دانشکده مهندسی کامپیوتر

انتقال داده

فرزان رحمانی

استاد: ابوالفضل دیانت

نیم‌سال دوم ۱۴۰۲-۱۴۰۱

فهرست مطالب

۱ گام اول

در پروژه قبلی متلب را نصب کرده بودیم.

۲ گام دوم

از اینترنت یک فایل wav پیدا می کنیم و با استفاده از کد های داده شده در جزوه آن را خوانده و دوباره ذخیره می کنیم.

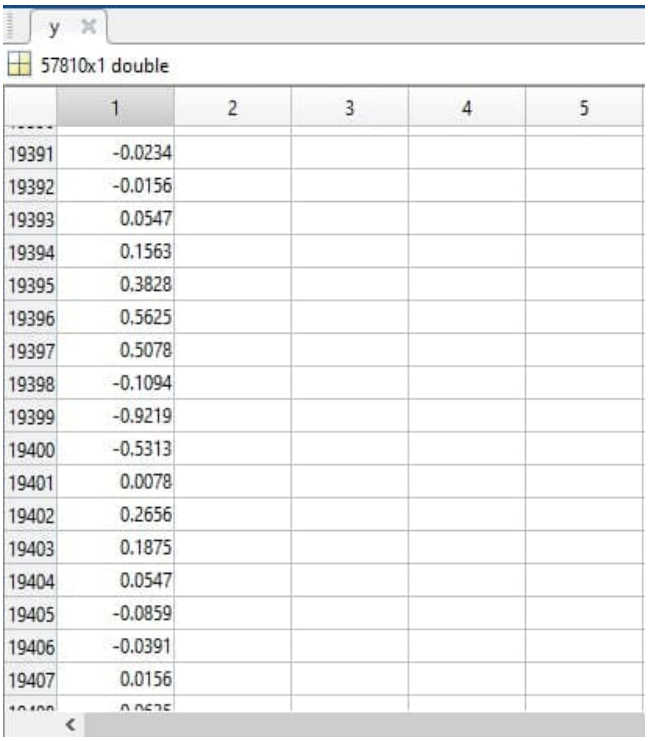
```
[y, Fs] = audioread('charge.wav');

player = audioplayer(y,Fs);
play(player);

audiowrite('handle.wav',y,Fs)
```

۳ گام سوم

- الفبای منبع در حالت ذکر شده چیست؟ همان طور که در تصویر مشخص است، y از اعداد اعشاری تا ۴ رقم اعشار تشکیل شده است.



y

57810x1 double

	1
19391	-0.0234
19392	-0.0156
19393	0.0547
19394	0.1563
19395	0.3828
19396	0.5625
19397	0.5078
19398	-0.1094
19399	-0.9219
19400	-0.5313
19401	0.0078
19402	0.2656
19403	0.1875
19404	0.0547
19405	-0.0859
19406	-0.0391
19407	0.0156

- سرعت تولید سمبل در منبع ذکر شده چه مقدار است ؟

$$F_s = \frac{|y|}{t} \qquad \rightarrow t = \frac{|y|}{F_s}$$

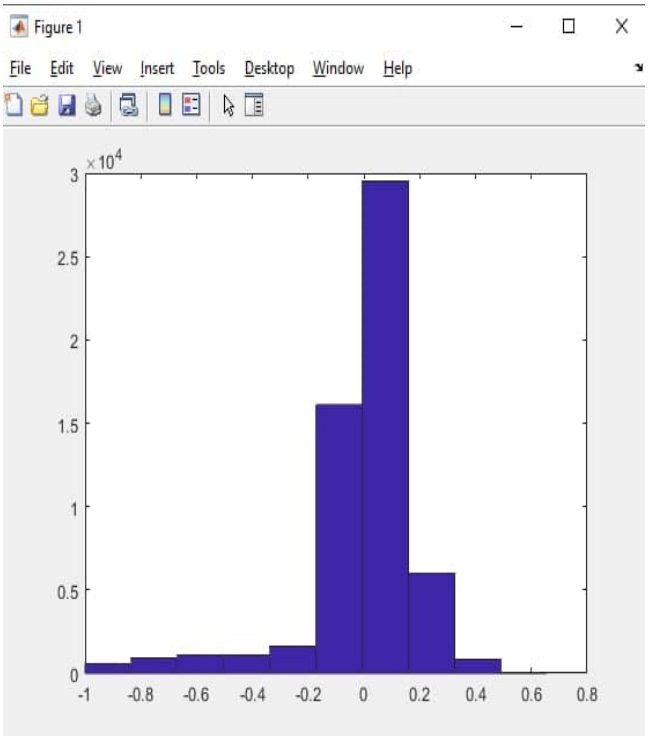
می دانیم که طول y برابر با 57810 است و مقدار F_s را نیز در دست داریم، پس با استفاده از روابط بالا $t = 5.2435$ به دست می آید که با توجه به خود فایل عدد به دست آمده درست است.

