# DSP Project

# Adaptive PCM

Group **9**:

- ➔V. Banu Theja (S20190020258)
- ➔Subash J (S20190020253)
- ➔Surya Sathvik T (S20190020254)

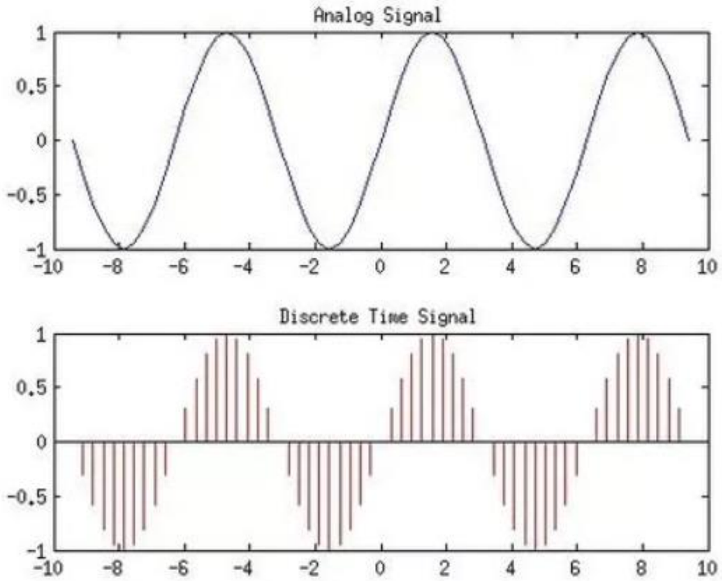
## 1.Background:

Pulse code modulation is a method that is used to convert an analog signal into a digital signal so that a modified analog signal can be transmitted through the digital communication network. PCM is in binary form, so there will be only two possible states high and low (0 and 1). We can also get back our analog signal by demodulation. The Pulse Code Modulation process is done in three steps Sampling, Quantization, and Coding.

APCM is achieved by adapting the quantizing levels to analog signal characteristics. We can estimate the values with the preceding sample values.

## 2.Overview

**Sampling:** Sampling is a process of measuring the amplitude of a continuous-time signal at discrete instants, converts the continuous signal into a discrete signal. For example, conversion of a sound wave to a sequence of samples. The Sample is a value or set of values at a point in time or it can be spaced. Sampler extract samples of a continuous signal, it is a subsystem ideal sampler produces samples that are equivalent to the instantaneous value of the continuous signal at the specified various points.




Sampled signal

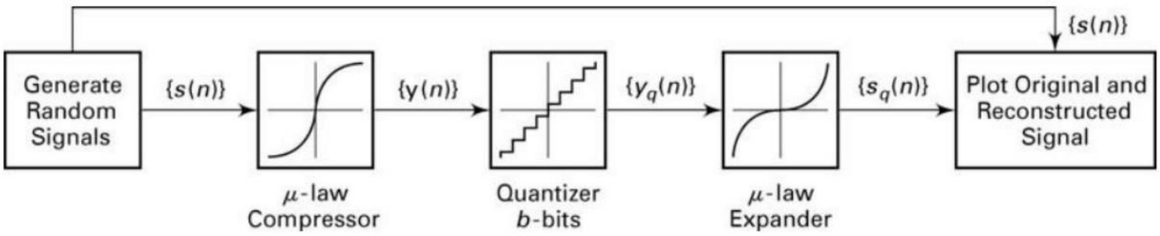
## 3. Problem Statement:

Designing a APCM model which compresses the signal and then quantizes it. The objective of this project is to gain familiarity with, and understanding of, APCM and its interface with a PCM encoder. The APCM encoder is inserted between the PCM compressor and the PCM expander

## 4. PCM:

Stages:

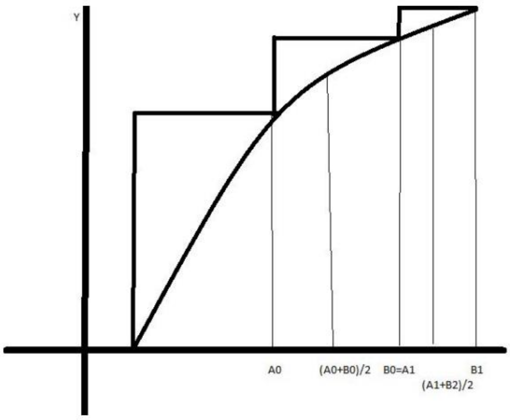
1. U-law Compressor
2. Quantizer
3. U-law Expander




## 5. APCM:

In APCM, the size of the length of quantization with respect to the signal is changed according the previous sampled values

