Tutorial of One Dimensional Discrete Fourier Transform (DFT): Theory, Implementation and MATLAB® programming

Kornkamol Thakulsukanant*and Vorapoj Patanavijit

Assumption University

Abstract

The Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT) are classical approaches to mathematically model signals and systems in the frequency and spatial (or temporal) domains, respectively. Due to worldwide implementation of Digital Signal Processing (DSP) during the last two decades, Discrete Fourier analysis has become one of the most useful mathematical techniques for analyzing digital signals and systems. Consequently, this article provides a tutorial for the Discrete Fourier Transform (DFT) on 1-dimensional (1-D) signals employing MATLAB®. While the Discrete Fourier analysis provides information for both spatial and frequency domains, this paper focuses on the frequency domain of the discrete signal.

Keywords: Discrete Fourier Transform (DFT), Inverse Discrete Fourier Transform (IDFT), Digital Signal Processing (DSP).

บทคัดย่อ

ผลการแปลงฟูเรียร์แบบวิยุตและผลการแปลงผกผันฟูเรียร์แบบวิยุตเป็นวิธีการคำนวณทางคณิตศาสตร์พื้นฐานสำหรับการสร้างแบบจำลองทางคณิตศาสตร์ของสัญญาณและระบบที่มีความไม่ต่อเนื่องเชิงเวลาเพื่อวิเคราะห์ทั้งในเชิงเวลาและความถี่ เนื่องจากการประยุกต์ใช้งานด้านการประมวลผลสัญญาณดิจิทัล (DSP) ในช่วงยี่สิบปีที่ผ่านมามีการเจริญเติบโตอย่างมาก ดังนั้นการวิเคราะห์โดยผลการแปลงฟูเรียร์แบบวิยุตจึงกลายเป็นเทคนิคทางคณิตศาสตร์ที่มีประโยชน์อย่างมากสำหรับการวิเคราะห์สัญญาณดิจิทัลและระบบดิจิคอล บทความนี้จึงนำเสนอหลักการและแนวคิดเชิงคณิตศาสตร์ผลการแปลงฟูเรียร์แบบวิยุตสำหรับสัญญาณหรือระบบแบบหนึ่งมิติและยังนำเสนอตัวอย่างการคำนวณการแปลงฟูเรียร์โดยใช้โปรแกรม MATLAB® เพื่อให้ผู้อ่านสามารถวิเคราะห์สัญญาณที่มีความไม่ต่อเนื่องทั้งในเชิงเวลาและความถี่

คำสำคัญ: ผลการแปลงฟูเรียร์แบบวิยุต, ผลการแปลงผกผันฟูเรียร์แบบวิยุต, การประมวลผลสัญญาณดิจิทัล

1. Introduction

1.1 History of Fourier Analysis

Fourier analysis was created and published by a French mathematician, Jean Baptiste Fourier, in 1822. Fourier analysis is composed of Fourier Series (FS) and Fourier Transform (FT) elements. FS is used as a mathematical model to represent all periodic signals in the frequency domain. Like FS, the FT can also be applied for periodic signals. However, when examining non-periodic signals FS is not appropriate, and FT must be used.

Three main groups can be used to classify the majority of the signals. These are analog or continuous-time signals, discrete-time signals, and digital or discrete signals. In the first group both time and amplitude are continuous. In the second group only amplitude is discrete and time remains continuous (e.g. sampling of analog signals). In the last group

* ผู้ประสานงานหลัก (Corresponding Author)
e-mail: kthakulsukanant@yahoo.com

both time and amplitude are discrete. Fourier analysis is an essential approach to transform all spatial domain signals into frequency domain signals by employing either FS or FT. In an inverse manner, the frequency domain signals can be used to reconstruct the spatial domain signal by utilizing the Inverse Fourier Transform (IFT).

1.2 Application of 1-D DFT

Fourier analysis is necessary in several applications areas as follows:

Telecommunication:

Various modulation techniques are employed to improve transmission performance such as saving transmission bandwidth and power and reducing noise interference. For examples, Amplitude Modulation (AM), Frequency Modulation (FM), and Phase Modulation (PM), are utilized in analog signal and Amplitude-Shift Keying, On-Off Keying, Quadrature Amplitude Modulation, Frequency-Shift Keying (FSK) and Phase-Shift Keying (PSK) are utilized in digital signals [Haykin and Moher, 2003; Haykin and Moher, 2007].

Digital signal processing (DSP):

Analog/digital filters such as Butterworth filters, Chebyshev filters, Elliptic continuous-time filters, and Finite Impulse Response (FIR) filters are employed to improve noise performance of the signals [Ingle, V. K & Proakis, J. G., 2007; A.V. Oppenheim & R.W. Schafer, 2009]. The first three are analog filters and the last one is the digital filter.

Circuit analysis:

Phasor analysis, frequency response analysis, and steady state response analysis are employed for determining and analyzing the signals properly.

From the modern research prospective, although the DFT is one of the classical mathematical tools for developing the research in both science and engineering for the last two decades, the modern advance algorithm techniques, which have been proposed during the last five years, are based on the framework of DFT. Various Advanced and Modern Codes and Modulations are based on DFT such as Bose-Chaudhuri-Hocquenghem (BCH) Codes [Vaezi, M. & Labeau, F., 2013], Orthogonal Frequency-Division Multiplexing (OFDM) Modulations [Wu, T-S. & Chung, C-D., 2014; Li, F., Li, X. and Yu, J., 2015], Quantized DFT Codes [Vaezi, M. & Labeau, F., 2014], Channel Estimation Techniques [Xiong, X., Jiang, B., Gao, X. and You, X., 2014], Digital Signal Processing such as analog and digital filters design/analysis [A.V. Oppenheim & R.W. Schafer, 2009], Filter Implementation [Edussooriya, C.U.S., Bruton, L.T., Agathoklis, P. and Gunaratne, T.K., 2013], Finite Impulse Response (FIR) filter [Edussooriya, C.U.S., Bruton, L.T., Agathoklis, P. and Gunaratne, T.K., 2013; Kamwa, I., Samantaray, S.R. and Joos, G., 2014], Signal Modeling and Compressive Sensing (CS) [Hu, L., Zhou, J., Shi, Z. and Fu, Q., 2013; Ha, P. H., Lee, W. and Patanavijit, V., 2014; Patanavijit, V. & Ha, P.H., 2013] and Signal Registration [Patanavijit, V., 2011].

