

Mathematical Data Science

Data Intensive Final Project

Faculty Advisor: Professor Peter Mucha

Digital Twins for Athletic Performance

A Statistical Integrative Physiology Approach to an Augmented Coupled Tank Model for
Human Metabolism, Physiology, and Fatigue in Endurance Sports

John DeForest

Abstract

This project develops and validates an individualized physiological digital twin model for endurance performance using test data from 53 runners. Extending Boillet's (2024) dynamic three-tank energy model and Lidar's (2023) metabolic breakdown, we simulate VO_2 kinetics, lactate accumulation, ventilatory load, and energy substrate use in response to graded treadmill protocols. Ventilatory and lactate thresholds are detected algorithmically, and model parameters such as efficiency (η), oxygen energy yield (C1), and anaerobic work capacity (W' non-ox) are inferred per subject. A novel correction method addresses systematic bias in raw $\text{RER} = \text{VCO}_2/\text{VO}_2$ data using global scaling constrained by plausible RER ranges. Model outputs match measured VO_2 trends with strong fidelity (mean RMSE = 0.365 L/min, MAPE = 11.9%) despite noisy data, and extend to estimate blood lactate, fatigue state, and metabolic demand composition under arbitrary workloads. The resulting digital twins not only replicate classical test behavior (e.g., 3-min all-out, threshold trials) but enable constant-load simulations and physiological inference in silico. This modeling framework provides a foundation for robust, data-grounded estimation of endurance potential and mechanistic understanding of performance limitation.

Introduction

Modeling human endurance performance requires integrating diverse physiological subsystems—oxygen uptake, energy production, and fatigue dynamics—into a coherent, mechanistically grounded framework (Sreedhara et al., 2019). While traditional exercise science research often isolates individual variables, pathways, or associations, recent advances in digital twin modeling aim to synthesize this fragmented knowledge into unified, predictive systems. This project extends such efforts by recreating and augmenting the tank-based energetic model introduced by Boillet et al. (2024), incorporating additional insights from Lidar et al. (2023), and adapting them to running-based exercise test data.

The core objective of this project is to develop a personalized simulation framework that estimates and predicts an athlete's physiological state and performance potential across a wide spectrum of exercise intensities. This begins with the reconstruction of Boillet's differential equation model, which captures the interaction of aerobic and anaerobic energy systems through a system of biophysically interpretable compartments. The framework is extended to incorporate additional physiological mechanisms, including dynamic blood lactate behavior, variable oxygen-to-energy efficiency ($C1$), and ventilatory energy cost modeling.

The resulting model is applied to treadmill-based threshold and VO_2 max tests from a cohort of 53 runners with diverse performance profiles. To support data fidelity and inference accuracy, automated procedures are implemented for ventilatory threshold detection and for correcting known sources of bias in raw metabolic cart data. The digital twin framework is used to simulate not only standard physiological testing protocols but also novel scenarios, such as constant-load time-to-fatigue estimates and muscle-to-blood lactate lag effects. In doing so, the model connects local muscular metabolism to whole-body oxygen and lactate behavior, enabling parameter inference for individuals from sparse test data.

The results demonstrate strong alignment between simulated and measured values for key outputs (e.g., VO_2 , metabolic power, RER, blood lactate), and provide interpretable parameter estimates (e.g., W' , η) with physiological meaning. This integrated modeling approach lays the groundwork for personalized diagnostics, endurance prediction, and future machine learning-based parameter estimation from partial test data.

Background

Human endurance performance is governed by a complex interplay of aerobic and anaerobic energy systems, muscle fiber characteristics, and metabolic feedback processes (Lidar et al., 2023). During prolonged or high-intensity exercise, the body must dynamically regulate oxygen uptake (VO_2), manage metabolite accumulation (e.g., lactate), and allocate energy across multiple physiological pathways to sustain effort (Van Der Zwaard et al., 2021). Traditional exercise science has characterized these processes via threshold testing (e.g., lactate or ventilatory turnpoints from incremental stages), isolated regression studies, and other focused but uncomprehensive analyses. However, such approaches often fail to capture the full time-dependent, nonlinear dynamics of performance or connect local muscle metabolism to system-wide outputs (Sreedhara, 2019).

Physiological Modeling of Energy Systems

A central concept in endurance modeling is that energy supply is partitioned among three major pathways:

- Oxidative (aerobic) system: Supplies the majority of energy in steady-state exercise. Characterized by oxygen uptake and efficient but slow energy release.
- Glycolytic (anaerobic) system: Breaks down carbohydrates without oxygen, producing energy quickly but also lactate.
- Phosphocreatine (PCr) system: Provides rapid energy for short bursts (seconds), but is quickly depleted.

Building on previous work by Margaria and Morton (Morton, 1986; Morton et al., 1990), Boillet et al. (2024) formalized these systems as a three-tank model, where each "tank" represents an energy reservoir (aerobic, glycolytic, phosphagen), and power output is governed by flow between them based on tank "heights" (pressures). The model is built as a system of coupled differential equations where energy flow constraints and physiological parameters control response behavior. Crucially, the Boillet model reproduces several hallmark features of exercise physiology, including threshold behaviors, VO_2 kinetics, and fatigue onset, while remaining interpretable and grounded in physical analogies (Gonzalez et al., 2019). This model will be explained in mathematical detail in the Methods section.

Understanding endurance energetics has importance beyond sport -- Aerobic metabolism underpins the energy production needed for life itself. Mitochondrial dysfunction—the

failure of aerobic energy systems—is implicated in diseases like Type 2 diabetes, cancer, cardiovascular disease, and Alzheimer’s (San Millán, 2023). Thus, endurance modeling can contribute both to performance optimization and broader biomedical understanding of how the body responds to energetic demands.

VO₂ Kinetics and Fatigue

VO₂ kinetics—the rate at which oxygen uptake adjusts to changes in exercise intensity—has been modeled since at least Hughson (1988) using exponential or multi-compartment dynamic models. At intensities below the first lactate threshold (LT1), VO₂ tends to reach steady state rapidly (Barstow, 2000). At higher intensities, particularly above the second threshold (LT2), VO₂ continues to rise ("drift") and can indicate rising fatigue (Saunders et al., 2000). Endurance athletes have been long tested for and characterized by their “VO₂max” – the maximal amount of oxygen their body can use during exercise (Seiler, 2010). Many different multi-component saturating exponential models have been constructed and proposed, which tend to try to best describe the following behavior drawn out by Jones et al. (2011):

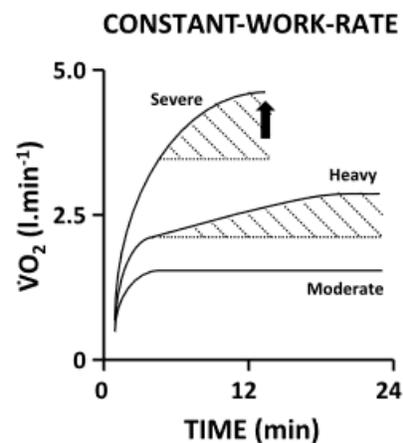


Figure 1: Conceptual model of VO₂ responses to constant work rate (speed or power) demand at different intensities, showing time constant variations and other observed and hypothesized effects. Taken from Jones, et al. (2011)

Fatigue arises from both local (muscular) and systemic causes, including metabolite buildup (especially lactate and associated ions), depletion of energy stores, and increased ventilatory and metabolic demand (Zuccarelli et al., 2018). Understanding the interplay of these factors, how they vary across individuals, and how they reveal themselves in ventilatory data is key to modeling endurance performance.

Lactate, Thresholds, and the Physiology of Exercise Intensity

Exercise intensity in endurance physiology is often stratified into distinct zones based on underlying metabolic transitions. These zones are defined by measurable breakpoints—called thresholds—which signal shifts in the dominant energetic systems. Two such thresholds are particularly important in modeling: the first lactate turnpoint (also called the first lactate threshold) (LT1) and the second lactate turnpoint (LT2). These correspond approximately to the onset of non-zero lactate accumulation above resting levels and the limit of sustainable lactate clearance, respectively (Caen et al., 2022; Stegmann et al., 1981).

Why Lactate?

As detailed by Van Der Zwaard et al. (2021), Lactate is a byproduct of anaerobic glycolysis, the metabolic pathway by which glucose is rapidly broken down in the absence of sufficient oxygen, primarily in Type II (fast-twitch) muscle fibers. When exercise intensity increases beyond what aerobic metabolism alone can support, anaerobic energy production is recruited—yielding lactate as a measurable consequence. Thus, lactate concentration functions as an indirect biomarker of anaerobic system engagement.

However, lactate is not merely a waste product. It is increasingly recognized as a fuel, signaling molecule, and buffer substrate, shuttled between muscle fibers and metabolically active tissues (Bartoloni et al., 2024). It is correlated with—but not causative of—fatigue. Still, because its concentration reflects an accumulation of fatigue-inducing conditions (e.g., hydrogen ion concentration, reduced pH, metabolite accumulation), lactate remains one of the most practical and interpretable indicators of metabolic stress and threshold crossing (Fischer et al., 2025; Poole et al., 2021).

Why Ventilatory Behavior?

While lactate is the most direct biochemical marker of metabolic threshold, it requires invasive or semi-invasive sampling. In contrast, ventilatory markers—changes in oxygen uptake (VO_2), carbon dioxide output (VCO_2), and minute ventilation (VE)—can be continuously measured using a metabolic cart (and facemask-tube apparatus). These variables respond to the increased buffering demand for acid byproducts of anaerobic metabolism, especially above LT2. As lactate accumulates, hydrogen ions (H^+) are buffered by bicarbonate (HCO_3^-), producing additional CO_2 (Green et al., 1983). This leads to a secondary rise in VE and VCO_2 , enabling the detection of ventilatory thresholds (VT1, VT2) that often correspond closely with LT1 and LT2 (Cerezuela-Espejo et al., 2019).

Therefore, ventilatory breakpoints are used as non-invasive surrogates for lactate thresholds, especially in modeling applications or real-time assessment, and both types of

thresholds are considered interchangeable to a degree, depending on the protocol and sensitivity of measurements.

Threshold Definitions and Their Modeling Relevance

- Lactate Threshold 1 (LT1):** The first inflection point where blood lactate concentration begins rising significantly above resting levels, typically around 2 mmol/L. Below this point, energy is almost entirely produced aerobically by Type I (slow-twitch) muscle fibers, with high efficiency and minimal fatigue. Fat oxidation is maximal near here. VO_2 increases linearly with power or speed, and the body operates in a true steady state (Morton et al., 1990).

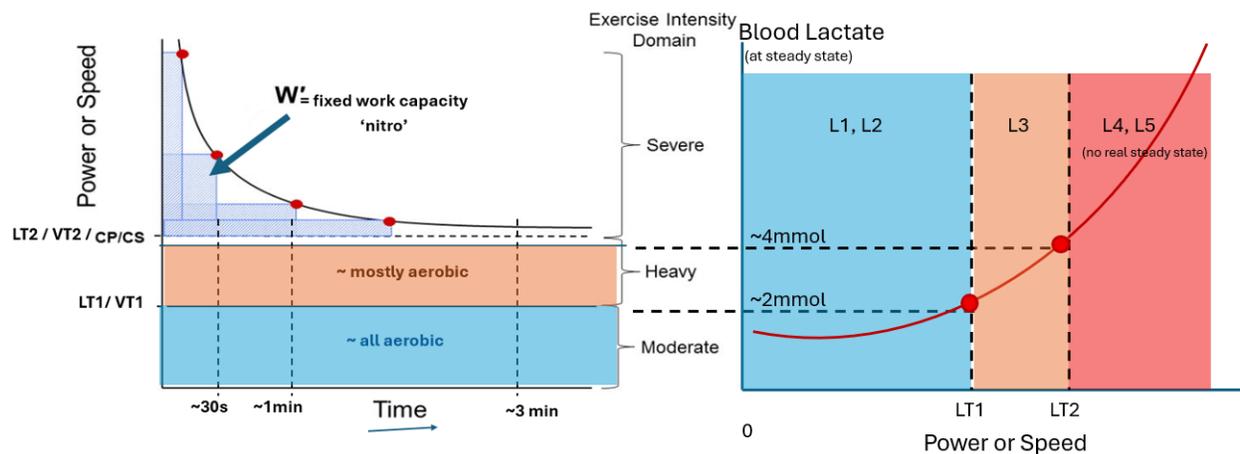


Figure 2: Conceptual lactate curve model (right) and corresponding critical power model (left), modified from Vanhatalo, 2016.

- Lactate Threshold 2 (LT2):** The second breakpoint, where lactate production exceeds clearance, typically around 4 mmol/L (though this varies). Exercise above this point enters the severe domain: VO_2 rises continuously (slow component), lactate accumulates rapidly, and fatigue onset is imminent. LT2 approximates Critical Power (CP), Ventilatory Threshold 2 (VT2), or Maximum Lactate Steady State (MLSS) depending on the testing method (Poole et al., 2021; Caen, 2022; 2024).

LT2 is closely connected to the Critical Power (CP) model -- Exercise below CP can be sustained aerobically, with manageable lactate levels and VO_2 kinetics reaching equilibrium (Chorley, 2020). Above CP, however, energy demand exceeds aerobic supply, leading to progressive lactate accumulation, VO_2 drift, and eventual exhaustion (Sreedhara 2019).

Mathematically, CP defines the asymptotic power limit in the two-parameter hyperbolic model of work and time to exhaustion:

$$W = CP \cdot t + W'$$

Where W is total work performed, CP is the critical power (in watts), and W' is a finite work capacity (in joules) representing the energy available above CP . The power intercept of the linear fit of constant-power exercise trials to exhaustion versus $1/\text{Time}$ is often used to find CP , but a 3-minute All-out test is also frequently utilized (where the subject is instructed to do maximal effort at every given second, rather than holding a stated power) (Vanhatalo, 2016; Caen, 2024).

In well-controlled graded exercise tests, CP corresponds closely to LT_2 and Ventilatory Threshold 2 (VT_2) as well as an RER near ~ 1.0 , and can therefore be used interchangeably for modeling purposes in this project (Caen et al., 2024; Korkmaz Eryilmaz & Polat, 2021). Specifically, CP serves as a reference point for estimating anaerobic system size, particularly the G -tank capacity (AG) in the Boillet model. The depletion of W' during high-intensity exercise directly informs how quickly the anaerobic reserve is used, and thus how the G tank drains under suprathreshold efforts.

While traditional CP models are limited in handling recovery and dynamic pacing, they remain physiologically interpretable markers of boundary conditions for fatigue onset, and provide critical anchors for calibrating dynamic simulation models like the one developed here (Sreedhara, 2019).

- **Zone Between LT_1 and LT_2 :** This intermediate zone (Zone 2/3 in some systems) reflects semi-steady conditions. Lactate accumulation occurs slowly but remains manageable. Both aerobic and anaerobic systems are active, and fatigue is a function of exercise duration and individual clearance capacity.

Blood vs. Muscle Lactate: Modeling Implications

Though muscle lactate is the true origin of production, it is difficult to measure directly (Van De Casteele et al., 2024). Instead, blood lactate, sampled via capillary collection, is used as a proxy. However, the kinetics of transport from muscle to blood involve delays and nonlinear saturation, influenced by perfusion, diffusion, transporter proteins (e.g., $MCT-1/4$), and metabolic reuse of lactate as fuel (Stegmann, 1981).

These considerations are critical in digital twin modeling, where muscle lactate is a state variable and blood lactate is a delayed, filtered observation used for validation. Without

accounting for these lags, any attempt to infer fatigue or threshold crossing from measured lactate would risk systematic error.

Recent research has shown that lactate itself is not necessarily what causes fatigue and failure, but rather a highly-correlated byproduct, fuel, and signaling molecule that can be used to infer the fatigue inducing accumulated metabolites and conditions that *do* induce fatigue. However, lactate is produced and cleared dynamically, and the relationship between muscle and blood lactate is nonlinear and time-lagged (Van Der Zwaard, 2021).

While direct muscle lactate measurement requires invasive biopsy, blood lactate can be sampled from finger or earlobe capillaries (Poole et al., 2021). Models that connect these compartments—such as first-order saturation or lagged regression—allow for indirect estimation of muscular states from more easily accessible data.

Digital Twin Modeling

A digital twin in this context refers to a personalized physiological model that can simulate how a specific individual's body would respond to various exercise intensities. Such a model requires identifying subject-specific parameters (e.g., $VO_2\text{max}$, muscle mass, mechanical conversion efficiency η , W' for anaerobic work capacity) from experimental data. Once fit, the model can be used to explore hypothetical training scenarios, simulate time-to-fatigue, or estimate unmeasured physiological states.

This thesis extends the Boillet framework by incorporating additional physiological mechanisms that enhance both interpretability and predictive realism. First, explicit modeling of blood lactate is introduced, drawing on empirical muscle–blood relationships from prior literature (e.g., Moneta, 1989) and adapting these dynamics to the tank-based structure of the original model.

Second, mechanical efficiency (η) and oxygen-metabolic conversion efficiency (C_1) are allowed to vary as functions of speed and respiratory exchange ratio (RER), moving beyond the limitations of fixed-parameter assumptions and capturing known nonlinearities in substrate utilization.

Finally, the simulation integrates conceptual and empirical advances from Lidar et al. (2023), who decomposed metabolic power into aerobic demand, ventilatory work, and metabolite clearance using synchronized VO_2 and VCO_2 data. Together, these enhancements allow the model to reflect more detailed internal physiology while remaining compatible with sparse, real-world test data.

Measurement and Data Challenges

Ventilatory and metabolic data—including VO_2 , VCO_2 , and respiratory exchange ratio (RER)—are typically collected using metabolic carts such as the Parvo TrueOne 2400 during graded treadmill step tests. However, in practice, such data often exhibit measurement noise, calibration drift, and missing values that can obscure underlying physiological patterns (Crouter et al., 2006; Lii et al., 2024).

To address these limitations, the present framework implements a global correction scheme for miscalibrated VO_2 and VCO_2 values, using constrained optimization to align observed RER trajectories with physiologically plausible bounds. In addition, ventilatory thresholds are automatically detected from raw time-series data by analyzing inflection points in VE/VO_2 , VE/VCO_2 , and RER traces, following established guidelines from prior work (e.g., Cerezuela-Espejo et al., 2018).

To accurately translate treadmill running speed and incline into external workload estimates, the model also integrates grade-adjusted pace (GAP) calculations, allowing biomechanically grounded inference of power demand across all test segments. Together, these innovations enable robust simulation of test data from over 50 runners and allow for meaningful interpretation of individual endurance traits through a computational lens.

Muscle Fiber Typology and Mechanistic Considerations

Although this project centers on modeling running performance from VO_2 , RER, and lactate dynamics, underlying muscle fiber composition remains a critical determinant of endurance capability, anaerobic reserve, and fatigue resistance. Numerous modeling assumptions—including energy system engagement thresholds, time constants of fatigue, and recruitment dynamics—are deeply tied to individual variation in muscle typology.

Conceptual Relevance of Fiber Type

Skeletal muscle fibers are broadly categorized into two primary types with distinct metabolic and functional characteristics. Type I fibers, also known as slow-twitch fibers, are highly oxidative due to their elevated mitochondrial density and rich capillary supply. These fibers are fatigue-resistant and are predominantly recruited during sustained, low-to-moderate intensity exercise (Bex et al., 2017). In contrast, Type II fibers, or fast-twitch fibers, are more glycolytic in nature (Vikne et al., 2012). They are capable of generating high power output over short durations but fatigue more rapidly, in part due to their lower oxidative capacity. These fibers become increasingly recruited as exercise intensity rises,

particularly above the second lactate threshold (LT2), where anaerobic energy systems play a more prominent role (Conde Alonso et al., 2020).

Type I fibers tend to have a smaller cross-sectional area than Type II fibers, but the intra-group (different types, genders, and training histories of athletes) and intra-individual variation tends to render anything but obvious conclusions (olympic 100m sprinters have more fast-twitch composition than an endurance cyclist, for example) feebly supported (Swinnen et al., 2024; Derave, 2024; Nuzzo, 2023).

The size principle of motor unit recruitment implies that Type I fibers tend to be recruited first, with Type II engagement scaling with torque and fatigue (as Type II fibers fatigue, it is suggested that more are recruited to cover the power demand) (Henneman, 1964; Dotan, 2012). This has clear implications for modeling endurance performance: the rate at which anaerobic stores are accessed, lactate is produced, and VO_2 kinetics drift all depend on the magnitude and timing of Type II fiber recruitment (Tesch et al., 1981).

Although muscle fiber type cannot be directly inferred from the VO_2 and lactate data used in this study, existing literature offers valuable insight into its downstream effects on measurable physiological variables. Fast-twitch–dominant athletes tend to exhibit higher peak torque and optimal cadence during maximal cycling efforts, consistent with findings from sprint studies (Wackwitz, 2025; Hautier, 1996).

The torque–cadence relationship itself reflects a tradeoff between gross mechanical efficiency, metabolite accumulation, and oxygen cost, with fiber composition influencing both the slope and intercept of this relationship (Van Vossel, 2023; Van der Zwaard, 2021; Barclay, 2004). In contrast, slow-twitch–dominant athletes typically sustain a higher fraction of their VO_2max at LT2, exhibit a more gradual rise in lactate during graded exercise, and often attain higher absolute VO_2max values (Barstow, 2000; Tesch, 1981).

Further, non-invasive proton magnetic resonance spectroscopy ($^1\text{H-MRS}$) studies have shown that muscle carnosine content correlates strongly with Type II fiber area, particularly in the gastrocnemius and soleus, providing a viable proxy for muscle typology in human subjects (Baguet, 2011). These findings help contextualize observed differences in test responses and suggest potential avenues for inferring fiber composition indirectly through extended physiological modeling.

A series of inter-study regressions was assembled from the literature, linking muscle fiber typology, muscle carnosine concentration, torque–cadence profiles, and maximal power outputs. Although the data were heterogeneous and underpowered for formal inference, the regressions demonstrated consistent physiological directionality. For example, estimated optimal cadence derived from fiber-type–based regression models aligned

closely with empirical values observed in sprint cycling protocols—approximately 121 ± 5 rpm from Baguet et al. (2011) compared to 118 ± 4 rpm reported by Hautier et al. (1996). Similarly, torque–cadence curves display characteristic negative slopes (around -1 Nm per rpm), with systematic shifts reflecting underlying fiber distribution (Wackwitz et al., 2025). These findings suggest potential modeling routes for estimating fiber-type contributions to fatigue dynamics and performance decay. In future work, such relationships could be formalized as priors or physiological constraints—embedding domain knowledge in the form of statistical regularization, or as a latent structure in hierarchical models.

Despite the conceptual appeal of ground-up fiber-informed digital twin modeling, this approach was not pursued as a primary analysis in the present study. First, the running test data lacked direct measurements of relevant indicators such as carnosine content, torque, or sprint-derived peak power. Second, the source studies varied widely in methodology, sample size, and athlete population, introducing inconsistency in how fiber type was quantified—whether by area, percentage, raw count, or mass-normalized estimates. Third, while fiber composition strongly influences contractile and metabolic properties, it is not the sole determinant of functional output -- Type II fibers, in particular, exhibit high adaptability toward oxidative or glycolytic function depending on training history and muscle environment. As noted by Laia et al. (2011), factors such as capillary density and intracellular ion concentrations further modulate fiber-specific performance potential. Finally, the present study focused on running rather than cycling, limiting the direct applicability of torque–cadence analyses and Wingate-derived anaerobic metrics, though both modalities engage large muscle groups under predominantly isometric contraction conditions at submaximal intensities (Swinnen et al., 2024).

Nonetheless, the literature review of typology-informed regressions provides valuable physiological context for several results observed in simulation—such as differences in lactate threshold onset relative to VO_{2max} , or variation in threshold oxygen uptake across subjects. These relationships reinforce the long-term modeling goal of individualizing energy system structure and fatigue profiles based on muscle-level parameters. While the original ambition of this work was to construct a fully bottom-up model, deriving macroscopic outcomes like VO_{2max} and threshold from individual fiber type distributions and cross-sectional areas, the available data proved insufficient to support such a granular approach (Hansen et al., 2002). Instead, this project demonstrates that meaningful physiological insight and simulation accuracy can be achieved through modifications to the Boillet model alone, without requiring direct inference of microscopic muscle properties.

Methods

Literature data were collected from tables and scraped from figures using WebPlotDigitizer where necessary (Rohatgi, 2021). Simulations run in Google Colab and PyCharm.

Running threshold and VO₂max test data were collected from Parvo metabolic cart and accessory sensor data from a treadmill setup (ParvoMedics TrueOne 2400) in a climate-controlled lab.

53 subjects (36 Male / 17 Female) of a range of sport backgrounds, though mostly runners, did both “eco” (threshold) and “max” (VO₂max) testing. Average weight was 71.79 ± 13.38 kg, VO₂max was 62.84 ± 10.09 ml/kg/min, and Age was 31.26 ± 6.08 yrs.

Much of the work done was based around two main interpretations of coupled dynamic ODE models, namely that of Boillet (2024) and Lidar et al. (2023), which are detailed below.

Boillet et al. (2024) 3-Tank Model Recreation

This project implements and extends the 2024 Boillet et al. dynamic three-tank model of endurance energetics. Boillet’s model conceptualizes the human energy system as a hydraulic analog: energy flows from multiple reservoirs, or “tanks”, into mechanical output based on height-driven pressure differentials and constrained flow capacities. The model is governed by nonlinear ordinary differential equations (ODEs) that simulate time-dependent energy dynamics during exercise at varying intensities.

This section details a rederivation and discrete-time implementation of Boillet’s model, and discusses the physiological rationale for each equation and parameter in relation to observable endurance data.

Conceptual Overview: The Three Energy Tanks

While the three energy systems have long been recognized in exercise physiology, the Boillet framework captures their interactions through a set of interpretable differential equations, cast in a hydraulic metaphor. The model treats the oxidative, glycolytic, and phosphagen systems as coupled reservoirs, each governed by depletion dynamics and flow constraints. The oxidative system (O) is modeled as an effectively infinite supply limited only by flow rate, while the glycolytic (G) and phosphagen (P) systems behave as finite tanks with second-order dynamics based on their respective depletion states. These are represented by tank “heights,” denoted $h(t)$ and $l(t)$, which evolve over time in response to energetic demands. Power output is modeled as a composite of flow rates from each system, with physiological-to-mechanical energy conversion modulated by an efficiency factor η . This structure supports dynamic simulation of fatigue, recovery,

and threshold behaviors under varying workloads. The mathematical formulation that follows specifies these interactions in detail.

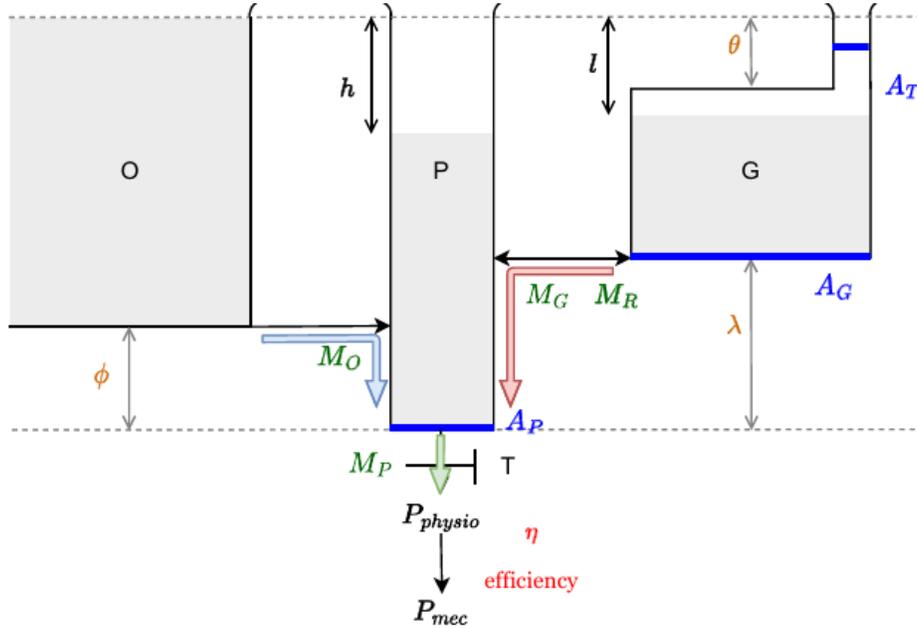


Figure 3: The 3-tank model as defined by Boillet (2024) in Figure 1B. The height of the whole system is 1.

Governing Equations

Boillet's model simulates the total physiological power output as:

$$P_{\text{physio}}(t) = \dot{V}_P(t) + D_{O \rightarrow P}(t) + \dot{V}_G(t)$$

Where:

- $\dot{V}_P(t)$: Flow from phosphagen tank P
- $\dot{V}_G(t)$: Flow from glycolytic tank G
- $D_{O \rightarrow P}(t)$: Flow from oxidative tank O into P (recharging P)

All flows represent energy rates in J/s. This total internal power is what the body must produce to sustain external work. However, the measurable mechanical power output is only a fraction:

$$P_{\text{mec}}(t) = \eta \cdot P_{\text{physio}}(t)$$

Where η is the gross efficiency (typically ~ 0.20 – 0.25 for cycling), the fraction of metabolic power that is converted to mechanical work (Lopez, 2023; Hansen, 2002; Boillet, 2024).

