國立臺灣大學電機資訊學院電信工程學研究所

博士論文

Graduate Institute of Communication Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Doctoral Dissertation

拉瑪努江和於信號處理及週期偵測之應用

Ramanujan’s Sum and Its Application to Signal Processing and Period Estimation

張國韋

Kuo-Wei Chang

指導教授：貝蘇章博士

Advisor: Soo-Chang Pei, Ph.D.

中華民國 107 年 5 月

May, 2018

doi:10.6342/NTU201800930

doi:10.6342/NTU201800930

誌謝

感謝眞主，感謝我的父母，感謝貝老師，感謝中華電信的主管和同事們，感謝 530 實驗室的學長姐與學弟妹。這篇論文獻給所有幫助我的人。

doi:10.6342/NTU201800930

doi:10.6342/NTU201800930

Acknowledgements

Thanks everyone who supports me.

doi:10.6342/NTU201800930

doi:10.6342/NTU201800930

摘要

本論文利用拉瑪努江和 (Ramanujan’s sum) 的特性，舉出兩種信號處理上的應用。其一爲建構零自相關的整數序列 (Integer zero autocorrleation sequence)，這在通訊系統上有廣泛的應用。他的原理是利用傅立葉轉換，把建構零自相關序列的數論問題，轉換成建構一個固定振幅信號 (constant amplitude) 的問題，這裡巧妙用上拉瑪努江和整數的特性。另一個應用是一維與二維的信號週期偵測。利用拉瑪努江和週期以及整數的特性，週期信號可以被分解成一個個因數的週期，往後便可以針對不同的週期進行不同的處理，如同以前的濾波器組 (filter bank)。最後我們針對拉瑪努江和只能找出因數週期的缺點，採用全相位快速傅立葉 (all phase FFT) 演算法來改進。全相位快速傅立葉演算法爲利用相位的差，求出非整數的頻率，改進了以往頻率只能在整數上的缺點，可以幫助我們進行週期偵測。我們也順帶提到全相位快速傅立葉演算法的一些其他應用，包含啁聲信號 (chirp signal) 的頻率追蹤、快速二維頻率偵測等等。

doi:10.6342/NTU201800930

doi:10.6342/NTU201800930

Abstract

This thesis presents two applications of Ramanujan's sum in the domain of signal processing. The first one is integer zero autocorrelation sequence construction, which is useful in modern communication system. The concept is to transform this number theoretic problem into a constant amplitude signal construction problem, by Fourier transform and the integer property of Ramanujan's sum. The second application is 1D and 2D period estimation. The periodic signal is separated into sub-period signals, just like filter bank. Each sub-period is a factor of the length. Finally we use all phase FFT to enhance the period estimation. All phase FFT use phase information to estimate frequency. The advantage is the frequency is non integer. Finally we propose some other applications of all phase FFT, such as chirp signal pitch tracking and fast 2D frequency estimation.

doi:10.6342/NTU201800930

doi:10.6342/NTU201800930

Contents

doi:10.6342/NTU201800930

doi:10.6342/NTU201800930

List of Figures

doi:10.6342/NTU201800930

doi:10.6342/NTU201800930

List of Tables

doi:10.6342/NTU201800930

doi:10.6342/NTU201800930

Chapter 1

Introduction

1.1 Definition and Properties of Ramanujan's Sum

A special sum of the roots of unity called Ramanujan's sum[1]

$$c_q(n) = \sum_{a=1,(a,q)=1}^{q} e^{2\pi i \frac{a}{q} n} \tag{1.1}$$

where $(a, q) = 1$ means that a only takes on values coprime to q. For example,

$$c_3(n) = [2, -1, -1, 2, -1, -1, 2, -1, ...] \tag{1.2}$$

$$c_6(n) = [2, 1, -1, -2, -1, 1, 2, 1, -1, ...] \tag{1.3}$$

$$c_7(n) = [6, -1, -1, -1, -1, -1, -1, 6, -1, -1, ...] \tag{1.4}$$

The sums have many good properties. First of all, they are all integer. This is amazing because the roots of unity are complex number. The summation over complex is in general also complex, but the Ramanujan's Sum is real and more surprisingly, integer. Secondly, the sums are periodic,

$$c_q(n+q) = c_q(n) \tag{1.5}$$

Additionally, since $c_q(n)$ is real, we have

$$c_q(n) = c_q^*(n) = \sum_{a=1,(a,q)=1}^{q} e^{-2\pi i \frac{a}{q} n} = c_q(-n) \tag{1.6}$$

where x^* is the complex conjugate of x. In other words, c_q is an even function.

Moreover, when signal length N is a multiple of q, Ramanujan's sum can be viewed as discrete Fourier transform (DFT) of a binary sequence. For instance, let $N = 6$

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \xrightarrow{\text{DFT}} \begin{bmatrix} 2 \\ 1 \\ -1 \\ -2 \\ -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \xrightarrow{\text{DFT}} \begin{bmatrix} 2 \\ -1 \\ -1 \\ 2 \\ -1 \\ -1 \end{bmatrix} \tag{1.7}$$

As we can see, $c_6(n)$ and $c_3(n)$, $n = 0, 1, ..., 5$ are constructed by 6-point DFT. Furthermore, since the binary sequences are orthogonal, the Ramanujan's sum c_6 and c_3 are also orthogonal. In general, if q_1 and q_2 are two different factors of N, then $c_{q_1}(n)$ and $c_{q_2}(n)$, $n = 0, 1, ..., N-1$ are orthogonal.

Based on those good properies, the Ramanujan's sum can be further used in analyzing long period time sequences such as stock price, DNA and protein coding[2, 3, 4]. In this thesis, we will introduce two applications in signal processing. The first is to construct integer zero autocorrelation sequences, and the second is period estimation. We define some useful functions and notations in the next section.

1.2 Preliminary

Unless otherwise specified, any signal or sequence x in this thesis is periodic with length N, and the index is n with module N, $n = 0, 1, 2, ..., N-1$. In other words, $x(-n) = x(N-n)$ and $x(N+n) = x(n)$. The discrete Fourier transform (DFT) of x is defined as

$$\hat{x}(k) = F\{x\} = \sum_{n=0}^{N-1} x(n) W_N^{nk} \tag{1.8}$$

where $W_N = e^{-2\pi i/N}$, $i = \sqrt{-1}$. If there is no ambiguity, W is used instead of W_N for convenience.

The delta function is

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases} \tag{1.9}$$

The autocorrelation of signal is defined as

$$R_{xx}(k) = \sum_{n=0}^{N-1} x^*(n-k)x(n) \tag{1.10}$$

where x^* is the complex conjugate of x. And a signal x is called zero-autocorrelation (ZAC) if $R_{xx}(k) = C\delta(k)$ for some nonzero constant C.

Some number theoretic definitions are given as follows. Let a and b be two integers, $a|b$ means a is a factor of b. The greatest common divisor (gcd) d them is denoted as $d = \gcd(a, b)$ or simply $d = (a, b)$. They are called coprime if $(a, b) = 1$. The least common multiple of them is denoted as $\operatorname{lcm}(a, b)$. The number in the form of $a + bi$ is called Gaussian integer. For a positive integer n, two integers a and b are said to be congruent modulo n, written:

$$a \equiv b \pmod n \tag{1.11}$$

if their difference $a - b$ is a multiple of n.

We define the set $I_k = \{i | 1 \leq i \leq k, \gcd(i,k) = 1\}$ to indicate all number between 1 and k which is relatively prime to k.

2D signal is represented in capital letter like X, the default size is $N \times N$ and the index is given by $X(n_1, n_2)$. Two dimensional discrete Fourier transform (2DDFT) is defined as

$$\hat{X}(k_1, k_2) = \sum_{n_1} \sum_{n_2} X(n_1, n_2) W^{n_1 k_1 + n_2 k_2} \tag{1.12}$$

doi:10.6342/NTU201800930

Chapter 2

First Application: Integer and Gaussian Integer Zero Autocorrelation Sequences

2.1 Motivation

Zero autocorrelation(ZAC) sequences have been extensively used in communication engineering, such as synchronization, CDMA [5, 6] and OFDM[7] system. They have also been applied to cryptography for constructing pseudo random sequences. In this work a special kind of ZAC sequences is considered, that is, integer ZAC (IZAC), because integer has the following advantages comparing to complex floating point number.

1) Integer requires less memory, for both saving and sending.

2) Arithmetic operations can be done faster and error-free.

3) The system can be implemented on hardware easily.

doi:10.6342/NTU201800930

There are some trivial IZAC . The most obvious one is

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

The second one, although less known, is still in simple form

$$\begin{pmatrix} N-2 \\ -2 \\ -2 \\ \vdots \\ -2 \\ -2 \end{pmatrix}$$

where N is the signal length. For example, $N = 5$ and $N = 6$

$$\begin{pmatrix} 3 \\ -2 \\ -2 \\ -2 \\ -2 \end{pmatrix} \text{ and } \begin{pmatrix} 4 \\ -2 \\ -2 \\ -2 \\ -2 \\ -2 \end{pmatrix}$$

When N is a composite number, constructing integer-valued ZAC from its factor by zero-padding is not difficult to think of. Let say $N = 15 = 3 \times 5$, by the examples given above, we can organize signals like

$$(3, 0, 0, -2, 0, 0, -2, 0, 0, -2, 0, 0, -2, 0, 0)$$

or

$$(1,0,0,0,0,-2,0,0,0,0,-2,0,0,0,0)$$

One natural question is, are there any non-trivial integer-valued ZAC? The answer is yes, such as

$$\begin{pmatrix} 2 \\ -1 \\ -1 \\ 2 \\ -1 \\ 5 \end{pmatrix}$$

Similarly, non-trivial Gaussian integer ZAC can also be found. In the following, we will introduce how to construct those sequences.

2.2 Related Works

There are two types of construction method for IZAC. The first type is solving equation directly. The other is by generating a constant amplitude (CA) sequence in frequency domain such that its discrete Fourier transform is integer. A constant amplitude sequence $x(n)$ is defined as

$$|x(n)| = C \tag{2.1}$$

for some constant C. The reason behind the second method is by the following theorem.

Theorem 2.2.1. *A sequence x is CA if and only if its DFT $\hat{x}$ is ZAC. Similarly, a sequence x is ZAC if and only if its DFT $\hat{x}$ is CA.*

Proof. See [8] □

doi:10.6342/NTU201800930

For convenience, we call the first method *time domain method* and the other *frequency domain method*. As we will see, time domain methods usually have to deal with non-linear equations, while frequency domain methods have to guarantee its DFT is integer. Both of them have some issues to conquer, but frequency domain methods are more systematic and easier to solve.

2.2.1 Time domain method I: Making extra constraints

Consider $N = 5$, the construction of IZAC is solving a five-variable equation.

$$x(0)x(1) + x(1)x(2) + x(2)x(3) + x(3)x(4) + x(4)x(0) = 0 \tag{2.2}$$
$$x(0)x(2) + x(1)x(3) + x(2)x(4) + x(3)x(0) + x(4)x(1) = 0 \tag{2.3}$$
$$x(0)x(3) + x(1)x(4) + x(2)x(0) + x(3)x(1) + x(4)x(2) = 0 \tag{2.4}$$
$$x(0)x(4) + x(1)x(0) + x(2)x(1) + x(3)x(2) + x(4)x(3) = 0 \tag{2.5}$$

where $x(n) \in \mathbb{Z}$. We can notice that eq. (2.4) is equivalent to eq. (2.3) and eq. (2.5) is equivalent to eq. (2.2). Thus, there are actually only two non-linear equations here. To solve this directly seems impossible, an alternative way is making more constraints. In particular, let $x(0) = x(1) = 1$, eq. (2.2) and eq. (2.3) become

$$1 + x(2) + x(2)x(3) + x(3)x(4) + x(4) = 0 \tag{2.6}$$
$$x(2) + x(3) + x(2)x(4) + x(3) + x(4) = 0 \tag{2.7}$$

Let $x(3) = a$ we can find

$$x(2) + x(4) = \frac{-1}{a+1} \tag{2.8}$$
$$x(2)x(4) = -2a + \frac{1}{a+1} \tag{2.9}$$

doi:10.6342/NTU201800930

$$\begin{aligned}\Rightarrow (x(2)-x(4))^2 &= \frac{1}{(a+1)^2} + 8a - \frac{4}{a+1} \\ &= \frac{8a^3+16a^2+4a-3}{(a+1)^2}\end{aligned} \tag{2.10}$$

Since $x(2)$, $x(3) = a$, and $x(4)$ are all rational (which can be modified into integer easily by multiplying the denominators), eq. (2.10) means there are two rational numbers y and a, where $y = (x(2)-x(4))(a+1)$, such that

$$y^2 = 8a^3 + 16a^2 + 4a - 3$$

This is exactly a well-known rational Elliptic Curves problem[9, 10]. Ellipitic Curves problem has been studied a lot for the past few decades, and its main application is cryptography[11]. In short, if there are two non-trivial rational points (y, a) on the Elliptic curve, then there are infinite rational points on it [9]. In this case, since we already know the answer $[-3, 2, 2, 2, 2]$ is integer ZAC, so when $x(0) = x(1) = 1$, $x(3) = a$ has two choices $1, -3/2$. By finding the next point we can get $a = -7/8$ and the new integer ZAC is

$$[8, 8, -12, -7, -52]$$

and the next is

$$[22, 22, 142, 462, -143]$$

and so on.

In general, making extra constraints can reduce the problem, and we have found a interesting connection to elliptic curve. However, the method is ad hoc and difficult to find a solution for arbitrary N.

2.2.2 Time domain method II:Linear equation method

Another way to generate ZAC is given in [12]. This method only works for Gaussian integer and signal length $N = p^k$ where p is a prime number.

It also reduces the problem by making more constraints. In the following we summarize this method by setting $N = 9$.

The first constraint is called grouping. The index $n = 0, 1, 2, ..., N-1$ is divided into several groups, and $x(n)$ is the same if n is in the same group. For $N = 9$, we have 3 groups

$$x(0) \tag{2.11}$$

$$x(3) = x(6) \tag{2.12}$$

$$x(1) = x(2) = x(4) = x(5) = x(7) = x(8) \tag{2.13}$$

One can notice that the index n is divided by $\gcd(n, N)$. In other words, if $\gcd(n_1, N) = \gcd(n_2, N)$, then n_1 and n_2 are in the same group. The varibles in this problem is reduced from 9 to 3. Moreover, the number of equations for ZAC is reduced into 2.

$$x^*[0]x(1) + x^*(1)x(0) + 3|x(1)|^2 + 2(x^*(3)x(1) + x^*(1)x(3)) = 0 \tag{2.14}$$

$$x^*[0]x(3) + |x(3)|^2 + x^*(3)x(0) + 6|x(1)|^2 = 0 \tag{2.15}$$

Since x is Gaussian integer, let $x(n) = x_n + iy_n$ and rewrite the above

$$2(x_0x_1 + y_0y_1) + 3(x_1^2 + y_1^2) + 4(x_1x_3 + y_1y_3) = 0 \tag{2.16}$$

$$2(x_0x_3 + y_0y_3) + (x_3^2 + y_3^2) + 6(x_1^2 + y_1^2) = 0 \tag{2.17}$$

Next, Equation (2.17) subtracts Equation (2.16).

$$(x_3 - x_1)(2x_0 - 3x_1 + x_3) + (y_3 - y_1)(2y_0 - 3y_1 + y_3) = 0 \tag{2.18}$$

Here comes the second constraint. Although Equation (2.18) is still a non-linear equation like Equation (2.2), we can use two linear equations

doi:10.6342/NTU201800930

to approximate it.

$$x_3 - x_1 = -y_3 - y_1 \tag{2.19}$$

$$2x_0 - 3x_1 + x_3 = 2y_0 - 3y_1 + y_3 \tag{2.20}$$

Note that the equations above can only represent part of the solutions in Equation (2.18). For instance,

$$x_3 - x_1 = 6 \tag{2.21}$$

$$2x_0 - 3x_1 + x_3 = 2 \tag{2.22}$$

$$y_3 - y_1 = -4 \tag{2.23}$$

$$2y_0 - 3y_1 + y_3 = 3 \tag{2.24}$$

may be a solution of Equation (2.18), but it is not a solution of Equations (2.19) and (2.20). To summarize, time domain methods face the non-linear equations directly. It is a hard problem without making extra constraints. In the following, we will introduce some frequency domain methods. By the Theorem 2.2.1, this number theoretic problem can be solved easily by Fourier transform.

2.2.3 Frequency domain method I:The Geometric sequence method

The method in this section has been submitted to Eusipco 2018.

For any length N, there are two trivial ZAC integer sequences[13]

$$x(n) = [1, 0, 0, 0, ..., 0] \tag{2.25}$$

$$x(n) = [N-2, -2, -2, -2, ..., -2] \tag{2.26}$$

Moreover, if N is even, the following sequence is also a ZAC integer sequence.

$$x(n) = \begin{cases} \frac{N-2}{2} & n = 0 \\ (-1)^{n-1} & n \neq 0 \end{cases} \tag{2.27}$$

Observe Equations (2.25) to (2.27), we can find two interest things. The first is that only the element $x(0)$ is different from the others. The second is that each $x(n), n \neq 0$ forms a geometric series. In Equation (2.25), $[0, 0, ..., 0]$ can be viewed as geometric series with any ratio r. In Equation (2.26), the ratio $r = 1$ while in Equation (2.27), the ratio $r = -1$.

Another example can be found in $N = 4$. The closed form solution for perfect integer sequence is

$$\left[-b, a, b, \frac{b^2}{a}\right] \tag{2.28}$$

where $a, b, \frac{b^2}{a}$ are integers. This is solved by brute force. The last three terms $\left[a, b, \frac{b^2}{a}\right]$ can also be viewed as geometric series, with ratio $r = \frac{b}{a}$.

