]FFFFFF**8.87**	[HTML]FFFF
$m = 6$	[HTML]FFFFFF**26.09**	[HTML]FFFFFF**0.72**	[HTML]FFFFFF**9.75**	[HTML]FFFF
$m = 7$	[HTML]FFFFFF**28.07**	[HTML]FFFFFF**0.72**	[HTML]FFFFFF**10.41**	[HTML]FFFF

Table 6.1: Values of different parameters used to compete the computational complexity between geodesic interpolation and SNIP based filtering

Chapter 7

Achievable Rates Using Matrix All-Pass Interpolation

Inorder to compare the performance of quantized matrix all-pass filter with the quantized geodesic interpolation, apart from the error measures, the achievable rate is also a reliable measure from MIMO-OFDM perspective.

This chapter describes the formulas to compute the achievable rates and the simulation results.

Consider the following channel model of a MIMO-OFDM system

$$\mathbf{y}[k] = \mathbf{H}[k]\mathbf{x}[k] + w[k], k \in \{0, 1, 2, ..., N_{FFT} - 1\} \tag{7.1}$$

The SVD of $\mathbf{H}[k]$ is

$$\mathbf{H}[k] = \mathbf{U}[k]\sigma[k]\mathbf{V}^H[k] = \sum_{i=1}^{N_s} \sigma_i[k]\mathbf{u}_i[k]\mathbf{v}_i^H[k] \tag{7.2}$$

It is known that inorder to achieve the maximum rate the transmitter needs to full knowledge of precoder matrices $\mathbf{V}[k]$ for all subcarriers (see [20, 25]). As we know, in a limited feedback setting the receiver estimates the precoder matrices $\mathbf{V}[k]$ for all subcarriers, but feedbacks only the quantized information of few subcarriers (due to small feedback rate). Say,

$$\mathbf{V}[k] \xrightarrow{\text{QUANTIZE}} \hat{\mathbf{V}}_{eq}[k] \tag{7.3}$$

When a linear receiver is employed, the symbols are recovered using the following receiver

$$\mathbf{G}[k] = \mathbf{F}[k]\mathbf{H}_{eq}[k] \tag{7.4}$$

$$\text{where } \mathbf{F}[k] = (P\hat{\mathbf{V}}_{eq}[k]^H H[k]^H H[k]\hat{\mathbf{V}}_{eq}[k] + a\mathbf{I}_{N_s}) \tag{7.5}$$

In this case, the achievable rate for the k-th subcarrier can be expressed as [20]

$$I_{LR}(\hat{\mathbf{V}}_{eq}[k]) = \sum_{i=1}^{N_s} \log_2(1 + \gamma_i[k]) \tag{7.6}$$

where $\gamma_i[k] = (\mathbf{F}[k]_{ii})^{-1} - a$ is the effective SNR for data received on the i-th singular vector on subcarrier k.

NOTE: $a = 0$ for Zero Forcing (ZF) inversion and $a = 1$ for minimum mean square error (MMSE) inversion.

7.1 Simulation and Discussion

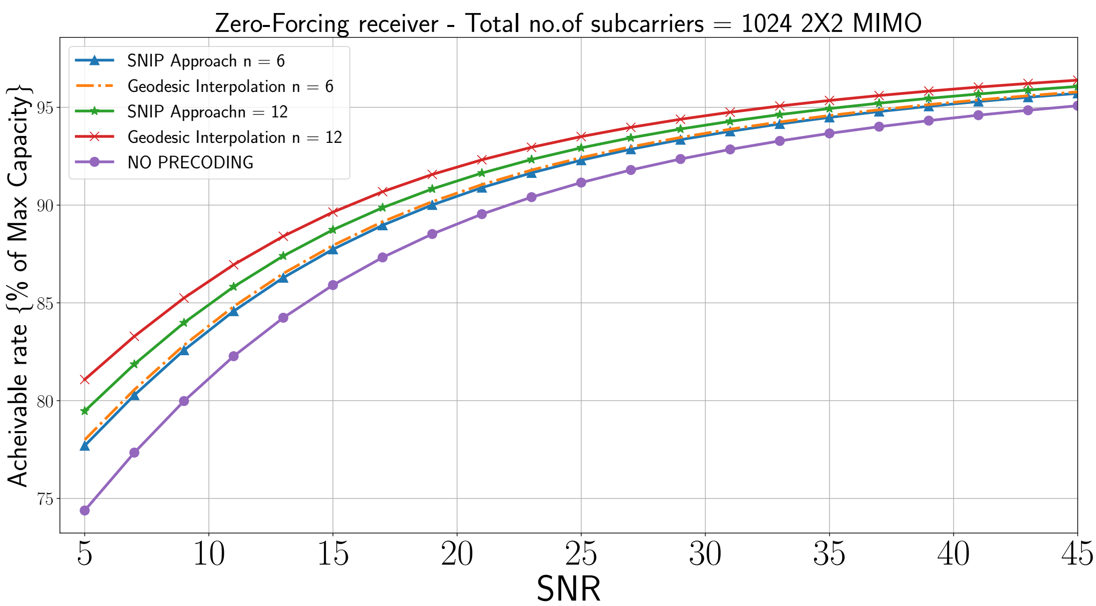


Figure 7.1: Achievable rates by a linear receiver with zero-forcing equalizer for a 2×2 MIMO-OFDM system, without any quantization

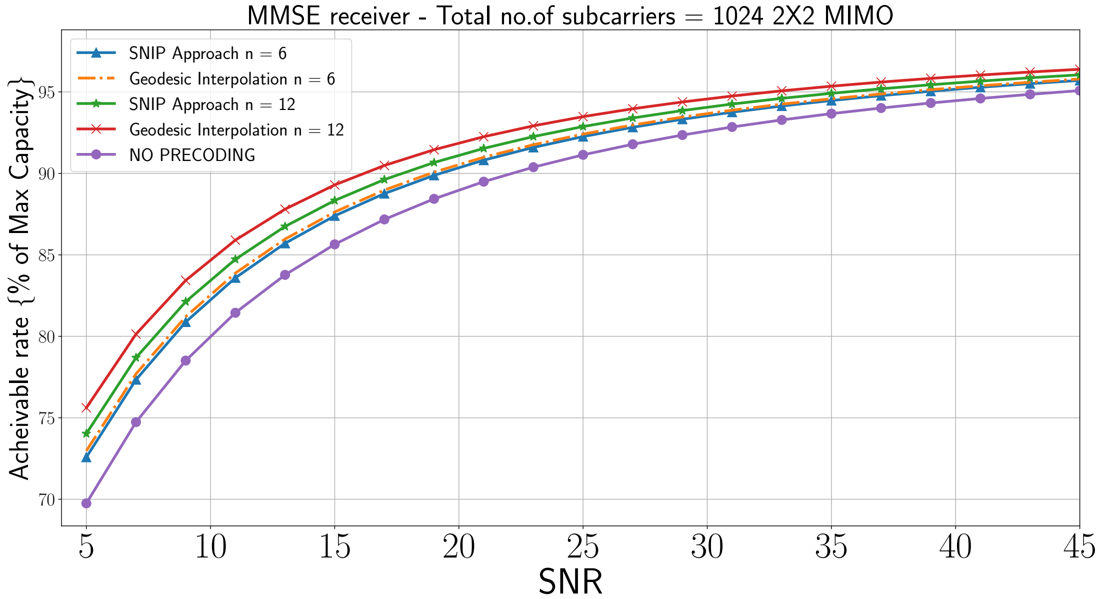


Figure 7.2: Achievable rates by a linear receiver with MMSE equalizer for a 2×2 MIMO-OFDM system, without any quantization

The figures Figure 7.1 and Figure 7.2 depicts the achievable rates by the SNIP based matrix all-pass filtering and geodesic interpolation on a 2×2 MIMO-OFDM system with a total of 1024 subcarriers and the feedback data-set size n.

The figure Figure 7.1 corresponds to linear receiver with Zero-Forcing (ZF) equalizer and the figure Figure 7.2 corresponds to linear receiver with Minimum Mean Squared Error (MMSE) equalizer.

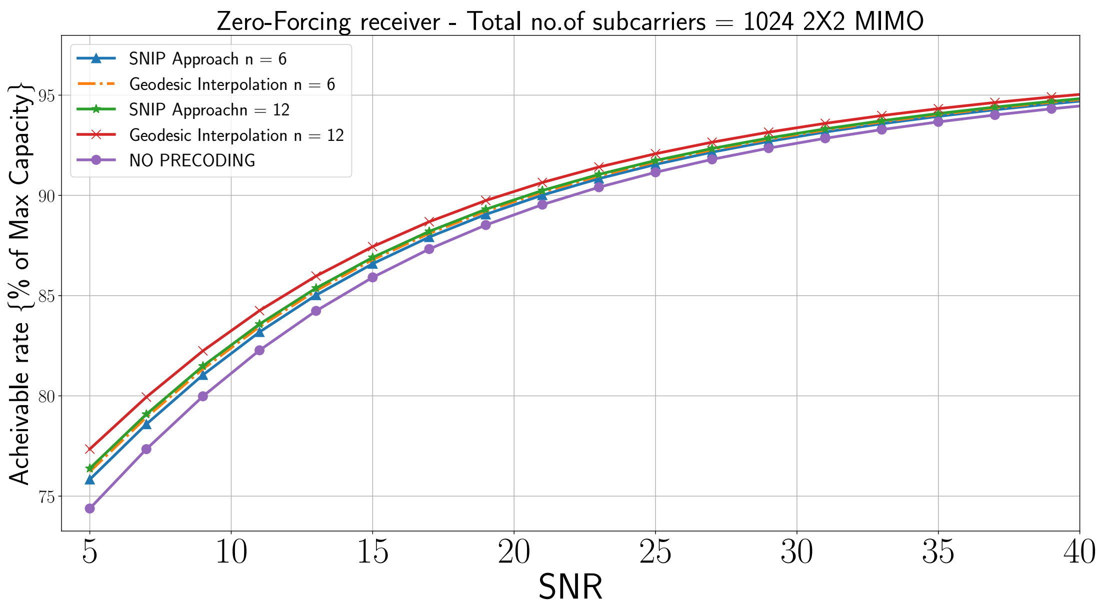


Figure 7.3: Achievable rates by a linear receiver with zero-forcing equalizer for a 2×2 MIMO-OFDM system, with quantization