## 6. Adaptive Quantizer




delt_1= B1-A1 = (B0-A0).*f

If y((2x+1)/2) > (y(n)+y(n+1))/2

f = 1.25

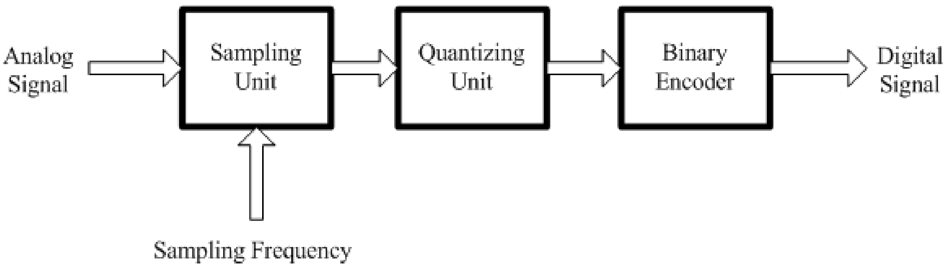
If y((2x+1)/2) = (y(n)+y(n+1))/2

f = 1;

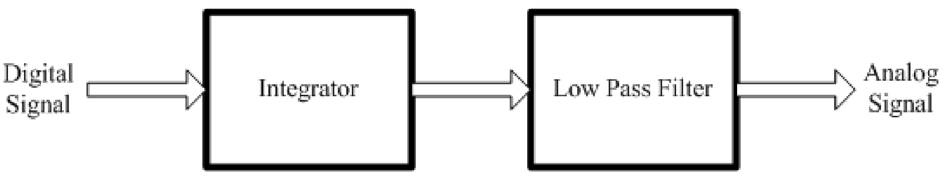
If y((2x+1)/2) < (y(n)+y(n+1))/2

f = 0.5;