From the observation above, a natural assumption is that any geometric series can be used to generate perfect integer sequence. With some calculation the assumption can be proved. The result is given as follows. Let r be an integer. A closed form perfect integer sequences is given as

$$x(n) = \begin{cases} -\sum\limits_{k=2}^{N-1} r^k & n = 0 \\ (1+r)r^n & n \neq 0 \end{cases} \tag{2.29}$$

Note that $x(n)$ is a geometric series from $n = 1$ to $N - 1$. Since $r = 0$ or $r = \pm 1$ gives only trivial result, from now on only the cases $r \neq 0, \pm 1$ are discussed. For a concrete example, let $N = 5$ and $r = 2$

$$x(n) = [-(2^2 + 2^3 + 2^4), 3 \cdot 2^1, 3 \cdot 2^2, 3 \cdot 2^3, 3 \cdot 2^4] \tag{2.30}$$

$$= [-28, 6, 12, 24, 48] \tag{2.31}$$

For another example, let $N = 6$ and $r = -3$

$$x(n) = \begin{bmatrix} -\sum_{k=2}^{5}(-3)^k \\ (-2)\cdot(-3)^1 \\ (-2)\cdot(-3)^2 \\ (-2)\cdot(-3)^3 \\ (-2)\cdot(-3)^4 \\ (-2)\cdot(-3)^5 \end{bmatrix} \tag{2.32}$$

$$= [180, 6, -18, 54, -162, 486] \tag{2.33}$$

The two examples above illustrate that both prime length $N = 5$ and composite length $N = 6$ can be used. In fact, there is no limitation on signal length. To prove Equation (2.29), by Theorem 2.2.1 we should check that $\hat{x}$ is CA. By definition,

$$\begin{aligned}
\hat{x}(m) &= \sum_{n=0}^{N-1} x(n) W_N^{nm} && (2.34)\\
&= -\sum_{k=2}^{N-1} r^k + \sum_{n=1}^{N-1} (1+r) r^n W_N^{nm} && (2.35)\\
&= \left(-\sum_{k=2}^{N-1} r^k\right) - (1+r) + (1+r) \\
&+ (1+r)\sum_{n=1}^{N-1} r^n W_N^{nm} && (2.36)\\
&= -\sum_{k=0}^{N-1} r^k + (1+r)\sum_{n=0}^{N-1} r^n W_N^{nm} && (2.37)\\
&= -\frac{1-r^N}{1-r} + (1+r)\frac{1-(rW_N^m)^N}{1-rW_N^m} && (2.38)\\
&= -\frac{1-r^N}{1-r} + (1+r)\frac{1-r^N}{1-rW_N^m} && (2.39)\\
&= (1+r)(1-r^N)\left(\frac{1}{1-rW_N^m} - \frac{1}{1-r^2}\right) && (2.40)
\end{aligned}$$

To prove $\hat{x}$ is CA, the following lemma is needed.

Lemma 2.2.2. *For any complex number $z = re^{i\theta}$, where $r = |z| \neq 1$,*

$$\left|\frac{1}{1-z} - \frac{1}{1-r^2}\right| = \left|\frac{r}{1-r^2}\right| \tag{2.41}$$

In other words, the magnitude of $\frac{1}{1-z} - \frac{1}{1-r^2}$ only depends on r and is independent from θ.

Proof.

$$\frac{1}{1-re^{i\theta}} - \frac{1}{1-r^2} = \frac{1}{1-r\cos(\theta)+ir\sin(\theta)} - \frac{1}{1-r^2} \tag{2.42}$$

$$=\frac{1-r\cos(\theta)-ir\sin(\theta)}{\left(1-r\cos(\theta)\right)^2+r^2\sin^2(\theta)} - \frac{1}{1-r^2} \tag{2.43}$$

$$=\left(\frac{1-r\cos(\theta)}{1-2r\cos(\theta)+r^2} - \frac{1}{1-r^2}\right) - i\frac{r\sin(\theta)}{1-2r\cos(\theta)+r^2} \tag{2.44}$$

Thus, let $Q = 1-2r\cos(\theta)+r^2$

$$\left|\frac{1}{1-re^{i\theta}} - \frac{1}{1-r^2}\right|^2 \tag{2.45}$$

$$=\left(\frac{1-r\cos(\theta)}{Q} - \frac{1}{1-r^2}\right)^2 + \left(\frac{r\sin(\theta)}{Q}\right)^2 \tag{2.46}$$

$$=\frac{(1-r\cos(\theta))^2}{Q^2} - 2\frac{1-r\cos(\theta)}{Q(1-r^2)} + \frac{1}{(1-r^2)^2} + \frac{r^2\sin^2(\theta)}{Q^2} \tag{2.47}$$

Note that $(1-r\cos(\theta))^2 + r^2\sin^2(\theta) = Q$

$$\left|\frac{1}{1-re^{i\theta}} - \frac{1}{1-r^2}\right|^2 \tag{2.48}$$

$$=\frac{Q}{Q^2} - 2\frac{1-r\cos(\theta)}{Q(1-r^2)} + \frac{1}{(1-r^2)^2} \tag{2.49}$$

$$=\frac{1-r^2}{Q(1-r^2)} - 2\frac{1-r\cos(\theta)}{Q(1-r^2)} + \frac{1}{(1-r^2)^2} \tag{2.50}$$

$$=\frac{-1+2r\cos(\theta)-r^2}{Q(1-r^2)} + \frac{1}{(1-r^2)^2} \tag{2.51}$$

$$=\frac{-Q}{Q(1-r^2)} + \frac{1}{(1-r^2)^2} = \frac{1-(1-r^2)}{(1-r^2)^2} = \left(\frac{r}{(1-r^2)}\right)^2 \tag{2.52}$$

which completes the proof. □

We can now prove $x(n)$ in Equation (2.29) is ZAC. By (2.40) and Lemma 2.2.2,

$$|\hat{x}(m)| = \left|(1+r)(1-r^N)\left(\frac{1}{1-rW_N^m} - \frac{1}{1-r^2}\right)\right| \tag{2.53}$$

$$= |(1+r)(1-r^N)| \cdot \left|\frac{1}{1-rW_N^m} - \frac{1}{1-r^2}\right| \tag{2.54}$$

$$= \left|\frac{r(1+r)(1-r^N)}{1-r^2}\right| = \left|\frac{r(1-r^N)}{1-r}\right| \tag{2.55}$$

in other words, $|\hat{x}|$ is constant when r is fixed. Thus, $\hat{x}$ is CA and by Theorem 2.2.1, x is ZAC.

This method looks like a time domain method, because the power series is used in time domain. However, the ZAC property is proved by Theorem 2.2.1, so it is actually a frequency domain method. This construction of ZAC works for Gaussian integer as well, by set $r = a + bi$. The main drawback of this method is that the dynamic range is too large. If $r = 2$ and $N = 100$, the max value of the ZAC sequence will be about $2^{100} \approx 10^{30}$, although it is still integer.

2.2.4 Frequency domain method I:Legendre sequence method

The methods in this section have been published in [14].

Using Legendre sequence and Gauss sum when N is prime

Recall that when N is prime, Legendre symbol is defined as

$$\left(\frac{n}{N}\right) = \begin{cases} 1, & \text{if } n \equiv x^2 (\bmod N), \text{for some } x \\ 0, & n \equiv 0 (\bmod N) \\ -1, & otherwise. \end{cases}$$

doi:10.6342/NTU201800930

And the Gauss sum is defined as

$$G(k) = \sum_{n=0}^{N-1} \left(\frac{n}{N}\right) e^{-2\pi ikn/N}$$

A well known result[15] is that

$$G(k) = \begin{cases} \left(\frac{k}{N}\right)\sqrt{N}, & N \equiv 1 (\bmod 4) \\ -\left(\frac{k}{N}\right) i\sqrt{N}, & N \equiv 3 (\bmod 4) \end{cases}$$

In other words, the Fourier transform of Legendre Sequences is almost CA, with the only exception on $k = 0$, the first point. The amplitude is $\sqrt{N}$ thus our goal is to find a Gaussian integer a and some integers b, and c such that

$$|a|^2 = b^2 + Nc^2 \tag{2.56}$$

Then a sequence that

$$f(n) = \begin{cases} a, & n = 0 \\ \left(\frac{n}{N}\right)\sqrt{N}c + bi, & n \neq 0 \end{cases} \tag{2.57}$$

is CA, and the DFT of $f(n)$ is ZAC in Gaussian integer. Before we prove this let's see some tiny examples of $N = 3$ and $N = 5$

Example 2.2.1. *For $N = 3$, the most trivial value we can choose is $a = 2, b = 1$,and $c = 1$. Thus*

$$f(n) = \left[2, 1i + \sqrt{3}, 1i - \sqrt{3}\right\}$$
$$\hat{f}(k) = F\left\{f(n)\right\} = \{2 + 2i, 2 - 4i, 2 + 2i]$$

As we can see, $f(n)$ is CA and $\hat{f}(k)$ is with Gaussian integer value. This example shows us a does not necessary to be complex.

Example 2.2.2. *N=3. Let's try a non-trivial value c=2,b=5,a=6+1i*

$$f(n) = \Big[6+1i, 5i+2\sqrt{3}, 5i-2\sqrt{3}\Big\}$$
$$\hat{f}(k) = \{6+11i, 6-10i, 6+2i]$$

From this example we can get a very different result from the previous one. Note that a has three alternatives $6-1i$, $1+6i$, and $1-6i$. They all give us different ZAC sequences.

Example 2.2.3. $N = 5, c = 1, b = 0, a = 2+1i$

$$f(n) = \Big[2+1i, \sqrt{5}, -\sqrt{5}, -\sqrt{5}, \sqrt{5}\Big\}$$
$$\hat{f}(k) = \{2+1i, 7+1i, -3+1i, -3+1i, 7+1i]$$

There are various ways to construct infinite triples (a, b, c) satisfied (2.56). Let $a = a_r + ia_i$, where a_r, a_i are integers. If we choose $c = 0$, clearly the (2.56) will reduce to $a_r^2 + a_i^2 = b^2$, and it has infinite solutions. If N is odd, we can choose any odd c and any even a_r. Now (2.56) becomes

$$a_r^2 + a_i^2 = b^2 + Nc^2$$
$$\Rightarrow a_i^2 - b^2 = Nc^2 - a_r^2$$

Since N and c are odd and a_r is even, $Nc^2 - a_r^2$ must be odd. Let $Nc^2 - a_r^2 = 2k+1$, then we can use $a_i = k+1$ and $b = k$. Similarly, for any even c and odd a_r, we can find proper a_i and b by the same trick.

We complete this subsection by proving the claim that $\hat{f}(k)$ is ZAC with Gaussian integer value.

Proof. Since $f(n)$ is CA by definition, thus $\hat{f}(k)$ is ZAC by Theorem 2.2.1. To prove they are all Gaussian integer, we actually give $\hat{f}(k)$ a closed form. If $N \equiv 1 (\bmod 4)$

$$\hat{f}(k) = \begin{cases} a + (N-1)bi, & k = 0 \\ \\ a + \left(\frac{k}{N}\right) Nc - bi, & k \neq 0 \end{cases}$$

If $N \equiv 3 (\bmod 4)$

$$\hat{f}(k) = \begin{cases} a + (N-1)bi, & k = 0 \\ \\ a - \left(\frac{k}{N}\right) Nci - bi, & k \neq 0 \end{cases}$$

Either case the $\hat{f}(k)$ is in Gaussian integer value since a is Gaussian integer and b, c are integers. □

Using GLS when N is $4k+1$ prime

When N is $4k+1$ prime, there is another way to increase the degree of freedom by using Generalized Legendre Sequences (GLS)[16, 17, 18]. GLS are originally applied to construct the eigenvectors of discrete Fourier transform (DFT) and generate a complete N-dimensional orthogonal basis[18], because they have the property that their Fourier transform is their conjugate multiply a constant whose absolute value is $\sqrt{N}$. All the values except the first one of GLS lies on unit circle, and when $N = 4k+1$ we can choose the ones which only contains $[1, i, -1, -i]$, in order to bound the sequences in gaussian integer. The detail steps is described as follows:

1)Choose a GLS x where the first value is 0 and the rest only contain $[1, i, -1, -i]$.

2)Take DFT of x. Now we have a sequence y that the first value is still 0, but the absolute value of rest all equal to $\sqrt{N}$.

3)The first value of y can then be chosen by a Gaussian integer $a+bi$ satisfying $a^2 + b^2 = N$, which always exists by Fermat's theorem. Now the sequence is constant amplitude.

4) The DFT of y is a ZAC with Gaussian integer.

It is easy to prove this sequence is really in Gaussian integer. In fact, the sequence is actually

$$F\left\{F\left\{x(n)\right\}+(a+bi)\delta(n)\right\}=Nx(N-n)+a+bi$$

where N, a, b, and all value in x are with Gaussian integer, so will be the sequence.

As a concrete example, assume $N=13$ and we choose a GLS x

$$x=[0,1,-i,1,-1,-i,-i,i,i,1,-1,i,-1]$$

And we choose $a+bi=3+2i$ since $3^2+2^2=13$. Thus the ZAC sequence is

$$\begin{aligned}&Nx(N-n)+a+bi=\\&13\left[0,-1,i,-1,1,i,i,-i,-i,-1,1,-i,1\right]+3+2i\\&=[3+2i,-10+2i,3+15i,-10+2i,\\&16+2i,3+15i,3+15i,3-11i,3-11i,-10+2i,\\&16+2i,3-11i,16+2i]\end{aligned}$$

This idea of this method can be extended by multiplying an integer or a Gaussian integer in step 2, and then we can have more choices in step 3. Let the length $N=a^2+b^2$, then we can find an integer or a Gaussian integer s, satisfying $|s|=\sqrt{c^2+d^2}$ with some integer c and d, and multiply this number to the GLS. The choices in step 3 can be doubled because

$$(a^2+b^2)(c^2+d^2)=(ad+bc)^2+(ac-bd)^2=(ad-bc)^2+(ac+bd)^2$$

doi:10.6342/NTU201800930

Thus by multiply different $c + di$ we can get different ZAC sequences. This gives us infinite choices.

For instance, if $N = 13$, originally we can only choose

$$a + bi = \{3 + 2i, 3 - 2i, 2 + 3i, 2 - 3i\}$$

When we multiply the GLS by $s = 5 = \sqrt{3^2 + 4^2}$, then in step 3 we have to find $a + bi$ such that $a^2 + b^2 = 25 \times 13 = 325$, so the choices we have now are:

$$[15 + 10i, 15 - 10i, 10 + 15i, 10 - 15i]$$

which are the original ones multiplyed by 5, and

$$[17 + 6i, 17 - 6i, 6 + 17i, 6 - 17i]$$

since $17^2 + 6^2$ is also 325. Similarly, when we multiply the GLS by $s = 2 + i$, we have to find $a + bi$ such that $a^2 + b^2 = 5 \times 13 = 65$, so the choices we have now are:

$$[4 + 7i, 4 - 7i, 7 - 4i, 7 + 4i, 8 - i, 8 + i, 1 + 8i, 1 - 8i]$$

where four of them are the original ones multiplied by $2 + i$.

$$[4 + 7i, 8 - i, 1 + 8i, 7 - 4i] =$$
$$(2 + i)\,[3 + 2i, 3 - 2i, 2 + 3i, 2 - 3i]$$

In summary, when N is a prime, LS can help us construct many interesting ZAC sequences in Gaussian integer. If N is a $4k+1$ prime, then we can further use GLS to construct more. The drawback of this method is that it only for prime number.

2.3 Generate IZAC using Ramanujan’s Sum

The method in this section has been published in [13].

Recall in Section 2.2.2, a signal is devided into several groups. We will use the concept in frequency domain. A signal is called gcd-delta if $s(n) = \delta(d - gcd(N, n))$ for some constant $d|N$. We express it as $s_{N,d}(n)$. For example,

$$s_{6,2}(n) = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, s_{6,6}(n) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

As we can see, gcd-delta function is actually grouping index n by $\gcd(n, N)$. However, the main difference between Section 2.2.2 and our method is that we use grouping in frequency domain, and ensure ZAC property by Theorem 2.2.1. The time domain will be integer by Ramanujan’s Sum. More precisely,

$$F\left\{s_{N,d}\right\} = \hat{s}_{N,d}(k) = c_{\frac{N}{d}}(k) \tag{2.58}$$

Proof.

$$\hat{s}_{N,d}(k) = \sum_{n=0}^{N-1} s_{N,d}(n) W_N^{nk} \tag{2.59}$$

$$= \sum_{n=0}^{N-1} \delta(d - \gcd(N, n)) W_N^{nk} \tag{2.60}$$

$$= \sum_{\gcd(N,n)=d} W_N^{nk} \tag{2.61}$$

Since $\gcd(N,n) = d$, we can let $n = n'd$ where $\gcd(N,n') = 1$.

$$\begin{aligned} \hat{s}_{N,d}(k) &= \sum_{\gcd(N,n)=d} W_N^{nk} \qquad (2.62) \\ &= \sum_{\gcd(N,n')=1} W_N^{n'dk} \qquad (2.63) \\ &= \sum_{\gcd(N,n')=1} W_{\frac{N}{d}}^{n'k} = c_{\frac{N}{d}}(k) \qquad (2.64) \end{aligned}$$

□

To construct ZAC sequence, the steps are described as follows

- Step 1. Given a signal length N, calculate all its factors d.

- Step 2. For every d choose a binary number $b_d = 0$ or 1 and a phase shift $W_N^{p_d}$, where p_d is any integer between 0 and $N-1$.

- Step 3. Let

$$g(n) = \sum_{d|N} (-1)^{b_d} W_N^{p_d n} s_{N,d}(n)$$

- Step 4. Then $\hat{g}(k) = F\{g(n)\}$ is an integer-valued ZAC squence.