In this article, DFT is emphasized to represent the frequency domain of discrete-time signals on 1-D [Haykin and Moher, 2003; Haykin and Moher, 2007; Ingle and Proakis, 2007; and Phillips *et al*., 2003; Schilling and Harris, 2005]. Moreover, this paper provides an analysis of DFT application as opposed to basic theory. Therefore, several examples on 1-D signals are demonstrated. This segment has been organized as follows: section 2 presents a theoretical of 1-D DFT and examples, and section 3 provides a conclusion.

2. Discrete Fourier Transform (DFT)

This section is subdivided into four subsections: section 2.1 presents the theoretical basic for DFT, section 2.2 and 2.3 provide synthetic cases for small and large sample numbers (n), respectively, and section 2.4 provides the real-world examples of signals.

2.1 Theory

This section briefly presents DFT and IDFT of 1-D signals. The algorithm of Fast Fourier Transform (FFT) [Duhamel and Vetterli, 1999; Phillips et al., 2003; Schilling and Harris, 2005] is employed for DFT and IDFT. The DFT (Discrete Fourier Transform) can be considered as an exceptional case of the DTFT (Discrete Time Fourier Transform), which is theoretically defined as

$$X(f)=\sum_{n=0}^{\infty}x(n)e^{(-jn2\pi fT)} \quad ,-\frac{f_s}{2}<f\leq\frac{f_s}{2} \tag{1}$$

The DTFT is a crucial mathematical tool, finding common application for discrete-time signals and systems. However, the direct computation of $X(f)$ requires an infinite number of floating-point operators (FLOPS). In another point of view, f is a continuous variable therefore $X(f)$ is comprised of an infinite number of data. Consequently, the DTFT is not practical for computer calculation.

For the discrete time signals $x(n)$ with finite duration ($\lim_{n\to\infty} x(n)=0$), the DTFT can be approximated as:

$$X(f)=\sum_{n=0}^{N-1}x(n)e^{(-jn2\pi fT)} \tag{2}$$

Therefore, we can interpret the DFT (Discrete Fourier Transform) as the set of N discrete values of DTFT $X(f)$. The mathematical representation for both DFT and IDFT are provided below

$$X(k)=\sum_{n=0}^{M-1}x(n)e^{-j2\pi kn/M} \tag{3}$$

$$x(n)=\frac{1}{M}\sum_{k=0}^{M-1}X(k)e^{j2\pi kn/M} \tag{4}$$

where $X(k)$ is a discrete function of the frequency variable, $x(n)$ is a finite discrete-time sequence, k and n are the integer numbers (0, 1, 2, …, M-1), M is a number of the selected samples to represent the discrete-time signal, and $j=\sqrt{-1}$ (in engineering application i is usually defined an electric current, thus to avoid confusion of notation this paper defines $j=\sqrt{-1}$).

2.2 Synthetic Cases for Small Sample Number (*n*)

The examples for 1-D signals based on small sample number, n, are illustrated in this section. Both mathematical and programmable calculations are provided in individual detailed steps so the reader can understand the concepts of DFT and IDFT using both methods.

Example 1: Unit step signal, $x1(n)=[0,0,1,1]$ with n = 4 samples.

Mathematical Calculation

This unit step signal, $x1(n)=[0,0,1,1]$, can be separated into four individual values as follows: $x1(0)=[0]$, $x1(1)=[0]$, $x1(2)=[1]$, $x1(3)=[1]$. In this example, M and k parameters equal to 4. Therefore, from Eq. (3) substitutes M = 4, we obtain

$$X1(k)=\sum_{n=0}^{3}\left(x1(n)e^{-j2\pi kn/4}\right)$$

$$= \sum_{n=0}^{3} \left(x1(n) e^{-j(\pi/2)kn} \right) \qquad (5)$$

Applying Euler's relation given below,

$$e^{\pm j\theta} = \cos\theta \pm j\sin\theta \qquad (6)$$

Thus, the exponential term in Eq. (5) becomes,

$$e^{-j(\pi/2)} = \cos(\pi/2) - j\sin(\pi/2) = -j$$

Substituting this result into Eq. (5), we obtain

$$X1(k) = \sum_{n=0}^{3} \left(x1(n)\left(-j\right)^{kn} \right)$$

Now $X1(k)$ on the above equation can be evaluated term by term as follows:

$k = 0;$
$$X1(0) = \sum_{n=0}^{3} \left(x1(n)\left(-j\right)^{0} \right) = \sum_{n=0}^{3} \left(x1(n) \right)$$
$$X1(0) = x1(0) + x1(1) + x1(2) + x1(3) = (0) + (0) + (1) + (1) = 2$$

$k = 1;$
$$X1(1) = \sum_{n=0}^{3} \left(x1(n)\left(-j\right)^{n} \right)$$
$$X1(1) = x1(0)(-j)^0 + x1(1)(-j)^1 + x1(2)(-j)^2 + x1(3)(-j)^3$$
$$X1(1) = (0)(1) + (0)(-j) + (1)(-1) + (1)(j)$$
$$X1(1) = 0 + 0 - 1 + j = -1 + j$$

$k = 2;$
$$X1(2) = \sum_{n=0}^{3} \left(x1(n)\left(-j\right)^{2n} \right)$$
$$X1(2) = x1(0)(-j)^0 + x1(1)(-j)^2 + x1(2)(-j)^4 + x1(3)(-j)^6$$
$$X1(2) = (0)(1) + (0)(-1) + (1)(1) + (1)(-1) = 0 + 0 + 1 - 1 = 0$$

$k = 3;$
$$X1(3) = \sum_{n=0}^{3} \left(x1(n)\left(-j\right)^{3n} \right)$$
$$X1(3) = x1(0)(-j)^0 + x1(1)(-j)^3 + x1(2)(-j)^6 + x1(3)(-j)^9$$
$$X1(3) = (0)(1) + (0)(j) + (1)(-1) + (1)(-j)$$
$$X1(3) = 0 + 0 - 1 - j = -1 - j$$

Thus, $X1(k)$ is $X1(k) = [2, -1 + j, 0, -1 - j]$. These values are in a rectangular co-ordinate form. However, $X1(k)$ can also be represented in a polar form as shown below,

$$X1(k) = 2\angle 0rad, 1.414\angle 0.75\pi rad, 0\angle 0rad, 1.414\angle -75\pi rad$$
$$= 2\angle 0rad, 1.414\angle 2.36rad, 0rad, 1.414\angle -2.36rad$$

All values of both rectangular co-ordinate and polar forms are contained in Table 1 below. The magnitude ($|X1(k)|$) and phase spectrums are illustrated in Figures 1 (c) and (e), respectively.

