Homework 1 Student Name: \_\_\_SHAURYA PANTHRI\_\_

AuE 8930: Machine Perception and Intelligence

Instructor: Dr. Bing Li, Clemson University, Department of Automotive Engineering

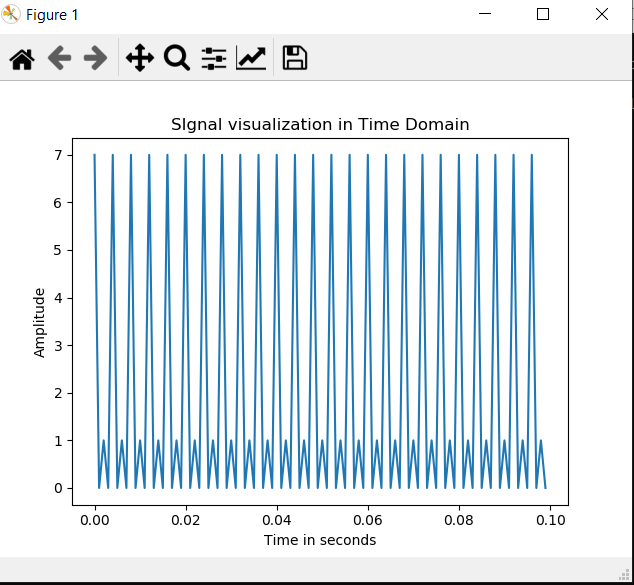
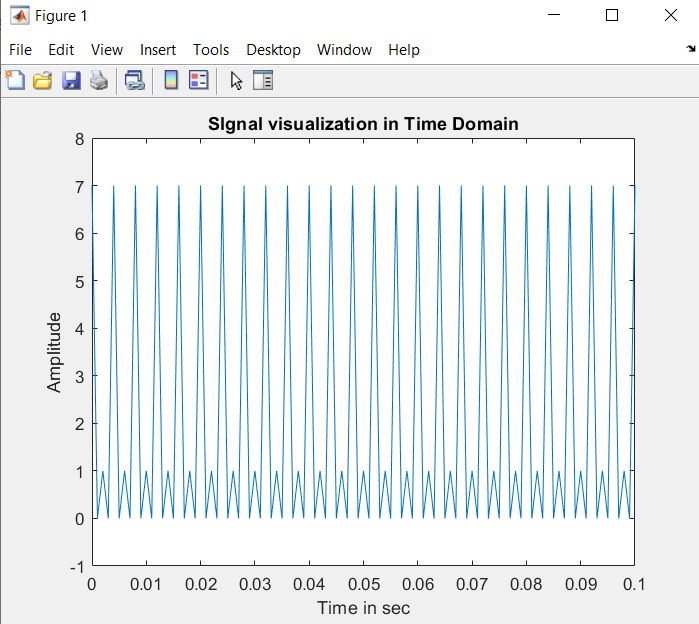
\* Refer to Syllabus for homework grading, submission and plagiarism policies;

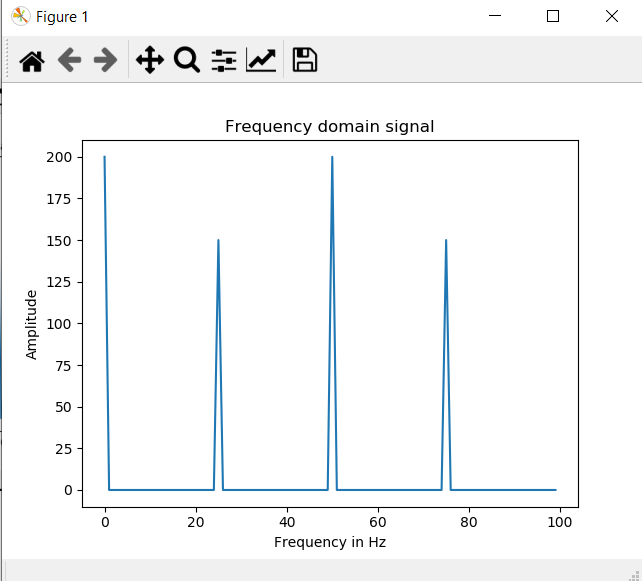
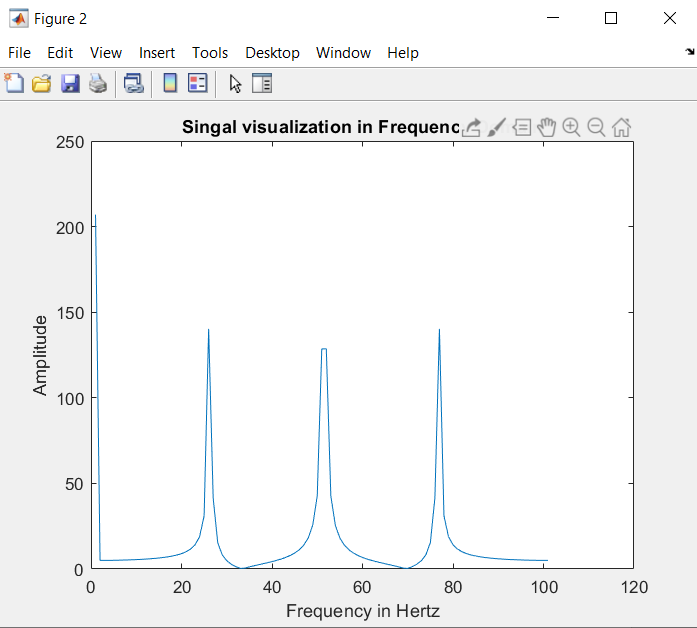
\* Submission to Canvas (Due: Tues. Feb. 4, 2020 11:59 pm), including:

* This document (with answers), and with your program results/visualization;
* A .zip file of source code (and data if any) with names indicating question number;

1. Visualize continuous period signal x(t) = 2 + 3 \* cos (500  t) + 2 \* cos (1000  t) + 3 \* sin (2000  t) in time-domain (axis: Amplitude and t) and visualize its digital Fast Fourier transform (axis: Amplitude and f). Given Sampling frequency as 1K HZ. (5 points)

ANSWER (1):

Figures from Python and MATLAB.

