6.3 SIGNAL PROCESSING

Sources of noise

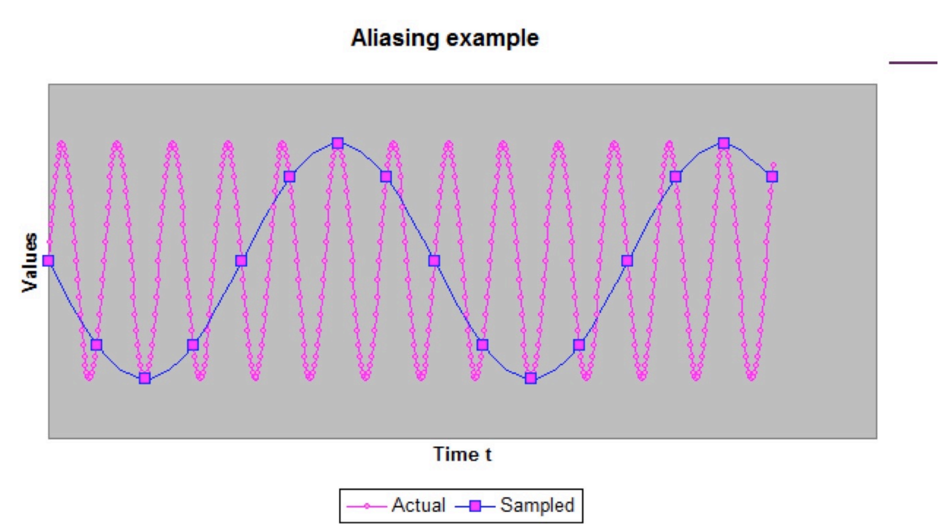
- skin motion artifact
- human error
- electronic noise (60 Hz)
- improper sampling

Angular velocities + accelerations

$$\text{velocity} = \omega = \frac{\Delta\theta_{hip}}{\Delta t}$$

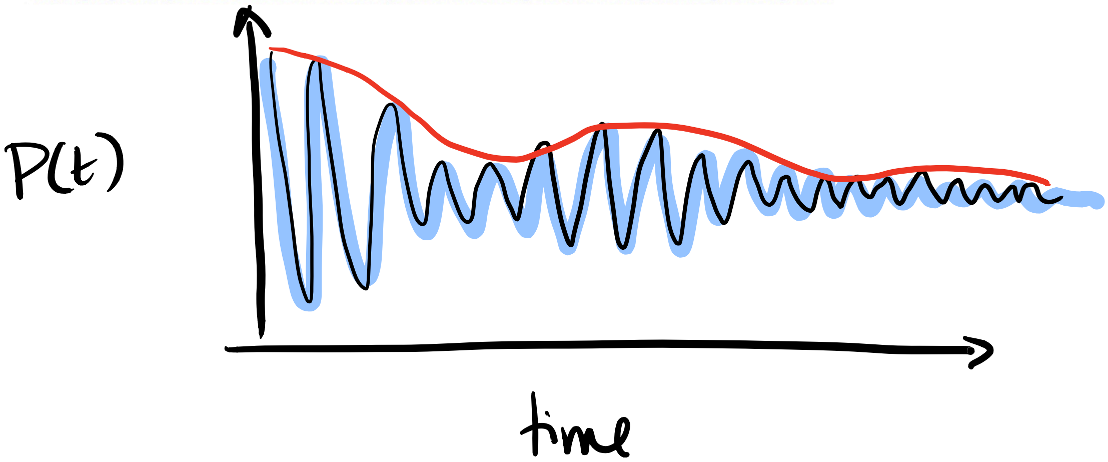
$$\text{Acceleration} = \alpha = \frac{\Delta\omega}{\Delta t}$$

Noise becomes a problem!