Programmable Calculation

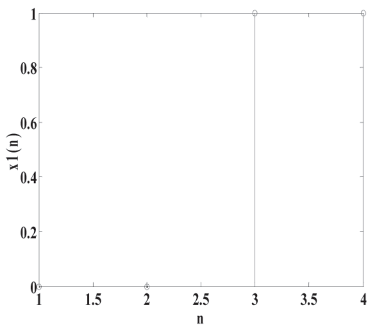
A unit step signal, $x1(n)$, with only 4 samples, is shown in Figure 1 (a). Figure 1 (b) provides a program for this signal. From the given program, two samples of zero and one are created as a unit step signal using *zeros*( ) and *ones*( ) functions, respectively. Placing *ones*( ) function after *zeros*( ) function indicates that the first sample of 'one' takes place next to the last sample of 'zero' and vice versa. A *stem*( ) function is employed to plot this $x1(n)$ signal in a discrete form. The functions of *xlabel*( ) and *ylabel*( ) are utilized to insert x- and y-labels, respectively.

Figures 1 (c) and (d) provide a magnitude spectrum (|*X*1(*k*)|) and its corresponding program, respectively. According to the program in Figure 1(d), a DFT of *x*1(*n*) signal (*X*1(*k*)) is computed using a function of *fft*( ). A function of *abs*( ) is employed to determine the absolute (magnitude) value of *X*1(*k*). Plotting and inserting x- and y-labels are completed utilizing *stem*( ), *xlabel*( ), and *ylabel*( ), respectively.

Table 1 reveals all values of both the time domain (*x*1(*n*)) and the frequency domain (*X*1(*k*)). A phase spectrum of *X*1(*k*) is demonstrated in Figure 1(e) using a program given in Figure 1 (f). Phase spectrum of *X*1(*k*) is determined using a function of *phase*( ). The last three functions mentioned in the previous paragraph are employed to complete the plot and insert the labels.

Table 1 Values of *x*1(*n*) in the spatial domain and *X*1(*k*) in the frequency domain.

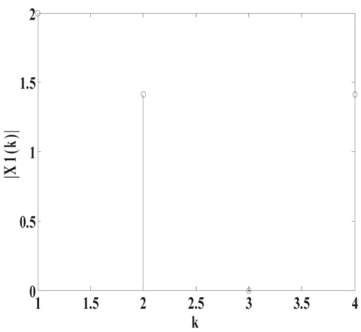
n/*k*	*x*1(*n*)	*X*1(*k*) (co-ordinate)	*X*1(*k*) (Polar)
1	0	2	$2\angle 0$ rad
2	0	-1+*j*	$1.414\angle 2.36$ rad
3	1	0	$0\angle 0$ rad
4	1	-1-*j*	$1.414\angle -2.36$ rad




```
>> x1 = [zeros(1,2) ones(1,2)];
% Create unit step signal x1(n)
>> stem(x1) % plot unit step signal
>> xlabel('n'); % Insert x-label
>>  ylabel('x1(n)');  %  Insert  y-
label
```

(a) (b)

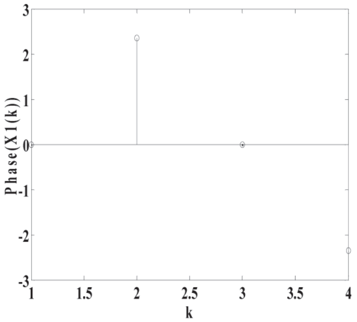
Figure 1 (a) A unit step signal, *x*1(*n*), in the spatial domain and (b) MATLAB® program for unit step signal, *x*1(*n*).




```
>> X1  = fft(x1); % Compute DFT  of
x1(n)( X1(k))
>> absX1 = abs(X1); % Determine all
absolute values of X1(k)
>> stem(absX1)% Plot |X1(k)|
>> xlabel('k'); % Insert x-label
>>  ylabel('|X1(k)|');  %  Insert  y-
label
```

(c) (d)

Figure 1 (c) The magnitude spectrum, |*X*1(*k*)| and (d) a MATLAB® program for magnitude spectrum, |*X*1(*k*)|.




```
>> PX1 = phase(X1); % Determine
phase of X1(k)
>> stem(PX1)% Plot phase of X1(k)
>> xlabel('k');% Insert x-label
>> ylabel('Phase(X1(k))');% Insert
y-label
```

(e) (f)

Figure 1 (e) The phase spectrum of $X1(k)$ and (f) a MATLAB® program for phase spectrum of $X1(k)$.

Example 2: Alternative +1 and -1 signal, $x2(n)=\left[1,-1,1,-1,1,-1\right]$ with $n = 6$ samples.

Mathematical Calculation

The signal of $x2(n)=\left[1,-1,1,-1,1,-1\right]$ can be separated into six individual values as follows: $x2(0)=\left[1\right]$, $x2(1)=\left[-1\right]$, $x2(2)=\left[1\right]$, $x2(3)=\left[-1\right]$, $x2(4)=\left[1\right]$, $x2(5)=\left[-1\right]$. M and k parameters equal to 6, and thus, Eq. (3) becomes

$$X2(k)=\sum_{n=0}^{5}\left(x2(n)e^{-j2\pi kn/6}\right)$$
$$=\sum_{n=0}^{5}\left(x2(n)e^{-j(\pi/3)kn}\right) \qquad (7)$$

Thus, from Eq. (7), $X2(k)$ can be evaluated utilizing Euler's relation in Eq. (6), as shown below:

$k = 0;$

$$X2(0)=\sum_{n=0}^{5}\left(x2(n)e^{0}\right)=\sum_{n=0}^{5}\left(x2(n)\right)$$
$$X2(0)=x2(0)+x2(1)+x2(2)+x2(3)+x2(4)+x2(5)$$
$$X2(0)=1-1+1-1+1-1=0$$

