To check if the comb frequency is mixing down to the brainwave region through both nonlinear and linear effects, you can use the following mathematical approaches:

1. **Beat Frequency Calculation (Linear Effects)**

- **Formula**: $f_{\text{beat}} = |f_1 - f_2|$
- **Application**: Calculate the difference between pairs of comb teeth frequencies to see if the result falls within the brainwave frequency range (0.5 Hz to 40 Hz).

2. **Heterodyne Frequency Calculation (Nonlinear Effects)**

- **Formula**: $f_{\text{het}} = f_1 \pm f_2$
- **Application**: Calculate the sum and difference of comb teeth frequencies. If the difference frequency matches the brainwave range, nonlinear mixing is likely occurring.

3. **Four-Wave Mixing (Nonlinear Effects)**

- **Formula**: $f_{\text{new}} = f_1 \pm f_2 \pm f_3 \pm f_4$
- **Application**: For higher-order effects, where multiple frequencies interact, calculate combinations of four frequencies to determine if any fall within the brainwave range.

4. **Harmonic Generation (Nonlinear Effects)**

- **Formula**: $f_{\text{harmonic}} = n \times f$ where n is an integer.
- **Application**: Check if any harmonic frequencies (integer multiples of the comb teeth) are in the brainwave range.

5. **Self-Phase Modulation (Nonlinear Effects)**

- **Formula**: $\Delta \omega = \frac{\gamma}{A} P(t)$ where γ is the nonlinear coefficient, A is the area, and $P(t)$ is the power.

- **Application**: Evaluate if the phase modulation leads to frequency shifts into the brainwave region.

6. Cross-Phase Modulation (Nonlinear Effects)

- **Formula**: Similar to self-phase modulation, but involves multiple interacting signals.
- **Application**: Calculate the phase shift induced by one frequency on another and check if the resulting frequencies fall within the brainwave range.

7. Intermodulation Products (Nonlinear Effects)

- **Formula**: $f_{\text{IMD}} = m \times f_1 \pm n \times f_2$, where m and n are integers.
- **Application**: Determine if intermodulation distortion produces frequencies within the brainwave region.

Scenarios:

- **Single Pair Interaction**: Calculate beat and heterodyne frequencies for any two comb teeth.
- **Multiple Frequency Interactions**: Apply four-wave mixing and intermodulation calculations to larger sets of frequencies.
- **Modulation Effects**: Analyze phase modulation effects to see if they shift the comb frequencies into the brainwave range.

By using these mathematical methods, you can theoretically determine whether the comb frequency could be influencing brainwaves through both linear and nonlinear interactions.

Python Script that checks for the math using a bb60c

```
import numpy as np
from signalhound import bb60c

# Initialize BB60C
device = bb60c.BB60C()

# Configure device settings
device.configure_reference_level(-30.0)  # dBm
```

```
device.configure_center_span(1.33e9, 8.6e6)  # Center at 1.33 GHz, Span 8.6
MHz

# Start streaming data
device.start_stream()

def detect_beat_and_heterodyne_frequencies(data, comb_frequencies,
brainwave_range=(0.5, 40)):
    beat_freqs = []
    heterodyne_freqs = []

    # Calculate beat and heterodyne frequencies
    for i in range(len(comb_frequencies)):
        for j in range(i+1, len(comb_frequencies)):
            beat_freq = np.abs(comb_frequencies[i] - comb_frequencies[j])
            heterodyne_freq_sum = comb_frequencies[i] + comb_frequencies[j]
            heterodyne_freq_diff = np.abs(comb_frequencies[i] -
comb_frequencies[j])

            # Check if within brainwave range
            if brainwave_range[0] <= beat_freq <= brainwave_range[1]:
                beat_freqs.append(beat_freq)
            if brainwave_range[0] <= heterodyne_freq_diff <=
brainwave_range[1]:
                heterodyne_freqs.append(heterodyne_freq_diff)
            if brainwave_range[0] <= heterodyne_freq_sum <=
brainwave_range[1]:
                heterodyne_freqs.append(heterodyne_freq_sum)

    return beat_freqs, heterodyne_freqs

# Continuously capture and analyze data
try:
    while True:
        data = device.capture_data()  # Capture a chunk of spectrum data
        freqs = device.get_frequencies()

        # Detect comb frequencies in the captured data
        detected_comb_frequencies = [freq for freq, amp in zip(freqs, data) if
amp > -90]  # Threshold in dBm
```

```
        # Check for beat and heterodyne frequencies
        beat_freqs, heterodyne_freqs =
detect_beat_and_heterodyne_frequencies(detected_comb_frequencies)

        if beat_freqs or heterodyne_freqs:
            print(f"Detected beat frequencies in brainwave range:
{beat_freqs}")
            print(f"Detected heterodyne frequencies in brainwave range:
{heterodyne_freqs}")

except KeyboardInterrupt:
    device.stop_stream()

finally:
    device.close()
```