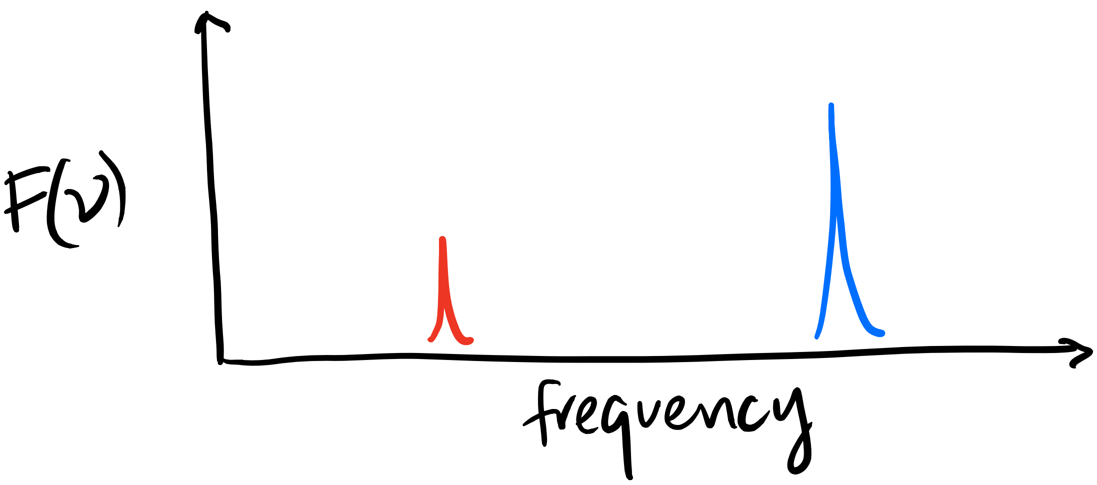
Nyquist sampling theorem:
Sampling freq ≥ 2 * highest freq of actual signal of interest

http://gregstanleyandassociates.com/whitepapers/FaultDiagnosis/Filtering/Aliasing/aliasing.htm



Convert to frequency domain using fast fourier transform (FFT)

How many frequencies are in this signal?



Which is the signal?

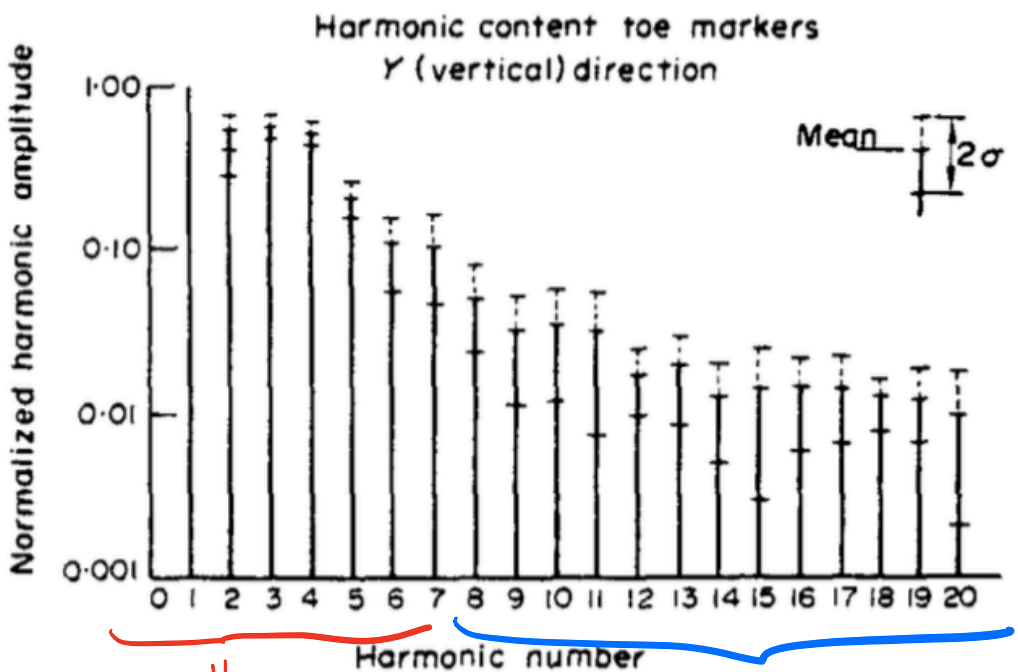
Walking → 120 steps/minute

Step frequency = 2 Hz

Stride frequency = 1 Hz ← which marker will we track to get this??

