**Signals and Systems**

**LAB 11:**

**Hints and Tips**

**Task 1:**

* krow = ceil(kk/2); %this just selects the relevant frequency from the table no need to explain this

kcol = rem(kk-1,2) + 1; %this just selects the relevant frequency from the table no need to explain this

* Identify the line in code which adds silence to the code
* Observe how concatenation is taking place

**Task 2:**

* Task 1 might be of some help here,

**Part 2:** The complete tone xx will follow the following pattern

xx= silence signal\_key1 silence signal\_key2 silence signal\_key3 ………

For example: keypressed=’2’ ‘4’ ‘A’

Final o/p would be: silence tone of no.2 silence tone of no.4 silence tone of no.A

duration of silence=0.05s

duration of signal\_key=0.2s

**Part 3:** Observe that the following code is specifically written for task 1 as the input is in the form of numbers and not any string:

**krow = ceil(kk/2);**

**kcol = rem(kk-1,2) + 1;**

therefore, it cannot be applied to task 2, for that purpose there is already few code lines provided in the skeleton of dtmfdial.m

* If **Find** command is used for a 2D matrix it then returns the indices of row and column.

**Task 3:** The maximum value of the frequency response of your bandpass filter at the cutoff frequency must be 1. You may have a look at figure 3.

Another useful tip: Use absolute value.

**Task4:**

**Part 1:** The output at any value of L should contain the response of all the 8 frequencies.

At the end of the function take care of the dimensions of **hh** it must be **Lx8.** (Workspace shows you the size of all the variables)

**Part 2:** Give it a read, for selection of L: Just check (by plotting as required in part 3) at which value of L all the frequencies are giving a response which does not overlap with the others. Remember: L must not be too large nor too small.

**Part3:** Plot them all as shown in figure 4.

**Task 5:** Scoring Function

Note: Write this function show it to one of the RAs and go for task 6 you’ll do the testing a with task 6.

This function simply checks for the frequencies present in a tone or a segment of the total signal (i.e one tone).

Here xx is the **segment** of the input signal having a single tone at a time (as per given by the user) encoded in it. First use dtmfcut then extract the **x\_segment** which is your xx for dtmfscore function.

hh is the response of bandpass filters that you wil get from the dtmfdesign function

You need to find the output ‘y’

And the apply the given condition , the function gives answer only in the form on ‘0’ or ‘1’. For a single valid tone/number pressed out of eight frequencies response in correspondence to two frequencies will be high ‘1’ and rest of the six will have ‘0’ value.

Note that: it takes hh for a single columns (i.e impulse response of only 1 frequency at a time)

For eg: when used in loop for each frequency check if tone ‘4’ is pressed dtmfscore will give output as: 0 1 0 0 1 0 0 0

**Task 6:** This is the final function which decodes the keys pressed by you.