$k = 1;$

$$X2(1)=\sum_{n=0}^{5}\left(x2(n)e^{-j(\pi/3)n}\right)$$

$$X2(1)=\begin{cases} x2(0)e^{0}+x2(1)e^{-j(\pi/3)}+x2(2)e^{-j(2\pi/3)} \\ +x2(3)e^{-j\pi}+x2(4)e^{-j(4\pi/3)}+x2(5)e^{-j(5\pi/3)} \end{cases}$$

$$X2(1)=\begin{cases} (1)(1)+(-1)\left[\cos(\pi/3)-j\sin(\pi/3)\right]+(1)\left[\cos(2\pi/3)-j\sin(2\pi/3)\right] \\ +(-1)\left[\cos(\pi)-j\sin(\pi)\right]+(1)\left[\cos(4\pi/3)-j\sin(4\pi/3)\right] \\ +(-1)\left[\cos(5\pi/3)-j\sin(5\pi/3)\right] \end{cases}$$

$$X2(1)=\begin{cases} 1+(-1)\left[0.5-j0.866\right]+\left[-0.5-j0.866\right]+(-1)(-1) \\ +\left[-0.5-j(-0.866)\right]+(-1)\left[0.5-j(-0.866)\right] \end{cases}=0$$

$k = 2;$

$$X2(2)=\sum_{n=0}^{5}\left(x2(n)e^{-j(2\pi/3)n}\right)$$

$$X2(2)=\begin{cases} x2(0)e^{0}+x2(1)e^{-j(2\pi/3)}+x2(2)e^{-j(4\pi/3)} \\ +x2(3)e^{-j2\pi}+x2(4)e^{-j(8\pi/3)}+x2(5)e^{-j(10\pi/3)} \end{cases}$$

$$X2(2)=\begin{cases} (1)(1)+(-1)\left[\cos(2\pi/3)-j\sin(2\pi/3)\right]+(1)\left[\cos(4\pi/3)-j\sin(4\pi/3)\right] \\ +(-1)\left[\cos(2\pi)-j\sin(2\pi)\right]+(1)\left[\cos(8\pi/3)-j\sin(8\pi/3)\right] \\ +(-1)\left[\cos(10\pi/3)-j\sin(10\pi/3)\right] \end{cases}$$

$$X2(2)=\begin{cases} 1+(-1)\left[-0.5-j0.866\right]+\left[-0.5-j(-0.866)\right]+(-1)(1) \\ +\left[-0.5-j0.866\right]+(-1)\left[-0.5-j(-0.866)\right] \end{cases}=0$$

$k=3;$ $$X2(3)=\sum_{n=0}^{5}\left(x2(n)e^{-j(n\pi)}\right)$$

$$X2(3)=\begin{cases} x2(0)e^{0}+x2(1)e^{-j(\pi)}+x2(2)e^{-j(2\pi)} \\ +x2(3)e^{-j3\pi}+x2(4)e^{-j(4\pi)}+x2(5)e^{-j(5\pi)} \end{cases}$$

$$X2(3)=\begin{cases} (1)(1)+(-1)\left[\cos(\pi)-j\sin(\pi)\right]+(1)\left[\cos(2\pi)-j\sin(2\pi)\right] \\ +(-1)\left[\cos(3\pi)-j\sin(3\pi)\right]+(1)\left[\cos(4\pi)-j\sin(4\pi)\right] \\ +(-1)\left[\cos(5\pi)-j\sin(5\pi)\right] \end{cases}$$

$$X2(3)=1+1+1+1+1+1=6$$

$k=4;$ $$X2(4)=\sum_{n=0}^{5}\left(x2(n)e^{-j(4\pi/3)n}\right)$$

$$X2(4)=\begin{cases} x2(0)e^{0}+x2(1)e^{-j(4\pi/3)}+x2(2)e^{-j(8\pi/3)} \\ +x2(3)e^{-j4\pi}+x2(4)e^{-j(16\pi/3)}+x2(5)e^{-j(20\pi/3)} \end{cases}$$

$$X2(4)=\begin{cases} (1)(1)+(-1)\left[\cos(4\pi/3)-j\sin(4\pi/3)\right]+(1)\left[\cos(8\pi/3)-j\sin(8\pi/3)\right] \\ +(-1)\left[\cos(4\pi)-j\sin(4\pi)\right]+(1)\left[\cos(16\pi/3)-j\sin(16\pi/3)\right] \\ +(-1)\left[\cos(20\pi/3)-j\sin(20\pi/3)\right] \end{cases}$$

$$X2(4)=\begin{cases} 1+(-1)\left[-0.5-j(-0.866)\right]+\left[-0.5-j0.866\right]+(-1)(1) \\ +\left[-0.5-j(-0.866)\right]+(-1)\left[-0.5-j0.866\right] \end{cases}=0$$

$k=5;$ $$X2(5)=\sum_{n=0}^{5}\left(x2(n)e^{-j(5\pi/3)n}\right)$$

$$X2(5)=\begin{cases} x2(0)e^{0}+x2(1)e^{-j(5\pi/3)}+x2(2)e^{-j(10\pi/3)} \\ +x2(3)e^{-j5\pi}+x2(4)e^{-j(20\pi/3)}+x2(5)e^{-j(25\pi/3)} \end{cases}$$

$$X2(5)=\begin{cases} (1)(1)+(-1)\left[\cos(5\pi/3)-j\sin(5\pi/3)\right]+(1)\left[\cos(10\pi/3)-j\sin(10\pi/3)\right] \\ +(-1)\left[\cos(5\pi)-j\sin(5\pi)\right]+(1)\left[\cos(20\pi/3)-j\sin(20\pi/3)\right] \\ +(-1)\left[\cos(25\pi/3)-j\sin(25\pi/3)\right] \end{cases}$$

$$X2(5)=\begin{cases} 1+(-1)\left[0.5-j(-0.866)\right]+\left[-0.5-j(-0.866)\right]+(-1)(-1) \\ +\left[-0.5-j0.866\right]+(-1)\left[0.5-j0.866\right] \end{cases}=0$$

Thus, $X2(k)$ is $X2(k) = [0,0,0,6,0,0]$. These values are in a rectangular co-ordinate form. However, $X2(k)$ can also be represented in a polar form as shown below,

$$X2(k) = 0\angle 0 rad, 0\angle 0\pi rad, 0\angle 0 rad, 6\angle 0 rad, 0\angle 0\pi rad, 0\angle 0 rad$$

All values of both rectangular co-ordinate and polar forms are contained in Table 2 below. The magnitude ($|X2(k)|$) and phase spectrums are plotted as shown in Figures 2 (c) and (e), respectively.

