



BIKE IQ TECHNICAL WHITEPAPER · V1.0 · APRIL 2026

# **Bike IQ — Gear Detection Whitepaper**

## **Knowing What Gear You Are In**

How Bike IQ Detects the Rider's Current Gear Without an Electronic Drivetrain

A BIKE IQ TECHNICAL WHITEPAPER

## Summary

Bike IQ can tell a rider which gear they are currently in, in real time, on a bike that has no electronic shifting and no gear sensor of any kind. The system uses the rider's speed, the cadence Bike IQ already estimates from phone motion, and the rider's own bike configuration (chainrings, cassette, wheel size). Out of those three inputs and some careful logic, it produces a gear estimate that is correct on the large majority of well-instrumented rides.

When the underlying signals are clean, the system reports the exact gear (for example, "52 by 17") with high confidence. When the signals are noisier, it falls back to a "gear zone" indication (easy, middle, hard) rather than show a number it cannot stand behind. This document explains the problem, the approach, what it does well, and what it does not.

---

## Why This Matters

Gear selection is one of the most consequential decisions a cyclist makes, and one of the least examined after the fact. A coach reviewing a workout can tell, from power and cadence alone, that a climb went badly. They cannot tell, without a gear feed, whether the rider was in the wrong gear for the gradient. A rider analyzing their own ride cannot tell where they ran out of cassette on a steep section, or where they were spinning out on a descent.

Most cyclists never see this data. Electronic drivetrains from SRAM and Shimano can broadcast gear position over Bluetooth, but they are expensive and uncommon. The vast majority of bikes on the road today have mechanical drivetrains and provide no electronic feedback at all. A gear estimate that works on any bike, mechanical or electronic, opens up an analysis dimension that has historically been the exclusive domain of high-end racing setups.

---

## Why It Is Hard

Estimating gear from indirect signals sounds simple. Physics tells you the relationship: at a given speed, the wheel turns at a known rate, and the cadence determines how many wheel revolutions correspond to one pedal revolution. Divide one by the other and you have the gear ratio. Match that ratio to the rider's drivetrain and you have the gear.

In practice, several things make this difficult.

**Many gear combinations produce nearly the same ratio.** A modern road bike with a 50/34 chainring and an 11-speed cassette has 22 possible gear combinations, but only 17 or so distinct ratios, and several of those are within a few percent of each other. The window of error required to distinguish neighboring gears is small.

**Cadence has to be right first.** The whole calculation rides on the cadence estimate. A 5 RPM cadence error at 30 kilometers per hour translates directly into picking the wrong cog. Any error in the front of the pipeline propagates straight into the gear output.

**The rider's bike configuration has to be right.** If the rider configured their tire size incorrectly, every wheel revolution computation is off by the same percentage, and every ratio shifts. The tires actually rolling on the road also vary with pressure and wear, so even a correct nominal tire size has some real-world drift.

**Shifts happen quickly and unpredictably.** A real cyclist can drop two cogs and a chainring in less than a second. A gear detector that smooths heavily to suppress noise will lag the actual shift by seconds, which is exactly the moment a rider most wants to know what gear they are in.

**Two-chainring drivetrains have an ambiguity problem.** On a 2x drivetrain, several rear cog and front chainring combinations can produce ratios that overlap. Without independent evidence of which front chainring is engaged, the system has to use temporal continuity (the rider was on the big ring a moment ago, and front shifts are rarer than rear shifts) to break ties.

---

## Our Approach

Bike IQ's gear detection system is built on top of the cadence estimator described in our cadence whitepaper. It adds a gear inference layer that takes the cadence estimate, the GPS speed, and the rider's drivetrain configuration, and produces a most-likely gear at every moment of the ride.

## **Ratio Computation**

At any moment, the wheel rotation rate is determined by the bike's speed and the configured tire size. Combined with the estimated cadence, this gives the effective gear ratio that the rider must currently be in. This is straightforward physics, not a guess.

## **Drivetrain Matching**

The computed ratio is matched against the actual ratios available on the rider's specific bike. Bike IQ stores each bike's drivetrain configuration (chainring sizes and cassette cog sizes) so the matching is exact rather than approximate. The closest gear within a small tolerance window wins.

## **Temporal Continuity**

A naive ratio match flips between neighboring gears every time the cadence estimate jitters by a fraction of a pedal stroke. Bike IQ's gear detector treats the gear sequence as a temporal trajectory, not an independent decision per moment. Holding a gear is cheap; shifting is expensive; shifting the front chainring is more expensive than shifting the rear. This temporal cost structure suppresses spurious switches without lagging real ones.

## **Front Chainring Logic**

For 2x drivetrains, the front chainring is treated as a separate state with its own switching cost. When the computed ratio sits in a region that only one chainring can produce (because the other chainring's range does not reach there), the front chainring is forced to that value. When the ratio sits in the overlap region where either chainring could be in use, the system stays on the chainring it was already on, unless several consecutive readings convincingly disagree.