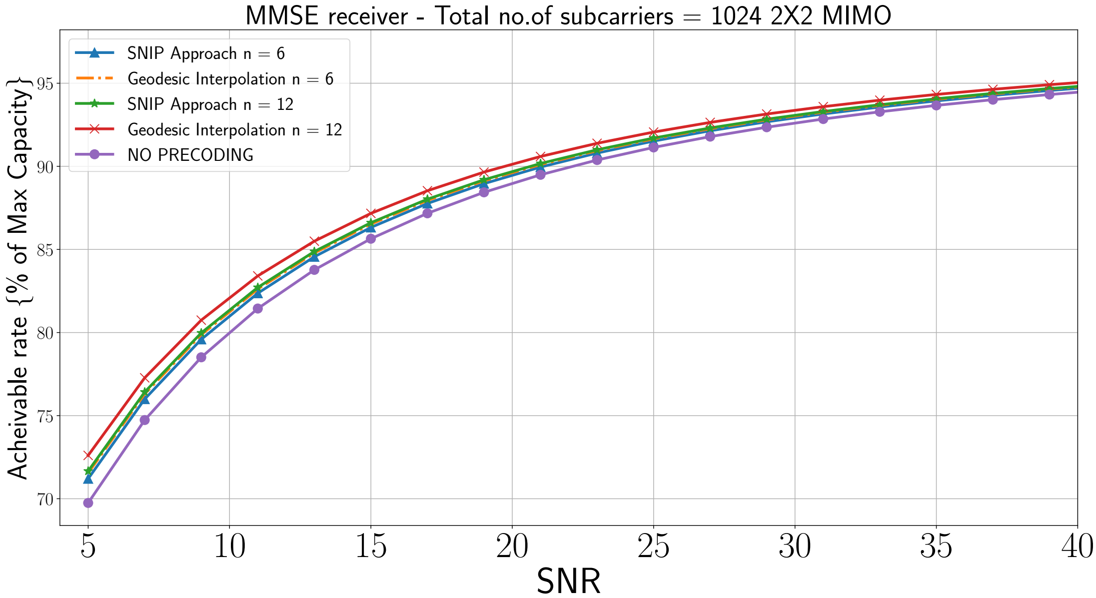


Figure 7.4: Achievable rates by a linear receiver with MMSE equalizer for a 2×2 MIMO-OFDM system, with quantization

It is evident from the plots that greater rates can be achieved by increasing the feedback data-set size (as expected). The performance of SNIP based filtering and Geodesic interpolation are closely comparable, however the filtering approach provides compact implementation and computational benefits.

Chapter 8

Matrix All-Pass Filter Design With Left Coprime Factorization

This chapter provides a brief overview of left co-prime factorized ($\mathbf{G}(z) = \mathbf{D}^{-1}(z)\mathbf{N}(z)$) matrix all-pass filter design and highlights the differences between this design and the right co-prime factorized ($\mathbf{G}(z) = \mathbf{N}(z)\mathbf{D}^{-1}(z)$) design discussed in chapter 4.

Let us look at look at the practical advantage of this left co-prime factorized matrix all-pass filter. We know, that computation cost and compactness in implementation has a significant importance in IIR digital discrete time filters.

Consider a LTI system with transfer matrix $\mathbf{G}(z) = \mathbf{D}^{-1}(z)\mathbf{N}(z)$, the input-output relation is

$$\mathbf{Y}(z) = \mathbf{G}(z)\mathbf{X}(z) \tag{8.1a}$$

$$= \mathbf{D}^{-1}(z)\mathbf{N}(z)\mathbf{X}(z) \tag{8.1b}$$

$$\Rightarrow \mathbf{D}(z)\mathbf{Y}(z) = \mathbf{N}(z)\mathbf{X}(z) \tag{8.1c}$$

Consider,

$$\mathbf{N}(z) = \mathbf{P}_0 + \mathbf{P}_1 z^{-1} + \mathbf{P}_2 z^{-2} \quad \text{and} \quad \mathbf{D}(z) = \mathbf{Q}_0 + \mathbf{Q}_1 z^{-1} + \mathbf{Q}_2 z^{-2}$$

$$\Rightarrow \mathbf{Q}_0\mathbf{y}[n] + \mathbf{Q}_1\mathbf{y}[n-1] + \mathbf{Q}_2\mathbf{y}[n-2] = \mathbf{P}_0\mathbf{x}[n] + \mathbf{P}_1\mathbf{x}[n-1] + \mathbf{P}_2\mathbf{x}[n-2]$$

Since, $\mathbf{N}(z)$ and $\mathbf{D}(z)$ are simple polynomials in z, the equation (8.1c) gives a LCCDE form, which gives compact realizations for direct time domain implementation. In fact, it gives simplest Direct form - I and Direct form - II realizations.

Now, we dive right into design of discrete matrix all-pass filter with left coprime factorization. The problem statement is same as the one mentioned in 4.1

Similar to the section 4.3, we carry out the induction based solution construction in this problem too.

Base Step: $n = 1, \mathbb{D} = (\omega_1, \mathbf{B}_1, \mathbf{\Gamma}_1)$.
Let, $z = e^{j\omega}$ and $z_1 = e^{j\omega_1}$. We first construct the following matrix polynomials. Define,

$$\begin{aligned} \mathbf{N}(z) &:= (z - z_1)\mathbf{I}_m + \frac{(\mathbf{A}_1 - \mathbf{I}_m)\mathbf{\Gamma}_1^{-1} z_1}{(1 + z_1)}(1 + z) \\ \mathbf{D}(z) &:= (z - z_1)\mathbf{I}_m + \frac{(\mathbf{A}_1 - \mathbf{I}_m)\mathbf{\Gamma}_1^{-1}\mathbf{A}^* z_1}{(1 + z_1)}(1 + z) \end{aligned} \tag{8.2}$$

Since, $\mathbf{\Gamma}_1$ is positive definite, $\mathbf{N}(z)$ and $\mathbf{D}(z)$ are well-defined. Correspondingly, we define $\mathbf{G}(z) := \mathbf{D}(z)^{-1}\mathbf{N}(z)$. This $\mathbf{G}(z)$ satisfies the constraint mentioned in (4.1a), (4.1b), and (4.1c). Thus, the base step is verified.

Inductive Step: We assume that our problem is solvable for $n-1$ points, and use this to prove that the problem is solvable for n points. First we suitably modify the given data set of $n-1$ points. To this end, we define the following $2m \times 2m$ matrix,

$$\mathbf{H}(z) = \begin{bmatrix} 2(z - z_1)\mathbf{I}_m - (z+1)(z_1+1)\mathbf{\Gamma}_1^{-1} & (z+1)(z_1+1)\mathbf{\Gamma}_1^{-1}\mathbf{A}_1^* \\ -(z+1)(z_1+1)\mathbf{A}_1\mathbf{\Gamma}_1^{-1} & 2(z - z_1)\mathbf{I}_m + (z+1)(z_1+1)\mathbf{A}_1\mathbf{\Gamma}_1^{-1}\mathbf{A}_1^* \end{bmatrix}.$$

Now, for each $i \in \{1, 2, \ldots n-1\}$, we define a modified data set as follows:

$$\begin{aligned} \widehat{\mathbf{B}}_i &= \mathbf{H}(e^{j\omega_{i+1}})\begin{bmatrix}\mathbf{I}_m & \mathbf{A}_{i+1}^T\end{bmatrix}^T \\ \widehat{\mathbf{\Gamma}}_i &= \mathbf{\Gamma}_{i+1} - \frac{(\mathbf{I}_m - \mathbf{A}_{i+1}^*\mathbf{A}_1){\mathbf{\Gamma}_1}^{-1}(\mathbf{I}_m - \mathbf{A}_1^*\mathbf{A}_{i+1})}{(1 - e^{j(\omega_{i+1} - \omega_1)})(1 - e^{j(\omega_1 - \omega_{i+1})})}. \end{aligned} \tag{8.3}$$

Both of the above modifications result in new data sets that also satisfy the conditions outlined in the section 4.2, since $\widehat{\mathbf{B}}_i$s are neutral and $\widehat{\mathbf{\Gamma}}_i$s are still Hermitian. Thus, the modified data set is $\widehat{\mathbb{D}} := \{(\omega_{i+1}, \widehat{\mathbf{B}}_i, \widehat{\mathbf{\Gamma}}_i) |\ i = 1, ..., n-1\}$.

It is important to note that the modifications in (8.3) are performed so that the Pick matrix of the new data set ($\widehat{\mathbb{D}}$) is the Schur complement of the Pick matrix of the original input data set ($\mathbb{D}$) with respect to $\mathbf{\Gamma}_1$ [17]. Thus, using the Schur complement property, we can argue that if the original Pick matrix for the data set $\mathbb{D}$ is positive definite, then the new Pick matrix for data set $\widehat{\mathbb{D}}$ will also be positive definite.