CODE:

**MATLAB:**

%% Problem 01

clc

clear all

close all

Fs=1000 %Sampling freq.

dt=1/Fs % Sampling time

time=0:dt:0.1

time\_data=2+ 3\*cos(500\*pi\*time) + 2\*cos(1000\*pi\*time) + 3\*sin(2000\*pi\*time)

figure()

plot(time,time\_data)

title('SIgnal visualization in Time Domain')

xlabel('Time in sec')

ylabel('Amplitude')

%% Problem 1 part 2 Fourier Transform of the function

freq= abs(fft(time\_data));

figure()

plot(freq)

title('Singal visualization in Frequency Domain')

xlabel('Frequency in Hertz')

ylabel('Amplitude')

**PYTHON:**

-\*- coding: utf-8 -\*-

"""question1.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1\_rUTsMBZZ2DbFslgNheIAqU4ad0GIlyN

"""

# Commented out IPython magic to ensure Python compatibility.

# %matplotlib inline

import numpy as np

import matplotlib.pyplot as plt

import math

#from \_\_future\_\_ import division

Fs=1000 # sampling frequency or rate

t0=0

t1=0.1

time=np.arange(t0,t1,1/Fs)

#time=np.linspace(0,0.1,0.0001) # time vector start,end and size

print(time)

amp2=2

amp3=3

w1=500 #omega for the function

w2=1000

w3=2000

time\_data=2+amp3\*np.cos(w1\*math.pi\*time)+amp2\*np.cos(w2\*math.pi\*time)+amp3\*np.sin(w3\*math.pi\*time)

#noise=0 # No noise the wave

"""plotting the function"""

plt.plot(time,time\_data)

plt.title('SIgnal visualization in Time Domain')

plt.ylabel('Amplitude')

plt.xlabel('Time in seconds')

plt.show()

#print(time\_data)

import sympy

#from sympy import fft

freq = np.abs(np.fft.fft(time\_data)) # frequency data for time domain using the fft function

""" Plotting the frequency domain function"""

plt.plot(freq)

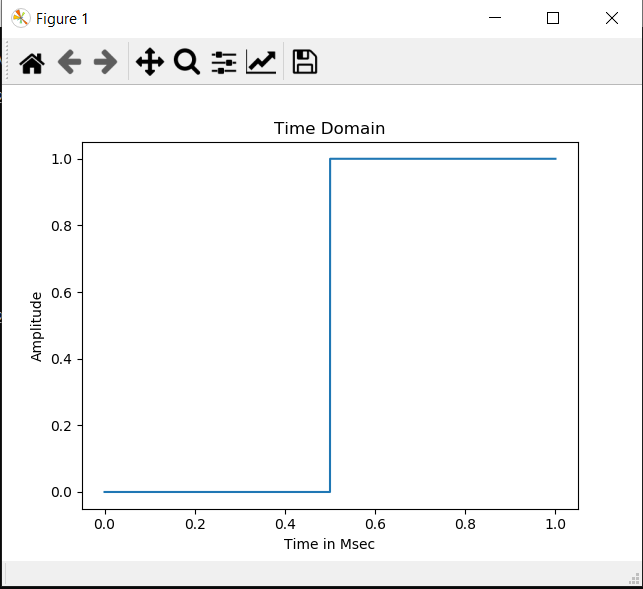
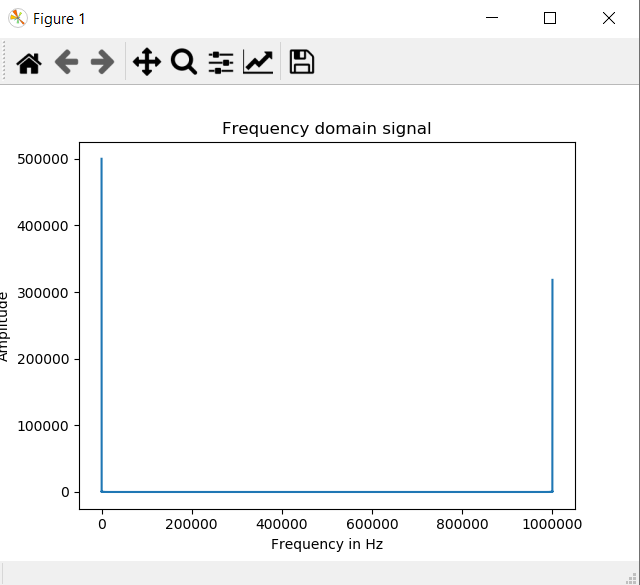
plt.title('Frequency domain signal')

plt.xlabel('Frequency in Hz')

plt.ylabel('Amplitude')

plt.show()

