CS 583 – Computational Audio -- Fall, 2021

Wayne Snyder
Computer Science Department
Boston University

Today: Fundamental Frequency Detection

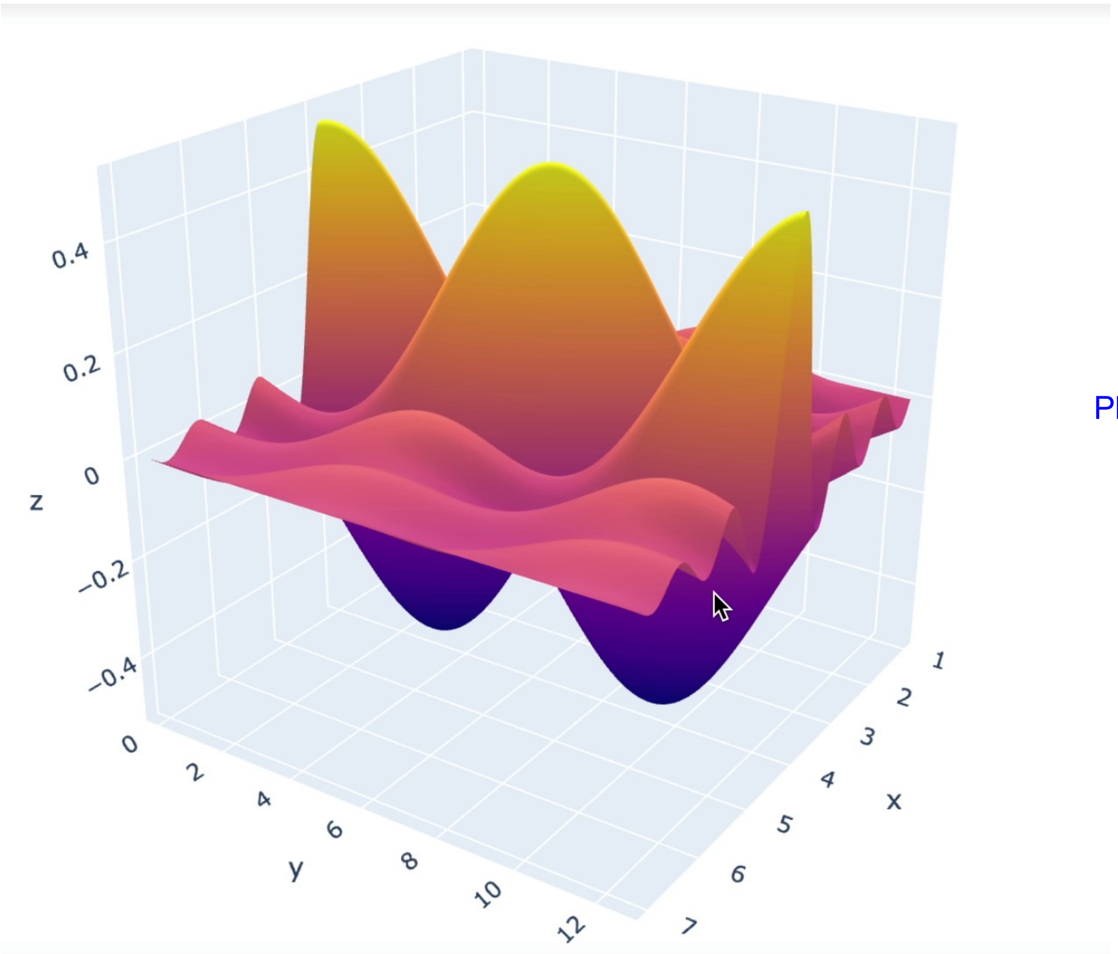
- Using Autocorrelation for F0 Detection
- Using interpolation to improve accuracy
- Problem: Which frequency is F0?

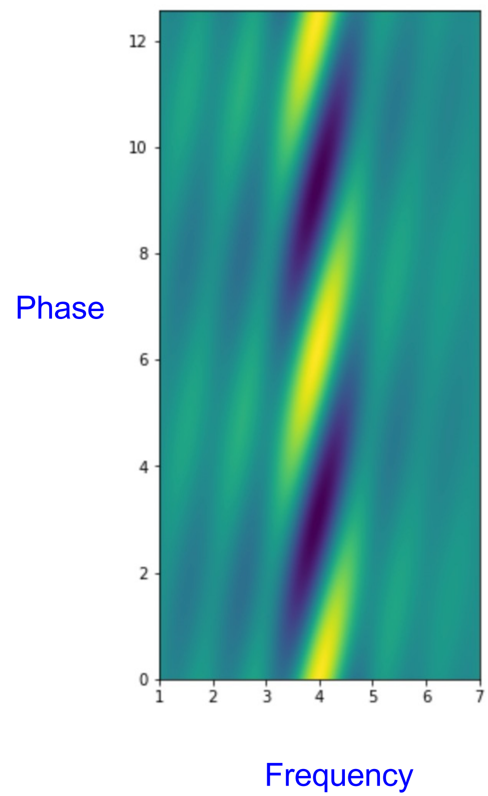
Computer Science

From last time….

Computer Science

Putting this together in a 3D surface plot with x = frequency and y = phase of the "probe wave," we can see that the maximum 0.5 is achieved only for the exact same frequency of 4 Hz and a phase difference of 0, $2\pi, 4\pi, etc.$





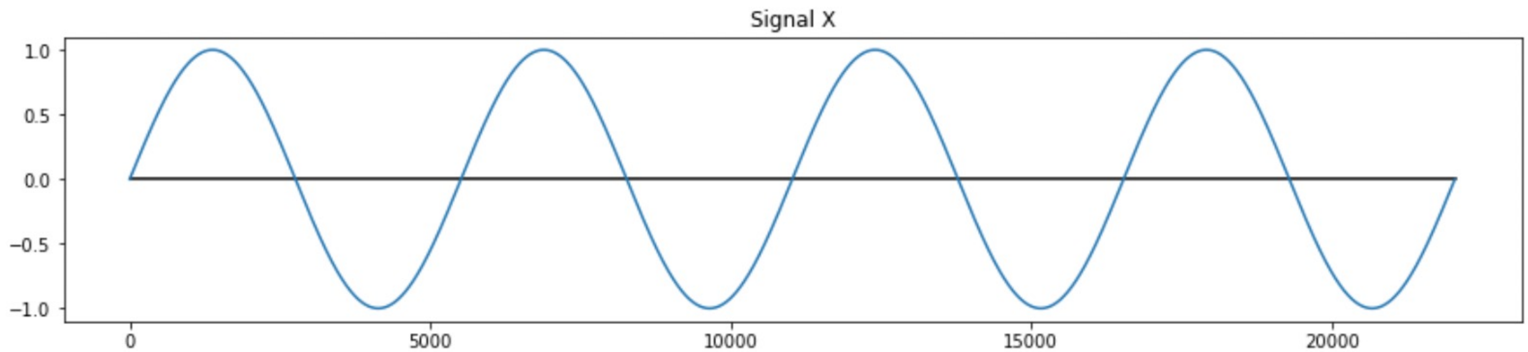
From last time…. Correction!

Computer Science

How to apply this to signals?

$$\rho_{X,Y} = \frac{E(X * Y) - \mu_X * \mu_Y}{\sigma_X * \sigma_Y}$$

We can simplify, since the mean of a sine waves is very close to 0.0:




```
mean(X) = 0.0
std(X) = 0.7071067811865476
```

Thus, the correlation of two sine waves is

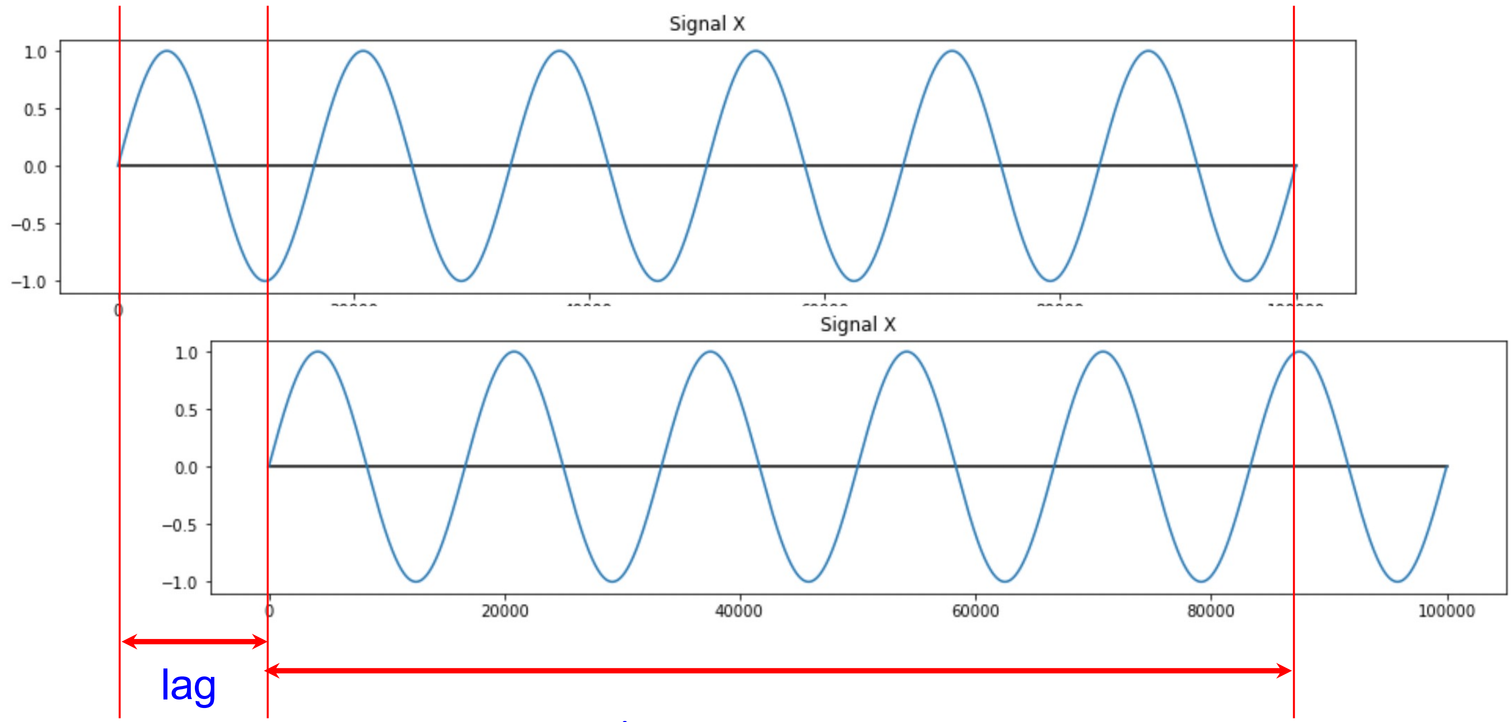
Dot Product

$$\frac{mean(X \cdot Y)}{\sigma_X \cdot \sigma_Y} = \frac{\sum_{k=0}^{N} X[k] \cdot Y[k]}{N \cdot \sigma_X \cdot \sigma_Y} = \frac{X@Y}{N \cdot \sigma_X \cdot \sigma_Y}$$

Fundamental Frequency Detection

As a warm up, we can explore the use of correlation to find the fundamental frequency of a signal…. by using the correlation of a signal with itself…

Auto-Correlation is the correlation of a signal with itself, with one detail: we will shift the signal before we perform the correlation. The shift in the signal is called the "lag" and is measured in units of samples:




```
AutoCorrelation(X,lag) =
        Correlation( X[: -lag], X[lag:] )
```

Fundamental Frequency Detection

If we plot the Auto-Correlation vs lags in the range 0 .. len(X)/2, the "peaks" in the resulting signal – remember, practically everything in this course is a signal! – show where the periods of the signal line up, and hence measure the period(s) of the periodic signal.

