https://www.rollingstone.com/product-recommendations/electronics/best-sleep-earbuds-1146657/

# Bed Bugs: Tracking Restlessness and Noise During Sleep

KAMERON MIZE, 5/11/2021

CEE5440—INSTRUMENTATION AND SIGNAL PROCESSING

PROFESSOR: DR. SARLO

# Background and Motivation

https://twitter.com/vthrl/status/1256239626941009922

Used with Drew’s permission

https://www.prevention.com/health/a20485882/loudest-alarm-clocks-for-heavy-sleepers/

# Objective

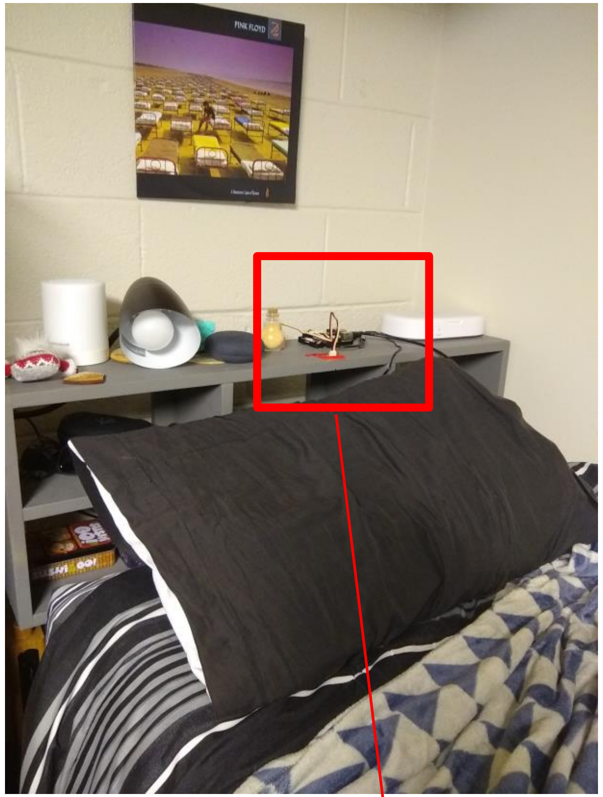
**Track restlessness through accelerometer data and determine if it is correlated to noise levels in my room.**

- Light data was discarded from original objective

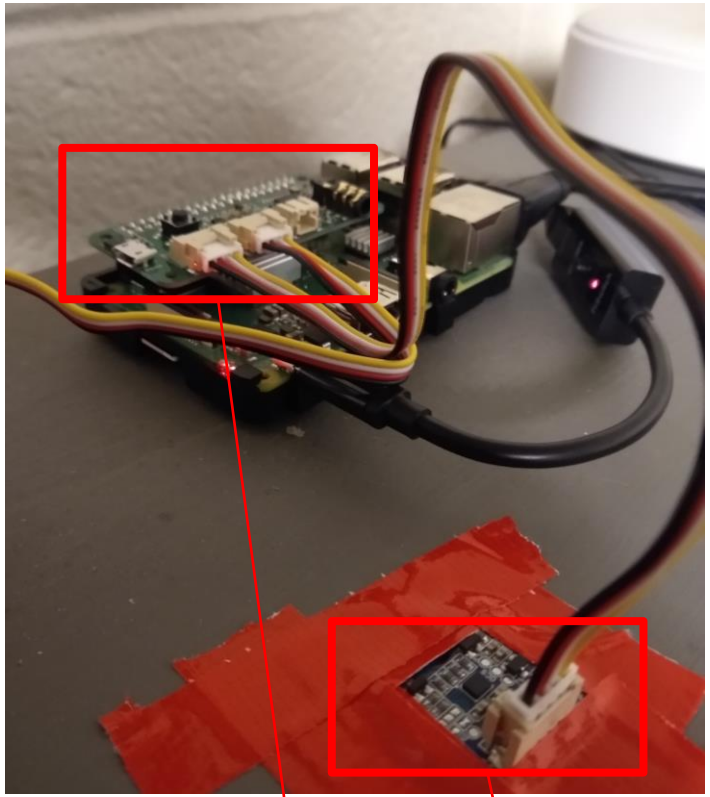
Plan

- Collect acceleration and audio data overnight
- Run event detection on acceleration to locate restless activity
- Take RMS of audio through the night
- Compare RMS of audio before events to nightly average