In repitive movements → frequencies will be at harmonics of stride frequency

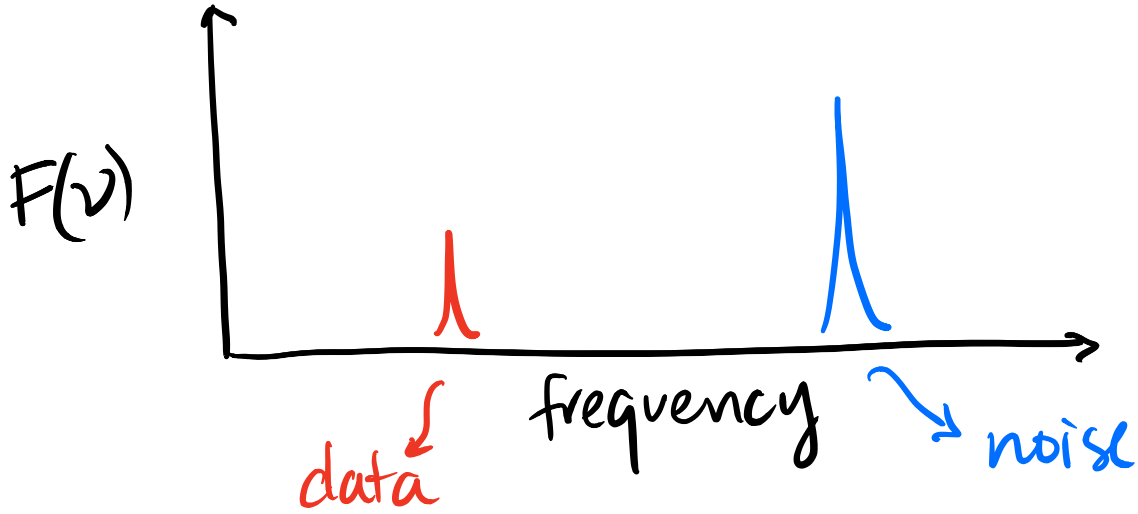
- Most of the data is below 6 Hz



data

noise

Noise tends to be random ⇒ tends to be high frequency ⇒ show Winter text



For biomechanics ⇒ tend to use low pass filter ⇒ Butterworth

CAREFUL! may be activity dependent!

Butterworth not so good for step/impulse inputs

- show some videos

Must choose appropriate cut off frequency

CAREFUL! may be activity dependent!

- Many options
- We will use residual analysis

For each cut off frequency (f_c), the residual (R) is difference between raw data (X) and filtered data ($\hat{X}$) of all datapoints (N)