Before we prove this, we provide another example to explain the ideas. Let $N = 6$, so $d = 1, 2, 3, 6$. We can arbitrarily choose $b_1 = 0, b_2 = 1, b_3 =$

$1, b_6 = 0$ and $p_1 = 2, p_2 = 1, p_3 = 2, p_6 = 0$. Since

$$s_{6,1}(n) = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, s_{6,2}(n) = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, s_{6,3}(n) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, s_{6,6}(n) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

therefore in step 3,

$$g(n) = \begin{pmatrix} 1 \\ -W_6^2 \\ -W_6^2 \\ -1 \\ -W_6^4 \\ -W_6^4 \end{pmatrix}$$

And the Fourier transform of $g(n)$ is

$$\hat{g}(k) = \begin{pmatrix} 2 \\ -1 \\ -1 \\ 2 \\ -1 \\ 5 \end{pmatrix}$$

The $\hat{g}(k)$ is an integer-valued ZAC as seen in the introduction. From the example above we can notice that $g(n)$ is CA, so by theorem Theorem 2.2.1 $G(k)$ is ZAC. We now formally prove this.

Theorem 2.3.1. *$g(n)$ in step 3 is CA.*

doi:10.6342/NTU201800930

Proof. Let $gcd(N, n) = g_n$,

$$
\begin{aligned}
g(n) &= \sum_{d|N} (-1)^{b_d} W_N^{p_d n} s_{_{N,d}}(n) \\
&= \sum_{d|N} (-1)^{b_d} W_N^{p_d n} \delta(d - gcd(N, n)) \\
&= \sum_{d|N} (-1)^{b_d} W_N^{p_d n} \delta(d - g_n) \\
&= (-1)^{b_{g_n}} W_N^{p_{g_n} n}
\end{aligned}
$$

so $g(n)$ is on unit circle for all n. In other words,

$$g^*(n)g(n) = 1 \, \forall \, n$$

which complete the proof. □

The last part is to verify $\hat{g}(k)$ is integer-valued.

Theorem 2.3.2. *$\hat{g}(k)$ in the step 4 is integer-valued.*

Proof. Since Fourier transform is a linear transformation,

$$
\begin{aligned}
\hat{g}(k) &= F\left\{g(n)\right\} \\
&= \sum_{d|N} (-1)^{b_d} F\left\{W_N^{p_d n} s_{_{N,d}}(n)\right\} \\
&= \sum_{d|N} (-1)^{b_d} \hat{s}_{_{N,d}}(k - p_d) \\
&= \sum_{d|N} (-1)^{b_d} c_{\frac{N}{d}}(k - p_d)
\end{aligned}
$$

Since $c_{\frac{N}{d}}$ is Ramanujan's sum which is integer, and the circular shift of an integer sequence is still integer sequence, thus $\hat{g}(k)$ is in fact a linear combination of integer sequence with coefficients ± 1. This proves that $\hat{g}(k)$ is integer. □

This method can be easily extended to Gaussian integer. In fact, Gaussian integer gives more degree of freedom. The concept is similar to the method in the previous section. Suppose we can find some integers a, b, c, e such that $a^2 + b^2 = c^2 + e^2$, or more simply using Pythagorean triple $a^2 + b^2 = c^2$, then the $(-1)^{b_d}$ can be substituted by $a + bi$ or $c + ei$. For example, let

$$\begin{aligned} g &= (3+4i)s_{6,6} + 5s_{6,1} + (4-3i)s_{6,2} + (3-4i)s_{6,3} && (2.65) \\ &= \begin{pmatrix} 3+4i \\ 5 \\ 4-3i \\ 3-4i \\ 4-3i \\ 5 \end{pmatrix} && (2.66) \end{aligned}$$

It is obvious that g is CA. The DFT of g is

$$\hat{g} = \begin{pmatrix} 24-6i \\ 1+11i \\ -3+3i \\ -2+2i \\ -3+3i \\ 1+11i \end{pmatrix} \tag{2.67}$$

As we can see, $\hat{g}$ is an Gaussian integer sequence with ZAC property.

2.4 Discussion

In summary, integer or Gaussian integer ZAC sequences can be generated by two kinds of method. The first kind is called time domain method. The non-linear equations are solved directly by adding more constraints. The second kind is called frequency domain method. By DFT and Theorem 2.2.1, the ZAC problem is transformed into a constant amplitude (CA) problem, which is easier to handle. In particular, using famous number theoretic sum like Gauss Sum and Ramanujan's Sum guarantees the integer property, so that abundant solutions can be found.

IZAC sequence construction is an important task in modern communication, and we have used Ramanujan's sum to solve it. Next, we will use another property of Ramanujan's sum, periodicity, in a different topic.

doi:10.6342/NTU201800930

Chapter 3

Second Application:Period Estimation

3.1 Motivation

The 1D period detection and estimation have become very popular recently, including different algorithms[19, 20, 21, 22, 23, 24, 25, 26] and theoretical study[27]. One might think period detection can be easily solved by frequency analysis, but in fact it is a hard problem. Especially when the period is integer. Consider a periodic signal whose period is 23,

$$x = [1, 2, 3, ..., 22, 23, 1, 2, 3, ..., 22, 23, 1, 2, 3, ...] \tag{3.1}$$

Since we do not know the period (otherwise we do not need period estimation), the signal length N might not be a multiple of 23. Let assume we take the first 100 points, and analyze $x(n), n = 0, 1, 2, ..., 99$, with 100 point DFT. The amplitude of spectrum is given in Figure 3.1. As we can see, the spectrum has many peaks, and the first few is at $k = 0, 4, 9, 13$,

doi:10.6342/NTU201800930

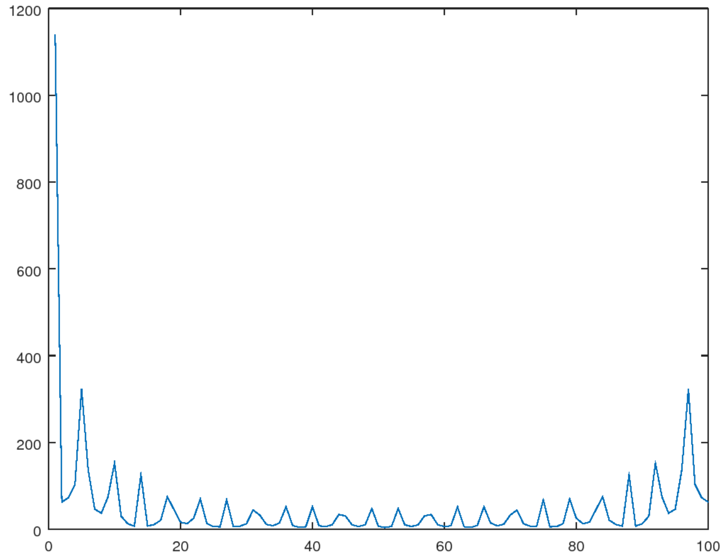

Figure 3.1: The spectrum of x, where x is defined in Equation (3.1).

because

$$100/23 \approx 4.3478 \approx 4 \tag{3.2}$$

$$100 \times 2/23 \approx 8.6957 \approx 9 \tag{3.3}$$

$$100 \times 3/23 \approx 13.043 \approx 13 \tag{3.4}$$

It is hard to inference the true period 23 from the numbers $4, 9, 13, ...$ We called this *critical length problem*. If the signal length N is selected perfectly like 92, i.e., N is a multiple of 23, then the spectrum peaks will be at $k = 0, 4, 8, 12, ...$ We will solve this in Chapter 4.

Another problem is called *mixed period problem*. Consider two periodic signal x_1 and x_2 with period 7 and 5 respectively. In particular, let

$$x_1 = [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7, ...] \tag{3.5}$$

$$x_2 = [1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2...] \tag{3.6}$$

Theoretically, $x = x_1 + x_2$ is also a periodic signal with period $7 \times 5 = 35$.

doi:10.6342/NTU201800930

But in practice, it is more desirable to determine and separate each period. It can not be easily done by traditional method like autocorrelation. In [23], the author used Ramanujan's sum to solve this problem. In the following we will briefly review his method, and extend to 2D case.

3.2 1D Period Estimation

3.2.1 1D period filterbank

Recall that in Section 2.3 we defined gcd-delta function $s_{N,d}(n)$. It is a binary function with only 0 and 1. From another point of view, the binary function can be considered as ideal filter. For example, ideal low pass filter can be performed by multiplying

$$H(k) = [1, 1, 1, ..., 1, 0, 0, ..., 0, 0, 1, 1, ..., 1] \quad (3.7)$$

in frequency domain. But unlike low pass filter, the N-point DFT of gcd-delta $s_{N,d}$ function is $c_{\frac{N}{d}}$ which is integer and periodic. The similarity and difference between ideal filter and gcd-delta function can be summarized as Table 3.1.

	Property	Ideal filter	gcd-delta
Freq Domain	Binary function	Yes	Yes
	Consecutive 1	Yes	No
Time Domain	Value	Sinc function(real)	Integer
	Periodic	No	Yes

Table 3.1: Similarity and difference between ideal filter and gcd-delta function

Traditional filterbanks divide input signal into several sub-band signals. Each sub-band signal contains certain frequencies. In practice, lowpass filter and highpass filter are often used for denoising or compression. Similarly, we can use gcd-delta function $s_{N,d}$ as *period* filterbanks, i.e., the input signal is divided into several sub-band signals and each sub-band contains certain *period*. For a toy example, let

$$x = \begin{bmatrix} 1 \\ -4 \\ 4 \\ -1 \\ 4 \\ 2 \end{bmatrix}, \; \hat{x} = \begin{bmatrix} 6 \\ -3+3\sqrt{3}i \\ -3+3\sqrt{3}i \\ 12 \\ -3-3\sqrt{3}i \\ -3-3\sqrt{3}i \end{bmatrix} \tag{3.8}$$

We now divide $\hat{x}$ into 4 channels by multiplying $s_{6,6}, s_{6,3}, s_{6,2}, s_{6,1}$.

$$\hat{x}_1 = \begin{bmatrix} 6 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \; \hat{x}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 12 \\ 0 \\ 0 \end{bmatrix}, \; \hat{x}_3 = \begin{bmatrix} 0 \\ 0 \\ -3+3\sqrt{3}i \\ 0 \\ -3-3\sqrt{3}i \\ 0 \end{bmatrix}, \; \hat{x}_6 = \begin{bmatrix} 0 \\ -3+3\sqrt{3}i \\ 0 \\ 0 \\ 0 \\ -3-3\sqrt{3}i \end{bmatrix}$$

Then by Inverse Fourier transform, the 4 sub-band signals are

$$x_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \; x_2 = \begin{bmatrix} 2 \\ -2 \\ 2 \\ -2 \\ 2 \\ -2 \end{bmatrix}, \; x_3 = \begin{bmatrix} -1 \\ -1 \\ 2 \\ -1 \\ -1 \\ 2 \end{bmatrix}, \; x_6 = \begin{bmatrix} -1 \\ -2 \\ -1 \\ 1 \\ 2 \\ 1 \end{bmatrix}$$

It is clear that

- $x_1 + x_2 + x_3 + x_6 = x$
- x_i and x_j are orthogonal to each other for all $i \neq j$.
- x_i is periodic with period i

Once we have separated the signal, further process can be implemented easily. For period estimation, the signal power is calculated for each period. By choosing peaks or ignoring the noise under certain threshold, the period candidates can be found. The estimated period is the least common multiplier (lcm) of these candidates. If multiple periods are desired (the *mix period problem*), then those candidates can be output directly.

In the next section, [28] provided a different view and a more efficient algorithm to achieve the same goal.

3.2.2 1D Impulse Train and Möbius Inversion

Some useful notation and theorem are given as follows. An impulse train $\Pi_{N,d}$ is

$$\Pi_{N,d} = \sum_{m=0}^{\frac{N}{d}-1} \delta(n - md) \tag{3.9}$$

with $d|N$. Alternatively, we can write

$$\Pi_{N,d}(n) = \begin{cases} 1 & n = md \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

An important property is that the DFT of an impulse train is another impulse train with some gain.

$$\begin{aligned} \hat{\Pi}_{N,d}(k) &= \sum_{n=0}^{N-1} \sum_{m=0}^{\frac{N}{d}-1} \delta(n - md) W_N^{kn} && (3.11) \\ &= \sum_{m=0}^{\frac{N}{d}-1} \sum_{n=0}^{N-1} \delta(n - md) W_N^{kn} && (3.12) \\ &= \sum_{m=0}^{\frac{N}{d}-1} W_N^{kmd} && (3.13) \end{aligned}$$

$$= \sum_{m=0}^{\frac{N}{d}-1} W_{\frac{N}{d}}^{km} \tag{3.14}$$

$$= \begin{cases} \frac{N}{d} & k = \frac{mN}{d} \\ 0 & \text{otherwise} \end{cases} \tag{3.15}$$

$$= \frac{N}{d}\Pi_{N,\frac{N}{d}} \tag{3.16}$$

For example,

$$\Pi_{6,2} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \hat{\Pi}_{6,2} = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 3 \\ 0 \\ 0 \end{bmatrix} = 3\Pi_{6,3} \tag{3.17}$$

Recall that gcd-delta function $s_{N,d} = \delta(d - \gcd(N,n))$. The relation between gcd-delta and impulse train is

$$\Pi_{N,d} = \sum_{d|d'|N} s_{N,d'} \tag{3.18}$$

Proof. Let $\gcd(n,N) = \bar{d}$. If $n = md$ then obviously $d|\bar{d}$. Thus,

$$\sum_{d|d'|N} s_{N,d'} = \sum_{d|d'|N} \delta(d' - \bar{d}) = 1 \tag{3.19}$$

On the other hand, if $n \neq md$, we can see $d \nmid \bar{d}$. Thus,

$$\sum_{d|d'|N} s_{N,d'} = \sum_{d|d'|N} \delta(d' - \bar{d}) = 0 \tag{3.20}$$

□

doi:10.6342/NTU201800930

For example,

$$\Pi_{6,2} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = s_{6,2} + s_{6,6} \tag{3.21}$$

Finally, we introduce Möbius function in number theory

$$\mu(n) = \begin{cases} 1 & \text{if } n \text{ is a square-free with an even number of prime factors.} \\ -1 & \text{if } n \text{ is a square-free with an odd number of prime factors.} \\ 0 & \text{if } n \text{ has a squared prime factor.} \end{cases} \tag{3.22}$$

For example, $\mu(1) = 1$, $\mu(3) = -1$, $\mu(15) = 1$, $\mu(4) = \mu(12) = 0$. The most important property of Möbius function is

$$\sum_{d|N} \mu(d) = \begin{cases} 1 & N = 1 \\ 0 & N > 1 \end{cases} \tag{3.23}$$

Now the main contribution in [28] is to rewrite Equation (3.18).

$$s_{N,d} = \sum_{d|d'|N} \mu(\frac{d'}{d})\Pi_{N,d'} \tag{3.24}$$

Take DFT for both sides.

$$\hat{s}_{N,d} = \sum_{d|d'|N} \mu(\frac{d'}{d})\hat{\Pi}_{N,d'} \tag{3.25}$$

$$\Rightarrow c_{\frac{N}{d}} = \sum_{d|d'|N} \mu(\frac{d'}{d})\frac{N}{d}\Pi_{N,\frac{N}{d}} \tag{3.26}$$

doi:10.6342/NTU201800930

It means the Ramanujan's sum can be calculated by impulse train. Note that

- μ only has value on $0, 1, -1$, which is very easy to implement.
- Impulse train, although with some gain $\frac{N}{d}$, is also easy to implement by hardware.

Therefore, any processing such as convolution can be implemented by impulse train, which is more efficient. Unfortunately, in [28], it just shows the coefficients are $0, 1, -1$, and does not know the value comes from Möbius function. In other words, it does not prove Equation (3.24). In this thesis we provide a proof.

Proof.

$$\sum_{d|d'|N} \mu(\frac{d'}{d})\Pi_{N,d'} = \sum_{d|d'|N} \mu(\frac{d'}{d}) \sum_{d'|d''|N} s_{N,d''} \tag{3.27}$$

$$= \sum_{d|d'|N} \sum_{d'|d''|N} \mu(\frac{d'}{d}) s_{N,d''} \tag{3.28}$$

$$= \sum_{d|d''|N} \sum_{d|d'|d''} \mu(\frac{d'}{d}) s_{N,d''} \tag{3.29}$$

$$= \sum_{d|d''|N} s_{N,d''} \sum_{d|d'|d''} \mu(\frac{d'}{d}) \tag{3.30}$$

$$\tag{3.31}$$

By Equation (3.23), we can obtain

$$\sum_{d|d'|N} \mu(\frac{d'}{d})\Pi_{N,d'} = \sum_{d|d''|N} s_{N,d''} \sum_{d|d'|d''} \mu(\frac{d'}{d}) \tag{3.32}$$

$$= \sum_{d|d''|N} s_{N,d''} \delta(d - d'') = s_{N,d} \tag{3.33}$$

□

If gcd-delta functions and impulse trains are organized by the sub-

doi:10.6342/NTU201800930

index in decreasing order,

$$S_6 = \left[s_{6,6}\; s_{6,3}\; s_{6,2}\; s_{6,1}\right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \Pi_6 = \left[\Pi_{6,6}\,\Pi_{6,3}\,\Pi_{6,2}\,\Pi_{6,1}\,\right] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.34}$$

Then Equation (3.18) and Equation (3.24) can be viewed as matrix product.

$$S_6 A_6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \Pi_6 \tag{3.35}$$

$$S_6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & -1 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \Pi_6 M_6 \tag{3.36}$$

$$\begin{tikzcd}
\text{gcd-delta} \arrow[r, shift left, "A_N"] \arrow[d, "\text{DFT}"'] & \text{impulse train} \arrow[l, shift left, "M_N"] \arrow[d, "\text{DFT}"] \\
\text{Ramanujan's sum} \arrow[r, shift left, "A_N"] & \text{weighted impulse train} \arrow[l, shift left, "M_N"]
\end{tikzcd}$$

Figure 3.2: Relations of gcd-delta, Ramanujan's sum and impulse train

In general, A_N and M_N is defined as

$$A_N(n,m) = \begin{cases} 1 & d_m | d_n \\ 0 & \text{otherwise} \end{cases} \tag{3.37}$$

$$M_N(n,m) = \begin{cases} \mu(\frac{d_n}{d_m}) & d_m | d_n \\ 0 & \text{otherwise} \end{cases} \tag{3.38}$$

where d_j is the jth divisor of N in decreasing order.