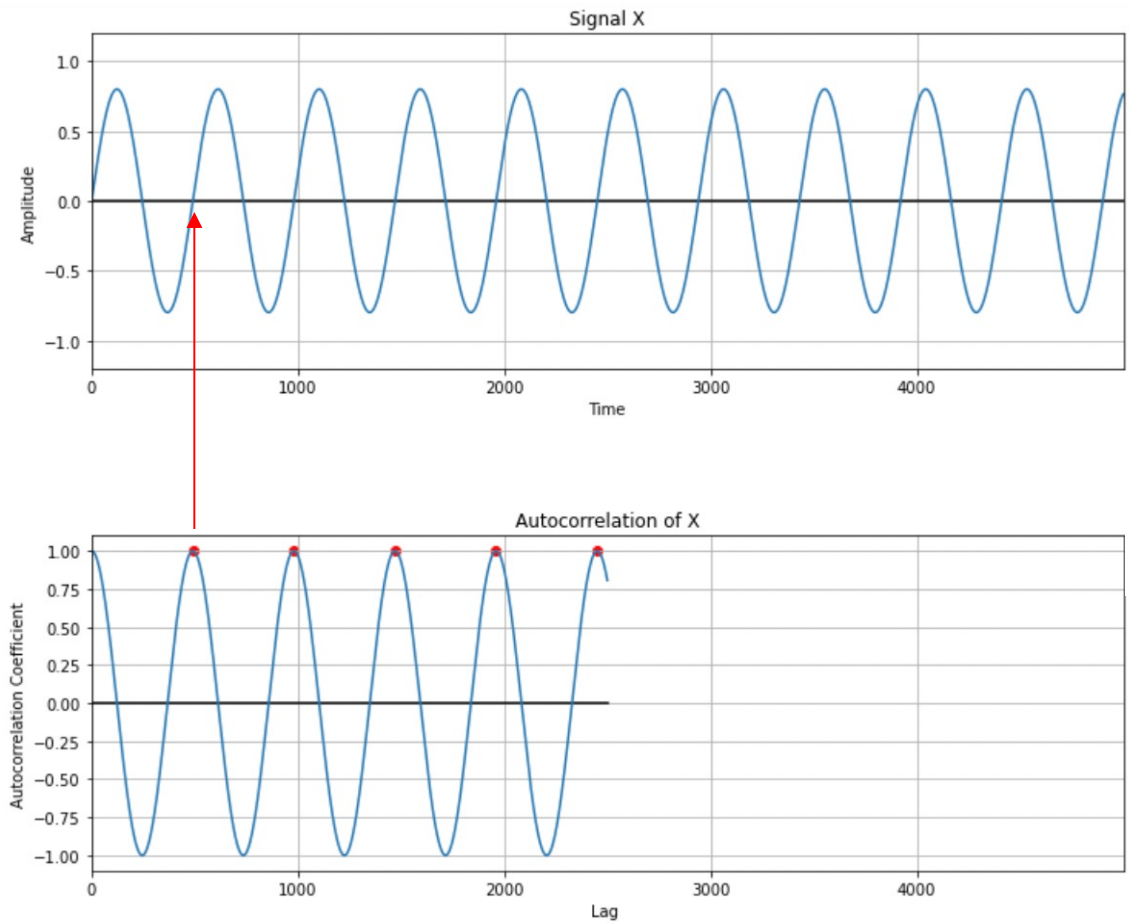
Several things to note:

- This works (in principle) for any periodic signal (not just simple sine waves);
- It is only practical to measure signals which have at least two full periods, which means you only consider lags up to len(X)/2;
- The "fundamental frequency" f_0 is the inverse of the shortest period; you will typically get peaks at f_0, $2f_0$, $3f_0$, …

Fundamental Frequency Detection

Computer Science

Examples of Peak Picking to determine F0:




```
Period found at 490 time units.
Fundamental Frequency =  90.0 Hz
```

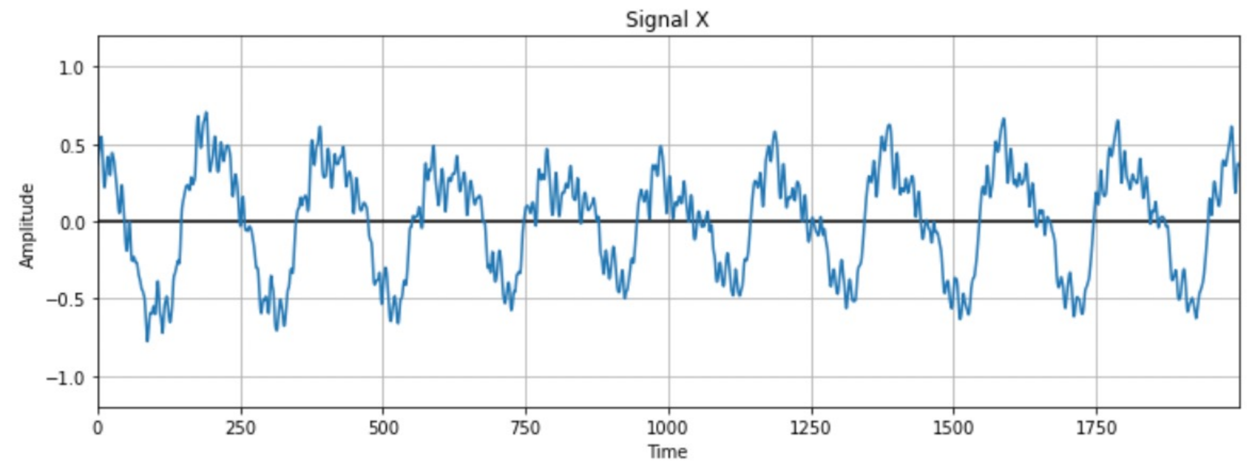
Calculation of F0:

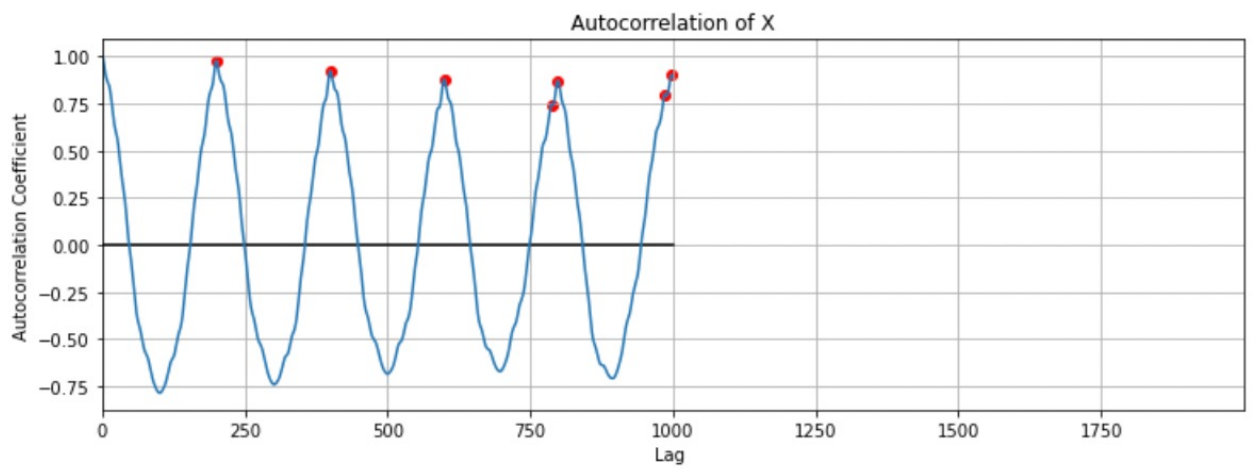
$$\text{Period in Seconds} = \frac{\text{Number of Samples}}{\text{Sample Rate}}$$

$$\text{Frequency in Hz} = \frac{1}{\text{Period in Seconds}}$$

Fundamental Frequency Detection

Examples of Peak Picking to determine F0:

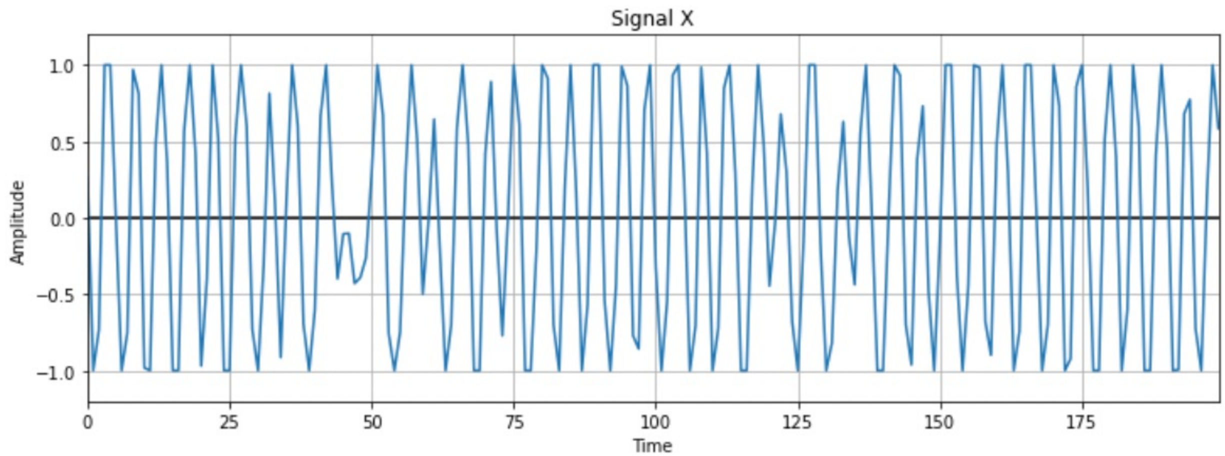


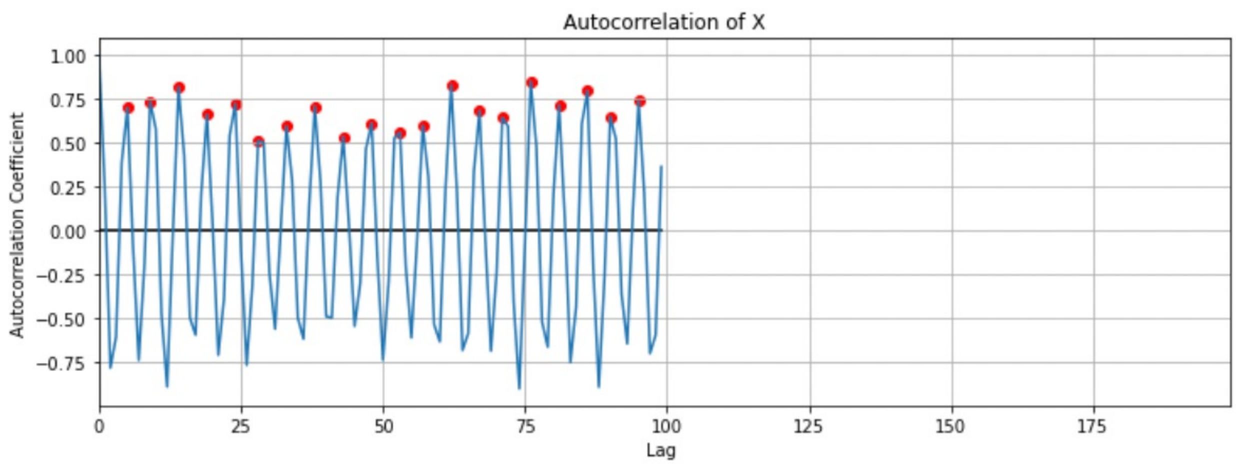


Fundamental Frequency Detection

Computer Science

Examples of Peak Picking to determine F0:





Fundamental Frequency Detection

Three issues with any frequency detector:

(1) Which peak in the autocorrelation graph is the fundamental frequency F0?

(2) Once we have selected a peak, how accurate is the method? How can we make it more accurate?

(3) If this works for single notes (monophonic signals), can it be extended to work for multiple simultaneous notes (polyphonic signals)?