Oxidative and Glycolytic Tank Dynamics

To simulate realistic energy contributions and fatigue, Boillet models the O and G tanks with second-order differential equations, capturing both capacity and rate limitations.

The relative heights of the tank drive the dynamics, assuming hydraulic-analog constant pressure at the top (surface).

- $h(t)$: depletion of phosphocreatine tank P
- $l(t)$: depletion of glycolytic tank G
- θ : G tank depletion threshold separating sub-LT1 power from LT2 dynamics
- λ : Critical value for glycolytic exhaustion (no energy left if l reaches $1 - \lambda$)
- ϕ : related to the rate of Phosphocreatine (P tank) concentration depletion relative to the current aerobic power, and is considered fixed by both Boillet and Behnke (1993).

$$\phi = 1 - (\Delta[PC]_{muscle} m_{muscle} C3) / (A_P * \%VO_2max) = 0.3$$

The key governing dynamics below LT1 (moderate intensity) are:

$$\frac{dl}{dt} = M_G \cdot \frac{h - l}{1 - \lambda} A_T, \quad \text{for } l < \theta$$

This describes the draining of the G tank (lactate accumulation) driven by potential lack of oxidative supply (based on the drain level of the P tank).

$$P_{physio} = M_O \cdot \frac{h}{1 - \phi} + A_P \cdot \frac{dh}{dt} + A_T \cdot \frac{dl}{dt}$$

Which results in an effective second-order system for $l(t)$:

$$a \cdot \frac{d^2l}{dt^2} + b \cdot \frac{dl}{dt} + c \cdot l = P_{physio}(t)$$

With constants:

$$\text{where } a = \frac{A_P A_G}{M_G} (1 - \lambda), \quad b = \left(\frac{M_O (1 - \lambda)}{M_G (1 - \phi)} + 1 \right) A_T + A_P, \quad c = \frac{M_O}{1 - \phi}$$

Where:

- M_O : Aerobic power or transfer rate (kJ/s), which has a maximal value equal to the subject's VO₂max converted via C₁ into kilojoules/s.
- A_T, A_P, A_G : Tank capacities (in kJ)
- M_G : Glycolytic transfer rate (kJ/s)

The model captures both transient kinetics and steady-state behavior. Initial conditions h, l , and dl/dt are all zero. The VO₂ response is then modeled as:

$$\dot{V}_{O_2}(t) = \frac{1}{\eta C_1} \left[P_{\text{physio}}(t) + \frac{M_O}{1 - \phi} \left(K_1 e^{r_1 t} \left(1 + r_1 \frac{A_T}{M_G(1 - \lambda)} \right) + K_2 e^{r_2 t} \left(1 + r_2 \frac{A_T}{M_G(1 - \lambda)} \right) \right) \right]$$

For a constant work demand rate, where r_1 and r_2 are the roots of the characteristic equation of the second order system. The exponential terms arise from solving the second-order ODE. Despite its complexity, at low intensity it behaves almost indistinguishably from a simple single-component saturating exponential for VO₂.

Thresholds and Physiological Zones

Below LT1: The G tank capillary section empties slowly, VO₂ increases linearly with power, and overall power is supported primarily by aerobic energy, fully at steady state.

Between LT1 and LT2: The main G tank begins depletion, lactate accumulates faster, and VO₂ shows a slow component. A_G now governs the equation:

$$\frac{dl}{dt} = M_G \cdot \frac{h - l}{1 - \lambda} A_G, \quad \text{for } l < \theta$$

Above LT2: G drains faster than O can replenish, and exhaustion occurs once thresholds are passed (i.e., $l(t) \geq 1 - \lambda$). The same dynamics apply as for Between LT1 and LT2.

Lactate dynamics are simple $l(t)$ -dependent exponentials as well:

$$[\text{La}]_m(t) = \frac{1}{C_2} \left[1_{l \leq \theta} \cdot l(t) \cdot A_T + 1_{l > \theta} \cdot ((l - \theta) \cdot A_G + \theta \cdot A_T) \right]$$

(using the indicator function **1**)

The maximum sustainable aerobic power (at Critical Power = LT2) is:

$$P_{\text{crit}} = \eta \cdot \frac{M_O(1 - \lambda)}{(1 - \phi) + \frac{M_O}{M_P}(1 - \theta - \lambda)} = \beta * M_O * \eta$$

Where $M_P = P_{max}/\eta = M_O + M_G$, i.e., the maximum physiological power supported by all systems combined. β is the fraction of VO_{2max} used at P_{crit} . Similarly, α , the aerobic power as a fraction of VO_{2max} that is done at LT1 is:

$$\alpha = \theta / (1 - \phi) = VO_{2 \text{ at } LT1} / VO_{2max}$$

$$\text{And } \lambda = 1 - \theta * \left(\left((1/\alpha) - (M_O/M_P) \right) / \left((1/\beta) - (M_O/M_P) \right) \right)$$

Parameter Estimation from Athlete Data

Many of the tank model's parameters map directly to physiological observables – eta is estimated by Boillet from low-intensity trials where all power is assumed supplied by aerobic sources:

$$\eta = P_{mec \text{ at } LT1} / (\%VO_{\{2\}max \text{ at } LT1} * VO_{2max} * C1)$$

Boillet uses anthropometric methods to estimate muscle mass and determine AP and AG. In this project, the process was inverted: starting from observed data, fit energy system parameters assuming a ~50% effective muscle mass of total body mass, given lean leg mass and volume ratios of Hopker et al. (2010).

From literature data, three constants are used by Boillet, for the respective tank conversions to power:

$$C1 = 20.9 \text{ J/ml of oxygen}$$

$$C2 = 100 \text{ J/mmol of accumulated muscle lactate}$$

$$C3 = 43.3 \text{ J/mmol of Phosphocreatine (PC)}$$

The tank widths ('capacities') are estimated as:

$$A_T = [La]_{muscle \text{ at } LT1} / \theta * m_{muscle} * C2$$

And:

$$A_P = [PC]_{muscle, max} * m_{muscle} * C3$$

Which then are paired with 3-min All-out Critical Power test data to find A_G , via the W' estimates of the non-oxidative power supplied (all the power done above CP = LT2):

$$A_G = (W_{\{mec \text{ non-ox}\}} / \eta) - A_P * l_{P_{crit}} - A_T * \theta / (l_{P_{crit}} - \theta)$$

Where, derived from the P_{crit} formula,

$$l_{P_{crit}} = \frac{1 - \lambda}{\left(\frac{M_O}{M_P} \cdot \frac{1 - \theta - \lambda}{1 - \phi}\right) + 1}$$

Discrete-Time Simulation

To implement the model computationally, we discretize the governing ODEs using a forward Euler method with time step Δt for the tank depletion levels. For example:

$$l_{t+1} = l_t + \Delta t \cdot M_G \cdot \frac{h_t - l_t}{1 - \lambda}$$

Higher-order derivatives were implemented as finite differences where needed, and the full state vector was advanced using an explicit integration scheme.

$$\ddot{l}_t = \frac{P_{\text{physio},t} - b_t \cdot \dot{l}_t - c_t \cdot l_t}{a_t}$$

where A is either AT or AG depending on whether l_t is below or above the threshold θ , and the coefficients a_t, b_t, c_t depend on the energy system regime (recruitment or recovery).

The updated tank levels are, from the first derivative of h :

$$\dot{h}_t = \dot{l}_t + \left(\frac{1 - \lambda}{M_G} \cdot A\right) \cdot \ddot{l}_t$$

$$h_{t+1} = h_t + \Delta t \cdot \dot{h}_t$$

$$l_{t+1} = l_t + \Delta t \cdot \dot{l}_t$$

Phosphocreatine concentration is derived from tank height as:

$$[\text{PCr}]_t = \frac{(1 - h_t) \cdot A_P \cdot 1000}{C_3 \cdot m_{\text{muscle}}}$$

Thus the modeled contribution of the P tank to mechanical power (after accounting for previous timestep PCr) is:

$$P_{\text{PCr},t} = ([\text{PCr}]_{t-1} - [\text{PCr}]_t) \cdot m_{\text{muscle}} \cdot C_3 \cdot \eta$$

Anaerobic glycolytic power follows the usage of the G tank:

$$P_{G,t} = A_G \cdot \dot{l}_t \cdot \frac{1000}{C_2}$$

And VO2 (aerobic) power follows from the remainder of the physiological power not covered by the anaerobic system(s):

$$P_{O,t} = P_{\text{physio},t} - P_{G,t} - P_{P,t}$$

While the G tank energy availability is monitored as:

$$E_{G,t} = (l_t > \theta) \cdot [A_G(1 - \theta - \lambda - (l_t - \theta))] + (l_t \leq \theta) \cdot [A_G(1 - \theta - \lambda) + \theta A_T - l_t A_T]$$

Failure was defined by:

- Energetic failure: G tank depletion
- Mechanical failure: Torque (estimated by power vs max power) demand exceeds capacity – often when Type II fibers fatigue
- Metabolic failure: Accumulated lactate > tolerance (e.g., >30 mmol/kg muscle)

Failure modes were not explicitly imposed on the model but rather considered subjectively during model evaluation, especially as lactate at task failure and exhaustion vary considerably (both for blood and muscular) (Tesch et al., 1981; Brooks et al., 2022).

Full code is in the appendix.

Replication of Boillet results

To ensure model validity, simulations were run using Cyclist 1 data from Boillet Table 2 (below), and ground-truthed against Boillet’s example performance figures. Cyclist 1 (and the other subjects in that study) is an elite female cyclist.

3-min All-out replication:

One of the constraints of the Boillet model is the G tank’s determination of the maximum possible power at any given moment in time – as the aerobic tank’s time constant is much too slow to contribute to instantaneous power but rather supports it. This conveniently allows us to impose the maximum power as one would normally do in the 3-minute Critical Power test, and iteratively figure out the athlete’s estimated response and potential power over the duration. Table 1 at right contains the twin-defining parameter values for Cyclist 1, from Boillet.

Table 1: Cyclist 1 Parameters from Boillet (2024)

Parameter	CYCLIST 1
ϕ	0.30
θ	0.43
λ	0.38
M_O (kJ/s)	1.34
M_P (kJ/s)	4.48
M_G (kJ/s)	9.15
M_R (kJ/s)	3.66
A_P (kJ)	27.79
A_G (kJ)	320.3
A_T (kJ)	11.27

Eta η for Cyclist 1 was estimated by taking peak simulated power from Boillet’s 3 minute all-out trial and converting through the stated MP value to get $\eta = 0.2098 W_{(mec)}/W_{(phys)}$

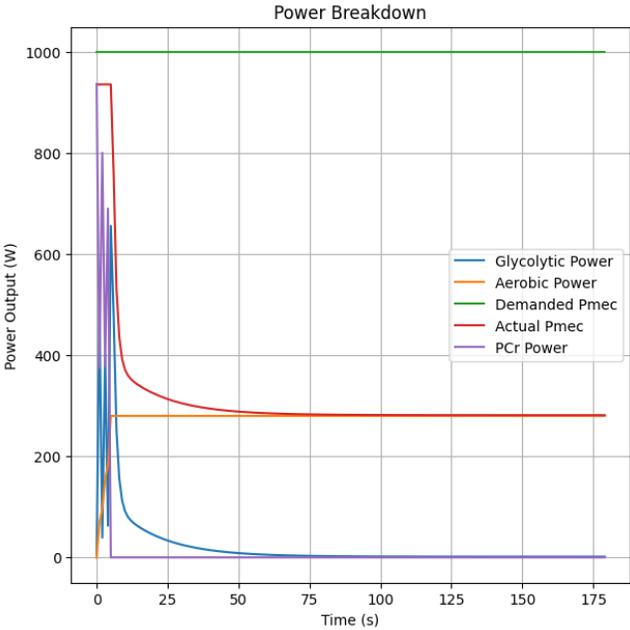


Figure 4: 3-min All-out Critical Power test, replicating Figure 3 of Boillet et al., 2024.

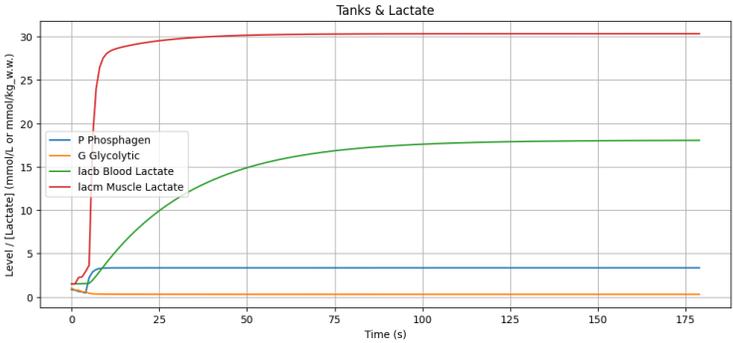


Figure 5: Corresponding lactate and tank utilization curves for Figure 4 CP test.

Note how according to the simulation, by the third minute, nearly all power output is fulfilled by aerobic power, the anaerobic glycogenic and phosphocreatine components all but gone. A good theoretical confirmation of the test validity of Critical Power as the maximum sustainable aerobic power, if executed correctly.

Basic Sub-LT1 Constant Power Replication:

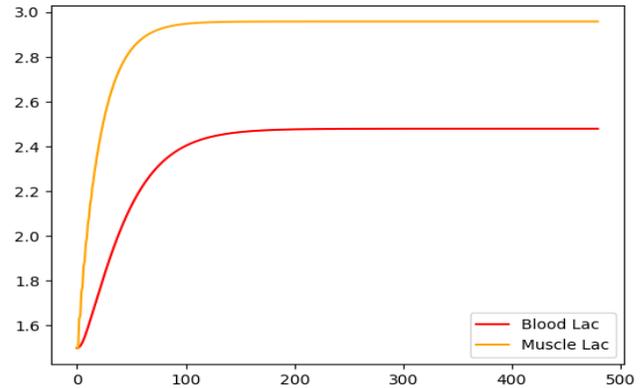
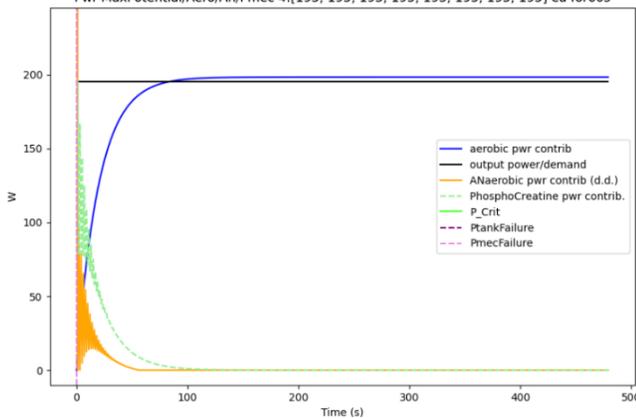


Figure 6A and B: Replication of Boillet Figure 2B: Sub-LT1 constant workload replication. Note the first-order behavior of the tank system's aerobic component (blue), and the stable lactate dynamics (right – orange and yellow).

Between-threshold (LT1, LT2) Constant Power Replication:

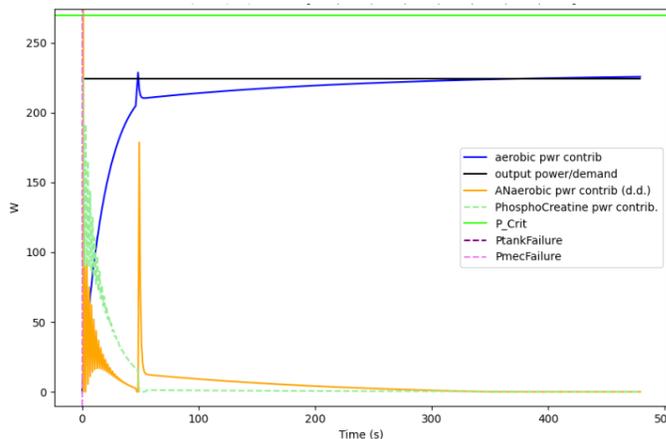


Figure 7: Between-threshold (L3) power replication of Boillet Figure 2D – showing some nonlinearity in the VO₂ and other power supply responses.

Note the non-linear dynamic result in both where the first threshold is crossed – the slow linear drift begins in the (blue) VO₂ response, and here the lactate will not reach an asymptote necessarily. The small blue spike is an artefact of the discrete solver calculating some oscillations in the tank dynamics, and isn't necessarily representative of the physical reality, but does not seem to impact the steady-state results.

Severe Doman >LT2 Replication:

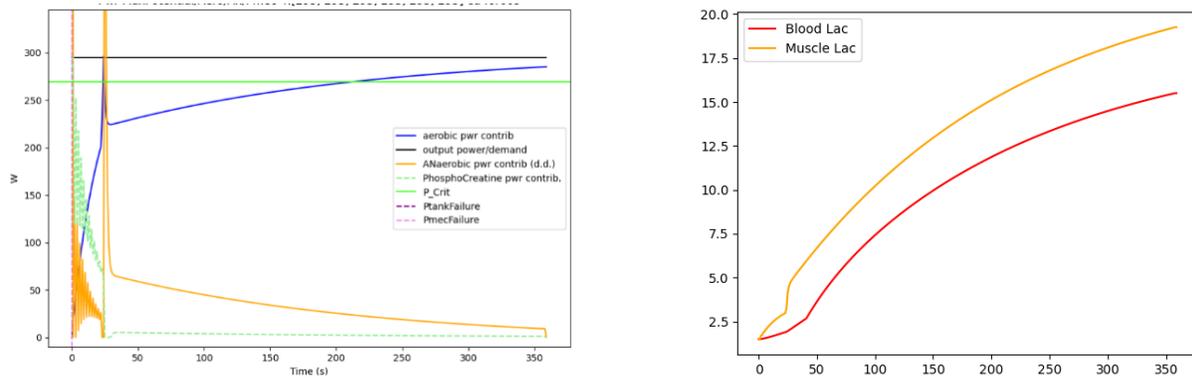


Figure 8A and B: Imposed power on Cyclist 1 above Critical Power – demand is not met by aerobic (blue) and lactate levels rise to levels that certainly indicate task failure.

Here the predicted lactate at time of failure matches that of Boillet 2F, following the same rough curvature that was determined in part from work by Green (1983).

Full VO2max Test Replication:

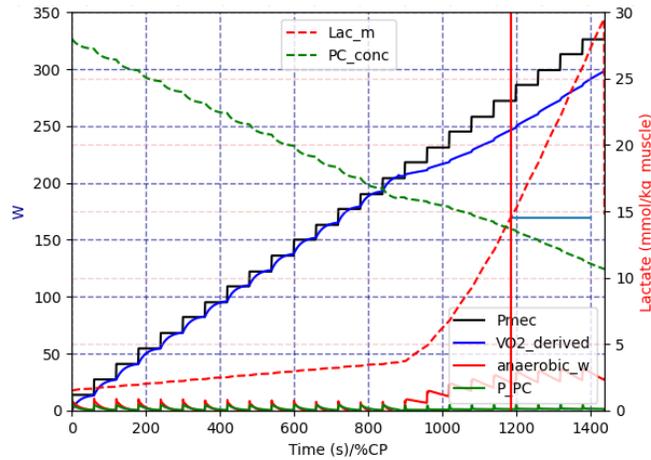


Figure 9: Incremental Step Test of 1-minute stages at increasing power until failure (vertical line) of Cyclist 1, matching that of Boillet figure 5B.

All of these simulation replications are in line with the findings Vanhatalo et al. (2016) of lactate profiles over and under critical power.

Lidar et al. (2023) Metabolic Breakdown Overview

The Boillet model conceptualizes energy supply through a three-tank system—oxidative (O), glycolytic (G), and phosphagen (P)—which dynamically respond to physiological power demands over time. However, while Boillet elegantly captures energy availability and depletion dynamics, it does not explicitly decompose the total physiological power into mechanistically distinct metabolic contributors such as ventilation cost or metabolite accumulation. This decomposition is where the work of Lidar et al. (2023) complements the Boillet framework.

Lidar’s 4-state variable coupled ODE model introduces a modular framework for decomposing total metabolic demand into physiologically meaningful components. First, the fundamental mechanical work component (MR_f) captures the baseline metabolic cost of performing external work, such as sustaining treadmill speed or mechanical cycling power. Second, the ventilatory demand (MR_{ve}) reflects the energetic burden placed on respiratory musculature, which rises with increasing ventilation and is modeled as a nonlinear function of the ratio VE/V_{Emax} . Finally, the accumulated metabolite demand (MR_{acc}) accounts for the additional energy required to buffer and clear byproducts such as lactate and hydrogen ions. This term scales nonlinearly with a proxy variable x_4 , which reflects the proportion of glycolytic energy reserve that has been depleted (or the accumulated muscle lactate). Together, these components sum to produce the total metabolic rate MR_{dem} , allowing the model to attribute energetic cost not only to overt work output but also to internal physiological stress and compensatory processes.

The above combined total metabolic demand in Lidar:

$$MR_{dem} = MR_{rest} + MR_f + MR_{ve} + MR_{acc}$$

And can be mapped to the Boillet model’s physiological power, P_{physio} , offering a more granular interpretation of the energy “sink” fed by the O, G, and P tanks.

Moreover, the aerobic metabolic rate in Lidar:

$$MR_{ae} = MR_{rest} + x_3$$

directly parallels Boillet’s modeled oxygen-derived power contribution at the muscular level, adding in the resting physiological load.

This mapping allows the Lidar formulation to be embedded as a post-processing layer in the digital twin simulations: Boillet’s simulated state variables (e.g. h, l , mechanical power P_{mec} , and VO_2 estimates) are passed into Lidar-style state variables (x_3, x_4) and equations to decompose the total power into constituent metabolic sources.

The hybrid dual-model structure offers several advantages that enhance both the interpretability and practical utility of the model. First, it improves physiological interpretability by explicitly attributing portions of the Boillet-derived energy expenditure to distinct costs such as ventilation or metabolite accumulation. This allows simulated outputs to align more closely with observed gas exchange patterns—particularly VO_2 , VCO_2 , and RER trajectories—offering a richer understanding of how different physiological systems contribute to total metabolic load. Second, the Lidar framework introduces predictive extensibility, as its parameterized equations for ventilatory and accumulation-related costs can be individualized and scaled for each subject if necessary.

This flexibility supports more accurate simulations across varied athletes and test conditions. Finally, the model offers diagnostic utility: mismatches between Lidar's estimated aerobic metabolic rate (MR_{ae}) and Boillet's predicted physiological power (P_{physio}) help identify situations in which aerobic supply is insufficient to meet total energetic demand—indicating recruitment of glycolytic or phosphagenic pathways.

Lidar's demand decomposition enhances Boillet's supply model by anchoring it to interpretable ventilatory and metabolic measurements. This integration allows the digital twin to both simulate energy system interactions and diagnose the physiological costs incurred at each moment of exercise.

Blood Lactate Gauge Integration – New Tanks Dynamics

In Boillet's base model, muscle lactate is a secondary computation based off of the G tank fill level, but to advance the ability of the model, it may be useful to directly add the lactate of the muscle and blood as their own gauges in the tank system. More than a visual effect, this explicit inclusion of blood lactate will allow exponentially more performance data to refine and reshape the model, as muscle lactate testing is very rare and invasive, especially in comparison to the current usage of easily accessible blood lactate testing worldwide.

Lactate is most closely tied to the glycolytic balance and anaerobic side of the model, and will thus have direct impact on muscle lactate estimates and then to Type 2 fiber relative proportions/sizes, due to their intrinsic fast-twitch position as lactate producers. The effects may well reach to impact threshold(s) estimates and oxidation fuel sources, as there is demonstrated zero-sum trade-off effect of aerobic vs anaerobic power – you can't have it all, necessarily (Van der Zwaard 2021, Seiler 2010).

Lactate Measurement

From ground-truthing with real lactate test data – blood lactate from portable analyzers (by far most common) are potentially accurate to 0.12 mmol/L typical error (Lidar, 2023) but usually the general trust is lower, at +/-0.2 or 0.3, assuming a good unbiased and clean sample (often a strong assumption) (Cerezuela-Espejo et al., 2018; Grassi et al., 1999). At higher (and during exercise) values can be more variant and contaminated, from a practical standpoint. Muscular lactate is measured by biopsy, with various procedures, either resulting in a concentration per wet or dry weight (in kg) (Green et al., 1983; Bangsbo et al., 1993). It is worth noting that muscular and blood lactate have different denominator units (kg wet weight of muscle vs Liters of blood), each of which is affected by different physiological factors, so this method has many limitations and assumptions.

Blood lactate values were taken with a Nova Biomedical Lactate Plus analyzer, which has been found to have relatively small proportional and fixed bias compared to gold standard reference analyzers, as well as being highly linear (Hart et al., 2013). Although within-blood sample variation is highly negligible, between sample variation remains fairly significant, and in practice ~1/10 readings is thrown out entirely due to lack of sample. From Hart's data the during-exercise standard deviation of the difference between Nova readings and ground truth reference was up to 1.43 mmol/L. Sweat and other contaminants can ruin the reading as well. At-rest readings are assumed to be more accurate, closer to an SD of 0.2-0.3.

Muscle Lactate Dynamics

Muscle lactate behavior is adopted from Boillet and is connected to the usage of the Glycolytic tank, both capillary AT and main section AG. The instantaneous value is:

$$[Lac]_m = \frac{1}{C2} * (D_{AT} + D_{AG}) * \frac{1}{mass_{muscle}} + RMLA$$

Where the sum of D is the current depletion level, not rate, of AT and AG:

$$l * AT * 1000 + (l - \theta) * AG * 1000 \text{ if } l > \theta$$

And mass is the effective muscle mass for the task.

Blood Lactate Dynamics

There is limited available work on concurrent muscle and blood lactate under controlled conditions similar to the present study, so a combination of empirical relationships, analytical formulae, and known constraints are employed. Generally, similar to oxygen

kinetics at low intensities, the behavior is roughly first-order saturation, with a response lag to muscular lactate (Caen et al., 2024; Green et al., 1983).

As a start, Chwalbinska-Moneta et al. (1989) did incremental cycling tests for applicable endurance-type lactate estimation, arriving at muscle-blood R^2 of 0.8369 with SE 0.676. Gorostiaga (2014) also did similar work, finding an R^2 of > 0.8 for muscle and blood lactate, but whose data was inapplicable due to highly intermittent exercise protocol and sparse lactate recordings that were not time-synced or necessarily steady-state representative.

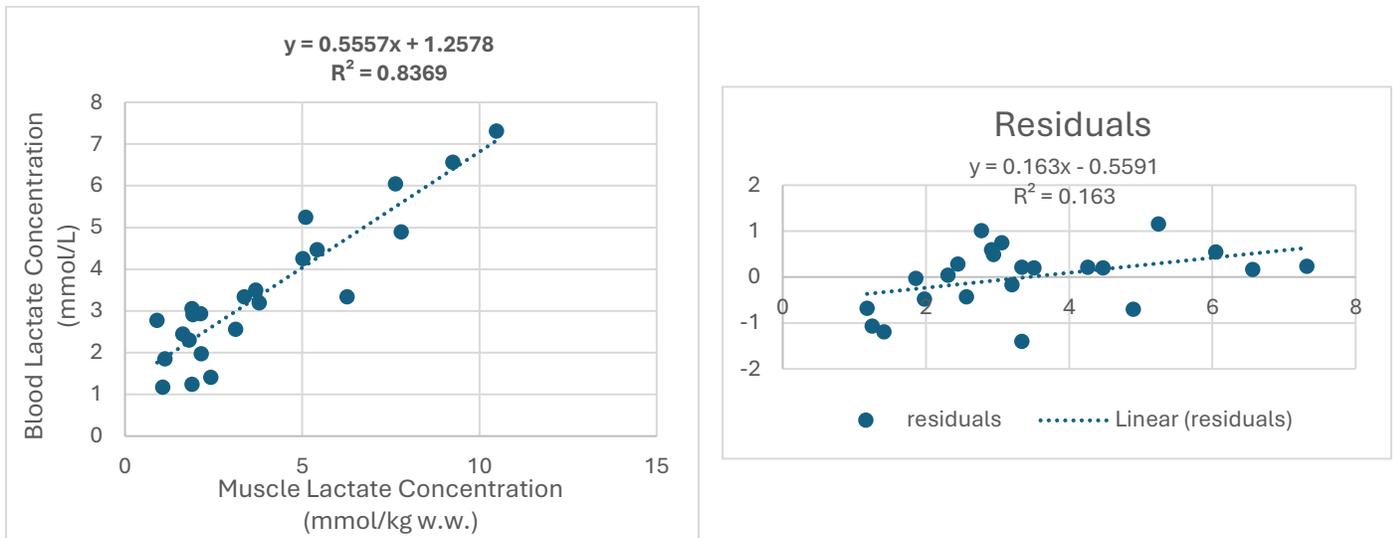


Figure 10A and B: Blood and muscle lactate concentration regression, with corresponding residual plot, data from Chwalbinska-Moneta et al. (1989).

With 21 Degrees of Freedom, the residual plot yielded a T-statistic of 1.9236 for a p-value of $0.068 > 0.05$, thus we fail to reject that the residuals are likely roughly homoscedastic, without a significant linear trend.

Blood lactate is modeled as tending towards a semi-steady-state, as set by muscular values from the regression, of the form:

$$[La]_b = a * [La]_m + b$$

To be grounded near $a=0.55$ and $b=1.2$, however a constraining factor is that some recent research has placed resting blood and muscle lactate as ~ 1.5 and ~ 3 mmol/L and mmol/kg w.w., respectively, as used and assumed by Boillet (Green et al. 1983, Bartoloni et al., 2024).

