Fourier Series and its Application in Radar Imaging

Geralt Heart

Advisor: Dr. Zhijun Qiao (MATH 4390)

April 30 2020

Abstract

Functions in mathematics are commonly approximated using Taylor and Maclaurin series to create polynomial function of x^n. Some functions are better approximated by sine and cosine rather than a polynomial function. The goal is to create an approximation of input data using Fourier series that will create a more accurate and faster approximation than a Taylor or Maclaurin series could. In addition, the usage of the Fast Discrete Fourier Transform will optimize Fourier Transform further.

1 Introduction

In this paper we will explore examples and develop an understanding for the Discrete Fourier Transform (DFT), and the Faster Discrete Fourier Transform (FFT). DFT computes Fourier transform from sampled values. The method that numerically performs the computation is done by the Fast Fourier transform algorithm FFT. DFT is a transform that actually acts on a sequence of values, rather than an interval of values as the case with Fourier Transform. However, Fourier Transform requires knowing the function on an interval (a,b). In applications, like the one in the MIMO radar imaging paper [6], the function is measured at a discrete set of points, in multi-dimension. To explain DFT, we consider a sequence of N real or complex number, as well as consider an example of a multidimension DFT. In addition, we will also take a look at using MATLAB as a viable resource for calculations that are not realistically done by hand.

2 Discrete Fourier Transform

Let

$$x_i,\ i = 0, 1, \dots, N-1$$

Let x represent the vector

$$x = (x_0, x_1, \dots, x_N - 1) = (x(i))_{i=0}^{N-1}$$

So it is a function over the discrete domain $\{0, 1, 2, \dots, N-1\}$ with usual addition and multiplication operation.

2.1 Properties

$$(x + y)(k) = x(k) + y(k)$$

$$(\mathrm{xy})(k) = x(k)\, y(k)$$

$$(\mathrm{ax})(k) = a(x(k)) \text{ where a is a scalar}$$

Definition Discrete Fourier Transfer Given an N-sequence

$$x = (x(0), x(1), .., x(N-1)),$$

we define the N-point discrete Fourier transform of x, DFT of x, by

$$F_N(x)(k) = X(k) = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x(j)\, e^{2\pi i j \frac{k}{N}}\ ,\ k = 0, 1, .., N-1\ ,\ i = \sqrt{-1}$$

So, the DFT of an N-sequence x is another N-sequence of complex numbers X=(X(0),...,X(N-1)). This is a similarity between FT and DFT; we use integration in FT, and summation in DFT.

As an example, we compute the 4-point DFT of $x = (0, 1, 1, -1)$: From the equation

$$F_4(x)(k) = X(k) = \frac{1}{\sqrt{4}} \sum_{j=0}^{3} x(j)\, e^{2\pi i j \frac{k}{4}}\ ,\ k = 0, 1, 2, 3$$

At k=0

$$F_4(x)(0) = X(0) = \frac{1}{2} \sum_{j=0}^{3} x(j)\, 1 = \frac{1}{2}[0 + 1 + 1 - 1] = \frac{1}{2}$$

At k=1

$$F_4(x)(1) = X(1) = \frac{1}{2} \sum_{j=0}^{3} x(j)\, e^{2\pi i j \frac{1}{4}}$$

$$= \frac{1}{2} \sum_{j=0}^{3} x(j)\, e^{\frac{\iota\pi j}{2}}$$

$$= \frac{1}{2}[x(0)\, e^{i\left(\frac{0\pi}{2}\right)} + x(1)\, e^{i\left(\frac{1\pi}{2}\right)} + x(2)\, e^{i\left(\frac{2\pi}{2}\right)} + x(3)\, e^{i\left(\frac{3\pi}{2}\right)}]$$

$$= \frac{1}{2}[0 + e^{i\frac{\pi}{2}} + e^{\iota\pi} - e^{i\frac{3\pi}{2}}]$$

Using the Euler Formula

$$e^{\iota\vartheta} = cos\theta + isin\theta$$

We get

$$e^{i\frac{\pi}{2}} = i, e^{\mathrm{i}\pi} = -1,\quad e^{i\frac{3\pi}{2}} = -i$$

$$F_4\left(x\right)\left(1\right)=\frac{1}{2}[0+i-1+i]$$

$$=\frac{1}{2}\left[-1+2i\right]=-\frac{1}{2}+i$$

At k=2

$$F_4\left(x\right)\left(2\right)=X\left(2\right)=\frac{1}{2}\sum_{j=0}^{3}x\left(j\right)e^{2\pi ij\frac{2}{4}}$$