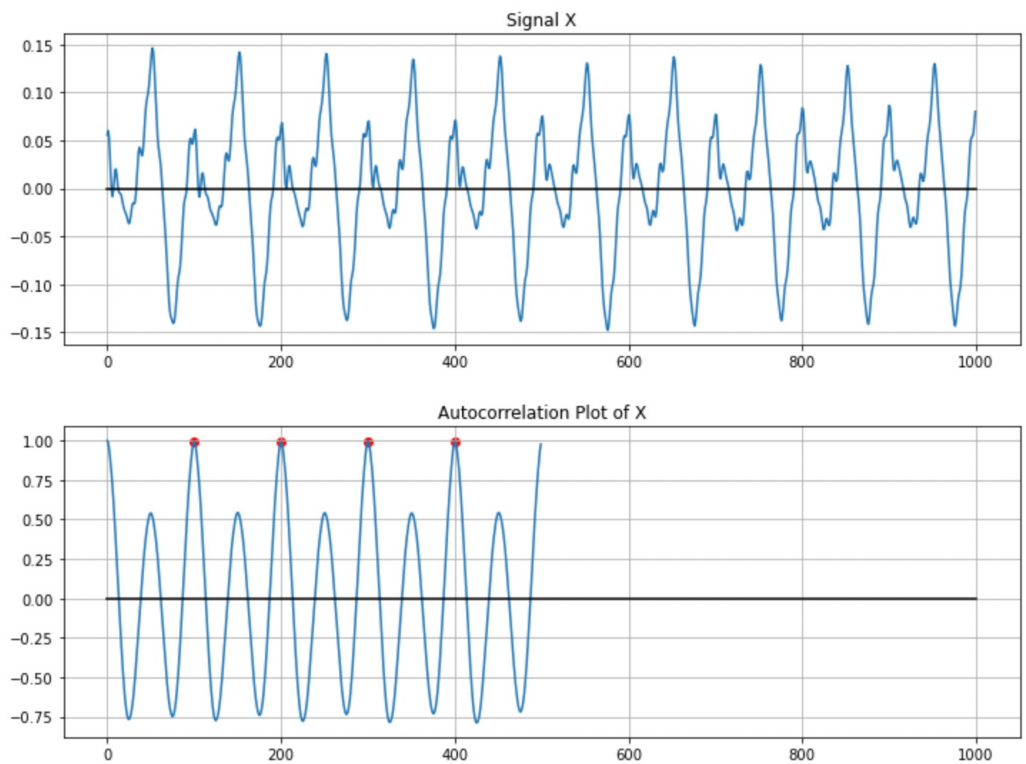
There are good solutions to (2), so we will discuss that issue first for the AC method.

Fundamental Frequency Detection

Computer Science

How accurate is the AC method?   How would we test this?

For example, here is the steel string guitar sound (not the beginning):




```
Detected Peaks:     [100 200 300 400]
Detected Period:     100.0
Detected Freq:       220.5
```

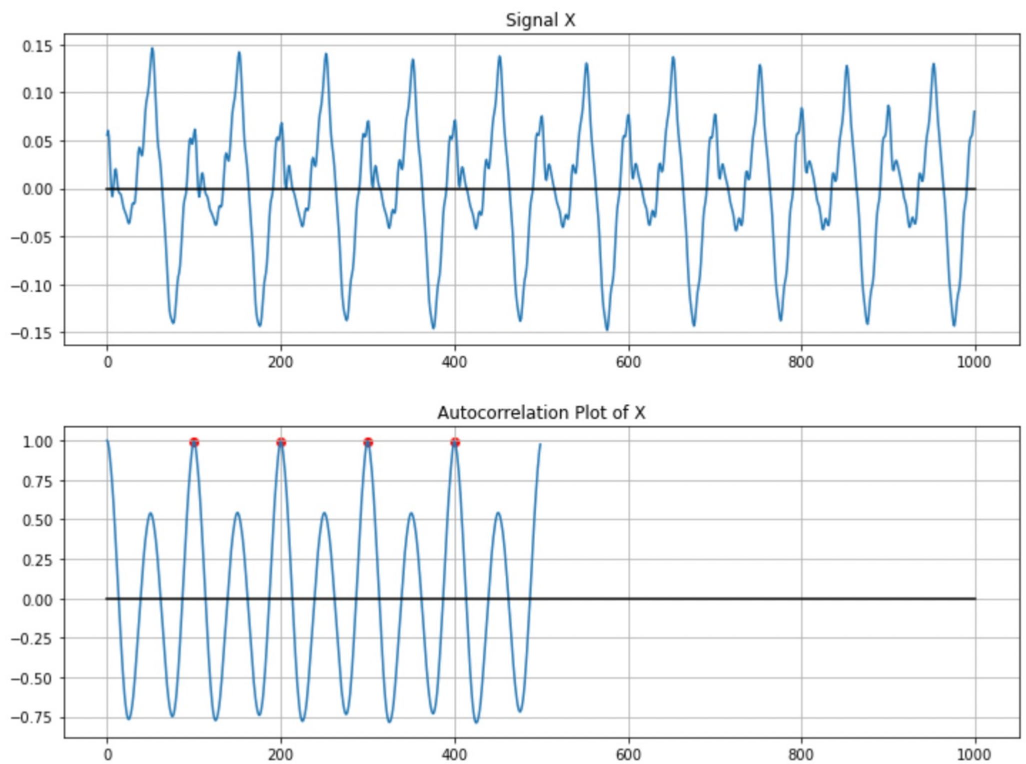
How do I decide if this pitch estimate is correct??

Fundamental Frequency Detection

Computer Science

How accurate is the AC method?   How would we test this?

For example, here is the steel string guitar sound (not the beginning):




```
Detected Peaks:     [100 200 300 400]
Detected Period:     100.0
Detected Freq:       220.5
```

How do I decide if this pitch estimate is correct??

I can't, without knowing the actual pitch outside my measurement!

So I can ONLY find out if two pitch algorithms are consistent!

Frequency Detection: How to test?

So we will adopt the following strategy:

Create a synthetic signal with a precise pitch (the “piano key frequencies”):

1. Simple sine waves of one frequency;
2. Combinations of several harmonic frequencies;
3. Signals created from the sample spectra (in the Intro notebook).

Determine the pitch detected by the algorithm, and calculate the

1. Percent error:

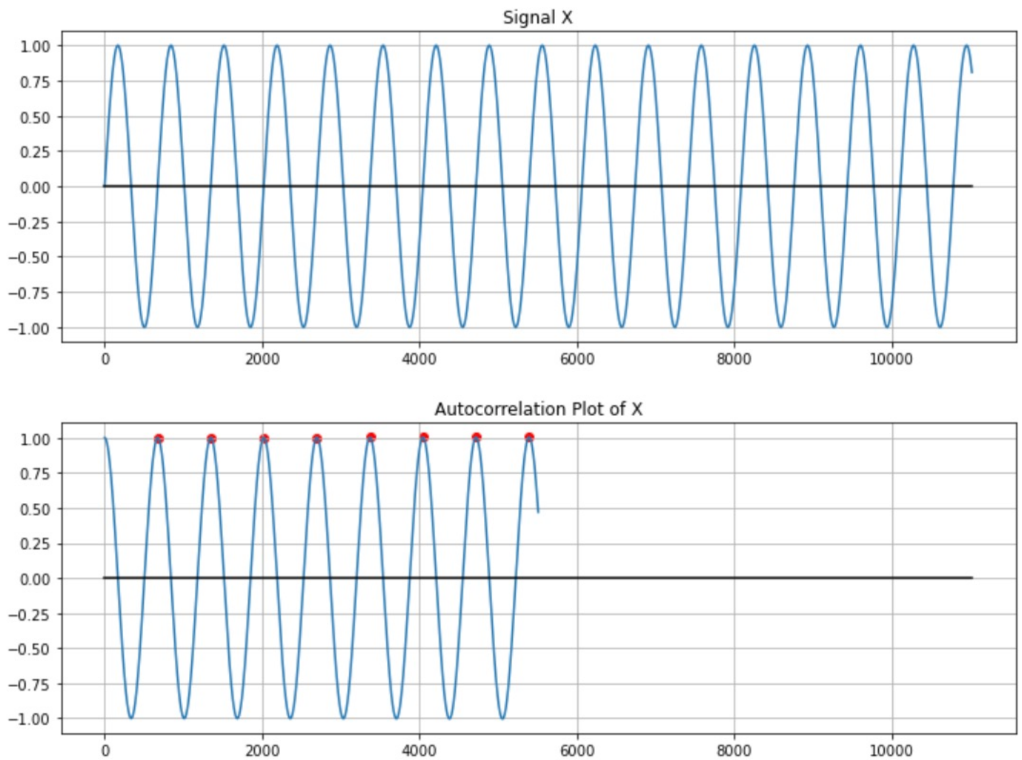
2. Error in “cents” (1/100 of a semitone – distance between piano key frequencies).

Cents are a good perceptual measure; humans without musical talent can usually distinguish between frequencies differing by 25 cents; those with musical talent can often distinguish between 2 – 5 cents.

Frequency Detection: How to test?

Computer Science

For the first test, we will create simple sine waves with the piano key frequencies, of duration 0.2 seconds. Here are some results:

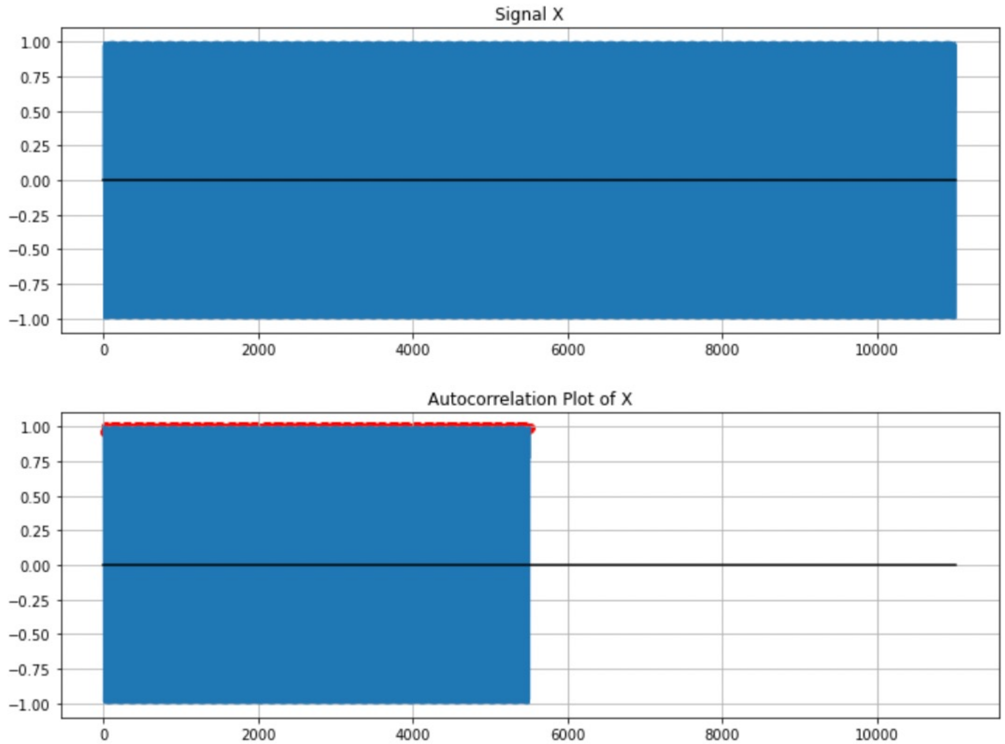



```
Detected Period:     675.0
Actual Period:       674.2460347761601
Detected Freq:       32.666666666666664
Actual Freq:         32.703195662574835
Percent Error:       -0.1117 %
Cents Error:         -1.9348
```

Frequency Detection: How to test?