If a resting blood lactate value is known for a subject, the intercept b is adjusted from the above used regression to go through *that* value and the resting muscular lactate value as well (otherwise assumed to be 3). The new intercept is thus, by rearrangement:

$$b' = [La]_b^{rest} a * [La]_m^{rest}$$

The reality is likely more complicated, as resting lactate values are often dependent on muscle fiber type and overall fitness, but that is underdetermined by the current available data and beyond the scope of this project.

Conditional blood lactate (that is, derived from muscle lactate) time constant of response values were inferred by standardized testing procedure of regular blood lactate steady states – minimum of 2.5 minutes up to the gold standard of 6 minutes at a given intensity to allow for full metabolic stabilization. Following the typical saturating exponential form assuming 3τ at the end of the stage (for 95% of steady-state value) results in

$$3\tau = [150,360] s$$

and thus $T_{blood} = [50,120]s$. Clamping muscle lactate between physiological norms (ADD source),

$$[La]_m \in [\sim 1.5, \sim 30] = [\lambda_{min}, \lambda_{max}]$$

mmol/kg w.w., the iterative update rule gives a conditional blood lactate time constant from:

$$\tau(\text{Blood Lactate}|\text{Muscle Lactate} = [La]_m) = \tau_{max} - \frac{([La]_m - \lambda_{min})}{(\lambda_{max} - \lambda_{min})} (\tau_{max} - \tau_{min})$$

From Boillet's simulations (Fig 2B,D, and 5A), muscular lactate time constants were around ~24s for some representative cases, putting the time-constant:

$$\tau(\text{Blood Lactate}|\text{Muscle Lactate} = [La]_m) = \tau_{blood} - \tau_{muscle} = \sim 26 - 96s$$

Which can be changed later if time-course of lactate is unsatisfactory.

The update step occurs when muscle lactate exceeds resting muscle lactate (RMLA):

$$[La]_b(\text{new}) = [La]_b(\text{old}) + ((a * [La]_m + b) - [La]_b(\text{old}))/\tau(B|M) * dt$$

Which mirrors classic saturating exponential behavior at a discrete timestep Δt , too.

Full algorithm code in appendix.

Running Data: Modeling

Test data were initially examined before building digital twins for characteristic and descriptive correlative analyses.

Description of Test Data

After removing derived test data (like % Calories from carbohydrates and fats, which are computed directly from RER, or METS which is directly computed from VO₂), we can get an understanding of how the base measured variables are interrelated.

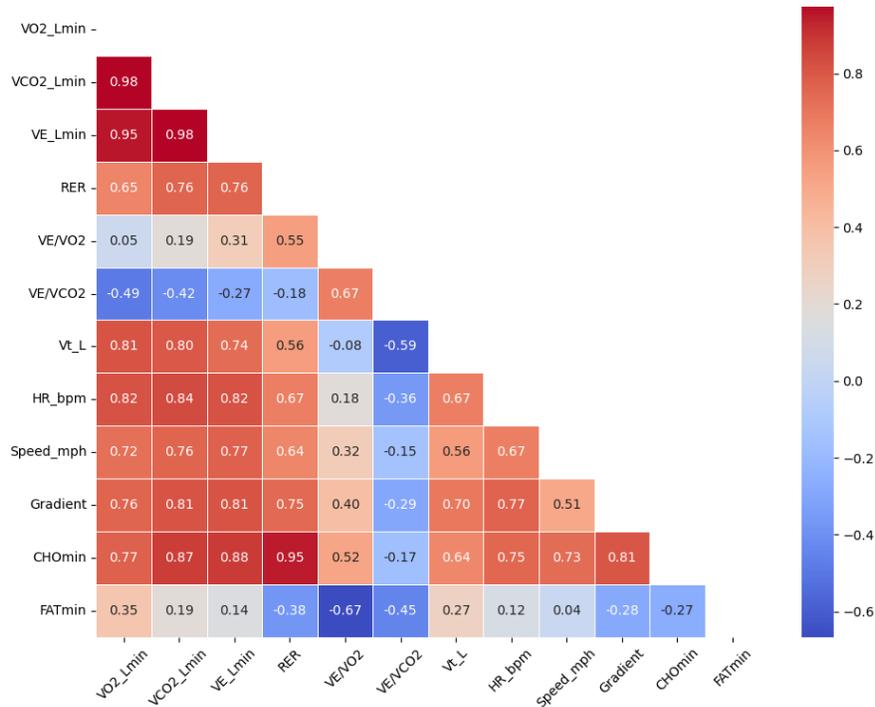


Figure 11: All-subject (n=52) average correlation matrix heatmap (for data from threshold tests only)

Most ventilatory variables have a moderate to strong positive correlation, with the exception of fat oxidation variables and VE/VCO₂, which decrease as intensity rises (and the other variables rise).

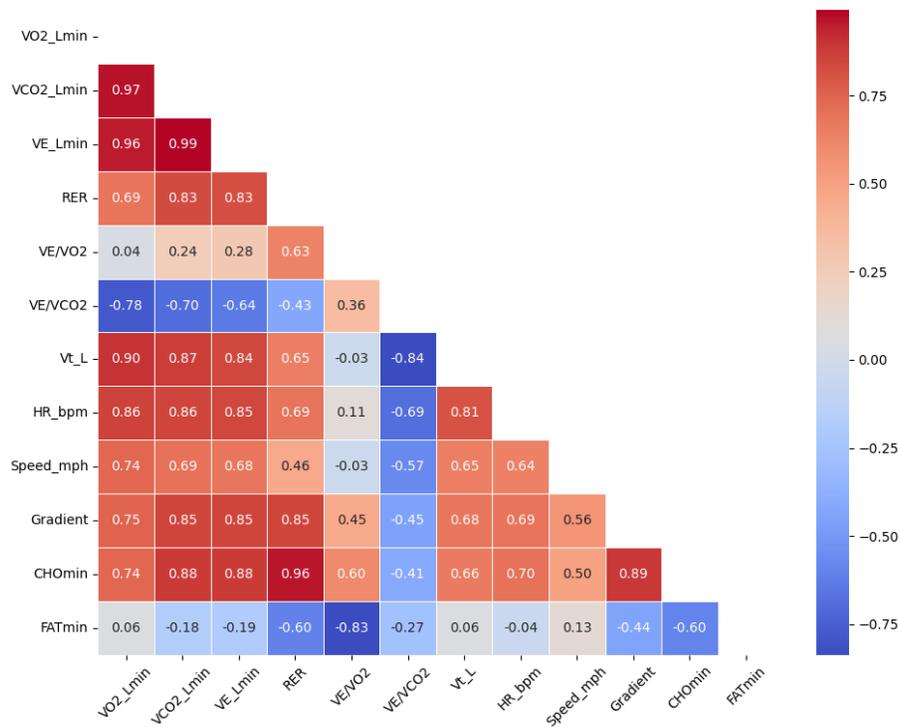


Figure 12: All-subject (n=55) average correlation matrix heatmap (for data from max tests only)

Due to the intensity range spanned in a maximal test, we might expect some of these correlations to be stronger than in the threshold test. To see if that is the case, the difference correlation heatmap is computed across all subjects, and is shown below.

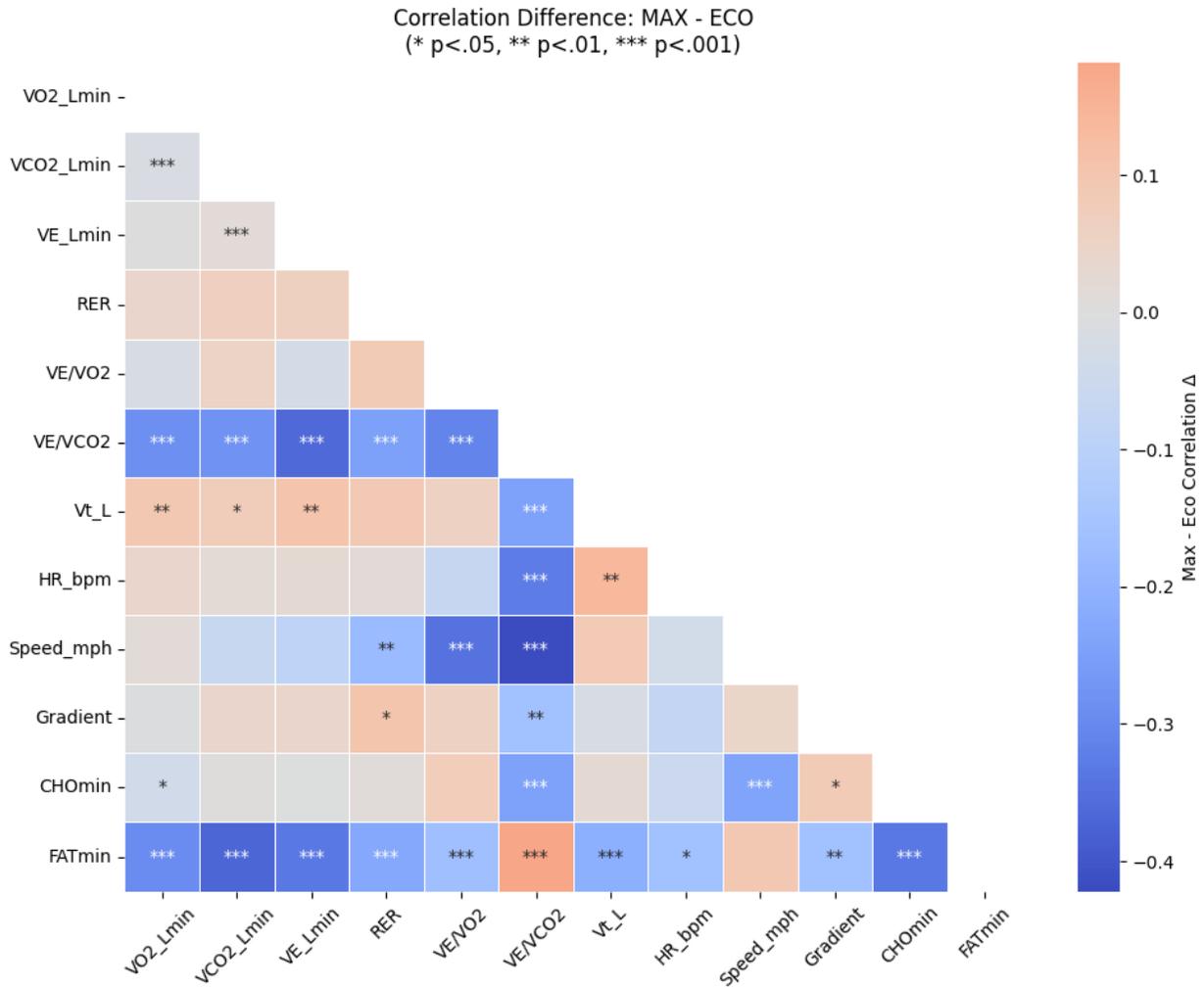


Figure 13: All-subject (n=52) average difference correlation matrix heatmap, showing significant correlative differences between max and threshold test data

In the max tests, Speed and VE/VCO2 tended to be much more strongly negatively correlated than in threshold tests, which seemed to be the case for a lot of variables paired with VE/VCO2. It makes sense that VCO2, produced in excess during high intensities, would be relatively higher during maximal exertion tests than in sub-maximal threshold assessments.

Ventilatory Data Correction

Typical threshold and VO₂max step test data looks something like this for RER, VO₂, and VCO₂, as speed increases every couple of minutes:

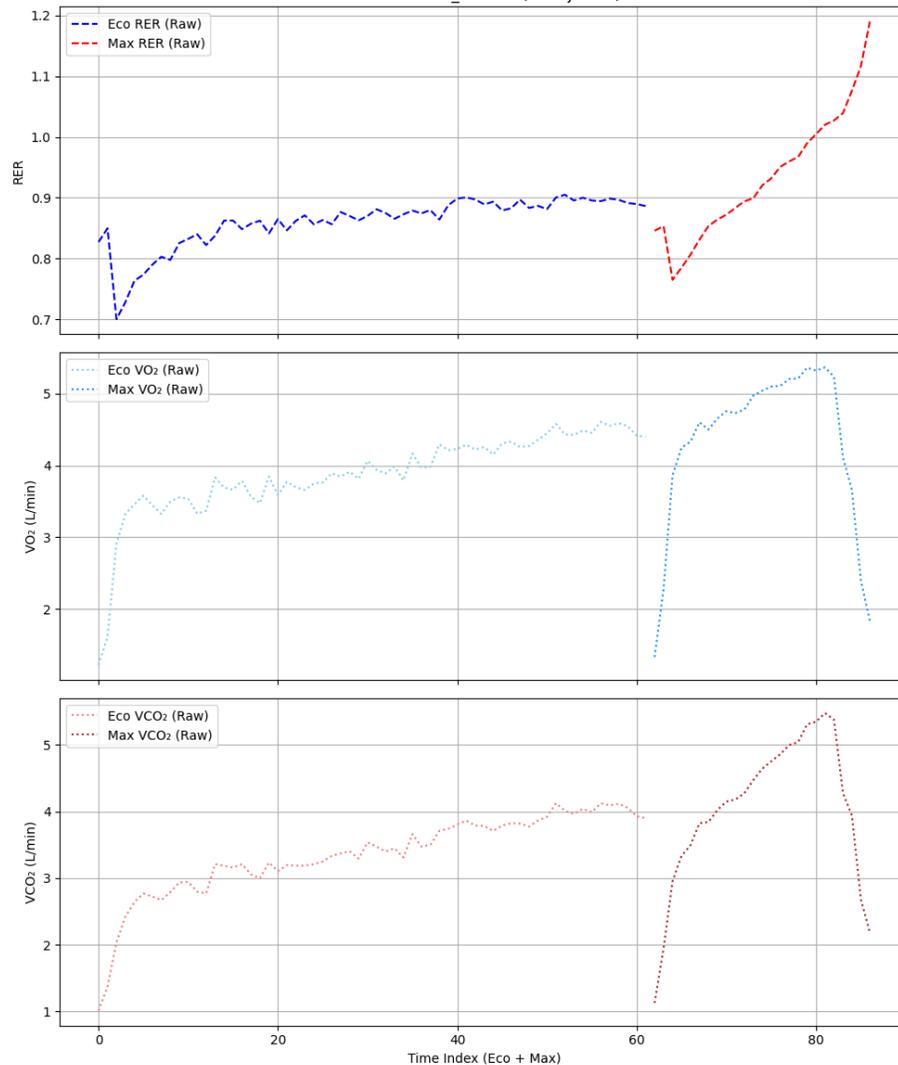


Figure 14A, B, and C: Example subject test data - progressive increases in intensity lead to higher ventilatory values for all three variables across the threshold ('eco') and especially during the maximal protocol. Single subject representative data appended together from the two tests – in reality, a ~10 minute full rest was given between test protocols.

Unfortunately, this was not the case for much of the data. RER for max tests often did not exceed 1.0, much less the 1.1+ that is usually expected just before exhaustion, as a criterion for attaining maximal effort. This was the case even for trained (high VO₂max athletes):

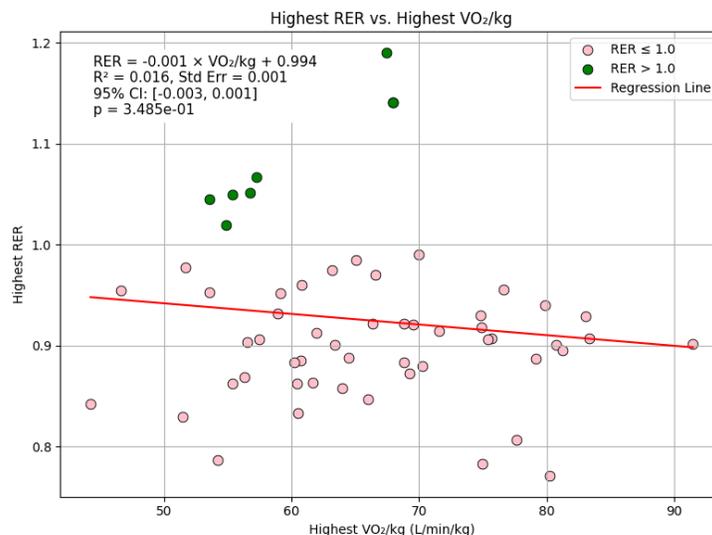


Figure 15: Running subject data, showing 84% of max tests that did not exceed 1.0 RER

Note that the slope term has a statistically insignificant p-value, thus it is unlikely that (lack of) fitness (as measured by VO₂max) is causal in any way to the RER errors – the idea that unfit people are unable to push themselves hard enough and/or fit people producing not enough lactate to cause significant buffering CO₂ effects does not seem to explain the homoscedastic appearing RER errors (Korkmaz Eryilmaz & Polat, 2021).

Furthermore, RER was often (especially in early stages of either eco and max tests) at physiologically-impossible sub-0.7 RER values

Crouter et al. (2006) and others have detailed accuracy and reliability specifications for the ventilatory data – which tends to be pretty precise and unbiased under most conditions for the Parvo TrueOne 2400 metabolic cart used. Unfortunately, during the test data collection period, readings were so errant (VO₂max readings far exceeding realistic values, and RER values that were far too low for maximal efforts) at one point and one of the gas hoses was found to have extra condensation which likely affected the calibration process. Thus, there is strong likelihood that there is some bias in the measurements, which will be corrected for.

A correctly calibrated Parvo was found to have the following between-day accuracy specifications against a criterion Douglas Bag control method (Crouter et al., 2006):

Table 2: Accuracy and reliability summary of Parvo metabolic cart for ventilatory data

	Mean error	95% PI low	95% PI high	CV (%)	SD (est.)
VE (STPD l/min)	-1.34	-11.91	9.22	7.3	5.283
VO ₂ (STPD l/min)	-0.04	-0.28	0.19	4.7	0.118
VCO ₂ (STPD l/min)	-0.03	-0.32	0.25	5.7	0.143

SD was estimated as the 95% PI range divided by 4, assuming normality for calibration data. The raw data distribution of RER values was as follows:

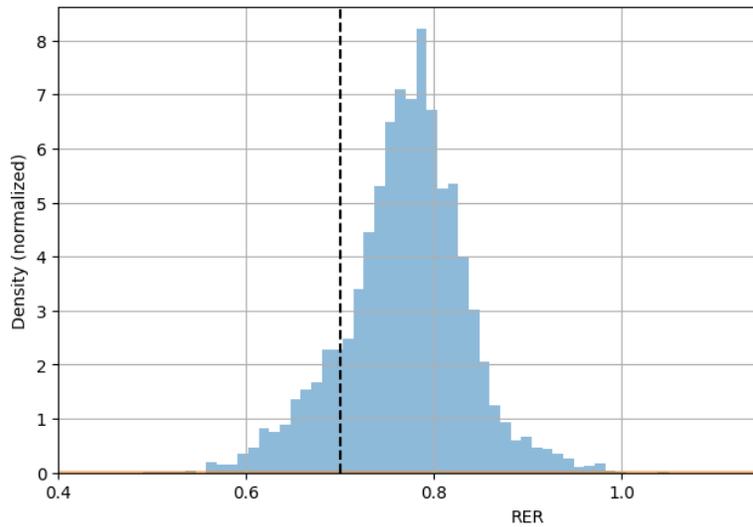


Figure 16: RER histogram, showing large portion of values under physiologically impossible 0.7 threshold (dotted line).

Perhaps it is more apt to say these are steady-state improbable and not predicted by peer-reviewed understanding, as early-exercise low RER is theoretically possible – due to potential time delays in lactic power usage from the glycolytic system (as the phosphocreatine system tends to cover rapidly any jumps in exertion) through the muscle, to the blood, to get buffered, and then exhaled as CO₂, while the VO₂ response potentially rises faster, leading to an artificially low (non-steady-state, and not fat-carb oxidation related) RER (Korkmaz Eryılmaz & Polat, 2021; Van Der Zwaard et al., 2021).

That can be said for *some*, but not all of the RER data:

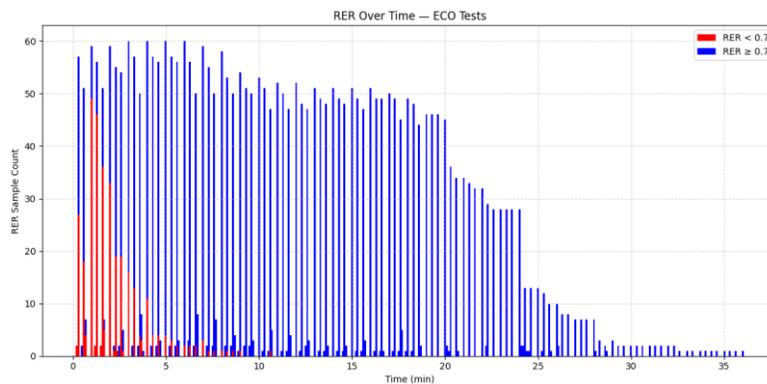


Figure 17: Eco (Threshold) test RER distribution over time.

Many of the RER erroneous values are outside the typical 1-minute window for potential slow boot-up of the body.

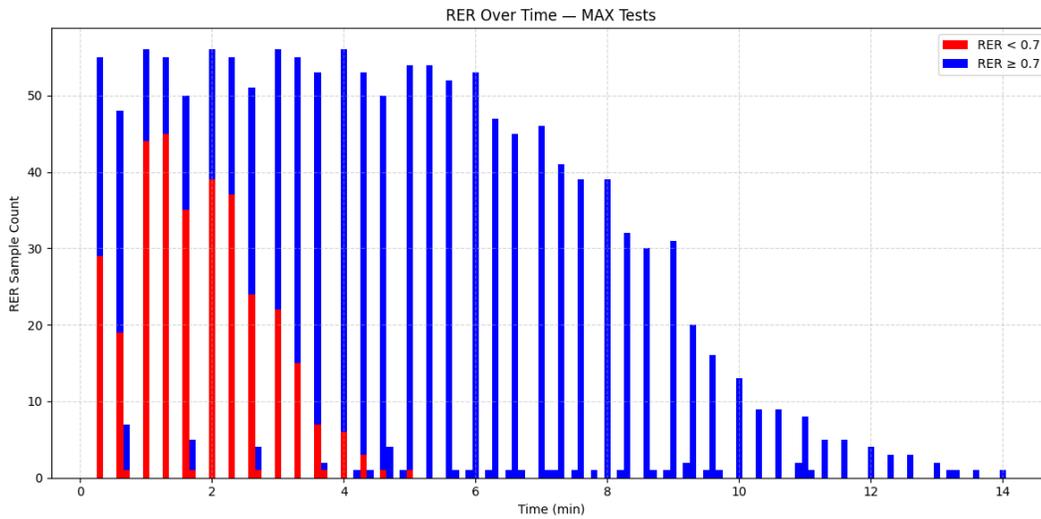


Figure 18: Max (VO₂max) test RER distribution over time.

There is a similar issue with the maximal test data. Paring out the lowest RER value helps only a little bit, a more structural correction is needed:

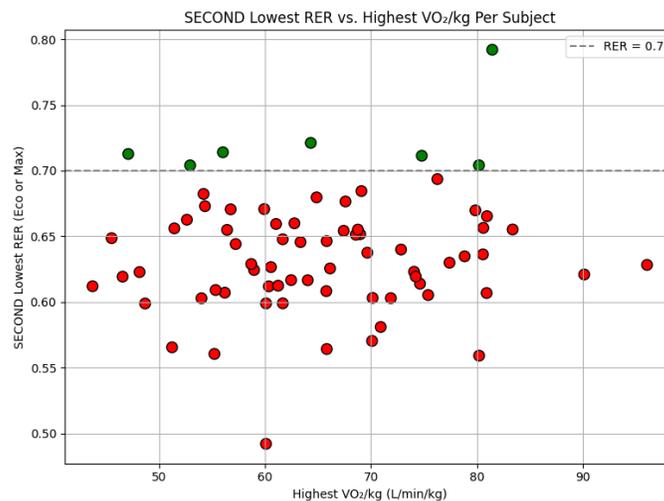


Figure 19: Second-lowest RER value versus VO₂max, for all subjects, to explore the idea that a single outlier per subject could have been throwing off the data.

To minimally adjust the VO₂ and VCO₂ signals using global scaling factors, while enforcing that all corrected RER values remain within a biologically credible range ($0.7 \leq \text{RER} \leq 1.4$), we assume that the measurement errors in VO₂ and VCO₂ are systematic and multiplicative in nature, such that the true values can be approximated by scaling the raw measurements with constants α and β . Global scaling was chosen as the machine

calibration was the same for both tests, and thus any bias is likely to be consistent across the testing period (<1hr).

$$\begin{aligned} \text{VO}_2^{\text{corrected}} &= \alpha \cdot \text{VO}_2^{\text{raw}}, \\ \text{VCO}_2^{\text{corrected}} &= \beta \cdot \text{VCO}_2^{\text{raw}} \end{aligned}$$

Thus the new corrected RER at each time point is defined as:

$$\text{RER}^{\text{corrected}} = \frac{\text{VCO}_2^{\text{corrected}}}{\text{VO}_2^{\text{corrected}}}$$

To find optimal correction parameters α and β , we minimized a cost function composed of three terms:

1. RER boundary violation penalties, encouraging all RER values to stay within [0.7, 1.4].
2. Regularization terms, penalizing deviation of α and β from 1, based on empirically validated measurement standard deviations ($\sigma_1 = 0.118$ for VO_2 , $\sigma_2 = 0.143$ for VCO_2).
3. Hard constraint enforcement, where any RER falling below 0.7 incurred an additional penalty scaled by a large constant (10^6), ensuring that the minimum corrected RER in a given dataset was ≥ 0.7 .

The optimization problem was formulated as:

$$\begin{aligned} \min_{\alpha, \beta} \sum_t & (\max(0, 0.7 - \text{RER}_t^{\text{corr}})^2 + \max(0, \text{RER}_t^{\text{corr}} - 1.4)^2) + \lambda_1(\alpha - 1)^2 + \lambda_2(\beta - 1)^2 \\ & + \text{Penalty}_{\text{if min RER} < 0.7} \end{aligned}$$

Where: $\lambda_1 = \frac{1}{\sigma_1^2}$, $\lambda_2 = \frac{1}{\sigma_2^2}$, $\sigma_1 = 0.118$ (VO_2), $\sigma_2 = 0.143$ (VCO_2)

For each subject, both economy (eco) and maximal (max) test files were analyzed. After loading and validating the data, the lowest raw RER value across both tests was identified. If the entire dataset satisfied $\text{RER} \geq 0.7$, no correction was applied and the raw data were preserved. Otherwise, the test with the lowest RER was selected as the fitting dataset for parameter estimation.

The optimal α and β values were then computed via bounded optimization (scipy.optimize.minimize), constrained to plausible ranges ($\alpha \in [0.85, 1.0]$, $\beta \in [1.0, 1.2]$). These scaling factors were subsequently applied uniformly to the full VO_2 and VCO_2 time series across both tests, and stored and saved in the subject's dataframe as `_corrected`. Derived columns (like RER, and carb/fat oxidation metrics) are then recomputed.

Full code is in appendix.

The results of the adjustment on a sample individual are as follows:

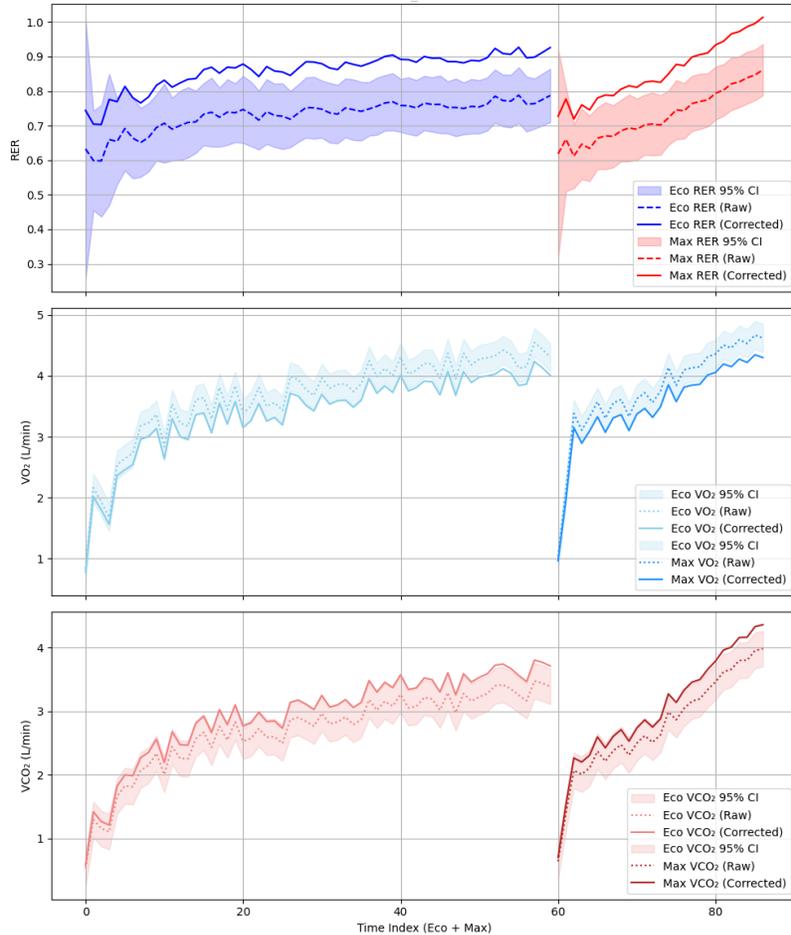


Figure 20: Example of single-subject pre- and post- correction RER curves from eco (threshold) and max (VO₂max) tests – appended together for compactness (in reality, roughly 5-10 minutes standing rest given in between).

