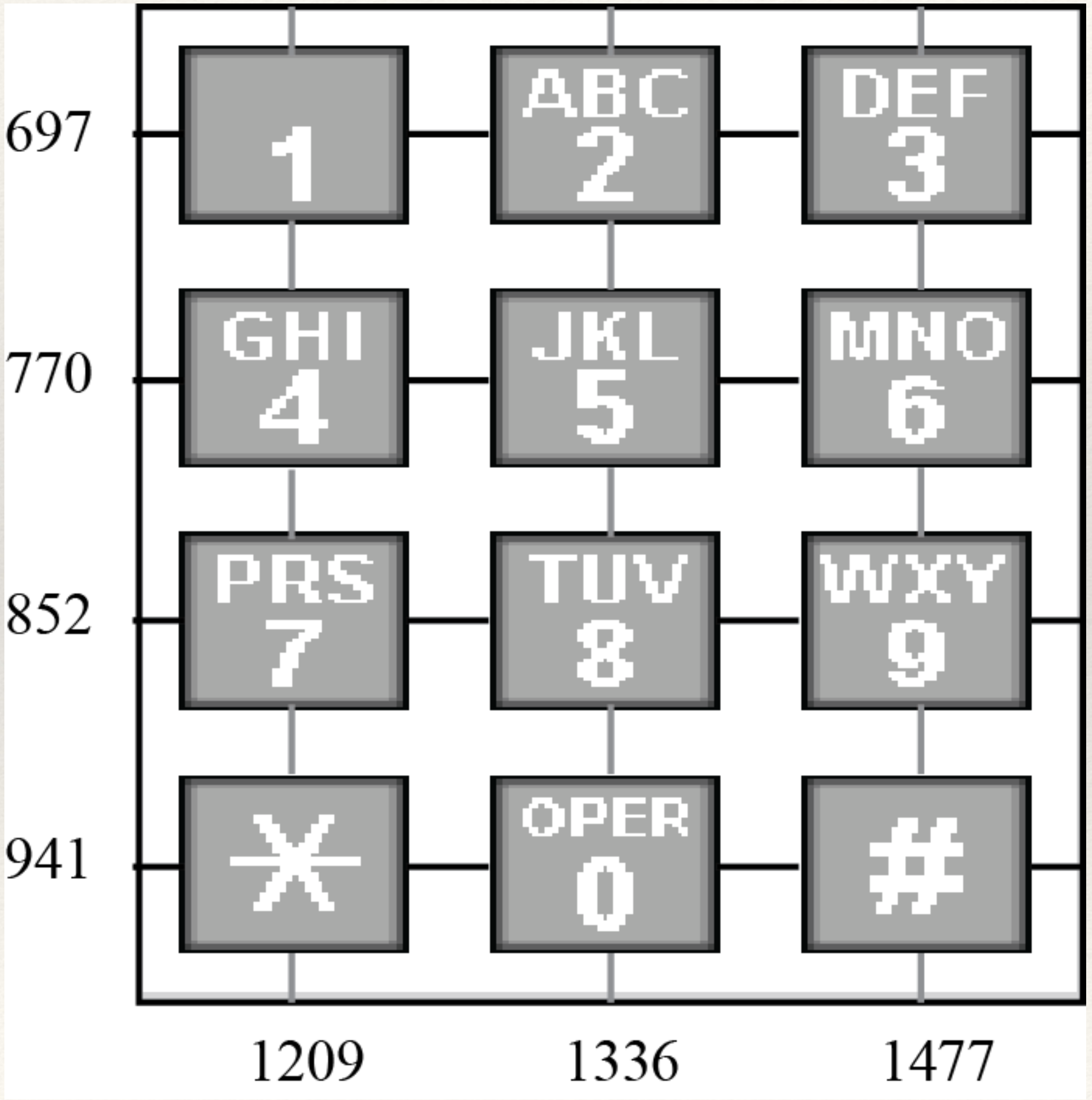
Understanding Phone dials

The phone keypad



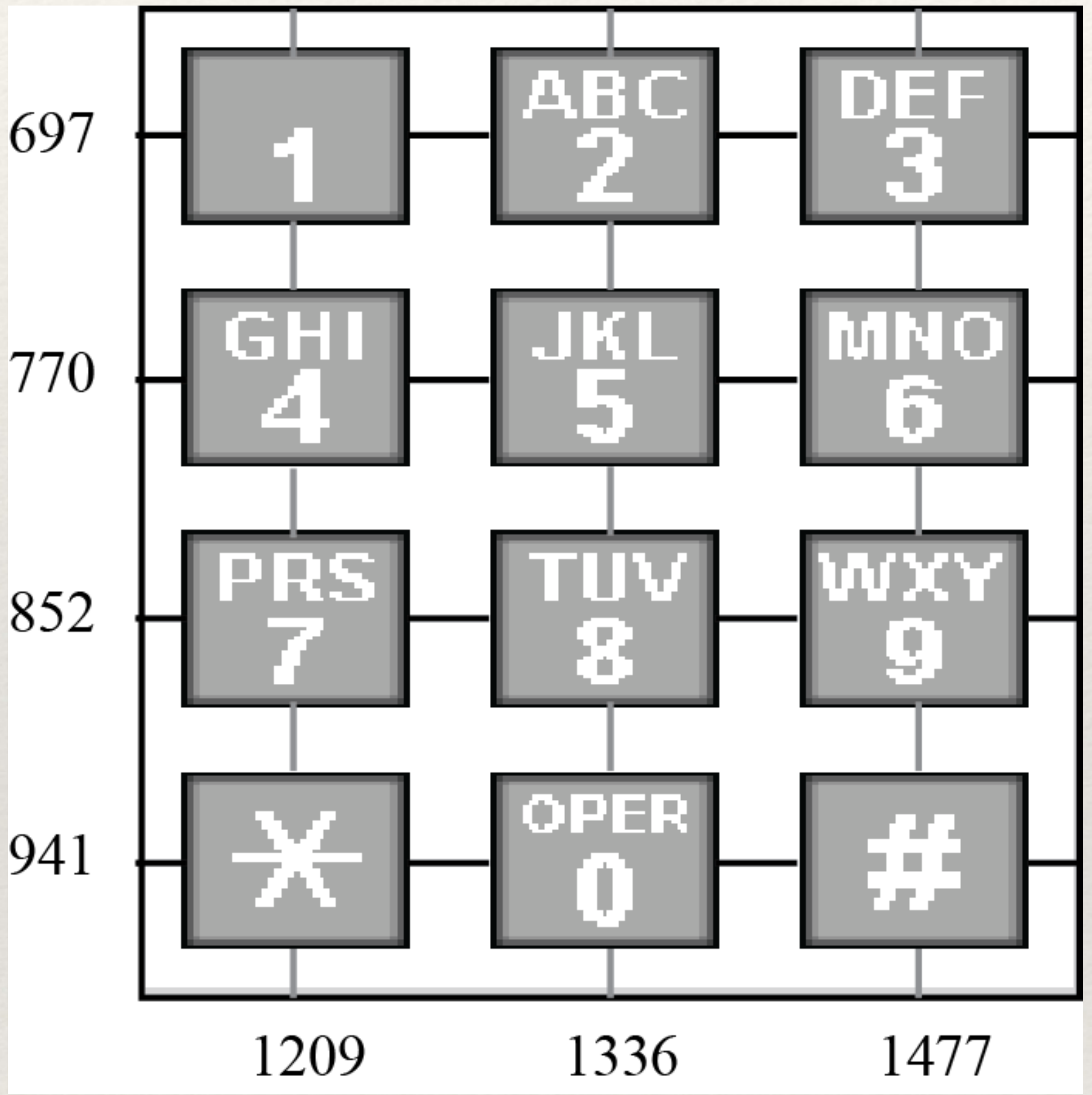
The keypad is organized over **four rows and three columns**.

Each row and each column is assigned a "**frequency**" (we will see later what this means).

A number is then defined by its row and column.

For example, the number 6 is defined by (770, 1477)

Dialing a number

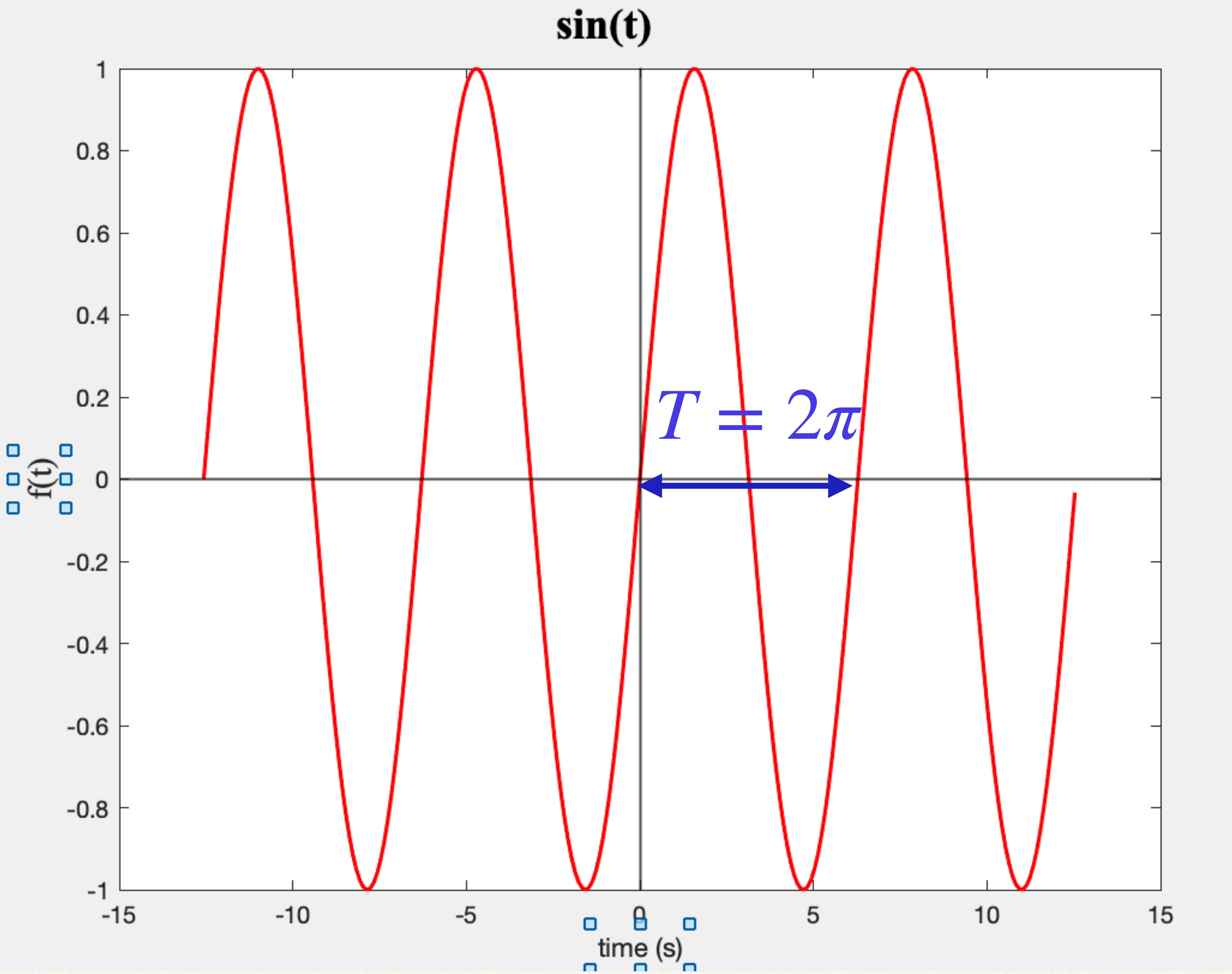


The phone generates **a digit as a sound**.

A **“sound”** is a **wave**, usually a sine function.

The phone generates a digit as the **sum of the sine waves** whose **frequencies** are defined by its **row position** and **column position**.