Using the original induction assumption, we know that we can solve for $\widehat{\mathbf{N}}(z), \widehat{\mathbf{D}}(z) \in \mathbb{C}^{m\times m}$ such that $\widehat{\mathbf{G}}(z) = \widehat{\mathbf{D}}(z)^{-1}\widehat{\mathbf{N}}(z)$ satisfies the matrix all-pass filter design problem for the new data set $\widehat{\mathbb{D}}$ containing $n-1$ points. Therefore, for $z_i := e^{j\omega_i}$, the following holds

$$\begin{bmatrix}\widehat{\mathbf{N}}(z_{i+1}) & -\widehat{\mathbf{D}}(z_{i+1})\end{bmatrix}\widehat{\mathbf{B}}_i = 0 \quad \forall \quad i = 1, \ldots, N-1$$

$$\widehat{\mathbf{B}}_i = \mathbf{H}(z_{i+1})\mathbf{B}_{i+1} \Rightarrow \begin{bmatrix}\widehat{\mathbf{N}}(z_{i+1}) & -\widehat{\mathbf{D}}(z_{i+1})\end{bmatrix}\mathbf{H}(z_{i+1})\mathbf{B}_{i+1} = 0.$$

We know that $\mathbf{H}(z_1)\mathbf{B}_1 = \mathbf{0}_{m\times 2m}$. Therefore, $\forall \quad i = 1, \ldots, N$, we have

$$\begin{bmatrix}\mathbf{N}(z) & -\mathbf{D}(z)\end{bmatrix} := \begin{bmatrix}\widehat{\mathbf{N}}(z) & -\widehat{\mathbf{D}}(z)\end{bmatrix}\mathbf{H}(z) \Rightarrow \mathbf{D}(z_i)^{-1}\mathbf{N}(z_i) = \mathbf{A}_i. \tag{8.4}$$

We now verify the unitary nature of $\mathbf{D}(e^{j\omega})^{-1}\mathbf{N}(e^{j\omega})$ as follows:

$$\begin{aligned}
\mathbf{N}(z)\mathbf{N}^*(z) - \mathbf{D}(z)\mathbf{D}^*(z) &= \begin{bmatrix}\mathbf{N}(z) & -\mathbf{D}(z)\end{bmatrix}\begin{bmatrix}\mathbf{I}_m & \mathbf{0}_{m\times m}\\ \mathbf{0}_{m\times m} & -\mathbf{I}_m\end{bmatrix}\begin{bmatrix}\mathbf{N}^*(z)\\ -\mathbf{D}^*(z)\end{bmatrix}\\
&= \begin{bmatrix}\widehat{\mathbf{N}}(z) & -\widehat{\mathbf{D}}(z)\end{bmatrix}[\mathbf{H}(z)]\mathbf{J}[\mathbf{H}(z)]^*\begin{bmatrix}\widehat{\mathbf{N}}^*(z)\\ -\widehat{\mathbf{D}}^*(z)\end{bmatrix}\\
&= \frac{-4(z_1-z)^2}{z_1 z}\begin{bmatrix}\widehat{\mathbf{N}}(z) & -\widehat{\mathbf{D}}(z)\end{bmatrix}\mathbf{J}\begin{bmatrix}\widehat{\mathbf{N}}^*(z)\\ -\widehat{\mathbf{D}}^*(z)\end{bmatrix}\\
&= \frac{-4(e^{j\omega_1}-e^{j\omega})^2}{e^{j(\omega_1+\omega)}}\left(\widehat{\mathbf{N}}(e^{j\omega})\widehat{\mathbf{N}}^*(e^{j\omega}) - \widehat{\mathbf{D}}(e^{j\omega})\widehat{\mathbf{D}}^*(e^{j\omega})\right) = 0.
\end{aligned} \tag{8.5}$$

The last equality can be inferred from the original induction assumption that $\widehat{\mathbf{D}}(e^{j\omega})^{-1}\widehat{\mathbf{N}}(e^{j\omega})$ satisfies the matrix all-pass filter constraints. Therefore, $\{\mathbf{N}(e^{j\omega})\mathbf{N}^*(e^{j\omega}) - \mathbf{D}(e^{j\omega})\mathbf{D}^*(e^{j\omega}) = 0\} \quad \forall\omega \in (-\pi, \pi]$, which implies $\mathbf{D}(e^{j\omega})^{-1}\mathbf{N}(e^{j\omega})$ is unitary for all $\omega \in (-\pi, \pi]$. Thus, we expand the expression of the solution for the original data set of n points as follows (from equation (8.4)):

$$\begin{aligned}
\mathbf{N}(z) &:= [2(z-z_1)\mathbf{I}_m - (z+1)(z_1+1)\mathbf{\Gamma}_1^{-1}]\widehat{\mathbf{N}}(z) + [(z+1)(z_1+1)\mathbf{A}_1\mathbf{\Gamma}_1^{-1}]\widehat{\mathbf{D}}(z)\\
\mathbf{D}(z) &:= [2(z-z_1)\mathbf{I}_m + (z+1)(z_1+1)\mathbf{A}_1\mathbf{\Gamma}_1^{-1}\mathbf{A}_1^*]\widehat{\mathbf{D}}(z) - [(z+1)(z_1+1)\mathbf{\Gamma}_1^{-1}\mathbf{A}_1^*]\widehat{\mathbf{N}}(z)
\end{aligned}$$

$$\mathbf{G}(z) := \mathbf{D}(z)^{-1}\mathbf{N}(z).$$

This $\mathbf{G}(z)$ satisfies the conditions in (4.1a), (4.1b), and (4.1c). This completes the mathematical induction steps.

8.1 Simulation results

The simulation results Simulating this left coprime factorized matrix all-pass filter design on the same data-set used in chapter 5 gives the error plots shown in Figure 8.1 and Figure 8.3.

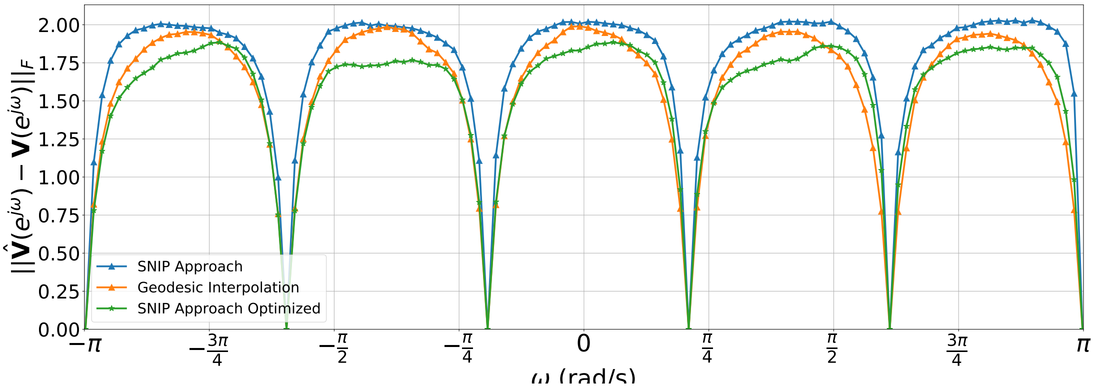


Figure 8.1: Frequency domain error measured for matrix all-pass filters constructed from samples for a 2×2 MIMO system, with Frobenius norm as error measure.

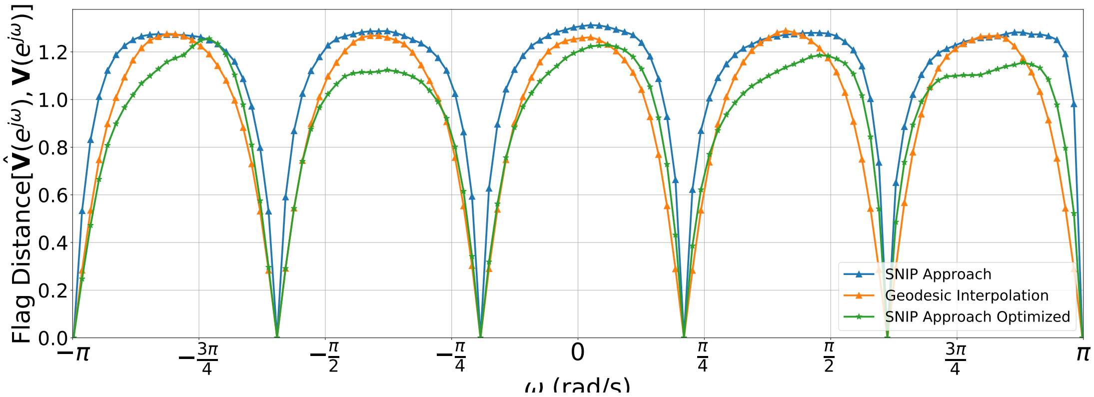


Figure 8.2: Frequency domain error measured for matrix all-pass filters constructed from samples for a 2×2 MIMO system, with Flag Distance as error measure.

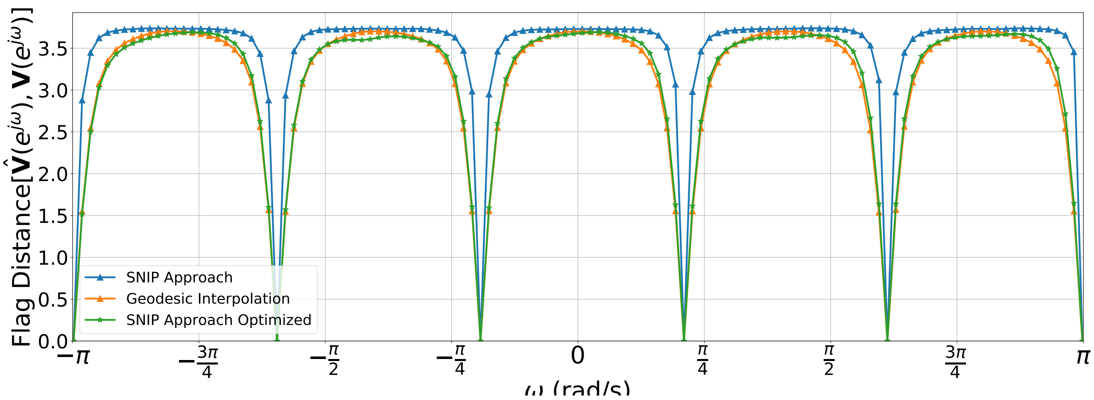


Figure 8.3: Frequency domain error measured for matrix all-pass filters constructed from samples for a 8×8 MIMO system, with Flag Distance as error measure.