# Data Collection—Hardware

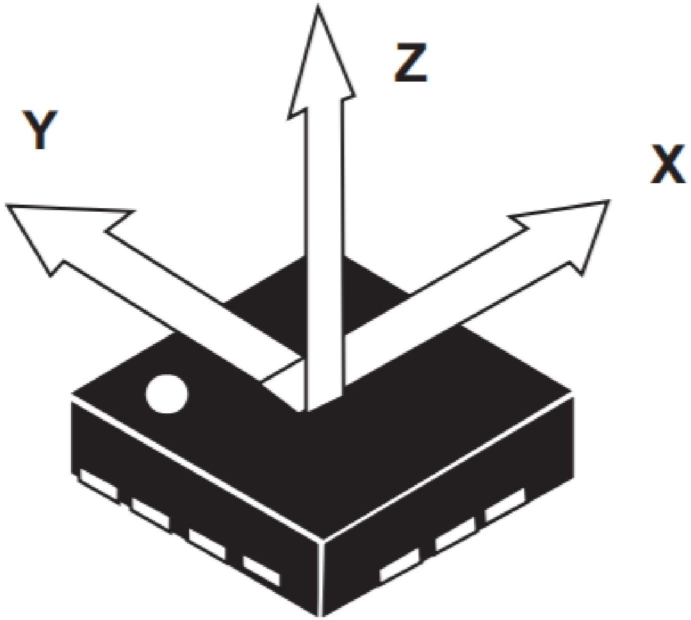


Data Collection Setup



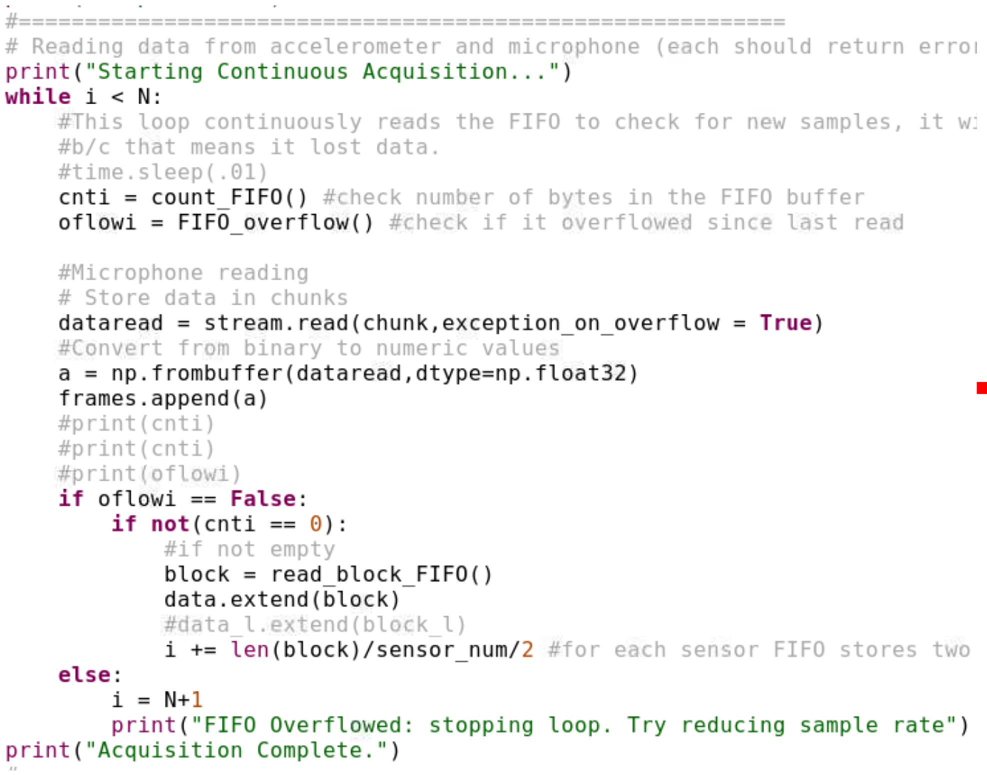
Microphone

Accelerometer



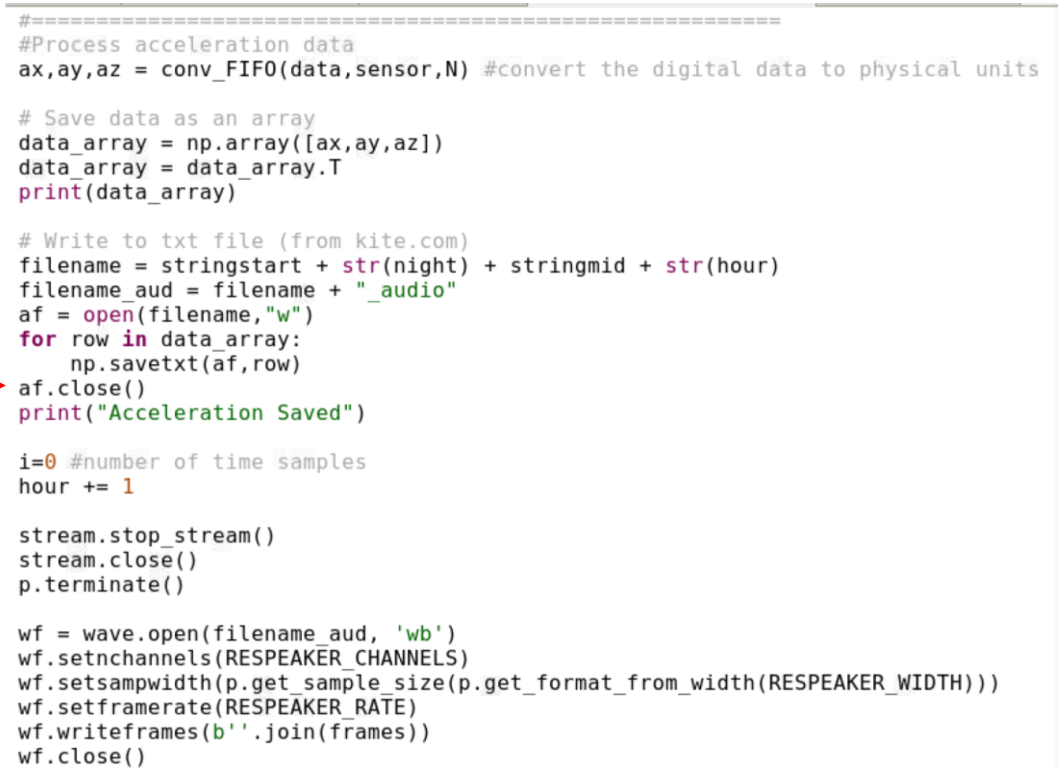
https://files.seeedstudio.com/wiki/Grove-6-Axis_AccelerometerAndGyroscope/res/LSM6DS3TR.pdf