1. Visualize discrete signal x(k) = 0 for k [0 499] and = 1 for k  [500 1000) s (sampling frequency as 1M HZ) in time-domain (Amplitude over t) and visualize its digital Fast Fourier transform (Amplitude over f), find its -3dB (called half-power) bandwidth frequencies (f\_low, f\_high) in frequency spectrum. (10 points)

Low Bandwidth Frequency = 0

High Bandwidth Frequency = 0.00851

**Code in Python:**

# -\*- coding: utf-8 -\*-

"""question2.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1hFqN0D8k59ShVLfMISViEOLzUYfqUZ81

"""

# Commented out IPython magic to ensure Python compatibility.

# %matplotlib inline

import numpy as np

import matplotlib.pyplot as plt

import math

Fs=1\*10\*\*6 # sampling frequency or rate

N=Fs #Time period

time=np.linspace(0,1,N) # time vector start,end and size

time\_data=[]

n=np.linspace(0,1000,1)

x=np.arange(0,1000,1)

#print(x)

count=0

while(count<=((len(time)/2)-1)):

time\_data.append(0)

count=count+1

count=0

while(count<=((len(time)/2)-1)):

time\_data.append(1)

count=count+1

print(len(time))

print(len(time\_data))

"""plotting the function"""

plt.plot(time,time\_data[0:len(time\_data)])

plt.title('Time Domain')

plt.ylabel('Amplitude')

plt.xlabel('Time in Msec')

plt.show()

"""Analyzing the singnal in the frequency domain"""

from scipy.fftpack import fft #import Fourier transform for converting to frequency domain

Fs=Fs/2 #Nyquist Shanon requirement

frequency=np.linspace(0.0,Fs,int(N/2))

#freq\_data=fft(time\_data) # frequecny data for time domain using the fft function

freq\_data=np.abs(np.fft.fft(time\_data))

y=2/N\*np.abs(freq\_data[0:np.int(N/2)])

""" Plotting the frequency domain function"""

plt.plot(freq\_data)

#plt.plot(y,frequency)

plt.title('Frequency domain signal')

plt.xlabel('Frequency in Hz')

plt.ylabel('Amplitude')

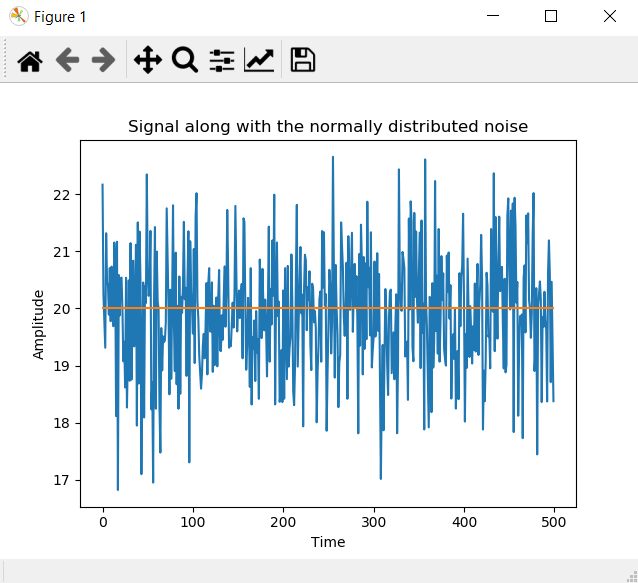
plt.show()

import scipy

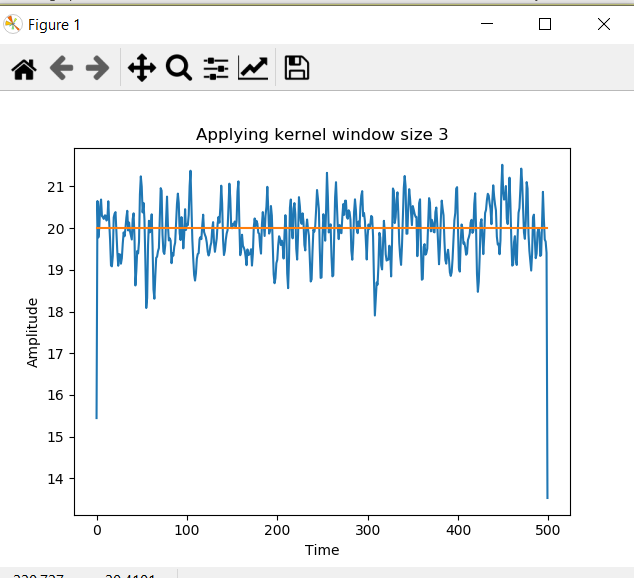
from scipy import signal

f,px=scipy.signal.periodogram(freq\_data)

1. For discrete signal x(k) = 20 for k [0 499], add a normally distributed random noise n(k) (mean 0, variance 1) to the signal, and get x(k) = x(k) + n(k). Then, apply a normalized (mean 0, standard deviation 1) [Gaussian kernel](http://dev.theomader.com/gaussian-kernel-calculator/) (windows size 3 and 11 respectively as a low pass filter, then rescale all elements to make sure the sum is 1) to perform convolution y(k) = x(k)  h(k) (h presents the impulse response, and in this case it’s the filter) by using basic arithmetic operations only.
2. Visualize both x(k) and x(k) in one figure (10 points)