7. Encoder and Decoder:




← PCM Encoder

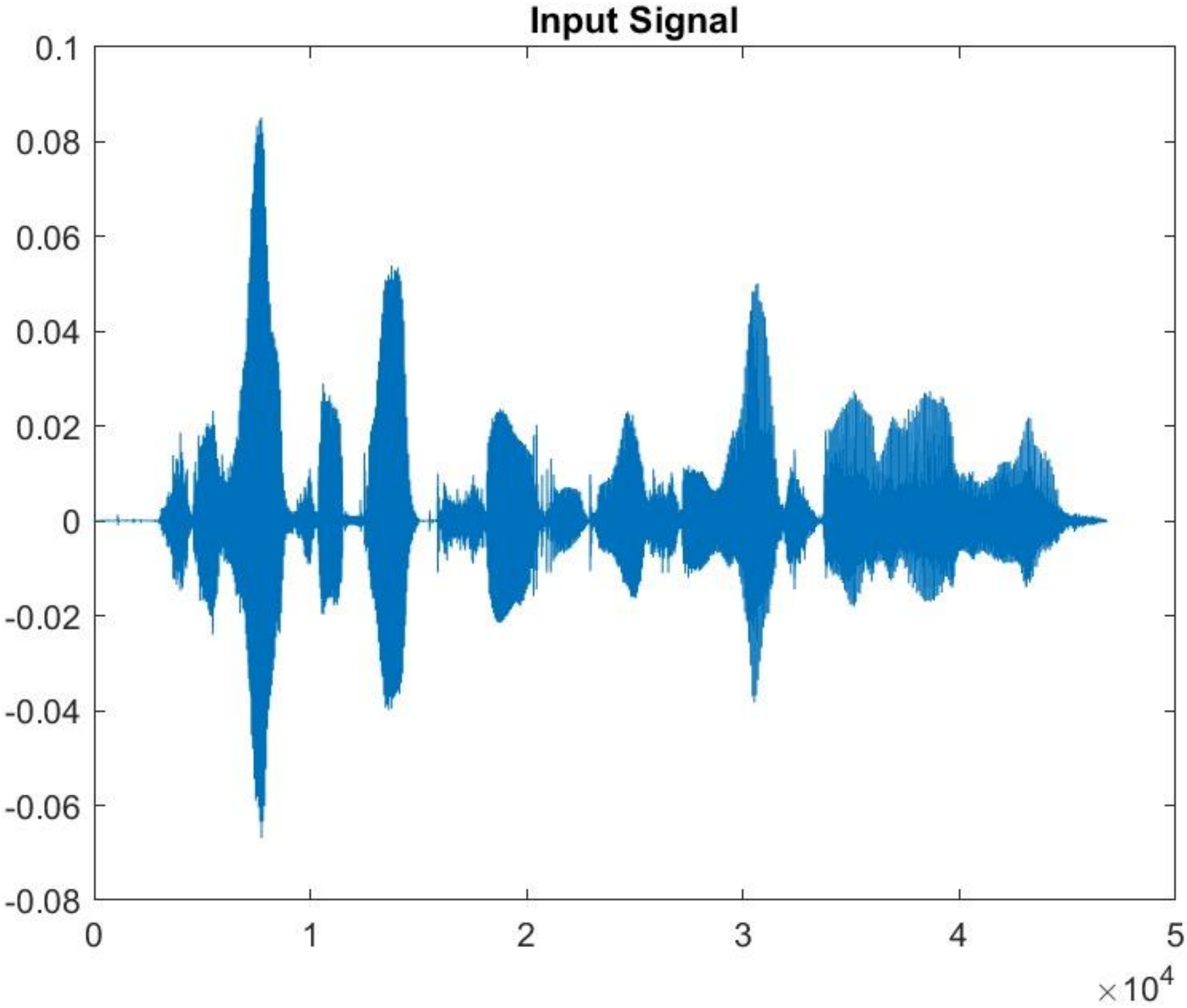


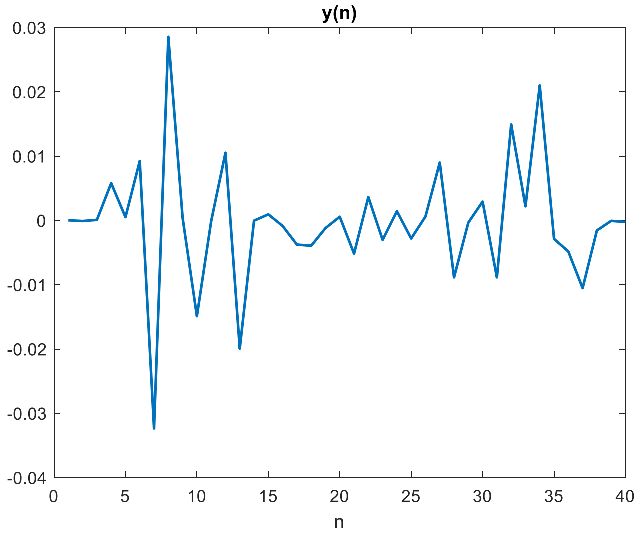

← PCM Decoder

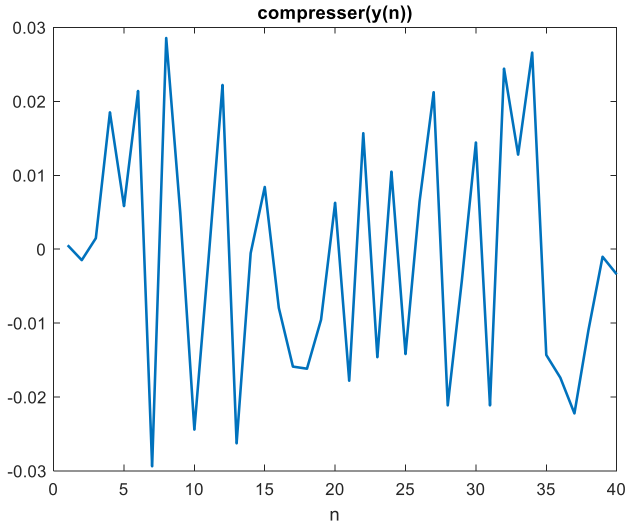
## 10. Outputs:

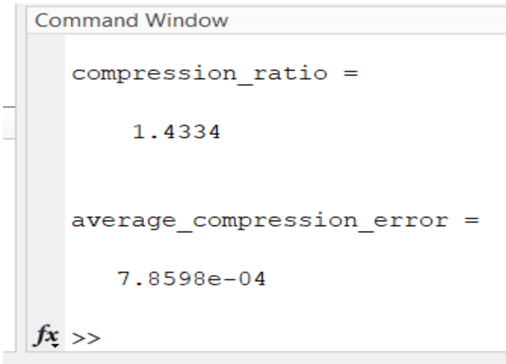
First, we have generated the original signal and then compressed it. Then using APCM technique, we have quantized the signal and then comparing it with the original signal. We did the APCM technique on different signals including an audio signal.