Computer Science

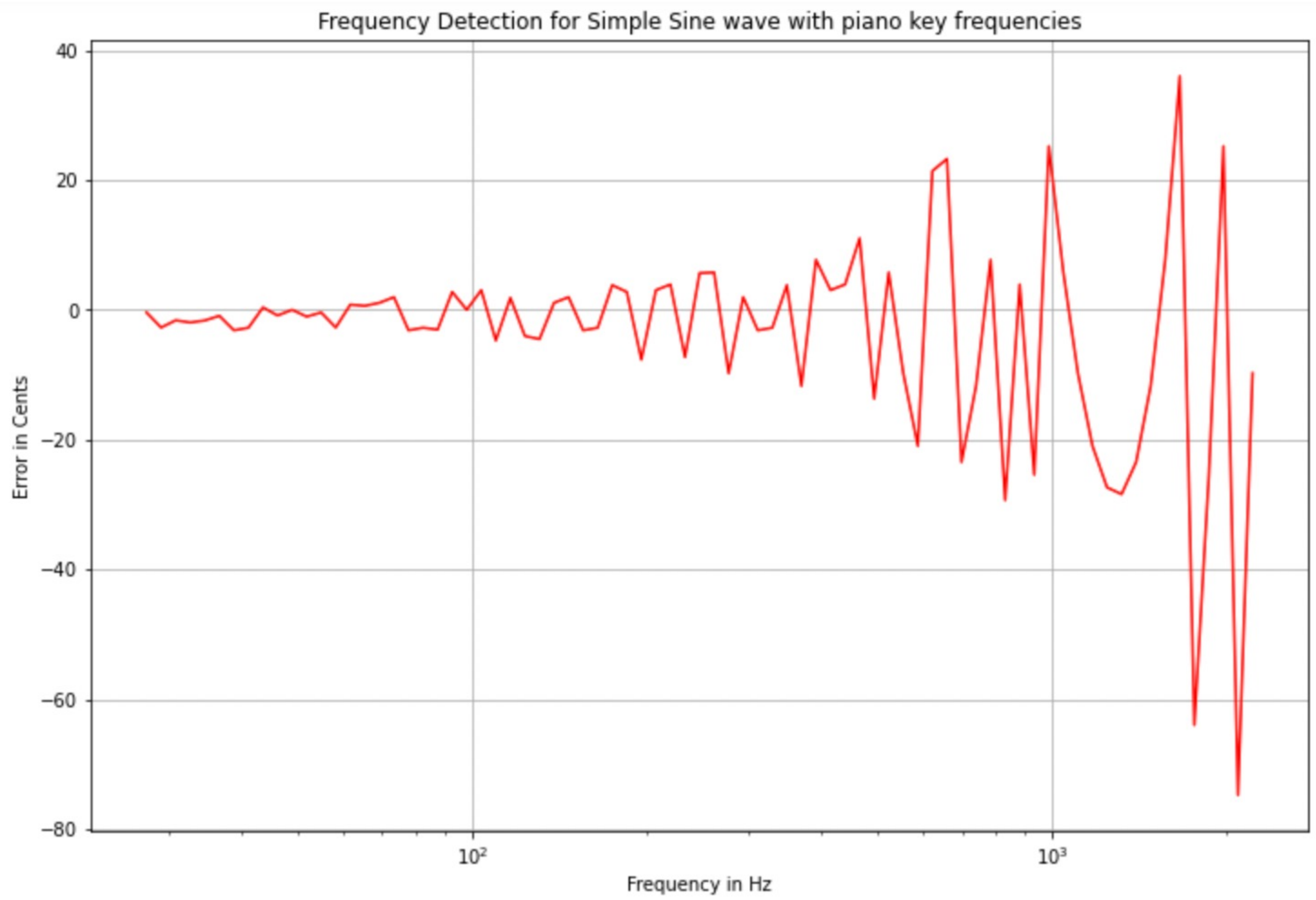
For the first test, we will create simple sine waves with the piano key frequencies, of duration 0.2 seconds. Here are some results:




```
Detected Period:    11.0
Actual Period:      10.535094293377474
Detected Freq:      2004.5454545454545
Actual Freq:        2093.004522404795
Percent Error:      -4.2264 %
Cents Error:        -74.7604
```

Frequency Detection: How to test?

Overall, the algorithm did worse as the frequencies got higher. WHY??



Frequency Detection: How to test?

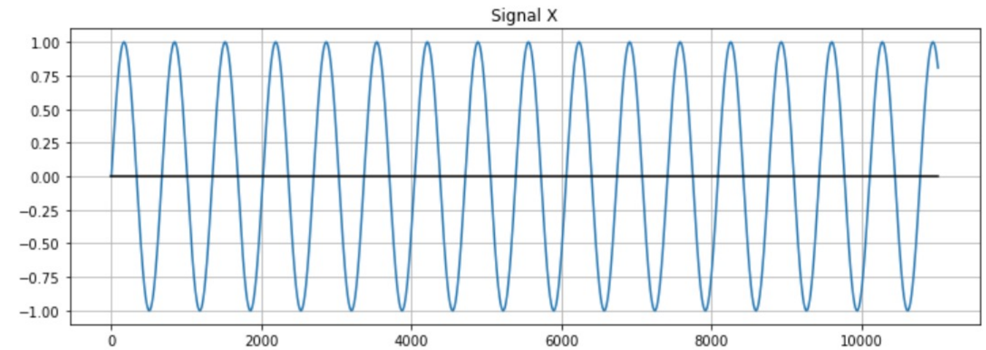
In general, it did WORSE as the frequency increased. WHY?

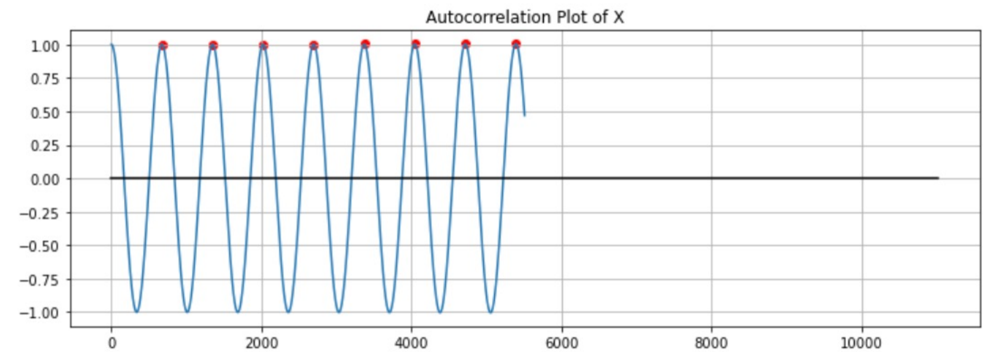
Very likely because as the frequency increases, the sampling rate produces a less and less accurate approximation of the signal.

Note that you can ONLY find period corresponding to a lag of an integral number of samples, which means that your accuracy is limited to the frequency distance between different lags.

For example, at C1, we got an error of -1.93 cents.

Here are the possible frequencies that can be detected at integral lags in this range:






```
Detected Period:     675.0
Actual Period:       674.2460347761601
Detected Freq:       32.666666666666664
Actual Freq:         32.703195662574835
Percent Error:       -0.1117 %
Cents Error:         -1.9348
PE Resolution:       -0.147929
Cents Resolution:    -2.562893
```

Lag	Frequency	Hz Error	Cents Error
676	32.6183432	-0.0483235	-2.5628932
675	32.6666667	0	0
674	32.7151335	0.04846686	2.56669291

Frequency Detection: How to test?

Computer Science

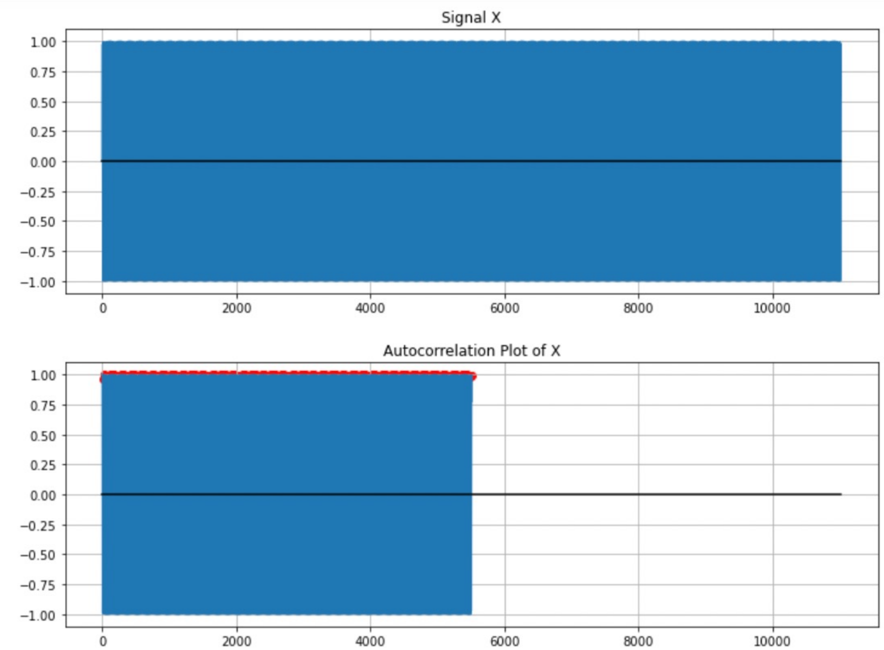
In general, it did WORSE as the frequency increased.   WHY?

Very likely because as the frequency increases, the sampling rate produces a less and less accurate approximation of the signal.

Note that you can ONLY find period corresponding to a lag of an integral number of samples, which means that your accuracy is limited to the frequency distance between different lags.

At C7, we got an error of -74.76 cents.

Here are the possible frequencies that can be detected at integral lags in this range:




```
Detected Period:      11.0
Actual Period:        10.535094293377474
Detected Freq:        2004.5454545454545
Actual Freq:          2093.004522404795
Percent Error:        -4.2264 %
Cents Error:          -74.7604
PE Resolution:        -8.333333
Cents Resolution:     -150.637059
```

Lag	Frequency	Hz Error	Cents Error
12	1837.5	-167.04545	-150.63706
11	2004.54545	0	0
10	2205	200.454545	165.004228

Frequency Detection: How to test?

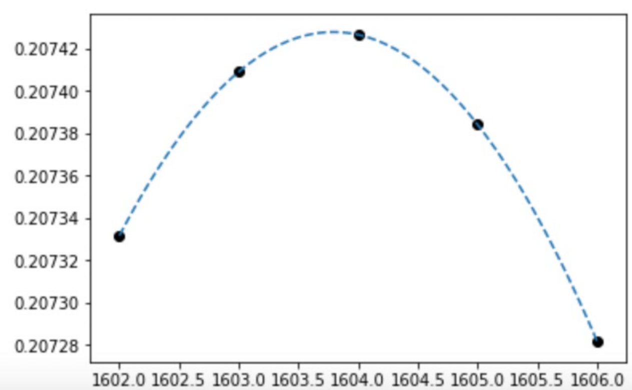
Computer Science

One idea IS to interpolate the resulting AC graph to find more accurate (meaning, floating point) sample locations for the peaks:

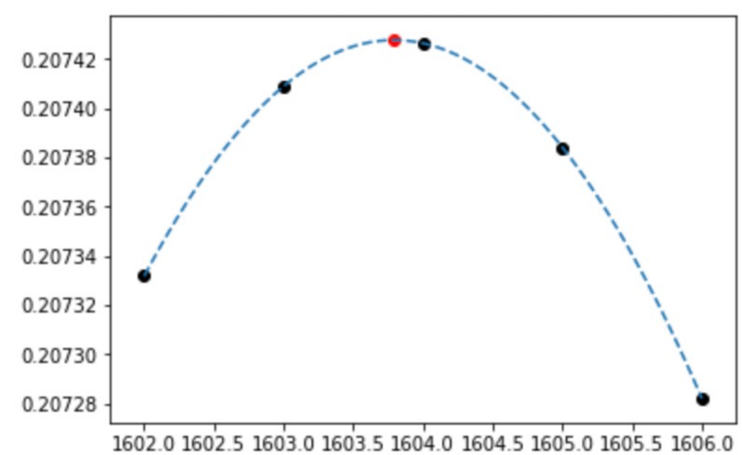
Improving the calculation using parabolic interpolation

When the period does not correspond to an integral number of samples, then interpolation can be used to improve the results. For example, in the previous example, we had the following detail around the first peak:

```
]: T = np.arange(1602,1607)
A1 = A[1602:1607]
f = interp1d(T,A1,kind='quadratic')
Tc = np.linspace(1602,1606,100)
Ac = [ f(x) for x in Tc ]
plt.plot(Tc,Ac,linestyle='--')
plt.scatter(T,A1,color='k')
plt.show()
```



Peak at sample 1604: 0.20742644

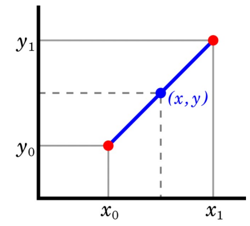


Peak at sample 1603.7914: 0.20742774

Frequency Detection: How to test?