$$=\frac{1}{2}\sum_{j=0}^{3}x\left(j\right)e^{\pi\imath j}$$

$$=\frac{1}{2}[x\left(0\right)e^{0}+x\left(1\right)e^{i\pi}+x\left(2\right)e^{2\pi i}+x(3)e^{3\pi i}]$$

$$F_4\left(x\right)\left(2\right)=\frac{1}{2}[0+1\left(-1\right)+1\left(1\right)+(-1)(-1)]$$

$$=\frac{1}{2}$$

At k=3

$$F_4\left(x\right)\left(3\right)=X\left(3\right)=\frac{1}{2}\sum_{j=0}^{3}x\left(j\right)e^{2\pi ij\frac{3}{4}}$$

$$=\frac{1}{2}[x\left(0\right)e^{0}+x\left(1\right)e^{i\frac{3\pi}{2}}+x\left(2\right)e^{3\pi i}+x(3)e^{\frac{9\pi}{2}i}]$$

$$=\frac{1}{2}[0+\left(-i\right)+\left(-1\right)+\left(-1\right)i]$$

$$=\frac{1}{2}\left[-1-2i\right]=-\frac{1}{2}-i$$

So the Discrete Fourier Transform of

$$x=(0,1,1,-1)$$

Is

$$\mathbf{X}=(\mathbf{\frac{1}{2}},-\mathbf{\frac{1}{2}}\mathbf{+i},\mathbf{\frac{1}{2}},-\mathbf{\frac{1}{2}}\mathbf{-2i})$$

3 Inverse Discrete Fourier Transform

Now the Inverse Discrete Fourier Transform (IDFT) is defined by:

$$F_N^{-1}(X)(k) = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} X(j) e^{-2\pi j i \frac{k}{N}}, k = 0, 1, \ldots, N-1$$

For

$$X = (X(0), X(1), \ldots, X(N-1))$$

And we can show:

$$F_N^{-1}(F_N(x))(k) = x(k), \ k = 0, 1, \ldots, N-1$$

Here we will compute the Inverse Discrete Transform for the Problem

Let

$$X = (\frac{1}{2}, -\frac{1}{2} + i, \frac{1}{2}, -\frac{1}{2} - 2i)$$

Then,

$$F_4^{-1}(X)(k) = \frac{1}{\sqrt{4}} \sum_{j=0}^{3} X(j) e^{-2\pi j i \frac{k}{4}}, k = 0, 1, 2, 3$$

Thus,

$$F_4^{-1}(X)(k) = \frac{1}{2} \sum_{j=0}^{3} X(j) e^{-2\pi j i \frac{k}{4}}$$

At k=0

$$F_4^{-1}(X)(0) = \frac{1}{2} \sum_{j=0}^{3} X(j) 1$$

$$= \frac{1}{2}(\frac{1}{2} + -\frac{1}{2} + i, \frac{1}{2}, -\frac{1}{2} - i)$$

$$= 0 = x(0)$$

At k=1

$$F_4^{-1}(X)(1) = \frac{1}{2} \sum_{j=0}^{3} X(j) e^{-2\pi j i \frac{1}{4}}$$

$$F_4^{-1}(X)(1) = \frac{1}{2}\sum_{j=0}^{3} X(j)\, e^{\frac{-\pi j}{2} i}$$

$$= \frac{1}{2}[X(0)(1) + X(1)\, e^{-i\frac{\pi}{2}} + X(2)\, e^{-i\pi} + X(3)e^{-i\frac{3\pi}{2}}]$$

$$= \frac{1}{2}\left[\frac{1}{2} + \frac{1}{2}i + 1 - \frac{1}{2} - \frac{1}{2}i + 1\right]$$

$$= 1 = x(1)$$

At k=2

$$F_4^{-1}(X)(2) = \frac{1}{2}\sum_{j=0}^{3} X(j)\, e^{-2\pi j i\frac{2}{4}}$$

$$F_4^{-1}(X)(2) = \frac{1}{2}\sum_{j=0}^{3} X(j)\, e^{-\pi j i}$$

$$= \frac{1}{2}[X(0)(1) + X(1)\, e^{-\pi i} + X(2)\, e^{-2\pi i} + X(3)e^{-3\pi i}]$$

$$= \frac{1}{2}[\frac{1}{2}(1) + (-\frac{1}{2} + i)(-1) + \frac{1}{2}(1) + (-\frac{1}{2} - i)(-1)]$$

