TTT4110 Project - Part 1

Table of Contents

In this project we will look into the practical application where sinusoidal signals are used to transmit information. This will be done through a touched-tone phone dialer. In this part we will generate the sinusoidal signals to transmit a phone number. Part 2 of the project will focus on extracting the information encoded in the sinusoidal signals.

Valid numbers and characters

```
valid_numbers=[1,2,3,4,5,6,7,8,9,0];
valid_chars=['*', '#'];
```

Frequencys

The frequencys are defined as upper and lower bound frequencys. They will be combined later to define the characteristic sinusoidal signal for each number and character.

```
lower_frequencies = [697, 770, 852, 941];
upper_frequencies = [1209, 1336, 1477];

Fs=8000;
```

Duration and delay

Each of the signals will have the duration *time* and there will be a delay between each signal of *delay*.

```
time=0.2; % duration in # sec.
delay=0.05; % delay in # sec.
```

Main code

The major part of the code is based in a single for-loop. This will greatly increase the possibility to reuse all the code without using separate function files.

```
phoneNr = input('Write phone number: ', 's'); % Input defined as string

% Initializing temporary variables
lower=0;
upper=0;

% Looping through all characters in the string with a for-loop:
for number=phoneNr
```

```
% Check if input is a valid character, and sets lower and upper
% frequencies accordingly
if(number == valid_chars(1) || number == valid_chars(2))
    switch number
        case '#'
            lower=lower_frequencies(4);
            upper=upper_frequencies(3);
        case '*'
            lower=lower_frequencies(4);
            upper=upper_frequencies(1);

    end
% Check if input is zero
elseif (number == '0')
    lower=lower_frequencies(4);
    upper=upper_frequencies(2);

% Handle numbers 1-9 accordingly, print error message if value is
% neither a number nor in the valid character set and continue to next input.
else
    number=str2num(number); % Trying to convert string-character to number
    if(isempty(number) || number==i || number==j) % Convertion returns and emp
        disp('Not a number or valid character, ignoring');
        continue; % Skip to next character from input string.

    end % End if

    % Calculates the 'upper' frequency of the sinusoidal signal based on
    % number value.
    switch mod(number, 3)
        case 0
            upper=upper_frequencies(3);
        case 1
            upper=upper_frequencies(1);
        case 2
            upper=upper_frequencies(2);
    end % End switch #1

    % Calculates the 'lower' frequency of the sinusoidal signal based on
    % number value.
    switch number
        case {1,2,3}
            lower=lower_frequencies(1);
        case {4,5,6}
           lower=lower_frequencies(2);
        case {7,8,9}
            lower=lower_frequencies(3);
    end % End switch #2
end

t=0:1/Fs:time;

xLower=sin(2*pi*lower*t); % Generate the 'lower' sinusoidal siganl
xUpper=sin(2*pi*upper*t); % Generate the 'upper' sinusoidal signal
```

```
    sound(xUpper+xLower, Fs); % Play the combined sinusoidal signal
    pause(delay+time); % Pause program to allow delay between signals
end % End for-loop
```

Published with MATLAB® R2013a