In summary, the relation of gcd-delta function, Ramanujan's sum and impulse train is described as Figure 3.2. Note that in this relation, the only signal which is *not* periodic is gcd-delta function. Another observation is that although impulse trains are easy to implement with hardware, they are *not orthogonal*. We can think M_N as an Gram-Schmidt orthogonalization process so that it is an upper triangular matrix. A final remark is that the relation can actually prove Ramanujan's sum is integer since A_N, M_N and the weights are all integers.

Next we will extend these results to 2D signal. As we will see, although the concept is very similar, the result is not trivial.

3.3 2D Period Estimation

Two dimensional period estimation is a hard problem since it has three parameters to determine: length, width and direction, unlike 1D peri-

doi:10.6342/NTU201800930

odic signal which has only one period parameter N such that $x(n+N) = x(n)$.

For example, the following 2D signals X_1 and X_2 share the same basic pattern

$$X_1 = \begin{pmatrix} a & b & a & b & a & b \\ c & d & c & d & c & d \\ e & f & e & f & e & f \\ a & b & a & b & a & b \\ c & d & c & d & c & d \\ e & f & e & f & e & f \end{pmatrix} \tag{3.39}$$

$$X_2 = \begin{pmatrix} a & b & e & f & c & d \\ c & d & a & b & e & f \\ e & f & c & d & a & b \\ a & b & e & f & c & d \\ c & d & a & b & e & f \\ e & f & c & d & a & b \end{pmatrix} \tag{3.40}$$

where the common basic pattern is

$$\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}$$

but they are two different periodic signals.

Recall that a 2D signal $X(n_1, n_2)$ is called periodic if there are integers a, b, c, d such that

$$X(n_1, n_2) = X(n_1 + a, n_2 + c) = X(n_1 + b, n_2 + d) \tag{3.41}$$

and we use boldface letter such as $\mathbf{P}$

$$\mathbf{P} = \begin{bmatrix} \mathrm{a} & \mathrm{b} \\ \mathrm{c} & \mathrm{d} \end{bmatrix} \tag{3.42}$$

to indicate the periodicity matrix. And it is well-known[29, 30] that the periodicity matrix can be uniquely represented as

$$\mathbf{P} = \begin{bmatrix} \mathrm{a} & \mathrm{b} \\ 0 & \mathrm{d} \end{bmatrix} \tag{3.43}$$

where $a > 0$, $d > 0$ and $0 \leq b < a$. We will use this representation throughout this article. The periodicity matrix of X_1 and X_2 in the (3.39) and (3.40) are

$$\mathbf{P}_1 = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}, \mathbf{P}_2 = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \tag{3.44}$$

respectively. The special case of $b = 0$ as (3.39) is called separable case. The advantage of using periodicity matrix is that a, b, d can directly point out the height, width and shift of the periodic pattern.

We now introduce the concept of multiple in 2D. Recall that in 1D, a signal x_1 with period 3

$$x_1 = [3, 1, 7, 3, 1, 7, 3, 1, 7, ...] \tag{3.45}$$

can also be considered as period 6, with the pattern $[3, 1, 7, 3, 1, 7]$. In other words, let x_2 be a general signal with period 6

$$x_2 = [a, b, c, d, e, f, a, b, c, ...] \tag{3.46}$$

we can say x_1 is a special case of x_2 with $a = 3, b = 1, c = 7, d = 3, e = 1, f = 7$. Moreover, the period of x_2, which is 6, is a multiple of 3, which is the period of x_1. This concept can be extended to 2D. For example,

doi:10.6342/NTU201800930

let $\mathbf{P}_3$ be

$$\mathbf{P}_3 = \begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix} \tag{3.47}$$

By definition, the pattern of $\mathbf{P}_3$ is

$$X_3 = \begin{pmatrix} a & c & e & a & c & e \\ c & e & a & c & e & a \\ e & a & c & e & a & c \\ a & c & e & a & c & e \\ c & e & a & c & e & a \\ e & a & c & e & a & c \end{pmatrix} \tag{3.48}$$

One can notice that X_3 is actually a special case of X_2, with $b = c, d = e, f = a$. Also notice that $\mathbf{P}_2$ which is the periodicity matrix of X_2, is a multiple of $\mathbf{P}_3$

$$\begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \tag{3.49}$$

In general, we call $\mathbf{P}_k$ is a multiple of $\mathbf{P}_j$ if there is a 2×2 matrix $\mathbf{M}$ such that $\mathbf{P}_k = \mathbf{P}_j\mathbf{M}$. We use the number theoretic symbol

$$\mathbf{P}_j | \mathbf{P}_k$$

to describe this relationship. Note that $\mathbf{M}$ is not necessary to be a periodicity matrix. Also note that matrix multiplying is not commutable. It means finding an $\mathbf{M}$ such that $\mathbf{P}_k = \mathbf{M}\mathbf{P}_j$ is useless. Moreover, since we only consider discrete patterns, any value in $\mathbf{P}$ and the related $\mathbf{M}$ should be integer. Furthermore, in 1D we have assumed each sub-period is a factor of signal length N, and in 2D we take a similar assumption. That

doi:10.6342/NTU201800930

is, every sub-period we want to detect is a factor of $\mathbf{N}$, where

$$\mathbf{N} = \begin{bmatrix} N & 0 \\ 0 & N \end{bmatrix}$$

Once we have defined the multiple, we can now define gcd and lcm. The least common multiple of $\mathbf{P_1}$ and $\mathbf{P_2}$ is the integral periodicity matrix $\mathbf{P}$ with the smallest determinant such that $\mathbf{P_1X} = \mathbf{P_2Y} = \mathbf{P}$ for some integral matriecs $\mathbf{X}$ and $\mathbf{Y}$. Similarly the greatest common divisor of $\mathbf{P_1}$ and $\mathbf{P_2}$ is the integral periodicity matrix $\mathbf{P}$ with the largest determinant such that $\mathbf{P_1} = \mathbf{PX}$ and $\mathbf{P_2} = \mathbf{PY}$ for some integral matriecs $\mathbf{X}$ and $\mathbf{Y}$.

3.3.1 2D Ramanujan's Sum and period filterbank

This work has been published in [31, 32].

Recall that in 1D, Ramanujan's sum can be viewed as the Fourier transform of a special kind of function called gcd-delta function. Thus , to define 2D Ramanujan's sum, we should first define 2D-gcd-delta function. Unlike 1D case, which only has 1 parameter d, 2D gcd-delta function has 3 parameters, in order to indicate the height, width and direction. A 2D-gcd-delta function $S_{N,d_1,d_2,t}(n_1, n_2)$ is defined as follows

$$S_{N,d_1,d_2,t}(n_1,n_2) = \begin{cases} 1 & if\ \gcd(n_1, N) = d_1, \\ & \gcd(n_2, N) = d_2, \\ & \frac{n_1}{d_1} - t\frac{n_2}{d_2} \equiv 0 \pmod{d} \\ 0 & \text{otherwise} \end{cases} \tag{3.50}$$

where $d = \gcd(N_1, N_2)$, $N_1 = \frac{N}{d_1}$ and $N_2 = \frac{N}{d_2}$, $n_1, n_2 = 0 \sim N - 1$ and $\gcd(t, d) = 1$.

For example,

$$S_{6,2,1,1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.51}$$

As we can see, the 2D-gcd-delta function is indeed a generalization of 1D gcd-delta function in a 2D plane. Like the 1D case, 2D-gcd-delta function only contains 0 and 1. In addition, the DFT of 2D-gcd-delta function is also a periodic signal.

$$\hat{S}_{6,2,1,1} = \begin{pmatrix} 2 & 1 & -1 & -2 & -1 & 1 \\ -1 & -2 & -1 & 1 & 2 & 1 \\ -1 & 1 & 2 & 1 & -1 & -2 \\ 2 & 1 & -1 & -2 & -1 & 1 \\ -1 & -2 & -1 & 1 & 2 & 1 \\ -1 & 1 & 2 & 1 & -1 & -2 \end{pmatrix} \tag{3.52}$$

The periodicity matrix of the above signal is

$$\mathbf{P} = \begin{bmatrix} 3 & 2 \\ 0 & 2 \end{bmatrix} \tag{3.53}$$

However, unlike the 1D case where the period of $\hat{s}_{N,d}$ can be easily obtained by $\frac{N}{d}$, the periodicity matrix of $\hat{S}_{N,d_1,d_2,t}$ can not be calculated trivially. In order to solve this problem, we need the following lemma.

Lemma 3.3.1. *If two integers t and d are coprime i.e. $\gcd(t,d) = 1$ then for any integer N there exists an integer a such that $\gcd(t + ad, N) = 1$*

Proof. First the integer N is factorized into power of serial prime integers $N = p_1^{e_1} p_2^{e_2} \ldots p_k^{e_k}$. Let Ω be the set of integers $\{j | d$ is a multiple of $p_j\}$.

Now rewrite $N = bh$ where $b = \prod_{j \in \Omega} p_j^{e_j}$. Obviously any a can satisfy $\gcd(t + ad, b) = 1$.

doi:10.6342/NTU201800930

On the other hand, since $\gcd(d,h)=1$, $d^{-1} \pmod h$ exists. We can now choose $a=(-t+1)d^{-1} \pmod N$ and this a proves the existence. □

To illustrate the lemma, let $t=8$, $d=27$ and $N=30=2\times3\times5$. Clearly $b=3, h=10$ and $27^{-1}\equiv 3 \pmod{10}$. Thus,

$$a=(-8+1)3=-21\equiv 9 \pmod{30} \tag{3.54}$$

we can easily verify that $t+ad=251$ is coprime to 30. Note that this lemma only provides existance, and in practice we can choose the smallest positive a. In this case we can choose $a=3$ and $t+ad=89$.

We can prove the following theorem.

Theorem 3.3.2. *Follow the definition of 2D gcd-delta, the periodicity matrix* $\mathbf{P}$ *of* $\hat{S}_{N,d_1,d_2,t}$ *is given as*

$$\mathbf{P}=\begin{bmatrix} \mathrm{N_1} & \frac{\mathrm{-d_2\cdot(t+ad)^{-1}}}{\mathrm{g}} \\ 0 & \frac{\mathrm{d_1}}{\mathrm{g}} \end{bmatrix} \tag{3.55}$$

where $g=\gcd(d_1,d_2)$, a *is the smallest integer such that* $(t+ad)$ *has an inverse in modulo* N_1 *and* $(t+ad)^{-1}$ *is the inverse in modulo* N_1.

Proof. Recall (24)-(27) in [31]

$$\hat{S}_{N,d_1,d_2,t}(k_1,k_2)=\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1}S_{N,d_1,d_2,t}(n_1,n_2)W_N^{-n_1k_1-n_2k_2} \tag{3.56}$$

$$=\sum_{\substack{r_1\in I_{N_1},r_2\in I_{N_2},\\ r_1-tr_2=0 \pmod d}} W_N^{-r_1d_1k_1-r_2d_2k_2} \tag{3.57}$$

$$=\sum_{\substack{r_1\in I_{N_1},r_2\in I_{N_2},\\ r_1-tr_2=0 \pmod d}} W_{N_1}^{-r_1k_1}W_{N_2}^{-r_2k_2} \tag{3.58}$$

$$=\sum_{q\in I_l} W_{N_1}^{-(t+ad)k_1q}W_{N_2}^{-k_2q} \tag{3.59}$$

$$=\sum_{q\in I_l} W_l^{-(\frac{l(t+ad)k_1}{N_1}+\frac{lk_2}{N_2})q} \tag{3.60}$$

where $l = \operatorname{lcm}(N_1, N_2)$. By (3.58), we can easily derive that

$$\begin{aligned}
&\hat{S}_{N,d_1,d_2,t}(k_1 + N_1, k_2) \\
&= \sum_{\substack{r_1 \in I_{N_1}, r_2 \in I_{N_2}, \\ r_1 - t r_2 = 0 \pmod d}} W_{N_1}^{-r_1(k_1+N_1)} W_{N_2}^{-r_2 k_2} && (3.61) \\
&= \hat{S}_{N,d_1,d_2,t}(k_1, k_2) && (3.62)
\end{aligned}$$

since $W_{N_1}^{-r_1 N_1} = 1$.

Let $r(k_1, k_2) = \frac{l(t+ad)k_1}{N_1} + \frac{lk_2}{N_2}$ and thus, by (3.60),

$$\begin{aligned}
\hat{S}_{N,d_1,d_2,t}(k_1, k_2) &= \sum_{q \in I_l} W_l^{-(\frac{l(t+ad)k_1}{N_1} + \frac{lk_2}{N_2})q} && (3.63) \\
&= \sum_{q \in I_l} W_l^{-r(k_1,k_2)q} && (3.64)
\end{aligned}$$

Note that r is periodic by

$$\begin{aligned}
&r(k_1 + \frac{-d_2 \cdot (t+ad)^{-1}}{g}, k_2 + \frac{d_1}{g}) \\
&= \frac{l(t+ad)}{N_1}\left(k_1 + \frac{-d_2 \cdot (t+ad)^{-1}}{g}\right) + \frac{l}{N_2}\left(k_2 + \frac{d_1}{g}\right) && (3.65)
\end{aligned}$$

Since $(t+ad)^{-1}$ is the inverse of $(t+ad)$ in modulo N_1, $(t+ad)^{-1}(t+ad) = 1 + sN_1$ for some integer s and (3.65) above becomes

$$\begin{aligned}
&r(k_1 + \frac{-d_2 \cdot (t+ad)^{-1}}{g}, k_2 + \frac{d_1}{g}) \\
&= \frac{l(t+ad)k_1}{N_1} - \frac{l(1+sN_1)}{N_1}\frac{d_2}{g} + \frac{l}{N_2}\left(k_2 + \frac{d_1}{g}\right) && (3.66) \\
&= r(k_1, k_2) - ls\frac{d_2}{g} - \frac{ld_2}{N_1 g} + \frac{ld_1}{N_2 g} && (3.67)
\end{aligned}$$

By $N_1 d_1 = N_2 d_2 = N$ therefore $\frac{d_2}{N_1} = \frac{d_1}{N_2}$ and (3.67) becomes

$$r(k_1 + \frac{-d_2 \cdot (t+ad)^{-1}}{g}, k_2 + \frac{d_1}{g})$$

doi:10.6342/NTU201800930

$$= r(k_1, k_2) - ls\frac{d_2}{g} \tag{3.68}$$

Now by (3.64) and (3.68) we can prove that

$$\hat{S}_{N,d_1,d_2,t}(k_1 + \frac{-d_2}{g} \cdot (t+ad)^{-1}, k_2 + \frac{d_1}{g})$$

$$= \sum_{q \in I_l} W_l^{-r(k_1 + \frac{-d_2 \cdot (t+ad)^{-1}}{g}, k_2 + \frac{d_1}{g})q} \tag{3.69}$$

$$= \sum_{q \in I_l} W_l^{-\left(r(k_1,k_2) - ls\frac{d_2}{g}\right)q} \tag{3.70}$$

$$= \sum_{q \in I_l} W_l^{-r(k_1,k_2)q} \tag{3.71}$$

$$= \hat{S}_{N,d_1,d_2,t}(k_1, k_2) \tag{3.72}$$

because $g = \gcd(d_1, d_2)$ and then d_2/g is an integer. By (3.62) and (3.72) the proof is completed. □

Example 3.3.1. *Consider* $\hat{S}_{6,2,1,1}$ *in* (3.52)*. It follows that* $N = 6$*,* $d_1 = 2$*,* $d_2 = 1$*,* $t = 1$ *and*

$$N_1 = \frac{N}{d_1} = 3$$
$$N_2 = \frac{N}{d_2} = 6$$
$$d = \gcd(N_1, N_2) = 3$$

From $(t + ad)$ *has inverse in modulo* N_1 *we can use* $a = 0$ *and* $(t + ad)^{-1} \equiv 1 \pmod 3$*. Therefore the periodicity matrix from Theorem 3.3.2 is*

$$\begin{bmatrix} 3 & 2 \\ 0 & 2 \end{bmatrix}$$

as we expected.

doi:10.6342/NTU201800930

Example 3.3.2. *Consider $S_{4,2,2,1}$.*

$$S_{4,2,2,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.73}$$

The 2D DFT of $S_{4,2,2,1}$ is

$$\hat{S}_{4,2,2,1} = \begin{pmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{pmatrix} \tag{3.74}$$

which has the periodicity matrix

$$\begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}$$

one can easily check that $a = 0$ and the theorem is correct.

Example 3.3.3. *The final example shows the importance of a. Let $N = 30$, $d_1 = 3$, $d_2 = 2$, $N_1 = 30/3 = 10$, $N_2 = 30/2 = 15$, $d = 5$ and $t = 4 \in I_d$. Note that if $a = 0$, $t = 4$ has no inverse in modulo 10. Therefore we must choose $a = 1$ and $t + ad = 9$. Since $9^{-1} \equiv 9 \pmod{10}$, the periodicity matrix is*

$$\begin{bmatrix} 10 & -2 \times 9 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 10 & 2 \\ 0 & 3 \end{bmatrix}$$

There are several good reasons to use 2D-gcd-delta functions $S_{N,d_1,d_2,t}$ as the sub-band. First of all, $S_{N,d_1,d_2,t}$ is disjointed to each other while different d_1, d_2, t can fulfill the whole $N \times N$ image. Secondly $\hat{S}_{N,d_1,d_2,t}$ is integral so fast and accurate algorithm can be found for image processing since the pixel value is also an integer. The final reason is that

2D-gcd-delta functions have closed form relationship to the periodicity matrix as we have seen in Theorem 3.3.2.