$$1 = x(2)$$

At k=3

$$F_4^{-1}(X)(3) = \frac{1}{2}\sum_{j=0}^{3} X(j)\, e^{-2\pi j i\frac{3}{4}}$$

$$F_4^{-1}(X)(3) = \frac{1}{2}\sum_{j=0}^{3} X(j)\, e^{-\frac{3\pi j}{2} i}$$

$$= \frac{1}{2}[X(0)(1) + X(1)\, e^{-i\frac{3\pi}{2}} + X(2)\, e^{-i(3\pi)} + X(3)\, e^{-i\left(\frac{9\pi}{2}\right)}]$$

$$= \frac{1}{2}[\frac{1}{2} + \left(\frac{-1}{2} + i\right)(i) + \frac{1}{2}(-1) + \left(-\frac{1}{2} - i\right)(-i)]$$

$$\frac{1}{2}\left[\frac{1}{2} - \frac{1}{2}i - 1 - \frac{1}{2} + \frac{1}{2}i - 1\right] = \frac{1}{2}[-2] = -1 = x(3)$$

Let us also consider the general case:

Theorem:

$$F_N^{-1}(F_N(x))(k) = x(k),\ k = 0, \ldots, N-1 \tag{1}$$

Which implies the uniqueness property of DFT $[if X = Y \ \rightarrow x = y]$

Proof

First we prove the orthogonality property:

$$\sum_{n=0}^{N-1} e^{2\pi in\frac{j-k}{N}} = 0,\ if\ j \neq k$$

The Left Hand Side

$$= \sum_{n=0}^{N-1} \left[e^{2\pi i\left(\frac{j-k}{N}\right)}\right]^{n} = 1 + e^{2\pi i\left(\frac{j-k}{N}\right)} + \left(e^{2\pi i\left(\frac{j-k}{N}\right)}\right)^{2} + \ldots + \left(e^{2\pi i\left(\frac{j-k}{N}\right)}\right)^{N-1}$$

The Right Hand Side is the N terms of geometric series with ratio $e^{2\pi i\left(\frac{j-k}{N}\right)}$

$$\text{the sum} = \frac{1 - e^{2\pi i\left(\frac{j-k}{N}\right)N}}{1 - e^{2\pi i\left(\frac{j-k}{N}\right)}} = \frac{1 - e^{2\pi i(j-k)}}{1 - e^{2\pi i\left(\frac{j-k}{N}\right)}} = \frac{1-1}{1 - e^{2\pi i}} = 0$$

So we get the orthonormality

$$\frac{1}{N}\sum_{n=0}^{N-1} e^{2\pi i\left(\frac{j-k}{N}\right)} = \begin{Bmatrix} 0\ if j \neq k \\ 1\ if\ j = k \end{Bmatrix}$$

The second part comes from the fact that when $j = k$, the left hand side is

$$\frac{1}{N}\left[1 + 1 + \ldots + 1\right] = \frac{N}{N} = 1$$

So, now back to equation

$$F_N^{-1}\left(F_N\left(x\right)\right)(k) = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1}\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1} x(j)e^{2\pi ij\frac{j}{N}}e^{-2\pi ij\frac{k}{N}}$$

$$= \frac{1}{N}\sum_{j=0}^{N-1} x\left(j\right)\sum_{n=0}^{N-1} e^{2\pi i(j-k)}\frac{n}{N} = \frac{1}{N}x\left(k\right)N = x\left(k\right)$$

Thus our verification of

$$F^{-1}F = I \tag{2}$$

Is complete.

We also notice that k in F(x)(k) and$F^{-1}\left(X\right)(k)$ is in the set {0,1,…,N-1}. However, because $e^{2\pi i\frac{k}{N}}$ is periodic, we can see that if we replace k by (k+N) in the exponent of the terms we get:

$$e^{2\pi ij\left(\frac{k+N}{N}\right)} = e^{2\pi ij\left(\frac{k}{N}\right)} * e^{2\pi ij\left(\frac{N}{N}\right)} = e^{2\pi ij\left(\frac{k}{N}\right)} * 1 = e^{2\pi ij\left(\frac{k}{N}\right)}$$

DFT is periodic because

$$X(k) = X(k+N) \ for \ all \ k$$

And also

$$x\left(k\right) = x\left(k+N\right) \text{ for all k}$$

So, both DFT & IDFT are periodic. Also from the definition we have

$$F_N\left(ax+by\right) = aF_N\left(x\right) + bF_N(y)$$

and

$$F_N^{-1}\left(ax+by\right) = aF_N^{-1}\left(x\right) + bF_N^{-1}(y)$$

Both F_N and F_N^{-1} are linear.