# Data Collection—Software



```python
#==========================================================
# Reading data from accelerometer and microphone (each should return erro
print("Starting Continuous Acquisition...")
while i < N:
    #This loop continuously reads the FIFO to check for new samples, it w
    #b/c that means it lost data.
    #time.sleep(.01)
    cnti = count_FIFO() #check number of bytes in the FIFO buffer
    oflowi = FIFO_overflow() #check if it overflowed since last read

    #Microphone reading
    # Store data in chunks
    dataread = stream.read(chunk,exception_on_overflow = True)
    #Convert from binary to numeric values
    a = np.frombuffer(dataread,dtype=np.float32)
    frames.append(a)
    #print(cnti)
    #print(cnti)
    #print(oflowi)
    if oflowi == False:
        if not(cnti == 0):
            #if not empty
            block = read_block_FIFO()
            data.extend(block)
            #data_l.extend(block_l)
            i += len(block)/sensor_num/2 #for each sensor FIFO stores two
    else:
        i = N+1
        print("FIFO Overflowed: stopping loop. Try reducing sample rate")
print("Acquisition Complete.")
```



```python
#==========================================================
#Process acceleration data
ax,ay,az = conv_FIFO(data,sensor,N) #convert the digital data to physical units

# Save data as an array
data_array = np.array([ax,ay,az])
data_array = data_array.T
print(data_array)

# Write to txt file (from kite.com)
filename = stringstart + str(night) + stringmid + str(hour)
filename_aud = filename + "_audio"
af = open(filename,"w")
for row in data_array:
    np.savetxt(af,row)
af.close()
print("Acceleration Saved")

i=0 #number of time samples
hour += 1

stream.stop_stream()
stream.close()
p.terminate()

wf = wave.open(filename_aud, 'wb')
wf.setnchannels(RESPEAKER_CHANNELS)
wf.setsampwidth(p.get_sample_size(p.get_format_from_width(RESPEAKER_WIDTH)))
wf.setframerate(RESPEAKER_RATE)
wf.writeframes(b''.join(frames))
wf.close()
```

# Data Collection—Software

Collected test data for tuning
- Acceleration at 104 Hz
- Audio at 30,000 Hz (worried about overflow)

Overflow
- Acceleration did not overflow
- Could not determine if audio overflowed, but no error was passed

Program failed after 253 minutes on both nights

# Signal Processing—Audio

Audio data was very noisy

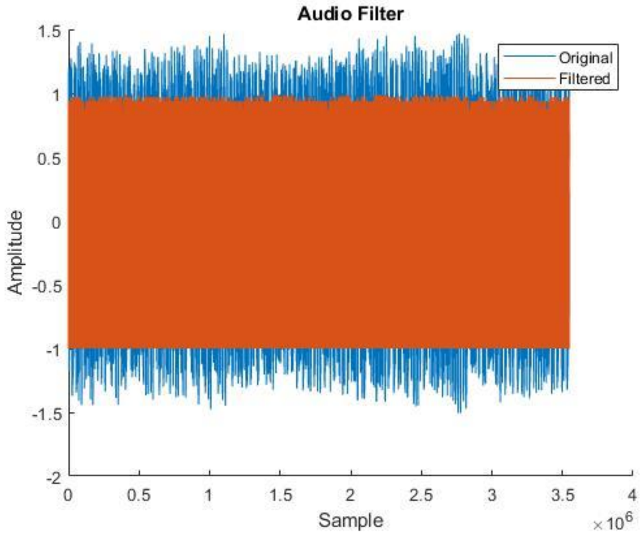
- Lots of white noise
- Contained chirps – not due to discontinuities
- Speech could be heard, but not understood

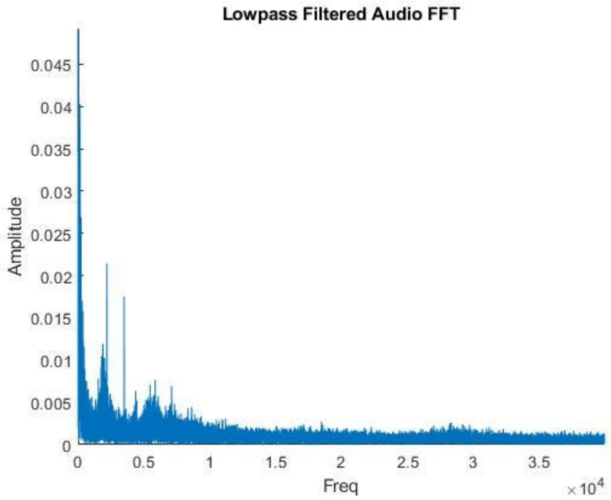
FFT

- Showed there was noise at all points in the spectrum

Filter

- Filtered using a 3rd order lowpass filter





# Signal Processing—Acceleration

Fairly Straightforward

- Collected at 104 Hz
- Lowpass filtered at Nyquist
- Event detection run on following data

