PRINCIPLES OF MODERN RADAR – ADVANCED TECHNIQUES

PROJECT TOPIC: RANGE ESTIMATION USING PYTHON

TEAM MEMBERS:

AKASH A (191EC102)

NAVRATAN (191EC133)

RAHUL KUMAR BAIRWA (191EC144)

SUBMITTED TO:

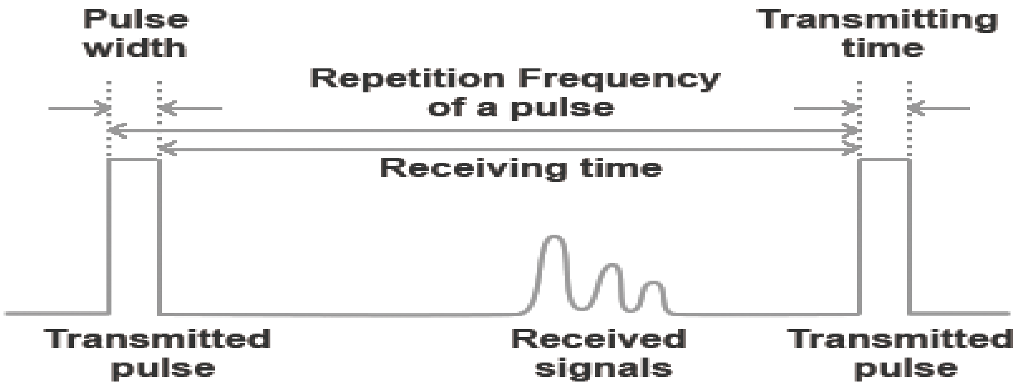
DR. PATIPATHI SRIHARI

Problem Statement:

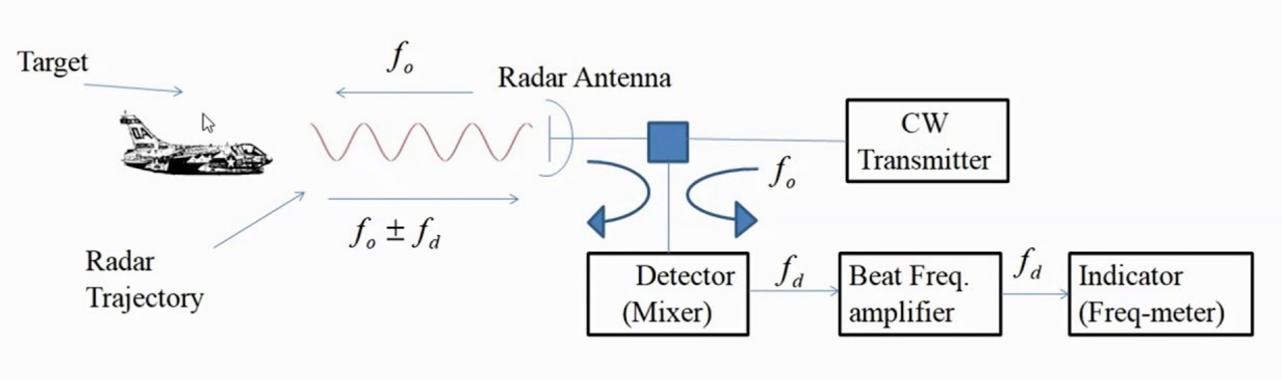
The purpose of FMCW range estimation is to estimate the range of a target. For example, a radar for collision avoidance in an automobile needs to estimate the distance to the nearest obstacle. The real time in phase and quadrature (I and Q) TI AWR1642 radar sensor data of moving car is provided for which implementation needs to be performed in Python for the provided MATLAB code.