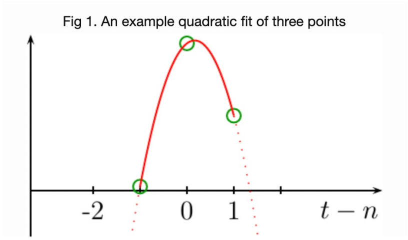
Interpolation can be done in MANY different ways, and more sophisticated methods are slower and (perhaps) more accurate:

Linear: fit a line to each pair of points,
(you did this in HW 01!)



Quadratic: Fit a parabola to the 3 points around the peak by solving the appropriate equations (remember CS 132?):

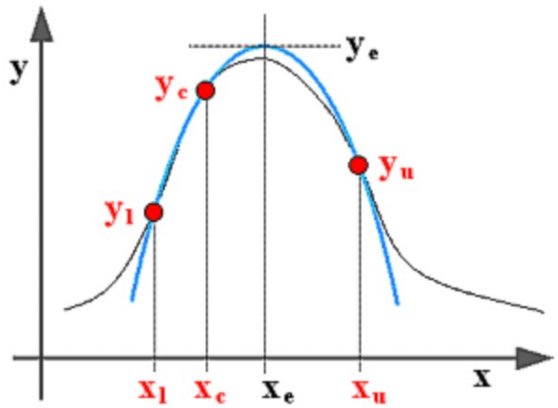
$A x_1^2 + B x_1 + C = y_1$
$A x_2^2 + B x_2 + C = y_2$
$A x_3^2 + B x_3 + C = y_3$



Cubic, Quartic, etc. (A degree K polynomial can be fitted exactly to K+1 points)

Frequency Detection: How to test?

Quadratic interpolation is widely used (and often criticized – see the link on the class web page) to find a more exact x coordinate for a peak. Once you have the parameters A, B, and C, you can take the derivative of the formula to find the maximum (remember Calc 1?), and the x coordinate *could be* a more accurate estimate of the period:



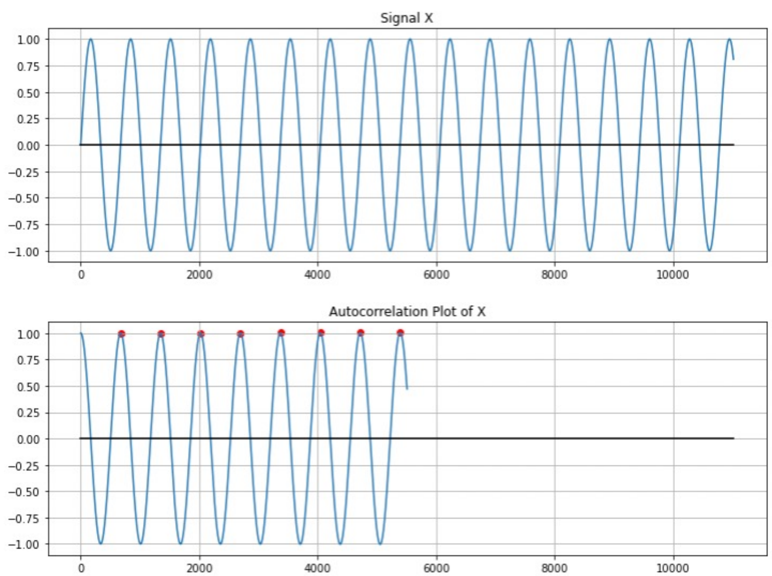
This can be done to refine peaks for F0 determination, and (next week) for refining peaks in spectra returned by the Fourier Transform.

Frequency Detection: How to test?

Computer Science

How does this strategy work? This CLEARLY does better with interpolation!

```
Single Sine wave of frequency 32.7032
```



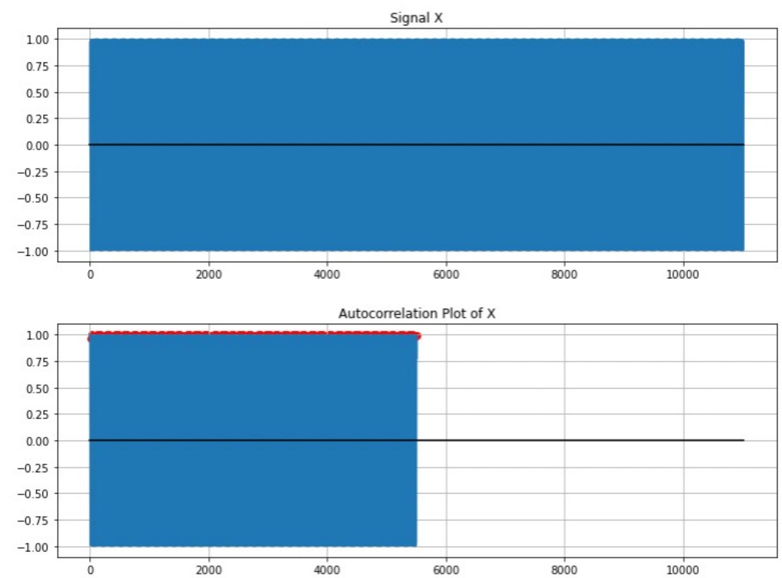

```
No interpolation

Detected Period:    675.0
Actual Period:      674.246
Detected Freq:      32.6667
Actual Freq:        32.7032
Percent Error:      -0.1117 %
Cents Error:        -1.9348
```

Cents Error: -1.9348

```
Interpolation

Detected Period:    674.6396
Actual Period:      674.246
Detected Freq:      32.6841
Actual Freq:        32.7032
Percent Error:      -0.0583 %
Cents Error:        -1.0103
```

Cents Error: -1.0103

```
Single Sine wave of frequency 2093.0045
```




```
No interpolation

Detected Period:    11.0
Actual Period:      10.5351
Detected Freq:      2004.5455
Actual Freq:        2093.0045
Percent Error:      -4.2264 %
Cents Error:        -74.7604
```

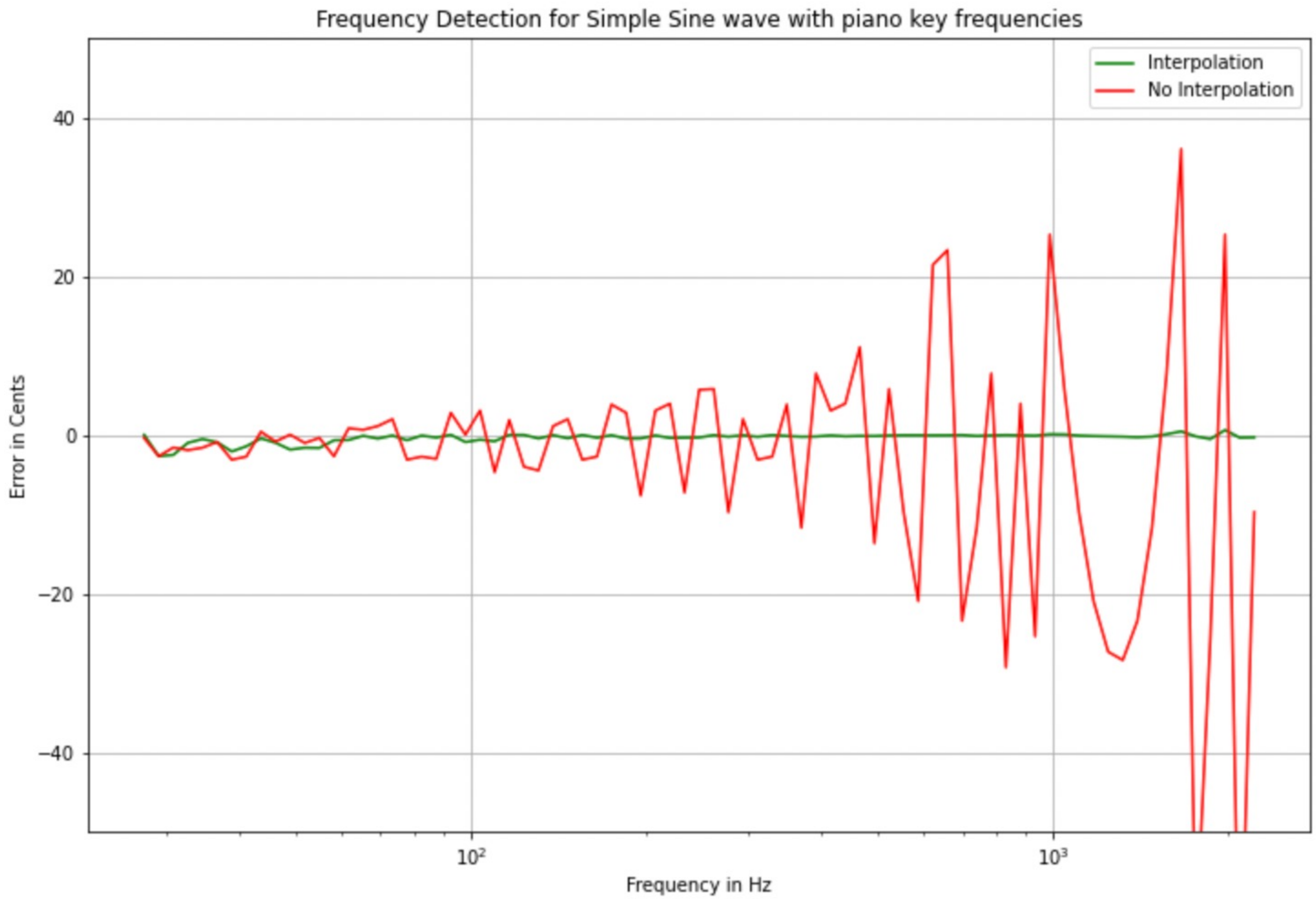
Cents Error: -73.7604

```
Interpolation

Detected Period:    10.5373
Actual Period:      10.5351
Detected Freq:      2092.5726
Actual Freq:        2093.0045
Percent Error:      -0.0206 %
Cents Error:        -0.3573
```

Cents Error: -0.3573

Frequency Detection: How to test?

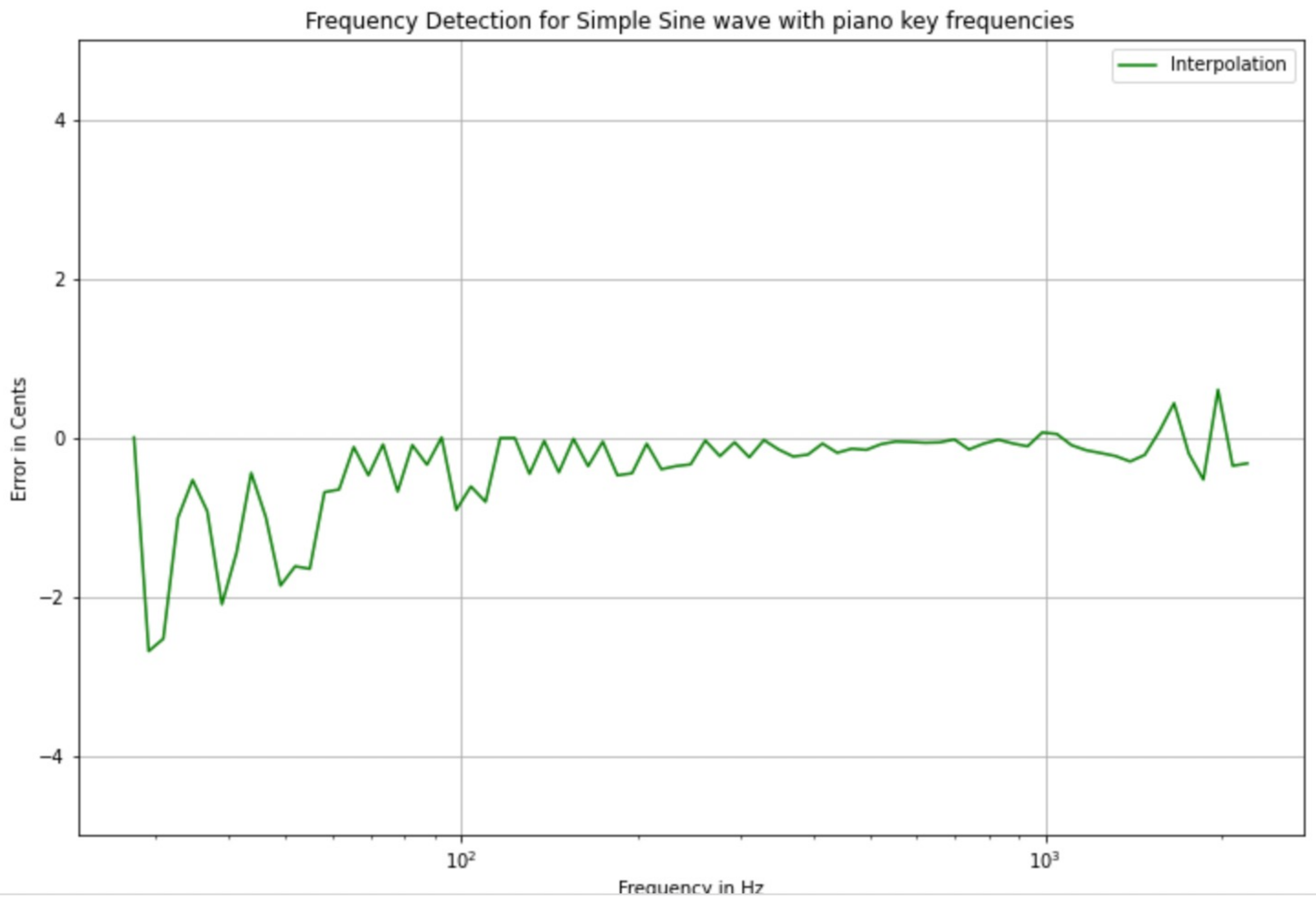
How does this strategy work?   This CLEARLY does better with interpolation!