Chapter 9

Lattice Structure Realization for Matrix All-Pass Filter

This chapter is the generalization of the work done in [26].

Given an all-pass transfer matrix of the following form

$$\mathbf{G}_n(z) = \mathbf{N}_n(z)\mathbf{D}_n^{-1}(z). \tag{9.1}$$

where n is the order of the matrix polynomials $\mathbf{N}(z)$ and $\mathbf{D}(z)$.

Our goal is to extract the unitary two-pair chain parameters $\{\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n, \mathbf{D}_n\}$ such that the remainder polynomial matrix is a unitary matrix of the form

$$z^{-1}\mathbf{G}_{n-1} = z^{-1}\mathbf{N}_{n-1}\mathbf{D}_{n-1}^{-1} \tag{9.2}$$

So, for a given all-pass transfer matrix $\mathbf{G}_n(z)$, the extraction of matrix digital two pair, leading to remainder all-pass transfer matrix $\mathbf{G}_{n-1}(z)$ is depicted in Figure 9.1.

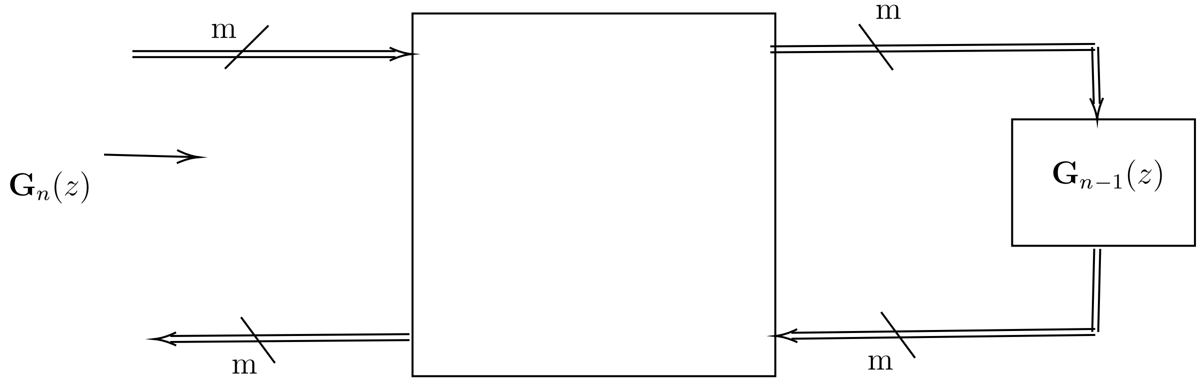


Figure 9.1: Matrix two-pair extraction

To this end, we divide the whole process of deriving $\mathbf{G}_{n-1}$ into two steps namely "*The Order Reduction Problem*" and "*Retention of unitary nature*".

9.1 The Order Reduction Problem

Given a unitary transfer matrix $\mathbf{G}_n(z)$ as defined in equation (9.1) with $\mathbf{D}_k(\infty) = \mathbf{I}_m$, we aim to solve the following problem: Extract a two pair with constant chain parameters $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ such that the remainder $\hat{\mathbf{G}}_n(z)$ is of the form

$$\hat{\mathbf{G}}_n(z) = \hat{\mathbf{N}}_n(z)\hat{\mathbf{D}}_n^{-1}(z) \tag{9.3}$$

$$\hat{\mathbf{N}}_n(z) := z^{-1}\sum_{i=0}^{n-1}\hat{\mathbf{A}}_{n,i}z^{-i}, \quad \hat{\mathbf{D}}_n(z) := \sum_{i=0}^{n-1}\hat{\mathbf{B}}_{n,i}z^{-i}. \tag{9.4}$$

As derived in [26], for order reduction, the following relations holds between $\mathbf{G}_n$ and $\hat{\mathbf{G}}_n$

$$\mathbf{D}_n(z) = \mathbf{A}\hat{\mathbf{D}}_n(z) + \mathbf{B}\hat{\mathbf{N}}_n(z) \tag{9.5}$$

$$\mathbf{N}_n(z) = \mathbf{C}\hat{\mathbf{D}}_n(z) + \mathbf{D}\hat{\mathbf{N}}_n(z) \tag{9.6}$$

We assume $\mathbf{D} = \mathbf{I}_m$ and solve for $\mathbf{A}, \mathbf{B}, \mathbf{C}$ that achieves the order reduction. Since $\hat{\mathbf{N}}_n(\infty) = 0$ (by definition), we see that $\mathbf{C}$ can be derived as follows

$$\mathbf{C}\hat{\mathbf{D}}_n(\infty) = \mathbf{N}_n(\infty). \tag{9.7}$$

From the equations (9.5) and (9.6), we observe that

$$\hat{\mathbf{D}}_n(z) = (\mathbf{A} - \mathbf{B}\mathbf{C})^{-1}[\mathbf{D}_n(z) - \mathbf{B}\mathbf{N}_n(z)] \tag{9.8}$$

Thus, from the definition of $\hat{\mathbf{D}}_k(z)$, it is evident that its order can be reduced to $n-1$ by choosing the $\mathbf{B}$ so that

$$[z^n\mathbf{D}_n(z)]_{z=0} = \mathbf{B}[z^n\mathbf{N}_n(z)]_{z=0} \tag{9.9}$$

$$\implies \mathbf{B} = [z^n\mathbf{D}_n(z)]_{z=0}[[z^n\mathbf{N}_n(z)]_{z=0}]^{-1} \tag{9.10}$$

Now, by substituting $z = \infty$ in equation (9.8), we arrive at $\mathbf{A} = \mathbf{I}_m$. Therefore, the chain parameters derived in this *order reduction step* are

$$\mathbf{A} = \mathbf{I}_m, \qquad \mathbf{B} = [z^n\mathbf{D}_n(z)]_{z=0}[[z^n\mathbf{N}_n(z)]_{z=0}]^{-1} \tag{9.11}$$

$$\mathbf{C} = \mathbf{N}_n(\infty), \qquad \mathbf{D} = \mathbf{I}_m \tag{9.12}$$

9.2 Retention of Unitary Nature

Firstly, let us look the conditions on chain parameters, inorder to retain the unitary nature in the remainder matrix transfer function.

Consider a general matrix digital two-pair {i.e a (m + m)-input and (m+m)-output system} as shown in Figure 9.2

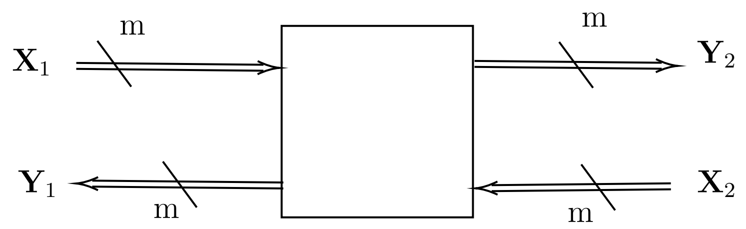


Figure 9.2: The digital matrix two-pair

$$\begin{bmatrix} \mathbf{X}_1 \\ \mathbf{Y}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_2 \\ \mathbf{X}_2 \end{bmatrix} \tag{9.13}$$

$$\begin{bmatrix} \mathbf{X}_1^* & -\mathbf{Y}_1^* \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_2^* & \mathbf{X}_2^* \end{bmatrix} \begin{bmatrix} \mathbf{A}^* & -\mathbf{C}^* \\ -\mathbf{B}^* & \mathbf{D}^* \end{bmatrix} \tag{9.14}$$

Now, multiplying the equation (9.14) with equation (9.13) we get,

$$\begin{bmatrix} \mathbf{X}_1^* & -\mathbf{Y}_1^* \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{Y}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_2^* & \mathbf{X}_2^* \end{bmatrix} \begin{bmatrix} \mathbf{A}^* & -\mathbf{C}^* \\ -\mathbf{B}^* & \mathbf{D}^* \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_2 \\ \mathbf{X}_2 \end{bmatrix} \tag{9.15}$$

$$\mathbf{X}_1^*\mathbf{X}_1 - \mathbf{Y}_1^*\mathbf{Y}_1 = \begin{bmatrix} \mathbf{Y}_2^* & \mathbf{X}_2^* \end{bmatrix} \begin{bmatrix} \mathbf{A}^* & -\mathbf{C}^* \\ -\mathbf{B}^* & \mathbf{D}^* \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_2 \\ \mathbf{X}_2 \end{bmatrix} \tag{9.16}$$

Since we are working with a system with unitary transfer matrix, $\mathbf{X}_1^*\mathbf{X}_1 - \mathbf{Y}_1^*\mathbf{Y}_1 = 0$. Therefore,

$$\begin{bmatrix} \mathbf{Y}_2^* & \mathbf{X}_2^* \end{bmatrix} \begin{bmatrix} \mathbf{A}^* & -\mathbf{C}^* \\ -\mathbf{B}^* & \mathbf{D}^* \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_2 \\ \mathbf{X}_2 \end{bmatrix} = 0 \tag{9.17}$$

$$\Longrightarrow \begin{bmatrix} \mathbf{A}^* & -\mathbf{C}^* \\ -\mathbf{B}^* & \mathbf{D}^* \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_{m\times m} \\ \mathbf{0}_{m\times m} & -\mathbf{I}_m \end{bmatrix} = \mathbf{J} \quad \text{(see chapter 4).} \tag{9.18}$$