Note that for the corrected RER values, they mostly fall outside of the 95% CI for the original RER values, this is to be expected given the physiological implausibility (bias) of the combination of VO₂ and VCO₂ values. On the other hand, the adjusted VO₂ and VCO₂ values often fall within the 95% CI, indicating the measurement miscalibration bias is nearly potentially accounted for in randomness of measurement, and thus acceptable as a reasonable correction to the bias.

A summary of the corrections applied to VO₂, VCO₂, and RER is below:

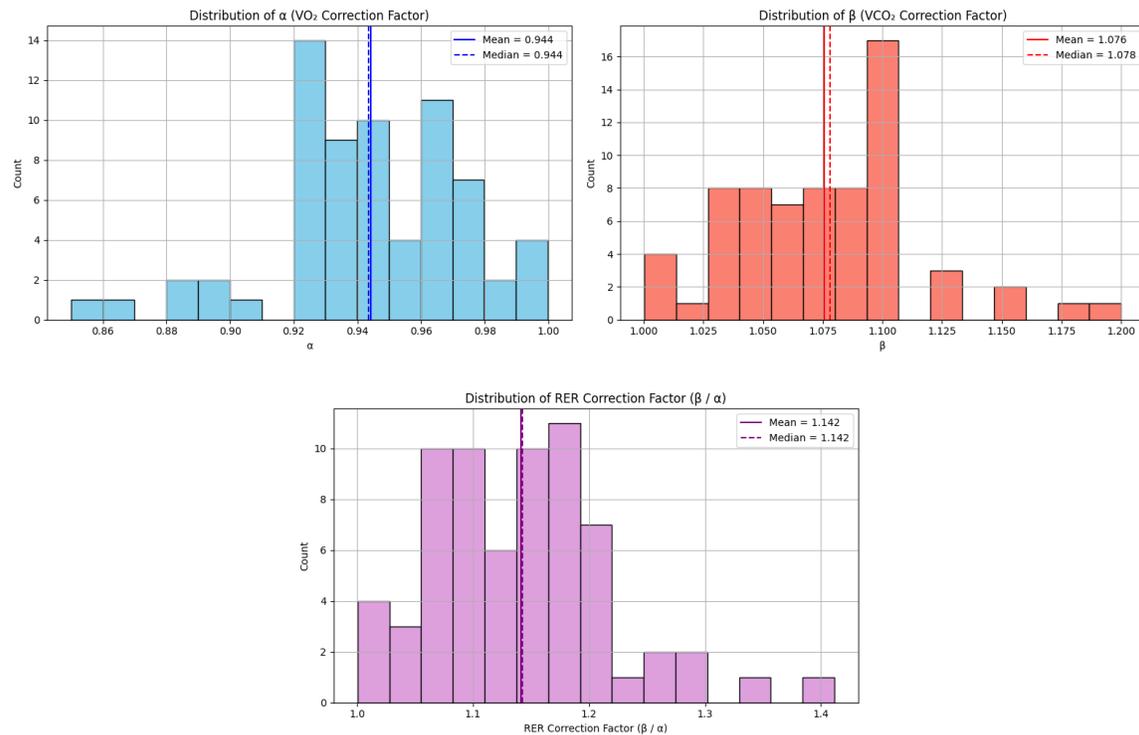


Figure 21A, B, and C: VO₂, VCO₂, and RER correction factors applied to subjects with implausible RER values (either at max or at low end of threshold testing).

Now the RER values are physiologically possible and thus hopefully the VO₂ and VCO₂ are more representative of their real underlying values.

Ventilatory ratio and Caloric and Substrate breakdowns (derived from VO₂ and VCO₂) were recomputed with the corrected data (based on literature on chemical energetics – Lii et al., 2024).

Ventilatory recalculations:

$$VE/VCO_2 = \frac{VE}{VCO_2}$$

$$VO_{2,kg} = \frac{VO_2 \cdot 1000}{\text{Body Mass}_{kg}}$$

Substrate oxidation in grams/minute:

$$CHO_{min} = \max(4.55 \cdot VCO_2 - 3.21 \cdot VO_2, 0)$$

$$FAT_{min} = \max(1.695 \cdot VO_2 - 1.701 \cdot VCO_2, 0)$$

Fraction of calories burned that came from carbohydrate oxidation:

$$\text{CHO}_{\%} = \min\left(\max\left(100 \cdot \frac{\text{RER} - 0.7}{0.3}, 0\right), 100\right)$$

Instantaneous accumulation of kcal:

$$\text{Ackcal}_{\text{inst}} = (3.9 \cdot \text{VO}_2 + 1.1 \cdot \text{VCO}_2) \cdot \Delta t$$

And cumulative kcal burned:

$$\text{Ackcal}_{\text{cumulative}} = \sum_{i=0}^t \text{Ackcal}_{\text{inst},i}$$

Instantaneous kcal from carbohydrates and fats:

$$K_{\text{CHO}} = \text{Ackcal}_{\text{inst}} \cdot \frac{\text{CHO}_{\%}}{100}$$

$$K_{\text{FAT}} = \text{Ackcal}_{\text{inst}} \cdot \left(1 - \frac{\text{CHO}_{\%}}{100}\right)$$

These values, although not as useful for direct model fitting, can be helpful for athletes planning fueling strategies for longer races – how much they might need to consume depends on their caloric expenditure and what kinds of fuel sources it comes from (Caen et al., 2024).

It is important to note that the above correction(s) does not remove the inherent noise in the ventilation data – for example even 20, 30, and 60s averaging over VO₂ breath-by-breath sampling still results in high variance for the same steady-state workload, even after filtering out the first minute of test data for each stage. Within-stage standard deviations were typically 3-7% of the average VO₂ (L/min) value *after* filtering and 20s-smoothing, when averaged across stages, for each subject. This comes into play in model evaluation, where some of the apparent gap between model and data may be related to the underlying variance in the data (and measurement of it) itself.

If Y = true VO₂ value (unknown) and Y_{pred} = model prediction,

And $Y_{\text{obs}} = Y + e_{\text{obs}}$ where we assume $e_{\text{obs}} \sim N(0, \sigma_{\text{obs}}^2)$

Then the observed model error is:

$$Y_{\text{pred}} - Y_{\text{obs}} = (Y_{\text{pred}} - Y) - e_{\text{obs}} = \text{model error} - \text{data noise}$$

Which means that

$$\text{observed error variance} = \text{model error variance} + \text{measurement error variance}$$

Thus to get the model intrinsic error:

$$\sigma_{\text{model}} = \sqrt{\sigma_{\text{model,observed}}^2 - \sigma_{\text{datanoise}}^2}$$

Grade-Adjusted Pace Model Refinement

Running power estimations would not be complete without consideration of the gradient, or incline of the running surface – it takes more energy to run uphill, due to fighting gravity and other smaller factors in changing running techniques. The VO₂max protocols often use a combination of speed increases and gradient to make sure that participants can run to metabolic exhaustion, and that failure comes from energetic failure more than inability to move one's feet fast enough.

Purely energetic computations of a “Grade-Adjusted Pace” (GAP) often miss out on the human realities of muscular limits and tendon properties at different velocities and tensions, and are often calibrated at extreme (non-realistic for most runners except elite trail runners) gradients (Minetti et al 2002). Here a combination of energetics and population economy data are used.

The work rate done against gravity is $Wg = mgv \sin(a)$ where a is the slope angle of inclination in radians. Since for typical treadmill gradients (<15%) $\tan(a) \approx \sin(a)$ and $\text{Grade} = \frac{\Delta h}{\Delta x} = \tan(a)$, $Wg = mgv \left(\frac{G\%}{100}\right)$ where m is mass, g is the force of gravity, v is the horizontal velocity in m/s and $G\%$ is the gradient (expressed as a percent).

This is combined with the varying energetic cost of flat ground running – against speed, that is. Using Davis's (Running Writings, 2023) extraction of Black et al. (2018) running economy data for flat ground across recreational and elite runners, a GAP is calculated from Davis' procedure:

- 1) Get metabolic cost of the flat ground speed
- 2) Get added metabolic cost of the hill gradient
- 3) Sum (1) and (2) to get total metabolic cost
- 4) Match (3) to equivalent flat ground speed that requires that cost

The outputs from this matching process were scraped across a range of speeds (6 to 13mph) and gradients (1 to 12%), and input into a second-degree polynomial in two variables for a full smooth surface that could capture non-linearities, of the form:

$$z_{pred}(x, y) = B_0 + B_1x + B_2y + B_3x^2 + B_4xy + B_5y^2$$

Where x is gradient, y is speed, and z_{pred} is predicted gap speed. Mean Squared Error (MSE) came out predictably (acceptably) small:

$$MSE = \frac{1}{n} \sum (z_i - z_{hat})^2 = 0.00297mph$$

The final form came out to:

$$GAP(mph) = 0.118 \cdot x + 0.60409 \cdot y + 0.00322 \cdot x^2 + 0.036 \cdot x \cdot y + 0.01838 \cdot y^2 + 2.02215$$

Which was applied across all test subjects, to both eco and max tests (some eco tests used a 1% gradient at higher speeds, to account for missing air resistance one might encounter outside). See appendix for full code.

Energy Equivalent of Oxygen Extension

Boillet uses a fixed average value of C1 as detailed by Di Prampero in Energetics of Muscular Exercise (ADD act SOURCE), at roughly 20.9 J/ml, which tends to be valid at low-middle steady-state intensities (around RER=.775). The molecular stoichiometric and thermodynamic relationship between the fat and carb balance (RER) in substrate oxidation is consistent with current industry knowledge – burning carbs is more efficient than burning fat.

For glycogen (carbs), the P/O2 is 6.2, and for FFA (fat) it is 5.6, thus the total substrate oxidation efficiency is some combination of the two. This may seem like a small adjustment but the effect is great (~11% change over the range) and may help explain better both the individual test data and more broadly the recent high-carb fueling pushes in endurance sports – allowing for races to be done at higher intensities as historically carbohydrate storage was a main limiter, and thus athletes tended to rely more on fat oxidation to preserve muscle glycogen.

The inferred relationship from Di Prampero is linear, and converted to J/ml from P/O2 is as follows:

$$C1 = 19.47 + 7.13 \cdot (RER - 0.7)$$

At an RER of ≥ 1 (where all oxidative energy is assumed to be from carbohydrates), the yield is 21.61 J/ml, while at RER =0.7 (all fat oxidation) the yield is 19.47J/ml.

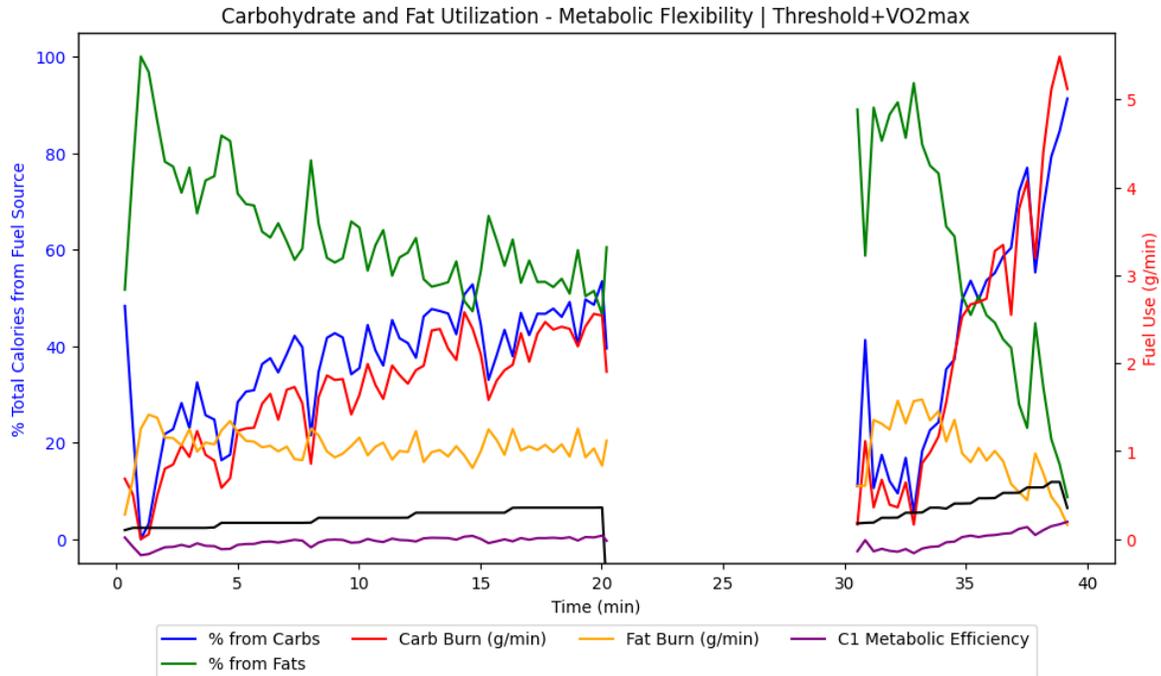


Figure 22: Sample VCO_2/VO_2 – and RER derived metabolic breakdown – fat burn and contribution to total energy supply indeed diminishes as intensity (black – speed) increases. Carbohydrate contribution and oxidation do the opposite. C1 increases.

Varying Eta estimation

On the whole, studies seem to find a decently linear relationship between steady speed of running on flat ground and the oxygen cost per kilogram (VO_2 – in $ml/kg/min$), although there is surprisingly limited data easily available to support this (Black et al., 2018). Within the current running dataset of those who completed a threshold test, we see a similar trend.

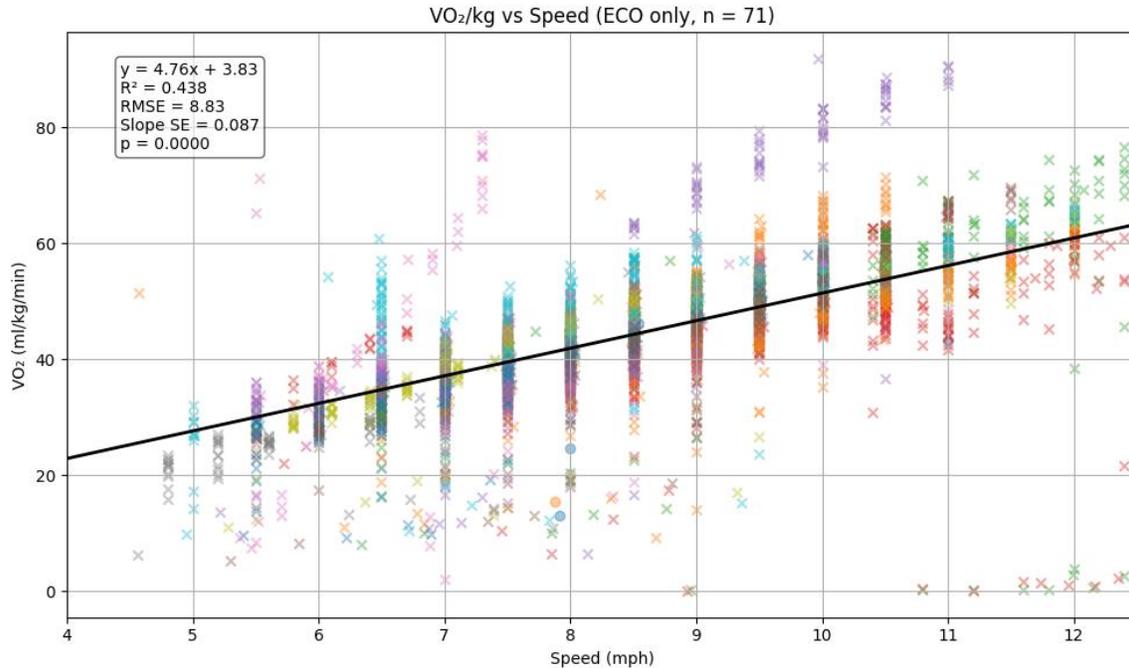


Figure 23: All-subject economy (threshold) test data – VO₂ energy cost of running versus the stage speed in mph.

On the whole there is good argument for linearity, but as we know from the central limit theorem when independent observations (e.g., individual athletes' VO₂ responses to running speed) are aggregated, the sampling distribution of their mean tends toward normality—even if the underlying individual relationships are nonlinear or heteroscedastic. This masks idiosyncrasies in each person's physiology. Similarly, the law of large numbers ensures that individual variability gets "washed out" when analyzing group means, pushing observed trends toward a smoother, often deceptively linear pattern. As a result, population-level linearity does not imply that the relationship between VO₂/kg and speed is linear within each subject. For physiological modeling and precision diagnostics, this distinction is critical—especially if individual metabolic costs or running economy deviate significantly from group behavior.

GAP Pace in mph, over the same VO₂ power estimate from below/at LT1 that Boillet uses for cycling efficiency, but recalculated at estimated steady state segments of the threshold tests, is used to capture the difficulty of running at higher speeds -- muscle contraction efficiency at the molecular and musculotendinous unit level is not necessarily constant (Hansen et al., 2002; Wackwitz et al., 2025). We assume a linear model, given the appearance of the data over the surveyed range.

Ventilation data are first filtered – Cleaning by outlier clipping by interquartile range (lower and upper bounds set at $q1 - 1.5 * iqr$ and $q3 + 1.5 * iqr$, respectively).

Efficiency values from the running data collected were typically in the range of 5.5 - 9 mph per kW of VO₂ (metabolic power) at steady state.

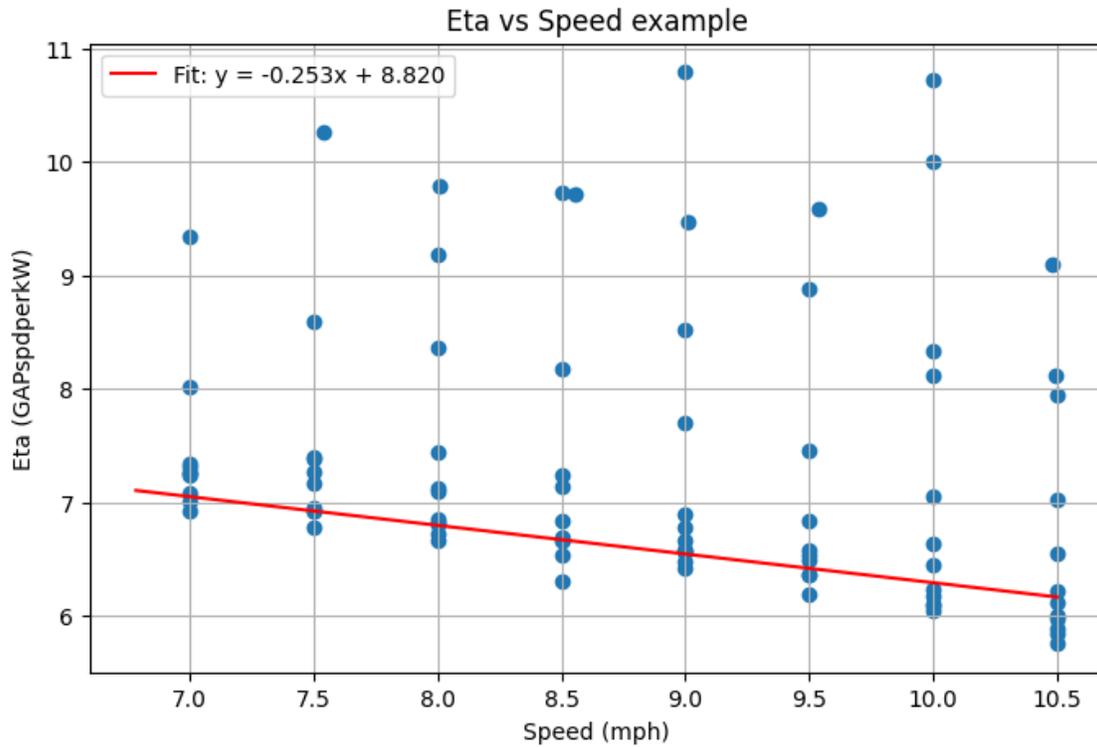


Figure 24: Example subject data of mechanical ('gross') efficiency over speed from eco (threshold test)

Note that the shown regression is fitted excluding some of the outlying points at y (Eta) values of ~8 or more, which are usually from weird timing errors from recovery periods between stages, or are from early in stages where the VO₂ has not had time to saturate to fulfill the power demand, and thus underestimate the VO₂ required to go a speed, overestimating the Eta efficiency.

Performance Pillars Influence Assessment

As a sanity check for the collected and corrected test data, standard exercise science regressions are computed, to check the placement of the data in the known context (Van Der Zwaard et al., 2021).

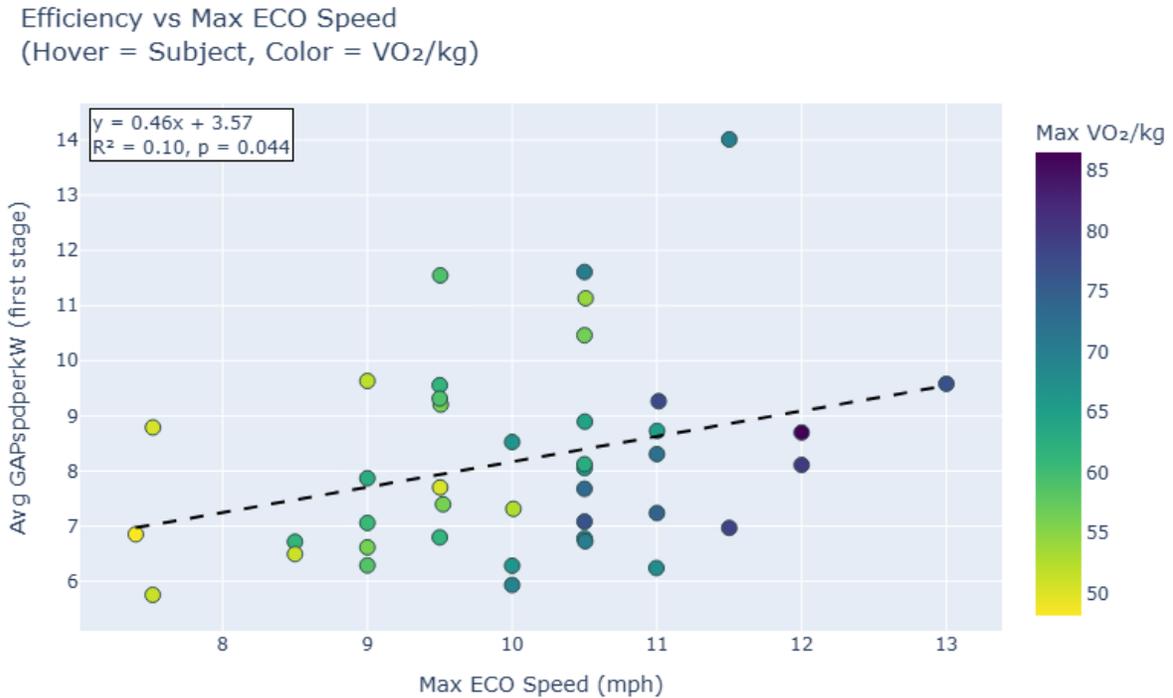


Figure 25: Efficiency of running at lowest stage versus maximal sustained stage in threshold test, colored by highest VO₂ (per/kg) reached in maximal test. Fewer subjects, as not everyone had both eco and max test data.

Y-axis efficiency (higher GAP speed per kW of energy demand on the body at low intensity) has a small but significant (at the 0.05 level) positive correlation with maximum eco (threshold test) treadmill speed – faster/better runners tend to be slightly more efficient. There is also a subjective trend of higher VO₂max values at higher speeds as well.

Shown are three pillars of endurance performance, or at least proxies for them – functional top speed, aerobic engine size (VO₂max), and efficiency of movement (also sometimes referred to as economy) (Behncke, 1993; Van Der Zwaard, 2021; Kim et al, 2021).

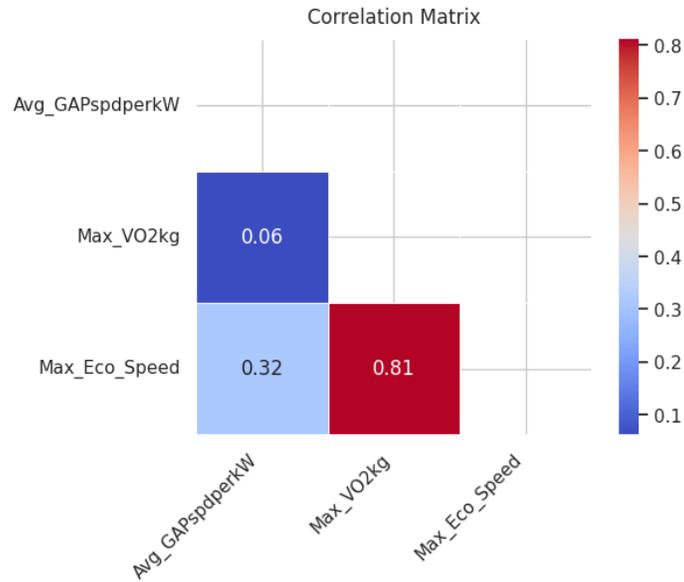


Figure 26: Correlation matrix of three pillar variables, seen as crucial to endurance performance

To understand the importance of efficiency and VO2max towards endurance performance in our dataset, we start with a simple multiple linear regression, of the form:

$$\text{Max_ECO_Speed} = \beta_0 + \beta_1 \cdot \text{Avg_GAPspdperkW} + \beta_2 \cdot \text{Max_VO}_{2\text{kg}} + \varepsilon$$

Using Ordinary Least Squares, we find from our data:

```

=====
Dep. Variable:      Max_Eco_Speed      R-squared:          0.729
Model:              OLS                Adj. R-squared:    0.714
Method:             Least Squares      F-statistic:       51.05
Date:               Fri, 23 May 2025    Prob (F-statistic): 1.71e-11
Time:               07:12:53           Log-Likelihood:    -38.997
No. Observations:  41                 AIC:               83.99
Df Residuals:      38                 BIC:               89.13
Df Model:           2
Covariance Type:   | nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
const	2.0609	0.819	2.518	0.016	0.404	3.718
Avg_GAPspdperkW	0.1828	0.058	3.146	0.003	0.065	0.300
Max_VO2kg	0.1011	0.011	9.388	0.000	0.079	0.123