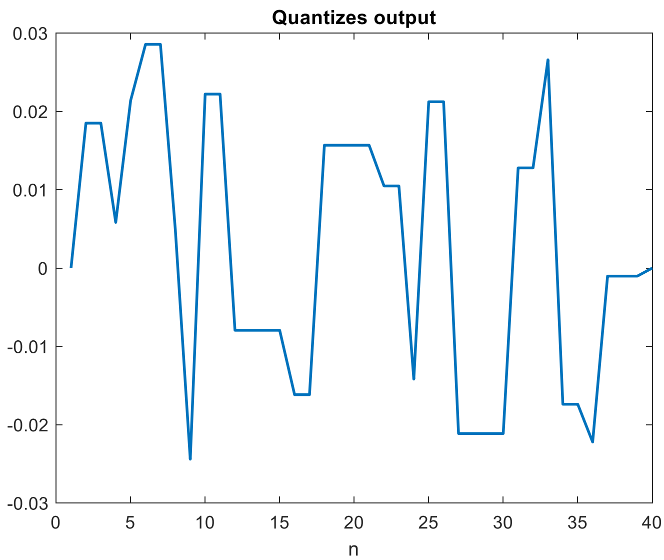
➔ **Audio Signal-**

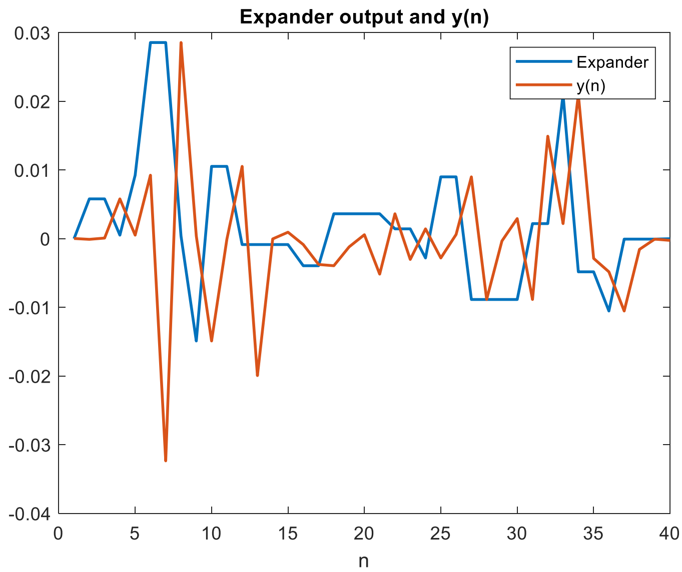



y(n)
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
0
5
10
15
20
25
30
35
40
n

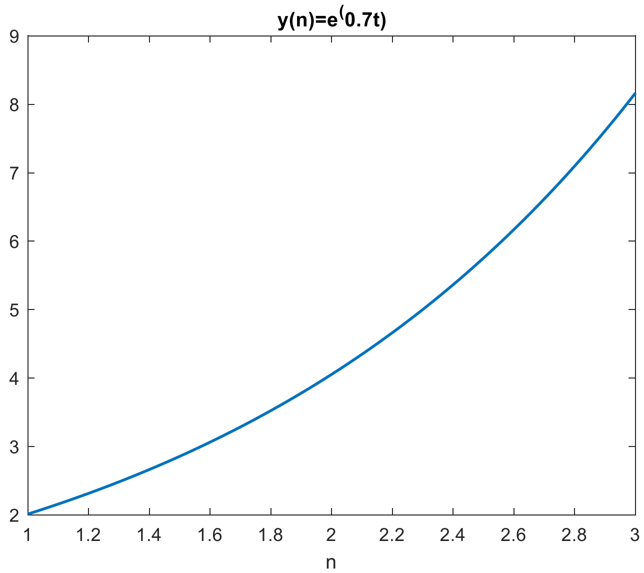

compresser(y(n))
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
0
5
10
15
20
25
30
35
40
n

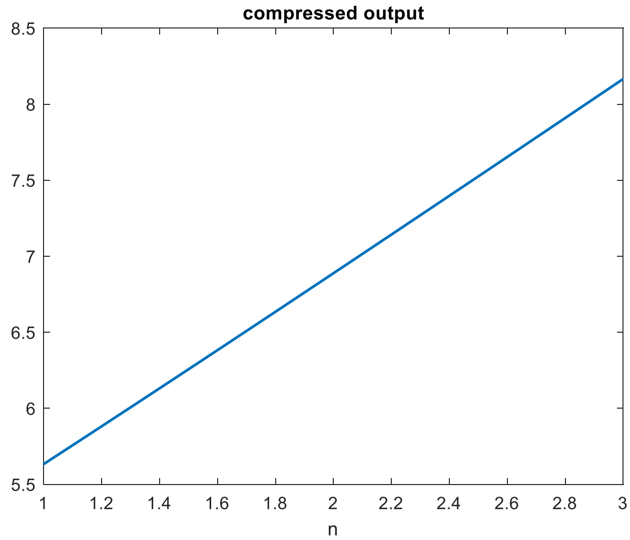

Command Window
compression_ratio =
1.4334
average_compression_error =
7.8598e-04
>>


Quantizes output
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
0
5
10
15
20
25
30
35
40
n