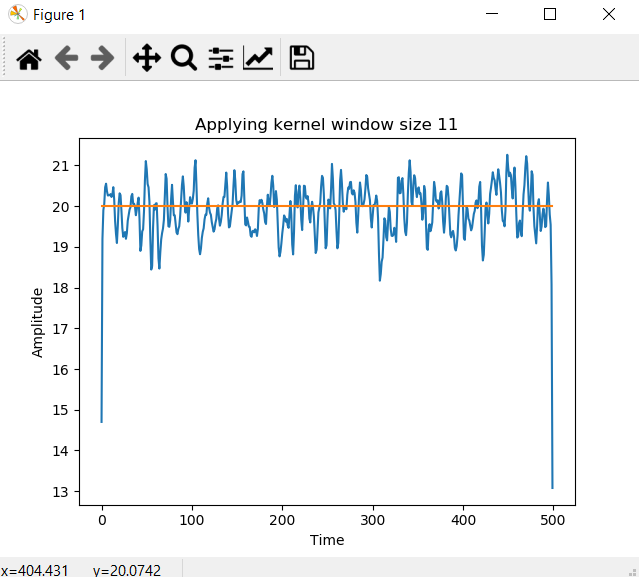


1. Visualize both x(k), and y(k) based on kernel window size 3 in one figure (10 points)



1. Visualize both x(k), and y(k) based on kernel window size 11 in one figure (5 points)

Tip: You may consider using zero-padded for edges during convolution operation



CODE:

# -\*- coding: utf-8 -\*-

"""question3.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1oqNxLB-NrIuXfYdUjjYG2ZZDMedn7BvX

"""

# Commented out IPython magic to ensure Python compatibility.

# %matplotlib inline

import numpy as np

from numpy import pi,exp

import matplotlib.pyplot as plt

from pylab import \*

mu=0

variance=1

sigma=np.sqrt(variance)

k=np.arange(500)

print(len(k))

x=[]

for i in range(len(k)):

x.append(20)

s = np.random.normal(mu, sigma, 500)

signal=x+s

plt.plot(k,signal,k,x)

plt.title('Signal along with the normally distributed noise')

plt.ylabel('Amplitude')

plt.xlabel('Time')

plt.show()

"""Analyzing the singnal in the frequency domain"""

from scipy.fftpack import fft #import Fourier transform for converting to frequency domain

xft=fft(signal)

plt.plot(xft)

plt.title('Frequency Domain')

plt.ylabel('Amplitude')

plt.xlabel('Time')

plt.show()

""" Kernel"""

""" Defining a function for performing the convolution"""

import numpy as np

def convolution(A,B):

lengthA=np.size(A)

lengthB=np.size(B)

C = np.zeros(lengthA + lengthB -1)

for m in np.arange(lengthA):

for n in np.arange(lengthB):

C[m+n] = C[m+n] + A[m]\*B[n]

return C

kern3=[0.27901,0.44198,0.27901]

kern11=[0.000003, 0.000229,0.005977,0.060598,0.24173,0.382925,0.24173,0.060598,0.005977,0.000229,0.000003]

C3=convolution(signal,kern3)

a=len(C3)-1

print(a)

C3=C3[1:a]

print(len(C3))

print(len(x))

plt.plot(k,C3,k,x)

plt.title('Applying kernel window size 3')

plt.ylabel('Amplitude')

plt.xlabel('Time')

plt.show()

# print(len(kern11))

# for i in range (len(k)):

# for j in range (len(kern3)):

C4=convolution(signal,kern11)

a=len(C4)-5

print(a)

C4=C4[5:a]

print(len(C4))

print(len(x))

plt.plot(k,C4,k,x)

plt.title('Applying kernel window size 11')

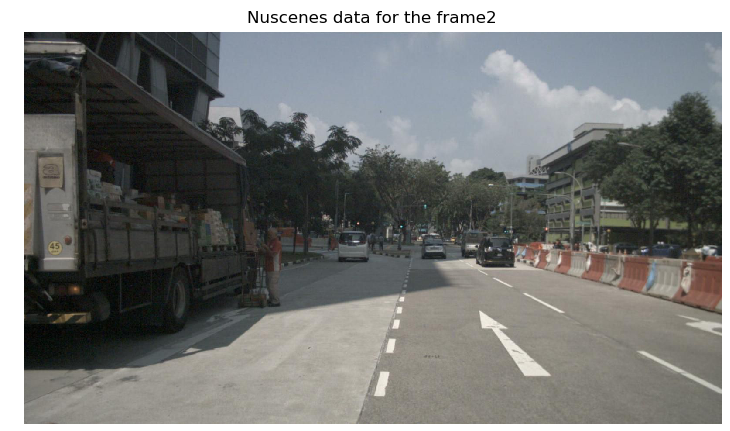
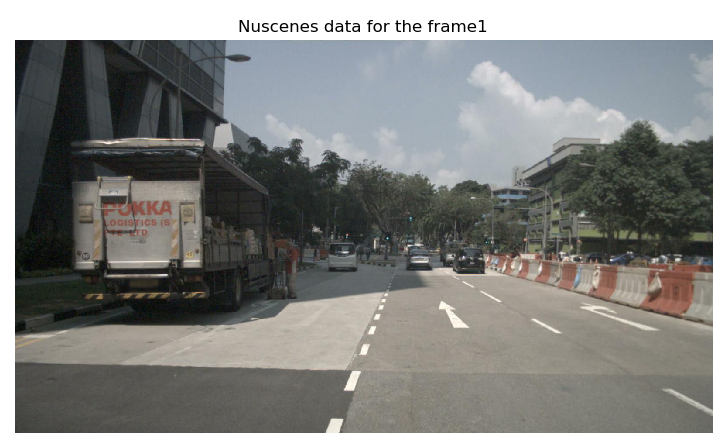
plt.ylabel('Amplitude')