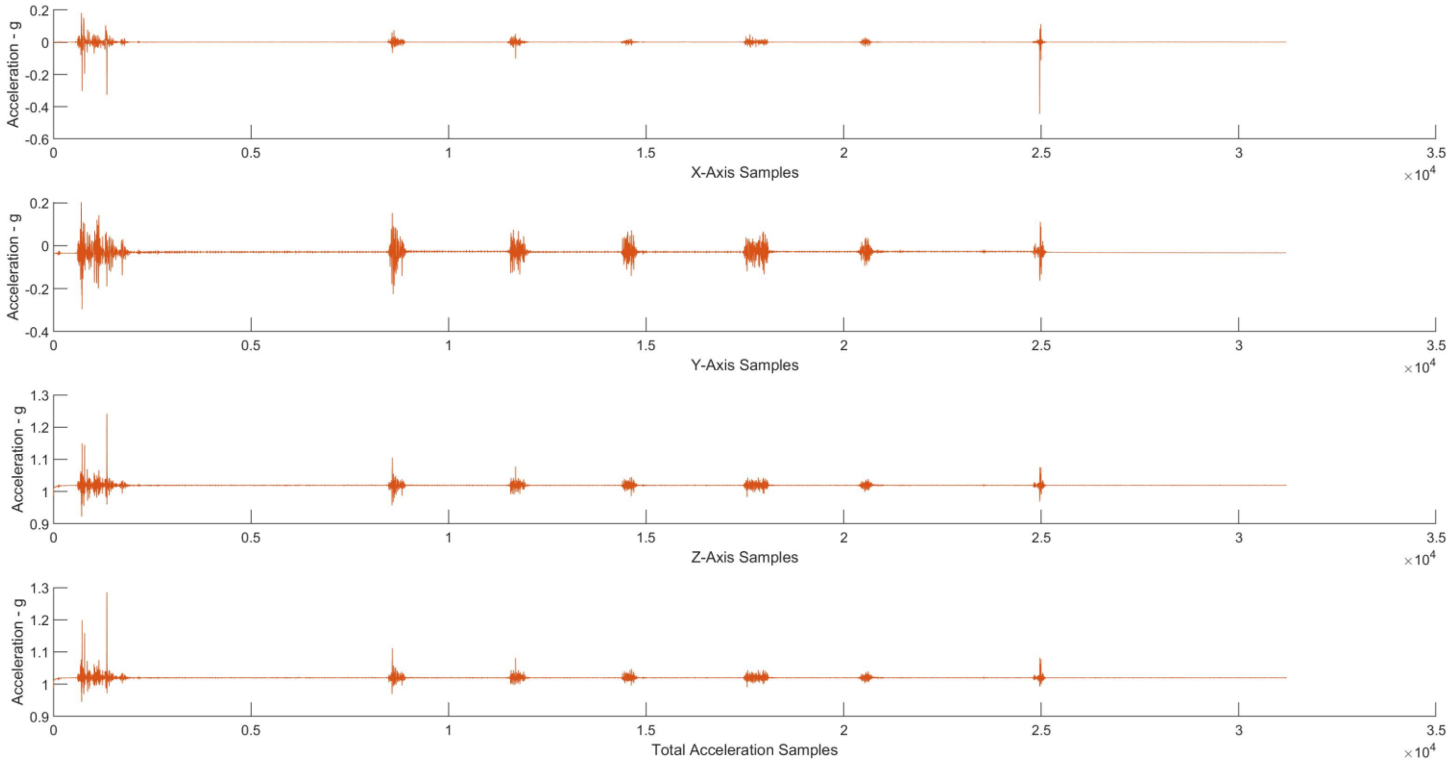
# Event Detection

Collected test data for tuning

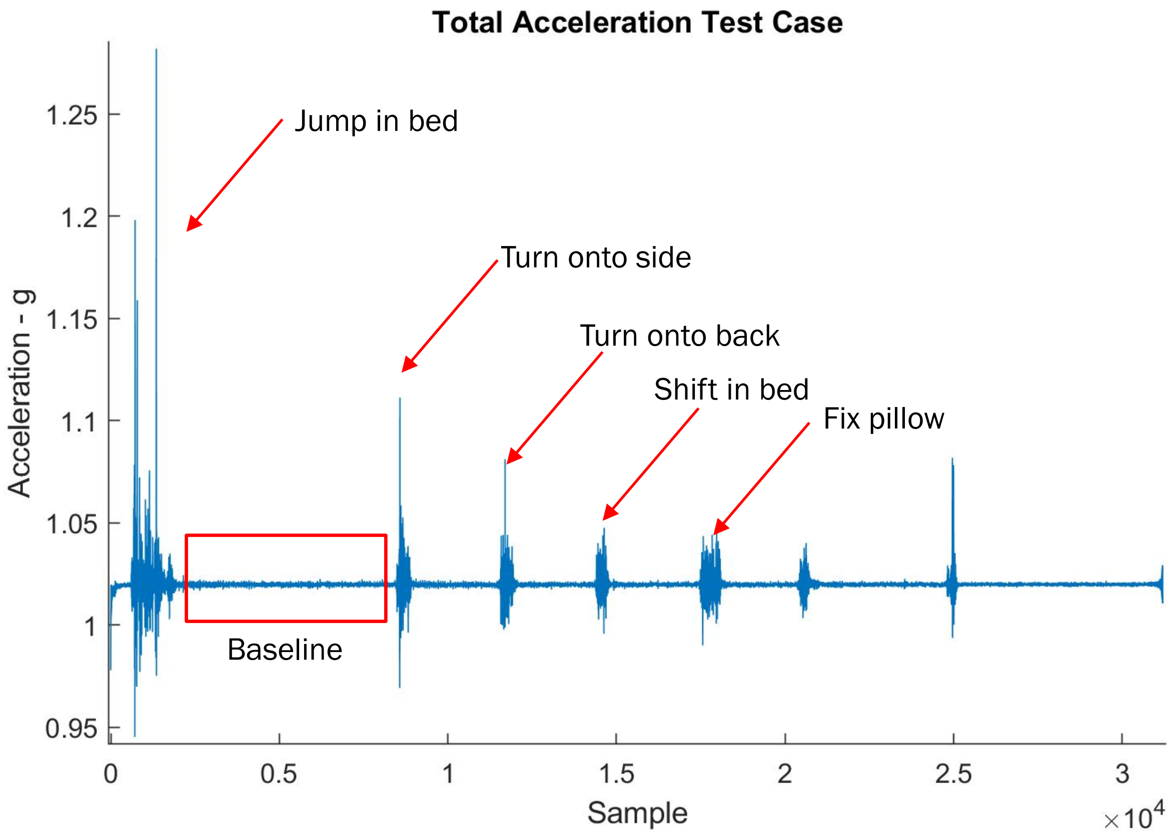
- Set to 100 Hz for filter (was really 104 Hz, fixed in actual night data)