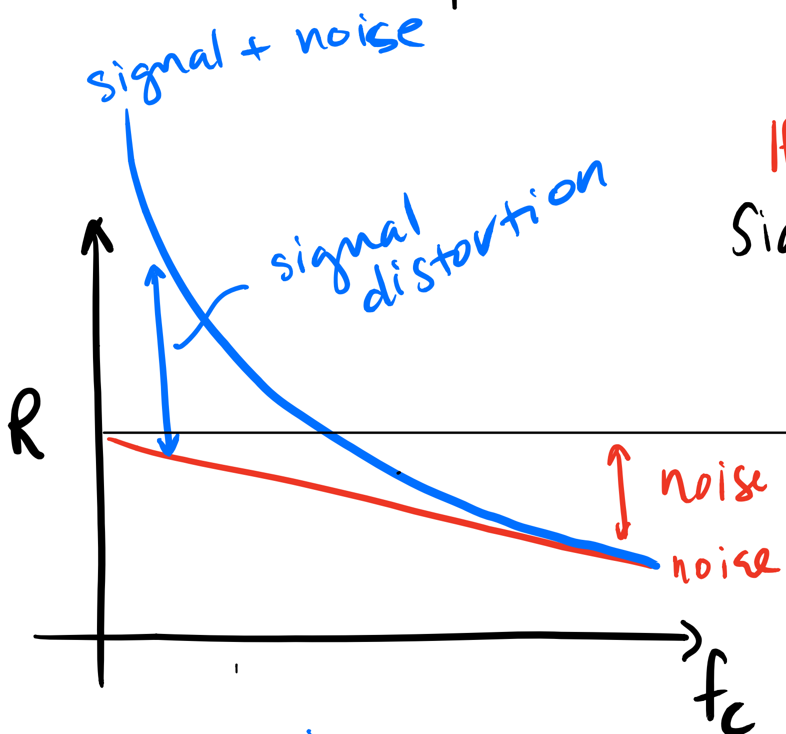
In Math-ese: $(x-\hat{x})$

all data $\sum_{i=1}^{N}(x_i-\hat{x}_i)$

avoid mean = 0 $\sqrt{\sum_{i=1}^{N}(x_i-\hat{x}_i)^2}$

repeat for different f_c

$R(f_c)=\sqrt{\sum_{i=1}^{N}(x_i-\hat{x}_i)^2}$

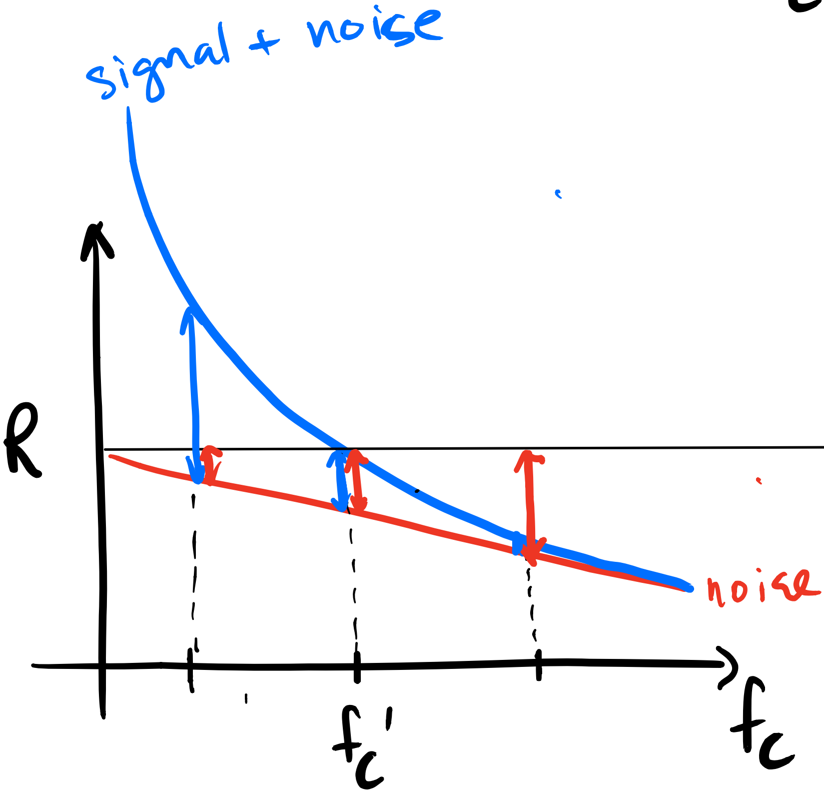


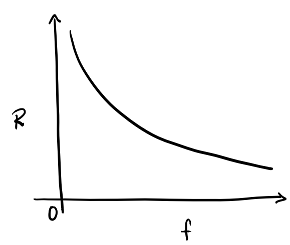
If all noise

Signal + noise $\Rightarrow$ residual increases as $f_c \downarrow$

noise passed through filter

For distorion = noise,

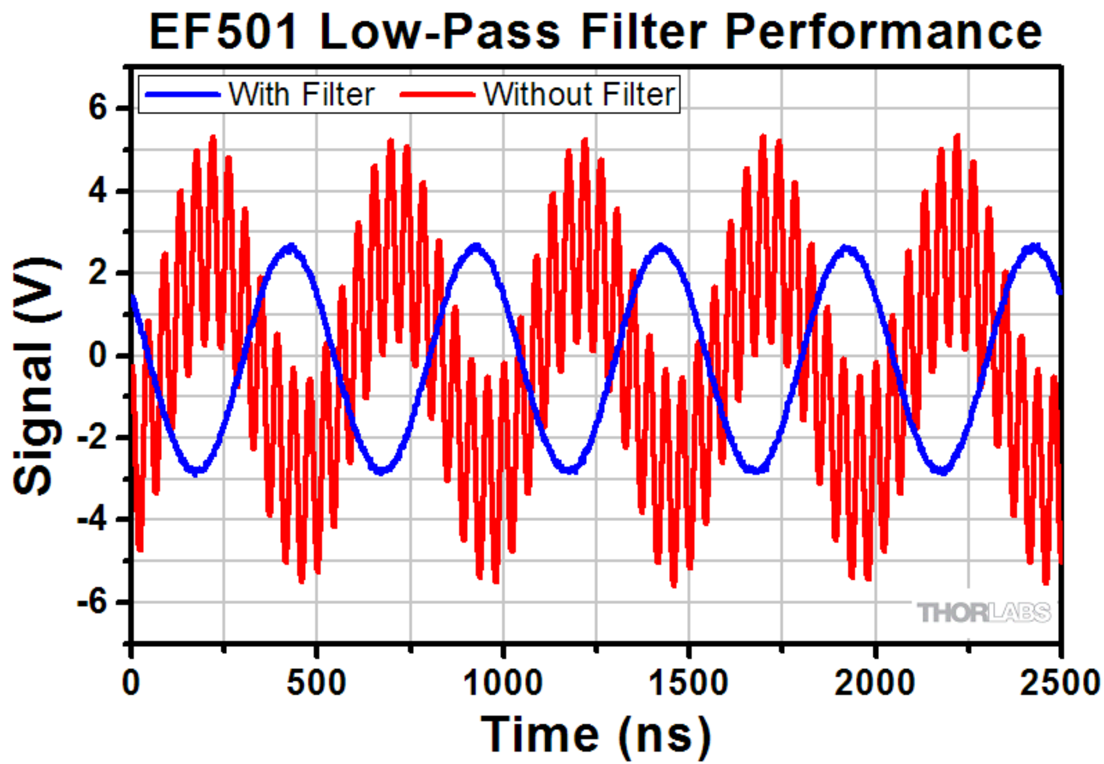




+ Lag due to filtering

Filtering causes a phase shift in the signal → seen as a "lag"

Use recursive or forward-backward filtering to get zero-lag (Matlab: filtfilt)



+ MATLAB Commands

[b,a]=butter(n,wn);

y=filtfilt(b,a,x);

$N_f =$ Nyquist frequency

$N_f = \frac{1}{2} f_s$

a,b = coefficients for Butterworth filter

$\frac{f_c \text{ (cutoff freq)}}{N_f}$

n = n^{th} order filter,