Because of the periodicity property of DFT

$$x\left(k\right) = x\left(k+N\right) \text{ for all k}$$

The following convolution definition is justified.

Consider two N-sequences x and y; both are N-periodic sequences on the integers, so we define their convolution by

$$\left(x*y\right)\left(k\right) = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1} x\left(j\right)y(k-j), \ k = 0, 1, \ldots, N-1$$

Since $(x*y)$ is also N-periodic

$$\left(x*y\right)\left(k\right) = \left(x*y\right)\left(k+N\right) \text{for all k}$$

Also we have

$$F_N\left(x*y\right) = F_N\left(x\right)F_N\left(y\right) = XY$$

And

$$F_N^{-1}\left(\text{XY}\right) = F_N^{-1}\left(X\right) * F_N^{-1}\left(Y\right) = x*y \tag{3}$$

Let us consider an example to verify the convolution theorem (3)

Let

$$x = (1, 1, 1, 2)$$

$$y = (-1, 1, 3, -2)$$

Let us find x*y using DFT & IDFT

Let

$X = F_4(x)$, $Y = F_4(y)$

$$X\left(k\right) = \frac{1}{\sqrt{4}}\sum_{j=0}^{3} x\left(j\right) * e^{2\pi ij\left(\frac{k}{4}\right)}$$

At k=0

$$X\left(0\right) = \frac{1}{2}\sum_{j=0}^{3} x\left(j\right) * 1 = \frac{1}{2}\left[1+1+1+2\right] = \frac{5}{2}$$

At k=1

$$X\left(1\right) = \frac{1}{2}\sum_{j=0}^{3} x\left(j\right) * e^{2\pi ij\left(\frac{1}{4}\right)} = -\frac{1}{2}i$$

At k=2

$$X\left(2\right) = \frac{1}{2}\sum_{j=0}^{3} x\left(j\right) * e^{2\pi ij\left(\frac{2}{4}\right)} = -\frac{1}{2}i$$

At k=3

$$X\left(3\right) = \frac{1}{2}\sum_{j=0}^{3} x\left(j\right) * e^{2\pi ij\left(\frac{3}{4}\right)} = \frac{1}{2}i$$

$$X = F_4\left(x\right) = \left(\frac{5}{2}, -\frac{1}{2}i, -\frac{1}{2}i, \frac{1}{2}i\right)$$

Similarly,

$$Y = F_4\left(y\right) = \left(\frac{1}{2}, -2+\frac{3}{2}i, \frac{3}{2}, -2-\frac{3}{2}i\right)$$

$$XY = \left(\frac{5}{4}, \frac{3}{4}+i, -\frac{3}{4}, \frac{3}{4}-i\right)$$

Now

$$F_4^{-1}\left(\mathrm{XY}\right)\left(k\right) = \frac{1}{2}\sum_{j=0}^{N-1}\left(\mathrm{XY}\right)\left(j\right)e^{-2\pi ij\left(\frac{k}{4}\right)}$$

$$\left(x*y\right)\left(k\right) = \frac{1}{2}[\left(\frac{5}{4}\right)\left(1\right) + \left(\frac{3}{4}+i\right)e^{-2\pi i(1)\left(\frac{k}{4}\right)} + \left(-\frac{3}{4}\right)e^{-2\pi i(2)\left(\frac{k}{4}\right)} + \left(\frac{3}{4}-i\right)e^{-2\pi i(3)\left(\frac{k}{4}\right)}$$

$$\left(x*y\right)\left(0\right) = 1$$

$$(x * y)(1) = 2$$

$$(x * y)(2) = -\frac{1}{2}$$

$$(x * y)(3) = 0$$

$$\mathbf{x * y = (1, 2, -\frac{1}{2}, 0)}$$

Now if we find the convolution of x and y without the transform,

$$(x * y)(k) = \frac{1}{\sqrt{4}} \sum_{j=0}^{3} x(j)\, y(k-j) = \frac{1}{2}[x(0)\, y(k) - x(1)\, y(k-1) + x(2)\, y(k-2) + x(3)\, y(k-3)]$$

$$\text{Note}: \ \textit{negative numbers will loop}: \ (-1, -2, -3) > (3, 2, 1)$$

At k=0

$$(x * y)(0) = \frac{1}{2}[x(0)\, y(0) + x(1)\, y(-1) + x(2)\, y(-2) + x(3)\, y(-3)]$$

$$= 1$$

At k=1

$$(x * y)(1) = \frac{1}{2}[x(0)\, y(1) + x(1)\, y(0) + x(2)\, y(-1) + x(3)\, y(-2)]$$

$$= 2$$

k=2

$$(x * y)(2) = \frac{1}{2}[x(0)\, y(2) + x(1)\, y(1) + x(2)\, y(0) + x(3)\, y(-1)]$$

$$= -\frac{1}{2}$$

k=3

$$(x * y)(3) = \frac{1}{2}[x(0)\, y(3) + x(1)\, y(2) + x(2)\, y(1) + x(3)\, y(0)]$$

$$= 0$$

$$\mathbf{x * y = \left(1, 2, -\frac{1}{2}, 0\right)}$$

So we see that the same answer is obtained with convolution or the DFT and IDFT.