```

=====
Omnibus:           0.850      Durbin-Watson:      2.113
Prob(Omnibus):    0.654      Jarque-Bera (JB):   0.921
Skew:             0.263      Prob(JB):           0.631
Kurtosis:         2.487      Cond. No.           526.
=====

```

Figure 27: OLS Multiple Regression Results for endurance 'pillar' variables

With $R^2 = .729$, 72.9% of the variance in max functional speed is explained by efficiency and VO2max, which is pretty strong. Both slope coefficients have p-values $< .005$, suggesting significance of predictors, and the intercept is significant at the $< .05$ level as well, despite a lack of physiological meaning.

For every ml/kg/min you increase your VO2max while holding efficiency constant, we see a $\sim .10$ (+-.01) mph increase in maximal sustainable speed.

Evaluating the model applicability under the OLS assumptions:

The Durbin-Watson (DW) value of 2.113 (near 2) indicates no issue of autocorrelation,

calculated by:
$$DW = \frac{\sum_{t=2}^n (e_t - e_{t-1})^2}{\sum_{t=1}^n e_t^2}$$

The high Omnibus and Jarque-Bera (from:
$$JB = \frac{n}{6} \left(Skew^2 + \frac{(Kurtosis-3)^2}{4} \right)$$

values indicate residuals are likely normal and not skewed or expressing excess kurtosis.

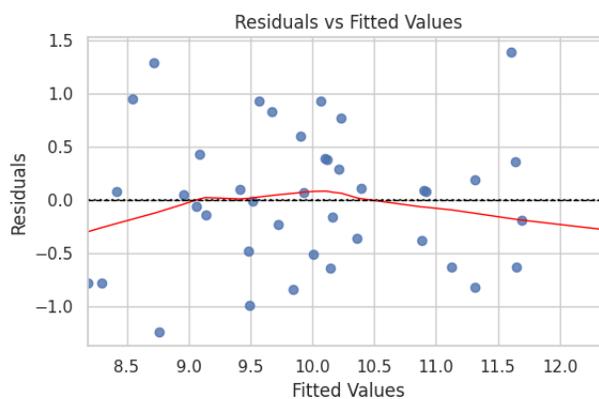


Figure 28: A visual inspection confirms the random scatter of the residuals of the OLS model above

As expected there is moderate collinearity given the Condition Number is 526 (>100), computed as
$$\text{Condition Number} = \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}}$$
, from the eigenvalues of the design matrix, which captures the relationships between the predictor variables.

These results fall in line with the expected and understood relationship between the predictor variables (Morton et al., 1990; Seiler, 2010; Van Der Zwaard, 2021).

Running Threshold Detection

To find running thresholds from the incremental stage tests performed, a combination of test-specific knowledge and ventilatory criteria was used (Cerezuela-Espejo et al., 2018; Green et al., 1983; Poole et al., 2021). To best capture effort and physiological response at a given intensity, the first minute of every stage was excluded, and the last two minutes of each stage data were averaged to define the stage effort.

First, LT2 is found from the last couple stages (given knowledge from helping run many of the tests and evaluating their data later) where the absolute ventilation ticks upward significantly above a rolling linear regression from previous stages while VE/VCO₂ and VE/VO₂ increase (any sustained increase, stage-to-stage).

LT1 is found from the remaining stages, knowing also from lab experience that LT1 and LT2 were highly unlikely to be neighboring stages, in combination with RER exceeding typical maximum fat oxidation values (~0.8-0.825) and VE/VO₂ increasing while VE/VCO₂ did not. This followed the established criteria of Cerezuela-Espejo et al. (2018) and established laboratory practices to the best approximation. In reality, other measures are also evaluated in real-time by expert physiologists, including absolute VO₂ and heart rate and subjective measures such as RPE, so this simplification has its limitations.

Algorithm full code in appendix, including stage-map outputs.

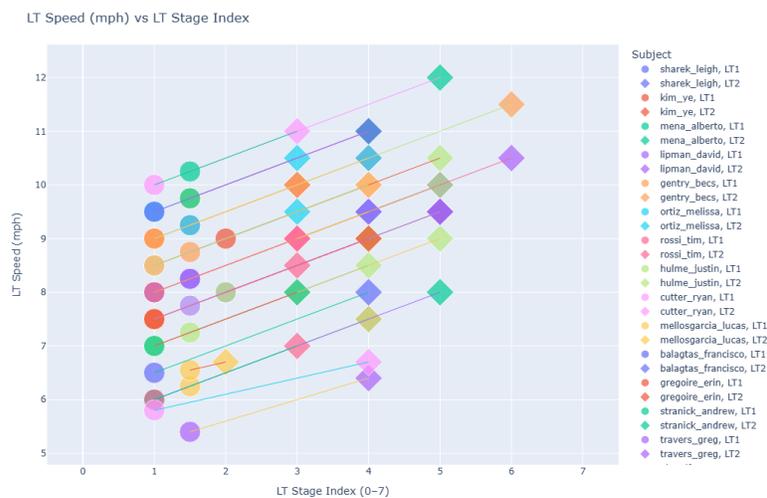


Figure 29: Threshold stage assignment sanity check for LT2 > LT1 and potential outliers from algorithmic decisions.

Unsurprisingly, subjects that had a higher VO2max had, on average, higher threshold speeds:

LT1 and LT2 Speed vs VO2max with Regression



Figure 30: LT (both) speeds versus VO2max of the subject, from their VO2max test. Separate linear regressions are included for the two types of thresholds.

Both threshold aerobic power values tend to be positively correlated with increased VO2max, though the scatter is substantial. Slopes are similar, interestingly. More surprisingly, was the finding of relative VO2max usage at thresholds, compared to the absolute VO2max potential of an athlete.

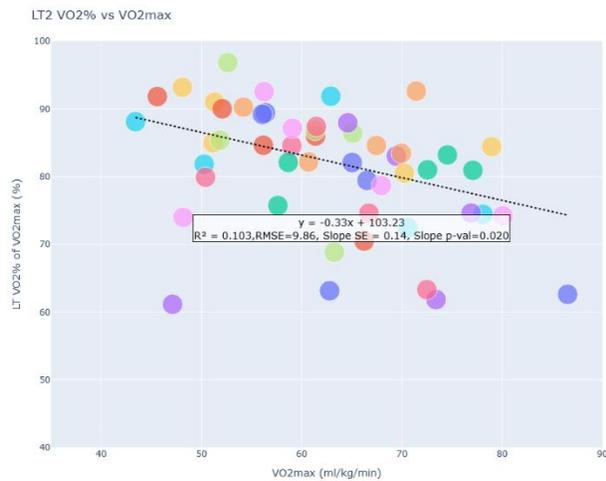
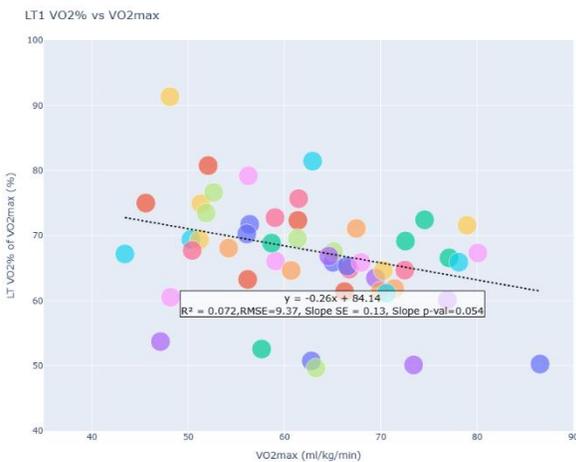


Figure 31A and B: relative (%) VO₂ utilization at LT1 (A) and LT2 (B) versus VO₂max of the subject, as determined by algorithmic threshold detection. Both with significant negative sloped linear regressions.

Based on prior research and general understanding of physiology (Van Der Zwaard et al., 2021; Barstow et al., 2000; Vanhatalo et al., 2016), we might expect that the more aerobically fit one is, the higher their threshold(s) might sit relative to their VO₂max. What we observe is the opposite – there is a tendency for those with low VO₂max values to have a high threshold (in terms of oxygen usage at an estimated threshold speed relative to their maximal measure oxygen usage), and vice versa for those with high VO₂max values.



Figure 32A and B: Absolute VO₂ utilization at LT1 (A) and LT2 (B) versus VO₂max of the subject, as determined by algorithmic threshold detection. Both with significant positive sloped linear regressions.

However, when considering non-percent VO₂ usage at thresholds, the expected trend appears – fitter individuals with higher VO₂max values are able to use more oxygen at their thresholds, ostensibly to run faster, as we saw earlier in Figure 25.

It is possible that the contrast seen here is an indication of a limitation of the study protocol – it is test-industry knowledge that often the 2.5-3min intervals are not sufficient to establish a perfect steady state for each stage in terms of metabolic and ventilatory response, and thus there is a tendency to overestimate threshold values, especially for people who are less fit – more of their energy is expected to come from anaerobic sources, and the VO₂ response time constant is larger (less quick) (Caen et al., 2024). This means for a ~3min stage, the last 2-minute data may underreport the oxygen energy cost of running at that pace, and thus the thresholds will often be set higher than that which is truly a steady state, leading to high relative (as % of VO₂max) threshold estimations.

There are other potential physiological limiters, especially with less-trained athletes, such as non-central fatigue (muscular, coordination, motivation, etc) that may disproportionately reduce their tested VO₂max relative to fitter individuals with more experience and physiological adaptation to high-intensity running (and exertion in general).

Metabolic Breakdown – Lidar (2023) Extensions

As introduced earlier, Lidar’s model extends traditional oxygen-based energy estimation by breaking down total metabolic demand into multiple physiologically distinct components. This enables integration with Boillet’s supply-side tank model by attributing Boillet’s total physiological power P_{physio} to underlying energetic drivers such as functional work, ventilatory effort, and accumulated metabolite load. The Lidar framework connects seamlessly to observed VCO₂, VO₂, and VE values and allows for richer inference of physiological burden beyond aerobic contribution alone.

Functional Work

The functional metabolic rate (MR_f) reflects the baseline energetic cost of external mechanical work (e.g., treadmill or cycling output). It is modeled as a linear function of mechanical power output P_{mec}, rescaled by body mass and a population average from the Lidar dataset:

$$MR_f = (A_f + B_f \cdot P_{mec}) \cdot \frac{m_{athlete}}{m_{Lidar}}$$

To match units across modalities, the slope coefficient B_f is scaled by the exercise efficiency (η) so that P_{mec} can be expressed in its observed form (e.g., running or cycling):

$$B_f = B_f \cdot \eta_{modality}$$

Accumulated Metabolites

The accumulated metabolite demand (MR_{acc}) captures energetic cost from lactate-related and other byproduct clearance processes. It scales nonlinearly with a variable x₄, representing the fraction of the anaerobic reserve that has been depleted, and is normalized to subject body mass:

$$MR_{acc} = (A_{acc} \cdot B_{acc} \cdot x_4 \cdot 1.03) \cdot \frac{m_{athlete}}{m_{Lidar}}$$

This form is a simplified version of Lidar’s formula, given that the original linear-quadratic distribution factor B_{acc} was so close to 1 (nearly completely linear). A_{acc} is a base metabolic factor, and B_{acc} modulates the contribution of linear versus nonlinear

accumulation effects. The scalar 1.03 accounts for slight correction in calibration bias based on Lidar's experimental tuning.

Physiological Power

Boillet's Pphysio is the fixed-eta, fixed-C1 total steady-state estimated metabolic demand on the body, based on the input mechanical work demand (assuming it was \leq the maximal mechanical output power of the athlete based on their current G tank depletion). The modification in this model is only that C1 is allowed to vary with substrate oxidation changes.

Aerobic Rate

Lidar's 'MRae' – the aerobic supply rate (tied closely to VO₂). The physiological power supplied by oxygen, defined explicitly from ventilatory data as:

$$MRae = (1.232 * RQ + 3.8149) * VO_2 * 4184/60$$

The above is adapted from McArdle et al. (2009). In Watts, with RER restricted to RQ [0.7, 1.0] as the relation is again to efficiency of substrate metabolism of fats vs carbs. VO₂ in L/min.

Individual testing with subject data revealed that RER didn't add much to the prediction of MRae, which was highly explainable on an per-individual basis by a linear regression with VO₂ (L/min) alone. Poor correlation between VO₂ and RER indicated limitations of any useful potential relationship during the test.

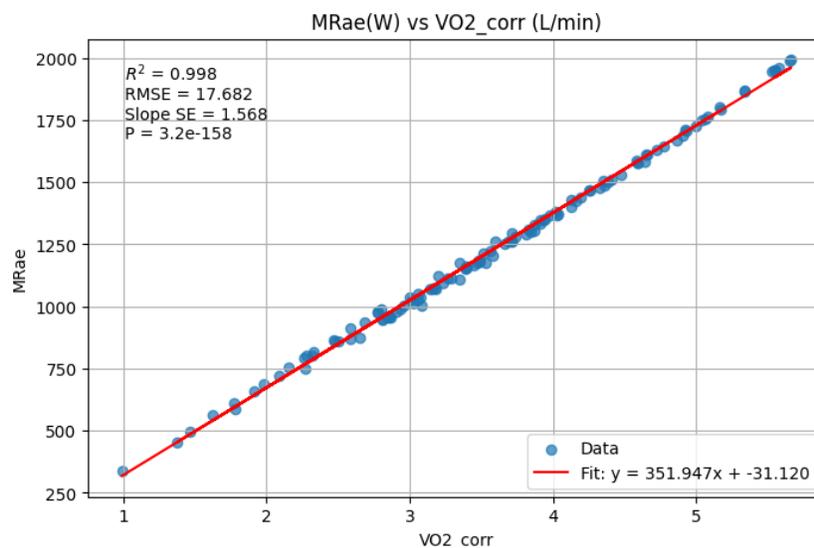


Figure 33: Metabolic rate (aerobic) versus (corrected) VO₂ (L/min) across a singular subject shows strong agreement.

For the digital twin construction, MRae is computed from simulated VO₂ metabolic power converted to L/min, which gets passed into an individualized regression (example shown above). These show up on each twin’s parameter list as Mrae slope and int(ercept).

Over all subjects, this fitting process seemed to be generally sound and consistent, in slope, intercept, and RMSE:

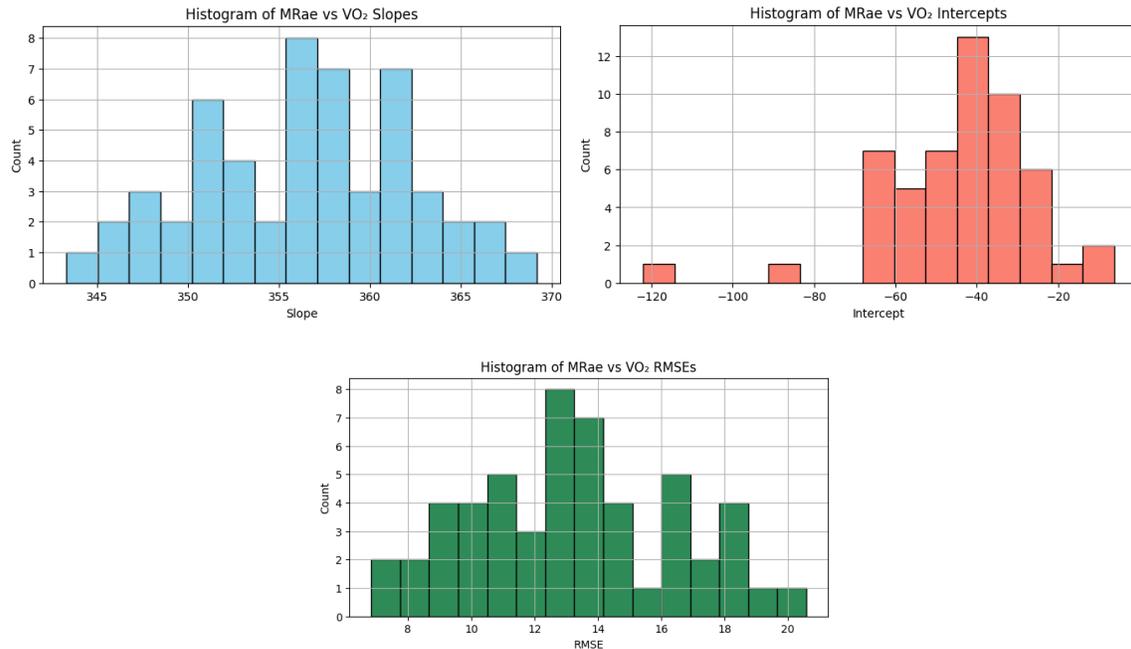


Figure 34A, B, and C: MRae equation parameters – slope (A), intercept (B), and RMSE (C) across all subjects fitted.

Ventilatory Demand

Lidar’s ‘MRve’ – metabolic demand due to ventilation (the energy requirement of the breathing muscles).

Lidar misstates the equation slightly, but the correct relationship that lines up with tuned values and physiological data (on average, used as a starting point) is:

$$MRve = 182 * \left(B \left(\frac{VE}{VE_{max}} \right) + (1 - B) \left(\frac{VE}{VE_{max}} \right)^2 \right)$$

Where $B = 0.93$ is the linear/quadratic distribution factor, MRve is in watts, and VE and VE_{max} are absolute minute- (current) and maximal ventilation values in L/min. Since total ventilation is not directly known from the twin, we attempt to use VO₂/VO₂max as a proxy, regressing both across all individuals, to make sure the relationship is identical and consistent, and across all datapoints, to make sure the proxy is accurate.

A linear fit between VE/VEmax and VO2/VO2max was unsuccessful – unsatisfactory individual R^2 values below 0.9 and slopes that ranged from ~0.8 to 1.1. Overall $R^2 = 0.87$:

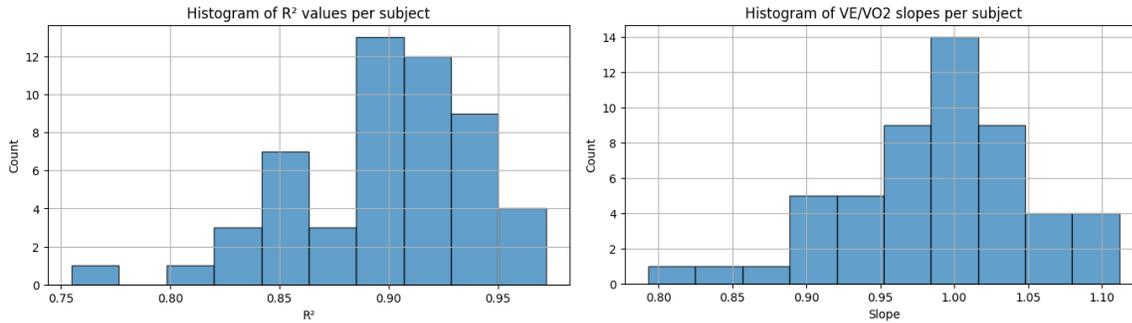


Figure 35A and B: R^2 and Slope ranges for attempted linear model for fractional ventilation and VO2.

After noting the general cubic trend, and changing the regression to a constrained least squares to require the intercept to go through zero (there is no oxygen where there is no air, and the opposite is assumed true for simplicity):

$$\frac{VE}{VE_{max}} = ax^3 + bx^2 + cx + 0 \text{ where } x = \frac{VO_2}{VO_{2max}}$$

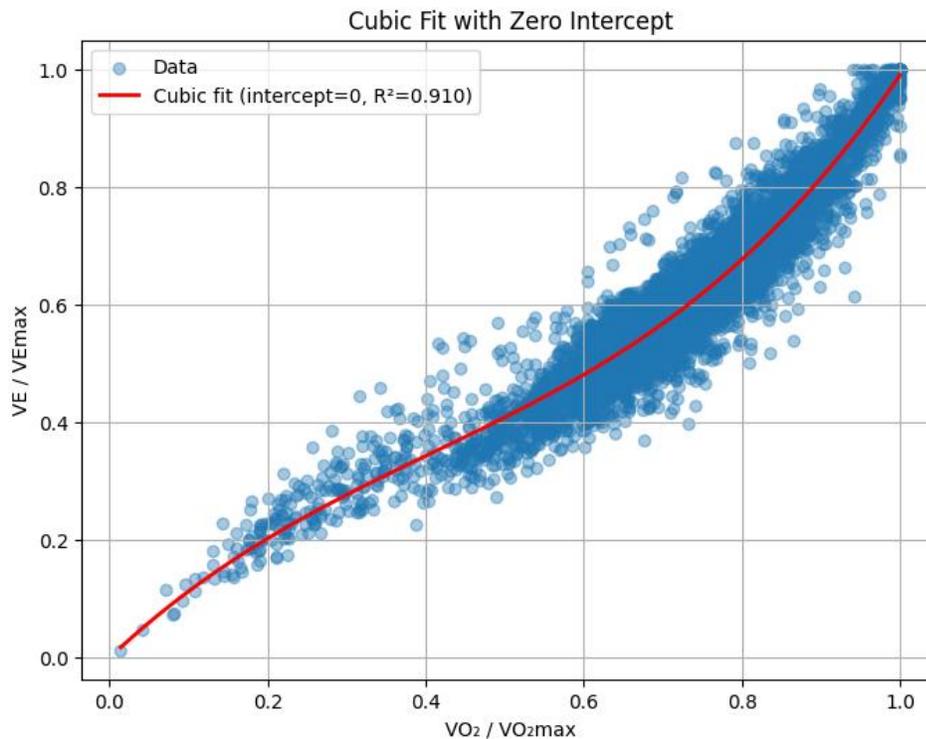


Figure 36: Cubic (intercept-constrained) fit of VE fraction versus VO2 fraction

The resulting conversion from VO2 fraction (x) to VE fraction (y) is:

$$y = 1.2499x^3 + -1.5287x^2 + 1.2687x$$

with a RMSE of 0.0507.

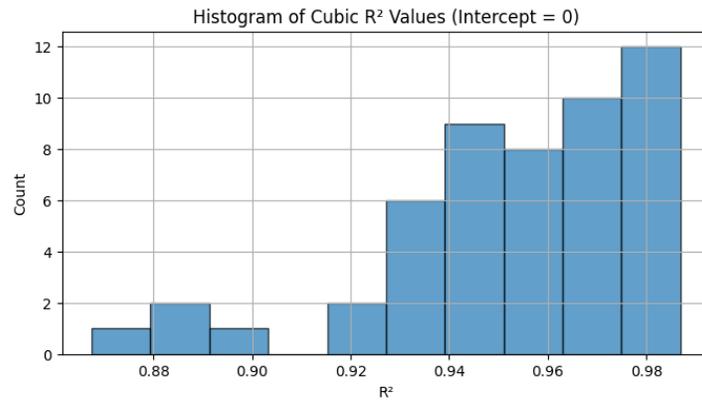


Figure 37: R^2 values across subjects with the new cubic fit.

Both overall R^2 and individual R^2 distribution was improved significantly with the cubic fit, especially given the highly variable nature of ventilatory data.

Replication of Lidar Protocol:

Based on best estimates of subject weight and metabolic characteristics (since many were only provided as averages across test subjects), we use the provided test power profile to simulate the response of an example athlete (as in Lidar (2023) Figure 7):

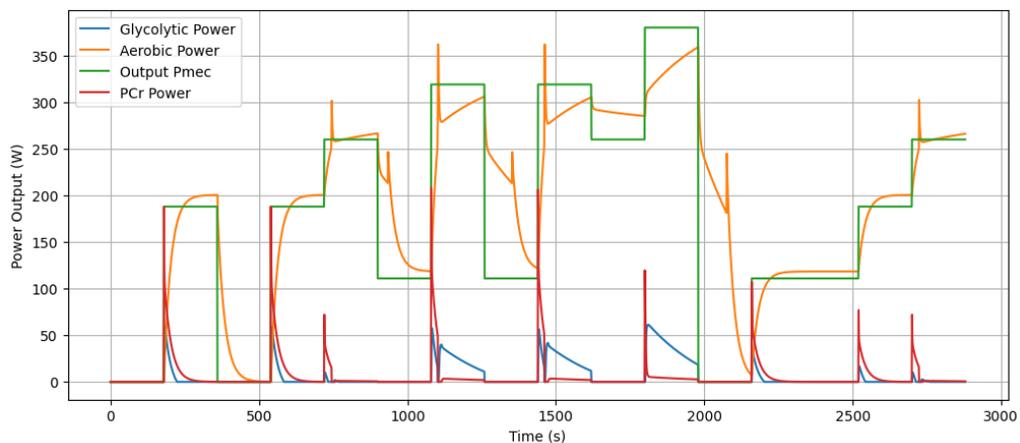


Figure 38: Power breakdown for Lidar protocol showing intermittent demand and varying aerobic and anaerobic contributions.

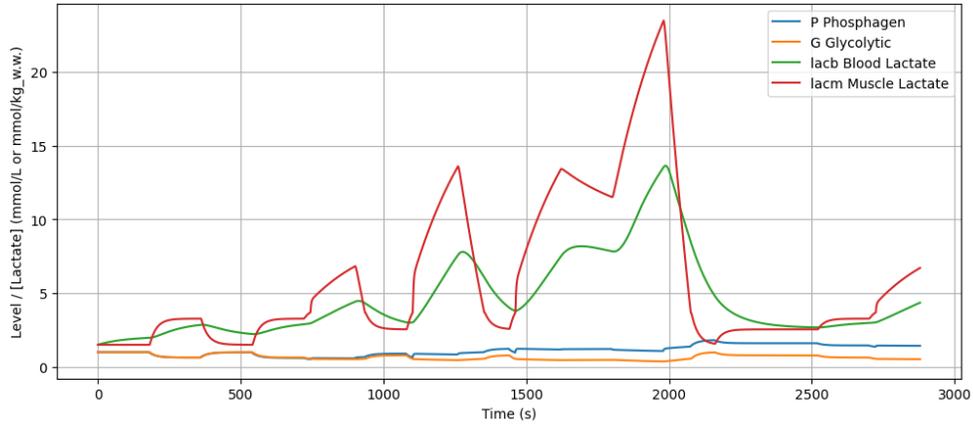


Figure 39: Tank and Lactate behavior for simulated protocol

Note: No Lidar comparison data, but the values seem physiologically plausible – high muscle and blood lactate at failure stage at ~2000 seconds.

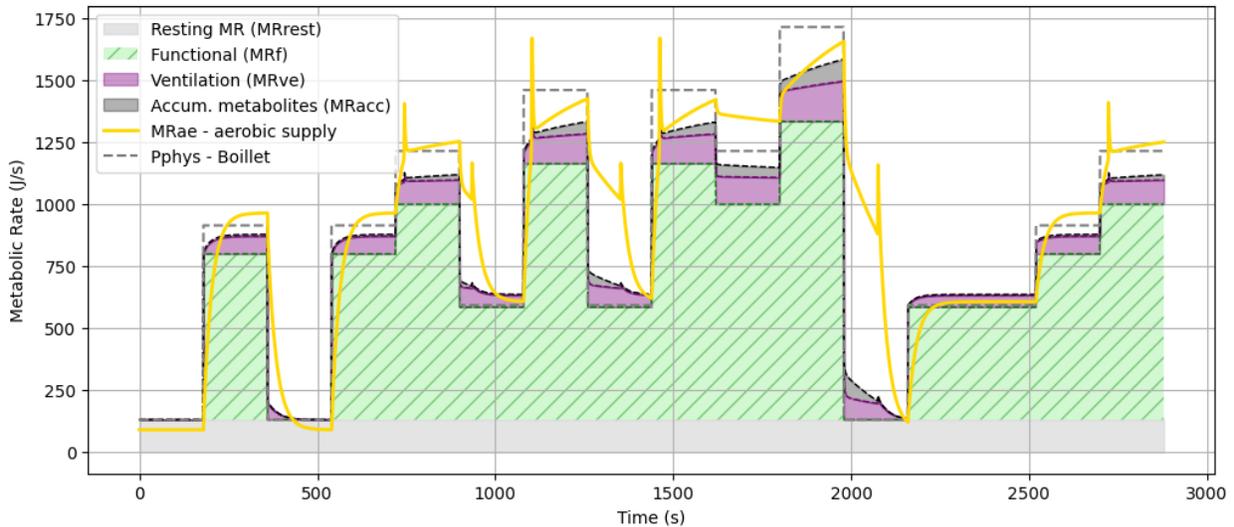


Figure 40: Metabolic breakdown during the aforementioned protocol replication, showing proper MRacc, MRf, MRrest, and MRve behaviors.

Notice the time-course of the ventilation and accumulated metabolite demands, and the aerobic supply attempting to match their increase. The occasional vertical spikes in MRae (yellow) are a byproduct of the coarse time steps of the discrete differential equation solver used, and although unsightly, do not take away from the steady-state estimates and overall fit.

Model deviations from power output are shown below, for comparison of aerobic power to estimated steady-state total power demand for the protocol imposed.

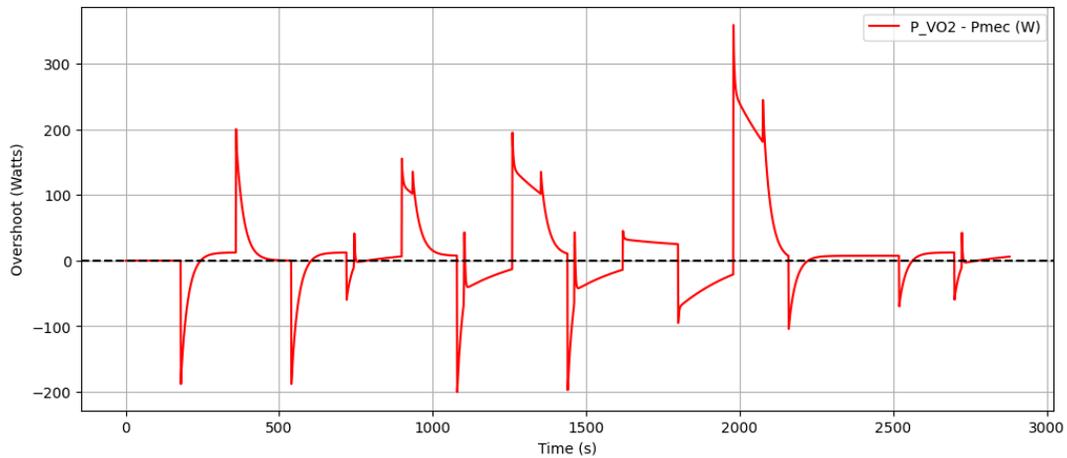


Figure 41: Aerobic power overshoot versus actual mechanical power demand for Lidar replication protocol.

Note the general convergence, especially in earlier stages where an aerobic steady state is reached (below or near LT1). The distinct large triangular regions that do present as significant deviations from the model are during recovery (ie; lower power demand) periods, as the VO2 and aerobic power drift back down, and are not considered problematic, as they match the general course of real measured VO2 – as can be seen in the “Measured MRae” curve of Lidar Figure 7A:

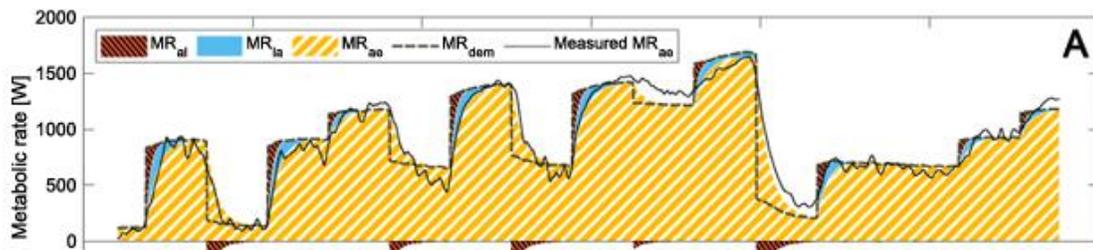


Figure 42: Lidar et al. (2023) Figure 7A – the deviation found by our model is the same for Lidar in the recovery periods – MRae exceeds MRdem (the power demand) as the aerobic engine winds down slowly.

The metabolic breakdown helps us understand some of the results of the Boillet-based simulations, especially in maximal effort settings where fatigue and failure are likely.

Consider the following replication of Boillet’s Cyclist 1 simulation of steady power in the severe domain to exhaustion, just above Critical Power (LT2):

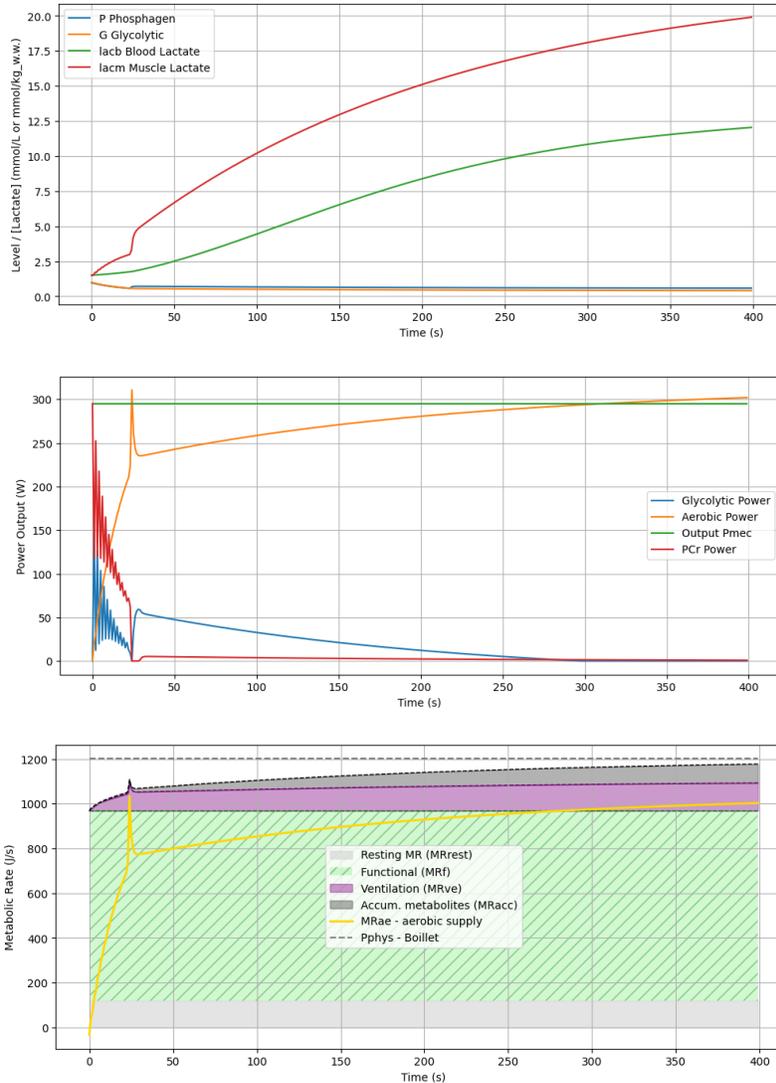


Figure 43A, B, and C: Cyclist 1 (from Boillet) simulation at >LT2 (CP), showing lactate dynamics (A), power breakdown (B), and Lidar metabolic breakdown (C), respectively.

Muscle lactate reaches the same rough point of nearing 20mmol/kg w.w. of muscle at Boillet’s estimated exhaustion point of ~350seconds (43A). From plot 43B, the aerobic power seems to fulfill the demand, potentially exceeding it, but we can see from 43C that the ventilation and accumulation of metabolites has increased the initial total power requirements such that the aerobic supply cannot keep up (only exceeding the functional power demand), and thus the glycolytic anaerobic system continues to contribute, as seen in the still-rising muscle and blood lactate values.

W' Estimation from VO2max Test

Boillet used a 3-min all-out trial to fit the Critical Power (CP) and W'non-oxidative (W') variables, along with other downstream anaerobic traits. The idea is that the glycolytic reserves are roughly fully depleted (the G tank is empty), thus due to a VO2max test's inherent to-failure maximal nature, we can approximate the theoretically fixed work that can be done above the critical power threshold by using LT2 and computing the work equivalent of the speed (Grade-Adjusted) and time done above it.