Frequency Detection: How to test?

Computer Science

However, there is still a lack of precision, but now at the LOW end!



Frequency Detection: How to test?

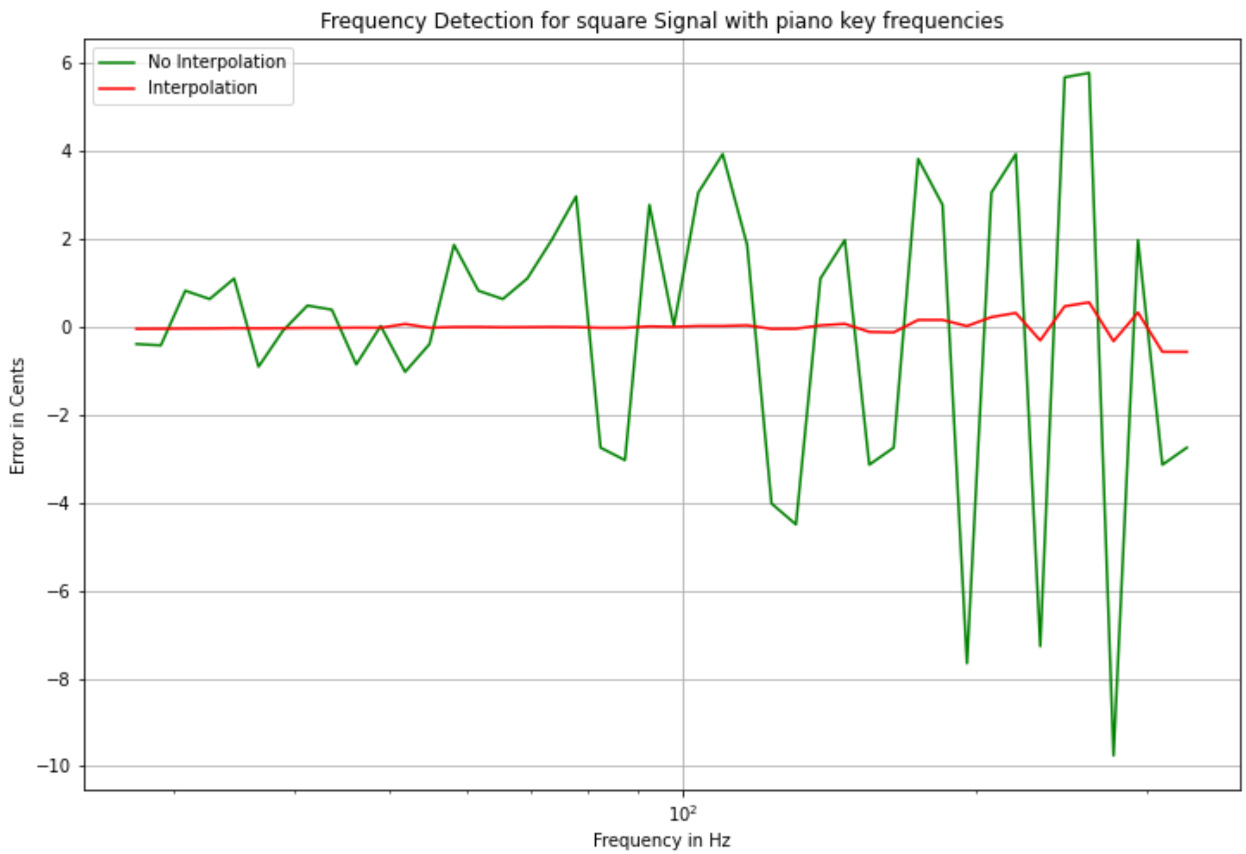
Computer Science

This is a problem with or without interpolation! This is probably because at lower frequencies, you don't have as many cycles to match in the overlap window, or perhaps because the interpolation points are *too* close together.

Frequency Detection: How to test?

Computer Science

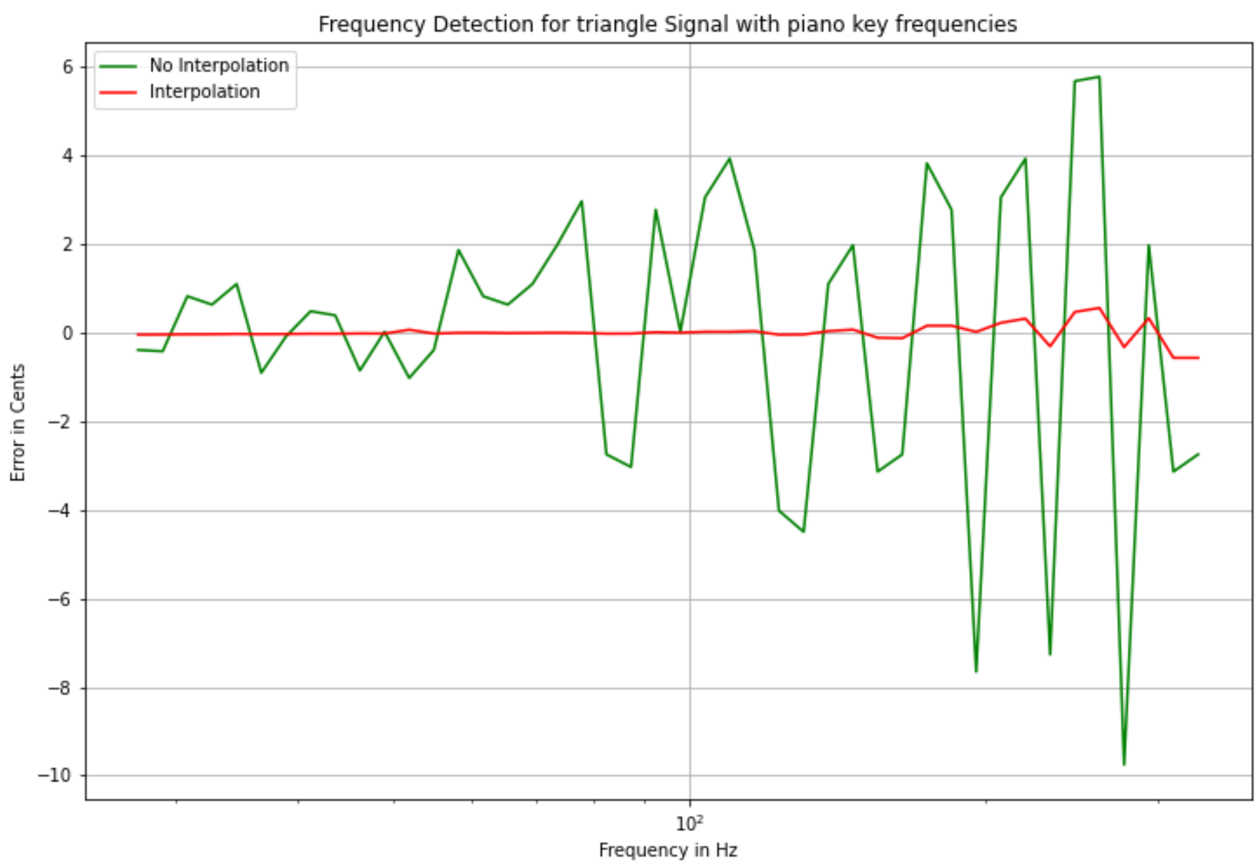
Results on sample spectra (square and triangle waves, steel string, clarinet, organ, and bell):



Frequency Detection: How to test?

Computer Science

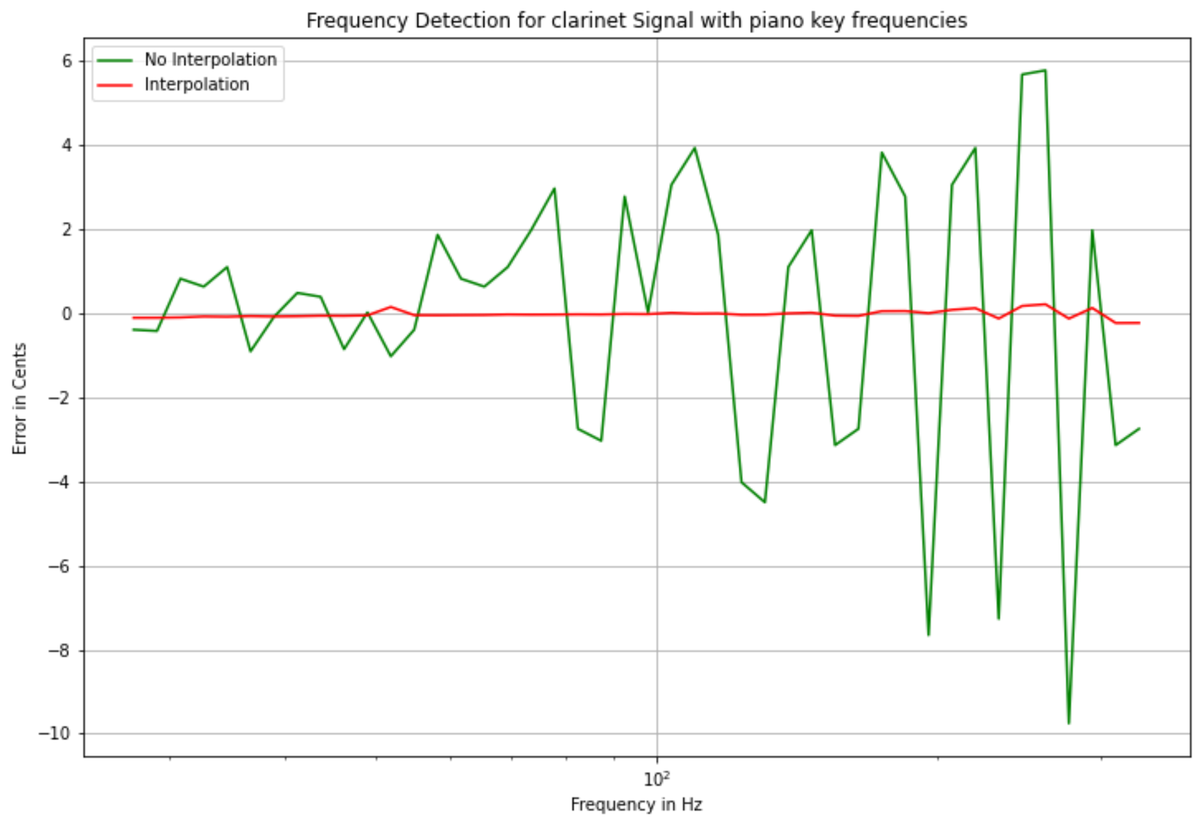
Results on sample spectra (square and triangle waves, steel string, clarinet, organ, and bell):



Frequency Detection: How to test?