Also we can consider a general case,

Let:

$$F_N\left(x * y\right) = F_N(x) F_N(y)$$

Because

$$F\left(x * y\right)(k) = F_N\left(\frac{1}{\sqrt{N}}\sum_{l=0}^{N-1}\left(x\left(l\right)y\left(k-l\right)\right)\right)$$

$$= \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}\frac{1}{\sqrt{N}}\sum_{l=0}^{N-1}x\left(l\right)y(j-l)e^{2\pi ij\frac{k}{N}}$$

$$= \frac{1}{\sqrt{N}}\sum_{l=0}^{N-1}\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}x\left(l\right)y\left(j-l\right)e^{2\pi ij\frac{k}{N}}$$

Now let $j - l = m$ in the internal sum

So we get

$$F_N\left(x * y\right)(k) = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}x(l)\frac{1}{\sqrt{N}}\sum_{m=-l}^{N-l-1}y\left(m\right)e^{2\pi i(m+l)\frac{k}{N}}$$

$$= \frac{1}{\sqrt{N}}\sum_{l=0}^{N-1}x(l)e^{2\pi il\frac{k}{N}}\frac{1}{\sqrt{N}}\sum_{m=-l}^{N-l-1}y\left(m\right)e^{2\pi i(m+l)\frac{k}{N}}$$

Because

$$y(m)e^{2\pi i(m+l)\frac{k}{N}}$$

Is periodic

$$F_N\left(x * y\right)(k) = X\left(k\right)Y(k)$$

$$= F_N(x)(k)F_N(y)(k)$$

$$F_N\left(x * y\right) = F_N\left(x\right)F_N\left(y\right) = XY$$

And because of the uniqueness of DFT, we also have:

$$F_N^{-1}\left(\mathrm{XY}\right) = F_N^{-1}\left(X\right) * F_N^{-1}\left(Y\right) = x * y$$

4 Faster Fourier Transform

The Fast Fourier Transform (FFT) is an efficient algorithm that was discovered in 1965 by J.W. Tukey and J.W. Cooley and is considered one of the century's most important contributions in Numerical Analysis because it reduces the number of elementary operations for computing the DFT.

So for DFT: $x = (x_0, \ldots, x_{N-1}) \to X = (X_0, \ldots, X_{N-1})$

$$X(k) = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x(j)\, e^{2\pi i j \frac{k}{N}}$$

Each sum requires N elementary operations,, and there are N k's, therefore the total number of elementary operations is N^2 for X.

However, if $N = N_1 N_2$, the number of operation can be reduced to$N(N_1 + N_2)$. If $N = 2^s$, the number of operations will be $N(2 + 2 + \ldots + 2_s) = 2sN = 2N \log_2 N$. Which is much faster than an N^2 number of operations if N is large.

We can approximate FT by DFT and then use FFT to get numerical approximation for FT

$$F(f) = \hat{f}(w) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x)\, e^{-iwx} \mathrm{dx}, -\infty < x < \infty$$

Now for the approximation using DFT

Choose M to be a large number, $\Delta x = (\frac{M}{N})$, and using Riemann sum approximation,

$$\hat{f}(w) = \frac{1}{\sqrt{2\pi}} \frac{M}{N} \sum_{j=0}^{N-1} f(j\frac{M}{N}) e^{-iwj\frac{M}{N}}$$

$$= \frac{1}{\sqrt{2N\pi}} \frac{M}{\sqrt{N}} \sum_{j=0}^{N-1} f(j\frac{M}{N}) e^{-iwj\frac{M}{N}}$$

Now evaluating

$$\hat{f} \ at \ w = \frac{2\pi k}{M}, \ for \ k = 0, 1, \ldots, N-1$$

We get

$$\hat{f}(\frac{2\pi k}{M}) \cong \frac{M}{\sqrt{2N\pi}} \frac{\mathbf{1}}{\sqrt{\mathbf{N}}} \sum_{\mathbf{j=1}}^{\mathbf{N-1}} \mathbf{f}(\mathbf{j}\frac{\mathbf{M}}{\mathbf{N}}) \mathbf{e}^{-\mathbf{i}(\frac{\mathbf{2\pi \times}}{\mathbf{N}})\mathbf{j}}$$

