Filtering and Smoothing

Embedded Systems

Based on Lecture Notes from Deepak Ganesan

Information and Noise in Signals

- Most sensor data is affected to some extent by **noise**
  - unexplained variations in the data
- Data analysis is often considerably simpler if this noise can be removed from the data.
  - This process is referred to by various terms - **data smoothing**, **filtering**, cleaning, and so on.
  - Remove the noise while retaining the important characteristics of the signal.
- Noise removal techniques can be divided into two class.
  - The first is **time-domain** approaches, which is the more intuitive way of approaching the problem.
  - The second is **frequency-domain** approaches, which removes noise that is periodic in nature.

Information and Noise in Signals

- Understanding how information is contained in the signals you are working
  - with, and what types of noise can **corrupt** the signal and make it difficult to extract the information.
- There are two ways that are common for information to be represented in naturally occurring signals.
  - Information represented in the **time domain** describes when something occurs and what the **amplitude** of the occurrence is.
    - e.g. temperature, your location
  - Information represented in the **frequency domain** is more indirect— many things in our universe show periodic motion
    - e.g. your heart beats in a quasi-periodic manner.
    - By measuring the **frequency**, **phase**, and **amplitude** of this periodic motion, information can often be obtained about the system producing the motion.

Noise in Accelerometer Data

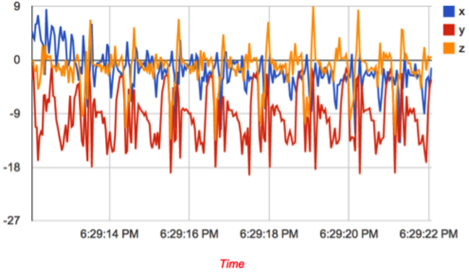


Figure: Typical accelerations along the three axes (x, y, and z) while walking.

- Some periodic pattern in this data, but really noisy
- The noise in the accelerometer signal can be categorized into two types.
  - **Intrinsic noise:** Electronic noise from the circuitry that is converting the motion into a voltage signal.
  - **External vibration noise:** Noise from the movements external to the sensor and therefore affects the readings.

Noise in Electrocardiogram (ECG) Data

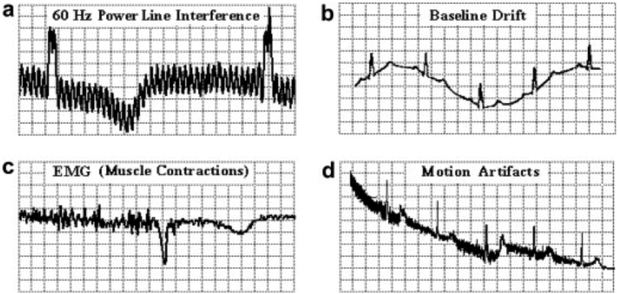


Figure: Typical ECG signal with different interference sources.

- Power line interference i.e. the 50Hz power line signal causes electromagnetic interference which is recorded by the ECG device.
  - Many other sources of ECG noise are present as well including those caused by breathing, muscle contractions, body movement ...

Noise in Audio Signals

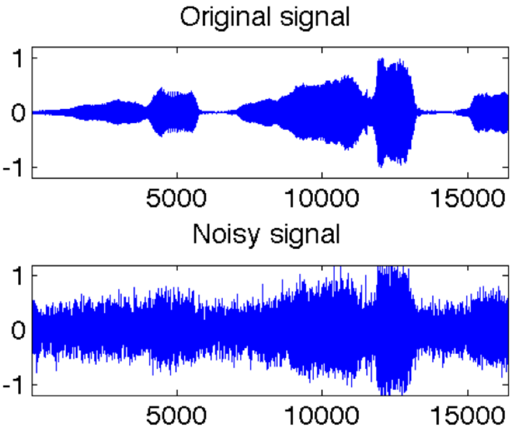


Figure: Noisy audio signal.

- The noise could be due to ambient sound.

Noise in GPS Data

Figure: Noisy GPS readings while driving in red. Actual trajectory in green.

- GPS readings can be noisy due to clock error, tropospheric delays, multipath effects due to buildings, weather conditions, and so on.

Time-series Smoothing and Filtering

- Many sources of noise tend to be random in nature.
  - Informally, this means that the noise has roughly equal amounts of positive and negative changes, and there is no pattern in the noise over time.
  - Formally, the noise is said to be **uncorrelated** in time, has zero mean, and finite variance.
- In this case, noise can be reduced by over-sampling the sensor and averaging the values.
  - Mathematically, if you have N samples of a random noise signal, and average these samples, your noise reduces by a factor of $\frac{1}{\sqrt{N}}$.