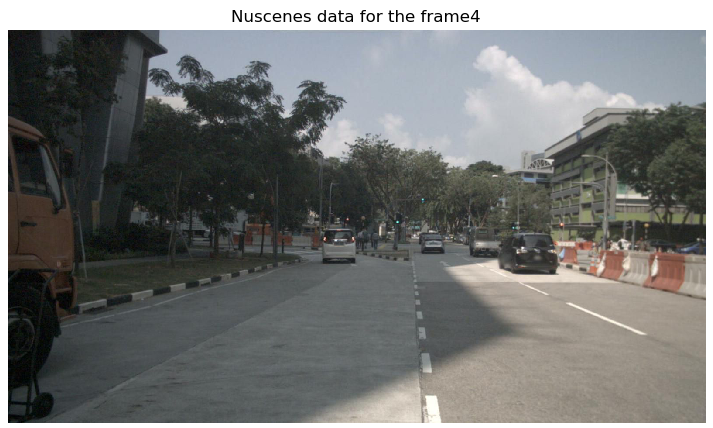
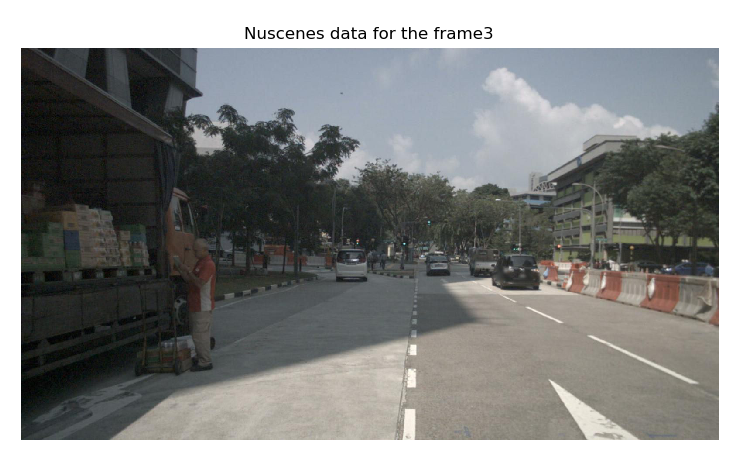
plt.xlabel('Time')

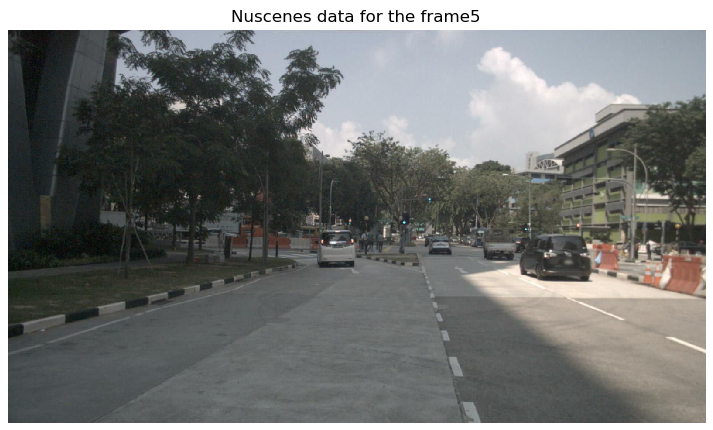
plt.show()

1. Find an online open dataset (such as but not limited to [ASTYX](https://www.astyx.com/development/astyx-hires2019-dataset.html), [KITTI](http://www.cvlibs.net/datasets/kitti/), [NUSCENES](https://www.nuscenes.org/)) containing 2D (or 3D) Radar data and its labelling, and pick up partial data from Radar dataset.
2. Visualize a continuous time frames (like a few seconds) for the Radar data in drawing visualization as a video; (5 points)

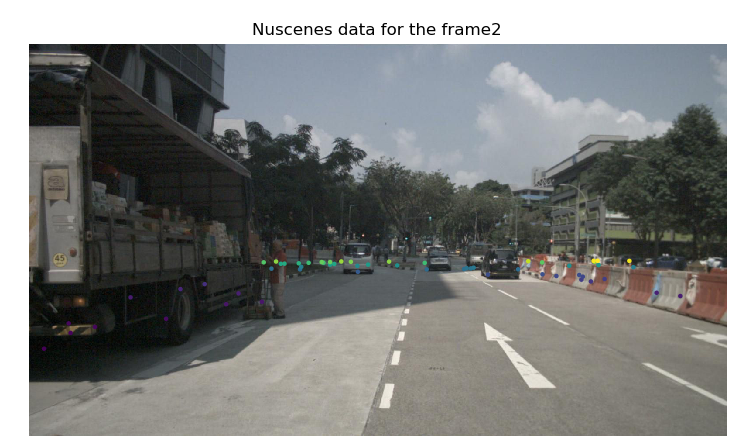
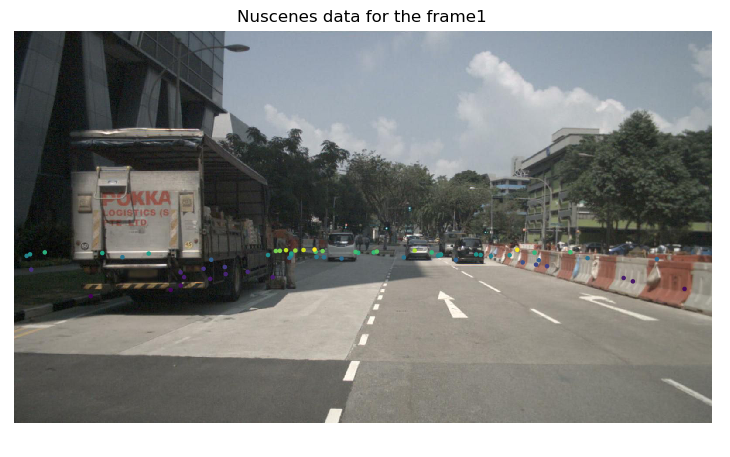
Answer:

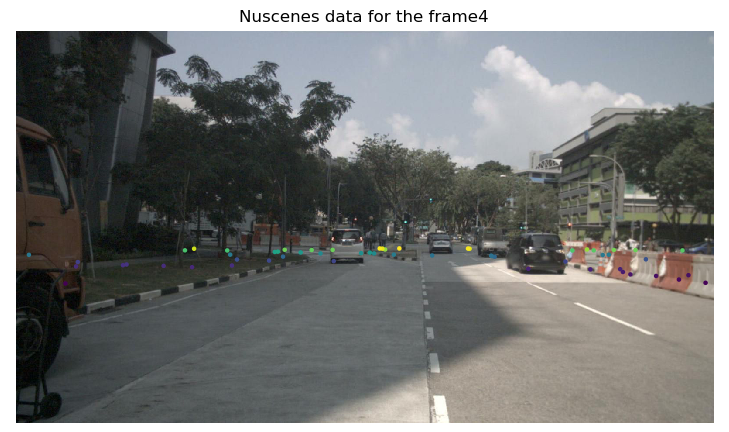


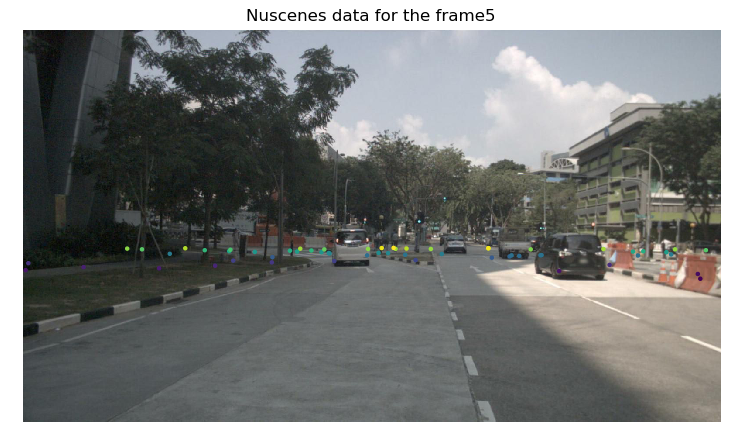




THE RADAR DRAWINGS AND LABELS:



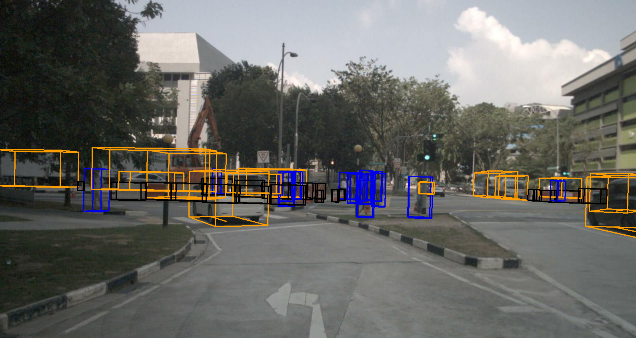


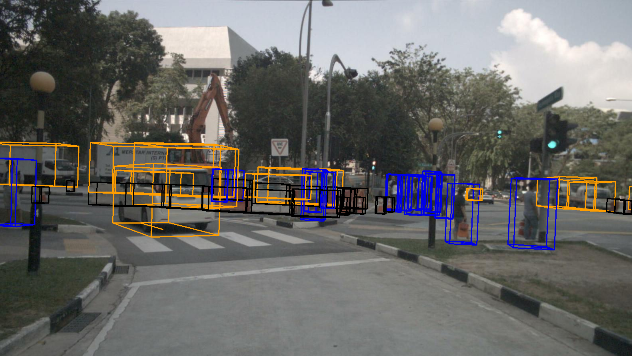


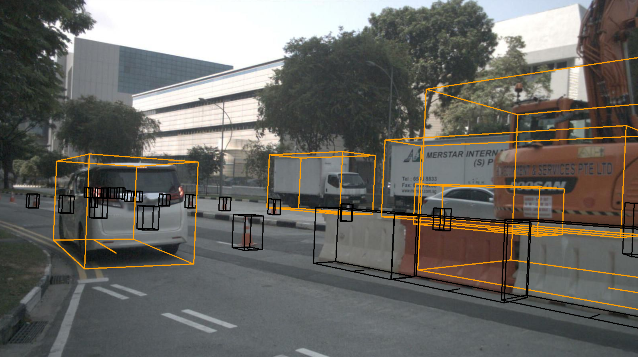
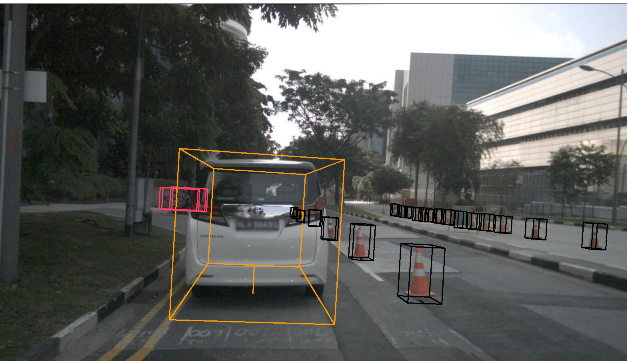
Objects labelled with intensity depicting the nearness to the Radar.