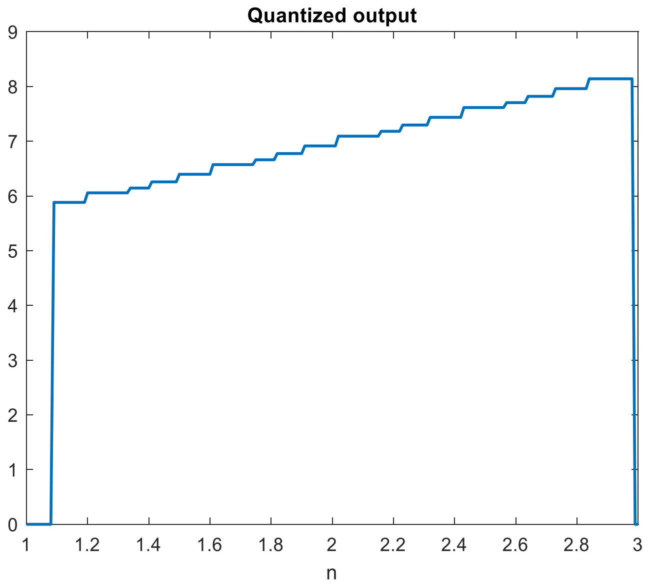

Expander output and y(n)
Expander
y(n)
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
0
5
10
15
20
25
30
35
40
n

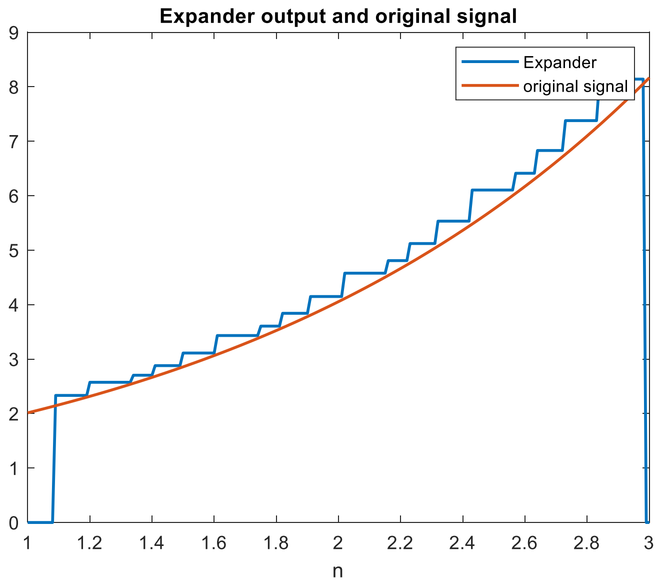
## ➔ y(n) = exp (t)



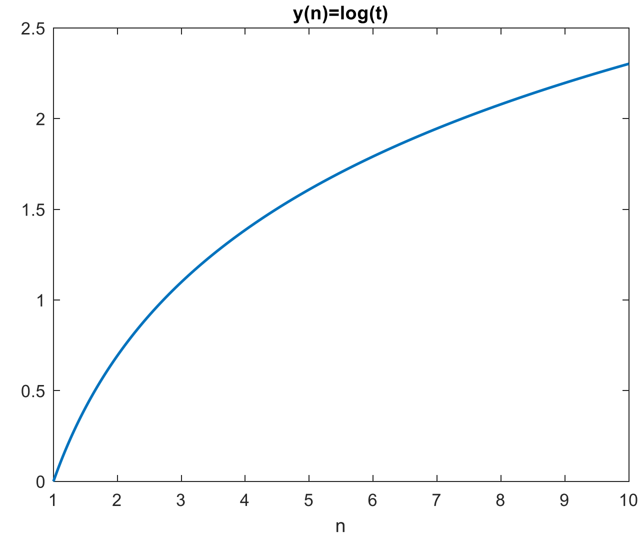



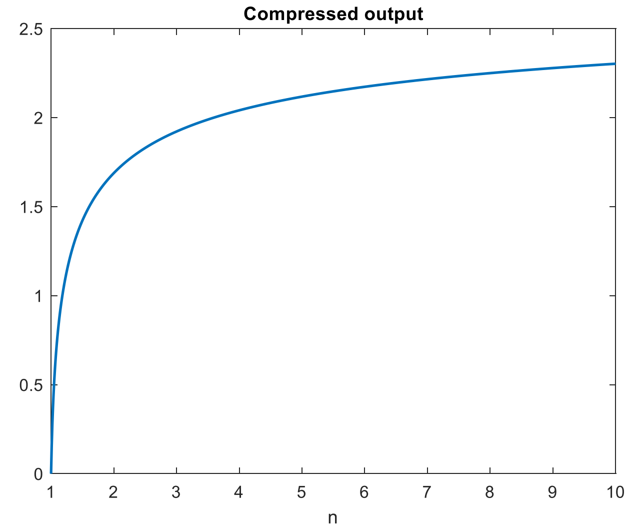

```
Command Window
>> APCM_e_power_t

compression_ratio =

    1.4080


average_compression_error =

   -2.4942

fx >>
```