Moving Average Smoothing

- Instead of averaging and reducing the number of samples, one can also perform a **moving average**.
- Accelerometer signal: $x = [x_1, x_2, x_3, \ldots x_n]$ where the index is the sample number.
- Moving average filter is for a window of **w**

$$s_i = \frac{x_i + x_{i+1} + \ldots + x_{i+w-1}}{w}$$

- Example: the output of the moving average filter for $w = 3$:

$$s_1 = (x_1 + x_2 + x_3)/3$$
$$s_2 = (x_2 + x_3 + x_4)/3$$
$$s_3 = (x_3 + x_4 + x_5)/3$$

- As you increase the smoothing window, the signal will look cleaner
  - But using too large a window since you will smooth out the important *characteristics* of the signal.

Exponential smoothing-I

- Moving average assumes random noise where the statistics of the noise *does not change* over time.
  - But what happens if the noise itself is a **time varying**?
- Exponential smoothing is to assign *exponentially decreasing* weights as the observation get older.

$$
\begin{aligned}
s_1 &= x_0 \\
s_t &= \alpha x_{t-1} + (1-\alpha)s_{t-1} = s_{t-1} + \alpha(x_{t-1} - s_{t-1}); \quad t > 1
\end{aligned}
$$

where $0 < \alpha < 1$ is the *smoothing factor*.

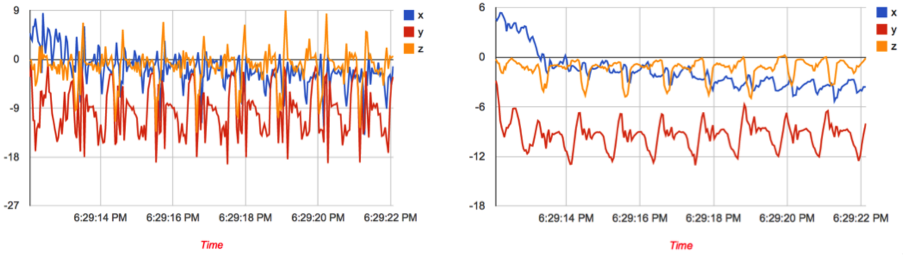
Exponential smoothing-II

- Why exponential?

$$
\begin{aligned}
s_t &= \alpha x_{t-1} + (1-\alpha) s_{t-1} \\
&= \alpha x_{t-1} + \alpha(1-\alpha) x_{t-2} + (1-\alpha)^2 s_{t-1} \\
&= \alpha\left[x_{t-1} + (1-\alpha) x_{t-2} + (1-\alpha)^2 x_{t-3} + \ldots\right] + (1-\alpha)^{t-1} x_0
\end{aligned}
$$

- The weights assigned to previous observations are in general proportional to the terms of the geometric progression $1, (1-\alpha), (1-\alpha)^2, (1-\alpha)^3, \ldots$.

Exponential smoothing-Example

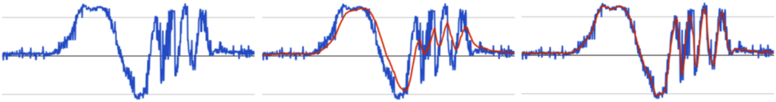


- Smoothing (right) after exponentially weighted smoothing with smoothing $= 6$ (i.e. $\alpha = \frac{1}{6}$)

Median Filtering

- When the noise appears like sudden spikes in the data, then the moving average and exponential smoothing methods are not the best methods.
- Exponential smoothing will remove noise, but has two issues:
  - It averages some of the peaks in the data and they don't have *the same amplitude*.
  - Averaging causes a time lag in the peaks i.e. the peaks are *shifted* slightly to the right of the original peak.
- Median filter:

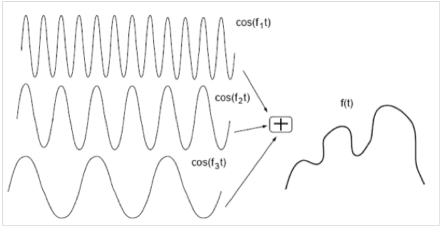
$$s_{n-2} = median(x_{n-2}, x_{n-1}, x_{n-1})$$



(a) Raw accelerometer readings (b) Exponential smoothing (c) Median filtering

Frequency-domain Filtering

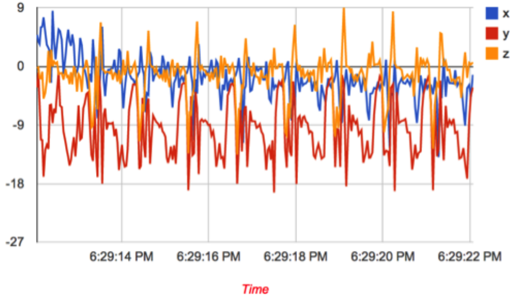
- Any periodic waveform can be expressed as the **sum** of an infinite set of *sine waves*—Jean Baptiste Fourier.