## **Confidence-Aware Output**

Not every moment produces a confident gear estimate. When the cadence input is noisy, the speed signal is glitching during a GPS dropout, or the rider is just starting the ride and personalization has not yet warmed up, the gear detector knows it is uncertain. In those moments it can either suppress output or fall back to a coarser "gear zone" indication (easy, middle, hard) rather than display a precise number it cannot defend.

## **Self-Calibration of Wheel Size**

Small errors in the rider's configured tire size shift every ratio by a constant percentage. The same self-calibration mechanism that powers the cadence estimator also corrects this here. After roughly

half a minute of clean pedaling, the system has converged on the rider's true effective wheel circumference and gear matches become noticeably tighter.

---

## How We Validated It

Validation for gear detection is more difficult than for cadence, because most bikes do not have a gear feed of any kind. Our validation set comes from research rides on bikes equipped with Bluetooth-broadcasting electronic drivetrains, where the true gear position from the drivetrain is recorded alongside the phone's motion data. Each second of the ride has a true gear (from the drivetrain) and an estimated gear (from Bike IQ) for direct comparison.

The current validation set spans multiple riders, multiple bike configurations from 1x setups with wide-range cassettes to 2x road racing setups, and rides under varied conditions including climbing, descending, urban traffic, and steady road riding.

The headline finding: when Bike IQ's cadence estimate is accurate (which it is on the large majority of rides), gear detection accuracy is around 90%. On rides where the cadence estimate degrades, gear accuracy degrades with it in a predictable way. The gear estimator is essentially a cadence-quality amplifier. Improvements to cadence accuracy translate directly into improvements here.

---

## What It Does Not Do Well Yet

**Mounting position matters more here than for any other metric.** Because gear detection sits on top of cadence, anything that hurts cadence accuracy hurts gear accuracy more. Riders whose phones produce a weak motion signal (unusual mounts, very smooth pedaling styles, extremely soft phone cases that damp the signal) see degraded gear detection.

**Front chainring transitions can briefly slip.** Front shifts are rarer than rear shifts, and the system is biased toward not switching the front chainring without strong evidence. On a real shift, the system catches up within a few seconds, but during that window the gear can be reported with the wrong chainring.

**Configured tire size has to be in the ballpark.** The self-calibration handles small errors, but a wildly incorrect tire size (for example, the wrong wheel diameter entirely) is outside the calibration's range. Riders should set the tire size correctly when they configure the bike.

**Very low cadences are harder.** The same low-cadence weakness in the cadence estimator surfaces here, and ratio matching with an off-by-half-cadence input picks the wrong gear consistently.

**Single-speed and fixed-gear bikes are not the target.** They have one gear, so detection is trivial but uninteresting. Internally geared hubs (Rohloff, Alfine) require a different configuration model that Bike IQ does not yet support.

---

## The Roadmap

**Better confidence display.** The system already knows when it is uncertain. The next step is exposing that uncertainty to the rider in a useful way, so that the on-screen gear is always either confident and exact or honestly approximate.

**Gear-zone fallback in the UI.** When confidence is low, the app should show “easy gear / middle gear / hard gear” rather than a number. This is a UX-side change rather than an algorithm change. It allows the metric to remain useful in conditions that defeat exact identification.

**Cross-ride personalization.** The wheel circumference calibration restarts every ride. Persisting it across rides, like the cadence range personalization, will make the first minute of every ride as accurate as the rest.

**Improvements that flow from cadence.** Every roadmap item from the cadence whitepaper, particularly the low-cadence robustness work, lifts gear accuracy along with it. Most of the gear team's attention is therefore on cadence rather than on gear-specific algorithms.

**Internally geared hubs.** A modest configuration change to support hub gears (Rohloff, Shimano Alfine, Sturmey-Archer) is on the backlog for riders on commuters and touring bikes.

**Shift-event detection.** Detecting the shift event itself, distinct from steady-state gear identification, is an interesting open problem. The signature of a chain moving across cogs may be visible directly in the motion signal. Early experiments are encouraging but not yet productized.

---

## What This Means for Riders

A cyclist on any bike, mechanical or electronic, gets a real-time gear readout in Bike IQ during the ride and a gear track in the post-ride analysis. For riders without an electronic groupset, this is information that has not previously been available to them at any price. For riders with electronic drivetrains, it is a complementary readout that does not require pairing, charging, or installing anything.

The system is honest about when it does not know. When the underlying signals support a precise gear, that is what the rider sees. When they do not, the rider sees an honest approximation rather than a confident wrong answer.

Gear detection is a downstream consequence of sensor-free cadence. Other metrics in Bike IQ chain off the same foundation in similar ways.

---

## About Bike IQ

Bike IQ is an iOS cycling app focused on turning the phone into a legitimate cycling computer for riders who want serious training metrics without buying serious training hardware. The app is built by a small independent team and ships regular research-driven updates to its beta and production users.

For press inquiries or technical follow-up, please contact the Bike IQ team at [contact@bikeiq.app](mailto:contact@bikeiq.app).

---

*This whitepaper describes research in active development. The systems and methods described herein are the subject of pending U.S. patent application(s), including U.S. Provisional Application No. 64/040,848 filed April 16, 2026. Numbers reported above reflect the state of the system as of the document date and may improve in subsequent releases.*