Computer Science

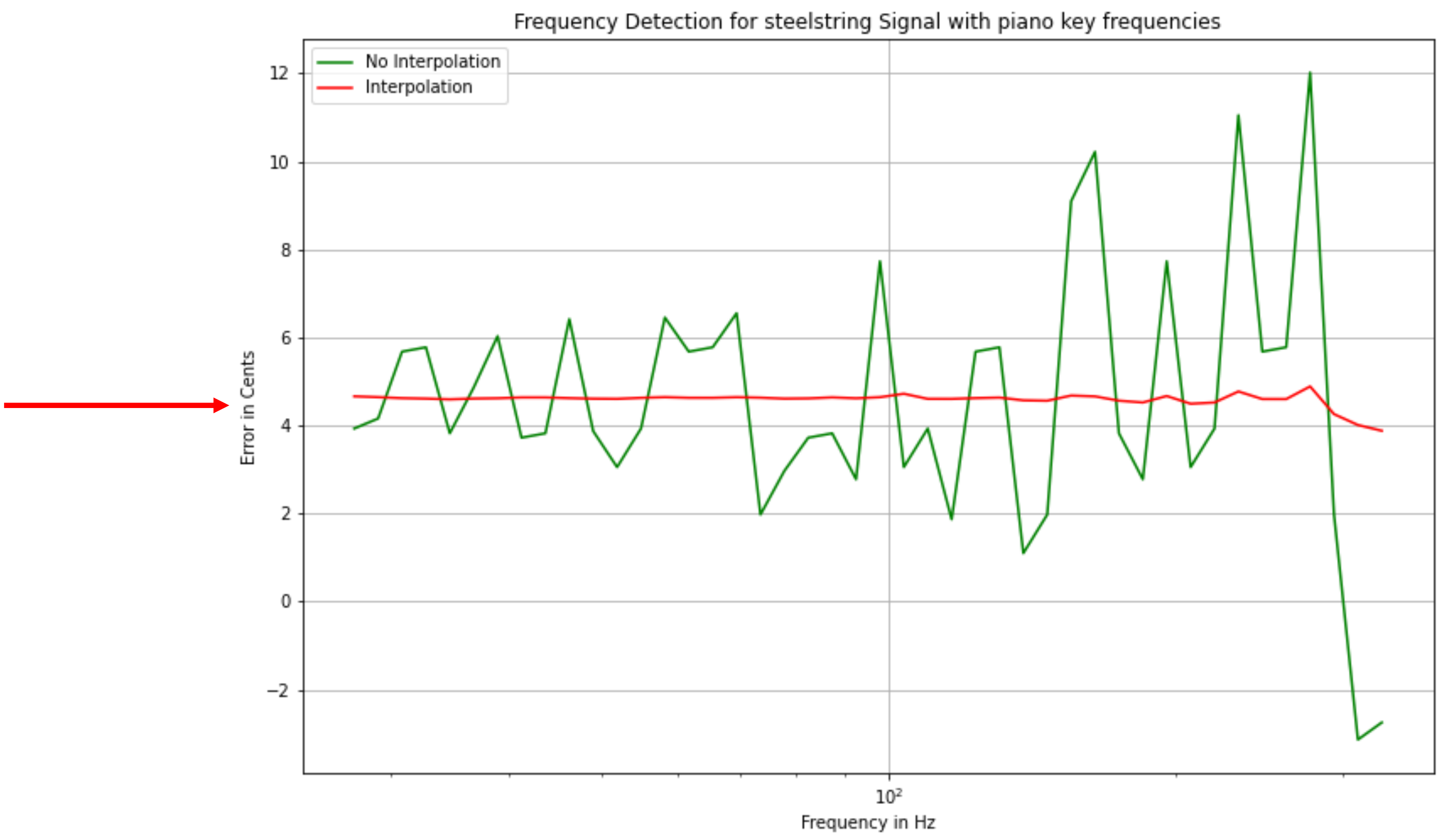
Results on sample spectra (square and triangle waves, steel string, clarinet, organ, and bell):



Frequency Detection: How to test?

Computer Science

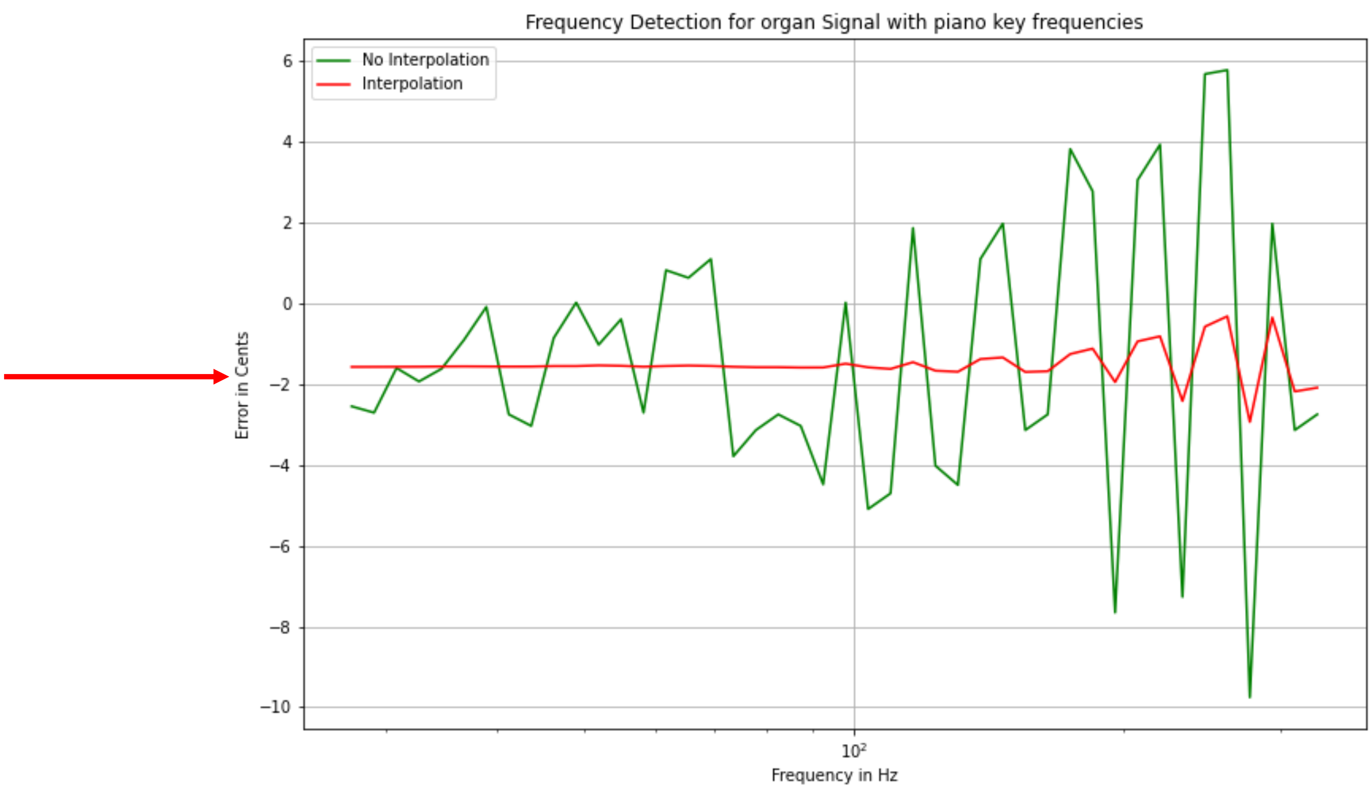
Results on sample spectra (square and triangle waves, steel string, clarinet, organ, and bell):



Frequency Detection: How to test?

Computer Science

Results on sample spectra (square and triangle waves, steel string, clarinet, organ, and bell):

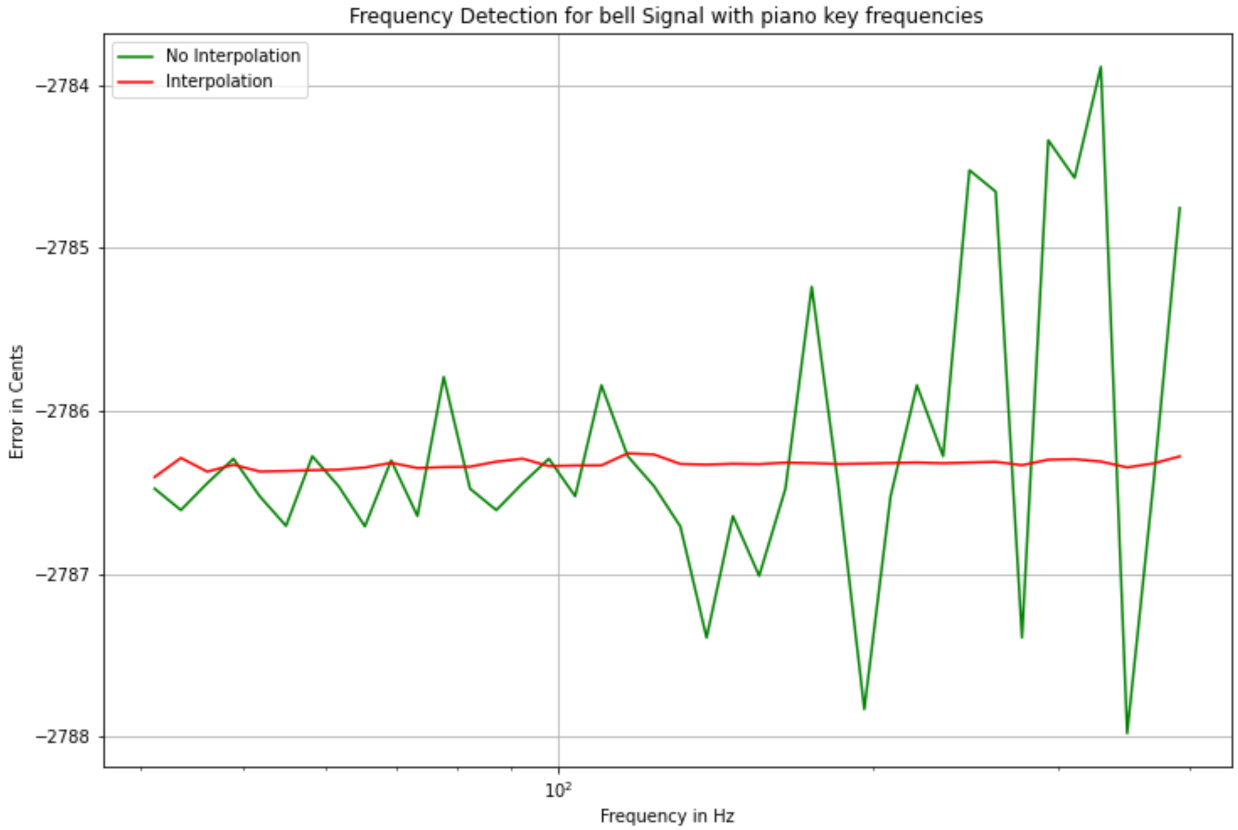


Frequency Detection: How to test?

Computer Science

Results on sample spectra (square and triangle waves, steel string, clarinet, organ, and bell):

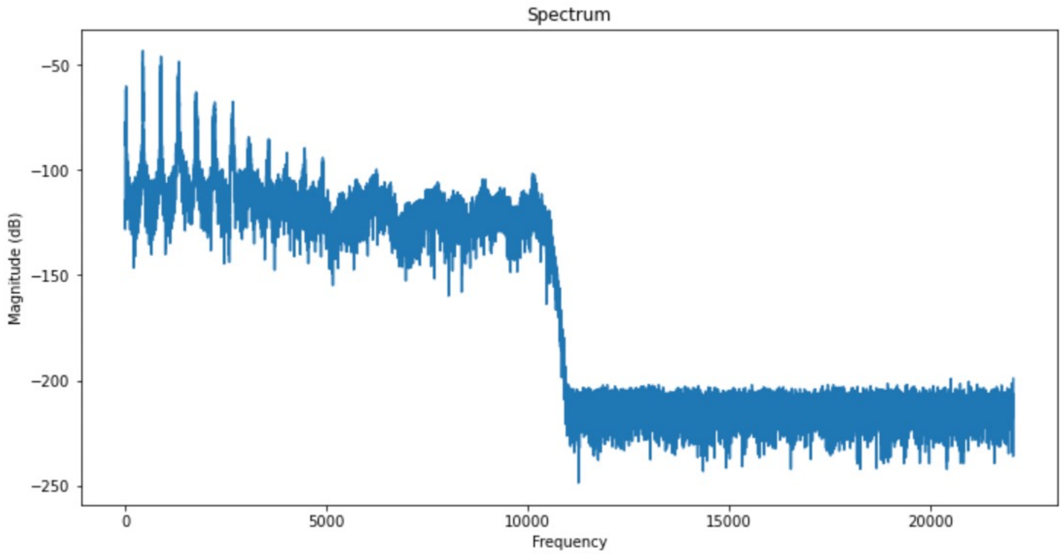
!!