Thus, in order to retain the unitary nature in the remainder polynomial matrix, the chain parameters $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ should satisfy the following conditions

$$\mathbf{A}^*\mathbf{A} - \mathbf{C}^*\mathbf{C} = \mathbf{I}_m \tag{9.19a}$$

$$\mathbf{D}^*\mathbf{D} - \mathbf{B}^*\mathbf{B} = \mathbf{I}_m \tag{9.19b}$$

$$\mathbf{A}^*\mathbf{B} - \mathbf{C}^*\mathbf{D} = 0 \tag{9.19c}$$

But, we observe that the chain parameters $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ derived in the section 9.1 are of the form

$$\mathbf{A}^*\mathbf{A} - \mathbf{C}^*\mathbf{C} = \mathbf{I}_m - [\mathbf{N}_n(\infty)]^*[\mathbf{N}_n(\infty)] \tag{9.20a}$$

$$\mathbf{D}^*\mathbf{D} - \mathbf{B}^*\mathbf{B} = \mathbf{I}_m - \mathbf{k}^*\mathbf{k} \tag{9.20b}$$

where $\mathbf{k} := [z^n\mathbf{D}_n(z)]_{z=0}[[z^n\mathbf{N}_n(z)]_{z=0}]^{-1}$. So, inorder to satisfy the conditions in equation (9.19) we scale the chain parameters $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ and redefine them as follows

$$\mathbf{A}_n := \left[\left[[\mathbf{I}_m - [\mathbf{N}_n(\infty)]^*[\mathbf{N}_n(\infty)]]^{1/2}\right]^*\right]^{-1} \tag{9.21a}$$

$$\mathbf{C}_n := [\mathbf{I}_m - [\mathbf{N}_n(\infty)]^*[\mathbf{N}_n(\infty)]\left[\left[[\mathbf{I}_m - [\mathbf{N}_n(\infty)]^*[\mathbf{N}_n(\infty)]]^{1/2}\right]^*\right]^{-1} \tag{9.21b}$$

$$\mathbf{D}_n := \left[\left[[\mathbf{I}_m - [\mathbf{k}]^*[\mathbf{k}]]^{1/2}\right]^*\right]^{-1} \tag{9.21c}$$

$$\mathbf{B}_n := \mathbf{k}\left[\left[[\mathbf{I}_m - [\mathbf{k}]^*[\mathbf{k}]]^{1/2}\right]^*\right]^{-1} \tag{9.21d}$$

These chain parameters defined in the equation (9.21) satisfy all three conditions in the equation (9.19). Now, the remainder polynomial $\mathbf{G}_{n-1} = \mathbf{N}_{n-1}\mathbf{D}_{n-1}^{-1}$ is defined as follows,

$$z^{-1}\mathbf{N}_{n-1}(z) = \mathbf{D}_n^{-1}\hat{\mathbf{N}}_n(z); \quad \mathbf{D}_{n-1}(z) = \mathbf{A}_n^{-1}\hat{\mathbf{D}}_n(z) \tag{9.22}$$

where, $\mathbf{A}_n$ and $\mathbf{D}_n$ are defined in equation (9.21).

Now, using these chain parameters $\{\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n, \mathbf{D}_n\}$, we extract the general $\mathbf{T}$ parameters (or scattering parameters) using the following relations.

$$\mathbf{T}_{11} = \mathbf{C}_n\mathbf{A}_n^{-1}; \qquad \mathbf{T}_{12} = \mathbf{D}_n - \mathbf{C}_n\mathbf{A}_n^{-1}\mathbf{B}_n \tag{9.23a}$$

$$\mathbf{T}_{21} = \mathbf{A}_n^{-1}; \qquad \mathbf{T}_{22} = -\mathbf{A}_n^{-1}\mathbf{B}_n \tag{9.23b}$$

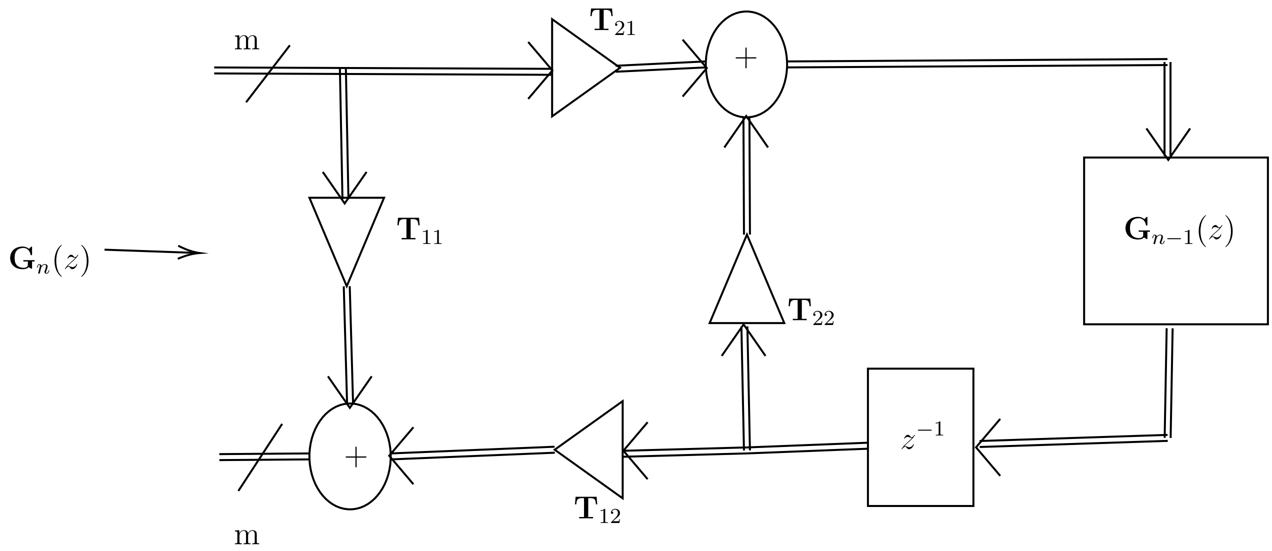


Figure 9.3: The matrix all-pass two pair lattice

Thus, by repeating the same procedure for $n-1$ times more, we get a constant matrix as remainder and thereby we achieve a lattice structure representation of a matrix all-pass filter just in terms of the constant **T**-parameter (or scattering parameter) matrices.

Moreover, from the simulations we observe that

$$\mathbf{T}_{11}^* = -\mathbf{T}_{22}; \quad \mathbf{T}_{21} = \mathbf{T}_{21}^*; \quad \mathbf{T}_{12} = \mathbf{T}_{12}^*$$
$$\mathbf{B}^* = \mathbf{C}; \quad \mathbf{A} = \mathbf{A}^*; \quad \mathbf{D} = \mathbf{D}^*$$

The following ideas may be considered for the future work in this direction:

- Cascading of multiple matrix all-pass filters becomes very easy by using the lattice structure realization. Thus filter might become scalable to data set size easily. The advantages of this scalability for matrix filters has to be explored.

- In scalar case the lattice structure provides a nice property of retaining the unitary nature even after quantizing the lattice coefficients, can we find any such property in lattice structure for matrix all-pass filters..?

- Lattice structure representation has many practical advantages in implementation of adaptive filtering and notch filtering, can we find such extensions through this matrix all-pass lattice structure.

Chapter 10

Conclusion

All-pass filtering of vector signals is typically done in the frequency domain using the DFT. However, this could lead to incorrect filtering, and may require more complex representation of filters. In this paper, we have presented a method to obtain a realizable (rational) matrix all-pass filter by extending the SNIP to the boundary case, leading to an LCCDE implementation. If the matrix valued phase response is specified only at n distinct values of Ω (or ω), we can obtain an n pole all-pass filter that exactly satisfies the phase constraints at these points. In addition, we show that, if the values of the group delay for the interpolating ω are specified, while often ensuring a compact representation. The group delay matrices at the interpolating points in the problem statement can be optimized or tuned to control the phase response for the remaining frequencies, which is also done in [7] for the scalar case. Simulations reveal that the method proposed can significantly outperform other optimization based approaches with much lower complexity. It is important to note that the performance of SNIP based matrix all-pass filter with group delay optimization is better than geodesic interpolation on the quantized unitary data set too. The right co-prime factorized matrix all-pass filters can be realized as lattice structures in cascaded form. SNIP based matrix all-pass filtering technique can be used to deign both right and left co-prime factorized matrix all-pass filters.

Future work would focus on stability of the solution under perturbation and approaches to minimize filter order as well as the problem of generating a good data sets that minimize the overfitting and undersampling for this boundary SNIP-based interpolation. The stability of the designed filter is also yet to be explored thoroughly. Benefits of matrix all-pass filters in terms of quantization and rounding off error in practical applications also constitute the problems with significant potential for further research.

Applications of lattice structure realization and their advantages are yet to be analyzed.

Appendix A

Discrete-Time Lossless Systems

This part focuses on proving the equation (4.7), using the literature of dissipative systems for discrete-time setting in [15]. We define

$$\mathbf{\Phi}(\zeta,\eta) = \sum_{k,l=0}^{n} \mathbf{\Phi}_{kl}\zeta^k\eta^l \in \mathbb{C}_s^{q\times q}[\zeta,\eta]$$

where $\mathbb{C}_s^{q\times q}[\zeta,\eta]$ denote the set of two-variable polynomial matrices satisfying $\mathbf{\Phi}(\zeta,\eta) = [\mathbf{\Phi}(\zeta,\eta)]^*$.

