

ECG Simulator Based on a Neural Network Trained With Real Patient Data

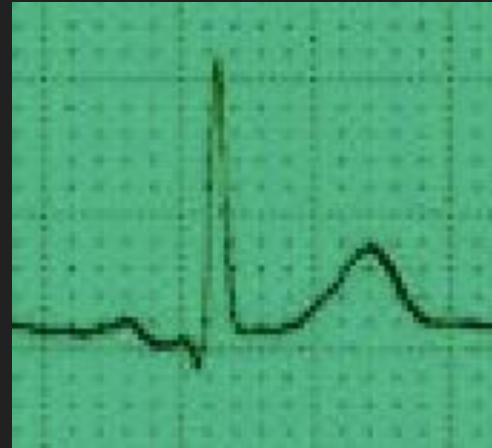
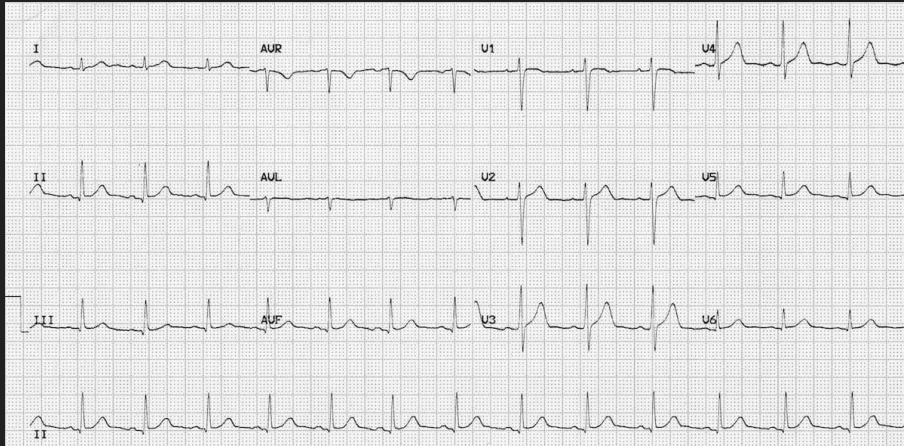
By

Raul Verduzco | Salvatore Gutierrez | Abisai Diaz

Data

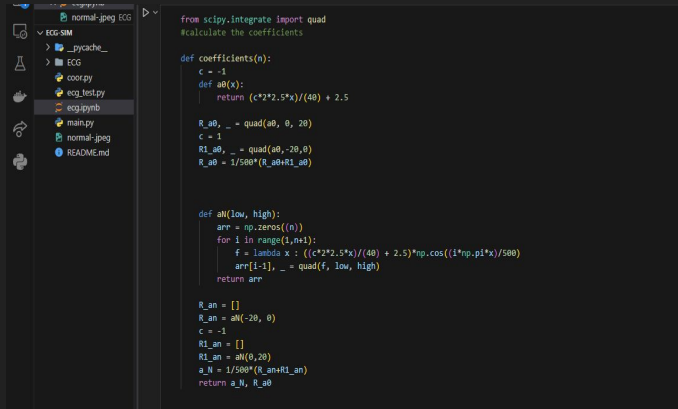
- Created Dataset

- Ex: normal sinus rhythm, bradycardia, tachycardia, Vfib ...
- Images with all six leads
- P waves, QRS wave, and T waves



Tools

- Python with Jupyter Notebook or Visual Studio Code
- Github
- Numpy library for math calculations
 - `np.cos` (cosine wave), `np.quad` (integrate), etc.