wn = cutoff freq must be btw. 0.0 < wn < 1.0, where 1.0 corresponds to ½ sampling rate

filtfilt = gives zero-lag recursive filtering

y = filtered version of signal x

+ MATLAB Commands

Example for data collected at 100 Hz:

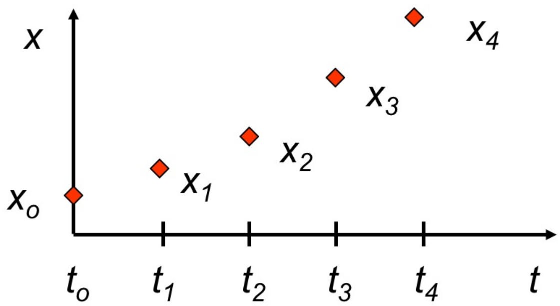
```
% filter the data

% 4th order Butterworth with 6Hz cut-off frequency

% use zero-phase forward-backward filtering


[b,a]=butter(n,wn);

y=filtfilt(b,a,x);
```

wn = cutoff freq must be btw. 0.0 < wn < 1.0, where 1.0 corresponds to ½ sampling rate

What should be the values for *n* and *wn*?

```
[b,a]=butter(4,0.12);
```

+ How to calculate velocity if we know position?



+ Three options:

Euler's Method
(Forward Difference Method)

$$\frac{d(x_n)}{dt} = \frac{x_{n+1} - x_n}{\Delta t}$$

Backward Difference Method

$$\frac{d(x_n)}{dt} = \frac{x_n - x_{n-1}}{\Delta t}$$

Three-Point Formula
(Centered-Difference
Formula)