The confounding factor is that not all of this work done above threshold is non-oxidative, despite it often being called such. Obviously some of the work is done by increased oxygen consumption, as evidenced by LT2 being a fraction of VO2max. Thus the extra work above threshold accounted for by increased VO2 is factored in by an integration with speed-dependent eta and varying speed (along with increased ventilation power demands), and then the remaining work is attributed to non-oxidative work, requiring more power as metabolite accumulation occurs, and eventually leading to failure (Caen et al., 2024; Miller et al., 2023; Jones et al., 2008).

Thus a new algorithm had to be developed to account for the division of energy sources, with a discrete-time approximation of the integration.

W' Algorithm

Retrieve CP (Critical Power) parameters from economy (threshold) test

$CP = LT2 \text{ Speed} \times 0.44704$: convert mph to m/s

$$\dot{V}O_{2,CP} = LT2 \text{ VO}_2 \text{ in L/min}$$

Get time-series test data from max test

For each time step t , extract:

$$v_{GAP}(t) = \text{GAP speed (mph)} \times 0.44704$$

$$\eta(t) = \text{GAPspdperkW} \times \frac{0.44704}{1000}$$

$$\dot{V}O_2(t) = \text{measured or corrected VO}_2 \text{ in L/min}$$

$$C_1(t) = \text{estimated oxygen energy equivalent [J/ml]}$$

$$\Delta t(t) = \text{change in time in seconds}$$

Adjust VO2 for ventilatory and metabolite cost (from Lidar et al.)

Normalize VO2 to max:

$$x(t) = \frac{\dot{V}O_2(t)}{\max(\dot{V}O_2) \text{ for the subject}}$$

Convert to estimated normalized ventilatory rate (as fraction of max):

$$ve_pct(t) = 1.2499 \cdot x(t)^3 - 1.5287 \cdot x(t)^2 + 1.2687 \cdot x(t)$$

Convert to estimated ventilatory rate contribution:

$$MR_{ve}(t) = 0.088 \cdot [0.93 \cdot ve_pct(t) + 0.07 \cdot ve_pct(t)^2]$$

Estimate accumulated metabolite contribution, based on Lidar average VE/Acc ratios:

$$MR_{acc}(t) = 0.36 \cdot MR_{ve}(t)$$

Adjusted oxygen cost is thus:

$$\dot{V}O_{2,adj}(t) = \dot{V}O_2(t) \cdot (1 - MR_{acc}(t) - MR_{ve}(t))$$

Estimate excess oxygen cost above CP

$$\dot{V}O_{2,excess}(t) = \max(0, \dot{V}O_{2,adj}(t) - \dot{V}O_{2,CP})$$

Convert to energy rate:

$$P_{excess}(t) = \dot{V}O_{2,excess}(t) \cdot 1000 \cdot C_1(t) \cdot \frac{1}{60}$$

Convert to equivalent speed demand:

$$v_{O_2}(t) = \frac{P_{excess}(t)}{\eta(t)}$$

Get time points above CP

$$above_CP(t) = \max(0, v_{GAP}(t) - CP)$$

Compute aerobic and non-aerobic work above CP

Total aerobic contribution above CP:

$$W_{ox} = \sum_t \left(\frac{v_{O_2}(t) \cdot \Delta t(t)}{\eta(t)} \right) \div 1000$$

Total non-oxidative contribution above CP:

$$W_{non-ox} = \sum_t \left(\frac{(above_CP(t) - v_{O_2}(t)) \cdot \Delta t(t)}{\eta(t)} \right) \div 1000$$

Both are returned for visualization, but only $W'_{\text{non-ox}}$ is used for the estimation of A_G , which has a straightforward computation identical to Boillet's formulas, described in Parameter Estimation earlier.

Full code for both W' and A_G estimation is in the appendix.

W' Distribution Description

An example W' visualization derived from the above algorithm for work done above CP is shown:

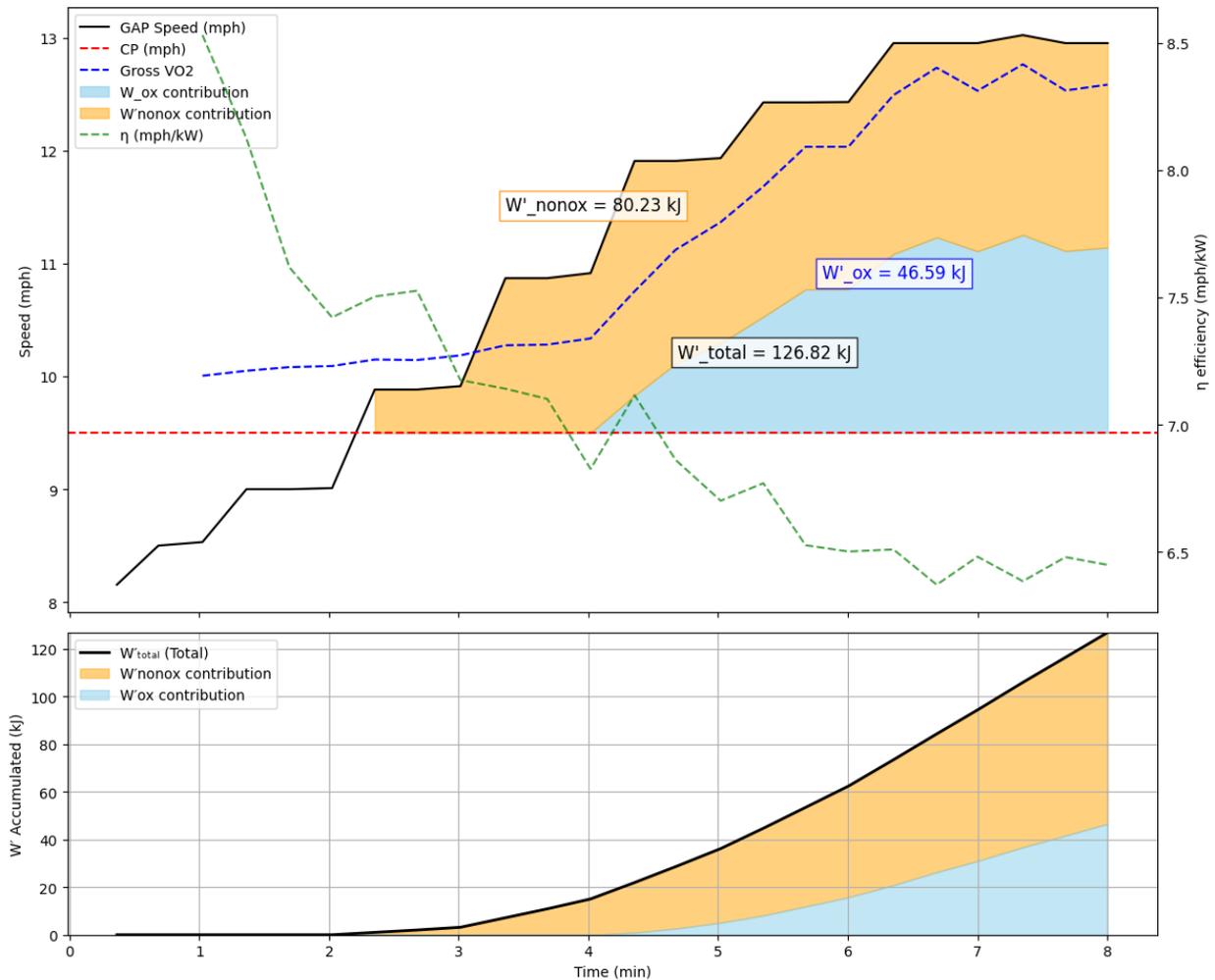


Figure 44A and B: GAP speed and W' component contributions to limited-work region over critical power, showing instantaneous power delivery distribution (A), and cumulative power source contribution (B).

Note the partial coverage of the supra-threshold work by aerobic increase – Gross VO₂ increase is dampened to the eventual, smaller W'_{ox} contribution by the ~8% and ~4% (near VO₂max, where VE is also high) that go towards covering ventilation power demands and clearing accumulated metabolites, respectively, instead of towards supplying running power.

Overall, for all subjects, the W' distributions were as follows:

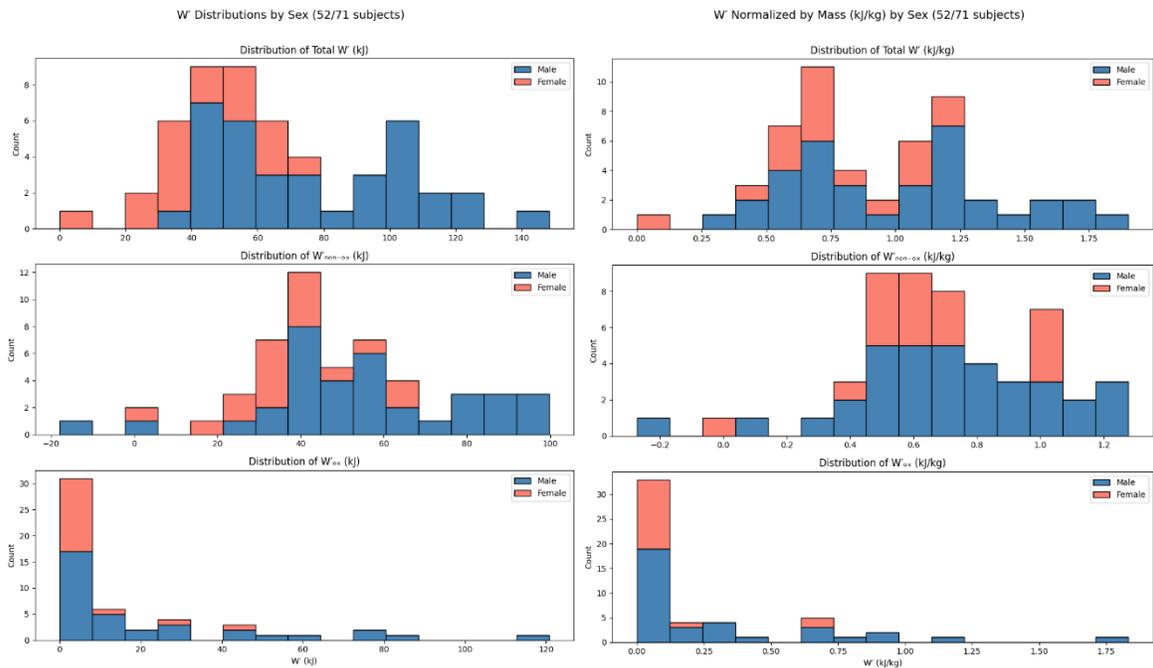


Figure 45A, B, C and 46A, B, C: Absolute (45) and weight-normalized (46) W' total (A), W'_{non-ox} (B), and W'_{ox} (C) distributions for the subset of subjects with both threshold and maximal test data, colored by sex.

Note that values of or near zero are likely due to threshold values (LT2 specifically) being estimated too high, either in the test protocol, or in the algorithmic determination, and can be considered limitations of the model.

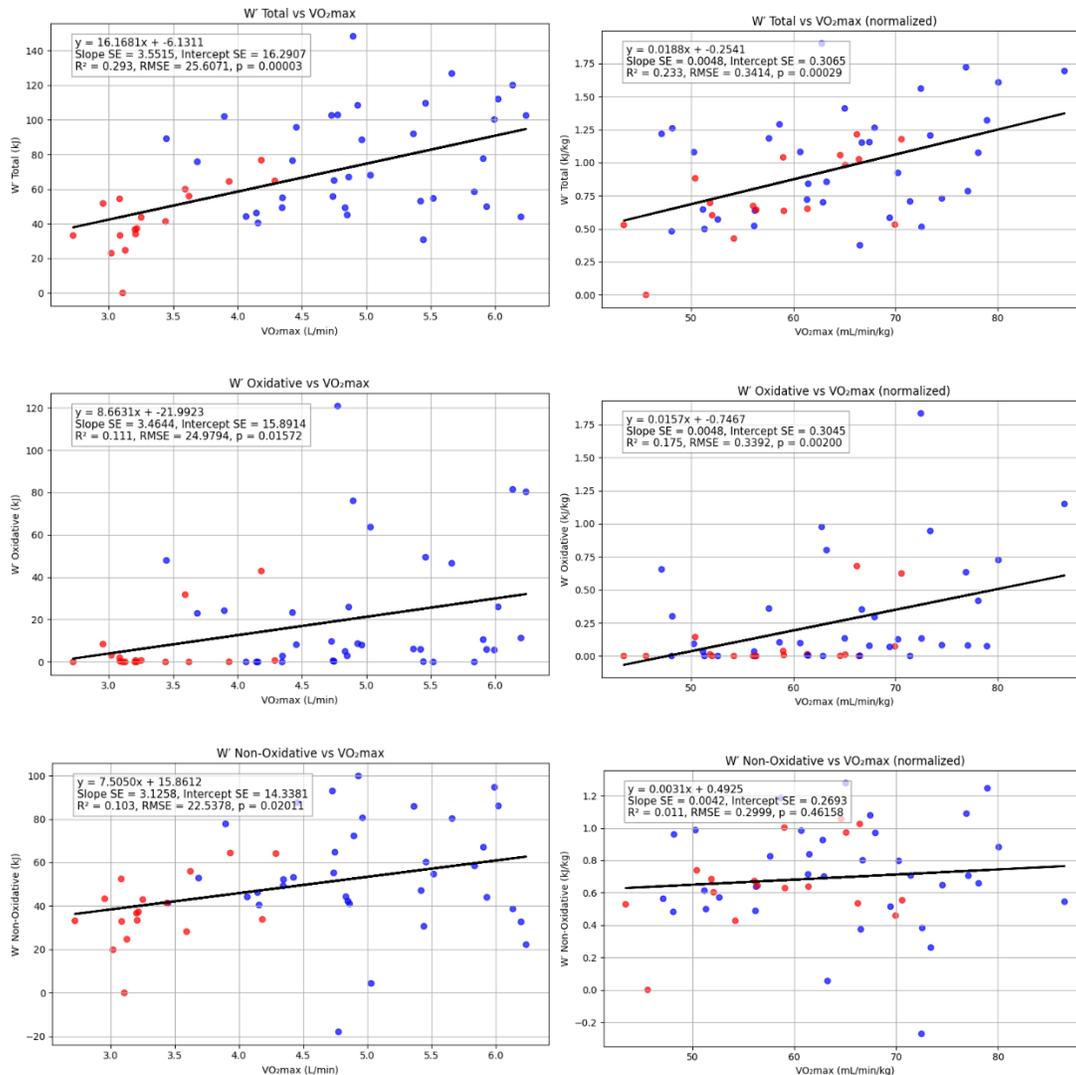


Figure 47A, B, C and 48A, B, C: Correlational assessments bet Absolute (45) and weight-normalized (46) W' total (A), W'non-ox (B), and W'ox (C) and absolute VO₂max (L/min – 47) and weight-normalized VO₂max (ml/kg/min – 48) for the subset of subjects with both threshold and maximal test data, colored by sex.

From the above, VO₂max generally correlates to more work able to be done above threshold. This could also be a product of the negative correlation between VO₂max and threshold expressed as a percent of maximal VO₂ – fitter people tend to have more “space” or capacity, above threshold, according to our data. These trends are weak and there isn’t a ton of data to suggest any strong direct causation here, especially given the derived nature of the W' variables. Women tend to have lower absolute W' values across the board.

In terms of relative (per-kilogram of body mass), there weren’t any super significant trends either - the correlations are weak at best, but there are some potential trends – higher

VO2max goes with higher oxidative potential over threshold, both absolute and per-kilo. The $p=0.10$ negative correlation we see between W' non-ox and VO2max could potentially warrant more study – to what extent is the lower W' non-ox caused by VO2 coverage of the demand versus just lacking anaerobic power? On a per-kilogram basis, women and men are less distinguishable in terms of estimated W' values.

None of the W' variables had significant relationships directly with speed at second threshold:

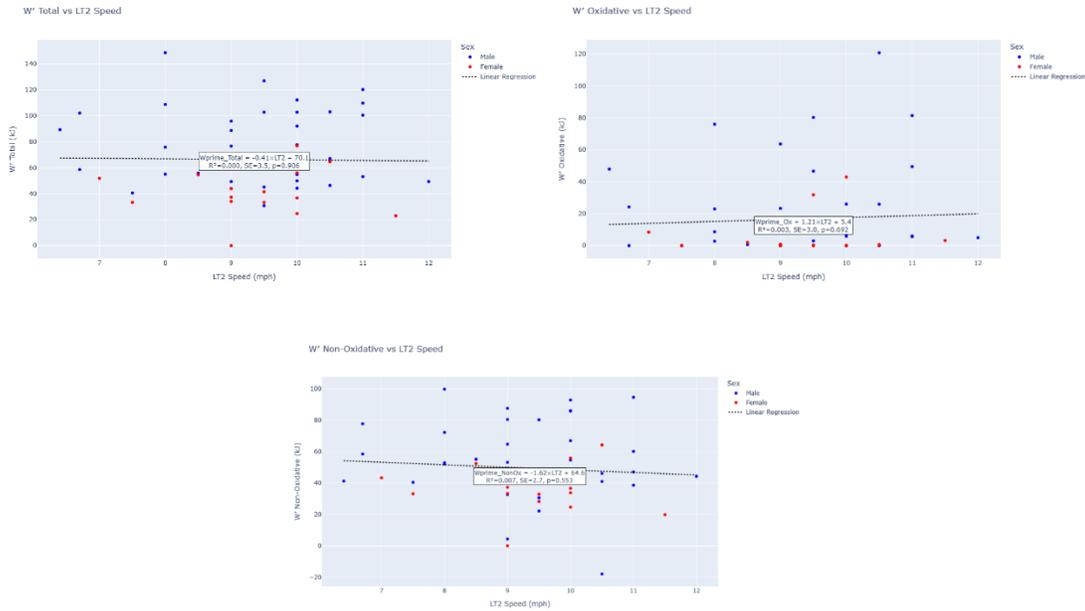


Figure 49A, B, C: W' total (A), W' ox (B), and W' non-ox (C) versus LT2 speed, with linear regressions fitted.

Notable that threshold speed has no significant correlation with W' . This emphasizes the divide between energetics and physical mechanical output, and emphasizes the need for a complete model – the anaerobic and top-end aerobic side of an athlete seems to have little bearing on the (mainly aerobically determined) second lactate threshold.

Results

Individual Amateur Runner Digital Twin

Using a case example of a digital twin for a moderately trained runner, with additional blood lactate data to validate the metabolite time-course.

Subject was chosen for expert-validated ventilatory and lactate thresholds and completeness of data – documented in real-time and cross-checked later.

Subject details

Weight: 73.6 kg, Age: 21 years old, VO₂max: 76 ml/kg/min

VT1 at 7.7 mph, VT2 at 9.5 mph (Stage 3, 7)

LT1 at 8 mph, LT2 at 9.5 mph (Stage 4, 7)

%VO₂max at LT1 = alpha = 0.600 = 60.0%

%VO₂max at LT2 = beta = 0.747 = 74.7%

Boillet model fitting

Table 3: Resulting subject parameter values from the twin auto-fitting processes described in Methods

Parameter	Value	Units
Aacc	114.0	W
PC_max	25.0	Mmol/kg w.w.
phi	0.300	
MO	2.040	kJ/s
eta(run)	6.931	GAP spd (mph) / kW
mraeslope	351.947	
mraeint	-31.120	
wgt	73.636	Kg
etaslope	-0.253	
etaint	8.820	
MG	10.567	kJ/s
MR	4.227	kJ/s
maxRER	1.029	
AT	26.281	kJ
AP	39.856	kJ
AG	471.220	kJ
lamb	0.445	

theta	0.420	
Wprimenonox	80.232	kJ
Wprimeox	46.589	kJ
Wprime	126.820	kJ
alpha	0.600	% of VO2max
beta	0.747	% of VO2max
musclemass	36.818	kg

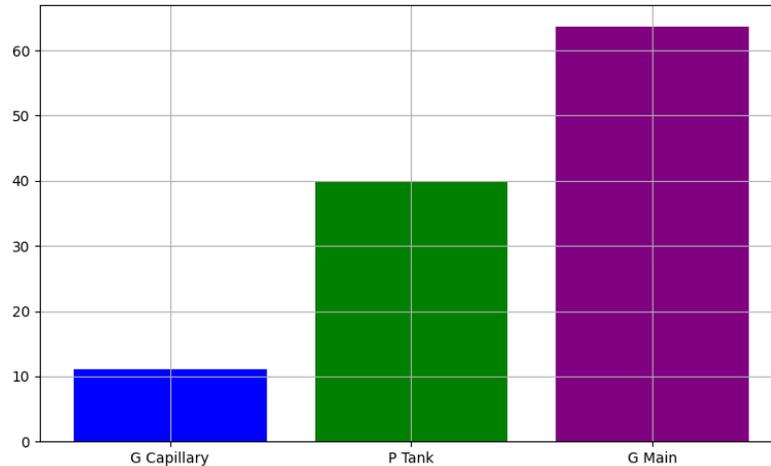


Figure 50: Non-oxidative tank capacity (total kiloJoules) breakdown for subject, showing relative sizes – the main glycolytic tank dominates in total energy storage.

Threshold and Max Tests

A digital twin created as per above was subjected to the threshold and maximal protocols of their own fitting data, and real measured blood lactate and measured VO2 were used for evaluation of the model fit, in addition to visual corroboration of expected model dynamics.

Measured VO2 was converted to MRae physiological power for direct comparison with modeled MRae. Since VO2/VO2max and VE/VEmax are so closely related, as found in the Energy Equivalents section, the model's performance on VO2 will be a good estimation of the VE predictive strength as well.

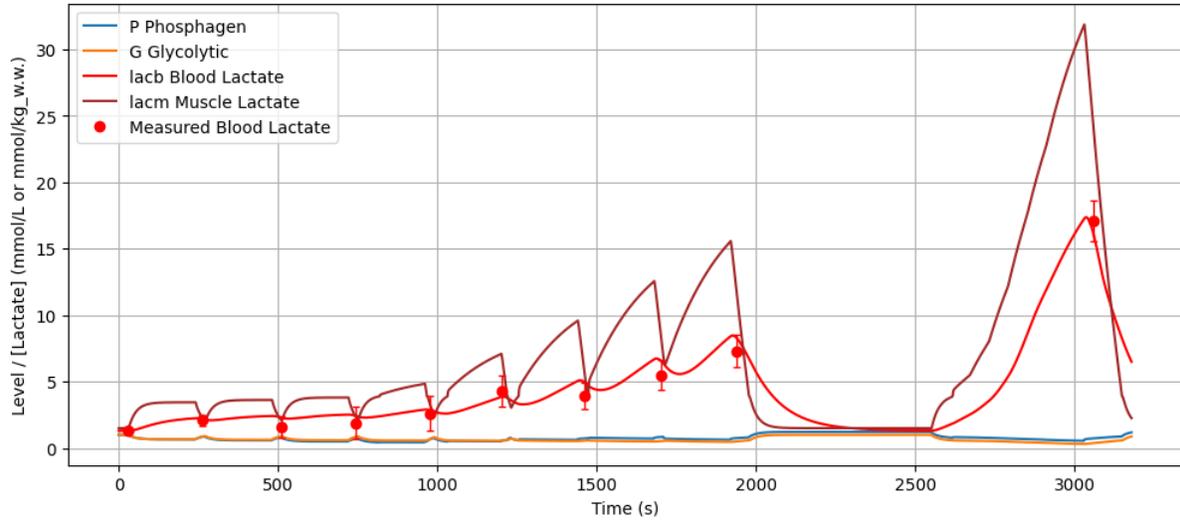


Figure 51: Muscle and Blood lactate (and tank dynamics) for simulated (lines) and real test data (dots – only Blood lactate) over the time course of the threshold and VO2max tests.

The model tends to slightly overestimate lactate at low and medium intensities. Perhaps this is a function of muscle lactate, or the population-style fit of blood lactate that is crudely applied to an individual’s lactate clearance dynamics (well-trained athletes may have better removal of lactate than the average person) – more sophisticated models are likely a good point of future modeling.

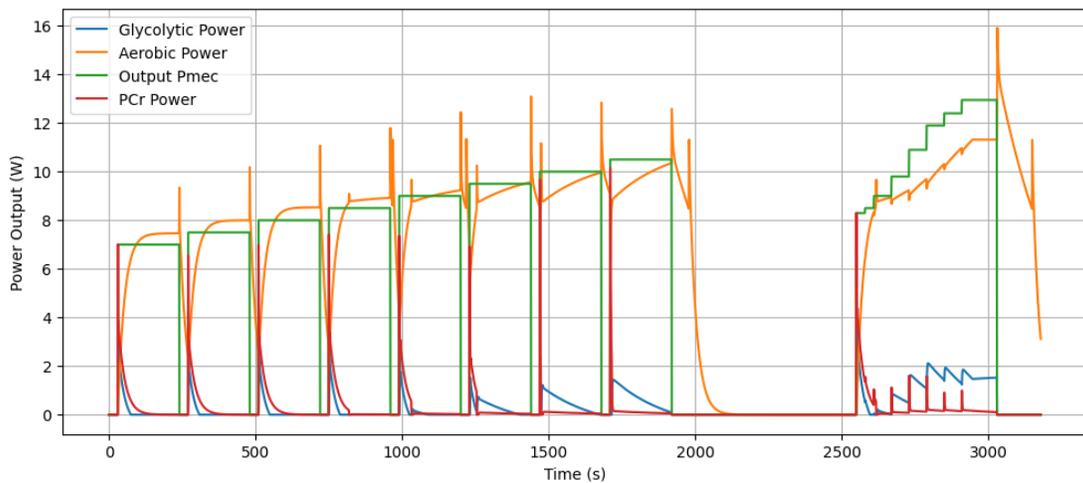


Figure 52: Power breakdown for subject over threshold and VO2max tests.

Again, ignore the yellow aerobic power spikes, they are after each work interval, where the applied physical work (speed) is reduced dramatically while the VO2 is still high and hasn’t come down yet, so the model thinks you are being really efficient (at low speed) and artificially spikes the power. This is a limitation of the computational logic, unfortunately.

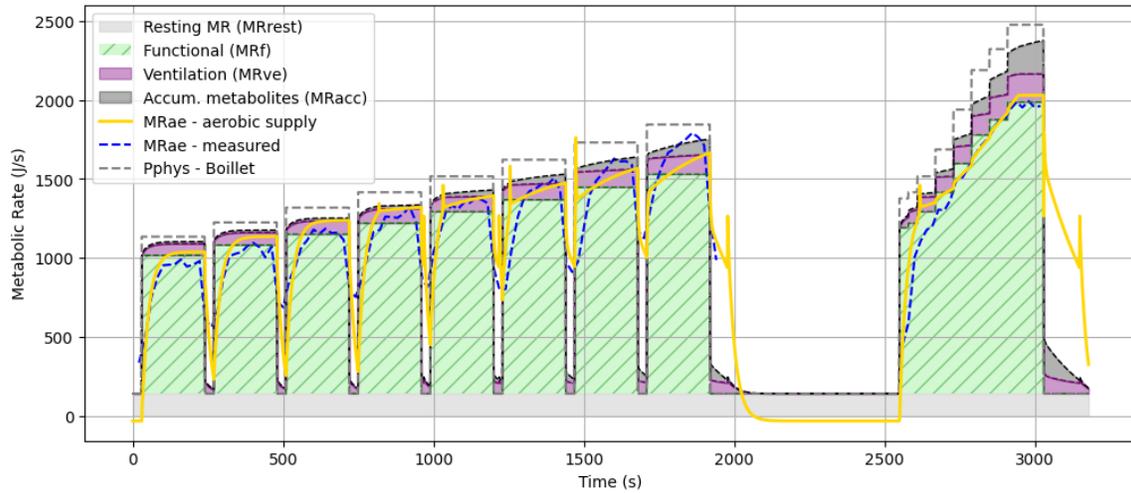


Figure 53: Metabolic supply and demand rates for subject over threshold and VO2max tests.

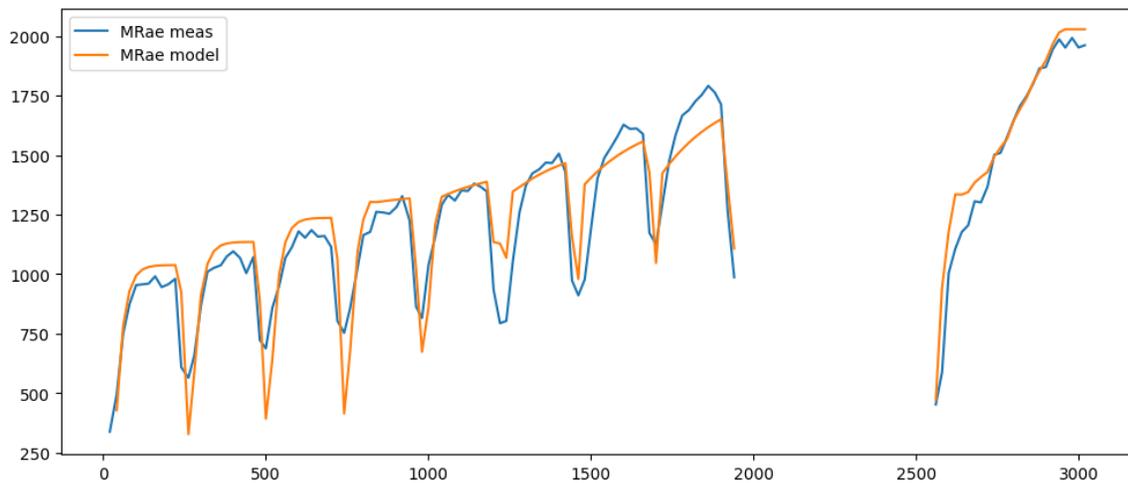


Figure 54: Extracted modeled and real (estimated) MRae from Figure 53, for accuracy comparison – both qualitative and quantitative. Visually the agreement seems decent.

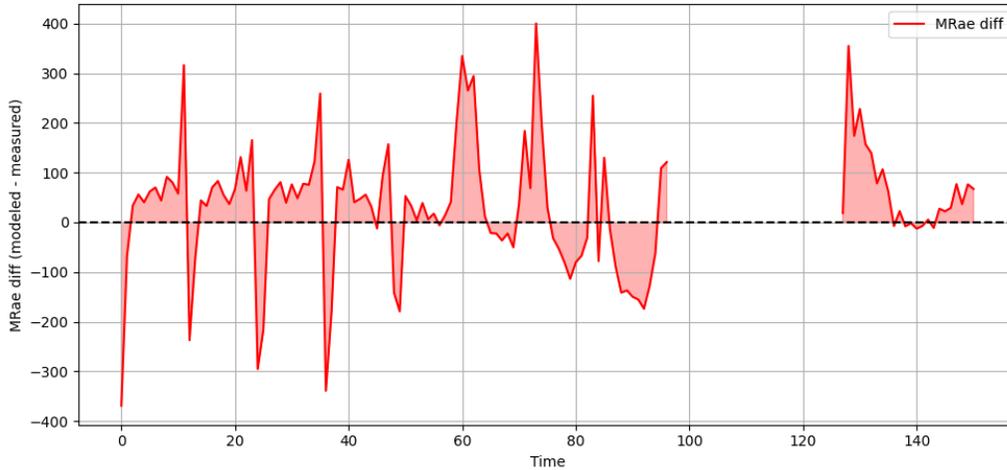


Figure 55: Modeled minus measured MRae (error) across threshold and VO2max tests

Visually, the model overestimates at low intensities (but perhaps at longer durations the body would find the same steady state (Caen et al., 2024)). At higher intensities there is some underestimation, although not for the vo2max test, which is interesting. This is potentially due to the MRae model not accounting for extra ventilation and accumulated metabolite power required at the higher intensities as explicitly. Further models may incorporate some sort of tuning to this extent. It is also possibly a result of the GAP algorithm overestimating speed adjustments for steeper gradients (which it seems to do, at least relative to other internet pace calculators). In this case, a more conservative GAP might result in an underestimated MRae at VO2max as well, similar to the later stages of the threshold test, for an overall underestimation of physiological aerobic power at intensities above LT1. Between stages (30s standing rest for lactate sampling) the model tended to overestimate the VO2 recovery down towards baseline levels more than was measured, mostly at lower intensities.

Several fit statistics were computed for the threshold-max data, including Mean Absolute Error, Root Mean Squared Error, Bias (Mean Error), Mean Absolute Percentage Error, and R^2 :

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| = 96.91 \text{ W}$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} = 131.4 \text{ W}$$

$$\text{Bias} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i) = 32.4 \text{ W}$$

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| = 10.4\%$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} = 0.87$$

Notable is that RMSE (131.4W) is 38.1% lower than Boillet's Critical Power testing model RMSE on the 3-min all-out ground-truth protocol, when converted to equivalent physiological power.

Within-stage real VO₂ (and thus MRae) variation was on the lower end of the 3-7% standard deviation percent of average, at 3.39%. Still, after converting 10.4% MAPE to RMSE (%) with the rough estimation that errors are normally distributed and symmetrical:

$$\text{RMSE \%} \approx 1.25 \text{ MAPE}$$

The remaining model error after controlling for data noise is still significant, at around 10%, but perhaps acceptable, given the early nature and quantity of estimations made on a relatively un-tuned model.

Steady-state trials

MLSS is the gold standard for maximal aerobic steady state, which is important for performance estimation for longer (>15min) duration maximal efforts – what intensity (speed or power) can be maintained? Usually this takes multi (3-7) day re-testing of 20-30 minutes that is costly, exhausting, and interrupts a lot of normal training and must be carefully controlled (Caen et al., 2024). Usually blood lactate levels are measured to determine (lack of) upward drift at each intensity tested.

Using the digital twin, constant efforts can be simulated at potential intensities instantly, to evaluate the potential lactate levels and other metabolic factors. Although early on some estimations put MLSS as commensurate with what LT2 tries to measure, more recent MLSS studies have found that it falls between LT1 and LT2 (both in intensity and lactate), sort of as a midpoint. Whether this is due to overestimation of LT2 from short-duration step protocols is debated but the general consensus is that MLSS is slightly lower. (Caen et al., 2022; Cerezuela-Espejo et al., 2018).

At simulated steady power (speeds: 7.5, 8.5, 9.5, 11.5 mph) for 20min:

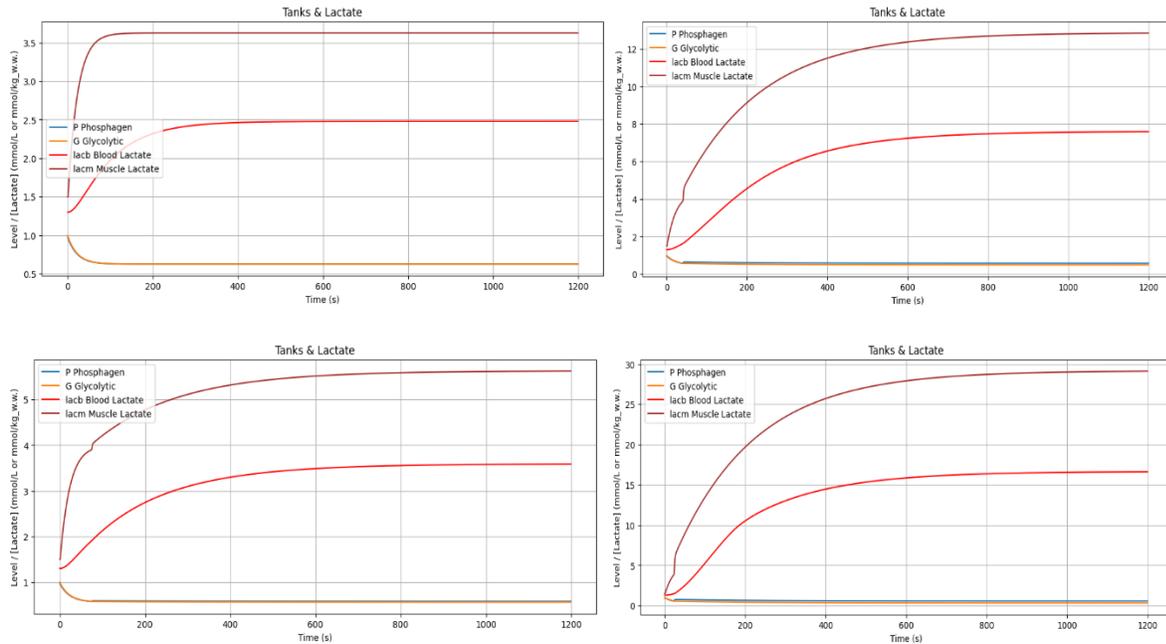


Figure 56A, B, C, D: Muscle and Blood lactate simulated curves for 20 minutes constant power at <LT1 (7.5mph, A-top left), >LT1 (8.5, B-bottom left), =LT2 (9.5, C-top right), > LT2(11.5, D-bottom right).

Note likely failure (or near) for the supra-LT2 test. Blood lactate values nearing maximal potential physiological values, as well as muscular lactate values.

The clear drawback here is that even above MLSS, at LT2, there is still a steady state reached (albeit at high lactate and nearer the end of the 20 minutes). In reality, a combination of reduced efficiency due to muscular fatigue and perhaps heat stress, along with accumulated metabolite demand and ventilation, might cause drift at LT2 over time (Caen et al., 2022; Zuccarelli et al., 2018; Bellinger et al., 2020). Another limitation of the model, and perhaps of the current body of knowledge on fatigue and durability, which is a rapidly emerging field of sports science.

53 Runner Cohort Digital Twin Summary

Overall, 53 subjects had both eco and max data. A few were discarded in the model fitting process for weird (often short) stage data and protocols, and a few were excluded for possessing an estimated W' that was negative.

For those with W' in valid ranges, but still ending up with negative AG (as sometimes was the case with early model testing, and for a few individuals), AG was set to a conservative 200 kJ, which tended to produce decent agreement. No further “hyper-parameter” tuning or correction was done.

Test data was appended together (eco + 10min rest + max) and aligned, and test speeds were converted to GAP speeds and processed to a full 1-hz power sequence input for the digital twin. Twin parameters were estimated from test data, and fed into the twin simulation as well. Similar to the individual twin evaluation, VO_2 and VCO_2 time course were compared to ground-truth ventilatory measurements.

See code in appendix for full data pipeline.

Results of parameter fitting and selection on subjects:

Table 4: Average Across-Subjects Parameter Summary

	Mean \pm SD	
PC_max	25.000 \pm 0.000	(left as constant)
phi	0.300 \pm 0.000	(fixed by Boillet)
MO	1.590 \pm 0.363	(VO_{2max} , in kJ/s)
eta(run)	8.184 \pm 1.788	(GAP mph/(kW))
mrae_slope	356.507 \pm 5.806	
mrae_int	-44.743 \pm 18.514	
weight	71.670 \pm 13.546	kg
eta_slope	-0.258 \pm 0.175	
eta_int	10.084 \pm 2.651	
MG	10.285 \pm 1.944	kJ/s
MR	4.114 \pm 0.778	kJ/s
AT	23.062 \pm 5.364	kJ
AP	38.791 \pm 7.332	kJ
AG	182.125 \pm 222.323	kJ (before correction(s))
lambda	0.391 \pm 0.079	
max_RER	1.048 \pm 0.090	
theta	0.476 \pm 0.069	
Wprime_nonox	50.157 \pm 21.884	kJ
Wprime_ox	14.073 \pm 22.104	kJ
Wprime_total	64.230 \pm 29.332	kJ
alpha	0.680 \pm 0.098	% of VO_{2max}