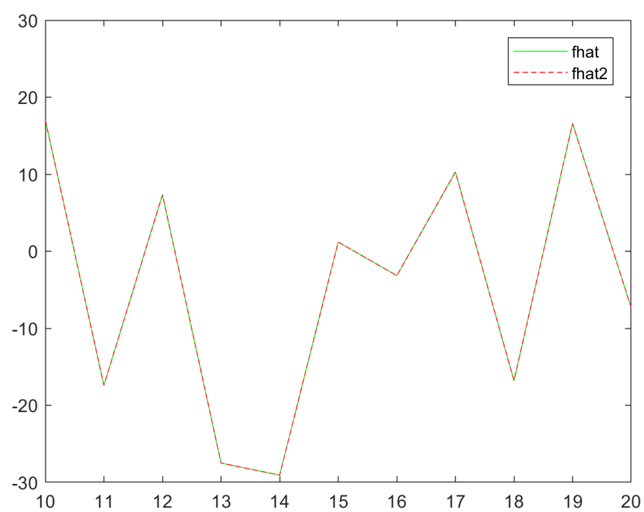
Therefore,

$$\hat{f}\left(\frac{2\pi k}{M}\right) \cong \frac{M}{\sqrt{2N\pi}} F_N^{-1}\left(f\left(j\frac{M}{N}\right)\right)(k), k = 0, 1, \ldots, N-1$$

Let us explore the usage of computer calculations in relation to discrete fourier series. Here we will be using MATLAB.

```
clear all; close all; clc
%fast discrete fourier transform FDFT
N=500;

f= randn(N,1);
fhat=fft(f);

w=exp(-i*2*pi/N);

for i=1:N
    for j=1:N
        DFT(i,j) = w^((i-1)*(j-1));
    end
end

fhat2=DFT*f; %should be equal to fhat

plot(real(fhat),'g')
hold on
plot(real(fhat2),'r--')
legend('fhat','fhat2')
axis([10 20 -30 30]);
%Notice that fhat and fhat2 overlap perfectly
```

Here we can see that the two lines match up perfectly



This example display a graphing of Single Input Single Output imaging or SISO. Next we will explore MIMO.

5 Multi-dimensional Discrete Fourier Transform

5.1 Background

MIMO radar systems transmit, through antennas, multiple probing signals. It is superior than a standard single phased array radar as it provides a higher resolution image and has higher detecting sensitivity.

Consider a MIMO radar system with M_t transmitter antennas and M_r receiver antennas. Let $x_m(n)$ denote the discrete-time base-band signal transmitted by the m_t transmit antenna, and let θ denote the location parameter of a target, for example its track angle and its range. The base-band signals at the target location can be described by:

$$\sum_{m=1}^{M_t} e^{-2j\pi f_o \tau_m(\theta)}$$

where f_o is the carrier frequency of the radar,

τ_m is the time needed by the signal emitted by the $(m_t) - th$ transmit antenna to arrive at the target.

N denotes the number of samples of each signal pulse.

5.2 Application

This application will show MIMO systems, which use multiple antennas at the transmitter and receive ends of a wireless communication system. We will use previously established data[7][8] and pre-existing built-in MATLAB code to display our MIMO data. This simulation covers an end-to-end system showing the encoded and/or transmitted signal, channel model, and reception and demodulation of the received signal. It is assumed here that the channel is known perfectly at the receiver for all systems.