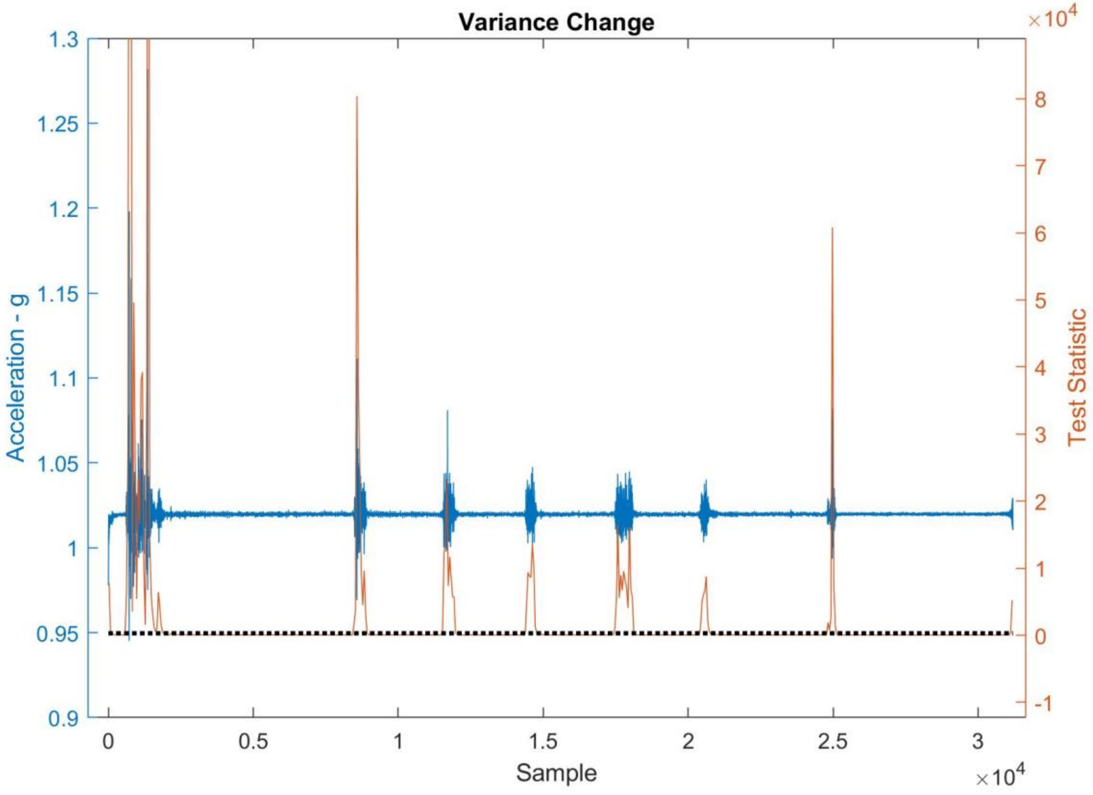
Event Detection

- Focused on variance
- Beta = 3
- N = 50, later changed to 104 to try to smooth and correspond to seconds