beta	0.828 \pm 0.102	% of VO_{2max}
muscle_mass	35.835 \pm 6.773	kg

Overall estimated alpha and beta values fall into accepted literature values for LT1 and LT2 as percents of VO2max (~68% and ~83%, respectively) (Kim et al., 2021; Cerezuela-Espejo et al., 2018). Maximal RER also falls into expected range of >1.0 at near-maximal efforts.

Parameter correlations across subjects were notable for a few pairings:

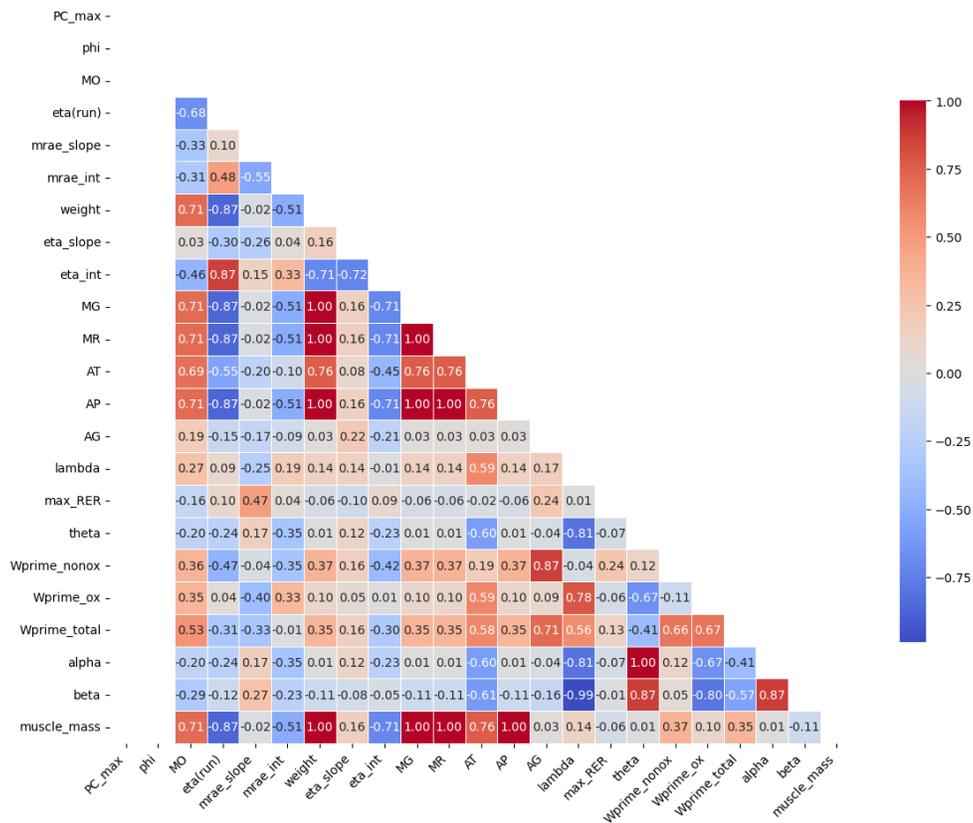


Figure 57: Parameter correlation heatmap for all subjects fitted successfully with digital twins

Correlations of 1.00 clearly indicate where direct relations exist in the estimation process, such as for muscle_mass and AP, MR, MG, etc. PC_max and phi were fixed values in the fitting process.

Higher theta, beta, and alpha were strongly positively correlated, as expected given their importance in determining the aerobic power (indirectly, by determining the delay of the onset of anaerobic power), and were negatively correlated with lambda, which tended to be lower (smaller) as the aerobic factors increased. Notably eta_int (the intercept term of the efficiency regression) tended to be negatively associated with anaerobic markers such as MG, MR, AT, and AP, as well as muscle mass – lighter and more aerobic athletes tended to be more efficient. The moderate positive association between MO and Wprime_total was also notable – higher VO2max goes along with higher work able to be done above

threshold, although the oxidative component of that W' had a smaller correlation. The moderate negative correlation between $\eta(\text{run})$ and MO seems to confirm some studies' findings that fitter athletes tend to be less efficient (or that less efficient athletes compensate by being fitter) (Mogensen et al., 2006; Lopez, 2023).

Athlete Subtype Classification

To see if the created digital twins could be separated into classes, just as we might categorize different types of athletes with a combination of subjective and objective traits (ie: fast, endurant, powerful, graceful), the model parameters were cast into Principle Component Analysis (PCA) to try to reduce the dimensionality to something more understandable than 15+ varying parameters.

Although not a majority of features have high correlations, the new orthogonal axes that PCA projects the data onto capture the maximum variance across the subjects' features, by decomposing the covariance matrix, and keeping the (2, in this case) largest eigenvalues and corresponding eigenvectors \mathbf{V} , that define the projection:

$$\mathbf{Z} = \mathbf{X}\mathbf{V}_k$$

Where $k=2$, and X is $n \times p$ and is the mean-subtracted original parameter-subject matrix (n subjects, p params each).

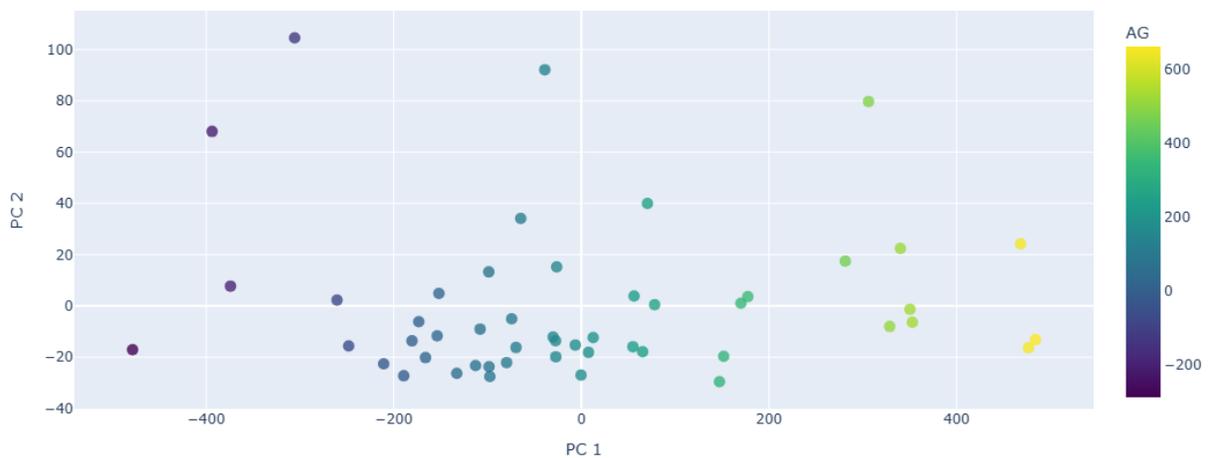


Figure 58: Principal Component Analysis projection onto PC1 and PC2, of the subjects' parameter values, colored by AG value.

Upon examining the underlying parameter data, it seemed that AG (anaerobic glycolytic capacity) was explaining a lot of the spread across PC1, so the color gradient was added to confirm. Since the negative AG values are pre-correction and likely represent an error somewhere in the model fitting process, this variance is sort of fake and model-induced, but nonetheless the sprint capacity of an athlete is undeniably a defining factor (Wackwitz

et al., 2025). The cumulative variance plot adds to the confirmation that much of the athlete parameter variance is explained by this PC1:

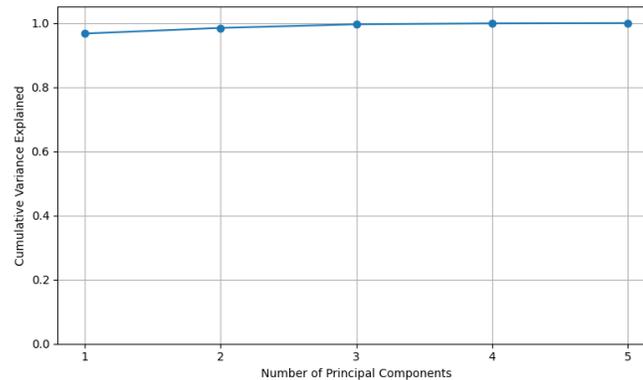


Figure 59: Cumulative explained variance by first five principal components

Nearly all (near 100%) of total variance in athlete parameters is explained by the first PC.

Overall model predictive results:

VO₂ (L/min) was chosen as the primary metric for model evaluation, being at the core of both ventilation and aerobic power, and capturing much of the essence of perceived endurance effort – it has intuition for many (Zhang et al., 2021; Van Der Zwaard et al., 2021). It also allows for 1:1 comparison with the Lidar model, who used MRae (the metabolic equivalent). Modeled versus measured VO₂ average errata are summarized below. Typical VO₂ values depend on weight, fitness, and intensity, but range between 2-5 L/min for most. Within-stage steady-state variation of VO₂ was on the order of 3-7%, which did not significantly change the model errata, like in the individual digital twin case study in the previous section.

To examine whether the model had a tendency to estimate certain intensities or types of tests better than others (as potentially indicated by the individual case study of the previous section), model VO₂, VCO₂, and MRae outputs were split into 3 sections: A (first half of the threshold test), B (second half of the threshold test), and Max (VO₂max test portion). Separate error metrics were computed in addition to the grouped overall errors.

Table 5: Mean +- SD of overall across-subjects VO2 model minus measured error statistics

	Overall	A-1st Half Eco	B-2nd Half Eco	VO2Max
MAE	0.278 ± 0.116	0.235 ± 0.091	0.261 ± 0.156	0.363 ± 0.139
MSE	0.154 ± 0.132	0.113 ± 0.107	0.145 ± 0.187	0.223 ± 0.165
RMSE	0.365 ± 0.143	0.311 ± 0.128	0.334 ± 0.184	0.444 ± 0.162
Bias	0.219 ± 0.133	0.209 ± 0.099	0.176 ± 0.200	0.299 ± 0.206
MAPE	11.905 ± 7.219	12.448 ± 9.937	11.571 ± 15.450	12.018 ± 4.670
R ²	0.664 ± 0.236	0.605 ± 0.163	0.481 ± 0.576	0.441 ± 0.541

VCO₂, although also modeled, was one more estimation removed from reality (requiring an RER estimation in the model) and thus was less relied upon, especially given the difficulty of estimating RER.

Overall VO₂ standard deviation of the RMSE was ~39.2%, indicating moderate reliability in line with Lidar’s 38.5% for their similarly intermittent test protocol (P3), though they found lower MAPE at around ~9%.

The VCO₂ errors are quite similar to those of VO₂, if a bit higher:

Table 6: VCO₂ Error Metrics Summary

	Mean ± SD
MAE	0.350 ± 0.190
MSE	0.263 ± 0.323
RMSE	0.459 ± 0.228
Bias	0.184 ± 0.239
MAPE (%)	14.784 ± 9.328
R ²	0.535 ± 0.582

For the above VO₂ model errors, pairwise T-Tests were computed for likely significant mean differences, and ranked in order of increasing p-value (full table in appendix):

Table 7: T-Test Significant Mean Results for VO₂ model error differences between test segments

Metric	Comparison	p-value	Significance
MAE	Overall vs Max	0.0000	***
MAE	A vs Max	0.0000	***
MAE	B vs Max	0.0000	***
MSE	Overall vs Max	0.0000	***
RMSE	Overall vs Max	0.0000	***
MSE	B vs Max	0.0000	***
MSE	A vs Max	0.0000	***
RMSE	A vs Max	0.0000	***
RMSE	B vs Max	0.0000	***
R ²	Overall vs Max	0.0000	***
MAE	Overall vs A	0.0001	***
RMSE	Overall vs A	0.0002	***
Bias	Overall vs Max	0.0006	***
Bias	A vs Max	0.0008	***

Bias	B vs Max	0.0011	**
R ²	Overall vs B	0.0016	**
MSE	Overall vs A	0.0052	**
RMSE	Overall vs B	0.0088	**
Bias	Overall vs B	0.0109	*
R ²	A vs Max	0.0324	*
MAE	Overall vs B	0.0727	*
R ²	Overall vs A	0.0928	*

Notable the VO₂max segment was significantly different than overall in terms of MAE, MSE, RMSE and Bias, as were individual A- and B-Threshold sections, which were also lower. R^2 trended slightly downwards between modeled and real VO₂ as the test (and intensity of the test) progressed, although the low value is important to consider alongside the MAPE, which hovered around ~11%.

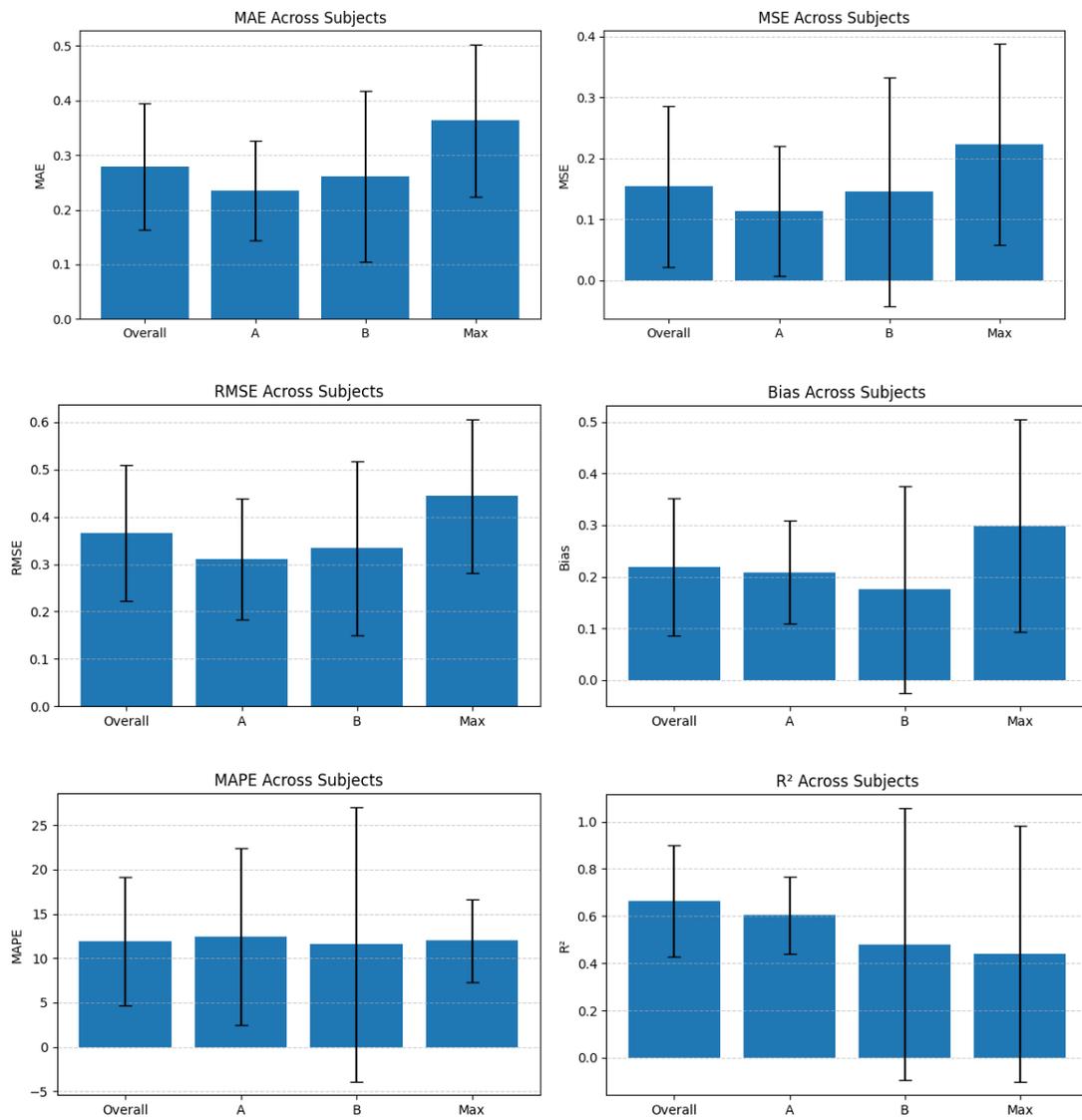


Figure 60A, B, C, D, E, F: Graphical representation of modeling errors across test segments for VO2 modeled versus measured data (previous page).

These results can seem less than encouraging without a visual understanding of the typical measurement variations and overall trends of the model.

Some of the overall positive bias between the model and real VO2 data – ie: overestimation may be due to the fact that measured VO2max was often a briefly-hit value, perhaps errant and often not sustained, especially for less-trained subjects. Thus basing the elite-cyclist model may have overestimated the ability to meet power demands aerobically for the running subjects in question as a whole.

Tank Simulation Animation

A fun outcome of this study is a real-time animation of the dynamics modeled, to further develop intuition and rapidly test ideas and parameter choice impacts. Currently set up as a locally-deployable script to run from a folder in a browser with Streamlit, it should be up and running on github shortly. Code in appendix.

🔒 Lock one to auto-calculate:

Power Torque Cadence

Power (W)	Torque (Nm)	Cadence (RPM)	Update Intvl (s)
282.74 - +	30.00 - +	90.00 - +	1.0 - +

MO (W overhead?)	θ (rad or deg)
1.34 - +	0.43 - +

(RE)Start Simulation

Stop and Clear Simulation

Pause and Save Simulation

Resume Simulation

Figure 61: Mini-dashboard showing a few potential parameters to live-tune the digital twin

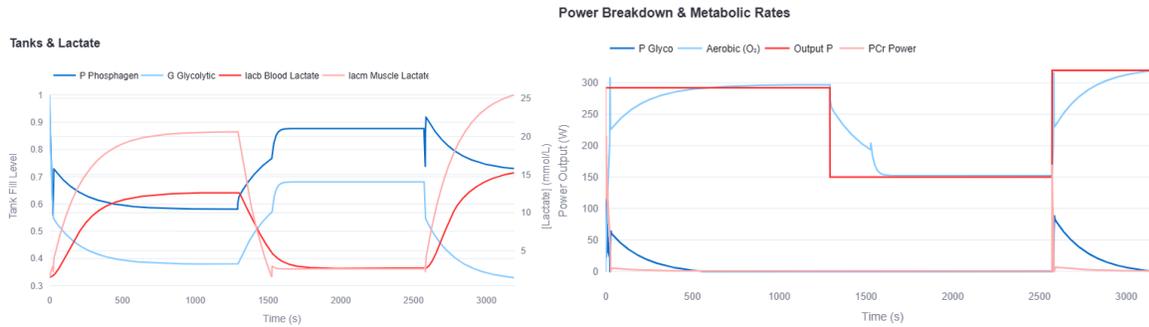


Figure 62A and B: Screenshot of live-output tank, lactate, power breakdown data streams

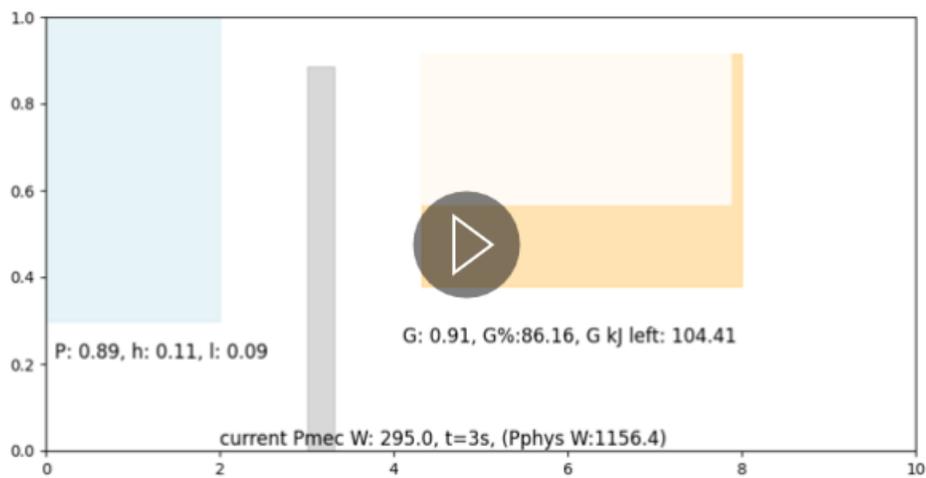


Figure 63: Tank animation from a Constant power test at above Critical Power, demonstrating desired asymptotic behavior of Glycolytic (G) tank. Simulations above used Cyclist 1 (from Boillet) parameter data.

The potential for this to live-simulate races and team-events is enticing, to be able to see effects in real-time and gain strong intuition as well as predictive power for better performance.

Discussion

This study presents a comprehensive digital twin framework capable of simulating physiological responses to endurance exercise using interpretable, data-driven models. By extending Boillet's three-tank theory of energy partitioning, the system introduces multiple novel components: dynamic mechanical and oxygen efficiency estimates, embedded lactate kinetics, and power scaling from grade-adjusted pace (GAP). The digital twin bridges a key gap between muscle-level theory and real-world treadmill gas exchange data, producing a time-resolved, multi-system portrait of endurance energetics.

One of the primary modeling innovations lies in the incorporation of dynamic efficiency terms. Rather than assuming fixed mechanical efficiency (η), the model allows this parameter to vary as a function of speed and fatigue, reflecting real-world shifts in substrate usage and muscular economy. Similarly, oxygen-metabolic efficiency (C_1), representing the volume of oxygen required per joule of metabolic work, is updated continuously during simulation, guided by inferred RER dynamics. Together, these variable efficiencies allow the model to reproduce the nonlinearities that define real exercise metabolism.

Another strength of the framework is its ability to reconstruct physiological thresholds without requiring invasive measurements. Using changes in VE/VO_2 , VE/VCO_2 , and RER slope, the model identified LT1 and LT2 inflection points with minimal manual intervention. This enables subject-specific calibration and facilitates high-throughput parameter estimation from treadmill tests alone. Across the cohort of 53 runners, the estimated LT1 and LT2 percentages of VO_2 max (mean $\alpha = 68\%$, $\beta = 83\%$) aligned with values from the broader literature (Kim et al., 2021; Cerezuela-Espejo et al., 2018).

To validate internal dynamics, the model was tested against VO_2 and blood lactate profiles for a well-characterized amateur runner with expert-validated thresholds. The twin accurately captured the rise and plateau of lactate under increasing workloads, with minor overestimations at lower intensities likely due to population-level assumptions about lactate clearance. Importantly, model VO_2 outputs closely followed measured values, with average MAPE of $\sim 11\%$ and RMSE of 0.365 ± 0.143 L/min across all subjects. These values fall within the error margins reported by Lidar (2023) for comparable treadmill protocols and represent a strong result given the model's complexity and minimal tuning.

Notably, the model supports physiologically grounded estimation of W' —the total anaerobic work capacity—and its oxidative versus glycolytic components. For instance, the case study subject exhibited a W' of 126.8 kJ, partitioned into 80.2 kJ non-oxidative and 46.6 kJ oxidative components. Such distinctions, grounded in empirical VO_2 max test data

and muscle–blood lactate regressions, enable more granular fatigue modeling than conventional critical power frameworks.

The model also supports inference from incomplete or noisy data, a frequent challenge in both field sport and clinical settings. Using only weight, VO_2 max, and GAP-estimated speeds, it is possible to simulate dynamic VO_2 , lactate accumulation, and ventilation profiles with plausible internal consistency. In this way, the digital twin can flag implausible threshold claims, reveal RER artifacts, or estimate fatigue under hypothetical pacing strategies. This positions the model as a powerful inverse inference engine, transforming sparse data into interpretable physiological narratives.

Still, several limitations remain. The discretized Euler solver produced some transient artifacts, especially during rapid transitions and short bouts (<20s), where numerical instability could yield brief overshoots. The lactate kinetics were also simplified, with a single saturation lag term governing clearance from muscle to blood, instead of more realistic known dynamics of production and clearance volumes (Bartoloni et al., 2024). More physiologically accurate models might incorporate perfusion, bicarbonate buffering, or liver metabolism. Furthermore, subject-specific values for muscle mass, PCr content, and tank widths were inferred from population means, potentially obscuring finer individual variability. Finally, no heart rate, thermal, or perceived exertion loops were integrated, leaving important feedback mechanisms unmodeled.

Despite these caveats, the digital twin robustly captured the essential dynamics of endurance exercise across a heterogeneous cohort in a new sport and provides a flexible base for future physiological modeling with lots of room for improvement.

Conclusion

This thesis introduced and validated a physiologically grounded digital twin of human endurance, linking Boillet’s muscular tank dynamics with treadmill-derived ventilatory data from 53 runners. Through layered extensions—dynamic efficiency modeling, lactate kinetics, and empirical VO_2 – VCO_2 coupling—the simulation system was able to reproduce not only key thresholds but also full time-course dynamics of fatigue, oxygen deficit, and energetic breakdown. More than a curve-fitting algorithm, the model serves as a hypothesis-generating tool and interpretive framework. It makes explicit the biophysical costs of exceeding LT2, shows how fat versus carbohydrate use shapes VO_2 demands, and highlights why even subtle changes in efficiency can alter fatigue onset. For instance, above-threshold efforts simulated at 11.5 mph showed non-recoverable lactate

accumulation and muscle tank depletion, illustrating the risk of pacing errors in high-stakes events. Across the cohort, observed VO_2 modeling errors were consistent with known measurement noise, suggesting the model faithfully reflects underlying physiology.

Looking ahead, the framework developed here provides a launching point for several impactful extensions. First, by treating the digital twin as an inverse problem, it becomes feasible to estimate VO_2max or lactate thresholds from submaximal effort data alone—especially valuable in clinical or large-cohort settings where maximal testing is impractical. Similarly, refining lactate and CO_2 clearance modeling through dynamic VCO_2 signals or individualized time constants would improve the realism of fatigue onset and recovery predictions. The model's structure is amenable to expansion: integration of heart rate, thermal strain, or perceived exertion feedback loops could yield a fuller picture of performance regulation under real-world stressors.