The number of sub-bands in 1D can be easily determined by the number of divisors of N. However in 2D, the number of sub-bands is equal to the number of cyclic sub-groups of $Z_N \times Z_N$, which is obtained in [33]. For example, let $N = 4$, there are 10 sub-bands

$$S_{4,4,4,1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \hat{S}_{4,4,4,1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \mathbf{P} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$S_{4,4,2,1} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \hat{S}_{4,4,2,1} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \end{bmatrix}, \mathbf{P} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

$$S_{4,4,1,1} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \hat{S}_{4,4,1,1} = \begin{bmatrix} 2 & 0 & -2 & 0 \\ 2 & 0 & -2 & 0 \\ 2 & 0 & -2 & 0 \\ 2 & 0 & -2 & 0 \end{bmatrix}, \mathbf{P} = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$$

$$S_{4,1,4,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \hat{S}_{4,1,4,1} = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 \\ -2 & -2 & -2 & -2 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{P} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$

doi:10.6342/NTU201800930

$$S_{4,2,4,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \hat{S}_{4,2,4,1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix}, \mathbf{P} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

$$S_{4,1,1,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \hat{S}_{4,1,1,1} = \begin{bmatrix} 2 & 0 & -2 & 0 \\ 0 & -2 & 0 & 2 \\ -2 & 0 & 2 & 0 \\ 0 & 2 & 0 & -2 \end{bmatrix}, \mathbf{P} = \begin{bmatrix} 4 & 3 \\ 0 & 1 \end{bmatrix}$$

$$S_{4,1,2,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \hat{S}_{4,1,2,1} = \begin{bmatrix} 2 & -2 & 2 & -2 \\ 0 & 0 & 0 & 0 \\ -2 & 2 & -2 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{P} = \begin{bmatrix} 4 & 2 \\ 0 & 1 \end{bmatrix}$$

$$S_{4,1,1,3} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \hat{S}_{4,1,1,3} = \begin{bmatrix} 2 & 0 & -2 & 0 \\ 0 & 2 & 0 & -2 \\ -2 & 0 & 2 & 0 \\ 0 & -2 & 0 & 2 \end{bmatrix}, \mathbf{P} = \begin{bmatrix} 4 & 1 \\ 0 & 1 \end{bmatrix}$$

$$S_{4,2,1,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \hat{S}_{4,2,1,1} = \begin{bmatrix} 2 & 0 & -2 & 0 \\ -2 & 0 & 2 & 0 \\ 2 & 0 & -2 & 0 \\ -2 & 0 & 2 & 0 \end{bmatrix}, \mathbf{P} = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}$$

$$S_{4,2,2,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \hat{S}_{4,2,2,1} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}, \mathbf{P} = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}$$

There is one important thing to notice. These 10 sub-bands do not cover all patterns directly. For example,

$$\mathbf{P} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

which is obviously a factor of $\mathbf{N} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$, is not in the sub-bands. This does not happen in 1D. Recall that in 1D, all the factors of N are considered. The pattern can be detected, however, by finding LCM of the peaks of 10 sub-bands.

3.3.2 2D LCM for Period Detection

Once the signal is decomposed into sub-band, the only job remains is calculating the least common multiplier (LCM) of those periods, in order to estimate the true period. In 1D, this calculation is trivial. However,

doi:10.6342/NTU201800930

in 2D the period is illustrated by 2×2 periodicity matrix, and the matrix LCM is not an easy task. To solve this we let

$$\mathbf{P_1} = \begin{bmatrix} a_1 & b_1 \\ 0 & d_1 \end{bmatrix} \tag{3.75}$$

$$\mathbf{P_2} = \begin{bmatrix} a_2 & b_2 \\ 0 & d_2 \end{bmatrix} \tag{3.76}$$

throughout this section.

Recall that the least common multiple of $\mathbf{P_1}$ and $\mathbf{P_2}$ is the periodicity matrix $\mathbf{P}$ with the smallest determinant such that $\mathbf{P_1X} = \mathbf{P_2Y} = \mathbf{P}$ for some matrices $\mathbf{X}$ and $\mathbf{Y}$. In other words,

$$\begin{bmatrix} a_1 & b_1 \\ 0 & d_1 \end{bmatrix} \begin{bmatrix} a_x & b_x \\ 0 & d_x \end{bmatrix} = \begin{bmatrix} a_2 & b_2 \\ 0 & d_2 \end{bmatrix} \begin{bmatrix} a_y & b_y \\ 0 & d_y \end{bmatrix} \tag{3.77}$$

$$= \begin{bmatrix} a & b \\ 0 & d \end{bmatrix} \tag{3.78}$$

From (3.77) we can deduce

$$a_1 a_x = a_2 a_y = a \tag{3.79}$$

$$a_1 b_x + b_1 d_x = a_2 b_y + b_2 d_y = b \tag{3.80}$$

$$d_1 d_x = d_2 d_y = d \tag{3.81}$$

Note that a, a_x and a_y only appear in (3.79). To minimize a we can directly choose

$$a_x = \frac{a_2}{\gcd(a_1, a_2)} \tag{3.82}$$

$$a_y = \frac{a_1}{\gcd(a_1, a_2)} \tag{3.83}$$

Equivalently, $a = \mathrm{lcm}(a_1, a_2)$ since $a_1 a_2 = \mathrm{lcm}(a_1, a_2) \cdot \gcd(a_1, a_2)$. On the

doi:10.6342/NTU201800930

other hand, from (3.81), however, we can not immediately set $d_x = d_2/\gcd(d_1, d_2)$ since it might controdict (3.80). To avoid this let

$$d_x = k\frac{d_2}{\gcd(d_1, d_2)} \tag{3.84}$$

$$d_y = k\frac{d_1}{\gcd(d_1, d_2)} \tag{3.85}$$

we should minimize k to minimize d. To find the minimum of k we rearrange (3.80) as

$$a_1b_x - a_2b_y = b_2d_y - b_1d_x \tag{3.86}$$

$$= k\frac{b_2d_1 - b_1d_2}{\gcd(d_1, d_2)} \tag{3.87}$$

$$= k(b_2q - b_1p) \tag{3.88}$$

where $q = d_1/\gcd(d_1, d_2)$ and $p = d_2/\gcd(d_1, d_2)$ are integers. The equation above is solvable if and only if

$$k(b_2q - b_1p) = t \cdot \gcd(a_1, a_2) \tag{3.89}$$

for some integer t. Let $s = \gcd(a_1, a_2)$, the minimum k is equal to

$$k_{\text{min}} = \frac{s}{\gcd(s, b_2q - b_1p)} \tag{3.90}$$

The proof is elementary. The term $\gcd(s, b_2q - b_1p)$ indicates the largest divider of s from which $b_2q - b_1p$ can provide, and k only has to provide the rest, which is $s/\gcd(s, b_2q - b_1p)$. For example let $s = 6$ and $b_2q - b_1p = 9$, since $s = 3 \times 2$ but $b_2q - b_1p$ can only provide the factor 3, one can easily check that minimum of k satisfied (3.89) is 2.

Once k_{min} is determined, by (3.85) we can obtain d_y and then from (3.81) we can obtain d. In order to calculate b, we first use (3.89) to calculate t. Then (3.88) becomes

$$a_1 b_x - a_2 b_y = t \cdot \gcd(a_1, a_2) \tag{3.91}$$

We can now use extended Euclidean algorithm [34] to obtain b'_x and b'_y such that

$$a_1 b'_x - a_2 b'_y = \gcd(a_1, a_2) \tag{3.92}$$

Finally $b_x = t \cdot b'_x$ and $b_y = t \cdot b'_y$ and by (3.80) b is obtained.

The discussion above are summarized in the Algorithm 1.

Algorithm 1 Calculate the least common multiple of two periodicity matrix.

```
Input:
    $\mathbf{P_1} = \begin{bmatrix} a_1 & b_1 \\ 0 & d_1 \end{bmatrix}$
    $\mathbf{P_2} = \begin{bmatrix} a_2 & b_2 \\ 0 & d_2 \end{bmatrix}$
Output:
    $\mathbf{P} = \begin{bmatrix} \mathrm{a} & \mathrm{b} \\ 0 & \mathrm{d} \end{bmatrix}$
    Such that $\mathbf{P_1 X} = \mathbf{P_2 Y} = \mathbf{P}$ for some matrices $\mathbf{X}$ and $\mathbf{Y}$.
$a = \mathrm{lcm}(a_1, a_2)$ and let $s = \gcd(a_1, a_2)$, $q = d_1/\gcd(d_1, d_2)$, $p = d_2/\gcd(d_1, d_2)$;
calculate $k = k_{\text{min}}$ from (3.90);
obtain $d_y$ from (3.85);
obtain $d$ from (3.81);
calculate $t$ by (3.89);
use extended Euclidean algorithm to obtain $b'_y$ from (3.92)
$b_y = t b'_y$;
obtain $b$ from (3.80), $b \equiv b \pmod{a}$; return
    $\mathbf{P} = \begin{bmatrix} \mathrm{a} & \mathrm{b} \\ 0 & \mathrm{d} \end{bmatrix}$
```

We end this section by a concrete example. Let

$$\mathbf{P_1} = \begin{bmatrix} 4 & 3 \\ 0 & 6 \end{bmatrix}, \mathbf{P_2} = \begin{bmatrix} 6 & 1 \\ 0 & 3 \end{bmatrix}$$

It follows that

$$\begin{aligned}
& a = \operatorname{lcm}(4,6) = 12 \\
& s = \gcd(4,6) = 2, q = 6/\gcd(6,3) = 2, p = 3/\gcd(6,3) = 1 \\
& b_2 q - b_1 p = -1, k = \frac{2}{\gcd(2,-1)} = 2 \\
& d_y = k\frac{d_1}{\gcd(d_1,d_2)} = 2\frac{6}{3} = 4 \\
& d = d_2 d_y = 12 \\
& 2 \cdot (-1) = 2t, t = -1 \\
& 4b'_x - 6b'_y = 2, b'_x = 2, b'_y = 1 \\
& b_y = tb'_y = -1 \\
& b = a_2 b_y + b_2 d_y \\
& \ = 6 \cdot (-1) + 1 \cdot 4 = -2 = 10 \pmod a
\end{aligned}$$

Thus,

$$\operatorname{lcm}(\mathbf{P_1}, \mathbf{P_2}) = \begin{bmatrix} 12 & 10 \\ 0 & 12 \end{bmatrix}$$

In order to prove that this is the common multiple with the smallest determinant, assume there is a smaller one $\mathbf{P_G}$. Note that $d_1 = 6$ so d must greater or equal to 6. On the other hand $a = \operatorname{lcm}(6,4) = 12$ is already minimum. Thus, we can assume

$$\mathbf{P_G} = \begin{bmatrix} 12 & \mathrm{b} \\ 0 & 6 \end{bmatrix}$$

for some constant b, such that $\mathbf{P_1X} = \mathbf{P_2Y} = \mathbf{P_G}$, in other words

$$\begin{bmatrix} 4 & 3 \\ 0 & 6 \end{bmatrix} \begin{bmatrix} 3 & b_x \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 1 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 2 & b_y \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 12 & b \\ 0 & 6 \end{bmatrix}$$

This implies $b = 4b_x + 3 = 6b_y + 2$. However, because $4b_x + 3$ is odd and $6b_y + 2$ is even, such b does not exist. This completes the proof.

3.3.3 2D Period Detection Examples

A toy example of 6×6

First we will show a toy example of 6×6. The input signal we choose is

$$X = \begin{pmatrix} 8 & 1 & 3 & 5 & 4 & 9 \\ 3 & 5 & 4 & 9 & 8 & 1 \\ 4 & 9 & 8 & 1 & 3 & 5 \\ 8 & 1 & 3 & 5 & 4 & 9 \\ 3 & 5 & 4 & 9 & 8 & 1 \\ 4 & 9 & 7 & 1 & 3 & 6 \end{pmatrix} \tag{3.93}$$

The periodicity matrix is

$$\mathbf{P} = \begin{bmatrix} 3 & 2 \\ 0 & 2 \end{bmatrix}$$

Note that we deliberately change two values of the last row $8 \to 7$ and $5 \to 6$ in order to demonstrate the robustness. To determine the period, the first step is to decompose the input image X into several periodic sub-band signals. The energy spectrum of X is

$$|\hat{X}| \simeq \begin{bmatrix} 180 & 2 & 0 & 2 & 0 & 2 \\ 0 & 2 & 0 & 2 & 0 & 2 \\ 0 & 53 & 0 & 2 & 45 & 2 \\ 0 & 2 & 0 & 2 & 0 & 2 \\ 0 & 2 & 45 & 2 & 0 & 53 \\ 0 & 2 & 0 & 2 & 0 & 2 \end{bmatrix} \tag{3.94}$$

As we can see, those 2s in the above equation can be ignored by peak finding, and the remaining parts can be mapping to 3 different sub-band signals.

$$S_{6,6,6,1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.95}$$

$$S_{6,2,1,1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.96}$$

doi:10.6342/NTU201800930

$$S_{6,2,2,2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.97}$$

Thus, the 3 corresponding periodicity matrices are

$$S_{6,6,6,1} \rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.98}$$

$$S_{6,2,1,1} \rightarrow \begin{bmatrix} 3 & 2 \\ 0 & 2 \end{bmatrix} \tag{3.99}$$

$$S_{6,2,2,2} \rightarrow \begin{bmatrix} 3 & 1 \\ 0 & 1 \end{bmatrix} \tag{3.100}$$

And finally the LCM of the above matrices is

$$\begin{bmatrix} 3 & 2 \\ 0 & 2 \end{bmatrix} \tag{3.101}$$

as we expected.

In this toy example we just demonstrated the guideline of our method. However, when the image size gets larger and the noise level is higher, it is difficult to determine the threshold that distinguish peaks or not. There are two solutions for this problem. First recall that the DFT of 2D-gcd-delta function is 2D version IIPF[25], we can use IIPF spectrum in order to enhance the peak. Second, as a rule of thumb, the LCM of the periodicity matrices is usually stable, which means if we sort the top k

doi:10.6342/NTU201800930

periodicity matrices (order by magnitude), namely $\mathbf{P_1}, \mathbf{P_2}, \mathbf{P_3}, ..., \mathbf{P_k}$, then

$$\mathbf{B_1} = \mathbf{P_1} = \text{Identity matrix (usually)} \tag{3.102}$$

$$\mathbf{B_2} = \mathbf{LCM}(\mathbf{P_1}, \mathbf{P_2}) = \mathbf{P_2} \text{ (usually)} \tag{3.103}$$

$$\mathbf{B_3} = \mathbf{LCM}(\mathbf{P_1}, \mathbf{P_2}, \mathbf{P_3}) = \mathbf{LCM}(\mathbf{B_2}, \mathbf{P_3}) \tag{3.104}$$

$$\mathbf{B_4} = \mathbf{LCM}(\mathbf{P_1}, ..., \mathbf{P_4}) = \mathbf{LCM}(\mathbf{B_3}, \mathbf{P_4})... \tag{3.105}$$

$$\mathbf{B_k} = \mathbf{LCM}(\mathbf{P_1}, ..., \mathbf{P_k}) = \mathbf{LCM}(\mathbf{B_{k-1}}, \mathbf{P_k}) \tag{3.106}$$

The LCM chain $\mathbf{B_1}, ..., \mathbf{B_k}$ is stable means there is some j such that $\mathbf{B_j} = \mathbf{B_{j+1}} = \mathbf{B_{j+2}}....$ We suggest that use $k = 10$ and look for 3 repeated $\mathbf{B}$ if any.

Computer generated image

In this settings we generate a 48×48 binary image with the periodicity matrix

$$\mathbf{P} = \begin{bmatrix} 4 & 1 \\ 0 & 3 \end{bmatrix}$$

as Figure 3.3. In order to test the robustness, we randomly remove some of the white pixels and obtain the input noisy image $I(n_1, n_2)$ as Figure 3.4.

