ISSN 2250-3153

Use of open-source language R-Programming for computation of Discrete Fourier Transform

Nadar Prince[1], Abhishek Sengupta[2], Juie Shah[1] and Yogesh Chandurkar[1]

[1]Department of Electronics and Telecommunication Engineering, Fr. Conceicao Rodrigues Institute of Technology, Vashi.
[2]Department of Computer Engineering, Fr. Conceicao Rodrigues Institute of Technology, Vashi.

Abstract – In this paper, we have made use of R-Programming, an open-source statistical computation and datamining language for computation of Discrete Fourier Transform. Computation of DFT is illustrated because it is the most fundamental concept in Digital Signal Processing and helps in establishing a platform for implementing other Digital Signal Processing algorithms, as a result, R-Programming can serve as an alternative to other software's like MATLAB, Octave, etc. At the same time, it can be extensively used with other statistical packages provided by R-Programming.

Index Terms- Scientific computation, Digital Signal Processing, R-Programming, open-source language.

I. INTRODUCTION

Computers have played a huge role in realizing the different mathematical theories and algorithms. Mathematical theories which will be difficult to realize using physical hardware or manual calculation have been made simpler by the use of computers. Computer programs enable the research individual to write a piece of code which is equivalent to mathematical formulas and interpretations. In the early days some of the key programs that played a vital role in this were COBAL, FORTRAN, C etc. Over the time, programming languages were developed specifically for mathematical computation. These were developed specifically for statistical computation and scientific computation. Scientific computation primarily dealt with realizing algorithms, theorems, formulas, etc related to signal processing, followed by digital signal processing.

One of the key programming language that played a huge role in this process was MATLAB. It provided most of the signal processing function which were in-built. This was widely accepted by the industry and the academic world. This lead to a wide popularity and huge demand for it. Over the time, a number of open-source clones emerged to compete with it. As it was a proprietary product and charged a hefty licensing fee, even to non-professionals and students. However, the main focus of these clones was to be an alternative to MATLAB. This became a limitation of these clones.

Modern day scientific computation demands an amalgamation between the different fields of science. One such programming language capable of performing the different fields of computation is R-Programming. It is one of the most efficient language for data-mining and statistical analysis. However, R-Programming has not been illustrated as a scientific computation tool. In this paper we have illustrated the computation of Discrete Fourier Transform using R-Programming. DFT is considered as one of the most
efficient and most commonly used DTSP tool. It finds application in almost every signal processing algorithm and is often the first step. In this paper we will be implementing DFT algorithm using elementary coding approach and also using a direct function from the 'signal' package.

II. MATHEMATICAL INTERPRETATION OF DISCRETE FOURIER TRANSFORM

As most of the signals are in analog form i:e the signals are in time domain. For mathematical computation we have to obtain their discrete form, which is done using sampling. Hence, using sampling, we can convert the signal from the time domain to discrete domain. This is mathematically denoted as:

$$x(t) \xrightarrow[\text{(Sampling)}]{t = n\,.\,Ts} x[n]$$

Frequency analysis is convenient to perform on any digital signal processing hardware and is generally preferred. This is obtained using Fourier transform. Thus, the discrete domain signal is converted to frequency domain using Fourier transform.

However, X(W) is a continuous-time representation and not computationally convenient. Therefore, we make use of Discrete-Fourier transform to obtain Frequency samples from the Fourier transform of the discrete-signal. Mathematically, it is represented as:

$$X[K] = \sum_{n=0}^{N-1} x[n].\, e^{\frac{-2\pi nK}{N}}$$

In general terms it is said that DFT provides the K-domain representation of the discrete-signal. The inverse is obtained using Inverse Discrete Fourier transform. Mathematically, it is represented as:

$$X[n] = \frac{1}{N}\sum_{n=0}^{N-1} X[K].\, e^{\frac{2\pi nK}{N}}$$

ISSN 2250-3153

Both these computational tools play a very important role in digital signal processing algorithm applications such as frequency analysis (spectrum analysis) of signals, power spectrum estimations and linear filtering. As there are a large number of computationally efficient algorithm present collectively known as Fast Fourier transform(FFT), they are of efficient practical use.

III. PROGRAMMING LANGUAGES FOR SCIENTIFIC COMPUTATION

Initially, the computation of these algorithm was done using assembly language. This was possible for smaller systems, but as the complexity increased assembly language was found inefficient. Therefore, the use of other high-level languages such a C, C++ was used. However, the most efficient and commonly used programming language for signal processing was MATLAB. It provided a signal processing toolbox, which made it the obvious choice for signal processing application. Most of the signal processing algorithm were present as an inbuilt function. MATLAB is one of the first language to focus mainly on signal processing. Because of its presence in the industry for a very long time, it has become the obvious choice for engineers and students. However, MATLAB is a proprietary software and has a hefty licensing fee. Over the time many open-source programming languages have been developed as an alternative to MATLAB. Software targeted as a complete alternative to it with syntax similar to MATLAB and offering the different functions of MATLAB. Or there are packages developed for specific languages. For example, sci-lab used with python etc. However, all these programming languages like Octave just intend to be an alternative to MATLAB. One of most powerful statistical computation language that has prevailed over the years is R-Programming. R-Programming is primarily a statistical computing programming language. It has found widespread application in statistical computing and has helped in developing many software's for data mining and data analysis. R-programming along with being a powerful statistical computing programming language also has an efficient graphical representation tool. This enables a better visualization of the different outputs of a particular algorithm etc.