1. Visualize objects by its labelled data on the above visualization; (15 points)

Answer: The labelled objects in the visualization on the video per frame are shown from the nuscenes data.

1. 2~3 pages of survey on a particular 1D physical signal related to vehicles (40 points), (such as: 1D Radar, vibration/friction/temperature/speed/… signals). The grading of this question is based on the contents which the survey covers:

- The importance of this signal measurement (5 points);

- The challenges of measuring this physical signal data (5);

- Existing solutions of measuring this physical signal (15);

- Existing problems of measuring this physical signal (5);

- There will be other grading factors (such as novelty, organization, et al) (10);

\* You are encouraged to include any drawing/table in the report;

\* Attention: Survey a particular 1D signal, not survey/compare multiple 1D signals.

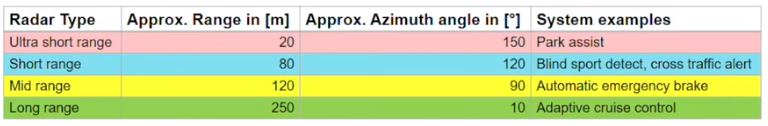
\* Attention: use “…” [1] to cite any sentence you literally copied and use … [1] to cite a content you referred to, with reference list in the end;

**1-D Radar signal survey**

Radar has a number of advantages over alternative sensing technologies for automotive applications for examples cameras, ultrasonic sensors, or LiDAR. They assist in certain automotive active safety systems and also in autonomous driving systems as well.

**1.The importance of this signal measurement.**

* They can instantaneously detect objects. It determines the range of the object.
* They have the unique ability to measure the velocity of the objects on the path via Doppler shift of radar signatures.
* They can perform well in certain adverse environment conditions like bad weather such as rain, fog, and snow.
* They are robust in bad light also that is in darkness and at night they successfully sense the object velocity and speed with accuracy and precision.
* Millimeter Wave FMCW RADAR are popular for automotive applications because they are small size, accuracy, light weight, low price and high-performance radars mounted on a single chip. They have smaller wavelength, high resolution, accuracy and ability to distinguish between the two distant objects.
* They have applications in the Adaptive cruise control, lane change assist, autonomous emergency braking and stop-and-go traffic jams.



**2.The challenges of measuring this physical signal data.**

* There is a requirement to assure that the transmitted signal is correct. This means that during the design verification it is made sure that the receiver responds to correct signals and there are no undesired signals emitted from the transmitter.
* At times there are challenges related to unexpected outputs that arise not only from unintended signals that arise from the pulse or spurious outputs due to radiation of internal oscillators, coupling from digital clocks, spurious oscillations with RF circuitry, pulse errors etc.
* The modern software defined radars work on the Digital Signal Processing and Digital Synthesis. Due to some errors in the computer codes for example some undesired filter values, logical errors or numerical expressions creates some short duration signals that are unrelated to the desired output. These signals generated can create problems in case they are amplified and transmitted. They can create signals that bear negligible relation to the desired output in the first place.
* Spurious emissions have a tendency to interfere with other services and may provide a distinctive signature if they are specific to a particular transmitter design.
* The frequency modulated continuous wave radars (FMCW) that are being adopted for automotive applications in the modern times need to keep linearity as it is one the most important performance parameter. Distortions occurs as the circuitry is used in severe environmental conditions and this distortion in frequency or non-linearity affects the measurement.[4] ([mmwave])

**3.Existing problems of measuring this physical signal.**

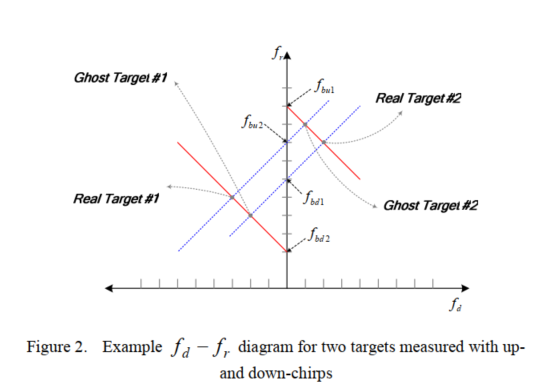
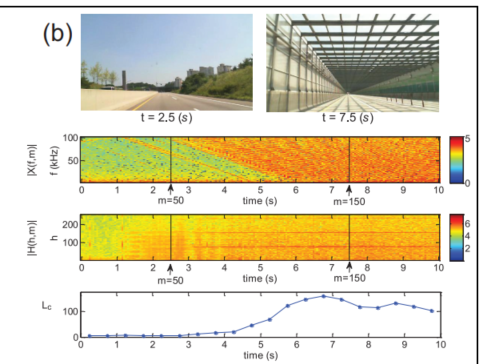
* Production testing requires that each unit must meet its specifications. During the production of the radars tuning and calibrating the assemblies is necessary and compensating if linearizers, calibrate analog modules and the amplifier components. The results are made accurate by repeating the steps to assure that the product will function as intended but as the components, but continued verification is required for desired performance[.[3]](https://www.tek.com/document/primer/fundamentals-radar-measurements)
* Although FMCW is an effective method, it possesses the certain ambiguities when it comes to multi-target situations.