Theory:

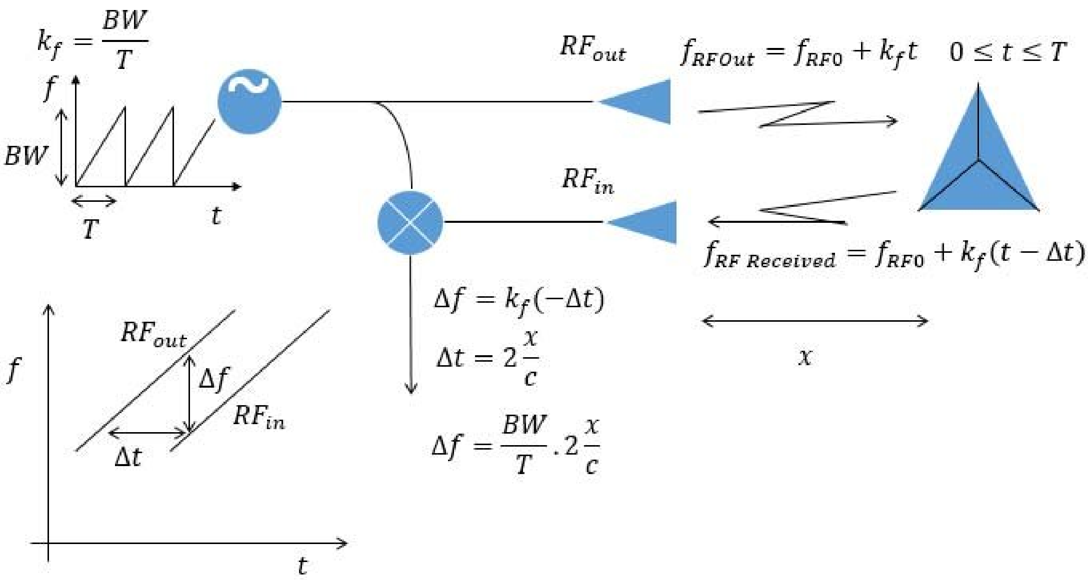
A radar is an instrument that is used to detect enemy obstacles. It is often termed as an electronic eye and offers advantages over other tools like cameras since they can be used at night as well as bad weather conditions like rain, fog, etc. Radars come in three categories – Pulsed Radars, Continuous Wave (CW) Radars Frequency Modulated Continuous Continuous Wave (FMCW) radars. Pulsed radars, send out pulses at regular intervals of time and are generally monostatic and are used for long range applicatons. CW and FMCW Radars send out a signal continuously. Since one antenna is needed for transmitting the signal continuously and another to receive the signal, CW and FMCW radars are typically bistatic. CW and FMCW radars are mainly used for short range applications. CW radars connot be used to estimate the range of the target since the signal is received continuously and it is not possible to determine the time when the signal reached the target. Their only use is for target detection or to determine the velocity of a moving target using Doppler shift. As a result, FMCW radars were developed which make it possible to measure the range of the target. The basic block diagrams of pulsed radars, CW radars and FMCW radars is as shown below:



Pulsed Radar



CW Radar



FMCW Radar

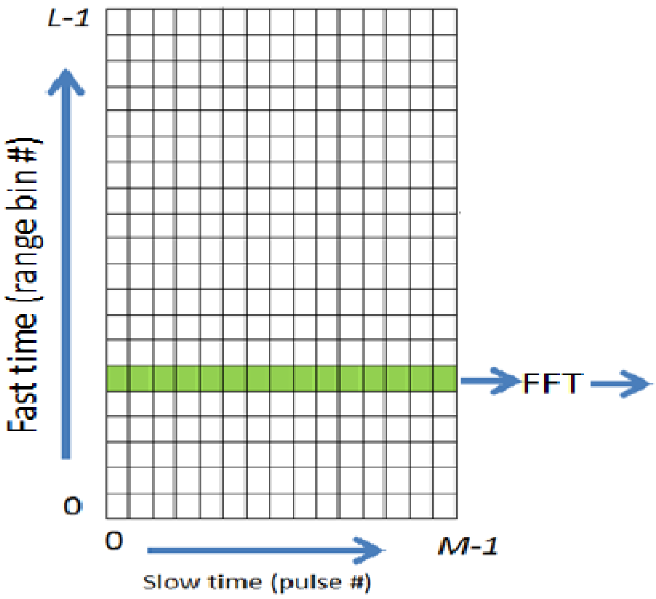
In an FMCW radar, the frequency of the transmitted signal is increased linearly with time as shown in the figure above. Such a waveform is known as a Linear Frequency Modulated (LFM) waveform. The received signal is simply a delayed version of the transmitted signal. We want to measure the time interval between the instant when the em wave left the target and when it returned. Since we know the slope of the waveform that we have transmitted (S), by measuring the frequency deviation Δf between the transmitted and received waveforms, we can measure the time taken by the em wave to return to the target using the relation:

$$S = \frac{\Delta f}{\Delta t}$$

Using the time difference Δt obtained above, we can estimate the range of the target as:

$$R = \frac{c\Delta t}{2}$$

Normally, in an FMCW radar, we use an Analog to Digital Converter at the receiver. The ADC samples the incoming em wave at the sampling rate which we also call as the fast time sampling rate. Each time the transmitter sends out a single ramp, the ADC samples the received signal which is also a ramp. Thus, a certain number of fast time samples are produced which we arrange in the form of a column. In a similar manner, for each of the ramp signals sent out, the same number of fast time samples are produced. These columns when concatenated together form a matrix called the data matrix. We also note that the matrix that is produced is complex and has both in phase as well as quadrature data. A single data matrix is produced for a single receiver. When multiple receivers are used, a separate data matrix is produced for each of the receivers which can be stacked together to form a data cube. The data matrix concept is illustrated in the diagram below:



Given the data matrix above, we need to estimate first the frequency deviation associated with the target. In order to do this, we must take the Fast Fourier Transform along the fast time axis. This is because the constant frequency shift appears as a linear phase shift (Phase is the integration of frequency) and when we take Fourier Transform of the linear phase shift, it gives an impulse function at the point of frequency shift. This arises from the following Fourier Transform property:

$$e^{j\omega_0 t} \Leftrightarrow \delta(\omega - \omega_0)$$

Thus, taking the FFT along the fast time axis will give a peak or amplitude at the point of the frequency shift while it will give zero at all other points. The higher the frequency shift value, the higher the range bin in which the peak will occur and thus, longer the range. Thus. The range bin in which the peak occurs is a measure of the range of the target. To estimate the exact range value, we use the fact that each range bin size is actually equal to the range resolution which is given by:

$$Range\,Resolution\,\Delta R = \frac{c}{2\beta} = \frac{c * Fast\,Time\,Sampling\,Rate}{2 * Slope * Number\,of\,Fast\,Time\,Samples\,per\,Ramp}$$

Thus, by plotting the range bin number multiplied by the range resolution on the x-axis and the FFT value obtained on the y-axis, we can claim that the point on the x axis at which the peak occurs is the actual range of the target. Further, since each column represents one FFT, we can integrate the results of the different columnns which will help us to get a better SNR value. Here non-coherent integration of pulses is used.

Details about the Radar Sensor:

The radar sensor used has a fast time sampling rate of 5 Mhz. 512 fast time samples are generated for each chirp. The radar operates at a frequency of 77GHz. The slope of the LFM waveform is given to be 8*10^12. The ramp time is given to be 56 µsec and the idle time is given to be 3 µsec. Four separate receivers are present. Using this information, we can calculate the range and Doppler Resolution of the radar as shown below:

$$Range\,Resolution\,\Delta R \;=\; \frac{c}{2\beta} \;=\; \frac{c * Fast\,Time\,Sampling\,Rate}{2 * Slope * No.\,of\;Fast\,Time\,Samples\,Per\,Chirp} \;=\; 0.183\,meters$$

$$Doppler\;Resolution\,\Delta v \;=\; \frac{\lambda * \Delta f_d}{2} \;=\; \frac{c}{2 * f * (ramp\,time + idle\,time) * Chirps\;per\;CPI} \;=\; 0.516\,m/sec$$

Here, the number of chirps in one CPI is taken to be 64 for our particular use case as will be discussed below.

