



به نام خدا
دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده مهندسی برق و کامپیوتر



سیگنال‌ها و سیستم‌ها

نیمسال دوم (99-00)

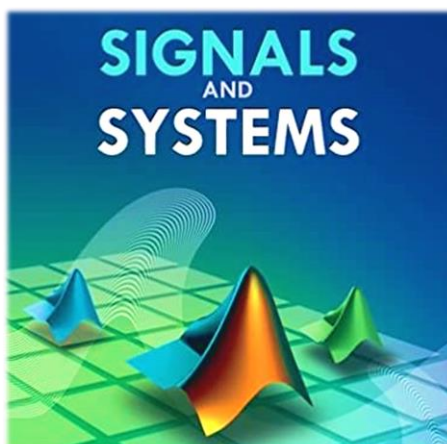
استاد: دکتر سعید اخوان

تمرین کامپیوتر 2

*Synthesis and Analysis
DTMF Signaling*

محمد مهدی عبدالحسینی

810 198 434



Signals and Systems

فہرست مطالب

بخش اول: سنٹر 1

بخش دوم: آنا لیر 2

بخش اول : سنتز

* m-file مربوط به این قسمت با نام synthesis.m پیوست شده است.

```

1 - clc; clear; close all;
2 - %% Initial Values
3 - fL = [697 770 852 941]; % Low Frequency Group
4 - fH = [1209 1336 1477]; % High Frequency Group
5 - fs = 8000; % Signal Sampling Frequency
6 - Ts = 1/fs; % Signal Sampling Time
7 - Ton = 0.1; % ON Time for each DTMF Signal (in second)
8 - Toff = 0.1; % OFF Time ; The Gap Between DTMF Signals (in second)
9 - t = 0:Ts:Ton;
10 - %% Enter a Number
11 - number = (num2cell(input("PLEASE ENTER A NUMBER TO SYNTHESIS : ", 's')));
12 - % now we have each number in a separate cell

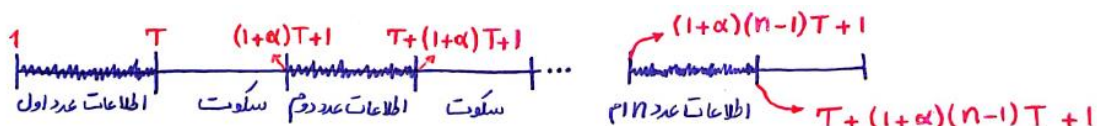
fx PLEASE ENTER A NUMBER TO SYNTHESIS : 810198434

13 - %% More Variables for Loop
14 -
15 - [~, N] = size(number); % How many digits do we have ?
16 - [~, T] = size(t);
17 -
18 - t = t(2:T); % Drop Zero Value of t
19 -
20 - T = T - 1; % T Equals to New Size of t
21 -
22 - alpha = round(Toff/Ton); % We need this to Calculate Silence Times (OFF Times)
23 -
24 - % Define Signals :
25 - yL=0; % Low Frequency Signal
26 - yH=0; % High Frequency Signal
27 - y=zeros(1, N * (1 + alpha)*T); % Final Encoded Signal

28 - %% For Loop
29 - for n = 1:N
30 -     L=0;H=0;
31 -     if strcmp(number(n), {'1'}) || strcmp(number(n), {'2'}) || strcmp(number(n), {'3'})
32 -         L = 1;
33 -     elseif strcmp(number(n), {'4'}) || strcmp(number(n), {'5'}) || strcmp(number(n), {'6'})
34 -         L = 2;
35 -     elseif strcmp(number(n), {'7'}) || strcmp(number(n), {'8'}) || strcmp(number(n), {'9'})
36 -         L = 3;
37 -     elseif strcmp(number(n), {'*'}) || strcmp(number(n), {'0'}) || strcmp(number(n), {'#'})
38 -         L = 4;
39 -     end
40 -     yL = sin(2*pi*fL(L)*t); % Low Frequency Signal
41 -     %=====
42 -     if strcmp(number(n), {'1'}) || strcmp(number(n), {'4'}) || strcmp(number(n), {'7'}) || strcmp(number(n), {'*'})
43 -         H = 1;
44 -     elseif strcmp(number(n), {'2'}) || strcmp(number(n), {'5'}) || strcmp(number(n), {'8'}) || strcmp(number(n), {'0'})
45 -         H = 2;
46 -     elseif strcmp(number(n), {'3'}) || strcmp(number(n), {'6'}) || strcmp(number(n), {'9'}) || strcmp(number(n), {'#'})
47 -         H = 3;
48 -     end
49 -     yH = sin(2*pi*fH(H)*t); % High Frequency Signal
50 -     %=====
51 -     % To calculate Final Signal we need to know Period of each Cycle;
52 -     % it means when do we have ON time and OFF time and next ON-OFF times ?
53 -     % This Code generates ON time Values at the beginning of the cycle
54 -     % and in the next Loop the next cycle begins with a gap between two cycles
55 -     y((1 + alpha)*(n - 1)*T + 1 : T + (1 + alpha)*(n - 1)*T) = (yL + yH)/2;
56 - end