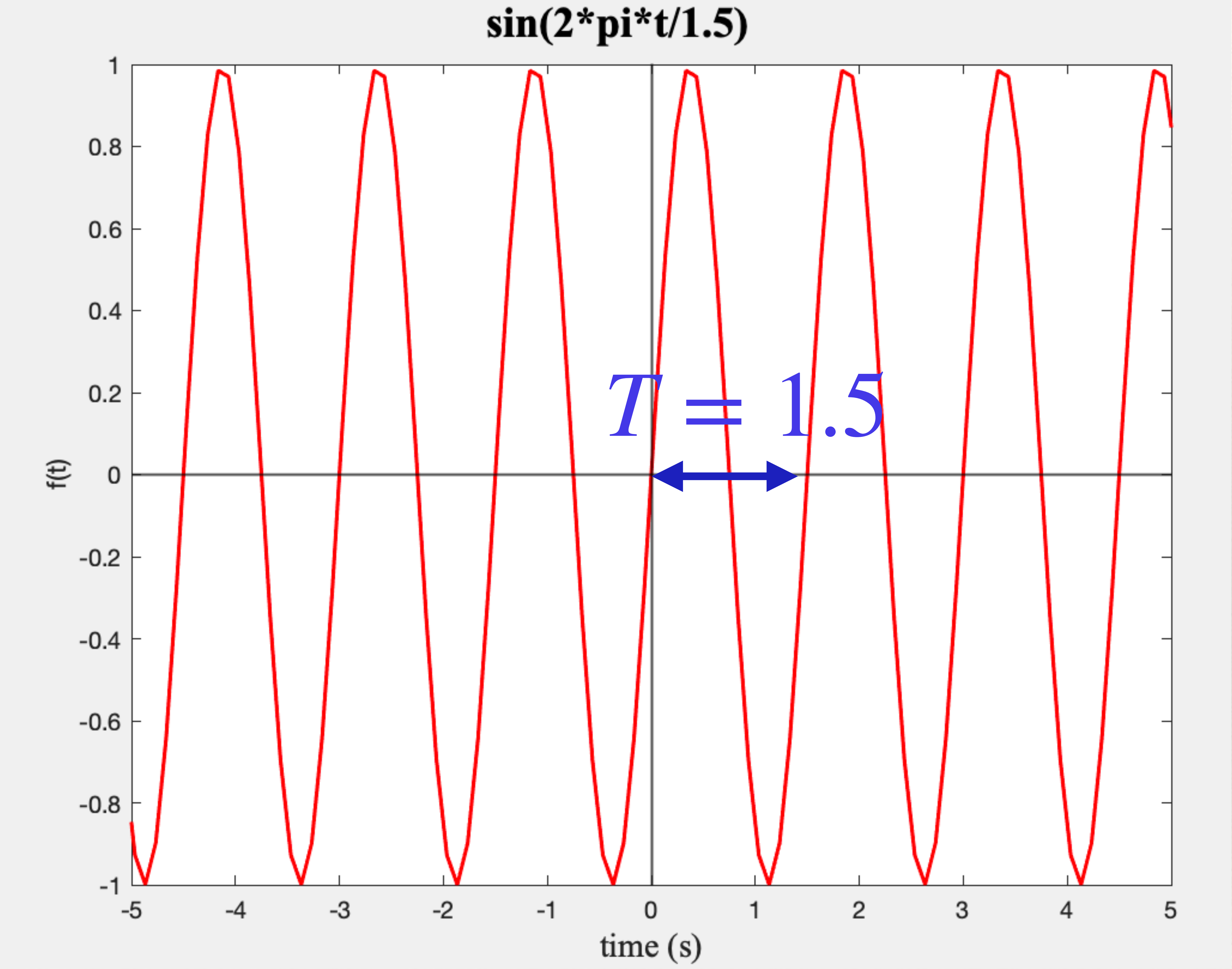
Reminder on sine functions



$$y(t) = \sin(t)$$

The sine function is periodic (i.e. it repeats itself), with a period T = 2π

Reminder on sine functions



The sine function can be modulated to change its period.

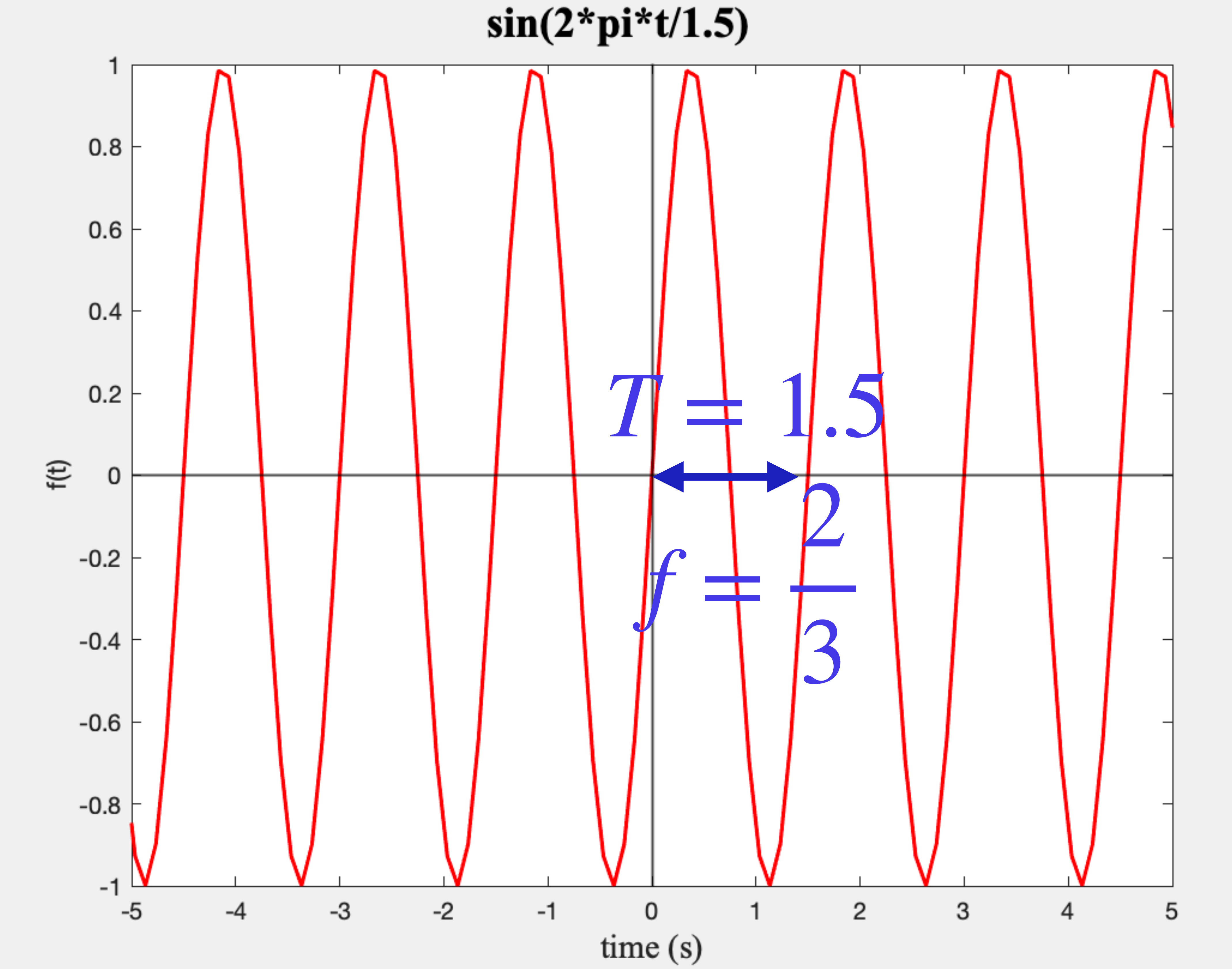
To assign a period T to a sine function,

$$y(t) = \sin\left(\frac{2\pi}{T}t\right)$$

Example (on the right): T=1.5

$$y(t) = \sin\left(\frac{2\pi}{1.5}t\right)$$

Reminder on sine functions



When the x axis represents time, the period T is the duration of the function before it starts repeating itself.

The inverse of the period is the frequency of the signal, f:

$$f = \frac{1}{T}$$

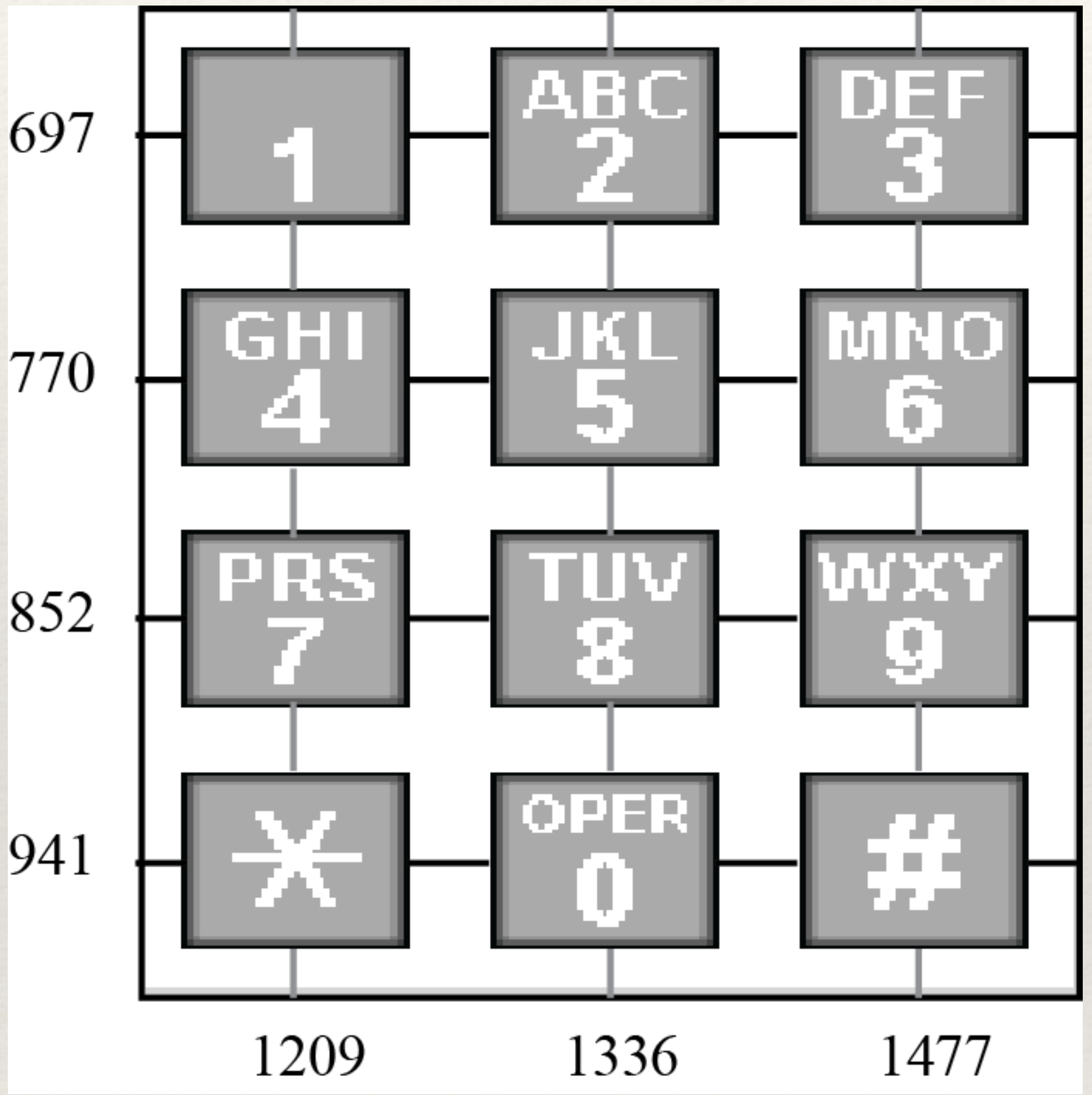
It defines the number of time the function repeats itself over 1 second.