Details about the Data Acquired:

As mentioned in the problem statement, the data obtained is for a moving car. The data is stored in the form of a binary file with 16 bits representing each of the samples. Upon reading this binary file, we get an array of 26214400 numbers. Since this data consists of both in phase and quadrature samples, we combine one in phase and one quadrature sample i.e. one I and one Q sample to get the complex data for further processing. Thus, we get a total of 13107200 samples. Since the number of samples in one chirp is 512 from the previous section, we can calculate the total number of chirps as:

$$Total\,Number\,of\,Chirps \;=\; \frac{13107200}{512} \;=\; 25600$$

Since we have four receivers in all, the number of chirps for a single receiver is one-fourth of the total number of chirps whichh can be calculated as 6400. For our use case, we have decided to estimate the range of the target 100 times and thus, we require 100 data frames. Thus, the number of chirps processed in a single frame can be calculated as:

$$Number\,of\,Chirps\,in\,a\,single\,frame \;=\; \frac{6400}{100} \;=\; 64$$

The data from these 64 chirps is processed coherently to achieve a better SNR and the range is estimated by taking the FFT as discussed in the theory.

Implementation:

The algorithm to estimate the range can be broken down into two sections – a preprocessing section and the main code. Initially, data is read from the binary file. In our use case, the code was developed using Python. The data was first extracted into a single-dimensional NumPy array. In case the number of bits per saple is less than 16, sign extension has been used. Then, the I and Q data was combined together to give complex samples. While doing this, care must be taken to note that the sensor data is arranged such that there are 2I samples first followed by two Q samples and so on. Then, data was rearranged so that a single column was created for each chirp. Finally, the data was arranged into a four dimensional NumPy array with each dimension representing a single receiver. It must be noted that while performing rearrangements, it must be done in Fortran order or column major order since the sensor data is arranged this way.

In the main code, we use the data only from the first receiver. We could equivalently reproduce this for any of the four receivers and the results would be similar. First we construct the data matrix for each of the 100 frames where each column represents the data for one frame. Thus each column would contain 64*512 elements. Now, a loop is run over each of the frames. In each iteration of the loop, one particular column is extracted and reshaped to form a data matrix with 512 fast time samples along the columns and 64 chirps along the rows. Thus the size of the data matrix is 64*512. A 1D FFT is then performed along each of the columns which would give a peak at the range bin in which the target is present as discussed in the theory. Finally, the data from each of the columns is added using non-coherent integration algorithm whose final output is given as below:

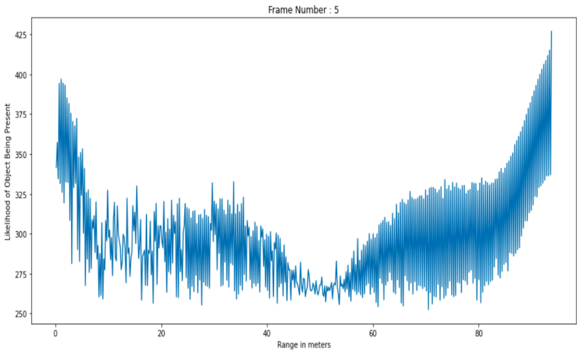
$$y[i] = \sum_{j=1}^{64} \left|\left(X[i][j]\right)\right|^2$$

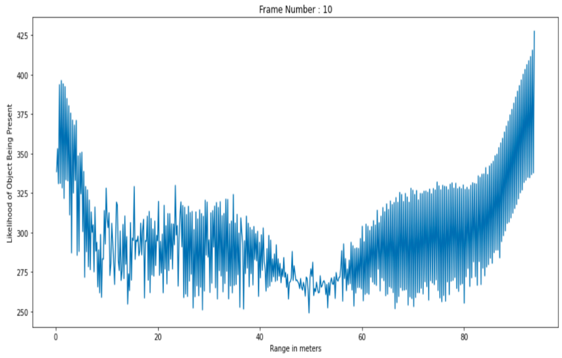
The final output is thus an array of 512 positive values each representing the likelihood of a target being present at a particular range bin. Finally a graph is plotted with the estimated range on the x-axis which is obtained from the range resolution multiplies by the index of the range bin for each of the 512 range bins and the likelihood of the target being present at that range which is the output y[i] which was obtained from the calculations above. For each of the 100 frames, a separate graph is obtained and by comparing each frame to its previous one, we can determine how the object is moving relative to its original position.