Let l_2^q be the set of square summable time-series vectors of size q. Consider discrete-time dynamical systems defined by $\Sigma = (\mathbb{Z}, \mathbb{C}^q, \mathscr{B})$, where $\mathscr{B} \subseteq (\mathbb{C}^q)^{\mathbb{Z}}$ is the set of trajectories satisfying some linear difference equations with constant coefficients. Let $\mathbf{w} = \mathbf{M}(\sigma)d$ be the image representation of the dynamical system Σ. Where, σ denotes a shift operator defined by $\sigma^T\mathbf{w}(t) := \mathbf{w}(t+T)$ and d is latent variable vector. It follows from the Proposition 3.2 in [15] that:

$$\text{For all } \mathbf{w} \in l_2^q \cap \mathscr{B}, \sum_{n=-\infty}^{n=\infty} \mathbf{Q_\Phi}(\mathbf{w})[n] = \mathbf{0}.$$

$$\Updownarrow$$

$$\text{There exists } a \quad \mathbf{\Psi}(\zeta,\eta) \in \mathbb{C}_s^{q\times q}[\zeta,\eta] \text{ such that } \forall n \in \mathbb{Z}$$

$$\mathbf{Q_\Psi}(\mathbf{w})[n+1] - \mathbf{Q_\Psi}(\mathbf{w})[n] = \mathbf{Q_\Phi}(\mathbf{w})[n] \quad \forall \mathbf{w} \in \mathscr{B}.$$

Now, substituting $\mathbf{w}[n] = \begin{pmatrix} \mathbf{x}[n] \\ \mathbf{y}[n] \end{pmatrix}$ gives

$$\mathbf{Q_\Psi}\begin{pmatrix} \mathbf{x}[n+1] \\ \mathbf{y}[n+1] \end{pmatrix} - \mathbf{Q_\Psi}\begin{pmatrix} \mathbf{x}[n] \\ \mathbf{y}[n] \end{pmatrix} = \mathbf{Q_\Phi}\begin{pmatrix} \mathbf{x}[n] \\ \mathbf{y}[n] \end{pmatrix}. \tag{A.1}$$

It is evident that, the infinite matrix (defined in [15]) for this case should be,

$$\tilde{\mathbf{\Phi}} = \mathbf{J} = \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_{m\times m} \\ \mathbf{0}_{m\times m} & -\mathbf{I}_m \end{bmatrix}$$

$$\text{so that, } \mathbf{Q}_{\mathbf{\Phi}}(\mathbf{w})[n] = \mathbf{x}^*[n]\mathbf{x}[n] - \mathbf{y}^*[n]\mathbf{y}[n]$$

We know from the definition of the Quadratic Difference Form in [15] that

$$\mathbf{Q}_{\mathbf{\Phi}}(\mathbf{w})[n] = \sum_{k,l} [\mathbf{x}^*[n+k] \quad \mathbf{y}^*[n+k]]\, \mathbf{\Phi}_{k,l} \begin{bmatrix} \mathbf{x}[n+l] \\ \mathbf{y}[n+l] \end{bmatrix}.$$

However, the quadratic difference form of our interest is as follows:

$$\mathbf{Q}_{\mathbf{\Phi}} \begin{pmatrix} \mathbf{x}[n] \\ \mathbf{y}[n] \end{pmatrix} = \begin{bmatrix} \mathbf{x}^*[n] & \mathbf{y}^*[n] \end{bmatrix} \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_{m\times m} \\ \mathbf{0}_{m\times m} & -\mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{x}[n] \\ \mathbf{y}[n] \end{bmatrix}$$

Therefore, for $k = 0, l = 0$,

$$\mathbf{Q}_{\mathbf{\Phi}} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} [n] = [\mathbf{x}^*[n] \quad \mathbf{y}^*[n]]\, \mathbf{\Phi}_{00} \begin{bmatrix} \mathbf{x}[n] \\ \mathbf{y}[n] \end{bmatrix}$$

where, $\mathbf{\Phi}_{00} = \mathbf{J}$. Thus, using (A.1), we observe that

$$\mathbf{x}^*[n]\mathbf{x}[n] - \mathbf{y}^*[n]\mathbf{y}[n] = \mathbf{Q}_{\mathbf{\Psi}} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} [n+1] - \mathbf{Q}_{\mathbf{\Psi}} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} [n] \tag{A.2}$$

Note: We refer to $\mathbf{Q}_{\mathbf{\Psi}} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$ as the *storage function*, and $\mathbf{x}^*[n]\mathbf{x}[n]-\mathbf{y}^*[n]\mathbf{y}[n]$ is referred to as the *dissipation rate* (or *supply rate*), and both are quadratic difference forms.

Computing the infinite summation on both sides of the equation (A.2) yields

$$\sum_{n=-\infty}^{0} \mathbf{x}^*[n]\mathbf{x}[n] - \mathbf{y}^*[n]\mathbf{y}[n] = \mathbf{Q}_{\mathbf{\Psi}} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} [n=1].$$

So, inorder to prove that $\sum_{n=-\infty}^{0} \mathbf{x}^*[n]\mathbf{x}[n] - \mathbf{y}^*[n]\mathbf{y}[n] > 0$ it is sufficient to show that $Q_{\mathbf{\Psi}} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} [1] > 0$

Claim 1: $\mathbf{Q_\Psi}\begin{pmatrix}\mathbf{x}[n]\\ \mathbf{y}[n]\end{pmatrix}$ is positive for all $n \in \mathbb{Z}$.

Note: The Proposition 3.2 in [15], which we have used to prove the difference equation relationship in (A.2), requires a stability condition on the dissipative system. Thus, we assume l_2 stability for our system.

Claim 2: The system is stable $\Rightarrow$ storage function is positive.

We know that, Storage function is always quadratic of state variables [16]. Therefore, we write Storage function as

$$\mathbf{Q_\Psi}\begin{pmatrix}\mathbf{x}[n]\\ \mathbf{y}[n]\end{pmatrix} = [\mathbf{q}[n]]^* \cdot \mathbf{K} \cdot [\mathbf{q}[n]] \tag{A.3}$$

The free evolution (i.e without input) can be written as a state space equation: $\mathbf{q}[n+1] = \mathbf{A} \cdot \mathbf{q}[n]$. Consider a situation where we start the system at $n = 0$, and no input is provided to the system (i.e $\mathbf{x}[n] = 0$). The system is then left to evolve over time (free evolution).

$$0 - \mathbf{y}^*[n]\mathbf{y}[n] = \mathbf{q}^*\,[n+1]\,\mathbf{K}\mathbf{q}[n+1] - \mathbf{q}^*[n]\mathbf{K}\mathbf{q}[n]$$

Now substitute,

$$\mathbf{q}[n+1] = \mathbf{A} \cdot \mathbf{q}[n] \quad \text{and} \quad \mathbf{y}[n] = c \cdot \mathbf{q}[n]$$

$$\begin{aligned}
&\Rightarrow - \mathbf{q}^*[n] \cdot c^* c \cdot \mathbf{q}[n] = \mathbf{q}^*[n]\mathbf{A}^*\mathbf{K}\mathbf{A}\mathbf{q}[n] - \mathbf{q}^*[n]\mathbf{K}\mathbf{q}[n]\\
&\Rightarrow \mathbf{q}^*[n]\,[\mathbf{A}^*\mathbf{K}\mathbf{A} - \mathbf{K} + c^*c]\,\mathbf{q}[n] = 0 \quad \forall n \in \mathbb{Z}\\
&\Rightarrow \mathbf{A}^*\mathbf{K}\mathbf{A} - \mathbf{K} = -c^*c \qquad (\textit{This is a Lyapunov equation})
\end{aligned}$$

Therefore, if $\mathbf{A}$ is stable[i.e {Poles of $\mathbf{A}$} lies inside Unit circle] then $\mathbf{K} \succ 0$ ($\mathbf{K}$ is positive definite). This proves *Claim 2.*

Thus, it follows from (A.3) that,

$$\mathbf{Q_\Psi}\begin{pmatrix}\mathbf{x}[n]\\ \mathbf{y}[n]\end{pmatrix} = \mathbf{q}^*[n] \cdot \mathbf{K} \cdot \mathbf{q}[n] > 0 \qquad \forall n \in \mathbb{Z}$$

This proves *Claim 1*, and thus we see $Q_{\mathbf{\Psi}}\begin{pmatrix}\mathbf{x}[1]\\ \mathbf{y}[1]\end{pmatrix} > 0$, which implies that

$$\sum_{n=-\infty}^{0} \mathbf{x}^*[n]\mathbf{x}[n] - \mathbf{y}^*[n]\mathbf{y}[n] > 0$$

for $\mathbf{x} \neq \mathbf{0}$. This proves (4.7).

Appendix B

Generalization of Matrix Filtering to Skew Hermitian

The philosophy behind the construction of the pick matrix and induction-based solution that was used in [5] for the vector all-pass filter design can be generalized to other interpolation problems with some modifications.

We hereby present one of such kind problem for interpolation of Skew Hermitian matrix data-set. (i.e we want to design a vector filter that outputs skew hermitian matrices at all frequencies). The problem statement is described as follows :
Given data $\mathbb{D} = \{ (\Omega_i, \mathbf{A}_i, \mathbf{\Gamma}_i) \mid$ i = 1,...,n $\}$, where $\Omega_i \in \mathbb{R}$, $\mathbf{A}_i \in \mathbb{C}^{m\times m}$ is skew-hermitian, $\mathbf{\Gamma}_i = \mathbf{\Gamma}_i^* \in \mathbb{C}^{m\times m}$ is positive semidefinite, we wish to obtain a rational transfer function matrix $\mathbf{G}(s) \in \mathbb{C}(s)^{m\times m}$ that satisfies

$$\mathbf{G}(\Omega_i) = \mathbf{A}_i \qquad \forall \quad i = 1, ..., n \tag{B.1a}$$

$$\mathbf{G}^*(\Omega_i) + \mathbf{G}(\Omega_i) = \mathbf{0}_m \quad \forall \qquad \Omega_i \in \mathbb{R} \tag{B.1b}$$

$$\begin{gathered} \text{if} \quad \mathbf{F}_i = -j\frac{d\mathbf{G}(\Omega_i)}{d\Omega} \qquad \forall \quad i = 1, ..., n \\ \text{then} \quad \mathbf{F}_i \quad \text{and } \mathbf{\Gamma}_i \quad \text{are similar matrices} \end{gathered} \tag{B.1c}$$

We observe that the Skew-hermitian interpolation is guaranteed by Equation (B.1b). The construction of solution will be very similar to the unitary case described in chapter 4.