The **frequency** is expressed in **Hertz (Hz)**

The function can then be written in 2 ways:

$$y = \sin\left(\frac{2\pi}{T}t\right) = \sin\left(2\pi f t\right)$$

Dialing a number



The phone generates a digit as the **sum of the sine waves** whose **frequencies** are defined by its **row position** and **column position.**

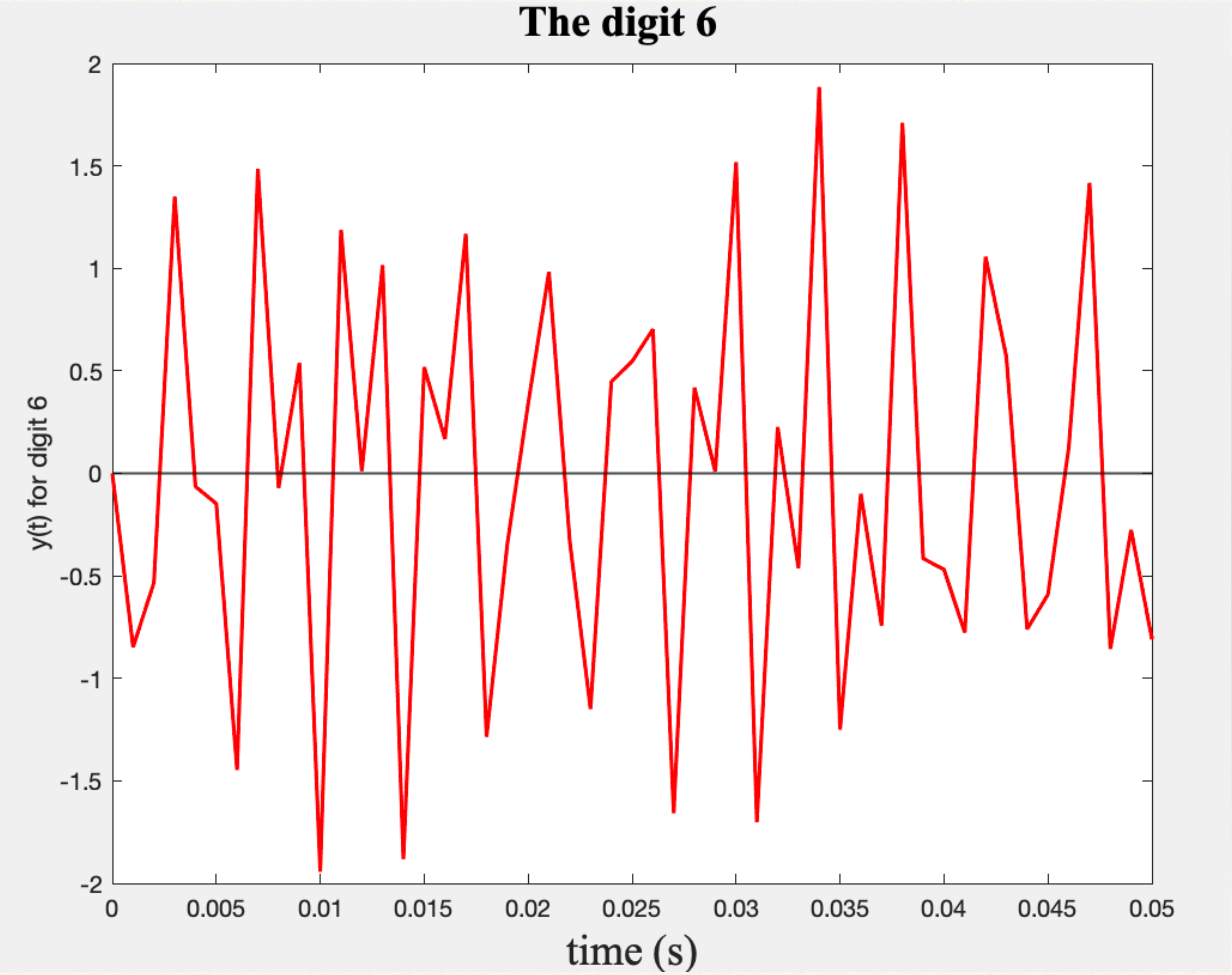
For example, for the digit 6:

$$y(t) = \sin(2\pi 770 t) + \sin(2\pi 1477 t)$$

Dialing a number: discrete signal

the digit 6:

$$y(t) = \sin(2\pi 770t) + \sin(2\pi 1477t)$$

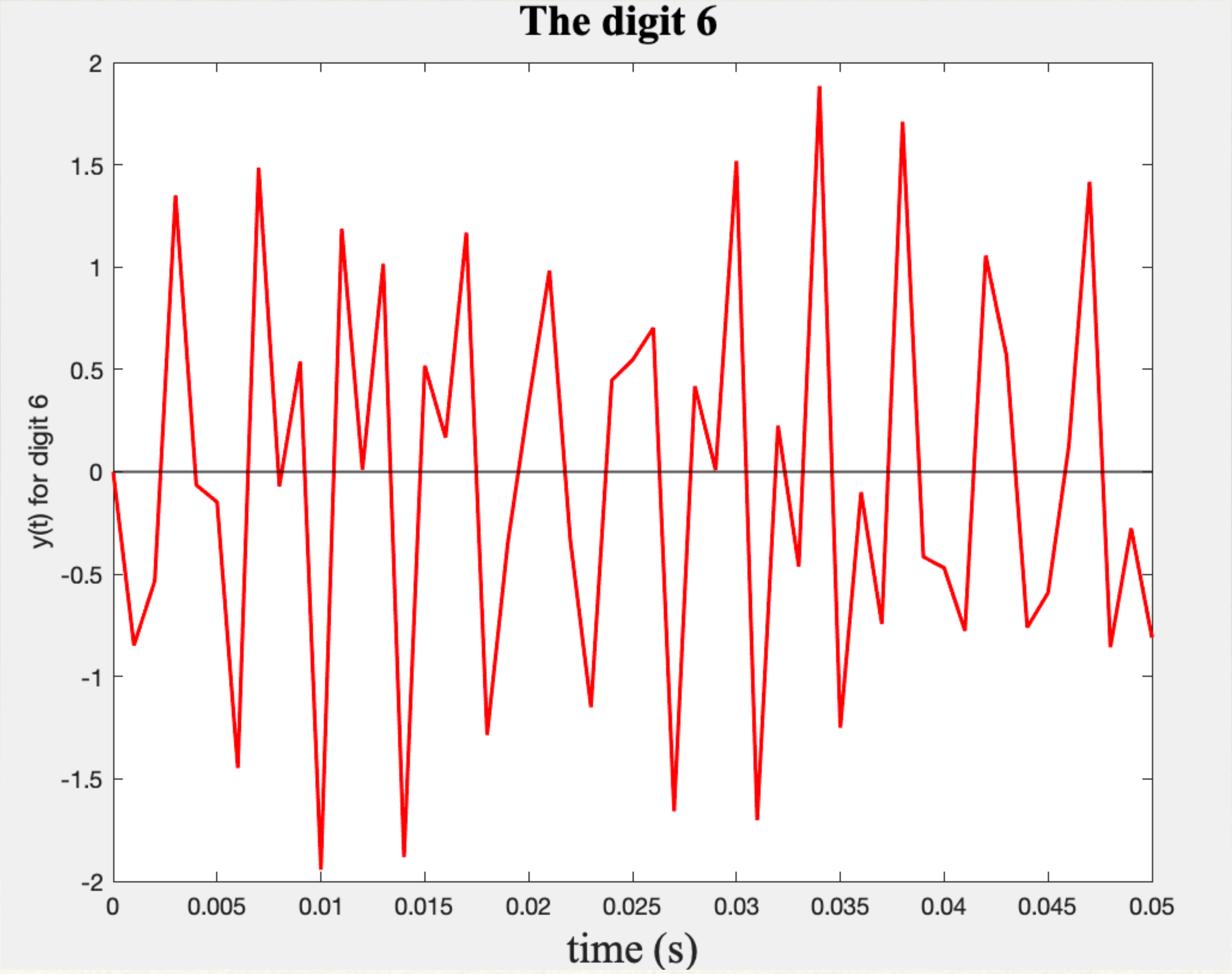


Dialing a number: discrete signal

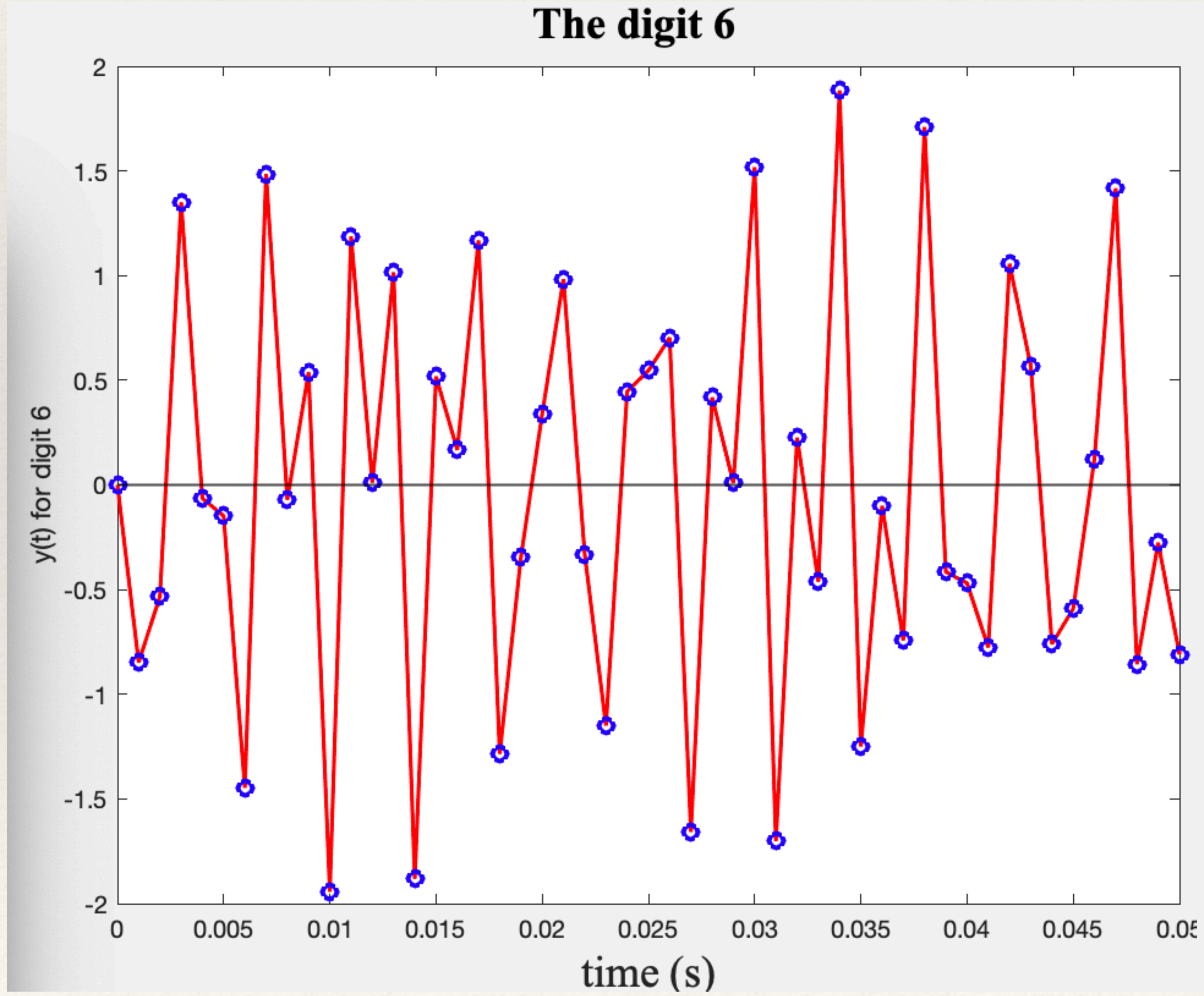
the digit 6:

$$y(t) = \sin(2\pi 770t) + \sin(2\pi 1477t)$$

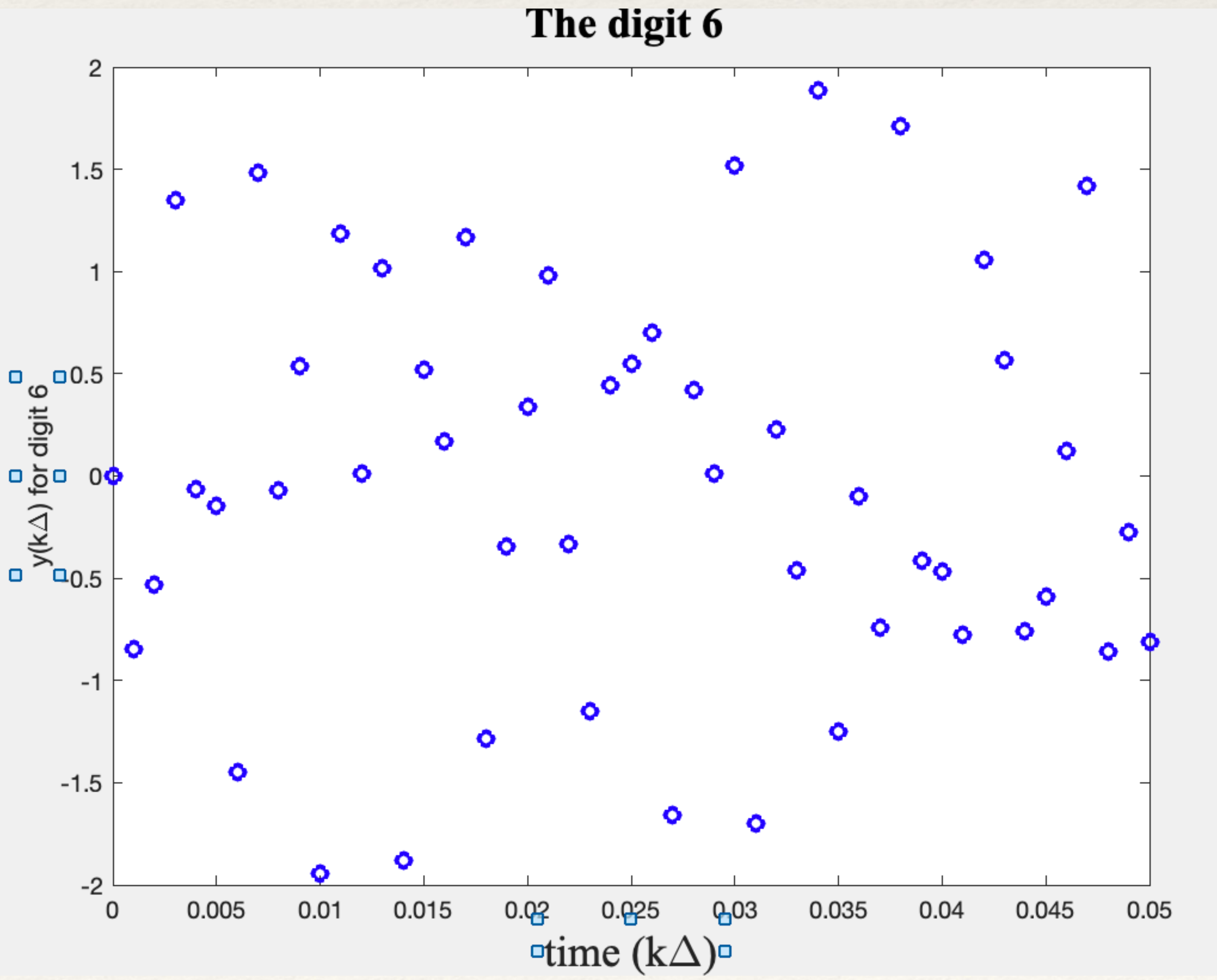
The phone transmits the sound signal "digitally": this means that it does not transmit the continuous signal, but a discrete signal in time, i.e. at only specific times. Those times are spaced uniformly, with a spacing Δ