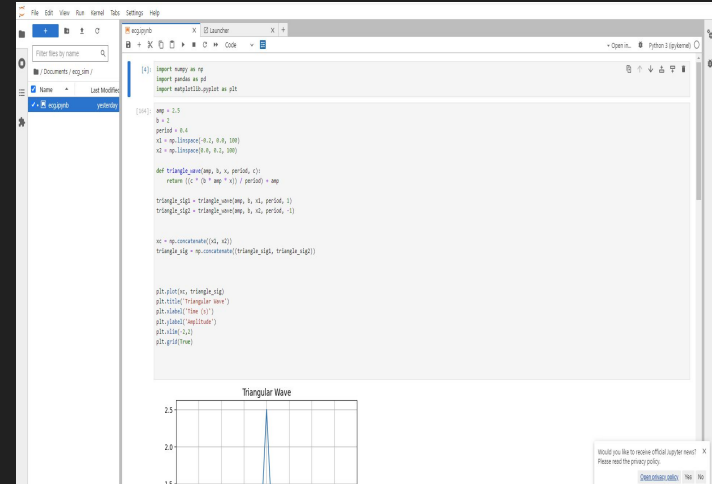


```
from scipy.integrate import quad
#calculate the coefficients

def coefficients(n):
    c = -1
    def a0(x):
        return (c**2*.5*x)/(40) + 2.5
    R_a0, _ = quad(a0, 0, 20)
    c = 1
    R1_a0, _ = quad(a0, -20, 0)
    R_a0 = 1/500*(R_a0+R1_a0)

def aN(low, high):
    arr = np.zeros(n)
    for i in range(1,n+1):
        f = lambda x : ((c**2*.5*x)/(40) + 2.5)*np.cos((1*np.pi*x)/500)
        arr[i-1], _ = quad(f, low, high)
    return arr

R_an = []
R_an = aN(-20, 0)
c = -1
R1_an = []
R1_an = aN(0,20)
a_N = 1/500*(R_an+R1_an)
return a_N, R_a0
```



Other Tools

- Math

- Basic Understanding of calculus concepts like integrating
- Other

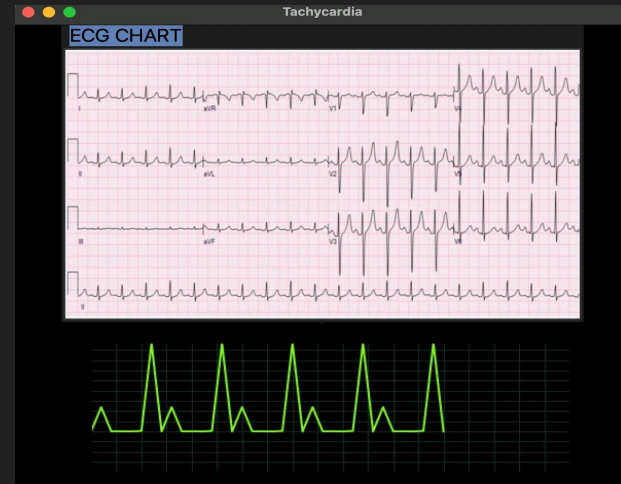
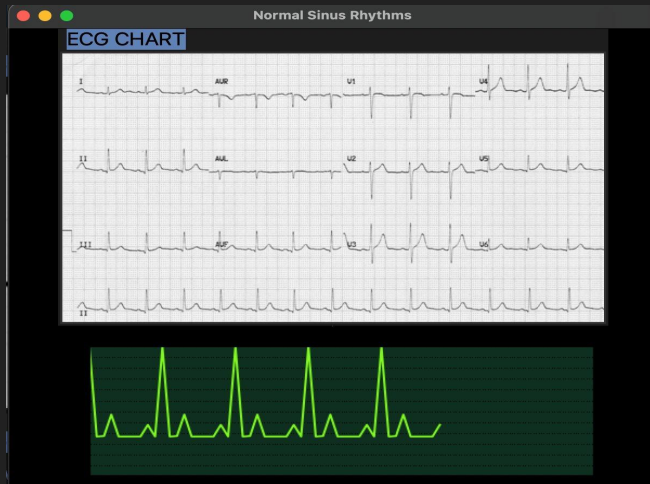
$$f(x) = \begin{cases} \left(\frac{-bax}{l} + a \right) & \text{if } (0 < x < \frac{l}{b}) \\ \left(\frac{bax}{l} + a \right) & \text{if } (-\frac{l}{b} < x < 0) \end{cases}$$