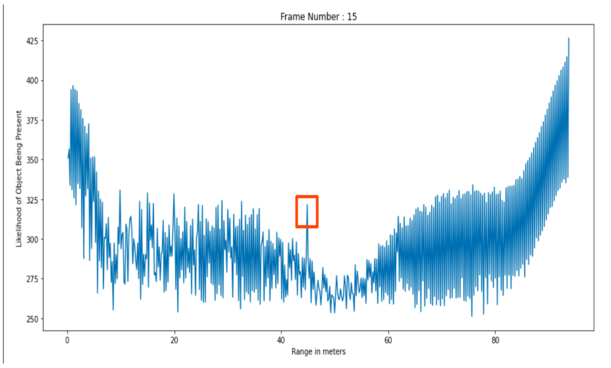
Note: It must be noted that whenever we perform rearrangements in Python, it is essential to follow the Fortran order or the column major order since this what is followed by default in the Matlab code.

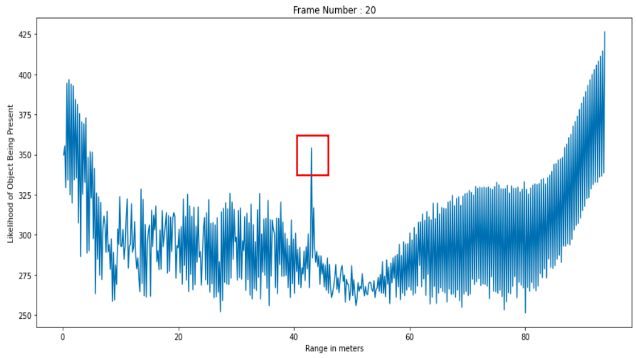
The libraries used for implementing the code were mainly Numpy for data processing and Matplotlib for drawing the plots. Sleep function from the OS library was also used to delay the process briefly so as to give some time duration from one frame to the next to make it possible for us to observe the motion of the target clearly.

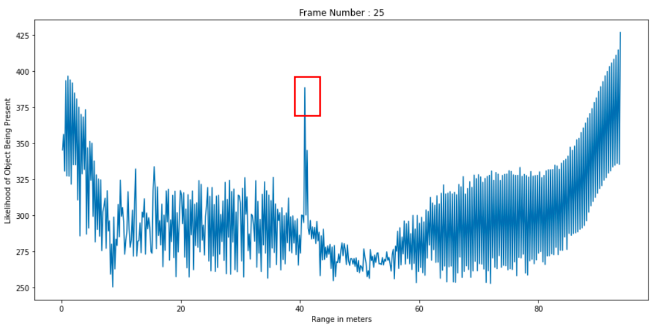
Final Results:

