Final Exam for Group A (20 Marks) **Apr 25, 2018**

Dept. of Electrical Engineering, IIT Madras
Applied Programming Lab Jan 2018 session

- ▷ **Time duration of exam is** 2.5 **hours**
- ▷ **Create a work directory called *your-roll-number (lower case)* and make it accessible only by you:**

```
chmod 700 your-roll-number
```

  Then change to that directory and do your work.

- ▷ **vector operations are a must.**
- ▷ **Label all plots. Add legends. Make the plots professional looking.**
- ▷ **Comments are not optional. They are required.**
- ▷ **pseudocode should be readable and neatly formatted.**
- ▷ **code should be written as part of a LyX file. The LyX file should be professional in appearance, and I will give marks for it.**
- ▷ **LyX file should be named *your-roll-number.lyx* (if you submit a .tex file it should be named *your-roll-number.tex*)**
- ▷ **PDF file should be named *your-roll-number.pdf***
- ▷ **Python code should be named *your-roll-number.py***
- ▷ **Python code should run!!**
- ▷ **Include the plots in the lyx file and generate a pdf.**
- ▷ **Internet will be turned off at the beginning of the exam.**
- ▷ **Leave the above three files in your working directory**
- ▷ **Late submission will result in reduced marks.**

An electron beam is injected into a 1 metre cavity and goes out the other side through an opening. The beam is perfectly uniform with one electron appearing every μsec. The electrons all have an initial velocity of 1000 metres/sec. The cavity has a voltage applied from entry to exit ports equal to

$$\frac{-eV}{m_e} = 2 \times 10^5 \sin(1000\pi t)$$

$15,000$ electrons are injected starting at time $t = 0$. The problem is to track the electrons through the cavity and record their time of exit. The total number of electrons collected over every 50μsec (call it "density") is recorded. This gives you 300 readings.

[part-1:5 marks] Write pseudocode on how the electrons are tracked and the density obtained vs time. Your algorithm should use the smallest size arrays possible and you should justify that size in this part. The report should be professional and all plots be properly labelled. The code should be part of the report, and well annotated.

[part-2:7 marks] Write python code to simulate the electron transit through the cavity and obtain the density vector. Plot the electron locations in cavity at a typical time and also the electron velocities (only a small number of electrons will be in the cavity at any given time). Also plot the last 256 points of density vector vs time in figure 1.

[part-3:4 marks] Fourier transform the last 256 samples of the density vector and plot the spectrum, without (figure 2) and with (figure 3) windowing. Determine the frequencies present.

[part-4:4 marks] Zero the DC part of the spectrum and take the inverse fourier transform. Plot the same in figure 1, to show the AC part of the density vector. Explain the differences between the two time signals.

Useful Python Commands (use "?" to get help on these from ipython)

```
from pylab import *
import system-function as name
Note: lstsq is found as scipy.linalg.lstsq
```

Python Commands (continued)

```
ones(List)
zeros(List)
range(N0,N1,Nstep)
arange(N0,N1,Nstep)
linspace(a,b,N)
logspace(log10(a),log10(b),N)
X,Y=meshgrid(x,y)
where(condition)
where(condition & condition)
where(condition | condition)
a=b.copy()
lstsq(A,b) to fit A*x=b
A.max() to find max value of numpy array (similalry min)
A.astype(type) to convert a numpy array to another type (eg int)
trunc(A) to truncate values of A to integer.
def func(args):
  ...
  return List
matrix=c_[vector,vector,...] to create a matrix from vectors
w=0.54+0.46*cos(2*pi*n/(N-1)) to generate a window of N points, where n=linspace(-N/
fftshift(v) to shift a vector from -max to +max to lie on unit circle.
```

Python commands (contd)

figure(n) to switch to, or start a new figure labelled n
plot(x,y,style,...,lw=...)
semilogx(x,y,style,...,lw=...)
semilogy(x,y,style,...,lw=...)
loglog(x,y,style,...,lw=...)
contour(x,y,matrix,levels...)
quiver(X,Y,U,V) # X,Y,U,V all matrices
xlabel(label,size=)
ylabel(label,size=)
title(label,size=)
xticks(size=) # to change size of xaxis numbers
yticks(size=)
legend(*List*) to create a list of strings in plot
annotate(str,pos,lblpos,...) to create annotation in plot
grid(Boolean)
show()