$$\begin{aligned} A_0 &= \frac{1}{P} \int_P s(x) dx \\ A_n &= \frac{2}{P} \int_P s(x) \cos\left(2\pi \frac{n}{P} x\right) dx \quad \text{for } n \geq 1 \\ B_n &= \frac{2}{P} \int_P s(x) \sin\left(2\pi \frac{n}{P} x\right) dx, \quad \text{for } n \geq 1 \end{aligned}$$

$$s_N(x) = A_0 + \sum_{n=1}^N \left(A_n \cos\left(2\pi \frac{n}{P} x\right) + B_n \sin\left(2\pi \frac{n}{P} x\right) \right)$$

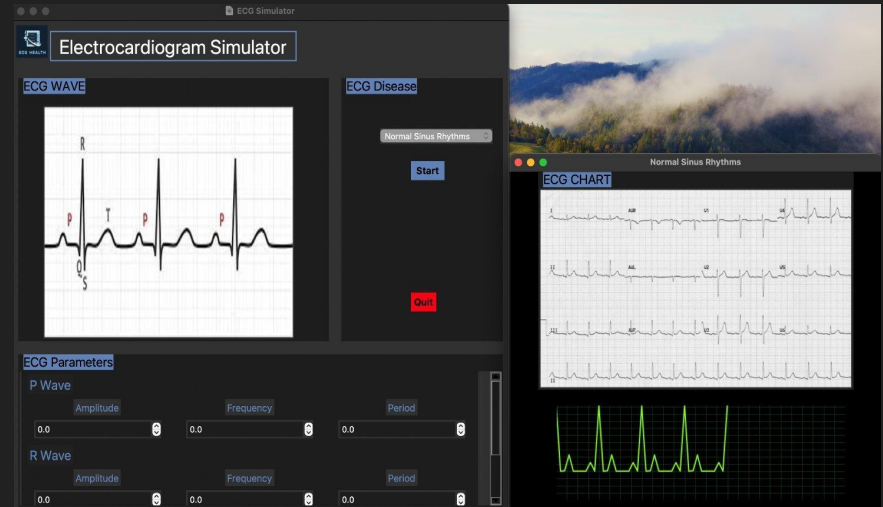
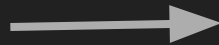
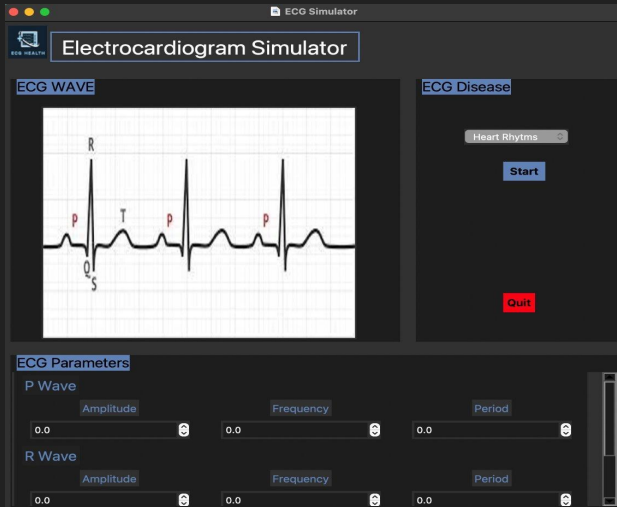
ECG rhythms

- Last week- Normal ECG
- Added: sinus arrhythmia, sinus bradycardia, sinus tachycardia, Vfib, Afib, and Unstable Angina



GUI

- Updated the GUI
 - Allow user to select ecg rhythms
 - Displays ECG chart and the simulation of the live ecg wave
 - Working on allowing the user to alter the wave based on parameter (amplitude, frequency, and period)



ECG Simulator Based on a Neural Network Trained With Real Patient Data

Chengwei Lei¹, Salvatore Gutierrez², Raul Verduzco Guillen³, Abisai Diaz⁴



¹Kern Medical Simulation Center

²Department of Computer and Electrical Engineering and Computer Science, CSU-Bakersfield

Research Background

Heart disease is a prominent issue in Kern County. Kern County has the fourth-highest number of deaths caused by heart disease. Nurses in Kern Medical Center train to learn and recognize patterns that may indicate common heart diseases. However, they still use pre-printed ECG results in their training. This may have the negative result of having them memorize the individual result instead of learning to recognize the pattern.