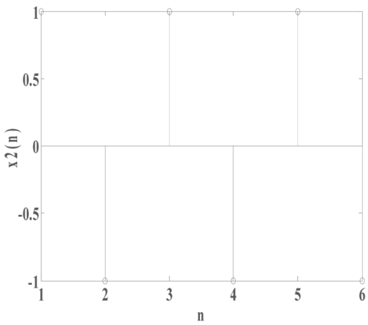
Programmable Calculation

Figure 2 (a) reveals an alternative +1 and -1 signal, $x2(n)$, with 6 samples taken. A program for constructing this signal is given in Figure 2 (b). $X2(k)$ is a DFT of this signal and its magnitude spectrum ($|X2(k)|$) is illustrated in Figure 2 (c). The program shown in Figure 2 (d) is employed to compute $X2(k)$ and plot this magnitude spectrum ($|X2(k)|$).

Table 2 provides all the values of $x2(n)$ and its corresponding DFT ($X2(k)$). A phase spectrum of $X2(k)$ is determined and plotted in Figure 2 (e) using the program provided in Figure 2 (f).

Table 2 Values of $x2(n)$ in the spatial domain and $X2(k)$ in the frequency domain.

n/*k*	*x2*(*n*)	*X2*(*k*) (co-ordinate)	*X2*(*k*) (Polar)
1	1	0	$0\angle 0$ rad
2	-1	0	$0\angle 0$ rad
3	1	0	$0\angle 0$ rad
4	-1	6	$6\angle 0$ rad
5	1	0	$0\angle 0$ rad
6	-1	0	$0\angle 0$ rad

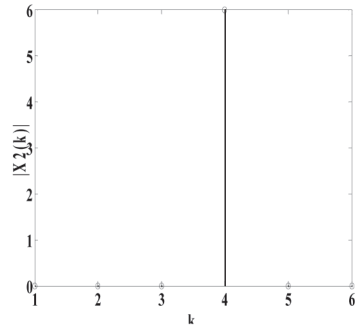


(a)

```
>> x2 =[1 -1 1 -1 1 -1]; % Create
alternative +1 and -1 signal x2(n)
>> stem(x2)
>> xlabel('n');
>> ylabel('x2(n)');
```

(b)

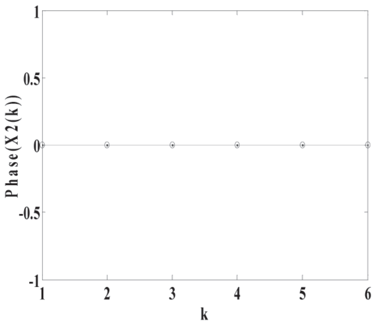
Figure 2 (a) An altenative +1 and -1 signals, $x2(n)$, in the spatial domain and (b) a MATLAB® program for alternative +1 and -1 signals, $x2(n)$.




```
>> X2 = fft(x2);
>> absX2 = abs(X2);
>> stem(absX2)
>> xlabel('k');
>> ylabel('|X2(k)|');
```

(c) (d)

Figure 2 (c) The magnitude spectrum, $|X2(k)|$ and (d) a MATLAB® program for the magnitude spectrum, $|X2(k)|$.




```
>> PX2 = phase(X2);
>> stem(PX2)
>> xlabel('k');
>> ylabel('Phase(X2(k))');
```

(e) (f)

Figure 2 (e) The phase spectrum of $X2(k)$ and (f) a MATLAB® program for the phase spectrum of $X2(k)$.

2.3 Synthetic Cases for Large Sample Number (n)

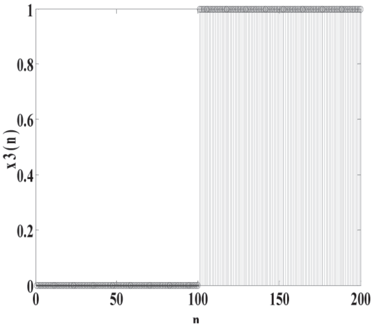
Two examples for 1-D signals, when n is large, are provided as follows:

Example 1: Unit step signal, $x3(n)$, with $n = 200$ samples.

In Figure 3 (a), in the first range of n samples (n =1-100), the values of $x3(n)$ are zero and in the rest of the samples (n = 101-200), the values of $x3(n)$ are one. A program for generating this unit step function is given in Figure 3 (b).

According to the given program, *zeros*( ) and *ones*( ) functions are used to create the unit step signal. A pair of numbers inside the parenthesis (i.e. (1,100)) of both functions, indicates how many samples of zeros and/or ones are taken (i.e. in this example, it is 100 samples). Thus, the first line of the program creates 100 samples of both zeros and ones, consecutively. A function of *stem*( ) is employed to plot a signal in a discrete form. The x- and y- labels are inserted by employing the functions of *xlabel*( ) and *ylabel*( ), respectively.

Figure 3 (c) demonstrates the magnitude spectrum ($|X3(k)|$) of unit step signal $x3(n)$. A program for creating this $|X3(k)|$ is revealed in Figure 3 (d). *fft*( ) function is used to compute a DFT of unit step signal ($X3(k)$). The magnitude spectrum of $x3(n)$ ($|X3(k)|$) is found by determining the absolute values of $X3(k)$ and dividing these data by a maximum value of $X3(k)$ (normalizing data). The functions of *abs*( ) and *max*((*max*( )) are used to determine the absolute and the maximum values of $X3(k)$, respectively. All data, x- and y- labels are plotted and inserted using the functions of *stem*( ), *xlabel*( ), and *ylabel*( ), respectively.

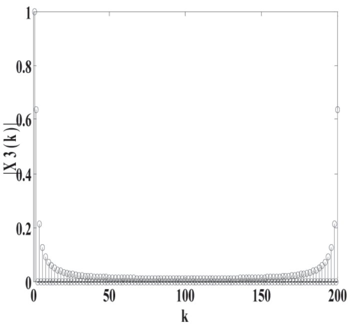



```
>> x3 = [zeros(1,100) ones(1,100)];
% Create unit step signal x3(n)
>> stem(x3)
>> xlabel('n');
>> ylabel('x3(n)');
```

(a) (b)

Figure 3 (a) A unit step signal, $x3(n)$, in the spatial domain and (b) a MATLAB® program for unit step signal, $x3(n)$.