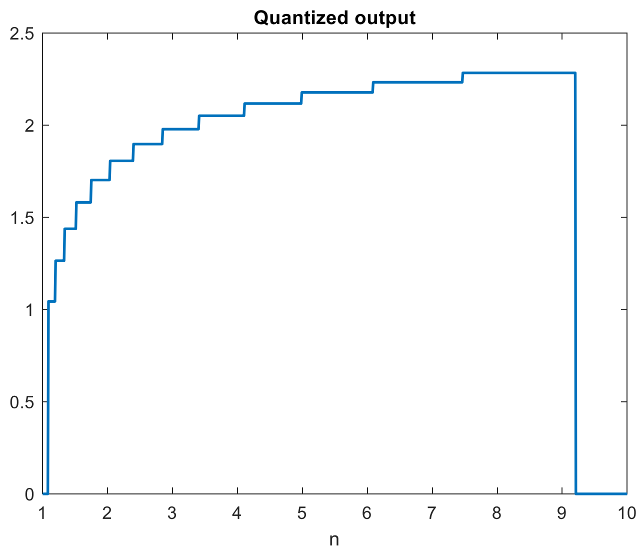



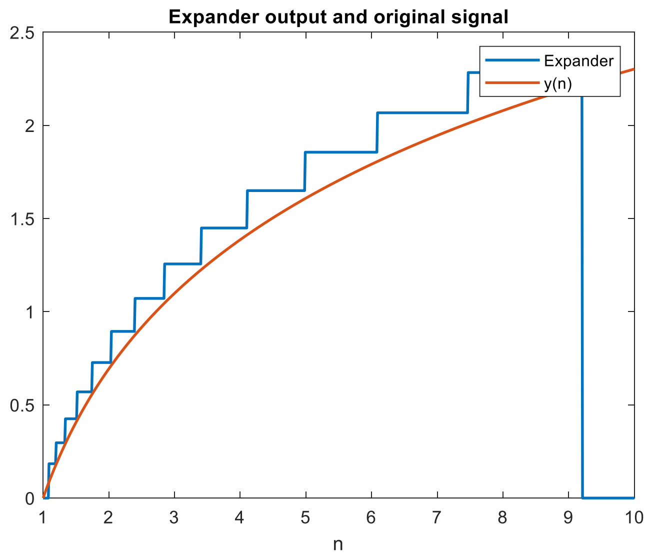
## ➔ y(n) = log(t)



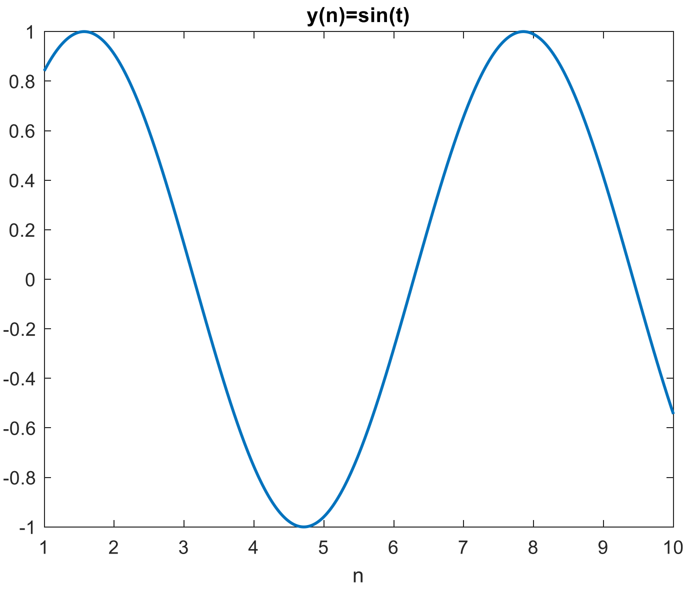



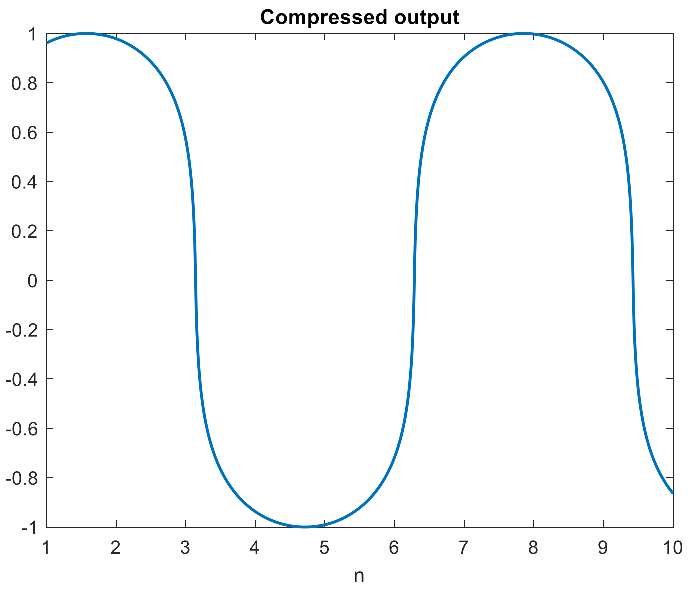

```
Command Window
>> APCM_log_t

compression_ratio =

    1.2059


average_compression_error =

   -0.4771

fx >>
```