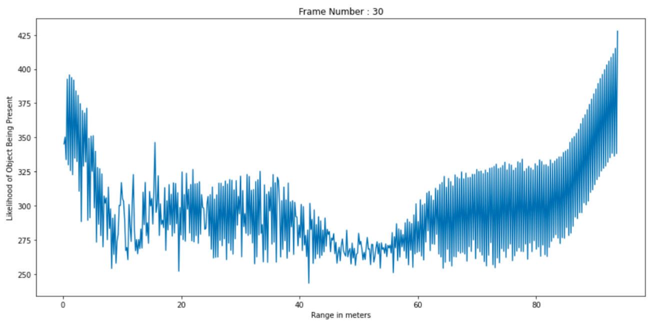


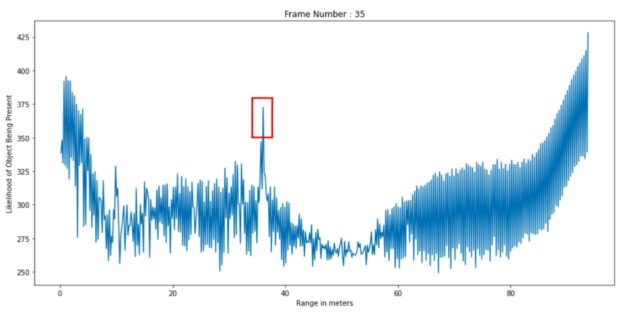


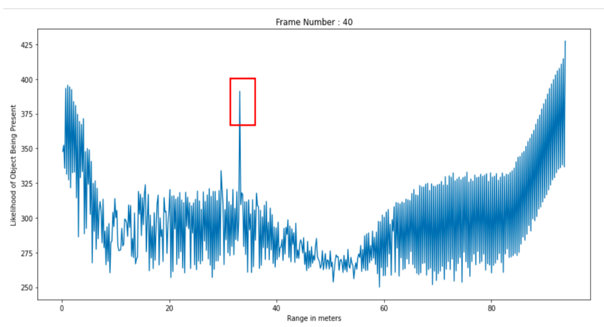


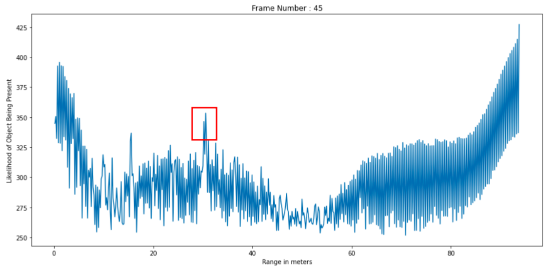


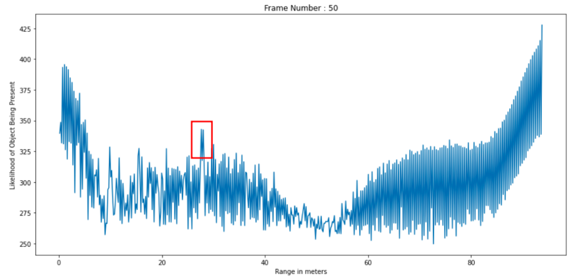


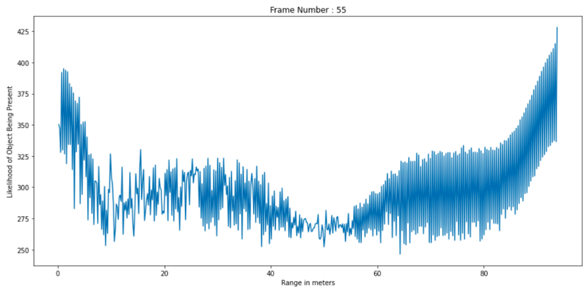


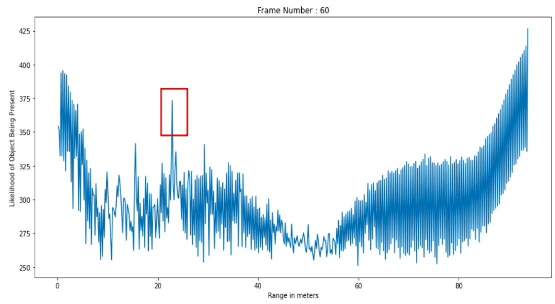


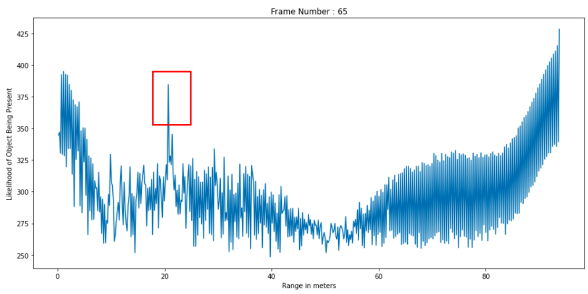


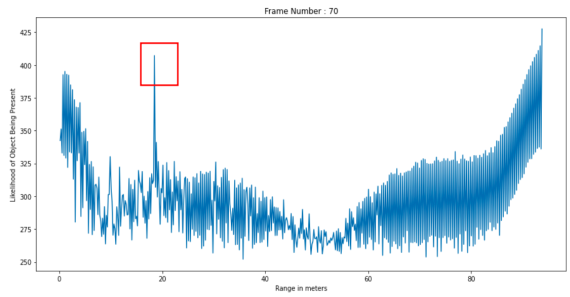

Frame Number : 45
Likelihood of Object Being Present
Range in meters

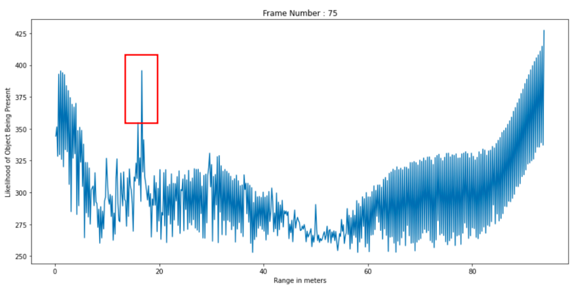

Frame Number : 50
Likelihood of Object Being Present
Range in meters

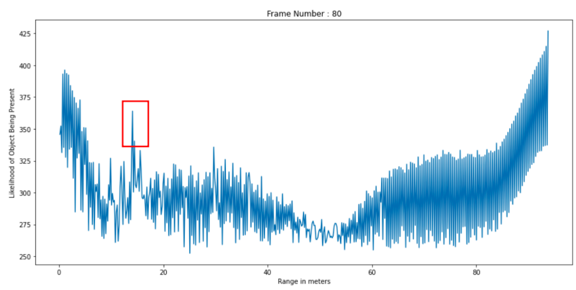

Frame Number : 55
Likelihood of Object Being Present
Range in meters

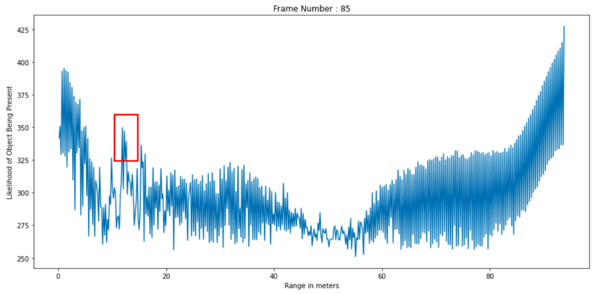

Frame Number : 60
Likelihood of Object Being Present
Range in meters

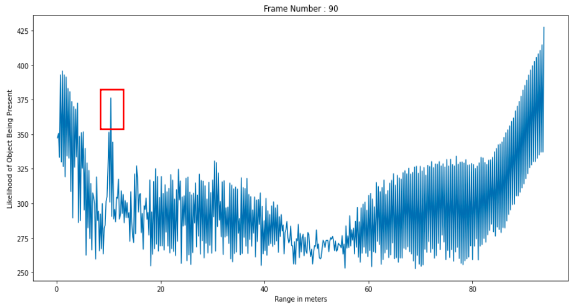

Frame Number : 65
Likelihood of Object Being Present
Range in meters

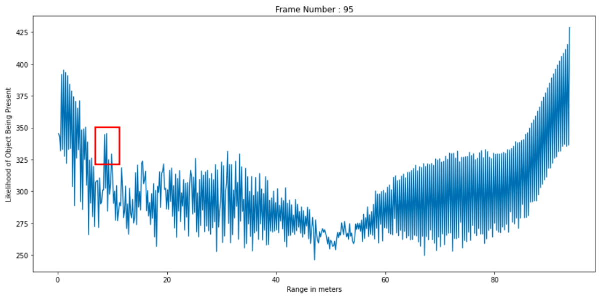

Frame Number : 70
Likelihood of Object Being Present
Range in meters