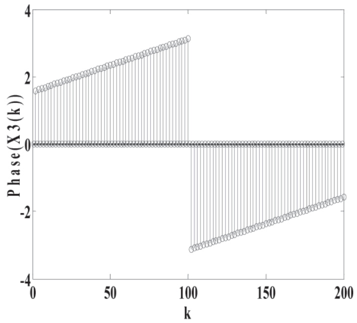
Phase of $X3(k)$ can be generated by using a program in Figure 3 (f). According to the program, phase of $X3(k)$ is found by utilizing the function of *phase*( ). Phase plot and labels are completed by employing the last three functions as mentioned in previous paragraph. Figure 3(e) shows this phase plot.




```
>> X3 = fft(x3);
>> absX3 = abs(X3); % Determine all
absolute values of X3(k)
>>   X3max   =   max(max(absX3));%
Calculate max. value of |X3(k)|
>>   magX3   =   (absX3/X3max);   %
normalize all values of |X3(k)|
>> stem(magX3)
>> xlabel('k');
>> ylabel('|X3(k)|');
```

(c) (d)

Figure 3 (c) The magnitude spectrum, $|X3(k)|$ and (d) a MATLAB® program for $|X3(k)|$.




```
>> PX3 = phase(X3);
>> stem(PX3)
>> xlabel('k');
>> ylabel('Phase(X3(k))');
```

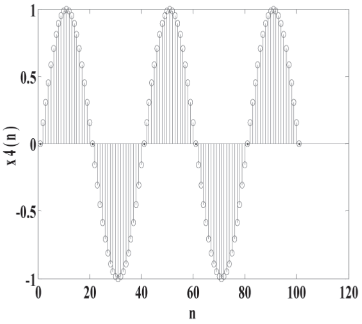
(e) (f)

Figure 3 (e) The phase spectrum of $X3(k)$ and (f) a MATLAB® program for phase spectrum of $X3(k)$.

Example 2: Sine signal, $x4(n)$, with $n = 100$ samples.

A sine signal, $x4(n)$, in spatial domain is demonstrated in Figure 4 (a). A function of *sin*( ) is employed to create this sine wave, which has 40 samples taken in each cycle.

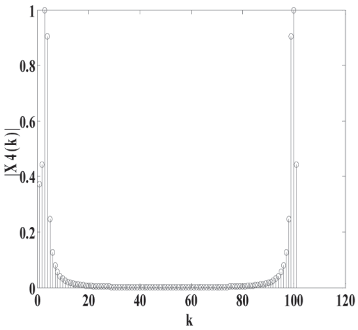
Figure 4 (b) illustrates a program for constructing this sine signal. A range and step of sample number (*n*) is defined in the first line of the program. *X*4(*k*) is a DFT of *x*4(*n*) and its magnitude spectrum (|*X*4(*k*)|) is revealed in Figure 4(c) by utilizing the program shown in Figure 4 (d). A phase spectrum of *X*4(*k*) is given in Figure 4 (e) by employing the program shown in Figure 4 (f).




```
>> n = 0:0.005:0.5; % Define range
and step of sample number (n)
>> x4 = sin(0.2*pi*50*n); % Create
sine wave x4(n)
>> stem(x4)
>> xlabel('n');
>> ylabel('x4(n)');
```

(a) (b)

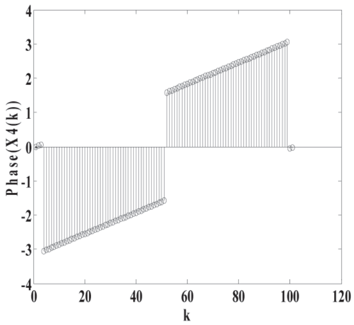
Figure 4 (a) A sine signal, *x*4(*n*), in spatial domain and (b) a MATLAB® program for sine signal, *x*4(*n*).




```
>> X4 = fft(x4);
>> absX4 = abs(X4);
>> X4max = max(max(absX4));
>> magX4 = (absX4/X4max);
>> stem(magX4)
>> xlabel('k');
>> ylabel('|X4(k)|');
```

(c) (d)

Figure 4 (c) The magnitude spectrum, |*X*4(*k*)|, of sine signal, *x*4(*n*) and (d) a MATLAB® program for |*X*4(*k*)|.




```
>> PX4 = phase(X4);
>> stem(PX4)
>> xlabel('k');
>> ylabel('Phase(X4(k))');
```

(e) (f)

Figure 4 (e) The phase spectrum of *X*4(*k*) and (f) a MATLAB® program for the phase spectrum of *X*4(*k*).

2.4 Real Cases for 1-D Signals

This section provides a real-world application of 1-D signals. Two demonstrations are provided as follows:

Example 1: Data a 60th row of Lena image, $x5(n)$, n = 128 samples.

Data at the 60th row of a Lena standard gray image of size 128×128 pixels, $x5(n)$ is selected as an example. Figure 5 (a) depicts a Lena image with pointed data at the 60th row as a representative of an actual case of 1-D signal of 128 data points. White lines above and below the 60th row represent data from 59th and 61st rows, respectively. These two rows of data are set to 255, which correspond to the maximum value or the brightest point of the image. A minimum value for the image pixel is 0, which corresponds to the darkest point in the image. Thus, all data from these two rows appear as white lines in order to distinguish the data from the 60th row from the other rows.

Figure 5 (b) illustrates a program to generate Figure 5 (a). An *imread*( ) function is used to read a real size 512×512 pixels of the Lena image (Lena_standard_gray.tif) and assign to a parameter "Image1". This image is then resized to 128×128 pixels employing a function of *imresize*( ). However, two parameters (inside the parenthesis) are required for this function. The first paramater indicates the input image, which needs to be resized (in this case it is the parameter "Image1"). The second parameter is the desired size (i.e. 128×128 pixels). This resized image is assigned to a parameter "A1". The commands of aa1(61, :) = 255 and aa1(59, :) = 255 are used to set all data from the 59th and 61st rows to 255 (or the brightest points). A parameter "aa1" in front of the parenthesis of both commands is a dummy variable and can be arbitrarily assigned any new name. The image is now ready to view by using a function of *imshow*( ).