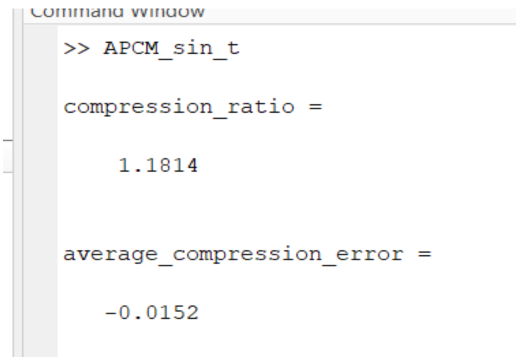


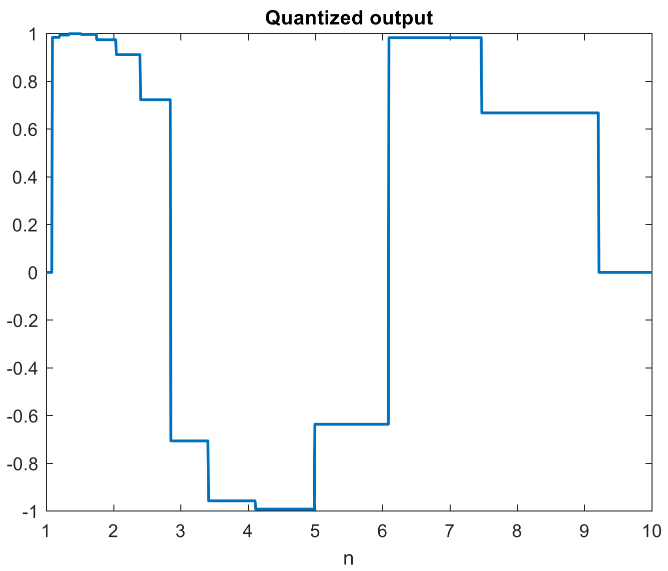
➔ **y(n) = sin(t)**

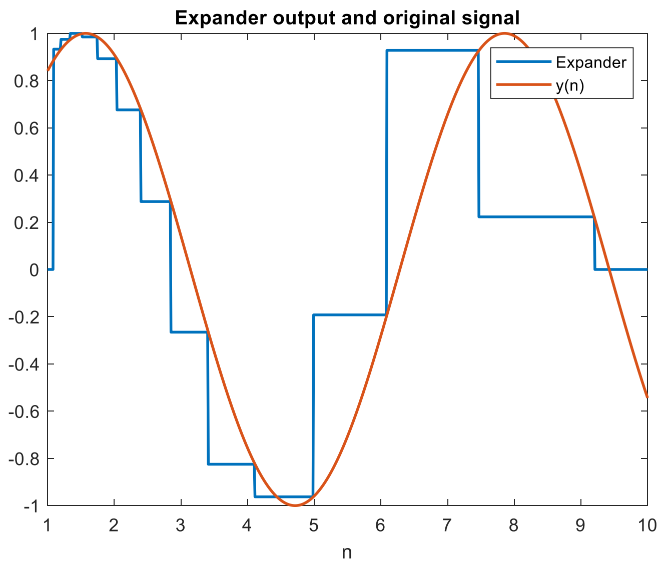


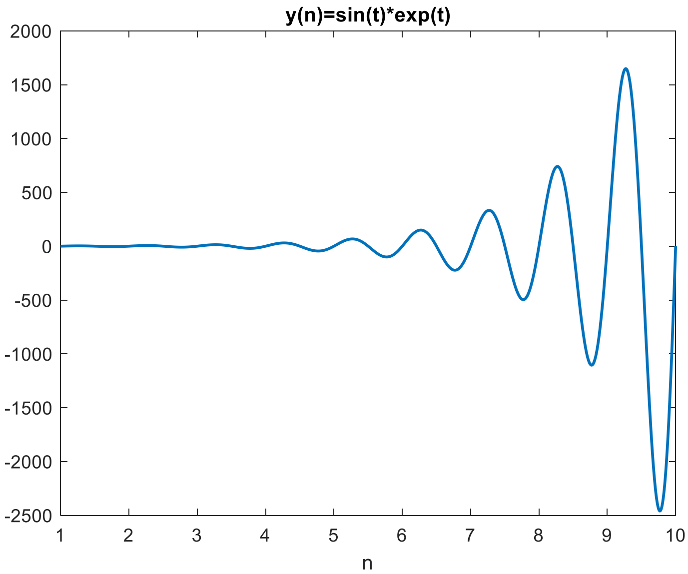



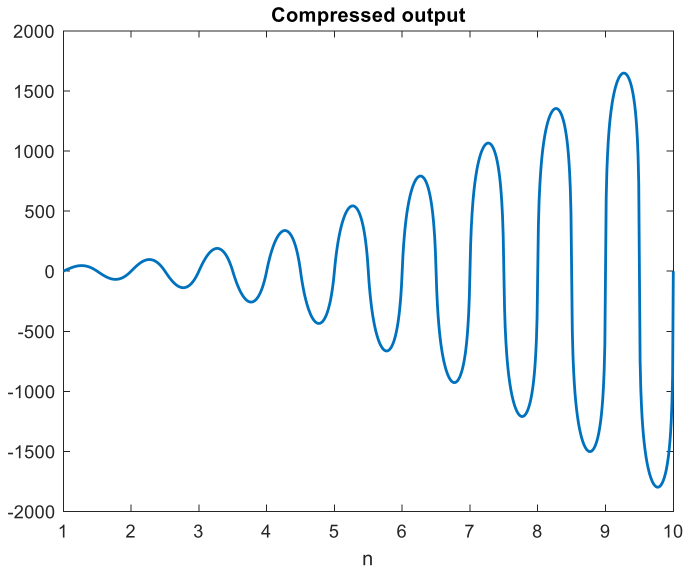

```
>> APCM_sin_t

compression_ratio =

    1.1814


average_compression_error =

   -0.0152
```



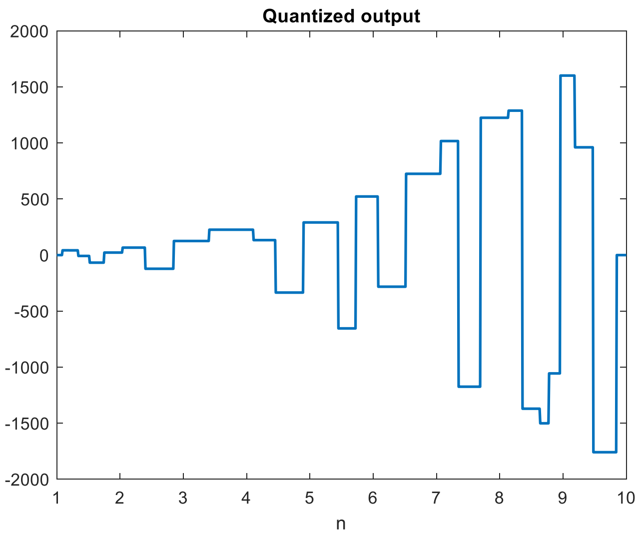



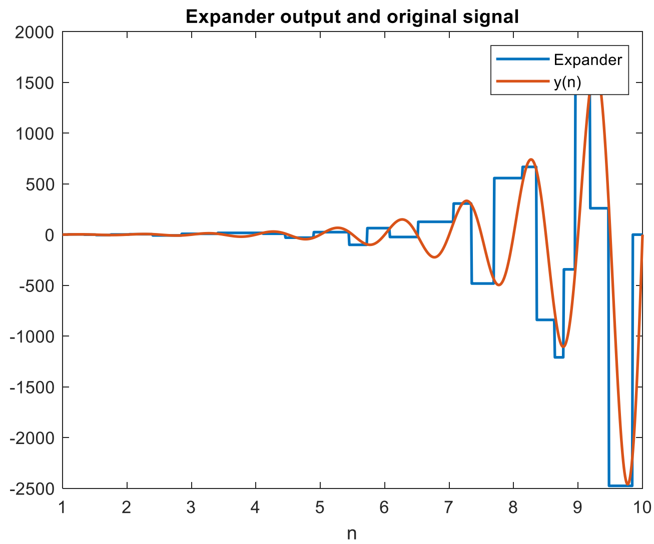
## ➔ y(n) = exp(t)*sin(t)




```
Command Window

compression_ratio =

    1.2076


average_compression_error =

   -7.7562

fx >>
```



**11. Conclusion:** Using adaptive pulse code modulation provide the basic analog-to-digital and digital-to-analog interfaces and represents one possible solution to achieving the goal of an economic and reliable companded PCM.

**12. Contribution:**

**Banu Theja V-** Compression and expansion and APCM for audio signal and exp function

**Subash J-** Quantization algorithm and APCM for sin and exp*sin function

**Surya Sathvik-** Sampling and APCM for log function

**13. Codes drive link:** Link

https://drive.google.com/drive/folders/1mUmMzbEXSEQ-8QesY3ouO3QCinA1PhC2?usp=sharing