This research is sectored into two parts with the goal of providing the nurses with a better method of learning different ECG patterns. The first is to create an ECG simulator that generates the patterns nurses need to learn. The second is to use machine learning and previous patient data to add variance similar to the variance in real patients. This will allow the nurses to train using more realistic data.

Methodology

ECG waves are periodic signals composed of a P wave, QRS complex, and T wave. Fourier analysis is a mathematical tool used to decompose these complex signals into simpler sinusoidal components. Fourier analysis allows us to generate accurate representations of the original waveforms through sinusoidal summation.

Fourier Series

$$s_N(x) = A_0 + \sum_{n=1}^N \left(A_n \cos\left(2\pi\frac{n}{P}x\right) + B_n \sin\left(2\pi\frac{n}{P}x\right) \right)$$

Q,R,S Wave

$$f(x) = \begin{cases} \left(\frac{-bx}{t} + a \right) & \text{if } (0 < x < \frac{t}{2}) \\ \left(\frac{bx}{t} + a \right) & \text{if } (-\frac{t}{2} < x < 0) \end{cases}$$

P,U WAVE

$$f(t) = a(t-c)^4 + A$$

ECG Waves

Typical Normal Lead II ECG compared to our simulated ECG



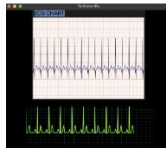
The QRS waves are 4ms spaced out making the heart rate normal

ECG for Sinus Bradycardia against the simulated Bradycardia



Bradycardia is a common rhythm that shows the R waves wider apart

Sinus Tachycardia



Tachycardia is known to have a faster heartbeat which is shown in both the chart and simulation

Preliminary Results

GUI:

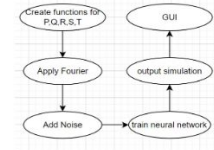


Parameters:
Controls inputs for independent waves

ECG simulator window

In-progress Machine Learning

Our next step involves adding noise to the ECG signals to enhance realism. Three main types of noise are typically encountered: random noise, baseline wander, and powerline interference. By incorporating these types of noise into our simulator, we aim to create more authentic ECG waveforms.



Conclusion and Future work

Heart disease can be a matter of life and death, so it is be adequately prepared before they begin working with real patients. Providing Kern Medical nurses with more realistic data can help them get accustomed to the fuzzier details before they work with real patients. The updated ECG simulator would help doctors and nurses by providing them with more realistic ECG for their training exercises.

Simulation of ECG

The image shows a screenshot of an "ECG Simulator" application window. The window title is "ECG Simulator" and it features a logo for "ECG HEALTH". The main interface is divided into several sections:

- ECG WAVE:** A central plot area displaying a simulated ECG waveform on a grid. The waveform shows a regular rhythm with labeled waves: P, Q, R, S, T, and another P. The R wave is the tallest, followed by the S wave, and then the T wave.
- ECG Disease:** A control panel on the right side. It includes a dropdown menu labeled "Heart Rhythms", a blue "Start" button, and a red "Quit" button.
- ECG Parameters:** A section at the bottom with two groups of sliders. The first group is for the "P Wave" and the second is for the "R Wave". Each group has three sliders labeled "Amplitude", "Frequency", and "Period", all currently set to "0.0".

The application window is overlaid on a scenic background image of a valley with rolling green hills, a winding dirt road, and a small pond, with mist or clouds hanging over the distant mountains.