B.1 Formulating Modified Pick Matrix

Modify the data-set $\mathbb{D}$ by replacing $\mathbf{A}_i \in \mathbb{C}^{m\times m}$ with $\mathbf{B}_i \in \mathbb{C}^{2m\times m}$

$$\mathbf{B}_i := \begin{bmatrix} \mathbf{I}_m \\ \mathbf{A}_i \end{bmatrix}, \qquad i = 1, 2, .., n. \tag{B.2}$$

Then inorder to retain the unitary nature of $\mathbf{A}_i$ we define *neutral* $\mathbf{B}_i$ that satisfy

$$\mathbf{B_i}^*\mathbf{J}\mathbf{B}_i = \mathbf{0}_{m\times m} \qquad \text{where} \quad \mathbf{J} = \begin{bmatrix} \mathbf{0}_{m\times m} & \mathbf{I}_m \\ \mathbf{I}_m & \mathbf{0}_{m\times m} \end{bmatrix} \tag{B.3}$$

Modified form of our data-set is $\mathbb{D} = \{ (\Omega_i,\ \mathbf{B}_i,\ \mathbf{\Gamma}_i) \mid \text{i} = 1,...,\text{n} \}$, where $\Omega_i \in \mathbb{R}$, $\mathbf{B}_i \in \mathbb{C}^{2m\times m}$ is neutral, $\mathbf{\Gamma}_i = \mathbf{\Gamma}_i^* \in \mathbb{C}^{m\times m}$ is positive semidefinite. we wish to obtain a rational transfer function matrix $\mathbf{G}(s) \in \mathbb{C}(s)^{m\times m}$ that satisfies Equation (B.1) with $\mathbf{A}_i$ replaced with $\mathbf{B}_{i,2}\ \mathbf{B}_{i,1}^{-1}$. where ,

$$\mathbf{B}_i = \begin{bmatrix} \mathbf{B}_{i,1}^T & \mathbf{B}_{i,2}^T \end{bmatrix}^T$$

$$\mathbf{B}_{i,1}, \mathbf{B}_{i,2} \in \mathbb{C}^{m\times m} \quad \forall \quad i = 1, ..., n$$

We now define $nm \times nm$ Pick matrix $\mathbf{P}$ as follows :

$$\mathbf{P}[(i-1)m+1 : im, (k-1)m+1 : km] = \mathbf{\Gamma}_i\delta_{ik} + \frac{\mathbf{B}_i{}^*\mathbf{J}\mathbf{B}_k(1-\delta_{ik})}{j(\Omega_k - \Omega_i)} \tag{B.4}$$

for $i, k \in \{1, .., n\}$.

B.2 Positive Definiteness of Pick Matrix:

Theorem 2. *Given data-set $\mathbb{D} = \{ (\Omega_i,\ \mathbf{A}_i,\ \mathbf{\Gamma}_i) \mid i = 1,...,n \}$ with Ω_i distinct, a rational transfer function matrix $\mathbf{G}(s) \in \mathbb{C}(s)^{m\times m}$ that satisfies Equation* (B.1) *exists only if the Pick matrix defined in equation* (B.4) *is positive definite*

Proof : We follow the similar technique followed in Theorem 1 . Here G is Skew-Hermitian. So, the supply rate for lossless dissipative dynamical systems is $\mathbf{u}^*\mathbf{y} + \mathbf{y}^*\mathbf{u}$. So,

$$\int_{-\infty}^{0} (\mathbf{u}^*\mathbf{y} + \mathbf{y}^*\mathbf{u})dt \geq \mathbf{0} \tag{B.5}$$

Now, consider $\mathbf{u} = \mathbf{v}e^{(\epsilon_i + j\Omega_i)t}$, where $\mathbf{v}$ is an arbitrary vector in $\mathbb{C}^m$, ϵ_i is a positive real number and Ω_i is the ith interpolation frequency . Since $\mathbf{y} = \mathbf{G}\mathbf{u}$, we get $\mathbf{y} = \mathbf{G}(\epsilon_\mathbf{i} + \mathbf{j}\mathbf{\Omega_i})\mathbf{v}e^{(\epsilon_i + j\Omega_i)t}$. Let $\mathbf{G_i} := \mathbf{G}(\epsilon_\mathbf{i} + \mathbf{j}\mathbf{\Omega_i})$.

Now substituting **u, y** in equation (B.5) we get

$$\int_{-\infty}^{0} (\mathbf{u}^*\mathbf{y} + \mathbf{y}^*\mathbf{u})dt = \frac{\mathbf{v}^*(\mathbf{G_i^*} + \mathbf{G_i})\mathbf{v}}{2\epsilon_i} \geq \mathbf{0}.$$

Taking limit $\epsilon_i \to 0^+$ in the above equation we get, $\lim_{\epsilon_i \to 0^+} \frac{\mathbf{v}^*(\mathbf{G_i^*}+\mathbf{G_i})\mathbf{v}}{2\epsilon_i} \geq \mathbf{0}$

Now, applying L'Hospital rule gives us the following,

$$\lim_{\epsilon_i \to 0^+} \frac{\mathbf{v}^*(\mathbf{G_i^*} + \mathbf{G_i})\mathbf{v}}{2\epsilon_i} = \lim_{\epsilon_i \to 0^+} \frac{\mathbf{v}^*[\frac{\mathbf{d}}{\mathbf{d\epsilon_i}}(\mathbf{G_i^*}) + \frac{\mathbf{d}}{\mathbf{d\epsilon_i}}\mathbf{G_i}]\mathbf{v}}{2}$$

$$\lim_{\epsilon_i \to 0^+} \frac{\mathbf{v}^*[\frac{\mathbf{d}}{\mathbf{d\epsilon_i}}(\mathbf{G_i^*}) + \frac{\mathbf{d}}{\mathbf{d\epsilon_i}}\mathbf{G_i}]\mathbf{v}}{2} = -j\mathbf{v}^*[\frac{\mathbf{d}}{\mathbf{d\Omega}}(\mathbf{G}(\mathbf{j\Omega_i})]\mathbf{v} = \mathbf{v}^*\mathbf{\Gamma_i}\mathbf{v}$$

So, we get $\mathbf{v}^*\mathbf{\Gamma_i}\mathbf{v} \geq \mathbf{0} \quad \forall \mathbf{v} \in \mathbb{C}^m, \forall i = 1, ..., n$

Next consider, $\mathbf{u} = \sum_{i=1}^{n} \mathbf{v_i} e^{(\epsilon_i + j\Omega_i)t}$, where $\mathbf{v}_i$ are arbitrary vectors in $\mathbb{C}^m$. Correspondingly $\mathbf{y} = \sum_{i=1}^{n} \mathbf{G}_i \mathbf{v_i} e^{(\epsilon_i + j\Omega_i)t}$. So, we observe that

$$\int_{-\infty}^{0} (\mathbf{u}^*\mathbf{y} + \mathbf{y}^*\mathbf{u})dt = \begin{bmatrix} \mathbf{v}_1^* & \mathbf{v}_2^* & ... & \mathbf{v}_n^* \end{bmatrix} \mathbf{T} \begin{bmatrix} \mathbf{v}_1^T & \mathbf{v}_2^T & ... & \mathbf{v}_n^T \end{bmatrix}^T$$

$$\text{where, } [\mathbf{T}]_{ik} := \lim_{\epsilon_i, \epsilon_k \to 0^+} \int_{-\infty}^{0} (\mathbf{G_i^*} + \mathbf{G_k}) e^{(\epsilon_k + \epsilon_i + j(\Omega_k - \Omega_i))t} dt$$

$$= \frac{(\mathbf{G_i^*} + \mathbf{G_k})}{j(\Omega_k - \Omega_i)} = \frac{(\mathbf{A_i^*} + \mathbf{A_k})}{j(\Omega_k - \Omega_i)}$$

Thus we see that **T** is in fact the Pick matrix (**P**) that we have defined in equation (B.4), we have

$$\begin{bmatrix} \mathbf{v}_1^* & \mathbf{v}_2^* & ... & \mathbf{v}_n^* \end{bmatrix} \mathbf{P} \begin{bmatrix} \mathbf{v}_1^T & \mathbf{v}_2^T & ... & \mathbf{v}_n^T \end{bmatrix}^T \geq \mathbf{0}$$

equality holds iff all $\mathbf{v}_i$'s are 0 vectors. Thus pick matrix **P** is positive definite.

B.3 Induction Based Solution

$$\mathbf{B}_i = \begin{bmatrix} \mathbf{I}_m & \mathbf{A}_i^T \end{bmatrix}^T$$

Base Step : n = 1. $\mathbb{D} = (\Omega_1, \mathbf{B}_1, \mathbf{\Gamma}_1)$

$$\mathbf{N}(s) = \mathbf{A}_1^* \mathbf{\Gamma_1^{-1}} \mathbf{A}_1 \quad \& \quad \mathbf{D}(s) = (\mathbf{s} - j\Omega_1)\mathbf{I}_m - \mathbf{\Gamma_1^{-1}} \mathbf{A}_1$$

These formulations are derived using the similar philosophy we have described in section IV of this paper. i.e Assume that

$$\mathbf{N}(s) = X \quad \& \quad \mathbf{D}(s) = (\mathbf{s} - j\Omega_1)\mathbf{I}_m - Y \quad \text{solve for X \& Y}$$

Correspondingly we define $\mathbf{G}$(s) := $\mathbf{N}$(s)$\mathbf{D}(\mathbf{s})^{-\mathbf{1}}$ .This $\mathbf{G}$(s) satisfies all 3 conditions of Equation (B.1). Thus, the base step is verified.

