

بسم الله الرحمن الرحيم
سید محمد مهدی رضوی
پاییز ۱۴۰۰
گزارش پروژه نظریه اطلاعات

گام سوم :

الف) الفبای منبع ما از نوع Double خواهد بود. و هر کدام ۱۵ رقم اعشار خواهند داشت.

```

Command Window

      Filename: 'E:\Mahdi_Razavi\RazaviSample2.wav'
CompressionMethod: 'Uncompressed'
      NumChannels: 1
      SampleRate: 44100
TotalSamples: 80641
      Duration: 1.8286
      Title: 'Recording (3)'
      Comment: []
      Artist: []
BitsPerSample: 16

```

با توجه به جدول فوق

$$۱۶ * ۸۰۴۶۱ = ۱,۲۹۰,۵۷۶$$

بیت برای ذخیره این سیگنال صوتی که در آن فقط بسم الله الرحمن الرحيم گفتم نیاز خواهیم داشت.

ب) سرعت تولید سمبل های ما با توجه به جدول فوق که با استفاده از `audioinfo` به دست آورده ایم

همان فرکانس خواهد بود که برابر ۴۴۱۰۰ خواهد بود.

کل زمانی که طول می کشد که ما صدا را بشنویم را می توانیم از تقسیم دو مقدار کل سمبل ها بر فرکانس بدست بیاریم.

اما با توجه به جدول فوق به سادگی درمی یابیم ۱,۸۲۸۶ ثانیه طول خواهد کشید.

$$\text{Duration} = \text{TotalSamples} / \text{SampleRate}$$

$$1.8286 = 80641 / 44100$$

ج) این مساله مربوط به قضیه نایکویست و تفاوت نوع نمونه گیری در این دو وسیله خواهد بود.

نوار کاست چون که به صورت پیوسته و درهر آن می چرخد ، سیگنال آنالوگ ما را به صورت آنالوگ نمونه برداری و ذخیره خواهد کرد.

اما در دستگاه mp3 player چون قضیه نایکویست برقرار است و فرکانس نمونه گیری بیشتر از دو برابر پهنای باند می باشد ، تمام سیگنال

آنالوگ را به صورت دیجیتال نمونه گیری خواهد کرد.

علت اصلی آن این است که فاصله زمانی موج های ارسالی آنقدر کم است که گوش ما قادر به تفکیک آن ها از یکدیگر نیست و میتواند موج اصلی را مطابق نایکویست بازپایی کند.

گام چهارم :

```
h = histogram(y, symbols);  
h.Normalization = 'probability';  
probabilities = h.Values;  
probabilities(end + 1) = 0;
```

با استفاده از کدهای فوق توانستیم نمودار هیستوگرام مربوط به سیگنال صوتی را رسم کنیم. سپس لیست احتمالات را به دست خواهیم آورد.

ب) با استفاده از قضیه اول شانن حداکثر میزان فشردگی سیگنال را به دست خواهیم آورد.

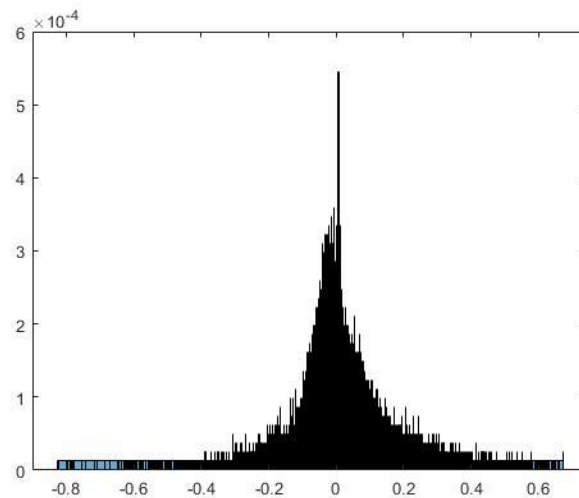
$$H(Y) = -\sum p(Y) * \log p(Y) = 1.335662e+01 = 13.35662$$

$$N * H(Y) = 80641 * 13.35662 = 1.077091e+06 = 1077091$$

طبق قضیه اول شانن تا میزان ۱۰۷۷۰۹۱ بیت می توان این سیگنال را فشرده سازی نمود.

این نتیجه به دست آمده منطقی نمی باشد زیرا که حجم فایل اصلی ما ۱,۲۶۴,۰۰۰ بیت خواهد بود.

ما می توانیم به گونه ای اطلاعات را کمپرس کنیم که اطلاعات از بین برود اما ما می خواهیم که با بخشی از اطلاعات بقیه اطلاعات را بازیابی کنیم.



گام پنجم :

```
dict = huffmandict( symbols , probabilities');  
code = huffmanenco(y(:,1) , dict);  
  
decodedy = huffmandeco(code ,dict);
```

الف) با استفاده از کدهای فوق توانستیم که این سیگنال را کدگذاری کنیم.

سایز فایل اولیه ۱۵۸ کیلوبایت است ولی سایز فایل کمپرس شده برابر ۲۱۰۹ کیلوبایت است.!!!!

با استفاده از کد هافمن مشاهده می کنیم که حجم فایل اولیه پس از کمپرس شدن ۱۰ برابر می شود. طبیعتا متوجه می شویم که این روش کدگذاری برای سیگنال های صوتی مناسب نیست.

چون در درجه اول تعداد سمبل های داده ما بسیار زیاد است.

در درجه دوم هم اینکه پراکندگی احتمالاتی این سمبل ها با توجه به نمودار هیستوگرام ما خیلی متفاوت نیست که تاثیر بسزایی در کدگذاری ما داشته باشد.

در درجه سوم به این علت است که هافمن کد کننده بدون اتلاف داده است بدین معنا که داده ها را کاملا از نو و با یک تغییر در چیدمان سمبل ها می سازد. در این سیگنال بهتر بود از روش های کدگذاری با اتلاف استفاده می کردیم.

ب)

$$(2109 * 8)^{Kbit} / 64^{Kbit/sec} = 263.625 \text{ sec}$$