The first step is to decompose the input image I into several periodic subband signals $\hat{S}_{48,d_1,d_2,t}$. To accomplish this we take 2D DFT to I, find peaks in $\hat{I}$ and then map those peaks to $S_{48,d_1,d_2,t}$. Recall that from [33] there are 230 cyclic subgroups of $Z_{48} \times Z_{48}$. The corresponding magnitude of the spectrum is illustrated as Figure 3.5. Note that each index at x-axis from 1 to 230 is actually mapping to a $S_{48,d_1,d_2,t}$ but does not indicate the frequency. In the ideal case, the spectrum is sparse which means there is only a few nonzero bins. However in our case since the image is defected, there is some random noise shown in Figure 3.5.

doi:10.6342/NTU201800930

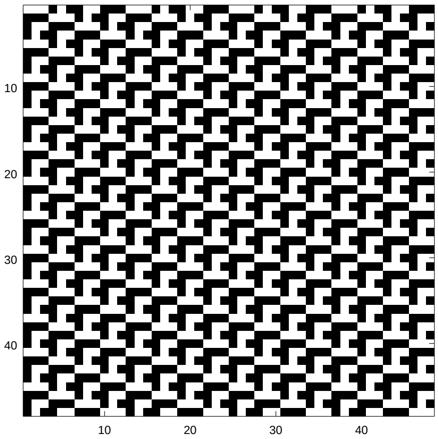

Figure 3.3: The original 2D Pattern

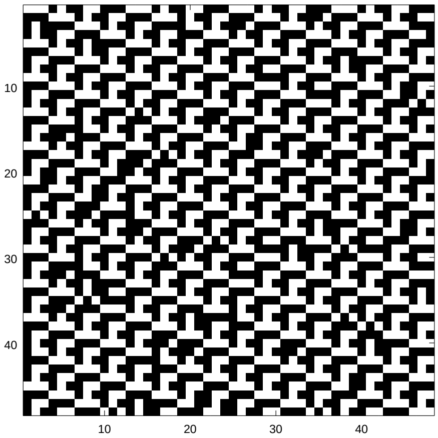

Figure 3.4: Some random removal from Figure 3.3

Nevertheless the noise is negligible for any peaks finding algorithm. The second step is to calculate the periodicity matrix of each $\hat{S}_{48,d_1,d_2,t}$. In practice, the calculation can be saved by table lookup method. Those relations can be precalculated if N is fixed. In this example, the 6 peaks from left to right are mapping to the following periodicity matrices

$$\text{The left most one at 1: } \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\text{, the DC} \tag{3.107}$$

doi:10.6342/NTU201800930

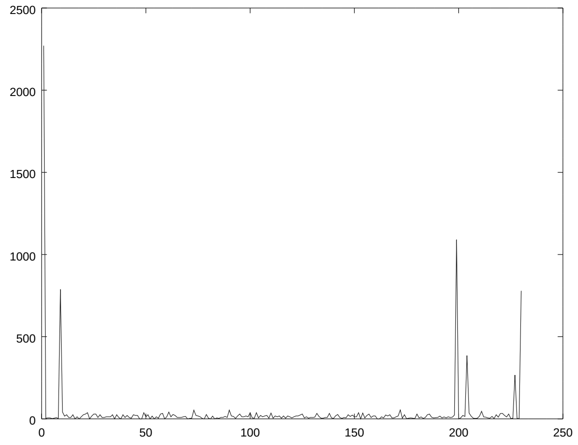

Figure 3.5: Spectrum obtained from Figure 3.4

$$\text{The second one at 9: } \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \tag{3.108}$$

$$\text{The third one at 199: } \begin{bmatrix} 4 & 3 \\ 0 & 1 \end{bmatrix} \tag{3.109}$$

$$\text{The fourth one at 204: } \begin{bmatrix} 4 & 1 \\ 0 & 3 \end{bmatrix} \text{, the ground truth} \tag{3.110}$$

$$\text{The fifth one at 227: } \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix} \tag{3.111}$$

$$\text{The right most one at 230: } \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \tag{3.112}$$

The final step is to calculate the least common multiplier of the matrices

doi:10.6342/NTU201800930

above. The LCM is

$$\mathbf{P} = \begin{bmatrix} 4 & 1 \\ 0 & 3 \end{bmatrix} \tag{3.113}$$

$$\text{Since } \begin{bmatrix} 4 & 1 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 1 \\ 0 & 3 \end{bmatrix} \tag{3.114}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 4 & 1 \\ 0 & 1 \end{bmatrix} \tag{3.115}$$

$$= \begin{bmatrix} 4 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 \\ 0 & 3 \end{bmatrix} \tag{3.116}$$

$$= \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.117}$$

$$\begin{bmatrix} 4 & 1 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 4 & 5 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix} \tag{3.118}$$

As we can see, although the ground truth period does not have the maximum magnitude in the spectrum, it can be obtained by the LCM of top 6 peaks. Furthermore, even we take wrong threshold and choose only top 4 peaks at 1, 9, 199 and 230, the LCM is still the correct. It means the LCM chain is stable from 4 to 6 and shows the robustness of this proposed method.

After the correct period is obtained, several applications can be immediately realized. For example, the original image can be recovered be averaging or voting from the repeated patterns. The result can be further used in image compression as we only use the periodicity matrix and the basic pattern to generate the whole image.

The second experiment of the computer generated image can be found from Figure 3.6 to Figure 3.9. The basic pattern in Figure 3.6 is

doi:10.6342/NTU201800930

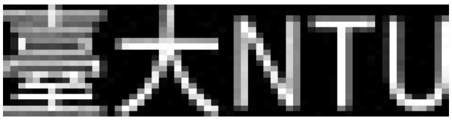

Figure 3.6: The image is a combination of Chinese characters and English letters. The width is 60 pixels and the height is 20 pixels. The meaning of those two Chinese characters is National Taiwan University (NTU)

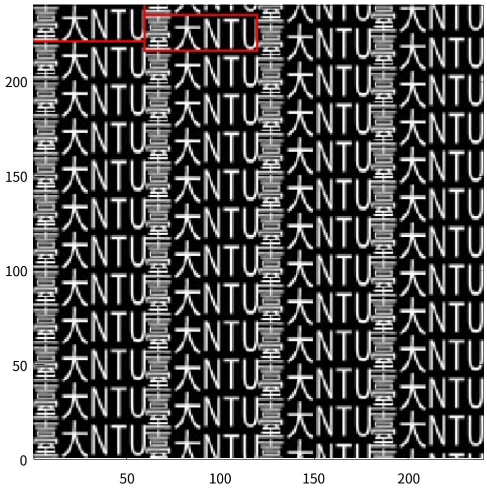

Figure 3.7: The 240x240 image repeating the Figure 3.6. The periodicity matrix is given in the text. The red box indicates the basic pattern and the direction of period

20x60 and the periodicity matrix in Figure 3.7 is

$$\begin{bmatrix} 20 & 5 \\ 0 & 60 \end{bmatrix}$$

To test the robustness, additive white Gaussian noise (AWGN) with signal to noise ratio (SNR) 3db is added, as in the Figure 3.8 . The spectrum is illustrated in Figure 3.9. Although the spectrum is not as clean as Figure 3.5, the peaks are still distinguishable. With the threshold 1000, the

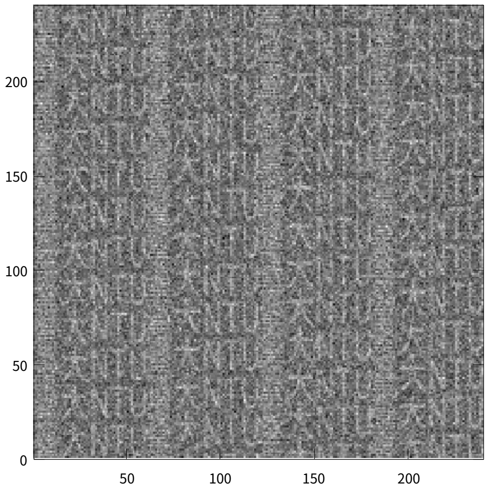

Figure 3.8: A 240x240 noisy image with the pattern in Figure 3.6. The added noise is white Gaussian with SNR=3db

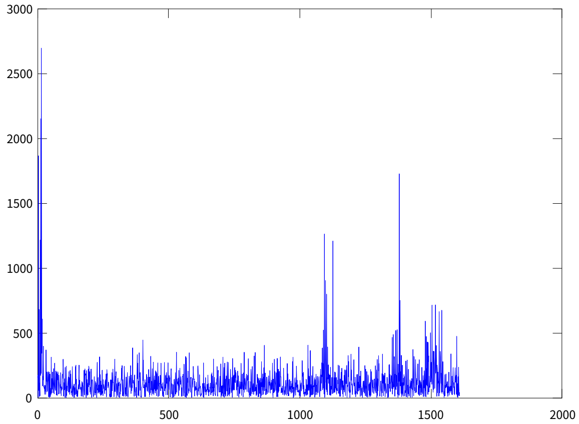

Figure 3.9: The spectrum from Figure 3.8. The first peak (DC) is not shown because the value is too large.

following periodicity matrices are chosen, ordered by magnitude

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 6 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}, \begin{bmatrix} 10 & 3 \\ 0 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 60 \end{bmatrix},$$

$$\begin{bmatrix} 20 & 3 \\ 0 & 4 \end{bmatrix}, \begin{bmatrix} 20 & 15 \\ 0 & 4 \end{bmatrix}, \begin{bmatrix} 20 & 9 \\ 0 & 12 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}$$

One can easily check the LCM chain is stable from 6 to 8 and $\mathbf{B_6} = \mathbf{B_7} = \mathbf{B_8} =$

$$\begin{bmatrix} 20 & 5 \\ 0 & 60 \end{bmatrix}$$

doi:10.6342/NTU201800930

as expected.

Real world image

In this experiment we use the artwork ”Pegasus” by the Dutch artist M. C. Escher[35]. The image is cropped and downsampled to 192×192. The length and width of the basic pattern, by observing the ”black Pegasus” on the top left, is around 96×48 and the direction is about $\frac{\pi}{4}$. Thus, the ground truth of the periodicity matrix is

$$\begin{bmatrix} 96 & 48 \\ 0 & 48 \end{bmatrix}$$

The number of cyclic subgroups of $Z_{192} \times Z_{192}$ is 950 and the corresponding magnitude of the spectrum is illustrated as Figure 3.11. We do not show the DC because the value is too large compared to others. Compare Figure 3.11 and Figure 3.5 we can notice that the real world image has much noise, but the peaks are still quite clear and distinguishable. The 6 non-DC peaks represent

$$\begin{bmatrix} 96 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 96 & 93 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 32 & 31 \\ 0 & 1 \end{bmatrix},$$
$$\begin{bmatrix} 32 & 1 \\ 0 & 3 \end{bmatrix}, \begin{bmatrix} 96 & 91 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} 64 & 42 \\ 0 & 1 \end{bmatrix}$$

respectively. And the LCM chain $\mathbf{B}$ is

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 96 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 96 & 24 \\ 0 & 24 \end{bmatrix}, \begin{bmatrix} 96 & 48 \\ 0 & 48 \end{bmatrix},$$
$$\begin{bmatrix} 96 & 48 \\ 0 & 48 \end{bmatrix}, \begin{bmatrix} 96 & 0 \\ 0 & 192 \end{bmatrix}, \begin{bmatrix} 192 & 0 \\ 0 & 192 \end{bmatrix}$$

doi:10.6342/NTU201800930

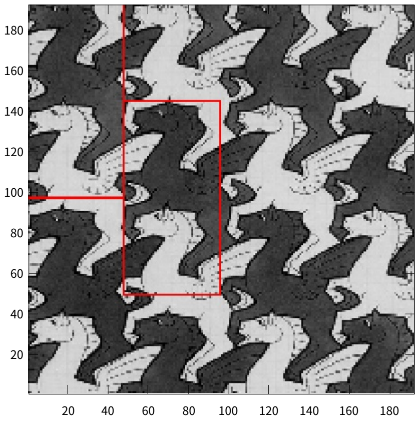

Figure 3.10: Pegasus, by M.C. Escher 1959. The image is cropped and downsampled to 192×192. The red box indicates the approximated basic pattern.

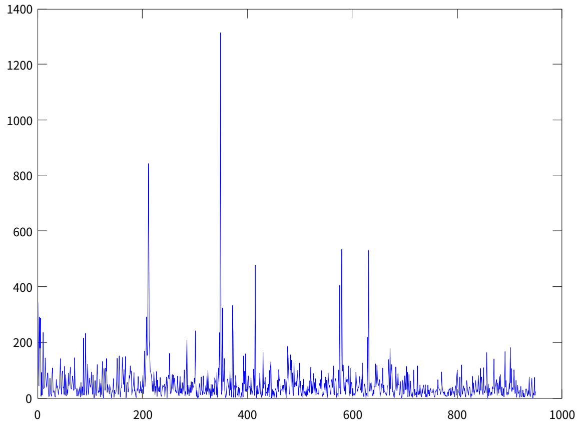

Figure 3.11: The spectrum from Figure 3.10. The first peak (DC) is not shown because the value is too large.

It does not repeat 3 times except $\mathbf{N} = \begin{bmatrix} 192 & 0 \\ 0 & 192 \end{bmatrix}$ (the whole image). So in this case we choose the matrix repeated twice $\mathbf{B_4} = \mathbf{B_5}$, which is exactly what we expected.

3.3.4 Discussion on Möbius Inversion in 2D

Recall that in 1D, the DFT of an impulse train with period d is another impulse train with period $\frac{N}{d}$. If we want to generalize the 1D result in Equation (3.18) and Equation (3.24) into 2D, the 2D impulse train must be defined. To simplify the notation, we do not use $S_{N,d_1,d_2,t}$ for the 2D

doi:10.6342/NTU201800930

gcd-delta function. Instead, we use $S_{N,\mathbf{P}}$ for the 2D gcd-delta function, where $\hat{S}_{N,\mathbf{P}}$ has periodicity matrix $\mathbf{P}$. We use the notation $\mathbf{I}$ for identity matrix and $\mathbf{N}$

$$\mathbf{N} = \begin{bmatrix} N & 0 \\ 0 & N \end{bmatrix}$$

as before. The row representation $\mathbf{P}^t$ is the same pattern as $\mathbf{P}$, but the matrix indicates the period from its row. For example, if

$$X = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

then

$$\mathbf{P} = \begin{bmatrix} 4 & 2 \\ 0 & 1 \end{bmatrix}, \mathbf{P}^t = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}$$

More precisely, the periodicity matrix suggests that

$$X(\begin{bmatrix} n_1 \\ n_2 \end{bmatrix}) = X(\begin{bmatrix} n_1 \\ n_2 \end{bmatrix} + \mathbf{P}\begin{bmatrix} m_1 \\ m_2 \end{bmatrix})$$

and the row representation suggests that

$$X([n_1, n_2]) = X([n_1, n_2] + [m_1, m_2]\mathbf{P}^t)$$

Note that the transpose of $\mathbf{P}$ is not necessarily equal to $\mathbf{P}^t$. We can now define impulse trains $\Pi_{N,\mathbf{P}}$ and discuss their properties.

$$\Pi_{N,\mathbf{P}}(n_1, n_2) = \begin{cases} 1, & \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} = \mathbf{P}\begin{bmatrix} m_1 \\ m_2 \end{bmatrix} \text{ for some } m_1, m_2 \\ 0, & \text{otherwise} \end{cases} \tag{3.119}$$

doi:10.6342/NTU201800930

For example, if $\mathbf{P} = \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}$,

$$\Pi_{6,\mathbf{P}}(n_1, n_2) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.120}$$

Like 1D impulse train, the DFT of 2D impulse train is another impulse train. Let the periodicity matrix of the original impulse train be $\mathbf{P}$ and the transformed impulse train be $\mathbf{Q}$, then we have

$$\mathbf{Q}^t\mathbf{P} = \mathbf{N} \tag{3.121}$$

the proof is similar to 1D case so we omit it here. Take the matrix in Equation (3.120) as example,

$$\hat{\Pi}_{6,\mathbf{P}}(k_1, k_2) = \begin{bmatrix} 6 & 0 & 6 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 6 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.122}$$

whose periodicity matrix is $\begin{bmatrix} 6 & 3 \\ 0 & 1 \end{bmatrix}$ and its row representation is $\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}$. Obviously,

$$\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 6 & 6 \\ 0 & 6 \end{bmatrix} = \begin{bmatrix} 6 & 0 \\ 0 & 6 \end{bmatrix}$$

doi:10.6342/NTU201800930

Now we can describe the relation between $S_{N,\mathbf{P}}$ and $\Pi_{N,\mathbf{P}}$

$$\Pi_{N,\mathbf{P}} = \sum_{\mathbf{Q}|\mathbf{P}} S_{N,\mathbf{Q}^t} \tag{3.123}$$

$$S_{N,\mathbf{Q}} = \sum_{\mathbf{Q}|\mathbf{P}|\mathbf{N}} \mu(\frac{\mathbf{P}}{\mathbf{Q}})\Pi_{N,\mathbf{P}^t} \tag{3.124}$$

where μ here is generalized Möbius function, with following recursive definition.

$$\sum_{\mathbf{P}|\mathbf{N}} \mu(\mathbf{P}) = \begin{cases} 1, & \mathbf{N} = \mathbf{I} \\ 0, & \text{otherwise} \end{cases} \tag{3.125}$$

The proof is exactly the same with 1D so we omit is here. Take Equation (3.120) as example, note that $\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}$ has four factors

$$\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.126}$$

and their row representations are

$$\begin{bmatrix} 6 & 3 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.127}$$

$$\Pi_{6,\mathbf{P}}(n_1, n_2) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.128}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.129}$$

we can find the periodicity matrices of above 2D-gcd-delta are the same as Equation (3.127). The inversion formula gives

$$S_{6,\mathbf{Q}}(n_1, n_2) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.130}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.131}$$

as we can see, 2D-gcd-delta function is linear combination of impulse train. Thus, the relation of Figure 3.2 still holds. One thing to remark is that if $\mathbf{P}$ is not cyclic, the Möbius function may not only be $0, 1, -1$. The minimum example of this is

$$\mathbf{P} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \tag{3.132}$$

which has five factors

$$\mathbf{P}_1 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \mathbf{P}_2 = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}, \mathbf{P}_3 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{P}_4 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \mathbf{P}_5 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.133}$$

As we can observe, $\mathbf{P}_2$, $\mathbf{P}_3$, $\mathbf{P}_4$ are not multiple to each other. Therefore, the A matrix like Equation (3.37) is given as

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

But $M = A^{-1}$, the Möbius matrix is

$$M = \begin{bmatrix} 1 & -1 & -1 & -1 & 2 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

It is no longer a matrix with only $0, 1, -1$.

To summarize, the 2D-gcd-delta function is still closely related to 2D impulse train by generalized Möbius function. These functions can be viewed as filterbank and applied to separate the signal into sub-period.

doi:10.6342/NTU201800930

The period detection is performed by calculating the LCM of those sub-periods. We have given concrete examples from 6×6 toy matrix, to real world image like [35]. But this method still has an issue. It assumes all sub-periods are the factor of N (1D case) or $\mathbf{N}$ (2D). We will find a way to solve the issue in the next section.

doi:10.6342/NTU201800930

Chapter 4

All phase FFT

This work is based on [36]. In this thesis, We use the concept but modify the details.