- اگر به همان فایل $handle$ گوش کنید شما چنین قطعی در صدا نمی یابید. چرا؟ چرا در هنگام تماشای یک فایل ویدئویی با این که تنها ۲۰تا ۳۰فریم در ثانیه پخش می شود، ولی شما هیچ گونه گسستگی در فیلم مشاهده نمی کنید؟ با توجه به قضیه

اگر از یک سیگنال باند محدود، با دو برابر نرخ نایکویست نمونه برداری کنیم، می توانیم به طور کامل از روی نمونه ها سیگنال پیوسته را بازیابی کنیم پس اگر نرخ نمونه برداری 2 برابر f_{max} باشد، بدونه هیچ گونه گسستگی شنیده می شود.

۴ گام چهارم

- ابتدا هیستوگرام y را رسم می کنیم.



- سپس با استفاده از اطلاعاتی که از هیستوگرام استخراج می کنیم، انتروپی این منبع را بدست می آوریم.

```
bins=hist(y)
p=bins/sum(bins)
Hx=-sum(p.*log2(p))
```

خروجی کد بالا $Hx = 1.9656$ شد.

- طبق قضیه اول شانون n متغیر تصادفی با انتروپی $H(x)$ را می توان به $nH(x)$ بیت بدون از دست دادن اطلاعات فشرده ساخت.

```
e=length(y)*Hx
```

خروجی کد بالا $e = 1.1363e + 05$ شد. یعنی این فایل را تا 113 کیلوبایت می توان فشرده کرد.

- به نظر شما نتیجه بدست آمده معقول است؟ اگر نیست چرا؟ تحقیق کنید و ببینید آیا می توان به مرزهای واقعی تری برای فشرده سازی دست یافت؟ نه معقول نیست، زیرا سمبل ها مستقل در نظر گرفته شدند ولی در واقعیت اینگونه نیست. همانند مثال کلمات، بعضی کلمات در کنار هم ظاهر نمی شوند یا برعکس و یا احتمال رخداد یک کلمه خیلی بیشتر از بعضی کلمات دیگر باشد.

۵ گام پنجم

ابتدا مقادیر $unique$ موجود در y را به دست می آوریم، سپس با استفاده از هیستوگرام تعداد هر کدام را به دست می آوریم و در نهایت احتمال هر کدام را محاسبه می کنیم.

```
symbols=unique(y)
bins = hist(y,symbols)
p=bins/length(y)
dict = huffmandict(symbols,p);
code = huffmanenco(y,dict);
```

در نهایت کد هافمن را تست می کنیم و با توجه به کد زیر به درستی آن پی می بریم.

```
sig = huffmandeco(code,dict);
isequal(y,sig);
```

محاسبه کنید که برای انتقال این فایل در یک لینک مخابراتی با سرعت $64kbit/s$ چه مقدار زمان لازم است؟ ابتدا کد هافمن به دست آمده را به فایل تبدیل می کنیم.

```
audiowrite('handle_code.wav',code,Fs);
```

حجم این فایل $634kB$ شد و اگر این مقدار را تقسیم بر $64kbit/s$ کنیم، عدد $80s$ به دست می آید.