- The inverse is also true: You can take any time-series pattern and break it down into a weighted sum of sinusoidal waves.

Frequency-domain Filtering

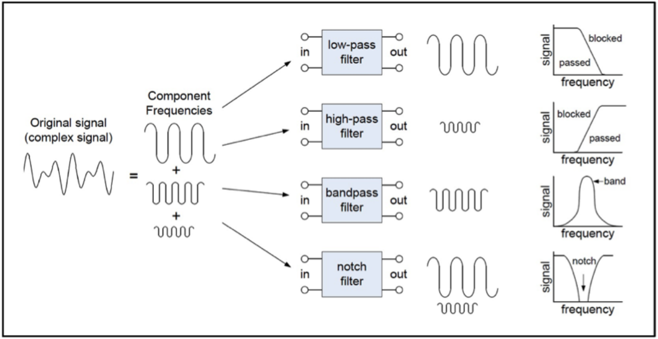
- The main idea is that noise in these waveforms are often concentrated in some frequencies, and not in others.



- The rate at which you walk is typically one or two steps a second; even if you run, the step rate is a few steps a second.
  - So the frequency of interest is only a few Hz.
  - Similarly, in the case of ECG, the useful frequencies of the electrical signals in the heart are between 0.5 - 150 Hz.

Frequency-domain Filtering

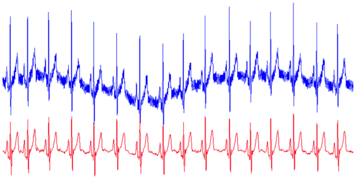
- Once you convert a signal to a weighted sum of sinusoidals, you can just *remove* all the sinusoidals whose periods are outside the range that you expect
  - What you are left with is a much cleaner signal!
- This core idea is that by converting data from the **time-domain** (which we normally look at) to the **frequency domain** (which is this new way of viewing data as sinusoidals)

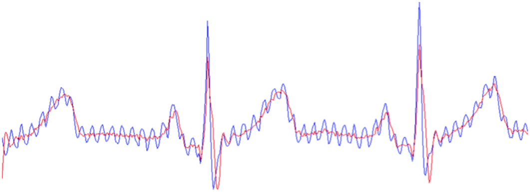


Frequency-domain Filtering

- **Low-pass filter:** passes signals with frequency lower than a certain cutoff frequency and attenuates (i.e. reduces the effect of) signals that are higher than the cutoff frequency.
- **High-pass filter:** passes signals with frequency higher than a cutoff frequency and attenuates signals that are lower than the cutoff.
- **Bandpass filter:** allows signals between certain frequencies to pass through but attenuates signals outside this band.
- **Notch filter:** attenuates the signal within a very small band but lets the other frequencies go through unaltered.

Example-ECG Noise Removal



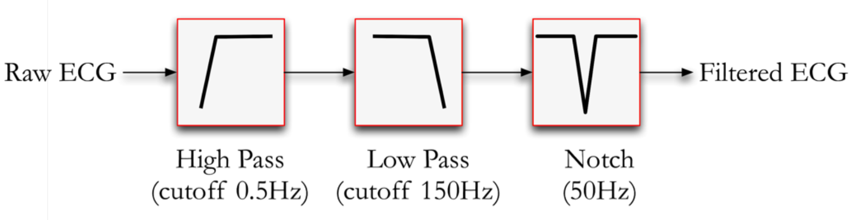


- **Baseline Wander:** A low-frequency component due to offset voltages in the electrodes, periodic breathing, and body movement.

- **Powerline Noise:** The frequency of alternating current in the electrical mains is typically around 50-60Hz.
- **High frequency Noise:** Various other electronic equipment in the vicinity of the ECG sensor including pacemakers, mobile phones, and other electronics can interfere with the ECG signal.

Example-ECG Noise Removal

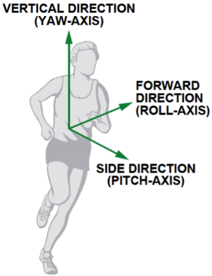
- The ECG signal of interest is between 0.5Hz to 150Hz
  - We can remove baseline wander by having a high-pass filter with a cutoff of 0.5Hz
  - We can remove high frequency noise by having a low-pass filter with a cutoff of 150Hz
  - This leaves us with powerline interference, which we can remove with a notch filter with a 50Hz cutoff.