Continuous signal

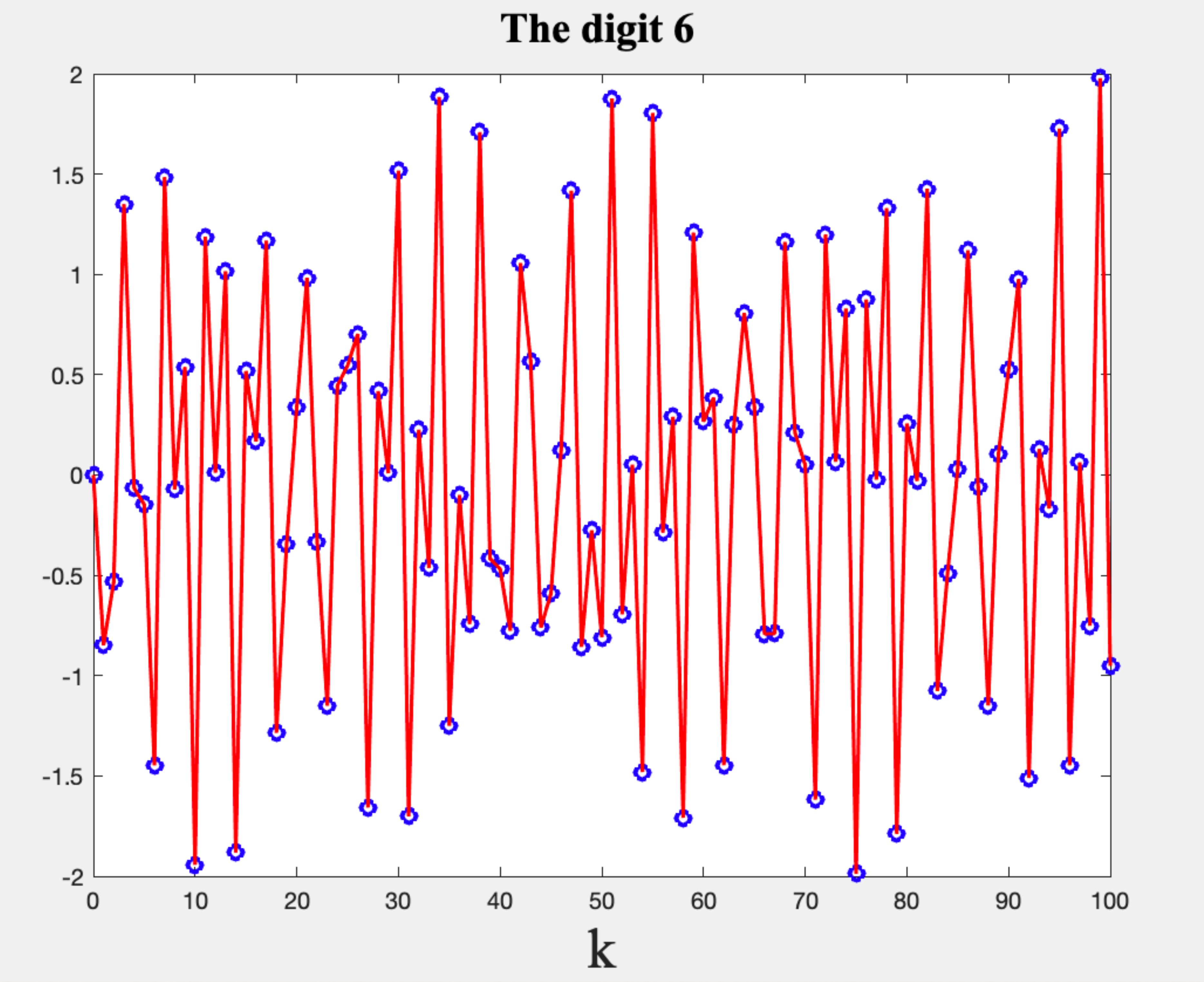


Discretization



Actual signal!

Dialing a number: discrete signal



$$y(k\Delta) = \sin(2\pi 770 k\Delta) + \sin(2\pi 1477 k\Delta)$$

How to choose Δ?

Δ is the **time interval between two sampled points**. It can be characterized by its inverse, Fs,

$$Fs = \frac{1}{\Delta}$$

Fs is the sampling rate: it defines how many points are sampled per second.

Proper discretization requires that **the sampling rate is at least twice the highest possible frequency in the signal**.

For a phone keypad, the highest frequency is 1477.

This means that $Fs > 2 \times 1477$

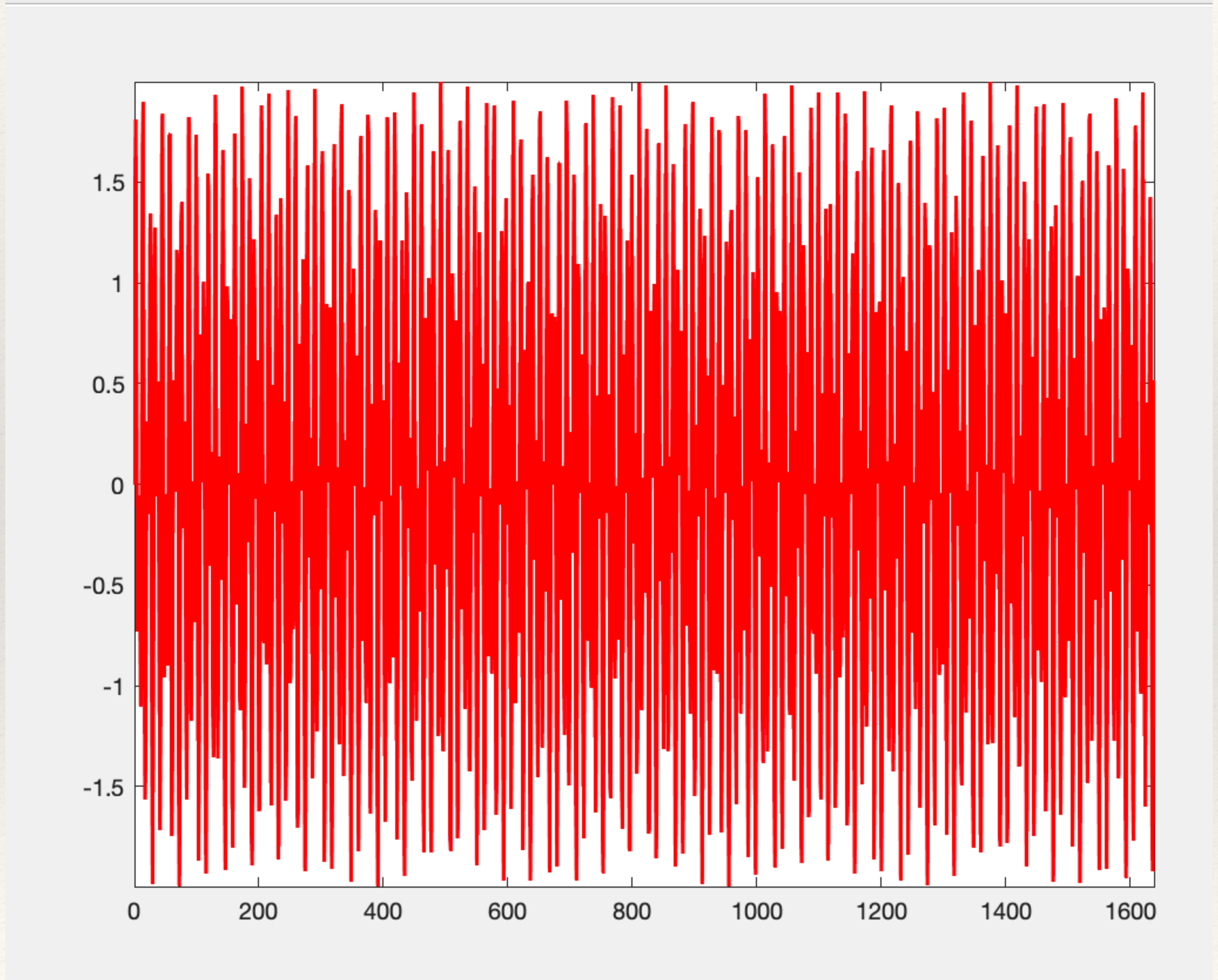
The phone industry has chosen the standard: Fs = 8192

Mimicking dialing a digit on Matlab

Generating the sound for the digit 6 on a phone keypad with Matlab:

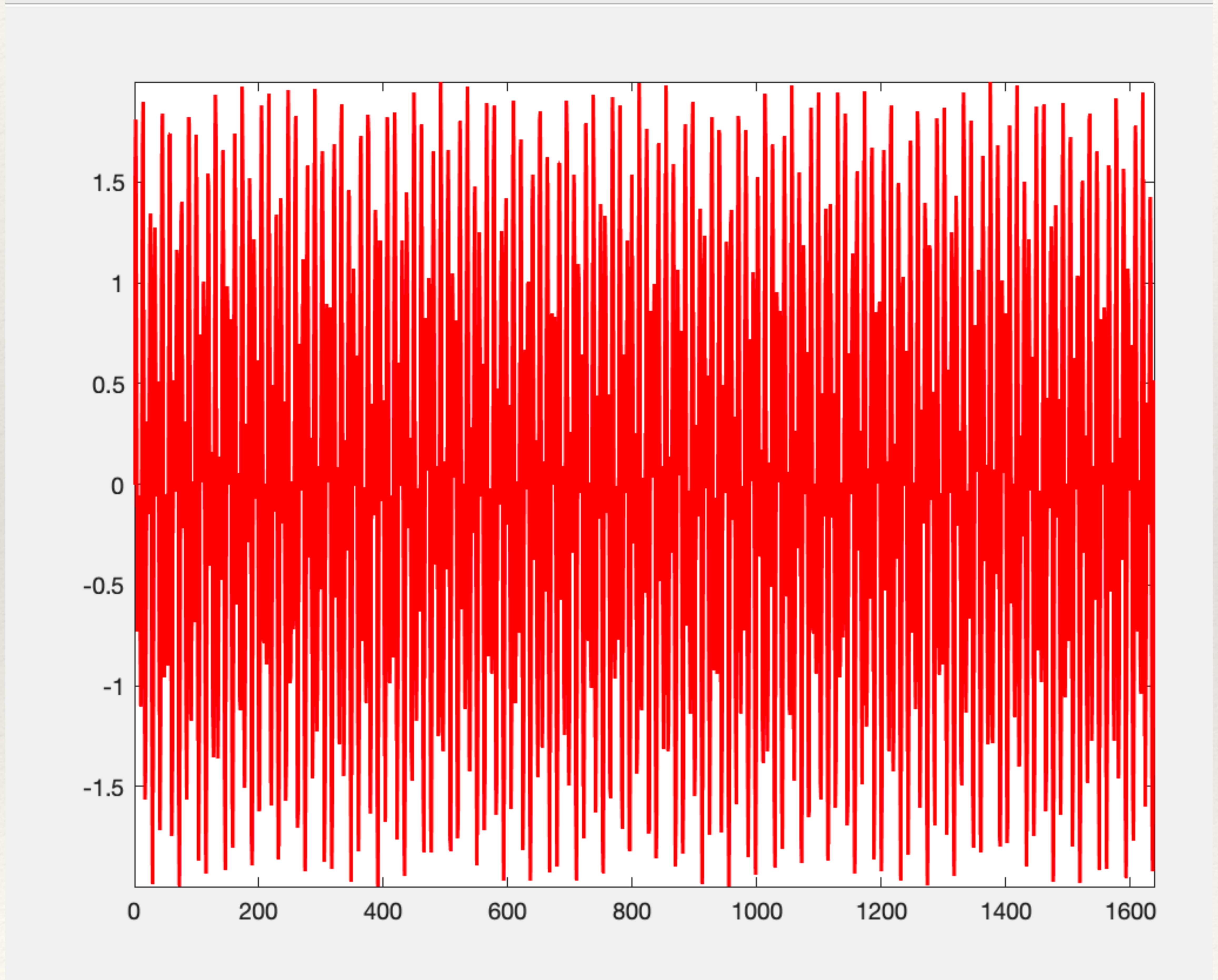
```
% Defining the sampling rate:
Fs = 8192;
% Computing the corresponding Delta in time:
Delta = 1/Fs;
% Defining the time for the signal: 0.2 seconds total, sampled by Delta:
time= 0: Delta : 0.2
% Generating the two signals corresponding to the row and column of 6:
y1  = sin ( 2 * pi * 770 * time);
y2  = sin ( 2 * pi * 1477 * time);
% Combining the two signals to generate the digit 6:
y   = y1 + y2;
% Plot this signal:
plot(time, y, '-r', 'LineWidth', 1.5)
% We can even play the sound:
sound(y)
```

Reverse Engineering:
Find a digit from its sound



Given a time signal (over N points), can we find the corresponding digit that was dialed?