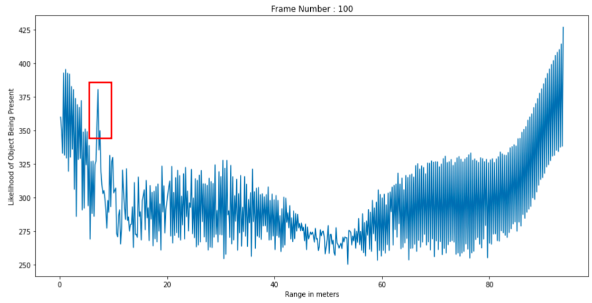

Frame Number : 75
Likelihood of Object Being Present
Range in meters


Frame Number : 80
Likelihood of Object Being Present
Range in meters









Observations:

The target begins to become visible at Frame 15 and is initially at a distance of approximately 45 meters from the radar. During successive frames, the target is found to move towards the radar since the peak corresponding to the target (enclosed in the red squares) comes closer and closer to the origin. Finally near frame 100, it is observed that the peak is less than 10 meters away from the radar which means that the target has come very close to the radar. There are also some instances where the target appears to be not visible or mixes into the clutter at which point it becomes impossible to separate the target from the clutter.

Conclusion:

An algorithm has been designed to estimate the range of a target. By implementing this algorithm, a moving car as a target was successfully detected.