Data at → the 60th row

(a)

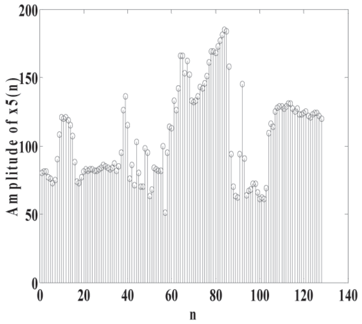
```
>> Image1 = imread ('Lena_standard_gray.tif'); %Read all data of
Lena image real size (512×512)
>> A1 = imresize (Image1, [128 128]);%Resize Lena picture to
128×128 and assign this resize image to a dummy variable A1
>> aa1 = A1; % Assign aa equals to data in A
>> aa1(61,:) = 255; % Set all data at the 61th row to 255
>> aa1(59,:) = 255; % Set all data at the 59th row to 255
>> imshow(aa1) %Show Lena picture, which has all data the 59th
and 61st rows set to 255
```

(b)

Figure 5 (a) The Lena image with pointed data at the 60th row, $x5(n)$, in the spatial domain and (b) a MATLAB® program for creating the Lena image.

Figure 5 (c) shows a plot of the original data (spatial domain) from the 60th row for the real case 1-D signals. From this figure, all values of the data are within the range of 0-255 (minimum and maximum values of image pixel). Figure 5 (d) provides a program to select the data from the 60th row and plot them. A command on the first line of this program is used to select a data of row 60th. The other rows of Lena image can also be used by simply changing the row number in the provided program.

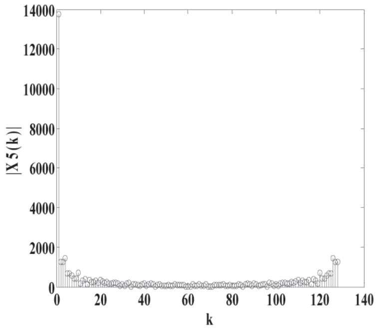
A magnitude spectrum (|$X5(k)$|) of this 1-D signal is given in Figure 5 (e). A program for creating this magnitude spectrum is shown in Figure 5 (f). A phase spectrum of $X5(k)$ is demonstrated in Figure 5 (g) by employing the program given in Figure 5 (h). All functions of *fft*( ), *abs*( ), *stem*( ), *phase*( ), *xlabel*( ), and *ylabel*( ) are applied here again to obtain both magnitude and phase spectrums.




```
>> x5 = A1(60,:); %Set B equals to
all data of row 60th of Lena, x5(n)
>> stem(x5)
>> xlabel('n');
>> ylabel('Amplitude of x5(n)');
```

(c) (d)

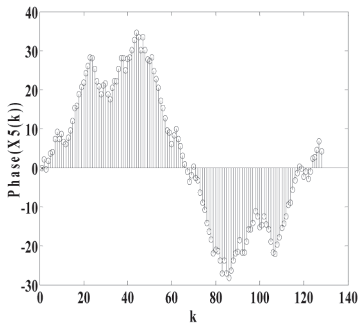
Figure 5 (c) Orignal data of Lena image by taking only data at the 60th row, $x5(n)$ and (d) a MATLAB® program for $x5(n)$.




```
>> X5 = fft(x5);
>> absX5 = abs(X5);
>> stem(absX5)
>> xlabel('k');
>> ylabel('|X5(k)|');
```

(e) (f)

Figure 5 (e) The magnitude spectrum of Lena at the 60th row, |$X5(k)$| and (f) a MATLAB® program for magnitude spectrum, |$X5(k)$|.




```
>> PX5 = phase(X5);
>> stem(PX5)
>> xlabel('k');
>> ylabel('Phase(X5(k))');
```

(g) (h)

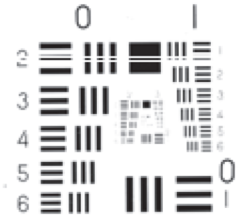
Figure 5 (g) The phase spectrum of Lena data at the 60th row, $X5(k)$ and (h) a MATLAB® program for phase spectrum of $X5(k)$.

Example 2: Data from the 33rd row of a Resolution chart image, $x6(n)$, n = 128 samples.

In this example a Resolution chart of size 128×128 pixels, $x6(n)$, is used as an example: data from the 33rd row is selected. Figure 6 (a) illustrates Resolution chart with pointed data at the 33rd row as a representation of 1-D signal. However, the actual size of this image is 256×256 pixels. Thus, this image must be resized. Moreover, data from the 32nd and 34th rows are set to 255 (the brightest point) in order to distinguish a data from the 33rd row from the other two. Figure 6 (b) reveals a program for generating the image in Figure 6 (a). All functions in the previous example are reapplied here. The data at the 33rd row, $x6(n)$, is plotted as shown in Figure 6 (c) using the program given in Figure 6(d).

A DFT of this data ($X6(k)$) is determined and its magnitude spectrum ($|X6(k)|$) is plotted as demonstrated in Figure 6 (e). The program provided in Figure 6 (f) is employed to complete this task. A phase spectrum of $X6(k)$ is demonstrated in Figure 6 (g) by employing the program shown in Figure 6 (h).

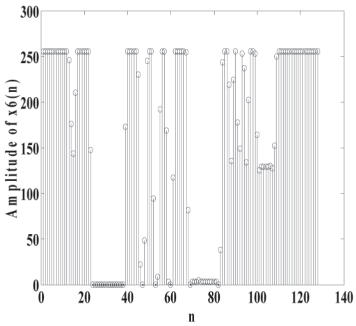
Data at the 33rd row →




```
>> Image2 = imread ('Resolution
chart.tif');
>> A2 = imresize (Image2, [128
128]);
>> aa2 = A2;
>> aa2(34,:) = 255;
>> aa2(32,:) = 255;
>> imshow(aa2)
```

(a) (b)

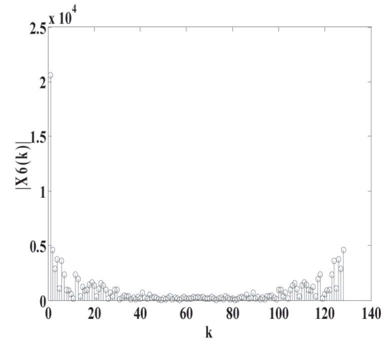
Figure 6 (a) The Resolution chart with pointed data at the 33rd row, $x6(n)$, in the spatial domain and (b) a MATLAB® program for creating Resolution chart in Figure 6 (a).




```
>> x6 = A2(33,:);
>> stem(x6)
>> xlabel('n');
>> ylabel('Amplitude of x6(n)');
```