Reverse Engineering:
Find a digit from its sound



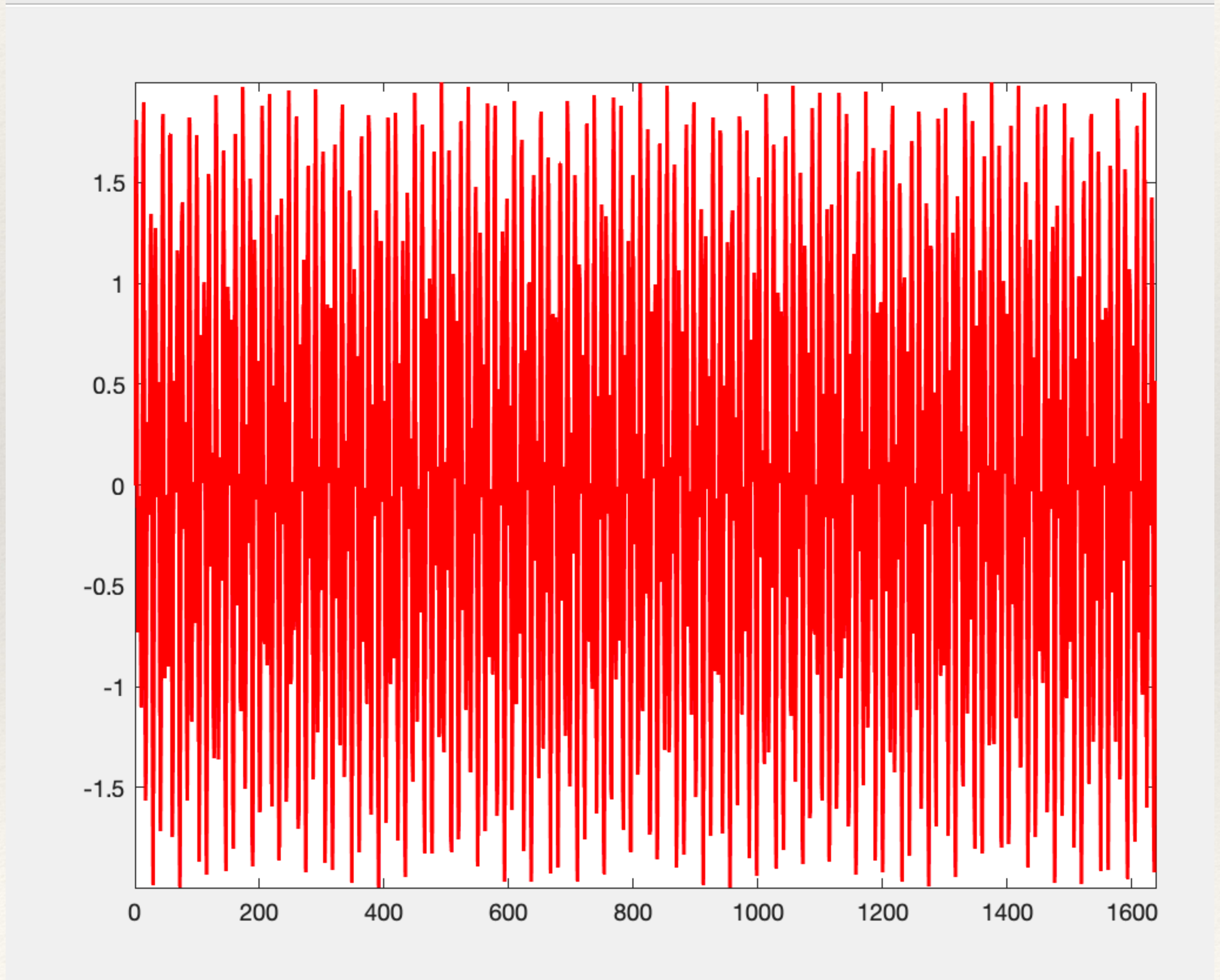
Given a time signal (over N points), can we find the corresponding digit that was dialed?

We know more than the time signal!:

- As this is a signal generated by a phone,
  - Fs = 8192 and
  - $\Delta = \frac{1}{Fs} = \frac{1}{8192}$
- The signal is a combination of two sines with different frequencies

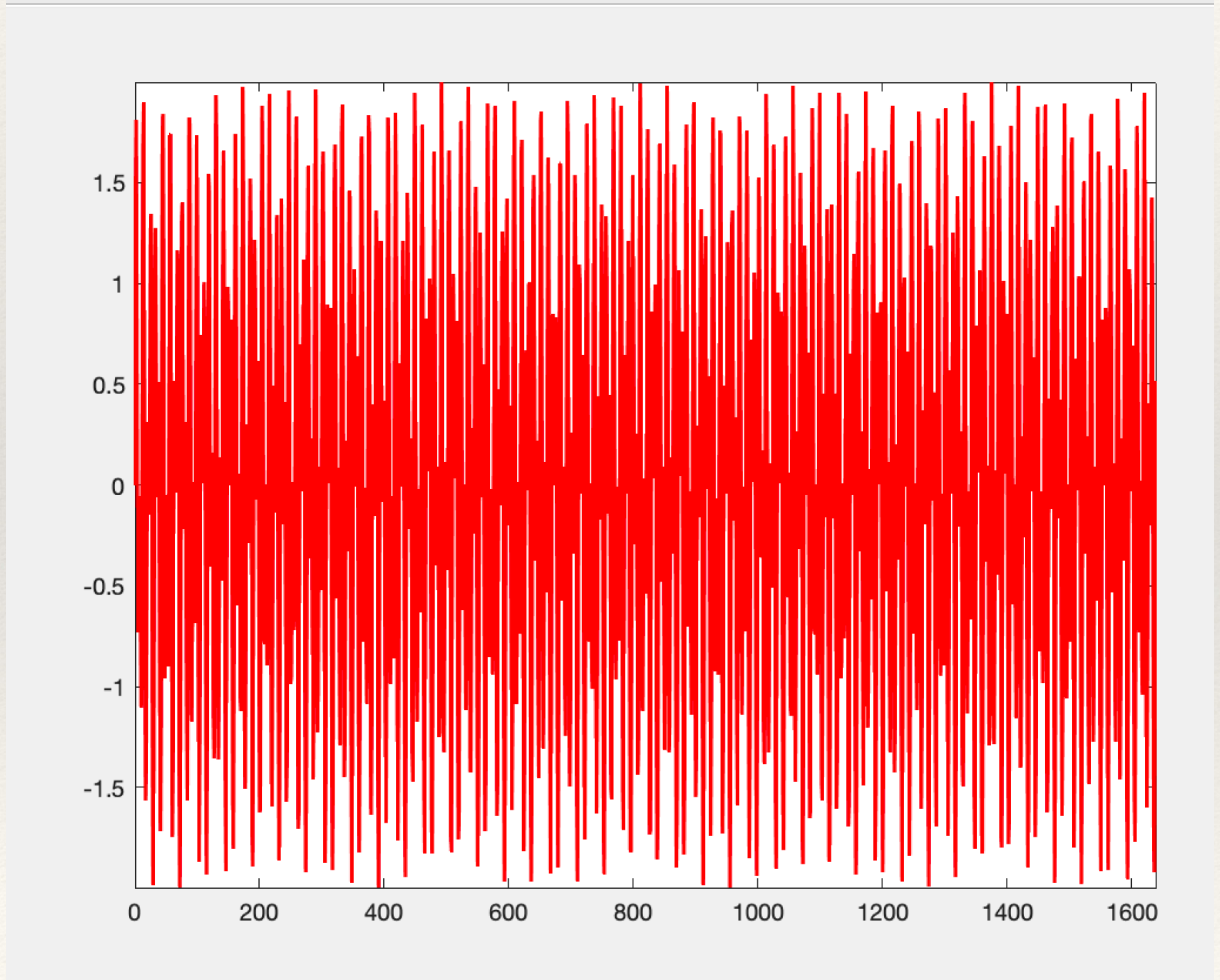
Reverse Engineering:
Find a digit from its sound

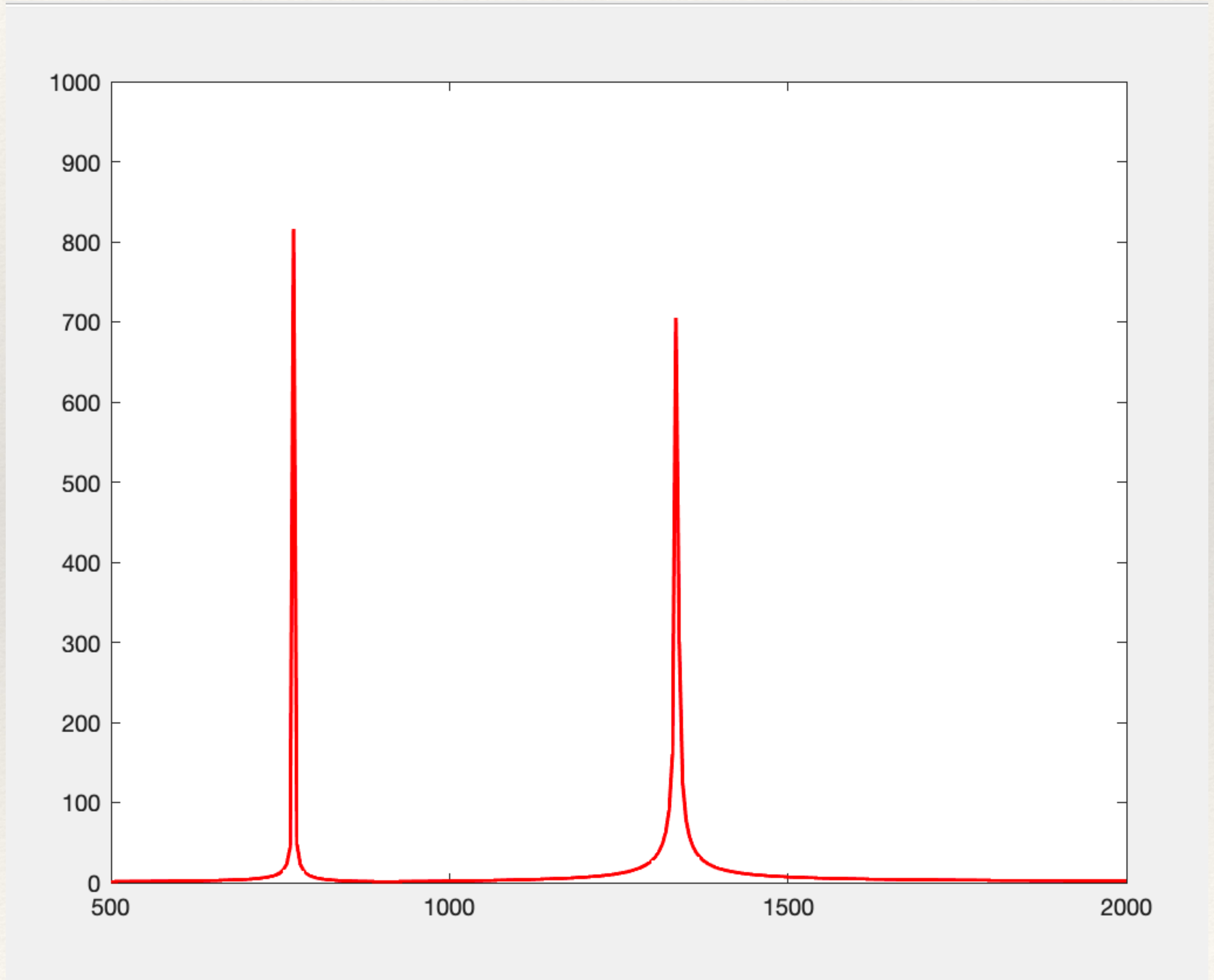
What we want: A tool that:



Reverse Engineering:
Find a digit from its sound

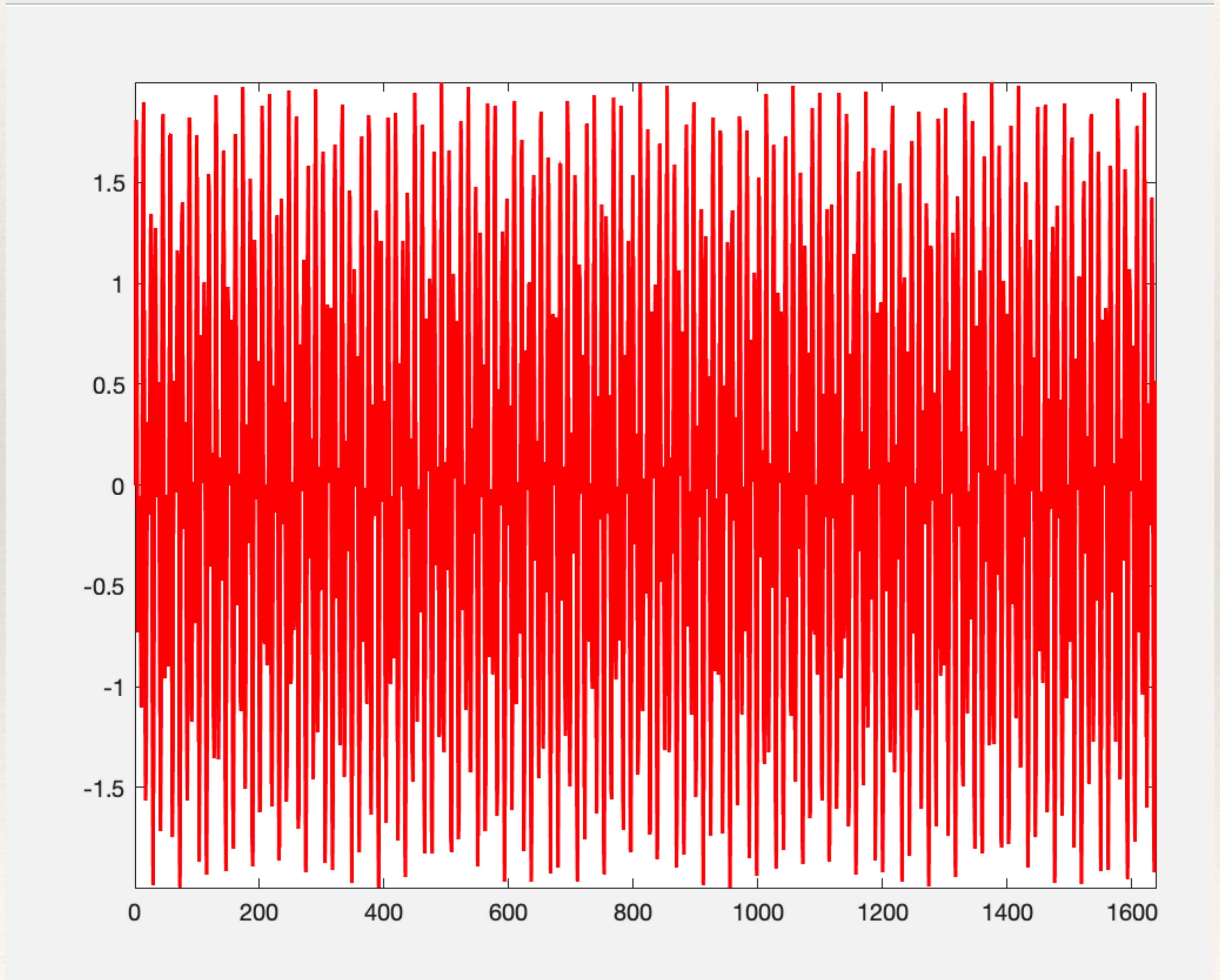
What we want: A tool that:

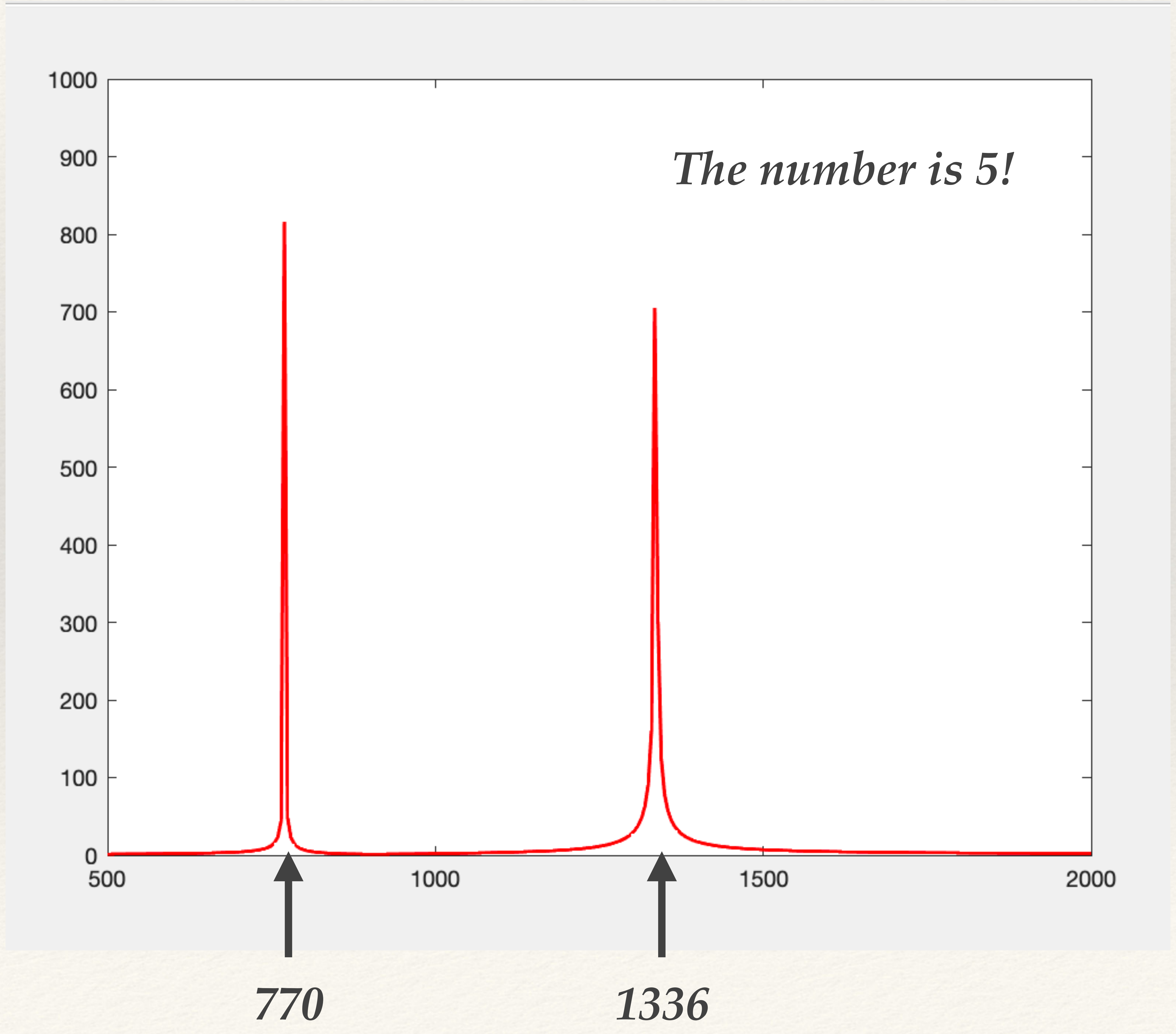




Reverse Engineering:
Find a digit from its sound

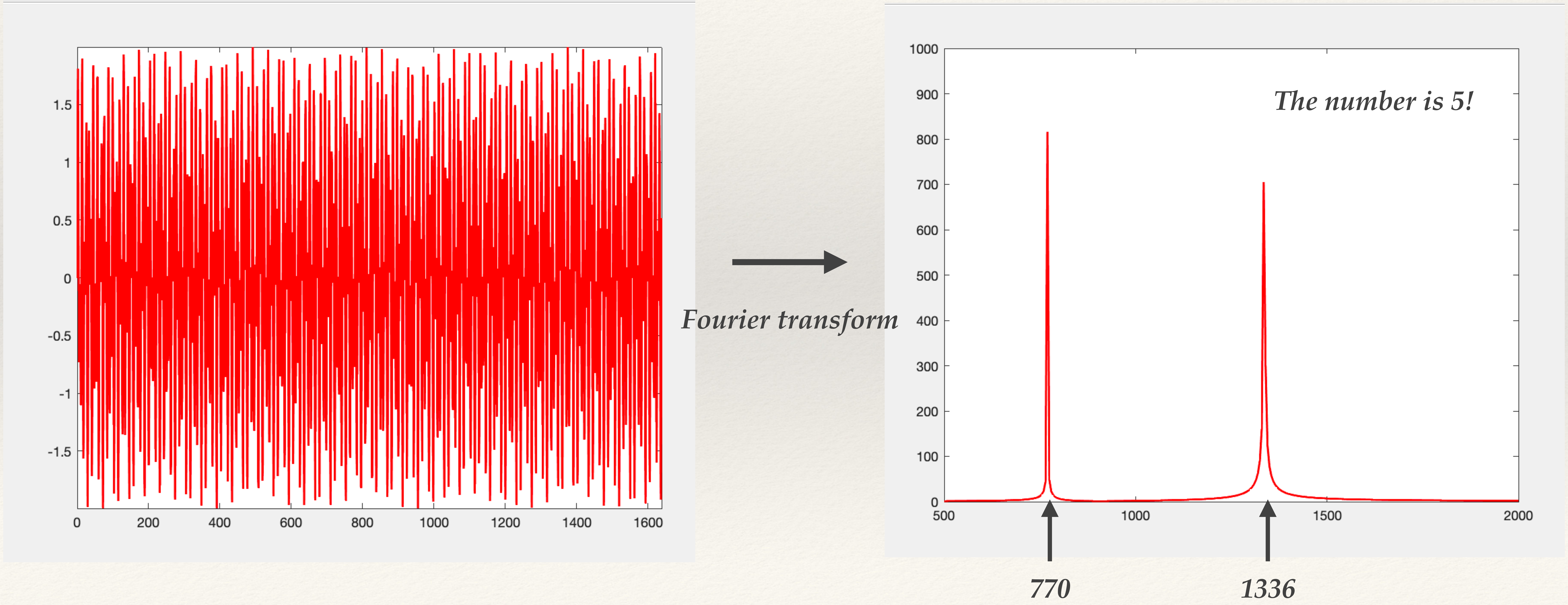
What we want: A tool that:





Reverse Engineering:
Find a digit from its sound

What we want: A tool that:



Basics of Fourier transform

Any periodic function f(t), with period T can be written as a sum of sine and cosine function:

$$f(t) = a_0 + \sum_k a_k \cos\left(\frac{2\pi}{T}kt\right) + b_k \sin\left(\frac{2\pi}{T}kt\right)$$

Defining the fundamental frequency $f_0 = \frac{1}{T}$

$$f(t) = a_0 + \sum_k a_k \cos\left(2\pi k f_0 t\right) + b_k \sin\left(2\pi k f_0 t\right)$$

Sum of cosines and sines with frequencies $f_0, \quad 2f_0, \quad 3f_0, \ldots,$

Basics of Fourier transform

$$f(t) = a_0 + \sum_k a_k \cos\left(2\pi k f_0 t\right) + b_k \sin\left(2\pi k f_0 t\right)$$

The coefficients a_k and b_k can be computed:

$$a_k = \frac{1}{T}\int_{-T/2}^{T/2} f(t)\cos(2\pi k f_0 t)dt$$

$$b_k = \frac{1}{T}\int_{-T/2}^{T/2} f(t)\sin(2\pi k f_0 t)dt$$

Basics of Fourier transform

$$f(t) = a_0 + \sum_k a_k \cos\left(2\pi k f_0 t\right) + b_k \sin\left(2\pi k f_0 t\right)$$