Frequency Detection: How to test?

Results on real signals:

```
flute_A4_1_mezzo-forte_normal.wav
```



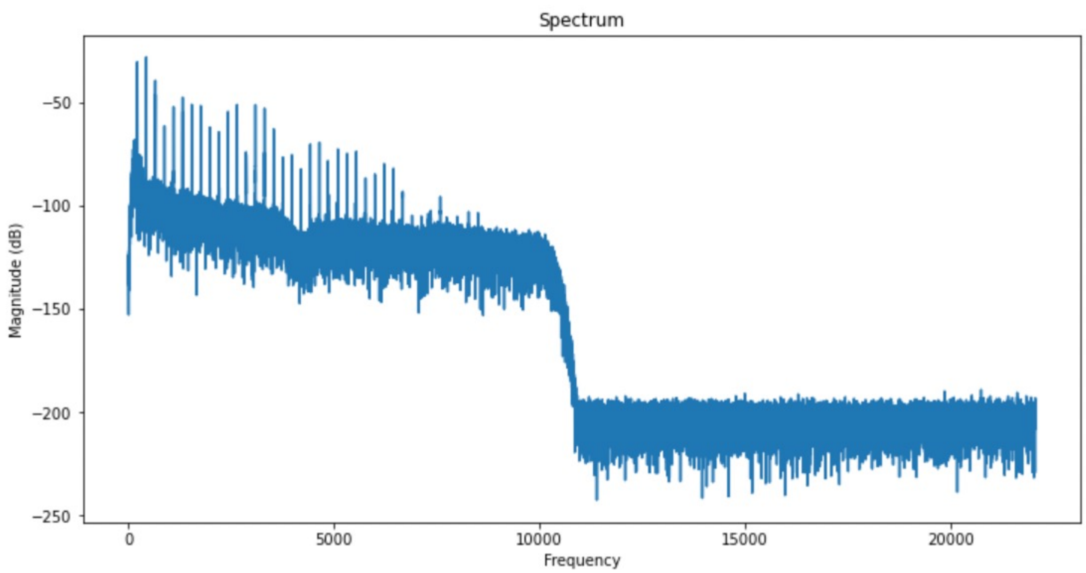

```
No interpolation

Measured Period: 100
Measured Frequency: 441.0

Interpolation

Measured Period: 99.5526
Measured Frequency: 442.9819
```

Frequency Detection: How to test?

Computer Science

Results on real signals:

```
SteelString.wav
```



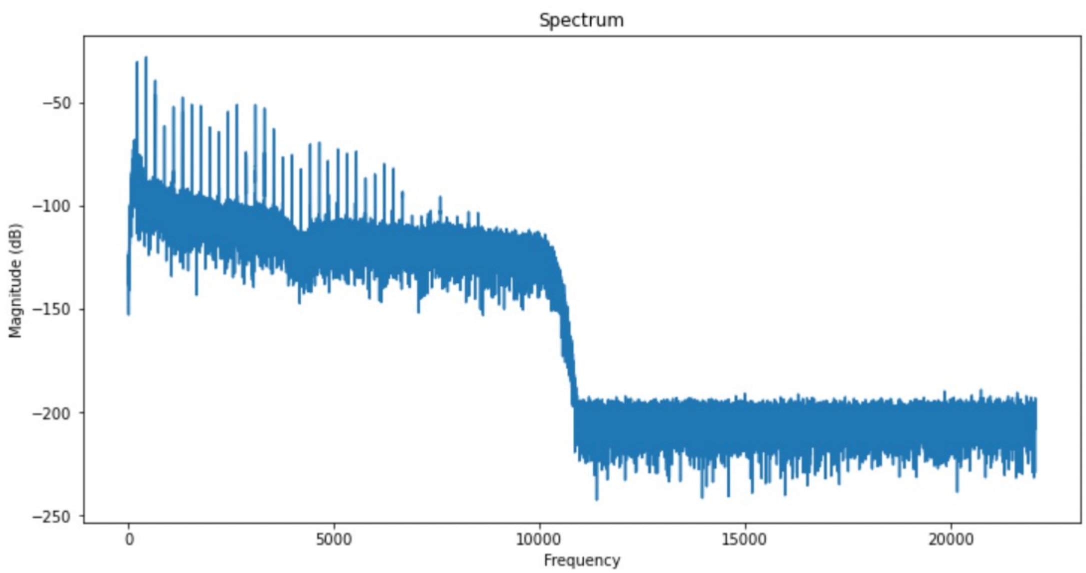

```
No interpolation

Measured Period: 200
Measured Frequency: 220.5

Interpolation

Measured Period: 199.7198
Measured Frequency: 220.8093
```

Frequency Detection: How to test?

Computer Science

Results on real signals:

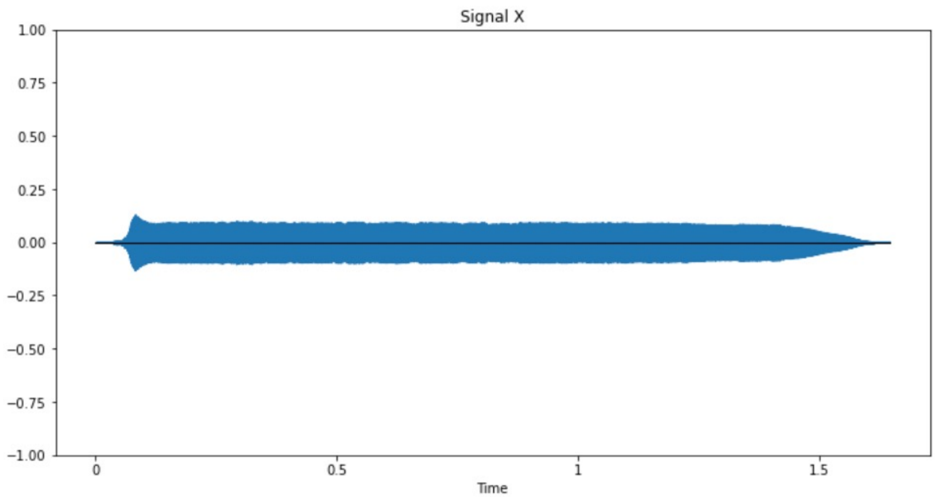
```
SteelString.wav
```

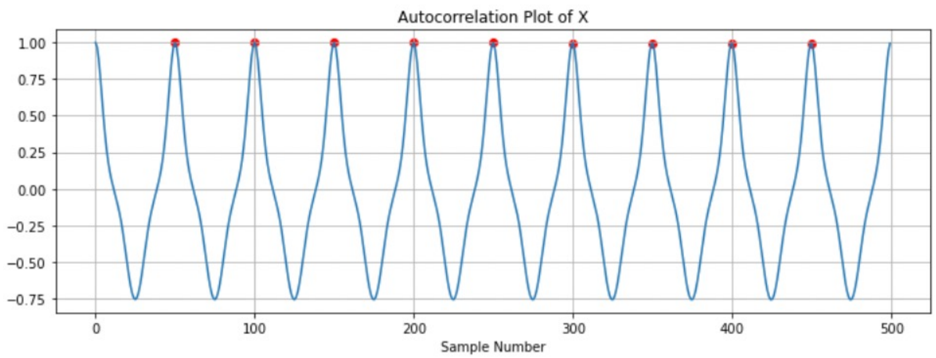



```
No interpolation

Measured Period: 200
Measured Frequency: 220.5

Interpolation

Measured Period: 199.7198
Measured Frequency: 220.8093
```

Frequency Detection: How to test?

For some instruments, it is easy to determine F0:

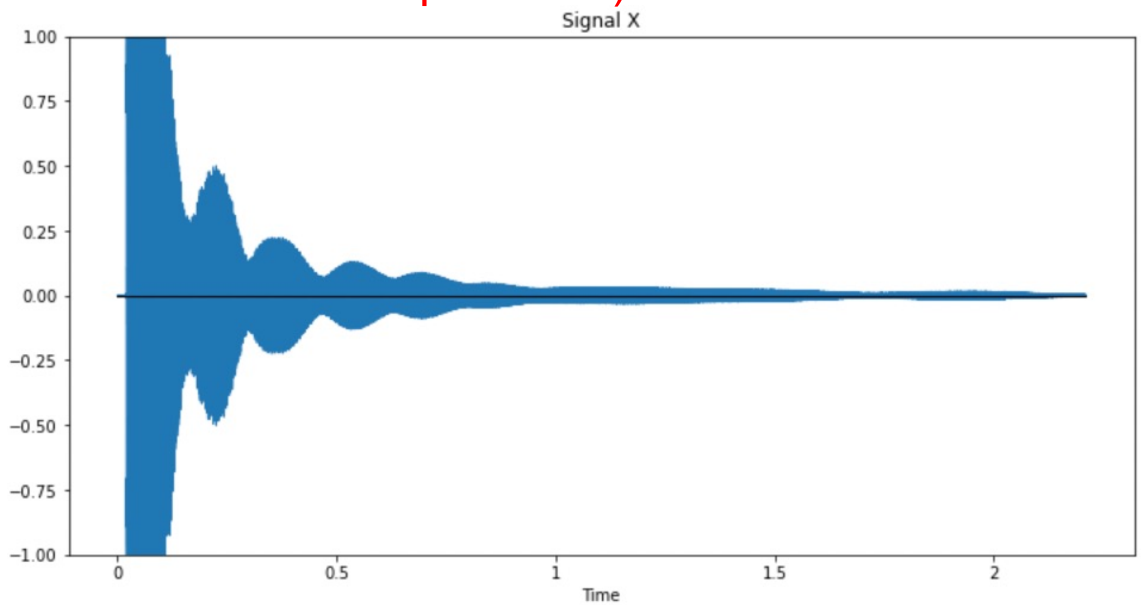


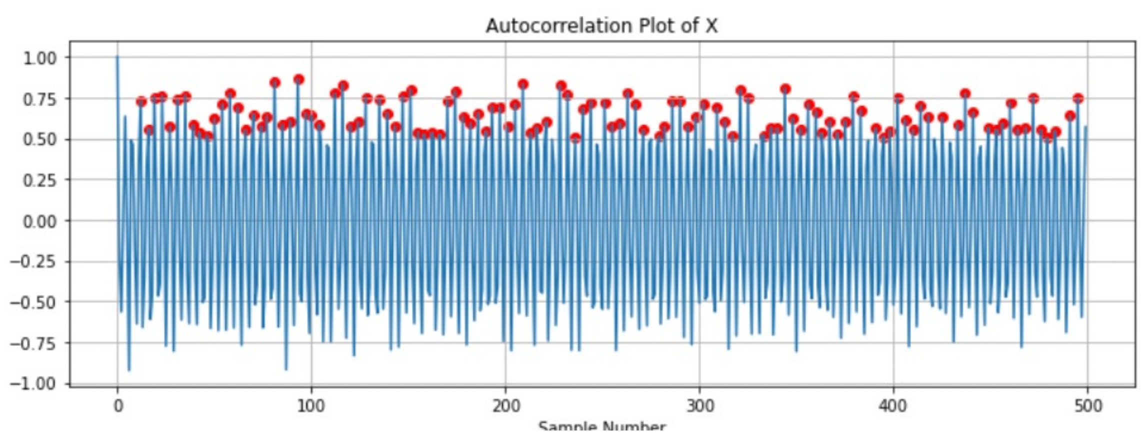


Frequency Detection: How to test?

Computer Science

For others, it is MUCH more complicated! This is a fundamental problem with pitch detection: Which AC peak is F)??





Frequency Detection: How to test?

Computer Science

For others, it is MUCH more complicated! This is a fundamental problem with pitch detection: Which AC peak is F)??