From a statistical perspective, the model is well-suited for probabilistic extensions. Bayesian estimation or Kalman filtering could be introduced to fit noisy metabolic cart data more robustly and to recover subject-specific parameters in the presence of uncertainty. This would support automated tuning of tank widths or efficiencies, enabling more general application across populations. In practical domains, the model could underpin race strategy simulators, personalized coaching tools, or diagnostic platforms for monitoring ventilatory limitations or metabolic disorders without the need for invasive testing.

Ultimately, this work reflects a broader aspiration: to shift the prevailing lens of exercise science away from isolated, one-dimensional studies and toward systems-level models that capture the interdependence of physiological processes. Rather than treating threshold values, lactate responses, or oxygen kinetics as discrete outcomes, the digital twin framework encourages us to view them as emergent properties of an integrated system. In doing so, it invites a more cohesive and mechanistic understanding of endurance performance—one that respects both the complexity of human physiology and the constraints of real-world data. The hope is that we don't just predict, but understand.

Acknowledgements

Thank you to Dave Uher for answering endless questions about physiological testing

Thank you to Professor Mucha for the mentorship, support, and enthusiasm

Thank you to Alice Boillet for helping me understand the 3-tank model

Special thanks to all my friends and family for encouraging me and putting up with my endless ponderings about thresholds and VO₂max testing.

References

Ahlquist, L. E., Jr, D. R. B., Sufit, R., Nagle, F. J., & Thomas, D. P. (n.d.). *The effect of pedaling frequency on glycogen depletion rates in type I and type II quadriceps muscle fibers during submaximal cycling exercise.*

Altenburg, T. M., Degens, H., Van Mechelen, W., Sargeant, A. J., & De Haan, A. (2007). Recruitment of single muscle fibers during submaximal cycling exercise. *Journal of Applied Physiology*, 103(5), 1752–1756. <https://doi.org/10.1152/jappphysiol.00496.2007>

Artiga Gonzalez, A., Bertschinger, R., Brosda, F., Dahmen, T., Thumm, P., & Saupe, D. (2019). Kinetic analysis of oxygen dynamics under a variable work rate. *Human Movement Science*, 66, 645–658. <https://doi.org/10.1016/j.humov.2017.08.020>

Baguet, A., Everaert, I., Hespel, P., Petrovic, M., Achten, E., & Derave, W. (2011). A New Method for Non-Invasive Estimation of Human Muscle Fiber Type Composition. *PLoS ONE*, 6(7), e21956. <https://doi.org/10.1371/journal.pone.0021956>

Bangsbo, J., Johansen, L., Graham, T., & Saltin, B. (1993). Lactate and H⁺ effluxes from human skeletal muscles during intense, dynamic exercise. *The Journal of Physiology*, 462(1), 115–133. <https://doi.org/10.1113/jphysiol.1993.sp019546>

Barstow, T. J., Jones, A. M., Nguyen, P. H., & Casaburi, R. (1996). Influence of muscle fiber type and pedal frequency on oxygen uptake kinetics of heavy exercise. *Journal of Applied Physiology*, 81(4), 1642–1650. <https://doi.org/10.1152/jappl.1996.81.4.1642>

Barstow, T. J., Jones, A. M., Nguyen, P. H., & Casaburi, R. (2000a). Influence of Muscle Fibre Type and Fitness on the Oxygen Uptake/Power Output Slope During Incremental Exercise in Humans. *Experimental Physiology*, 85(1), 109–116. <https://doi.org/10.1111/j.1469-445X.2000.01942.x>

Barstow, T. J., Jones, A. M., Nguyen, P. H., & Casaburi, R. (2000b). Influence of Muscle Fibre Type and Fitness on the Oxygen Uptake/Power Output Slope During Incremental Exercise in Humans. *Experimental Physiology*, 85(1), 109–116. <https://doi.org/10.1111/j.1469-445X.2000.01942.x>

Bartoloni, B., Mannelli, M., Gamberi, T., & Fiaschi, T. (2024). The Multiple Roles of Lactate in the Skeletal Muscle. *Cells*, 13(14), 1177. <https://doi.org/10.3390/cells13141177>

Behncke, H. (1993). A mathematical model for the force and energetics in competitive running. *Journal of Mathematical Biology*, 31(8), 853–878. <https://doi.org/10.1007/BF00168050>

Bellinger, P., Desbrow, B., Derave, W., Lievens, E., Irwin, C., Sabapathy, S., Kennedy, B., Craven, J., Pennell, E., Rice, H., & Minahan, C. (2020). Muscle fiber typology is associated with the incidence of overreaching in response to overload training. *Journal of Applied Physiology*, 129(4), 823–836. <https://doi.org/10.1152/jappphysiol.00314.2020>

Beltman, J. G. M., Sargeant, A. J., Van Mechelen, W., & De Haan, A. (2004). Voluntary activation level and muscle fiber recruitment of human quadriceps during lengthening contractions. *Journal of Applied Physiology*, 97(2), 619–626. <https://doi.org/10.1152/jappphysiol.01202.2003>

Bex, T. (n.d.). *NMR-BASED APPLICATIONS IN ELITE SPORTS PERFORMANCE*.

Bex, T., Baguet, A., Achten, E., Aerts, P., De Clercq, D., & Derave, W. (2017). Cyclic movement frequency is associated with muscle typology in athletes. *Scandinavian Journal of Medicine & Science in Sports*, 27(2), 223–229. <https://doi.org/10.1111/sms.12648>

Billat, V. L., Mille-Hamard, Petit, & Koralsztein. (1999). The Role of Cadence on the $\dot{V}O_2$ Slow Component in Cycling and Running in Triathletes. *International Journal of Sports Medicine*, 20(7), 429–437. <https://doi.org/10.1055/s-1999-8825>

Black, M. I., Fulford, J., Debenedictis, L., Wylie, L. J., Williams, E., Armstrong, N., & Jones, A. M. (2018). Is There an Optimal Speed for Economical Running? *International Journal of Sports Physiology and Performance*, 13(1), 75–81. <https://doi.org/10.1123/ijsp.2017-0015>

Boillet, A., Messonnier, L. A., & Cohen, C. (2024). Individualized physiology-based digital twin model for sports performance prediction: A reinterpretation of the Margaria–Morton model. *Scientific Reports*, 14(1), 5470. <https://doi.org/10.1038/s41598-024-56042-0>

Brooks, G. A., Arevalo, J. A., Osmond, A. D., Leija, R. G., Curl, C. C., & Tovar, A. P. (2022). Lactate in contemporary biology: A phoenix risen. *The Journal of Physiology*, 600(5), 1229–1251. <https://doi.org/10.1113/JP280955>

Caen, K., Bourgois, J. G., Stassijns, E., & Boone, J. (2022). A longitudinal study on the interchangeable use of whole-body and local exercise thresholds in cycling. *European Journal of Applied Physiology*, 122(7), 1657–1670. <https://doi.org/10.1007/s00421-022-04942-2>

Caen, K., Poole, D. C., Vanhatalo, A., & Jones, A. M. (2024). Critical Power and Maximal Lactate Steady State in Cycling: “Watts” the Difference? *Sports Medicine*, 54(10), 2497–2513. <https://doi.org/10.1007/s40279-024-02075-4>

Cerezuela-Espejo, V., Courel-Ibáñez, J., Morán-Navarro, R., Martínez-Cava, A., & Pallarés, J. G. (2018). The Relationship Between Lactate and Ventilatory Thresholds in Runners: Validity and Reliability of Exercise Test Performance Parameters. *Frontiers in Physiology*, 9, 1320. <https://doi.org/10.3389/fphys.2018.01320>

Chavarren, J. (n.d.-a). *Cycling ef®ciency and pedalling frequency in road cyclists*.

Chavarren, J. (n.d.-b). *Cycling ef®ciency and pedalling frequency in road cyclists*.

Chwalbinska-Moneta, J., Robergs, R. A., Costill, D. L., & Fink, W. J. (n.d.-a). *Threshold for muscle lactate accumulation during progressive exercise*.

Chwalbinska-Moneta, J., Robergs, R. A., Costill, D. L., & Fink, W. J. (n.d.-b). *Threshold for muscle lactate accumulation during progressive exercise*.

Conde Alonso, S., Gajanand, T., Ramos, J. S., Antonietti, J.-P., & Borrani, F. (2020). The metabolic profiles of different fiber type populations under the emergence of the slow component of oxygen uptake. *The Journal of Physiological Sciences*, 70(1), 27. <https://doi.org/10.1186/s12576-020-00754-1>

Cross, T. J., Winters, C., Sheel, A. W., & Sabapathy, S. (2014). Respiratory Muscle Power and the Slow Component of O₂ Uptake. *Medicine & Science in Sports & Exercise*, 46(9), 1797–1807. <https://doi.org/10.1249/MSS.0000000000000306>

Crouter, S. E., Antczak, A., Hudak, J. R., DellaValle, D. M., & Haas, J. D. (2006). Accuracy and reliability of the ParvoMedics TrueOne 2400 and MedGraphics VO2000 metabolic systems. *European Journal of Applied Physiology*, 98(2), 139–151. <https://doi.org/10.1007/s00421-006-0255-0>

Da Eira Silva, V., Painelli, V. D. S., Shinjo, S. K., Ribeiro Pereira, W., Cilli, E. M., Sale, C., Gualano, B., Otaduy, M. C., & Artioli, G. G. (2020). Magnetic Resonance Spectroscopy as a Non-invasive Method to Quantify Muscle Carnosine in Humans: A Comprehensive Validity Assessment. *Scientific Reports*, 10(1), 4908. <https://doi.org/10.1038/s41598-020-61587-x>

Da Silva, J. C. L., Tarassova, O., Ekblom, M. M., Andersson, E., Rönquist, G., & Arndt, A. (2016). Quadriceps and hamstring muscle activity during cycling as measured with intramuscular electromyography. *European Journal of Applied Physiology*, 116(9), 1807–1817. <https://doi.org/10.1007/s00421-016-3428-5>

Dotan, R., Mitchell, C., Cohen, R., Klentrou, P., Gabriel, D., & Falk, B. (2012). Child—Adult Differences in Muscle Activation—A Review. *Pediatric Exercise Science*, 24(1), 2–21. <https://doi.org/10.1123/pes.24.1.2>

[*Ergebnisse der Physiologie N°89*] Erich Heinz (auth.)—*Reviews of Physiology, Biochemistry and Pharmacology, Volume 89 (1981, Springer) [10.1007_BFb0035262]*—Libgen.li.pdf. (n.d.).

Fischer, J., Hävecker, F., Ji, S., Wahl, P., & Keller, S. (2025). Modeling lactate threshold in cycling—Influence of sex, maximal oxygen uptake, and cost of cycling in young athletes. *European Journal of Applied Physiology*. <https://doi.org/10.1007/s00421-025-05744-y>

Foss, Ølvind, & Hallén, J. (2004). The most economical cadence increases with increasing workload. *European Journal of Applied Physiology*, 92(4–5). <https://doi.org/10.1007/s00421-004-1175-5>

Foss, Ølvind, & Hallén, J. (2005). Cadence and performance in elite cyclists. *European Journal of Applied Physiology*, 93(4), 453–462. <https://doi.org/10.1007/s00421-004-1226-y>

Gaitanos, G. C., Williams, C., Boobis, L. H., & Brooks, S. (1993). Human muscle metabolism during intermittent maximal exercise. *Journal of Applied Physiology*, 75(2), 712–719. <https://doi.org/10.1152/jappl.1993.75.2.712>

Gouzi, F., Maury, J., Molinari, N., Pomiès, P., Mercier, J., Préfaut, C., & Hayot, M. (2013). Reference values for vastus lateralis fiber size and type in healthy subjects over 40 years old: A systematic review and metaanalysis. *Journal of Applied Physiology*, 115(3), 346–354. <https://doi.org/10.1152/japplphysiol.01352.2012>

Grassi, B., Quaresima, V., Marconi, C., Ferrari, M., & Cerretelli, P. (1999). Blood lactate accumulation and muscle deoxygenation during incremental exercise. *Journal of Applied Physiology*, 87(1), 348–355. <https://doi.org/10.1152/jappl.1999.87.1.348>

Green, H. J., Hughson, R. L., Orr, G. W., & Ranney, D. A. (1983). Anaerobic threshold, blood lactate, and muscle metabolites in progressive exercise. *Journal of Applied Physiology*, 54(4), 1032–1038. <https://doi.org/10.1152/jappl.1983.54.4.1032>

Gregory, C. M., & Bickel, C. S. (2005). Recruitment Patterns in Human Skeletal Muscle During Electrical Stimulation. *Physical Therapy*, 85(4), 358–364.

<https://doi.org/10.1093/ptj/85.4.358>

Hansen, E. A., Andersen, J. L., Nielsen, J. S., & Sjøgaard, G. (2002). Muscle fibre type, efficiency, and mechanical optima affect freely chosen pedal rate during cycling. *Acta Physiologica Scandinavica*, 176(3), 185–194. <https://doi.org/10.1046/j.1365-201X.2002.01032.x>

Hart, S., Drevets, K., Alford, M., Salacinski, A., & Hunt, B. E. (2013). A method-comparison study regarding the validity and reliability of the Lactate Plus analyzer. *BMJ Open*, 3(2), e001899. <https://doi.org/10.1136/bmjopen-2012-001899>

Hautier, C. A., Linossier, M. T., Belli, A., Lacour, J. R., & Arsac, L. M. (1996). Optimal velocity for maximal power production in non-isokinetic cycling is related to muscle fibre type composition. *European Journal of Applied Physiology and Occupational Physiology*, 74(1–2), 114–118. <https://doi.org/10.1007/BF00376503>

Henneman, E., Somjen, G., & Carpenter, D. O. (1965). FUNCTIONAL SIGNIFICANCE OF CELL SIZE IN SPINAL MOTONEURONS. *Journal of Neurophysiology*, 28(3), 560–580.

<https://doi.org/10.1152/jn.1965.28.3.560>

Hopker, J. G., Coleman, D. A., Gregson, H. C., Jobson, S. A., Von Der Haar, T., Wiles, J., & Passfield, L. (2013). The influence of training status, age, and muscle fiber type on cycling efficiency and endurance performance. *Journal of Applied Physiology*, 115(5), 723–729.

<https://doi.org/10.1152/jappphysiol.00361.2013>

Hopker, J., Passfield, L., Coleman, D., Jobson, S., Edwards, L., & Carter, H. (2009). The Effects of Training on Gross Efficiency in Cycling: A Review. *International Journal of Sports Medicine*, 30(12), 845–850. <https://doi.org/10.1055/s-0029-1237712>

Hughson, R. L., Sherrill, D. L., & Swanson, G. D. (1988). Kinetics of VO₂ with impulse and step exercise in humans. *Journal of Applied Physiology*, 64(1), 451–459.

<https://doi.org/10.1152/jappl.1988.64.1.451>

Iaia, F. M., Perez-Gomez, J., Thomassen, M., Nordsborg, N. B., Hellsten, Y., & Bangsbo, J. (2011). Relationship between performance at different exercise intensities and skeletal muscle characteristics. *Journal of Applied Physiology*, 110(6), 1555–1563.

<https://doi.org/10.1152/jappphysiol.00420.2010>

Iii, F. M., Hines, B., Martin, C., McCain, A., Tedder, E., & Calle, Y. L. (2024). *VO₂ Master Analyzer versus Parvo Medics TrueOne 2400 Canopy System for assessing Resting Metabolic Rate and Oxygen Consumption.*

Jenkins, N. D. M., Miramonti, A. A., Hill, E. C., Smith, C. M., Cochrane-Snyman, K. C., Housh, T. J., & Cramer, J. T. (2017). Greater Neural Adaptations following High- vs. Low-Load Resistance Training. *Frontiers in Physiology*, 8, 331.

<https://doi.org/10.3389/fphys.2017.00331>

Jenkins, N. D. M., Miramonti, A. A., Hill, E. C., Smith, C. M., Cochrane-Snyman, K. C., Housh, T. J., & Cramer, J. T. (2021). Mechanomyographic Amplitude Is Sensitive to Load-Dependent Neuromuscular Adaptations in Response to Resistance Training. *Journal of Strength and Conditioning Research*, 35(11), 3265–3269.

<https://doi.org/10.1519/JSC.0000000000003276>

Jenkins, N. D. M., Rogers, Emily. M., Banks, N. F., Muddle, T. W. D., & Colquhoun, R. J. (2021). Increases in motor unit action potential amplitudes are related to muscle hypertrophy following eight weeks of high-intensity exercise training in females. *European Journal of Sport Science*, 21(10), 1403–1413.

<https://doi.org/10.1080/17461391.2020.1836262>

Jones, A. M., Grassi, B., Christensen, P. M., Krstrup, P., Bangsbo, J., & Poole, D. C. (2011). Slow Component of $\dot{V}O_2$ Kinetics: Mechanistic Bases and Practical Applications. *Medicine & Science in Sports & Exercise*, 43(11), 2046–2062.

<https://doi.org/10.1249/MSS.0b013e31821fcfc1>

Jones, A. M., Wilkerson, D. P., DiMenna, F., Fulford, J., & Poole, D. C. (2008). Muscle metabolic responses to exercise above and below the “critical power” assessed using³¹P-MRS. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 294(2), R585–R593. <https://doi.org/10.1152/ajpregu.00731.2007>

Kim, T. H., Han, J. K., Lee, J. Y., & Choi, Y. C. (2021). The Effect of Polarized Training on the Athletic Performance of Male and Female Cross-Country Skiers during the General Preparation Period. *Healthcare*, 9(7), 851. <https://doi.org/10.3390/healthcare9070851>

Korkmaz Eryilmaz, S., & Polat, M. (2021). Correlation of maximal respiratory exchange ratio with anaerobic power and maximal oxygen uptake in anaerobic trained athletes. *Pedagogy of Physical Culture and Sports*, 25(4), 261–266.

<https://doi.org/10.15561/26649837.2021.0408>

Krstrup, P., Sanderlund, K., Mohr, M., Gonzalez-Alonso, J., & Bangsbo, J. (2004). Recruitment of fibre types and quadriceps muscle portions during repeated, intense knee-extensor exercise in humans. *Pflügers Archiv - European Journal of Physiology*, 449(1), 56–65. <https://doi.org/10.1007/s00424-004-1304-3>

- Lidar, J., Ainegren, M., & Sundström, D. (2023). Development and validation of dynamic bioenergetic model for intermittent ergometer cycling. *European Journal of Applied Physiology*, 123(12), 2755–2770. <https://doi.org/10.1007/s00421-023-05256-7>
- Lievens, E., Bellinger, P., Van Vossel, K., Vancompernelle, J., Bex, T., Minahan, C., & Derave, W. (2021). Muscle Typology of World-Class Cyclists across Various Disciplines and Events. *Medicine & Science in Sports & