

Breathing From ACC Formula Sheet

PolarH10 repository-specific breathing-volume approximation

Provenance

This breathing-volume model is repository-specific code ported from the existing Unity runtime. It is not presented as a published external method reference.

- This ACC breathing-volume model has not yet been externally validated.
- A validation study is actively being worked on.

Data path

- input stream: PMD accelerometer samples from the Polar H10
- preprocessing: exponential moving average on the raw ACC signal
- calibration: estimate a center point and a principal breathing axis from a calibration window
- projection: map filtered ACC onto the chosen breathing axis
- bounds: derive lower and upper projection bounds from calibration quantiles
- output: convert the projected signal into a 0..1 breathing-volume estimate

In the notation below, $a_{raw}(t)$ is the raw accelerometer vector at time t , $a_f(t)$ is the filtered vector, α is the smoothing factor, c is the calibration center, u is the dominant breathing axis, $p(t)$ is the scalar projection onto that axis, and Q_{lower}/Q_{upper} are the calibration quantile operators used to define a practical operating range. The tracker works in accelerometer coordinates internally and only turns the signal into a single breathing scalar after the projection step.

Core approximation

Filtered ACC sample:

$$a_f(t) = (1 - \alpha)a_f(t - 1) + \alpha a_{raw}(t)$$

This smooths the raw accelerometer stream so the later breathing estimate is less sensitive to frame-level jitter.

Calibration center:

$$c = \frac{1}{N} \sum_{i=1}^N a_f(i)$$

This is the average filtered posture during calibration and acts as the reference point for later motion.

Principal-axis projection:

$$p(t) = \langle a_f(t) - c, u \rangle$$

where u is the dominant calibration axis estimated from the calibration-window covariance.

This projection turns 3-D chest motion into one signed scalar that should rise and fall mainly along the learned breathing direction.

If the optional XZ model is available, the tracker also keeps a 2-D projection in the chest plane:

$$p_{xz}(t) = \langle (x(t), z(t)) - c_{xz}, u_{xz} \rangle$$

This is the same idea as $p(t)$, but constrained to the chest plane for the optional XZ model.

Lower and upper projection bounds come from calibration quantiles:

$$b_{min} = Q_{lower}(p), \quad b_{max} = Q_{upper}(p)$$

These bounds define the expected low and high ends of the breathing excursion without trusting a single noisy minimum or maximum sample.

The runtime then applies the configured edge-ease and optional adaptive-bounds updates before mapping the projection to volume:

$$v(t) = \text{clamp} \left(\frac{p(t) - b_{min}}{b_{max} - b_{min}}, 0, 1 \right)$$

This maps the projected motion into a normalized 0..1 breathing volume, where values near 0 sit near the lower calibration bound and values near 1 sit near the upper calibration bound.

The final base volume uses either the 3-D axis model or the XZ model, depending on the tracker settings. The displayed output may then be inverted if the configured inhale/exhale direction is flipped.

State classification

The tracker derives **Inhaling**, **Exhaling**, and **Pausing** from the first difference of the output volume:

$$\Delta v(t) = v(t) - v(t - 1)$$

Positive Δv means the estimated volume is rising, negative Δv means it is falling, and values near zero mean the tracker sees little movement in either direction.

- **Inhaling** when Δv exceeds the configured positive threshold
- **Exhaling** when Δv is below the negative threshold
- **Pausing** otherwise

Alignment status

There is no external paper claim here. The correct interpretation is:

- this sheet documents what the app computes
- the model is intended as a practical ACC-only breathing approximation
- external validation is still pending

References

- <https://mesmerprism.github.io/PolarH10/reference/breathing-workflow.html>
- <https://mesmerprism.github.io/PolarH10/reference/references.html>