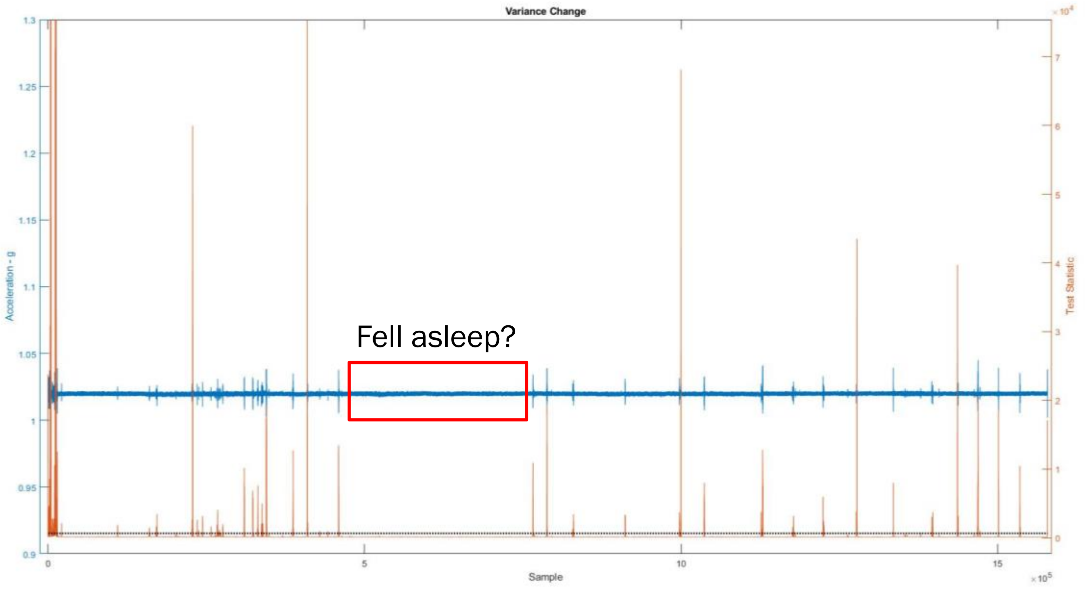


# Event Detection

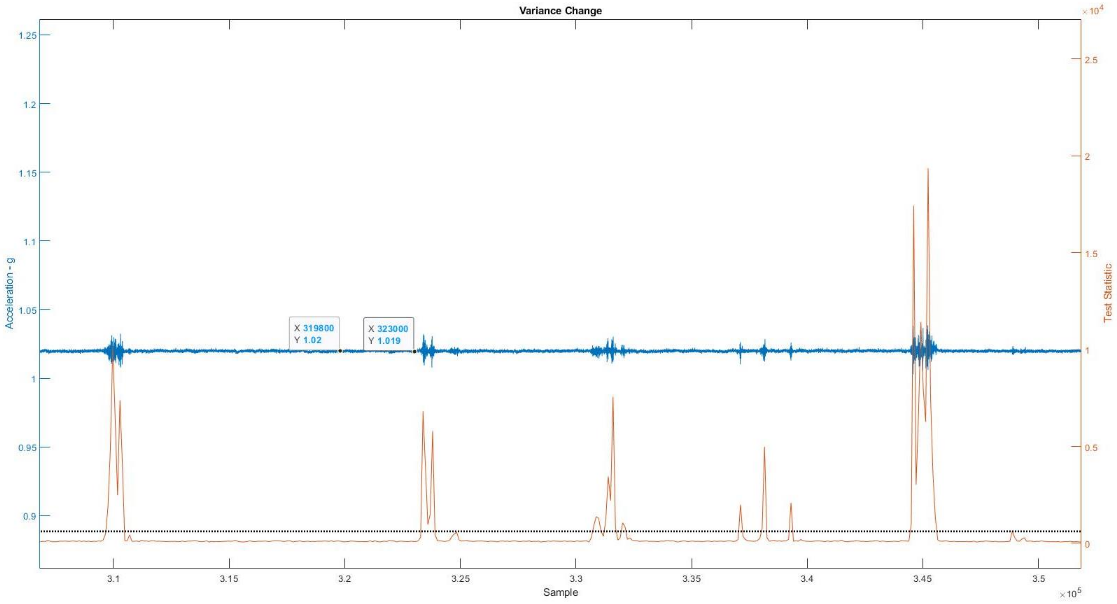




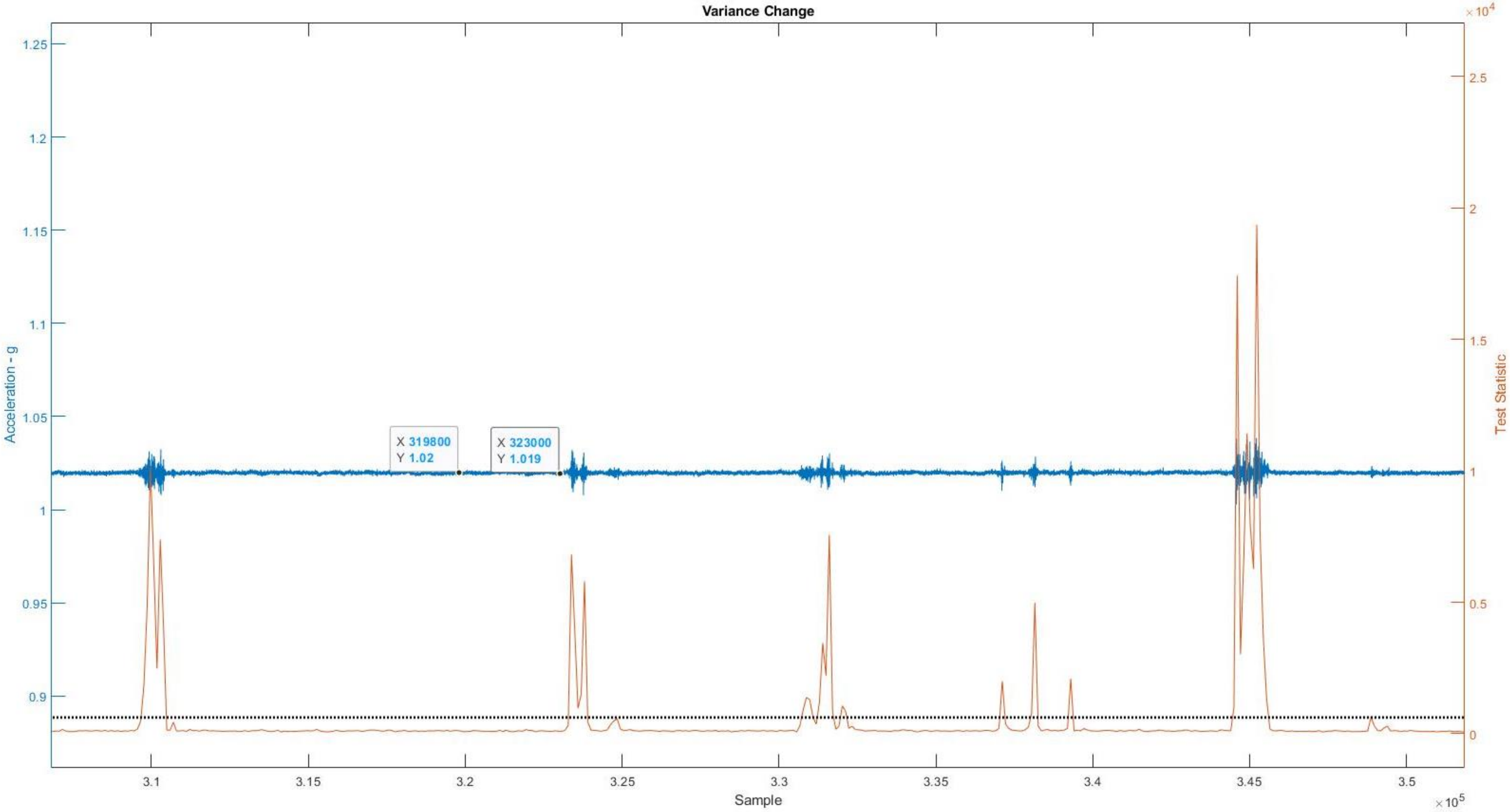
# Results



Fell asleep?