The coefficients a_k and b_k can be computed:

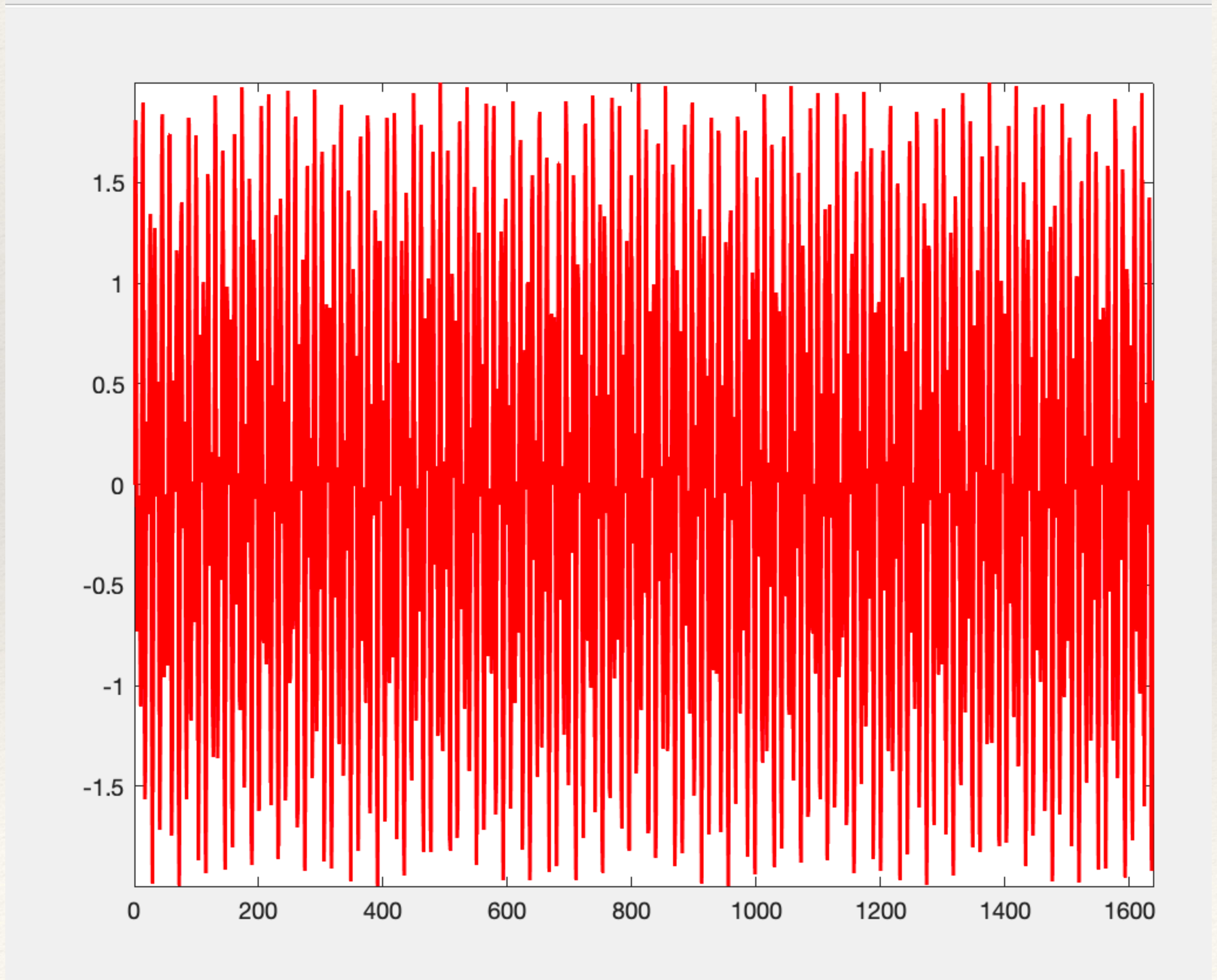
$$a_k = \frac{1}{T}\int_{-T/2}^{T/2} f(t)\cos(2\pi k f_0 t)dt$$

$$b_k = \frac{1}{T}\int_{-T/2}^{T/2} f(t)\sin(2\pi k f_0 t)dt$$

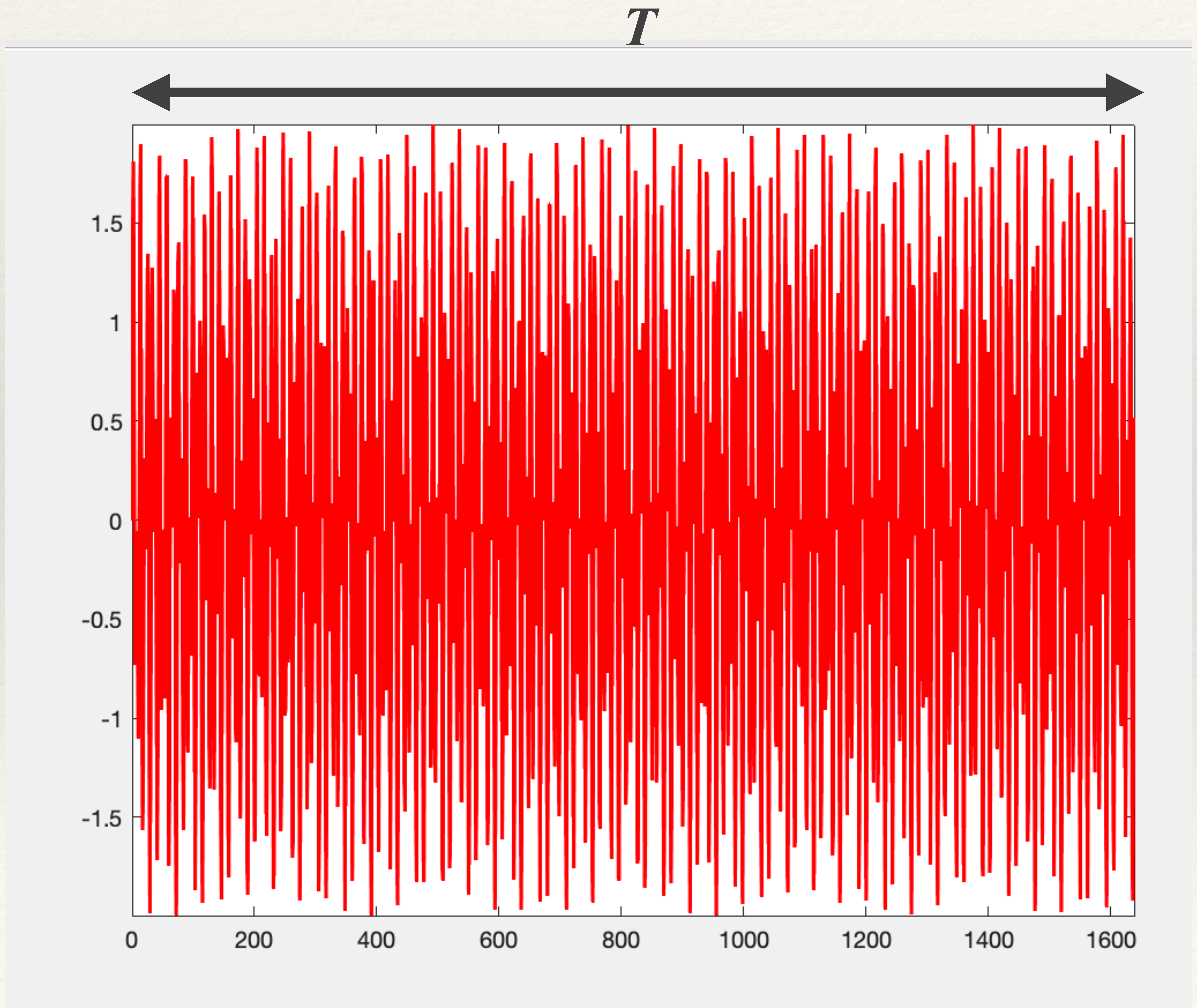
This is exactly what the Fourier transform computes!

It finds how much the different frequencies kf_0 intervene in the signal!

Reverse Engineering:
Find a digit from its sound



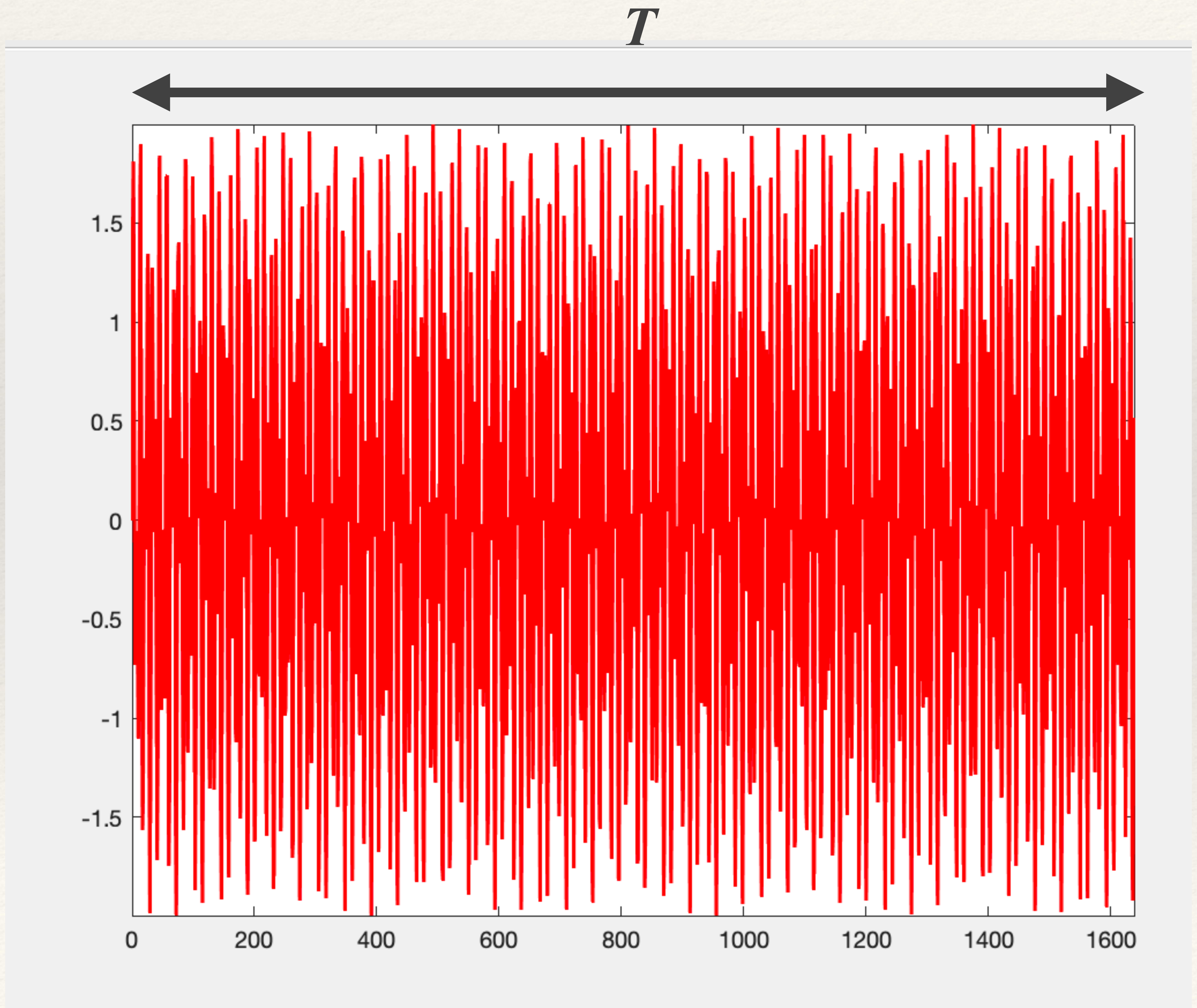
Reverse Engineering:
Find a digit from its sound