```

محتوای خط 55 بصورت زیر بدست آمده است :



```

57 %% Save and Play
58 audiowrite("y.wav",y,fs);
59 sound(y,fs);

```

بخش دوم: آنالیز

* m-file مربوط به این قسمت با نام analysis.m پیوست شده است.

ابتدا یک منبع اطلاعاتی از کیبورد میسازیم تا بتوانیم در ادامه سیگنال مجهول را با آن مقایسه کنیم.

```

1 - clc; clear; close all;
2 - [a,fs] = audioread("a.wav");
3 - sound(a,fs);
4 - %% Generate Source Signals
5 - fL = [697 770 852 941]; % Low Frequency Group
6 - fH = [1209 1336 1477]; % High Frequency Group
7 - Ts = 1/fs; % Signal Sampling Time
8 - Ton = 0.1; % ON Time for each DTMF Signal (in second)
9 - Toff = 0.1; % OFF Time ; The Gap Between DTMF Signals (in second)
10 - t = 0:Ts:Ton;
11 - [~, T] = size(t);
12 - t = t(2:T); % Drop Zero Value of t
13 - T = T - 1; % T Equals to New Size of t
14 - alpha = round(Toff/Ton); % we need this to Calculate Silence Times (OFF Times)
15 - % Source Signals :
16 - y = zeros(4,3,T);
17 - for L = 1:4
18 -     yL = sin(2*pi*fL(L)*t); % Low Frequency Signal
19 -     for H = 1:3
20 -         yH = sin(2*pi*fH(H)*t); % High Frequency Signal
21 -         y(L, H, :) = (yL + yH)/2; % Source Signal
22 -     end
23 - end

```

در مرحله بعد کار را بر روی سیگنال مجهول آغاز میکنیم. در قدم اول یک ماتریس دوبعدی از داده‌های غیرسکوت میسازیم که سطرهای آن داده‌های هر رقم است و ستون‌های آن نشان‌دهنده شماره هر رقم است. (شماره هر رقم یعنی اینکه داده‌های هر سطر مربوط به بوق چندم در صوت میباشد.)

```

24 %% Extracting "a.wav" T(on) ?
25 [aSize, ~] = size(a);
26 numOfNumbers = (aSize / T) * 1/2; % 1/2 = Ton/(Ton + Toff)
27 x = zeros(T, numOfNumbers);
28 for n = 1:numOfNumbers
29     % Drop Gaps (OFF Times)
30     x(:, n) = a((1 + alpha)*(n - 1)*T + 1 : T + (1 + alpha)*(n - 1)*T);
31 end

```

در ادامه سعی داشتم ضریب همبستگی را بطور کلی بدست بیاورم که به این نتیجه رسیدم داده‌های مخرج تقریباً به ازای همه ورودی‌ها ثابت می‌مانند. بنابراین تنها به محاسبه $\sum x[n]y[n]$ اکتفا کردم.

```

32 %% r = Correlation Coefficient (x,y)
33 % y2n = sum(y.^2,3);
34 % y2n approximately for all keyboard is the same ; so we don't need to use it for finding r(max)
35 xyn = zeros(4,3,numOfNumbers); % xyn = sigma x[n]*y[n]
36 for n = 1:numOfNumbers
37     for L = 1:4
38         for H = 1:3
39             xyn(L, H, n) = sum((x(:, n)') .* reshape(y(L,H,:), [1,800]), 2);
40         end
41     end
42 end

```

برای اینکه بیشتر با ساختار xyn ، که ادعا میکنیم برای یافتن $r(\max)$ کافیت آشنا بشیم، پیشنهاد میکنم به ارقام ماکزیمم در ماتریس xyn توجه کنید و آن را با عدد پرینت شده در خروجی مقایسه کنید.