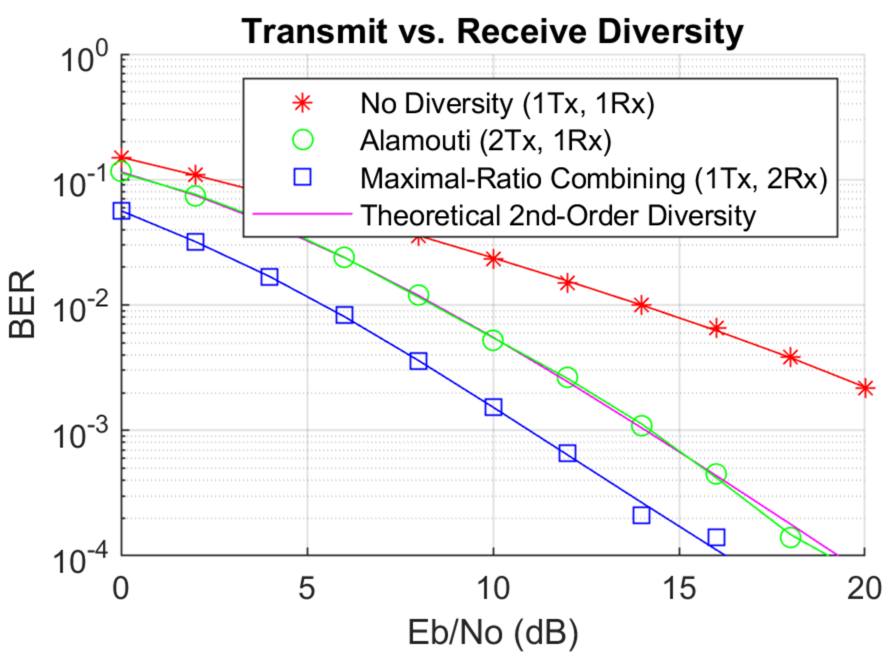
```
%%
% We start by defining some common simulation parameters
frmLen = 100;       % frame length
numPackets = 1000;  % number of packets
EbNo = 0:2:20;      % Eb/No varying to 20 dB
N = 2;              % maximum number of Tx antennas
M = 2;              % maximum number of Rx antennas


% and set up the simulation.

% Create comm.BPSKModulator and comm.BPSKDemodulator System objects(TM)
P = 2;              % modulation order
bpskMod = comm.BPSKModulator;
bpskDemod = comm.BPSKDemodulator('OutputDataType','double');

% Create comm.OSTBCEncoder and comm.OSTBCCombiner System objects
ostbcEnc = comm.OSTBCEncoder;
ostbcComb = comm.OSTBCCombiner;

% Create two comm.AWGNChannel System objects for one and two receive
% antennas respectively. Set the NoiseMethod property of the channel to
% 'Signal to noise ratio (Eb/No)' to specify the noise level using the
% energy per bit to noise power spectral density ratio (Eb/No). The output
% of the BPSK modulator generates unit power signals; set the SignalPower
% property to 1 Watt.
```

```
awgn1Rx = comm.AWGNChannel(...
    'NoiseMethod', 'Signal to noise ratio (Eb/No)', ...
    'SignalPower', 1);
awgn2Rx = clone(awgn1Rx);

% Create comm.ErrorRate calculator System objects to evaluate BER.
errorCalc1 = comm.ErrorRate;
errorCalc2 = comm.ErrorRate;
errorCalc3 = comm.ErrorRate;

% Since the comm.AWGNChannel System objects as well as the RANDI function
% use the default random stream, the following commands are executed so
% that the results will be repeatable, i.e., same results will be obtained
% for every run of the example. The default stream will be restored at the
% end of the example.
s = rng(55408);

% Pre-allocate variables for speed
H = zeros(frmLen, N, M);
ber_noDiver  = zeros(3,length(EbNo));
ber_Alamouti = zeros(3,length(EbNo));
ber_MaxRatio = zeros(3,length(EbNo));
ber_thy2     = zeros(1,length(EbNo));


% Set up a figure for visualizing BER results
fig = figure;
grid on;
ax = fig.CurrentAxes;
hold(ax,'on');

ax.YScale = 'log';
xlim(ax,[EbNo(1), EbNo(end)]);
ylim(ax,[1e-4 1]);
xlabel(ax,'Eb/No (dB)');
ylabel(ax,'BER');
```

```matlab
fig.NumberTitle = 'off';
fig.Renderer = 'zbuffer';
fig.Name = 'Transmit vs. Receive Diversity';
title(ax,'Transmit vs. Receive Diversity');
set(fig, 'DefaultLegendAutoUpdate', 'off');
fig.Position = figposition([15 50 25 30]);

% Loop over several EbNo points
for idx = 1:length(EbNo)
    reset(errorCalc1);
    reset(errorCalc2);
    reset(errorCalc3);
    % Set the EbNo property of the AWGNChannel System objects
    awgn1Rx.EbNo = EbNo(idx);
    awgn2Rx.EbNo = EbNo(idx);
    % Loop over the number of packets
    for packetIdx = 1:numPackets
        % Generate data vector per frame
        data = randi([0 P-1], frmLen, 1);

        % Modulate data
        modData = bpskMod(data);

        % Alamouti Space-Time Block Encoder
        encData = ostbcEnc(modData);

        % Create the Rayleigh distributed channel response matrix
        %   for two transmit and two receive antennas
        H(1:N:end, :, :) = (randn(frmLen/2, N, M) + ...
                         1i*randn(frmLen/2, N, M))/sqrt(2);
        %   assume held constant for 2 symbol periods
        H(2:N:end, :, :) = H(1:N:end, :, :);

        % Extract part of H to represent the 1x1, 2x1 and 1x2 channels
        H11 = H(:,1,1);
        H21 = H(:,:,1)/sqrt(2);
```

```
    H12 = squeeze(H(:,1,:));

    % Pass through the channels
    chanOut11 = H11 .* modData;
    chanOut21 = sum(H21.* encData, 2);
    chanOut12 = H12 .* repmat(modData, 1, 2);

    % Add AWGN
    rxSig11 = awgn1Rx(chanOut11);
    rxSig21 = awgn1Rx(chanOut21);
    rxSig12 = awgn2Rx(chanOut12);

    % Alamouti Space-Time Block Combiner
    decData = ostbcComb(rxSig21, H21);

    % ML Detector (minimum Euclidean distance)
    demod11 = bpskDemod(rxSig11.*conj(H11));
    demod21 = bpskDemod(decData);
    demod12 = bpskDemod(sum(rxSig12.*conj(H12), 2));

    % Calculate and update BER for current EbNo value
    %   for uncoded 1x1 system
    ber_noDiver(:,idx)  = errorCalc1(data, demod11);
    %   for Alamouti coded 2x1 system
    ber_Alamouti(:,idx) = errorCalc2(data, demod21);
    %   for Maximal-ratio combined 1x2 system
    ber_MaxRatio(:,idx) = errorCalc3(data, demod12);

end % end of FOR loop for numPackets

% Calculate theoretical second-order diversity BER for current EbNo
ber_thy2(idx) = berfading(EbNo(idx), 'psk', 2, 2);

% Plot results
semilogy(ax,EbNo(1:idx), ber_noDiver(1,1:idx), 'r*', ...
         EbNo(1:idx), ber_Alamouti(1,1:idx), 'go', ...
```

```
                EbNo(1:idx), ber_MaxRatio(1,1:idx), 'bs', ...
                EbNo(1:idx), ber_thy2(1:idx), 'm');
    legend(ax,'No Diversity (1Tx, 1Rx)', 'Alamouti (2Tx, 1Rx)',...
           'Maximal-Ratio Combining (1Tx, 2Rx)', ...
           'Theoretical 2nd-Order Diversity');

    drawnow;
end  % end of for loop for EbNo

% Perform curve fitting and replot the results
fitBER11 = berfit(EbNo, ber_noDiver(1,:));
fitBER21 = berfit(EbNo, ber_Alamouti(1,:));
fitBER12 = berfit(EbNo, ber_MaxRatio(1,:));
semilogy(ax,EbNo, fitBER11, 'r', EbNo, fitBER21, 'g', EbNo, fitBER12, 'b');
hold(ax,'off');

% Restore default stream
rng(s);
```