Figure 1Ghost targets misdetected in range velocity processing

* + Separation of the ranges and the velocities for each target with the wide relative velocity and range
  + Identification of the correct combinations of the beat frequencies for the multiple targets.

There are two real and two ghost targets that is no physical representation of a reflection object. These ‘ghost targets’ and ‘missing targets’ in range velocity processing are problematic.[[2]](https://ieeexplore.ieee.org/document/5309873) (E. Hyun and J.-H. Lee)

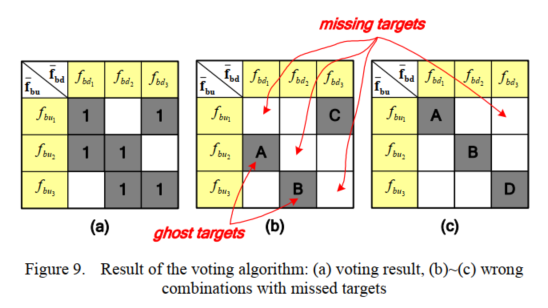
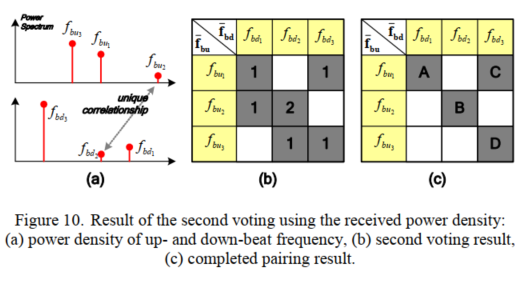
****

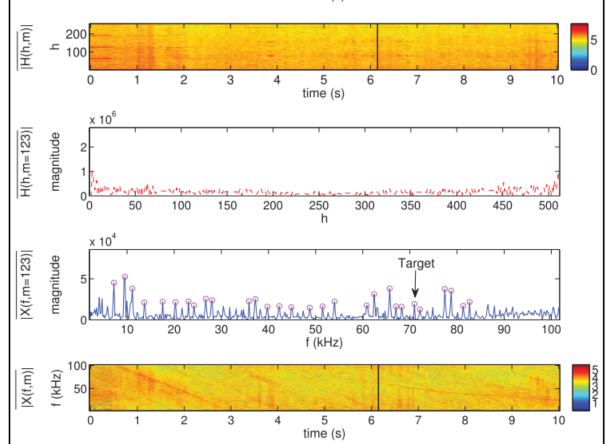
* **Detection problems in radars due to cluttering:** Radars perform better than other sensors on poor weather conditions or environmental conditions of the roads. But the signal quality of these radars is deteriorated by downsizing when man made structures on the road that is iron tunnels and guardrails or soundproof walls with high reflectivity against electromagnetic waves generate harmonic clutters that occur due to periodic structures on the road. This cluttering causes significant performance issue with the radar systems.

Figure 2 Analysis of level of harmonic clutters under various road conditions.

**4.Existing solutions of measuring this physical signal.**

* Calibration problems during the production cycle: The automated testing is performed to reduce the chances of operator errors that might occur through manual operation and interpreted testing equipment. The reproducibility is assured regardless of production personnel changes, training requirements. [3] (“Fundamentals of Radar Measurements)
* A pairing method that utilizes the voting technique can overcome complex signal processing to identify the correlative beat frequencies from all the combinations.
  + To combine the correct pair of frequencies for each target an additional continuous wave is required. As the doppler frequency is measured inn the period of unmodulated wave the ambiguity between the reflected and doppler can be resolved.
  + The power spectrum density of the beat frequency is used to vote the pairs with strongest relation for the first voting. The power densities of the (fbu2, fbd2 ) have the same value as they are received from target B hence this pair get the vote in the second voting.

* Harmonic clutter recognition and suppression: The spectral characteristics of the radar from different roads are compared on a harmonogram to recognize the harmonic clutters by extracting the peak components. For suppression the magnitude response of |H(h.m)| is extracted with negligible loss by using cell averaging constant false alarm algorithm by designing a threshold with the level of harmonic clutter as three times the mean power of harmonogram at normal road condition[. (J.-E. Lee)[1](https://journals.sagepub.com/doi/pdf/10.1177/1550147717729793)]

*Figure 3 An example of clutter suppression with residual in an iron tunnel*

References

* “Fundamentals of Radar Measurements, ”. T.-r.-m.-F.-2. (n.d.).
* 06-Feb-2020]., S. [.-m.-m.-a.-r.-t.-c. (n.d.). *[mmwave].*
* E. Hyun and J.-H. Lee, “. m.-t. (n.d.).
* J.-E. Lee, H.-S. L.-H.-C.-W.-C. (n.d.).
* StackPath. [Online]. Available: https://www.mwrf.com/technologies/test-measurement/document/21849594/overcome-mmwave-automotive-radar-testing-challenges-pdf-download. [Accessed: 06-Feb-2020]. (n.d.).