4.1 Motivation

All phase FFT (apFFT)[36] uses the phase information to determine more accurate frequency. It has been successfully used in many areas such as spectrum analysis, power management, radar singal processing and microgrid monitoring[37, 38, 39, 40, 41, 42, 43]. In the traditional DFT, if the signal length N is given, the frequencies can only be $k = 0, 1, 2, ..., N - 1$. As we have seen in Section 3.1, it will result in *critical length problem*. All phase FFT provides a method to calculate exact frequencies. The concept is simple. It uses the difference of phase to estimate the frequency. Recall that a continuous signal $x(t)$ with only one frequency can be formulated as

$$x(t) = ae^{i(2\pi ft+\phi)}$$

where $a > 0$ indicates the amplitude, f indicates the frequency and ϕ is the phase of that signal x. To calculate the frequency f, we can sample from two different time t_1 and t_2. The phase at each time $\angle x(t_j)$ is $2\pi f t_j +$

doi:10.6342/NTU201800930

ϕ, $j = 1, 2$. Therefore,

$$\angle x(t_2) - \angle x(t_1) = 2\pi f(t_2 - t_1) \tag{4.1}$$

$$\Rightarrow f = \frac{\angle x(t_2) - \angle x(t_1)}{2\pi(t_2 - t_1)} \tag{4.2}$$

Thus, f can be determined by phase. When signal is discrete, the formula is almost the same, and we can notice that this method is not integer, so the *critical length problem* can be solved.

However, there are some problems on this method:

- What if there are multiple frequencies?
- $x(t)$ might be real, e.g. $x(t) = \cos(2\pi ft)$.
- The amplitude is not determined.

We will discuss how to deal with these problem in the next section.

4.2 Definition and Properties

The discrete apFFT algorithm is implemented by the Algorithm 2. In the step 2 of the algorithm, a window function is used in order to reduce the interference of other frequencies. In step 3, we take FFT of y. It means the phase is obtained from frequency domain. Therefore, whether the signal x is real or complex, the phase can always be determined. In step 5, we reconstruct the signal with accurate frequency to determine the phase ϕ, and further, use f and ϕ to reconstruct again, to determine the amplitude. Note that in the reconstruction, the same window function must be performed, otherwise the phase and amplitude will not be accurate.

The main drawback of apFFT is that in step 4, local maximum must be obtained. This will not only increase the computation time, but also cause some mistakes. In particular, if two frequencies are very closed to each other, then local maximum finding may wrongly detect just one of them. Moreover, although all phase FFT uses window function to avoid interference or crosstalk between frequencies, the window bandwidth is the bottleneck. If the distance between two frequencies are smaller than the window bandwidth, then all phase FFT will fail. Nevertheless, in most cases, the frequencies are located sparsely, so this may not be an issue. Another property or disadvantage of all phase FFT is that the

```
Algorithm 2 All phase FFT
Input: Signal $x$ with length $N+1$, i.e. $x(n)$ with $n = 0, 1, 2, ..., N$.
Output: $a_j, f_j, \phi_j$ such that
    $$x(n) = \sum_j a_j e^{i(\frac{2\pi f_j n}{N} + \phi_j)}$$
Let $x_1(n) = x(n)$, $n = 0, 1, 2, ..., N-1$, $x_2 = x(n)$, $n = 1, 2, 3, ..., N$;
Take a window function $h(n)$, such as Hamming window or Blackman window, calculate $y_1(n) = h(n)x_1(n)$, $y_2(n) = h(n)x_2(n)$;
Take $N$ point FFT on $y_1$ and $y_2$;
Find peaks on $|\hat{y}_1(k)|$, let the index of these maximum be $k_j^*$;
For each $j$:
    $$f_j = \frac{N(\angle\hat{y}_2(k_j^*) - \angle\hat{y}_1(k_j^*))}{2\pi} \tag{4.3}$$
    $$z_j = h(n)e^{i2\pi f_j n} \tag{4.4}$$
    $$\phi_j = \angle\hat{y}_1(k_j^*) - \angle\hat{z}_j(k_j^*) \tag{4.5}$$
    $$r_j = h(n)e^{i(2\pi f_j n + \phi_j)} \tag{4.6}$$
    $$a_j = |\frac{\hat{y}_2(k_j^*)}{\hat{r}_j(k_j^*)}| \tag{4.7}$$
    return $a_j, f_j, \phi_j$
```

separated signals are not orthogonal. Recall that in standard IFFT

$$x(n) = \sum_k \hat{x}(k)W^{nk}$$

which means the signal is separated into N sub-bands. Those sub-bands W^{nk} are orthogonal to each other because k is an integer. However, if a signal is separated into

$$x(n) = 3e^{\frac{2\pi in\sqrt{2}}{N}} + e^{\frac{2\pi in\sqrt{5}}{N}}$$

one can easily check the inner product of two sub-bands will not equal to zero for any length N.

Remark: Algorithm 2 has one main difference with [36]. We use $N+1$ points instead of $2N-1$. This will reduce some accuracy because the input information is less, but the latency will also be reduced.

We now provide a concrete example to see how Algorithm 2 works. Let $N = 200$, and the input signal x is

$$x(n) = \sum_{k=1}^{3} a_k e^{i(\frac{2\pi f_k n}{N} + \phi_k)} \tag{4.8}$$

where

$$\begin{aligned} &f_1 = 12.5781, f_2 = 32.4147, f_3 = 49.377 \\ &a_1 = 2.3, a_2 = 1.7, a_3 = 4.2 \\ &\phi_1 = \frac{\pi}{3}, \phi_2 = \frac{\pi}{8}, \phi_3 = \frac{11\pi}{15} \end{aligned}$$

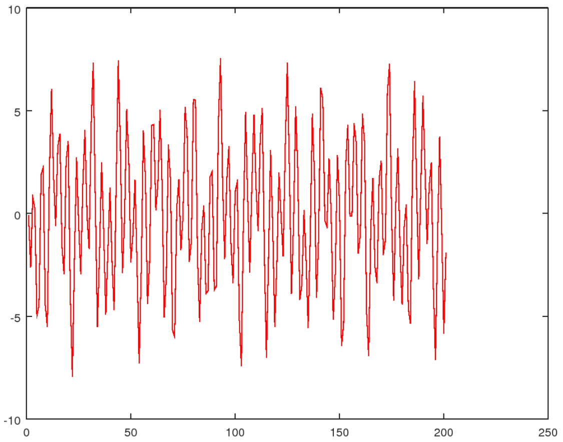

Figure 4.1: The real part of x in Equation (4.8)

doi:10.6342/NTU201800930

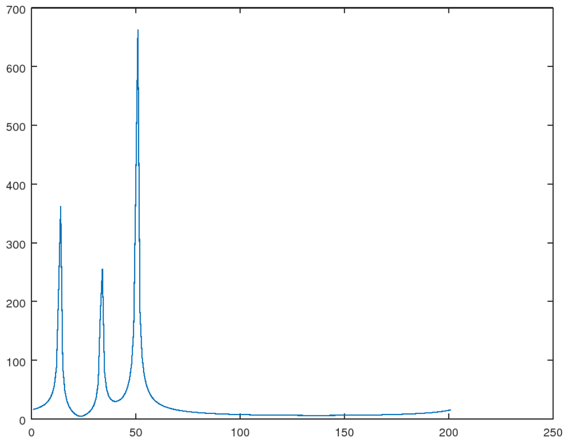

Figure 4.2: The power spectrum of x in Equation (4.8), where window is not performed.

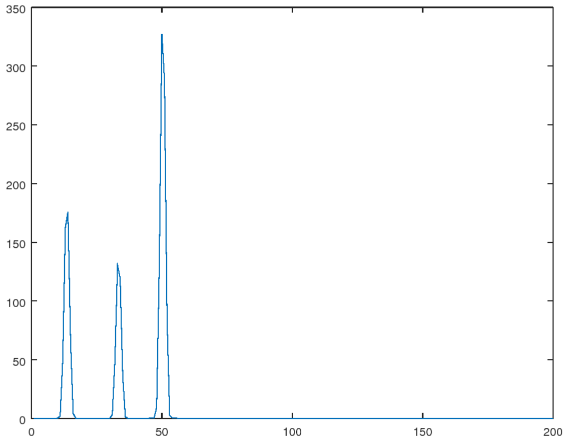

Figure 4.3: The power spectrum of x in Equation (4.8), where window is performed.

The real part of this signal is illustrated in Figure 4.1. If we perform DFT directly, as Figure 4.2, we can see the power leakages of these three signals. In Figure 4.3, where Blackman window is performed, the spectrum is simpler. This can be very helpful for the next step, peak finding. After finding the peaks, the phases at those peaks can be found.

doi:10.6342/NTU201800930

The phases of first part ($x(n), n = 0, 1, 2, ...199$) are

$$p_1 = [-0.27163, 1.68897, -2.80091]$$

And the second part ($x(n), n = 1, 1, 2, ...200$)

$$p_2 = [0.12353, 2.70731, -1.24968]$$

So the estimated frequencies are

$$(p_2 - p_1) \times \frac{N}{2\pi} = [12.578289, 32.414725, 49.377016]$$

As we can see, the errors are very small. Once the frequencies are determined, the phases and amplitudes can be calculated easily.

$$\phi = [1.0465935, 0.39259048, 2.3037725], a = [2.2998256, 1.7000168, 4.2000321]$$

All of the errors are less than 10^{-4}.

4.3 Applications of apFFT

4.3.1 Determine the proper frequency for period estimation

In this section, we will demonstrate how to use all phase FFT to enhance the period estimation. We give a toy example first. Let $N = 100$ and

$$x = [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7, ...,]$$

Note that the signal x has period 7 which is not a factor of N. After apFFT, the frequencies are

$$f = \begin{bmatrix} 0 \\ 14.285722 \\ 28.571754 \\ 42.857115 \\ 57.142885 \\ 71.428246 \\ 85.714278 \end{bmatrix}$$

Although f is not an integer vector, we can find N/f (ignore the DC term)

$$\frac{N}{f} = \begin{bmatrix} 6.9999961 \\ 3.4999601 \\ 2.3333349 \\ 1.7499991 \\ 1.4000064 \\ 1.1666668 \end{bmatrix}$$

We can notice that the first one is very close to the true period 7. In fact, the values are close to

$$\frac{N}{f} = \begin{bmatrix} 7 \\ 7/2 \\ 7/3 \\ 7/4 \\ 7/5 \\ 7/6 \end{bmatrix}$$

It means f contains many harmonic frequencies, with fundamental frequency $f_0 = 14.2857$. The integer period can be found easily by using

doi:10.6342/NTU201800930

N/f_0. Furthermore, we can use the period p to truncate the original signal. In this example, the largest number below $N = 100$ which is multiple of 7 is 98, so we can reduce the length into 98 and perform other processing. This is very useful when p is not prime number, which means there might be sub-periods in the signal. Once the length is reduced to the proper length, all methods in Chapter 3 can be applied.

Next, we provide a more realistic example. Suppose a given signal x in Figure 4.4, which is a periodic function with noise. The length $N = 483$ and the period $p = 103$. The first few frequencies apFFT obtain are

$$f = \begin{bmatrix} 4.6891 \approx \frac{N}{p} \\ 14.0696 \approx \frac{3N}{p} \\ 23.4459 \approx \frac{5N}{p} \\ 32.8236 \approx \frac{7N}{p} \\ 42.2038 \approx \frac{9N}{p} \end{bmatrix}$$

Note that in this signal, only odd harmonic exists. The reason is this signal is almost a odd signal.

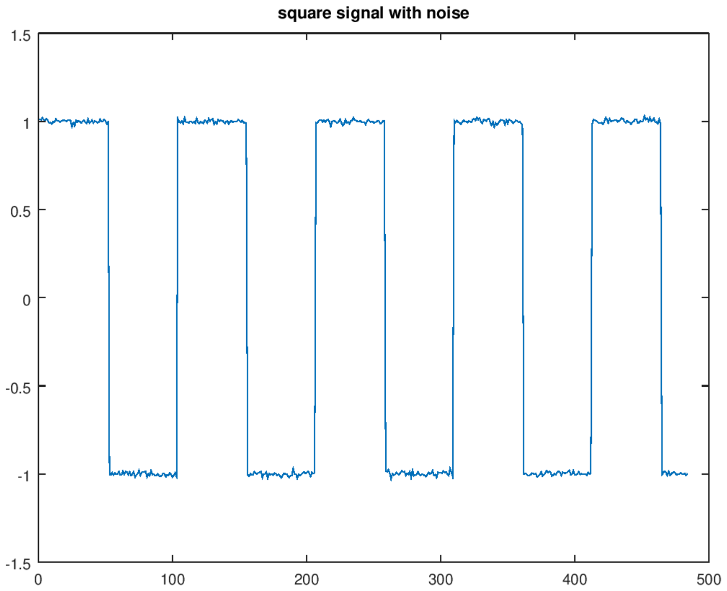

Figure 4.4: The step function with noise

doi:10.6342/NTU201800930

4.3.2 1D/2D frequency estimation

In this section we will provide a fast way to estimate the frequency from a 2D signal by using two 1D apFFT. The concept is very simple. Since the apFFT uses phase information to estimate frequency, we can use it again on another dimension. To illustrate the idea, let

$$X(n_1, n_2) = \sum_k a_k e^{i2\pi\left(\frac{f_k n_1 + h_k n_2}{N} + \phi_k\right)}$$

where f_k is the frequency on the first dimension, and h_k is on the second. Now let

$$x_1 = X(n_1, m), x_2 = X(n_1, m+1)$$

for some fixed m. And by using all phase FFT, the estimated frequencies, amplitudes and phases from x_1 are

$$[f_{x_1}, a_{x_1}, \phi_{x_1}] = [f_k, a_k, \left(\frac{2\pi h_k m}{N} + \phi_k\right)]$$

Similarly, the estimated frequencies, amplitudes and phases from x_2 are

$$[f_{x_2}, a_{x_2}, \phi_{x_2}] = [f_k, a_k, \left(\frac{2\pi h_k (m+1)}{N} + \phi_k\right)]$$

Note that f_{x_2} and a_{x_2} are actually redundant. By

$$\phi_{x_2} - \phi_{x_1} = \frac{2\pi h_k}{N}$$

We can obtain the frequencies on the second dimension. Once f_k, a_k, h_k are found, we can further calculate ϕ_k like 1D case.

The advantage of the proposed method is lower complexity. Only two rows or columns are used, so the complexity of frequency estimation is $O(N \log N)$ instead of $O(N^2 \log N)$. The drawback of this method is the assumption that f_k and h_k are sparse and not very close to each other.

Recall that all phase FFT use window function and take local maximum, and it will fail if two frequencies are near.

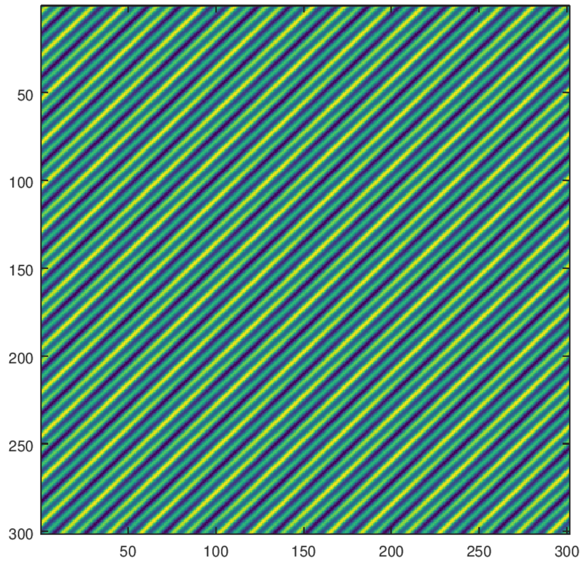

Figure 4.5: 2D signal with two frequencies

We use the following signal as example.

$$X = e^{i2\pi\left(\frac{f_1 n_1 + f_2 n_2}{N}\right)} + 2e^{i2\pi\left(\frac{f_3 n_1 + f_4 n_2}{N}\right)}$$

where

$$f_1 = 12.531; f_2 = 14.196; f_3 = 35.981; f_4 = 33.52;$$

The real part of this signal is given in Figure 4.5. Note that f_1 and f_2 can be near since they are on different dimensions. By all phase FFT, the estimated frequencies on n_1 are

$$12.53103; 35.98102$$

which is very good approximation of f_1 and f_3. The frequencies on n_2 are

$$14.20639; 33.52696$$

doi:10.6342/NTU201800930

As we can see, this result is still acceptable, but not as good as the estimations on n_1. This is not surprising because we only take two samples on this dimension. Another reason is error propagation. The phase estimation on n_1 already induce some error, and we use the result from it. Nevertheless, consider we only use 1D all phase FFT twice, instead of 2D FFT, this result might have potential on mobile device, where the computation power is limited.

4.3.3 Chirp rate tracking

In this section we use all phase FFT to trace the frequency of a chirp signal. Since the frequency is not represented by integer, the slope can be more accurate. The chirp signal we use in this section is

$$c(n) = \sum_k a_k e^{i2\pi \frac{\alpha_k n^2 + f_k n}{L}}$$

where α_k is the chirp rate and f_k is the based frequency. To trace the varying frequency, we use sliding window and select only $N+1$ points on $c(n)$. In other words,

$$x_m(n) = c(m+n)$$

where $n = 0, 1, 2, ..., N$. In practice, m is moving half of the window size N. For example, if $N = 200$, we will take $x_0, x_{100}, x_{200}, ...$ as the input. All phase FFT will apply to each x_j to get the instantaneous frequency. The time-frequency plane will look like Figure 4.6. This signal is

$$c(n) = e^{(0.01n^2+200n)\frac{i2\pi}{4000}} + 2e^{(-0.01n^2+560n)\frac{i2\pi}{4000}}$$

As we can see, the frequencies are traced well. Recall that all phase FFT already chooses local peaks, so on the time-frequency plane only two lines are shown. Other methods, such as short-time Fourier transform, will produce two lines with thin width. The drawback of this method is when two chirp come across, it will fail at the crossing area. As shown in Figure 4.7, where $\alpha_1 = 0.1$ and $\alpha_2 = -0.05$ with the same other settings, the first chirp (α_1, red) is affected when the second is near. This makes sense because of the crosstalk.