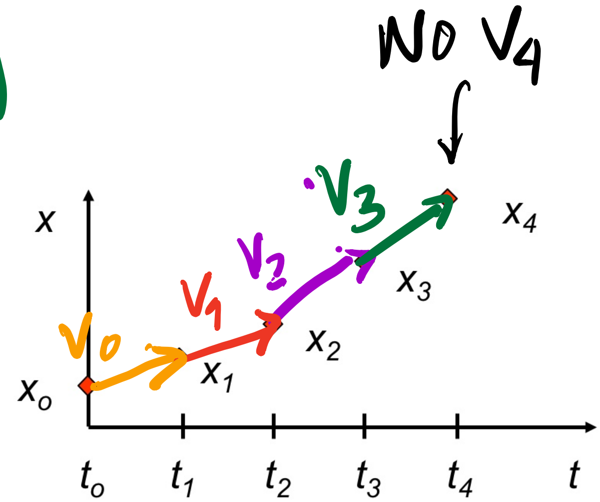
$$\frac{d(x_n)}{dt} = \frac{1}{2}\left[\frac{x_{n+1} - x_{n-1}}{\Delta t}\right]$$

Euler's Method (Forward Difference Method)

Simplest!

velocity

$$\frac{dx_n}{dt} = \frac{x_{n+1} - x_n}{\Delta t}$$



If $x: 1 \rightarrow n$ datapoints

$\frac{dx}{dt} : 1 \rightarrow (n-1)$

Error $\sim f(\Delta t)$

$\downarrow \Delta T$, more accurate

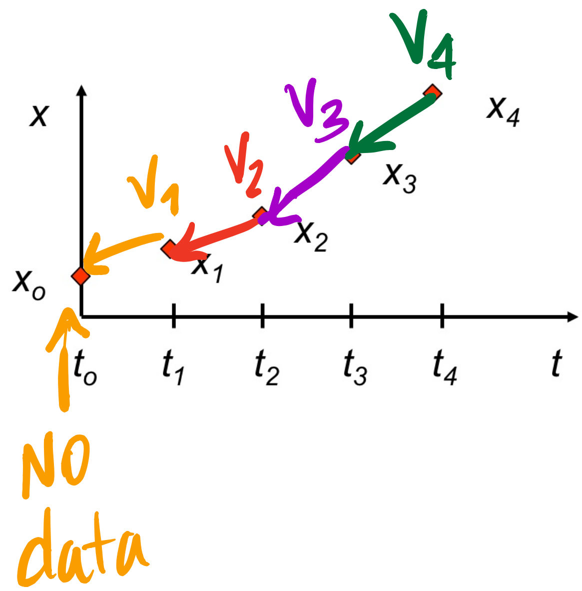
Matlab function diff

diff $= [x_n - x_{(n-1)}]$ vector

$\Delta t \rightarrow$ constant

+ Backward Difference Method

Just like forward, but backward

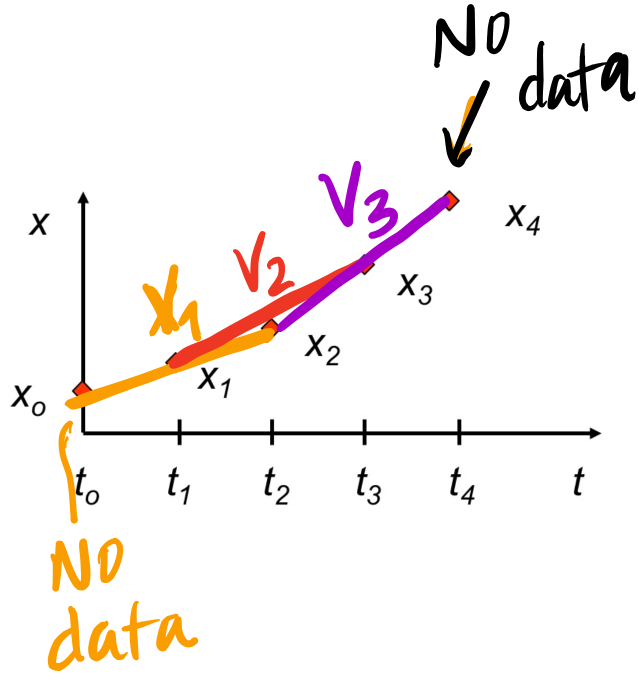


$$\frac{dx_n}{dt} = \frac{x_n - x_{n-1}}{\Delta t}$$

$$\text{error} \sim (\Delta t)$$

Three-Point Formula (Centered-Difference Formula)

- Average of forward & backward



$$\frac{dx_n}{dt} = \frac{1}{2}\left(\frac{x_{n+1} - x_{n-1}}{\Delta t}\right)$$

error $\sim$ Δt^2 :)

Comp. more expensive :(