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

Angular velocities + accelerations

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

\[ \text{Acceleration} = \alpha = \frac{\Delta \omega}{\Delta t} \]

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

In repetitive movements → frequencies will be at harmonics of stride frequency
- Most of the data is below 6 Hz

Noise tends to be random ⇒ tends to be high frequency ⇒ show winter text
For biomechanics, it is common to use low pass filters, such as the Butterworth filter, but be careful, as it may be activity dependent. The Butterworth filter is not as good for step/impulse inputs.

- Show some videos

Most choose an appropriate cut-off frequency.

- Many options
- We will use residual analysis

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

\[
\text{all data: } \sum_{i=1}^{N} (x_i - \bar{x_i}) \\
\text{avoid mean = 0: } R(f_c) = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \bar{x_i})^2}{\sum_{i=1}^{N} (x_i - \bar{x})^2}}
\]

Repeat for different \( f_c \).

**If all noise**

Signal + noise \( \Rightarrow \) residual increases as \( f_c \downarrow \).

Signal + noise \( \Rightarrow \) noise passed through filter.

For distortion = 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

\[ [b, a] = \text{butter}(n, \text{wn}); \]
\[ y = \text{filtfilt}(b, a, x); \]

\[ a, b = \text{coefficients for Butterworth filter} \]

\[ n = n^{\text{th}} \text{ order filter}, \]

\[ \text{wn} = \text{cutoff freq must be btw. } 0.0 < \text{wn} < 1.0, \text{ where } 1.0 \text{ corresponds to } \frac{1}{2} \text{ sampling rate} \]

\[ \text{filtfilt} = \text{gives zero-lag recursive filtering} \]

\[ y = \text{filtered version of signal x} \]
Example for data collected at 100 Hz:

% filter the data

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

% use zero-phase forward-backward filtering

\[
[b,a]=\text{butter}(n,wn);
\]

\[
y=\text{filtfilt}(b,a,x);
\]

\[
\text{wn} = \text{cutoff freq must be btw. } 0.0 < \text{wn} < 1.0, \text{ where } 1.0 \text{ corresponds to } \frac{1}{2} \text{ sampling rate}
\]

What should be the values for \( n \) and \( wn \)?

\[
[b,a]=\text{butter}(4,0.12);
\]
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

\[ \text{diff} = [x_n - x_{n-1}] \text{ vector} \]

\( \Delta t \rightarrow \text{constant} \)
Backward Difference Method

Just like forward, but backward

NO data

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

error \sim (\Delta t)
Three-Point Formula (Centered-Difference Formula)

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

- Average of forward and backward

\[
\text{error} \sim \Delta t^2
\]

\[
\text{comp. more expensive}
\]