IV. USING R-PROGRAMMING

For implementing DFT algorithm using R-Programming, we will have two different approaches. Using the elementary coding approach, we can implement the formulae or we can make use of the direct 'fft' function present in the 'signal' package in R-Programming.

A. *Using Elementary coding approach*

Once the 'signal' package is installed, we need to include it in code.

```
#DFT using Sum Formula and 'fft' Function
library(signal);
```

The discrete sequences is to be provided by the user. For this user provides the input using the scan function and this is stored as vector.

```
#Enter the sequence
xn <- scan();
print("Entered Sequence is ::");
print(xn);
```

In the next step we need to obtain the length of the input sequence provided. The value the length is stored in the variable 'N'. This is needed in the further computation.

```
#Get the length of the sequnce
N = length(xn);
print("Length of the sequence ::")
print(N);
```

According to the formula of DFT we need to perform summation on the formula. The value of the summation is stored in the variable 'sum' and printed for each value of k.

```
#Get DFT by Sum Formula
print("DFT by Sum Formula ::")
for(k in 1:N)
{
  sum = 0;
  for(n in 1:N)
  {
    sum = sum + xn[n]*exp((-1i*2*pi*(n-1)*(k-1))/N);
  }
  print(sum);
}
```

B. *Using Direct function*

R-Programming is packed with a number of packages. Using the 'signal' package DFT of a given sequence can be computed directly using the 'fft' function. This function is similar to 'fft' function in MATLAB. However, in order to use this function, we have to import the 'signal' package in the code, as mentioned before.

```
#Get DFT directly by 'fft' function
print("DFT by 'fft' function ::");
Xk <- fft(xn);
print(Xk);
```

V. OBSERVATIONS

The entire sequence is provided in a stream, followed by a new line stroke. Input provided by the user is as shown below:

```
> #Enter the sequence
> xn <- scan();
1: 1 2 3 4
```

Output i:e the DFT corresponding to the input sequence provided is as shown below:

```
> print(Xk);
[1] 10+0i -2+2i -2+0i -2-2i
```

Output obtained from computation is correct, and is verified when theoretically solved.

VI. CONCLUSION

As the computed result matches with the theoretical result, it is established that R-Programming can be effectively used for other Digital Signal Processing techniques. However, it goes without doubt that, being an open-source project, the support will come over-time. In a short time, different communities will spurge with relevant tutorial and support.

The underlying fact that, a language is available, that can provide you with the same functionality that a proprietary software provides you with no cost and good efficiency, will make it a favorable choice.

R-Programming is not meant merely for signal processing based computations. The different packages provide an expanded horizon, which makes it a preferred choice for integrating Digital Signal Processing with the different datamining techniques.

ACKNOWLEDGMENT

We would like to thank Mr. Yogesh Chandurkar, Assistant Professor (FCRIT, Vashi), for helping us understand the different concepts of Digital Signal Processing and introducing us to R-Programming.

We would also like to thank our colleagues for useful and beneficial criticism, which helped us to shape the paper.

Finally, we would like to express our humble gratitude to the different online communities for sharing their repositories and documentation related to R-Programming and the packages associated with it.

REFERENCES

[1] John G. Proakis and Dimitris G. Manolakis, “Digital Signal Processing: Principles, Algoritms and Applications”, 3rd ed., Prentice-Hall International, New Jersey, USA, 1996.

[2] Alan V. Oppenheim, Ronald W. Schafer with John R. Buck, “Discrete-Time Signal Processing”, 2nd ed., Prentice-Hall International, New Jersey, USA, 1999.

[3] Mark Gardener, “Beginning R: The Statistical Programming Language”, First Edition, 2013.

[4] Hitesh Goyal and Surender Singh, “Big Data Analysis Using R”, IJARCSSE, Volume 5, Issue 9, September 2015.

AUTHORS

First Author – Nadar Prince, Third Year student, Department of Electronics and Telecommunication Engineering, FCRIT, Navi Mumbai, India.
jprince444@gmail.com

Second Author – Abhishek Sengupta, Third Year student, Department of Computer Engineering, FCRIT, Navi Mumbai, India.
sengupta1280@gmail.com

Third Author – Juie Shah, Second Year student, Department of Electronics and Telecommunication Engineering, FCRIT, Navi Mumbai, India.
juieshah@gmail.com

Fourth Author – Mr. Yogesh Chandurkar, Assistant Professor, Department of Electronics and Telecommunication Engineering, FCRIT, Navi Mumbai, India.
ynarayanc@gmail.com

Correspondence Author – Nadar Prince, Third Year student, Department of Electronics and Telecommunication, FCRIT, Navi Mumbai, India.
jprince444@gmail.com