Filtering Takeaways

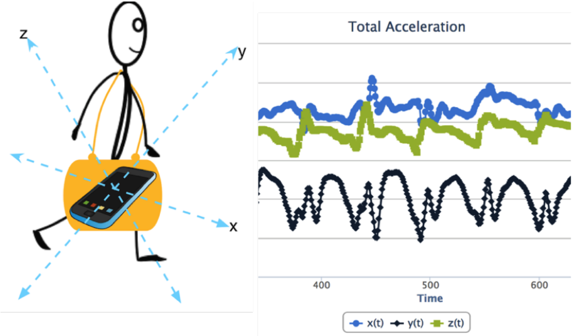
- Noise removal is the foundation for any analytics that you wish to perform with sensor data.
  - In practice, it is often important that you try to identify the **source** and **type** of noise in the signal, since that will allow you to remove the noise more effectively.
- Any method that you use for removing noise can come with some unwanted artifacts that you may have to deal with in later stages of analysis.
  - Make sure that you know the advantages and disadvantages of the filtering approaches that you are using before you start using them in your data processing.

Accelerometer



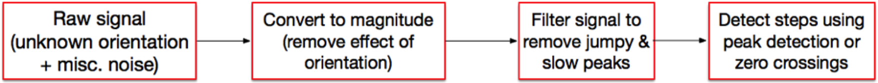
- The three components of motion for an individual (and their related axes) are forward (roll), vertical (yaw), and side (pitch).
- The 3-axis accelerometer senses acceleration along its three axes: x, y, and z.
- The pedometer will be in an unknown orientation
  - The measurement accuracy should not depend critically on the relationship between the motion axes and the accelerometer's measurement axes.

Example: Step Detection



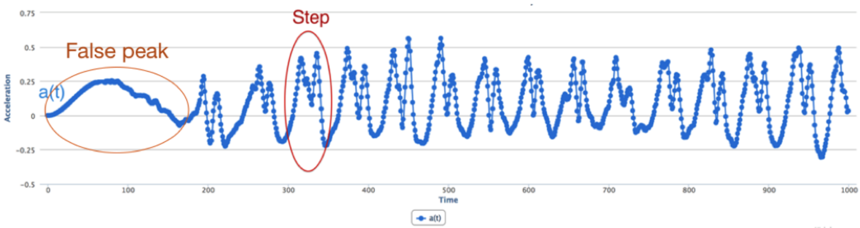
- Since acceleration changes as a result of a step can result in changes along all the three axes, so we need to design an orientation-independent algorithm to detect steps.

Example: Step Detection



- The key insight in our method is to convert the 3-axis signal into a one axis magnitude signal, and then extract steps from this signal.
- **Step 1-) Extract signal magnitude:** Take the magnitude of the entire acceleration vector i.e. $\sqrt{x^2 + y^2 + z^2}$, where x, y, and z are the readings of the accelerometer along the three axes.

Example: Step Detection

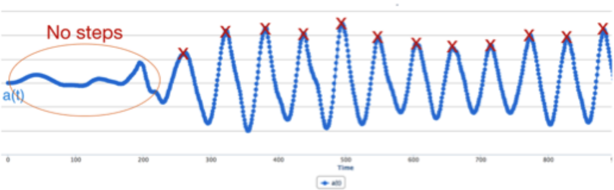


- **Step 2-) Filter the signal to remove noise:** is to remove noise, and extract the specific signal corresponding to walking.
  - *Jumpy peaks:* Since the phone is often carried in a pocket/purse, it can jiggle a little with each step.
  - *Short peaks:* Small peaks can occur when a user is using a phone (e.g. making a call or using an app).
  - *Slow peaks:* Slow peaks can occur when the phone is moved or due to movements of the leg while sitting (if the phone is in the pant pocket)

Example: Step Detection

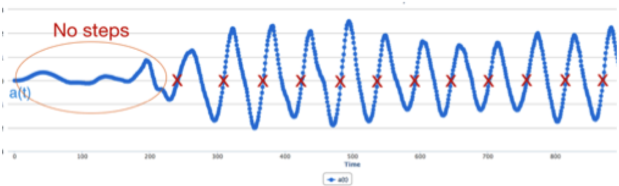
- To remove these sources of noise, we are going to use *frequency-domain* noise removal.
- We need to remove high frequency variations like jumpy peaks and low frequency variations like slow peaks.
  - A simple solution is to use a filter that keeps only frequencies relating to walking and removes the rest.
  - For example, we know that typical walking pace may be under three steps a second (3 Hz) and over half step a second (0.5Hz)
  - Remove all frequencies above 5 Hz and below 0.5 Hz (just to give some margin for error).
  - Note that this method would not be able to detect running or bicycling, which may have higher pace.

Example: Step Detection



- **Step 3-) Detecting Steps:** Look for large peaks and use that to detect steps.
  - Another approach is to take the derivative (slope) of the smoothed acceleration signal. The derivative changes from negative to positive (or positive to negative) exactly when a step occurs, so you can just count the number of times the derivative changed from negative to positive to detect the number of steps that occurred.

Example: Step Detection



- Another possibility is to do subtract the mean for each window and look at zero crossings i.e. times when the signal crosses from the negative to positive in the upward direction.