6 Conclusion

The motivation for the above presentation was the MIMO paper [6]. The paper shows an application of DFT for electro-magnetic waves. MIMO communications involves sending several messages to several receivers and develops radar imaging. The MIMO research uses filtering to remove the corruption of transmitted message to derive useful information. The information gained with MIMO Radar can be used to get red of false targets and create a better resolution image. The paper does that through multi-dimensional DFE with multiple sending and receiving by solving multi-dimensional wave equations. I used the paper as a guide for the presentation of simple DFT case, and for future possible research.

References

1. Ames, William F. *Numerical Methods for Partial Differential Equations* . Acad. Pr., 1985.

2. Asmar, Nakhlé H. *Partial Differential Equations with Fourier Series and Boundary Value Problems* . Dover, 2016.

3. Davis, John M. *Introduction to Applied Partial Differential Equations* . W.H. Freeman & Co., 2013.

4. Farlow, Stanley J. *Partial Differential Equations for Scientists and Engineers* . Dover Publications, Inc., 2016.

5. Kreyszig, E. *Advanced Engineering Mathematics* . John Wiley & Sons, 2006.

6. Y.Cao, J.F. Lopez,A. Martinex, and Zhijun Qiao: *A Mathematical Model for MIMO Imaging.* Spieditial-library.org, 2012

7. Jian Li, Petre Stoica: *MIMO Radar Signal Processing*; John Wiley Publishing

8. S.M. Alamouti, *A Simple transmit diversity technique for Wireless Communications*, IEEE Journal on Selected Areas in Communications, Vol.16 No. 8, Oct. 1998, pp 1451-1458