

به نام خدا



دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

تمرین کامپیوتری ۲

سیگنال ها و سیستم ها

دکتر اخوان

عرفان پناهی ۸۱۰۱۹۸۳۶۹

بهار ۱۴۰۰

فهرست

بخش الف (سنتز)..... صفحه ۲ ([لینک](#))

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

*** خروجی مربوط به این قسمت با نام `y.wav` پیوست شده است.

بخش ب (آنالیز)..... صفحه ۳ ([لینک](#))

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

بخش الف : سنتز

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

- اسکریپت نوشته شده برای این قسمت به صورت زیر است:

```
function synthesis(input_str)
    keypad = ['1' '2' '3';
              '4' '5' '6';
              '7' '8' '9';
              '*' '0' '#'];
    fr = [697 770 852 941];
    fc = [1209 1336 1477];
    fs = 8000;
    Ts = 1 / fs;
    Ton = 0.1; % Ton = Toff
    t = Ts:Ts:Ton;
    off = 0 * t;
    y_final = [];
    for k = input_str
        [i,j] = find(keypad == k);
        y1 = sin(2 * pi * fr(i) * t);
        y2 = sin(2 * pi * fc(j) * t);
        y = (y1 + y2) / 2;
        y_final = [y_final y off];
    end
    sound(y_final,fs);
    audiowrite('y.wav',y_final,fs);
end
```

- حال به عنوان ورودی ، شماره دانشجویی (۸۱۰۱۹۸۳۶۹) را به این تابع می دهیم:

```
>> synthesis('810198369')
```

*** خروجی مربوط به این قسمت با نام **y.wav** پیوست شده است.

بخش ب : آنالیز

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

توضیح عملکرد: همانطور که در توضیحات صورت پروژه نیز گفته شده است، ابتدا بازه های زمانی T_{on} را جدا کرده و از قسمت های T_{off} پرش می کنیم (گام حلقه را به این صورت تعیین میکنیم). سپس به کمک معیار همبستگی، سعی می کنیم بیشترین شباهت بین سیگنالی که از T_{on} گرفته شده و سیگنال های ناشی از هر کلید را پیدا کنیم. در نهایت کلیدی که بیشترین شباهت را دارد انتخاب کرده و سراغ T_{on} بعدی می رویم.

- اسکریپت نوشته شده برای این قسمت به صورت زیر است:

```
function code = analysis(filename)
    keypad = ['1' '2' '3';
              '4' '5' '6';
              '7' '8' '9';
              '*' '0' '#'];
    [x,fs] = audioread(filename);
    fr = [697 770 852 941];
    fc = [1209 1336 1477];
    Ts = 1/fs;
    Ton = 0.1;
    t = Ts:Ts:Ton;
    off = 0*t;
    r = [];
    code = [];
    for k = 1:2*length(t):length(x)
        new = x(k:k+length(t)-1);
        new = transpose(new);
        for j = 1:3
            for i = 1:4
                y1 = sin(2*pi*fr(i)*t);
                y2 = sin(2*pi*fc(j)*t);
                y = (y1 + y2)/2;
                r(i,j) = sum(new.*y)/(sum(new.^2)*sum(y.^2))^0.5;
            end
        end
        ind_key = r == max(max(r));
        key = keypad(ind_key);
        code = [code key];
    end
end
```

- حال به عنوان ورودی ، برای اطمینان از صحت عملکرد این تابع خروجی بخش قبل (**y.wav**) را به تابع می دهیم:

```
>> analysis('y.wav')

ans =

    '810198369'
```

- در نهایت ورودی **a.wav** را به تابع می‌دهیم تا رمزگشایی شود:

```
>> analysis('a.wav')
```

```
ans =
```

```
'810198'
```