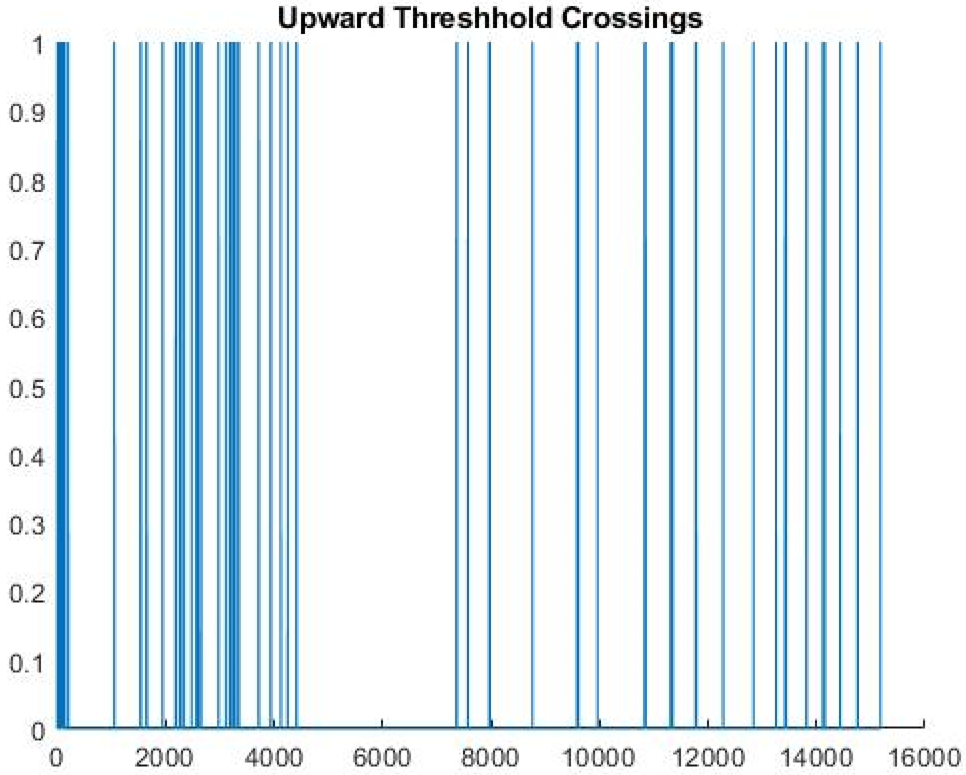


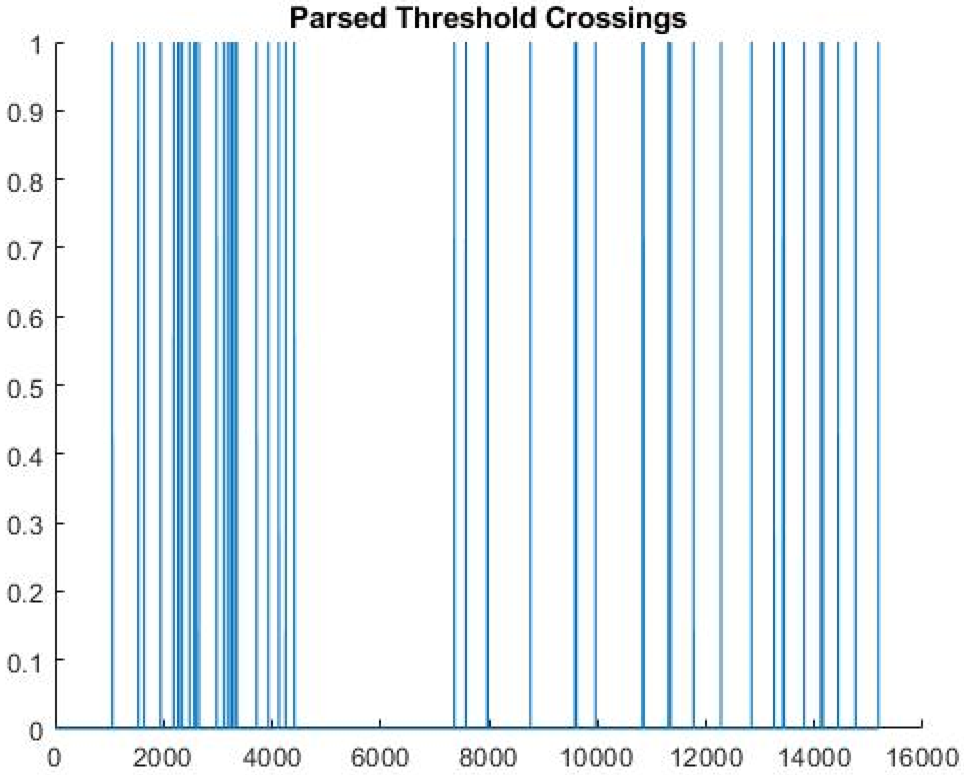
# Results



Variance Change
Acceleration - g
Test Statistic
Sample
1.25
1.2
1.15
1.1
1.05
1
0.95
0.9
×10^4
2.5
2
1.5
1
0.5
0
3.1
3.15
3.2
3.25
3.3
3.35
3.4
3.45
3.5
×10^5
X 319800
Y 1.02
X 323000
Y 1.019

# Results


Upward Threshhold Crossings
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
2000
4000
6000
8000
10000
12000
14000
16000


Parsed Threshold Crossings
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
2000
4000
6000
8000
10000
12000
14000
16000

# Results

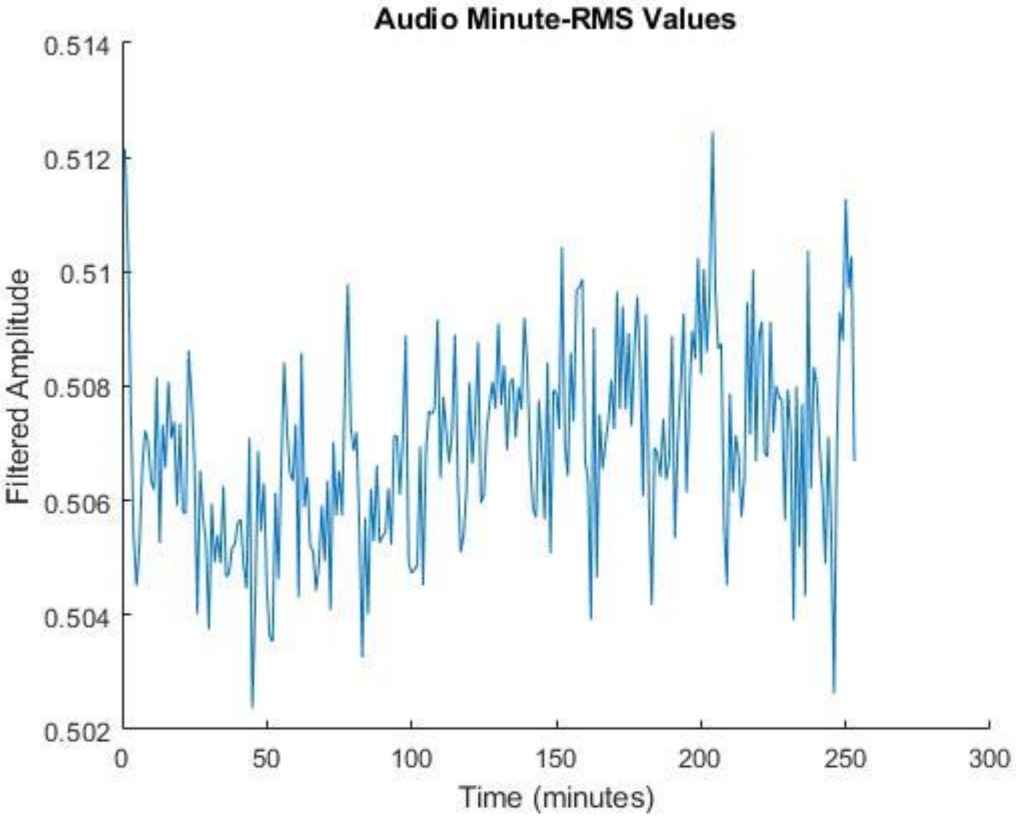
Audio RMS Values

- Appears slightly higher later in the night
- Very noisy
- Only slight variations in actual values

Event vs. Overall RMS

- Overall: 0.5069
- Event: 0.5065

Almost exactly the same! Audio is likely too noisy, but I couldn't figure out how to get it nicer. Either that or noise makes me sleep easier.



# Conclusion and Future Work

- Figure out how to get less noisy audio data
  - New microphone setup?
  - Different settings on current hat?
- Extend time that can be recorded
  - Must identify error causing halt
- Identify Important noise frequencies from test data in room and exclude everything else

# Questions?