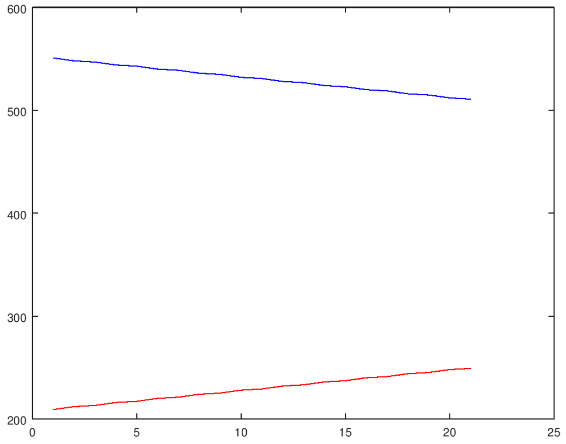

Figure 4.6: Signal with two chirp rate, represented in time-frequency plane by all phase FFT. Red:$\alpha_1 = 0.01$,Blue:$\alpha_2 = -0.01$ Details are referred to the text.

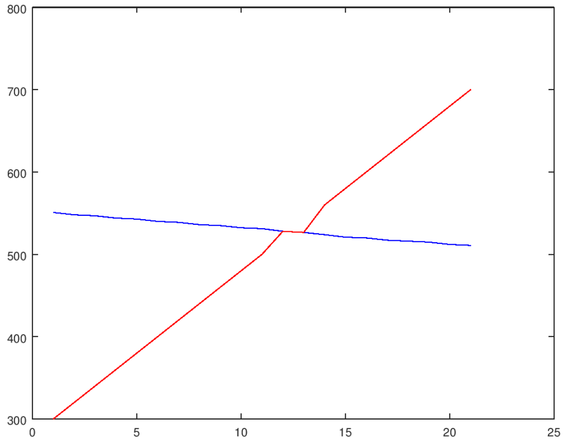

Figure 4.7: Signal with two chirp rate, represented in time-frequency plane by all phase FFT. Red:$\alpha_1 = 0.1$,Blue:$\alpha_2 = -0.05$. The red chirp is affected when blue is near.

doi:10.6342/NTU201800930

4.3.4 Sparse FFT and Signal reconstruction

This concept of sparse FFT[44, 45] can be simplified as follows:

- Detect a band where a frequency exists.
- Apply a band pass filter and estimate that frequency easily.
- Eliminate the estimated signal.
- Loop the process above until the remaining signal is too small.

We can find apFFT is very similar to sparse FFT. First of all, finding the local peaks can be viewed as frequency detection. Secondly, the frequency estimation is simple. Finally, both of them assume the frequencies are sparse and far away. Therefore, we can apply a technique from sparse FFT to further enhance all phase FFT. The trick is called coprime downsampling, which often used for fast detection. Let x be a signal with length N. p and q are two integer where $\gcd(p,q) = 1$. The downsampled signals

$$x_p(n) = x(np), x_q(n) = x(qn)$$

Note that x_p and x_q have length $\frac{N}{p}$ and $\frac{N}{q}$ respectively. Because downsampling induce aliasing, the frequency f_p calculated by x_p, might come from

$$f_t = f_p, f_p + \frac{N}{p}, f_p + \frac{2N}{p}, ...$$

where f_t is the true frequency of x. Similarly,

$$f_t = f_q, f_q + \frac{N}{q}, f_q + \frac{2N}{q}, ...$$

The intersection of the two set $\{f_p, f_p + \frac{N}{p}, f_p + \frac{2N}{p}, ...\}$ and $\{f_p, f_p + \frac{N}{p}, f_p + \frac{2N}{p}, ...\}$ is the true frequency. The motivation of this method is we can use fewer sampling points and thus, reduce the complexity. However, if $\frac{N}{p}$ is not an integer, traditional frequency finding method will wrongly estimate f_p. Luckily, all phase FFT can solve this easily. Moreover, the true frequency f_t might not be integer, either. This shows the advantage of apFFT.

To illustrate the idea, let $N = 1000$, $p = 17$ and $q = 23$. The signal is

$$x(n) = e^{i2\pi \frac{f_1 n}{N}} + 2e^{i2\pi \frac{f_2 n}{N}}$$

where $f_1 = 855.77$ and $f_2 = 310.431$. The signal x_p, which is x downsampled by p, has length $N/p \approx 59$. Similarly the length of x_q is only 43. The apFFT gives

$$\begin{aligned} f_p &= [16.313, 32.240] \\ f_q &= [6.0832, 29.6831] \end{aligned}$$

Note that

$$\begin{aligned} f_p(1) + 5\frac{N}{p} &= 16.313 + 5000/17 \approx 310.431 = f_2 \\ f_p(2) + 14\frac{N}{p} &= 32.24 + 14000/17 \approx 855.77 = f_1 \\ f_q(1) + 7\frac{N}{q} &= 6.0832 + 5000/23 \approx 310.431 = f_2 \\ f_q(2) + 19\frac{N}{p} &= 29.6831 + 19000/23 \approx 855.77 = f_1 \end{aligned}$$

As we can observe, although we just use $59 + 43 = 102$ points, which is $1/10$ of N, the frequencies can still be estimated accurately.

The main difference of sparse FFT and all phase FFT is orthogonality. The sparse FFT can detect, estimate and reconstruct the frequencies recursively. More precisely in sparse FFT, if a_k, f_k, ϕ_k is detected, we can eliminate that frequency.

$$x_r(n) = x(n) - a_k e^{i(\frac{2\pi}{N} f_k n + \phi_k)} \tag{4.9}$$

where x_r is the remaining signal. However, for all phase FFT, each f_k is not orthogonal. Eliminating one frequency may affect another. To solve this problem, we propose a band elimination method. To simplify the problem we consider a single frequency signal. Let

$$x(n) = e^{i(\frac{2\pi}{N}fn} \tag{4.10}$$

The DFT of x is

$$\begin{aligned}
\hat{x}(k) &= \sum_{k=0}^{N-1} x(n)W^{nk} \qquad (4.11)\\
&= \sum_{k=0}^{N-1} e^{i(\frac{2\pi}{N}(f-k)n} \qquad (4.12)\\
&= \frac{1-e^{i2\pi(f-k)}}{1-e^{i2\pi\frac{(f-k)}{N}}} \qquad (4.13)\\
&= e^{i\pi(f-k)\frac{N-1}{N}}\frac{\sin(\pi(f-k))}{\sin(\pi\frac{f-k}{N})} \qquad (4.14)
\end{aligned}$$

In other words, when f is determined, the frequency response of all k can be represented in closed form. Therefore, when we eliminate the frequency with non integer f, we can actually remove several k near f. For example, let $N = 1000$ and $f = 855.77$. The frequency response near 850 is illustrated in Figure 4.8. As we can see, the decay is very fast. In practice, we remove the part between 850 and 860. Since the frequencies we remove are now integer, the orthogonality is preserved.

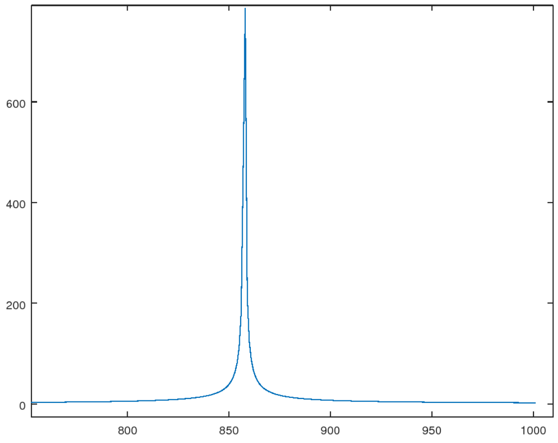

Figure 4.8: Absolute value of $\hat{x}$, where x is given in Equation (4.10), $f = 855.77$. The decay is very fast near 850.

doi:10.6342/NTU201800930

Chapter 5

Conclusion

In this thesis we review the Ramanujan's sum and propose two applications. The first application is integer zero autocorrelation (integer ZAC) sequence construction. This takes advantage of the integer property of Ramanujan's sum. The construction becomes much simpler in frequency domain. We also review and propose some other methods such as Legendre sequence and Gauss sum for prime length. Ramanujan's sum method can be viewed as complement for composite length. Therefore, integer ZAC for arbitrary length has been solved.

The second application is 1D and 2D period estimation. Two dimensional period estimation is a hard problem since it has three parameters to determine: length, width and direction, unlike 1D periodic signal which has only one period parameter. We have defined 2D Ramanujan'sum and discussed their period. The signal is then separated into sub-period signals. We also introduce 2D Least common multiplier (2D LCM) for period estimation. Concrete examples are provided.

Finally, to handle the problem that the period might not be a factor of given signal length, we review the all phase FFT. This algorithm use phase information to estimate frequency so the result can be non integer. The period can be estimate accurately by the inverse of fundemental frequency. We also propose some other applications of all

doi:10.6342/NTU201800930

phase FFT, such as chirp signal pitch tracking and fast 2D frequency estimation.

doi:10.6342/NTU201800930

Bibliography

[1] Srinivasa Ramanujan. On certain trigonometrical sums and their applications in the theory of numbers. *Trans. Cambridge Philos. Soc*, 22(13):259–276, 1918.

[2] Michel Planat, Milan Minarovjech, and Metod Saniga. Ramanujan sums analysis of long-period sequences and 1/f noise. *EPL (Europhysics Letters)*, 85(4):40005, 2009.

[3] Changchuan Yin, Xuemeng E Yin, and Jiasong Wang. A novel method for comparative analysis of DNA sequences by ramanujan-fourier transform. *Journal of Computational Biology*, 21(12):867–879, 2014.

[4] Wei Hua, Jiasong Wang, and Jian Zhao. Discrete ramanujan transform for distinguishing the protein coding regions from other regions. *Molecular and cellular probes*, 28(5):228–236, 2014.

[5] E. Carni and A. Spalvieri. Synchronous cdma based on the cyclical translations of a cazac sequence. *IEEE Transactions on Wireless Communications*, 8(3):1144–1147, March 2009.

[6] Ho-Hsuan Chang, Shieh-Chiang Lin, and Chong-Dao Lee. A cdma scheme based on perfect gaussian integer sequences. *AEU - International Journal of Electronics and Communications*, 75:70 – 81, 2017.

doi:10.6342/NTU201800930

[7] S. F. A. Shah and A. H. Tewfik. Perfectly balanced binary sequences with optimal autocorrelation. In *2007 14th IEEE International Conference on Electronics, Circuits and Systems*, pages 681–684, Dec 2007.

[8] J.J. Benedetto, I. Konstantinidis, and M. Rangaswamy. Phase-coded waveforms and their design. *Signal Processing Magazine, IEEE*, 26(1):22–31, Jan 2009.

[9] Joseph H Silverman and John Torrence Tate. *Rational points on elliptic curves*, volume 9. Springer, 1992.

[10] Neal I Koblitz. *Introduction to elliptic curves and modular forms*, volume 97. Springer Science & Business Media, 2012.

[11] Ian Blake, Gadiel Seroussi, and Nigel Smart. *Elliptic curves in cryptography*, volume 265. Cambridge university press, 1999.

[12] K. J. Chang and H. H. Chang. Perfect gaussian integer sequences of period p^k with degrees equal to or less than $k+1$. *IEEE Transactions on Communications*, 65(9):3723–3733, Sept 2017.

[13] Soo-Chang Pei and Kuo-Wei Chang. On integer-valued zero autocorrelation sequences. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, pages 1–4, Oct 2013.

[14] Soo-Chang Pei and Kuo-Wei Chang. Perfect gaussian integer sequences of arbitrary length. *Signal Processing Letters, IEEE*, 22(8):1040–1044, Aug 2015.

[15] Godfrey Harold Hardy and Edward Maitland Wright. *An introduction to the theory of numbers*. Oxford university press, 1979.

doi:10.6342/NTU201800930

[16] L-K Hua. *Introduction to number theory*. Springer Science & Business Media, 2012.

[17] MR Schroeder. Number theory in science and communications, 2nd, 1986.

[18] Soo-Chang Pei, Chia-Chang Wen, and Jian-Jiun Ding. Closed-form orthogonal dft eigenvectors generated by complete generalized legendre sequence. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55(11):3469–3479, 2008.

[19] P. P. Vaidyanathan and P. Pal. The farey-dictionary for sparse representation of periodic signals. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 360–364, May 2014.

[20] P.P. Vaidyanathan. Ramanujan Sums in the Context of Signal Processing-Part I: Fundamentals. *Signal Processing, IEEE Transactions on*, 62(16):4145–4157, Aug 2014.

[21] P.P. Vaidyanathan. Ramanujan Sums in the Context of Signal Processing-Part II: FIR Representations and Applications. *Signal Processing, IEEE Transactions on*, 62(16):4158–4172, Aug 2014.

[22] P.P. Vaidyanathan. Ramanujan-sum expansions for finite duration (fir) sequences. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 4933–4937, May 2014.

[23] S. V. Tenneti and P. P. Vaidyanathan. Nested periodic matrices and dictionaries: New signal representations for period estimation. *IEEE Transactions on Signal Processing*, 63(14):3736–3750, July 2015.

doi:10.6342/NTU201800930

[24] S. V. Tenneti and P. P. Vaidyanathan. A unified theory of union of subspaces representations for period estimation. *IEEE Transactions on Signal Processing*, 64(20):5217–5231, Oct 2016.

[25] S. C. Pei and K. S. Lu. Intrinsic integer-periodic functions for discrete periodicity detection. *IEEE Signal Processing Letters*, 22(8):1108–1112, Aug 2015.

[26] S. Deng and J. Han. Signal periodic decomposition with conjugate subspaces. *IEEE Transactions on Signal Processing*, PP(99):1–1, 2016.

[27] S. V. Tenneti and P. P. Vaidyanathan. Critical data length for period estimation. In *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1226–1229, May 2016.

[28] P. P. Vaidyanathan and S. Tenneti. Efficient multiplier-less structures for ramanujan filter banks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6458–6462, March 2017.

[29] J. Cassels. *An Introduction to the Geometry of Numbers*. Springer-Verlag, 1971.

[30] J. Kovacevic and M. Vetterli. The commutativity of up/downsampling in two dimensions. *IEEE Transactions on Information Theory*, 37(3):695–698, May 1991.

[31] Soo-Chang Pei and Kuo-Wei Chang. Integer 2-d discrete fourier transform pairs and eigenvectors using ramanujan¡¦s sum. *IEEE Signal Processing Letters*, 23(1):70–74, 2016.

[32] Soo-Chang Pei and Kuo-Wei Chang. Two-dimensional period estimation by ramanujan’s sum. *IEEE Transactions on Signal Processing*, 65(19):5108–5120, 2017.

[33] Mario Hampejs, Nicki Holighaus, László Tóth, and Christoph Wiesmeyr. Representing and counting the subgroups of the group $Z_m \times Z_n$. *Journal of Numbers*, 2014.

[34] Henri Cohen. *A Course in Computational Algebraic Number Theory*. New York: Springer-Verlag, 1993.

[35] M.C. Escher. Pegasus, 1959.

[36] Huang Xiaohong, Wang Zhaohua, and Chou Guoqiang. New method of estimation of phase, amplitude, and frequency based on all phase fft spectrum analysis. In *2007 International Symposium on Intelligent Signal Processing and Communication Systems*, pages 284–287, Nov 2007.

[37] X. Lu and Y. Zhang. Phase detection algorithm and precision analysis based on all phase fft. In *International Conference on Automatic Control and Artificial Intelligence (ACAI 2012)*, pages 1564–1567, March 2012.

[38] Z. Yonghui, C. Xi, and Z. Xiyuan. Power harmonic analysis based on all-phase fft1. In *2010 2nd International Conference on Signal Processing Systems*, volume 3, pages V3–576–V3–579, July 2010.

[39] D. Zhang and P. Le. Application of modified all-phase fft to extract signals in microgrid for condition monitoring and control purposes. In *2015 Australasian Universities Power Engineering Conference (AUPEC)*, pages 1–5, Sept 2015.

[40] Si Wei Tan, Zhi Liang Ren, and Jiong Sun. Parameter estimation of low-frequency signal with negative frequency contribution. In *Vehicle, Mechatronics and Information Technologies*, volume 380 of *Applied Mechanics and Materials*, pages 3457–3460. Trans Tech Publications, 11 2013.

doi:10.6342/NTU201800930

[41] A. Boughambouz, A. Bellabas, B. Magaz, T. Menni, and M. E. M. Abdelaziz. Improvement of radar signal phase extraction using all phase fft spectrum analysis. In *2017 Seminar on Detection Systems Architectures and Technologies (DAT)*, pages 1–4, Feb 2017.

[42] G. Yang, J. Tian, S. Wu, and S. L. Wu. A novel frequency discrimination method based on all phase fft for anti-towed decoy. In *IET International Radar Conference 2015*, pages 1–7, Oct 2015.

[43] W. Lv, C. Shen, F. Gui, Z. Tian, and D. Jiang. Real-time spectrum analyzer based on all phase fft spectrum analysis. In *2013 Fourth International Conference on Digital Manufacturing Automation*, pages 966–969, June 2013.

[44] Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Simple and practical algorithm for sparse fourier transform. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1183–1194. Society for Industrial and Applied Mathematics, 2012.

[45] S. H. Hsieh, C. S. Lu, and S. C. Pei. Sparse fast fourier transform by downsampling. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5637–5641, May 2013.

doi:10.6342/NTU201800930