Inductive Step : We assume that our problem is solvable for $n-1$ points and prove that the problem is solvable for n points. First we modify our data-set for $n-1$ points. For that define,

$$\mathbf{H}(s) := \begin{bmatrix} (s - j\Omega_1)\mathbf{I}_m - \mathbf{\Gamma_1^{-1}}\mathbf{A}_1^* & -\mathbf{\Gamma_1^{-1}} \\ -\mathbf{A}_1\mathbf{\Gamma_1^{-1}}\mathbf{A}_1^* & (s - j\Omega_1)\mathbf{I}_m - \mathbf{A}_1\mathbf{\Gamma_1^{-1}} \end{bmatrix}$$

Now, we define

$$\hat{\mathbf{B}}_i := \mathbf{H}(s)\begin{bmatrix} \mathbf{I}_m & \mathbf{A}_i^T \end{bmatrix}^T$$

$$\hat{\mathbf{\Gamma}}_i := \mathbf{\Gamma}_{i+1} - \frac{(\mathbf{A}_{i+1}^* + \mathbf{A}_1)\mathbf{\Gamma}_1{}^{-1}(\mathbf{A}_1^* + \mathbf{A}_{i+1})}{(\Omega_{i+1} - \Omega_1)^2}$$

The modifications are done so that the Pick matrix of the new data set is the Schur complement of the Pick matrix of the old data set for the n points w.r.t. $\mathbf{\Gamma}_1$.

With this, now the solution for the original data-set of n points is as follows

$$\mathbf{N}(s) = [(s - j\Omega_1)\mathbf{I}_m - \mathbf{A_1^*}\mathbf{\Gamma_1^{-1}}]\hat{\mathbf{N}}(s) + [\mathbf{A}_1^*\mathbf{\Gamma}^{-1}\mathbf{A}_1]\hat{\mathbf{D}}(s)$$
$$\mathbf{D}(s) = [(\mathbf{s} - j\Omega_1)\mathbf{I}_m - \mathbf{\Gamma}_1^{-1}\mathbf{A}_1]\hat{\mathbf{D}}(s) - [\mathbf{\Gamma}^{-1}]\hat{\mathbf{N}}(s)$$

$$\mathbf{G}(s) := \mathbf{N}(s)\mathbf{D}(s)^{-1}$$

B.4 Implementation Results on MIMO

We implemented the above discussed SNIP based interpolation method for Skew-hermitian data on 4×4 MIMO, with size of data-set = 4. The skew-hermitian data set is generated from wireless MIMO rayleigh fading channel($\mathbf{H}$) by peroforming SVD as follows.

$$\mathbf{H} = \mathbf{U\Sigma V^*} \qquad \text{where, } \mathbf{U} \text{ is unitary}$$
$$\mathbf{H} = (\mathbf{U\Sigma U^*})(\mathbf{UV^*})$$

$(\mathbf{U\Sigma U^*})$ is hermitian , hence $-1j(\mathbf{U\Sigma U^*})$ is Skew-hermitian.

We took ITU Vehicular A profile as wireless channel model. Skew-hermitian matrices($\mathbf{U}$) are generated at certain frequencies (referred to as data-set & interpolation frequencies respectively) At the frequencies other than those in data-set we interpolate the skew-hermitian matrices using the above developed induction based solution. To measure the accuracy of interpolation method we plot the error between the interpolated skew-hermitian matrix($\hat{\mathbf{V}}(\mathbf{e}^{\mathbf{j}\omega})$) at a frequency ω and the skew-hermitian matrix that is realized from our channel matrix($\mathbf{H}$) at the same frequency ω The plots shown below is error plot with Frobenius norm as performance metric.

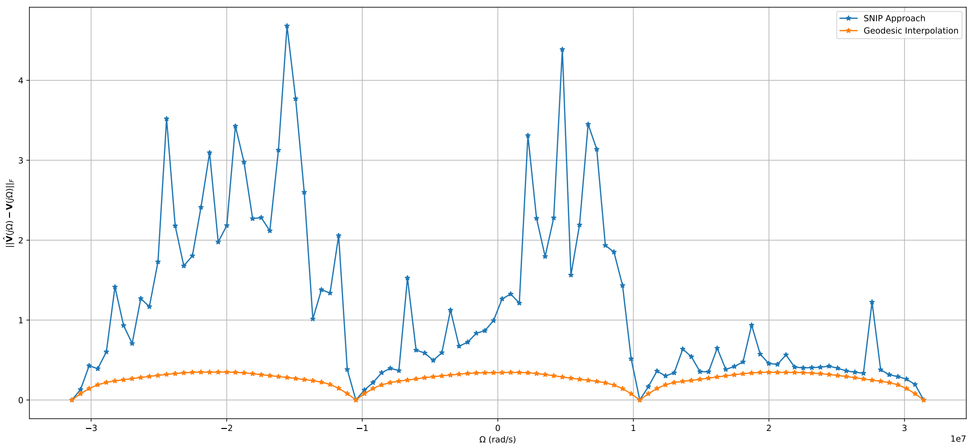


Figure B.1: MIMO 4 × 4 with Frobenius norm as error measure

It is evident from the above plots that the output of our filter exactly matches with the skew-hermitian matrix in data-set at the frequencies in data-set. At the non interpolating points our approach it worse than the geodesic interpolation but still we achieved time domain realizable rational filter by design method followed above.We can extend this skew-hermitian interpolation method to discrete-time using Bilinear transformation and thus we get an equivalent LCCDE that leads to simpler implementation in time domain.

NOTE: There is a lot of scope for future work in this direction, to design a filter with further optimizations in the design.

References

[1] D.J. Love and R.W. Heath. Multimode precoding for mimo wireless systems. 53(10):3674–3687, 2005.

[2] H. Karaa, R. S. Adve, and A. J. Tenenbaum. Linear precoding for multiuser mimo-ofdm systems. In *2007 IEEE International Conference on Communications*, pages 2797–2802, 2007.

[3] David J. Love, Robert W. Heath, Vincent K. N. Lau, David Gesbert, Bhaskar D. Rao, and Matthew Andrews. An overview of limited feedback in wireless communication systems. 26(8):1341–1365, 2008.

[4] D. J. Love and R. W. Heath. Limited feedback unitary precoding for spatial multiplexing systems. 51(8):2967–2976, 2005.

[5] P. Rapisarda and J. C. Willems. The Subspace Nevalinna Interpolation Problem and the most powerful unfalsified model. In *36th IEEE Conference on Decision and Control*, volume 3, pages 2029–2033, 1997.

[6] M. Saito D.C. Youla. Interpolation with positive real functions. *Journal of the Franklin Institute*, 284, Issue 2:77–108, 1967.

[7] K. Appaiah and D. Pal. All-pass filter design using Blaschke interpolation. 27:226–230, 2020.

[8] Vladimir Bolotnikov. Boundary Interpolation by Finite Blaschke Products. In *Complex Analysis and Dynamical Systems*, pages 39–65. Springer, 2018.

[9] Nadia Khaled, Bishwarup Mondal, Geert Leus, Robert W. Heath, and Frederik Petre. Interpolation-based multi-mode precoding for mimo-ofdm systems with limited feedback. 6(3):1003–1013, 2007.

[10] N. Khaled, B. Mondal, R.W. Heath, G. Leus, and F. Petre. Quantized multi-mode precoding for spatial multiplexing mimo-ofdm system. In *VTC-2005-Fall. 2005 IEEE 62nd Vehicular Technology Conference, 2005.*, volume 2, pages 867–871, 2005.

[11] D.J. Love, R.W. Heath, W. Santipach, and M.L. Honig. What is the value of limited feedback for MIMO channels? 42(10):54–59, 2004.

[12] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[13] S. Fan and J. M. Kahn. Principal modes in multimode waveguides,. *Optics Letters*, 30(2):135–137, 2005.

[14] J. C. Willems. Dissipative Dynamical Systems Part I: General Theory. *Archive for Rational Mechanics and Analysis*, 45(5):321–351, January 1972.

[15] Osamu Kaneko and Takao Fujii. Discrete-time average positivity and spectral factorization in a behavioral framework. *Systems & Control Letters*, 39(1):31–44, 2000.

[16] H.L. Trentelman and J.C. Willems. Every storage function is a state function. *Systems & Control Letters*, 32(5):249–259, 1997. System and Control Theory in the Behavioral Framework.

[17] Jean H. Gallier. *Notes on the Schur Complement*. December 2010.

[18] Jihoon Choi, Bishwarup Mondal, and Robert W. Heath. Interpolation based unitary precoding for spatial multiplexing mimo-ofdm with limited feedback. 54(12):4730–4740, 2006.

[19] ITU-R Recommendation. Guidelines for evaluation of radio transmission technologies for imt-2000. *Rec. ITU-R M. 1225*, 1997.

[20] S. Nijhawan, A. Gupta, K. Appaiah, R. Vaze, and N. Karamchandani. Flag Manifold-Based Precoder Interpolation Techniques for MIMO-OFDM Systems. 69(7):4347–4359, 2021.

[21] Renaud-Alexandre Pitaval, Ashvin Srinivasan, and Olav Tirkkonen. Codebooks in flag manifolds for limited feedback mimo precoding. In *SCC 2013; 9th International ITG Conference on Systems, Communication and Coding*, pages 1–5. VDE, 2013.

[22] Renaud-Alexandre Pitaval. Coding on flag manifolds for limited feedback mimo systems. 2013.

[23] Google Inc. Google colaboratory.

[24] Tarkesh Pande, David J. Love, and James V. Krogmeier. Reduced feedback mimo-ofdm precoding and antenna selection. *IEEE Transactions on Signal Processing*, 55(5):2284–2293, 2007.

[25] Wei Dai, Youjian Liu, Brian Rider, and Vincent K. N. Lau. On the information rate of mimo systems with finite rate channel state feedback using beamforming and power on/off strategy. *IEEE Transactions on Information Theory*, 55(11):5032–5047, 2009.

[26] P. Vaidyanathan and S. Mitra. A general family of multivariable digital lattice filters. *IEEE Transactions on Circuits and Systems*, 32(12):1234–1245, 1985.