(c) (d)

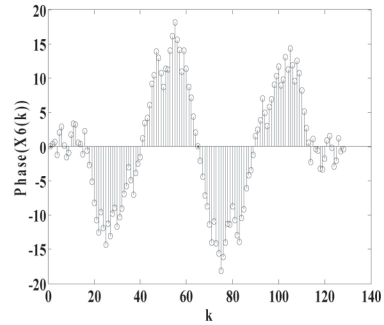
Figure 6 (c) Orignal data of a Resolution chart generated from the data at the 33rd row, $x6(n)$ and (d) a MATLAB® program for $x6(n)$




```
>> X6 = fft(x6);
>> absX6 = abs(X6);
>> stem(absX6)
>> xlabel('k');
>> ylabel('|X6(k)|');
```

(e) (f)

Figure 6 (e) The magnitude spectrum of the Resolution chart data from the 33rd row, $|X6(k)|$ and (f) a MATLAB® program for magnitude spectrum, $|X6(k)|$.




```
>> PX6 = phase(X6);
>> stem(PX6)
>> xlabel('k');
>> ylabel('Phase(X6(k))');
```

(g) (h)

Figure 6 (g) The phase spectrum of the Resolution chart data from the 33rd row, $X6(k)$ and (h) a MATLAB® program for phase spectrum of $X6(k)$.

3. Conclusion

The main objectives of this article were to provide the reader the 1-D DFT theoretical concept and an implementation using the MATLAB® program. Therefore, several examples on both 1-D synthetic and real-world cases were demonstrated. Numerous figures of spatial and frequency domains were provided – accompanied by their cognate MATLAB® programs in an easy to follow format.

Moreover, the authors also apply DFT in the research areas of the Compressive Sensing (CS) [Ha, P. H., Lee, W. and Patanavijit, V., 2014; Patanavijit, V. & Ha, P.H., 2013] for signal modeling in frequency domain and the signal registration [Patanavijit, V., 2011] for reconstructing the higher resolution signal by using phase information.

4. References

Duhamel, P. & Vetterli, M. (1990). Fast Fourier Transforms: a tutorial review and a state of the art. *Journal of Signal Processing*, 19 (4), 259-299.

Haykin, S & Moher, M. (2003*). Communication Systems*. John Wiley & Sons.

Haykin, S & Moher, M. (2007). *Introduction to analog & digital communications*. John Wiley & Sons.

Ingle, V. K & Proakis, J. G. (2007*). Digital signal processing using MATLAB* (*International Student Edition*). Thomson.

Phillips, C. L., Parr, J. M. & Riskin, E. A. (2003). *Signals, systems, and transforms*. Pearson Education International.

Schilling, R. J. & Harris, S. L. (2005). *Fundamentals of digital signal processing: Using MATLAB®*. Thomson.
Vaezi, M. & Labeau, F. (2013). Systematic DFT frames: Principle, eigenvalues structure, and applications. *IEEE Transactions on Signal Processing*, 61 (15), 3774-3885.
Wu, T-W. & Chung, C-D. (2014). Spectrally precoded DFT-based OFDM and OFDMA with oversampling. *IEEE Transactions on Vehicular Technology*, 63 (6), 2769-2783.
Li, F., Li, X. & Yu, J. (2015). Performance comparison of DFT-spread and pre-equalization for 8×244.2-Gb/s PDM-16QAM-OFDM. *Journal of Lightwave Technology*, 33 (1), 227-233.
Vaezi, M. & Labeau, F. (2014). Generalized and extended subspace algorithms for error correction with quantized DFT codes. *IEEE Transactions on Communications*, 62 (2), 410-422.
Xiong, X., Jiang, B., Gao, X. & You, X. (2014). DFT-based channel estimator for OFDM systems with leakage estimation. *IEEE Communications Letters*, 17 (8), 1592-1595.
Oppenheim, A.V. & Schafer, R.W.(2009). *Discrete-Time Signal Processing*. Prentice Hall.
Edussooriya, C.U.S., Bruton, L.T., Agathoklis, P. & Gunaratne, T.K. (2013). Low-complexity maximally-decimated multirate 3-D spatio-temporal FIR cone and frustum Filters. *IEEE Transactions on Circuits and Systems I*, 60 (7), 1845-1856.
Kamwa, I., Samantaray, S.R. & Joos, G. (2014). Wide frequency range adaptive phasor and frequency PMU algorithms. *IEEE Transactions on Smart Grid*, 5 (2), 569-579.
Hu, L., Zhou, J., Shi, Z. & Fu, Q. (2013). A fast and accurate reconstruction algorithm for compressed sensing of complex sinusoids. *IEEE Transactions on Signal Processing*, 61 (22), 5744-5754.
Ha, P.H., Lee, W. & Patanavijit, V. (2014). An introduction to compressive sensing for digital signal reconstruction and its implementation on digital image reconstruction. In Proceedings of the 2014 International Electrical Engineering Congress (iEECON 2014), Pattaya, Thailand, Mar. 2014, pp. 1-4.
Patanavijit, V. & Ha, P.H. (2013). A novel frequency domain image reconstruction based on the Tikhonov regularization and robust estimation technique for compressive sensing. In Proceedings of The 10th Annual International Conference of Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2013), ECTI Association Thailand, Krabi, Thailand, May 2013, pp. 1-6.
Patanavijit, V. (2011). The empirical performance study of a super resoluton reconstruction based on frequency domain from aliased multi-low resolution images. In Proceedings of The 34th Electrical Engineering Conference (EECON-34), Ambassador City Jomtien Hotel, Pataya, Chonburi, Thailand, Dec., pp. 645-648.

Authors

Dr.Kornkamol Thakulsukanant
Martin de Tours School of Management and Economics (MSME Building 2nd floor)
Assumption University (Suvarnabhumi Campus), 88 Moo 8 Bang Na-Trad Km.26,
Bangsaothong, Samuthprakarn, Thailand, 10540
e-mail: kthakulsukanant@yahoo.com

Asst. Prof. Dr.Vorapoj Patanavijit
Vincent Mary School of Engineering (VMS Building 2nd floor)
Assumption University (Suvarnabhumi Campus), 88 Moo 8 Bang Na-Trad Km.26,
Bangsaothong, Samuthprakarn, Thailand, 10540
e-mail: patanavijit@yahoo.com