عدد اول	val(:, :, 1) =			عدد چهارم	val(:, :, 4) =		
	-0.0375	41.3993	0.0111		119.3213	70.7959	70.8744
	0.0710	41.5078	0.1196		48.3267	-0.1987	-0.1202
	65.1534	106.5902	65.2020		48.3606	-0.1648	-0.0863
	-0.1972	41.2396	-0.1486		48.3356	-0.1897	-0.1112
عدد دوم	val(:, :, 2) =			عدد پنجم	val(:, :, 5) =		
	119.3213	70.7959	70.8744		-0.0739	-0.1644	33.1739
	48.3267	-0.1987	-0.1202		-0.0371	-0.1276	33.2107
	48.3606	-0.1648	-0.0863		65.1337	65.0432	98.3815
	48.3356	-0.1897	-0.1112		-0.2281	-0.3186	33.0198
عدد سوم	val(:, :, 3) =			عدد ششم	val(:, :, 6) =		
	-0.0402	41.4309	0.0368		-0.0375	41.3993	0.0111
	-0.0901	41.3810	-0.0131		0.0710	41.5078	0.1196
	-0.1596	41.3115	-0.0826		65.1534	106.5902	65.2020
	61.3907	102.8618	61.4677		-0.1972	41.2396	-0.1486

با بررسی داده‌ها در ماتریس xyn میتوان گفت تقریباً کار تمام است و فقط سه قدم زیر باقی می‌ماند:

- ✓ قدم اول: یافتن ماکزیمم در ماتریس xyn .
- ✓ قدم دوم: نسبت دادن عددی در کیبورد به آن که در همان جایگاه قرار دارد.
- ✓ قدم سوم: پرینت عدد نهایی.

```

43 %% Find The Number from r(max)
44 number = -1*ones(1,numOfNumbers);
45 maxValue = max(xyn,[],[1 2]); % xyn(max) ==> r(max) :)
46 maxIndex = mod(find(xyn == maxValue),12);
47 for n = 1:numOfNumbers
48     if maxIndex(n) == 1
49         number(n) = 1;
50     elseif maxIndex(n) == 2
51         number(n) = 4;
52     elseif maxIndex(n) == 3
53         number(n) = 7;
54     elseif maxIndex(n) == 4
55         number(n) = '*';
56     elseif maxIndex(n) == 5
57         number(n) = 2;
58     elseif maxIndex(n) == 6
59         number(n) = 5;
60     elseif maxIndex(n) == 7
61         number(n) = 8;
62     elseif maxIndex(n) == 8
63         number(n) = 0;
64     elseif maxIndex(n) == 9
65         number(n) = 3;
66     elseif maxIndex(n) == 10
67         number(n) = 6;
68     elseif maxIndex(n) == 11
69         number(n) = 9;
70     elseif maxIndex(n) == 0
71         number(n) = '#';
72     end
73     if number(n) < 10
74         number(n) = number(n) + 48; % Using ASCII Table
75     end
76 end

```

```

77 disp(['ANALYSIS RESAULT : ' char(number)]);

```

```
ANALYSIS RESAULT : 810198
```

```
fx >>
```

در مرحله چاپ خروجی اندکی با مشکل مواجه شدم. مشکل این بود که هنگام چاپ '*' و '#', بجای چاپ خودشان، عدد آسکی آنها را در خروجی چاپ میکرد. این مشکل با چاپ خروجی به شیوه خط 77 به مشکل جدیدی منجر شد. مشکل جدید این بود که این بار کاراکتر آسکی را بجای اعداد 0 تا 9 چاپ میکرد. این مشکل با مراجعه به جدول آسکی و قطعه کد در خطوط 73 تا 75 حل شد.

پروژه واقعاً زیبا و مفید بود.

ممنون از طراح

MJP