1) The "period" T of the signal is defined as its total length:

$$T = N\Delta = \frac{N}{Fs}$$

Reverse Engineering:
Find a digit from its sound



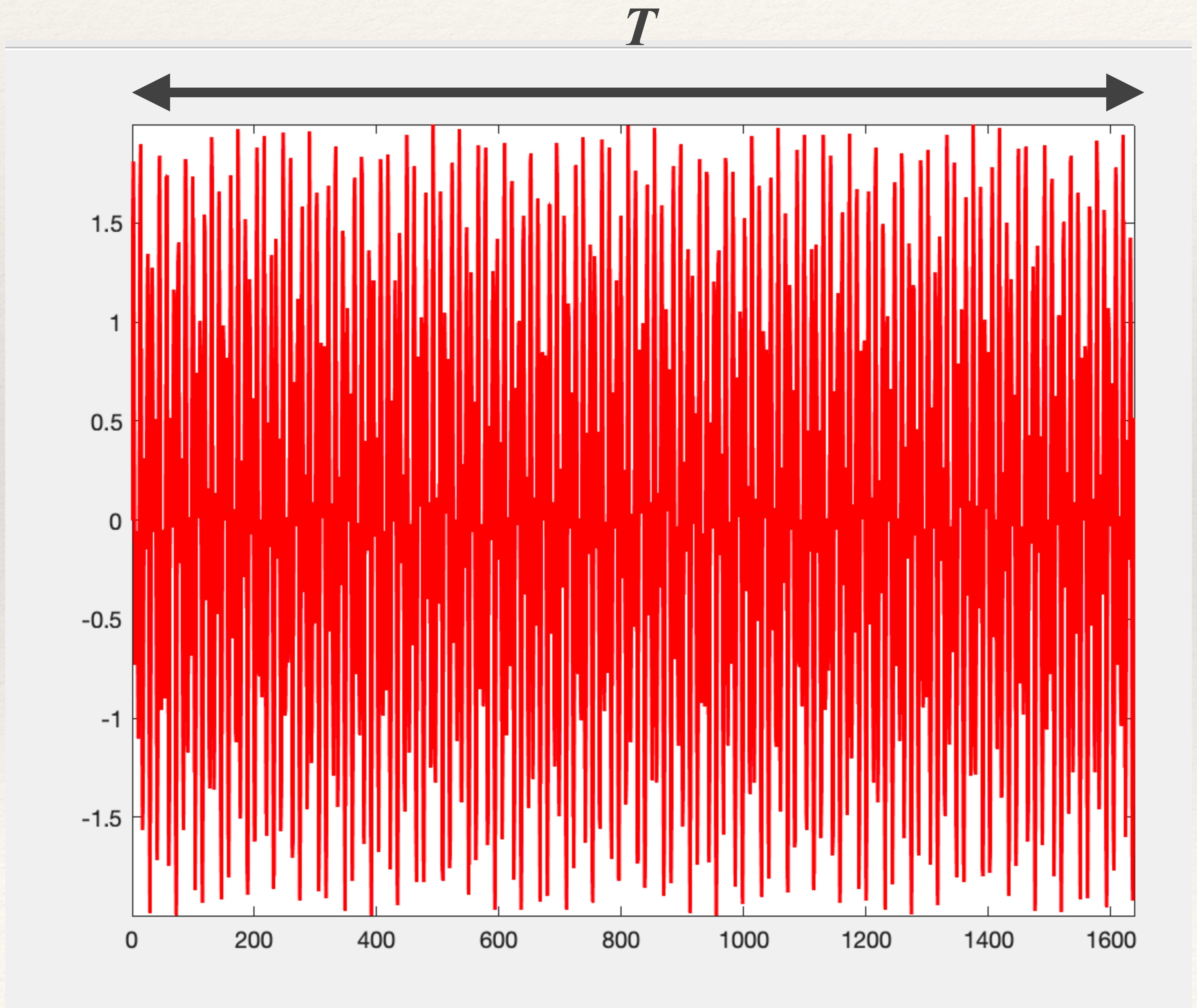
1) The “period” T of the signal is defined as its total length:

$$T = N\Delta = \frac{N}{Fs}$$

2) The “fundamental frequency” f_0 is the inverse of the “period”:

$$f_0 = \frac{1}{T} = \frac{F_s}{N}$$

Reverse Engineering:
Find a digit from its sound



1) The "period" T of the signal is defined as its total length:

$$T = N\Delta = \frac{N}{Fs}$$

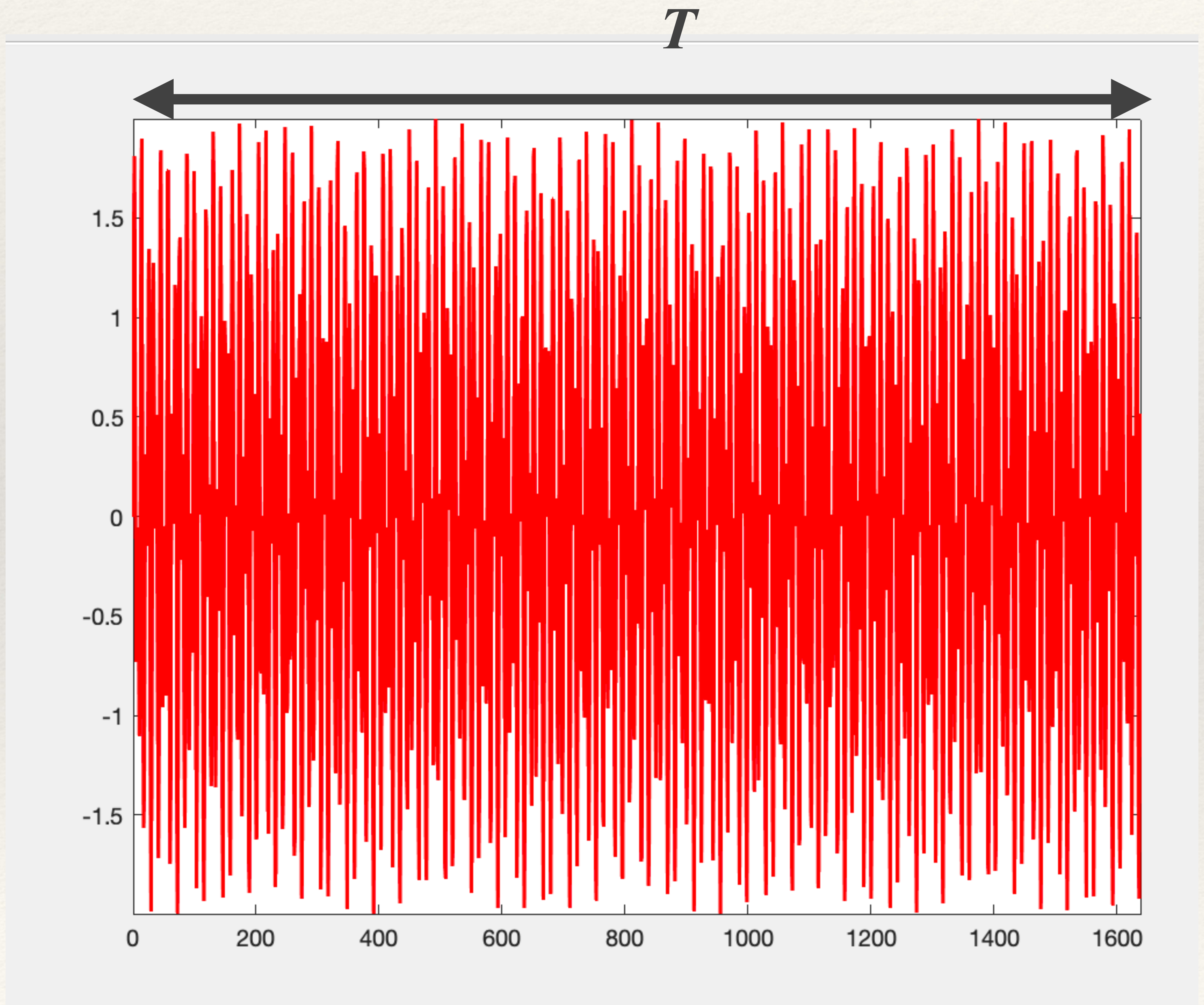
2) The "fundamental frequency" f_0 is the inverse of the "period":

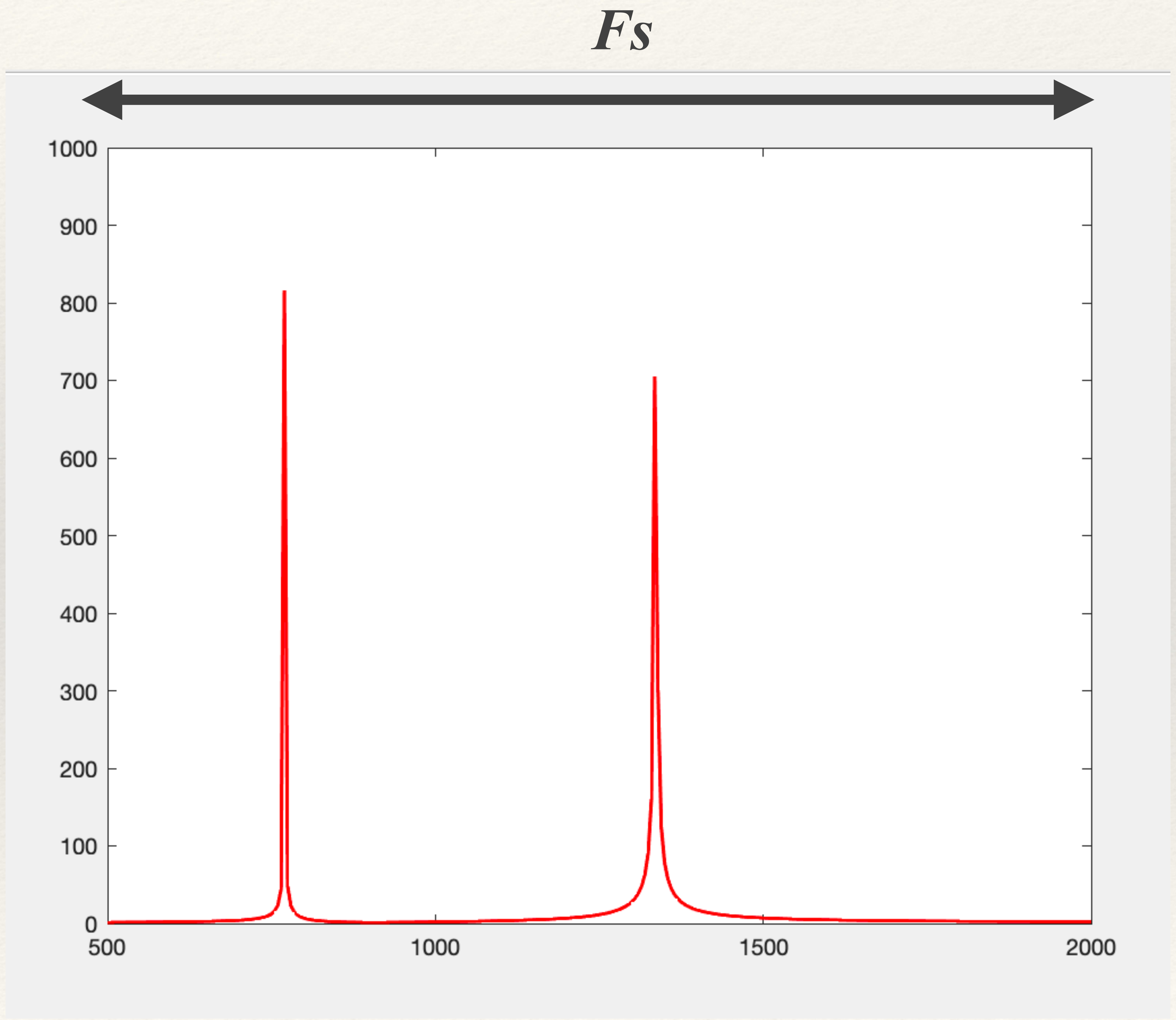
$$f_0 = \frac{1}{T} = \frac{F_s}{N}$$

3) Compute the coefficients a_k and b_k for frequency (kf_o) using Fourier transform.

If there are N values in the time signal, the Fourier transform will compute N values.

Reverse Engineering:
Find a digit from its sound





Matlab script to analyse a signal y

```
% Defining the phone company sampling rate:
Fs = 8192;
Delta = 1/Fs;
% Length of time signal y(t) and corresponding “period’ T:
nval = length(y);
T = Delta*nval;
% Define the fundamental frequency identified by Fourier: f_0 = 1/T
f0 = 1/T;
% The Fourier is computed over a range of frequencies (0, f_0, 2*f_0…)
% over nval values; define this list:
freq= 0: f0 : Fs-1;
% Compute Fourier transform; get abs to combine a_k and b_k
f= abs(fft(y);
% Plot
plot(freq, f, ‘-r’, ‘LineWidth’, 1.5);
% Limit to frequency range of phone dials
xlim([ 500 1500]);
```

Additional code to find peaks automatically

```
%
% Find peak position
%
[peak_amp peak_loc]=findpeaks(f,'Minpeakheight',200);
%
% peak_loc is in point; convert to frequency
%
freq=f(peak_loc);
%
% Only keep values in DTMF range
%
peak_freq=freq(freq < 1700);
peak_height = peak_amp(freq < 1700);
%
hold on
str=num2str(int32(peak_freq));
text(peak_freq(1),peak_height(1),str(1));
text(peak_freq(2),peak_height(2),str(2));
```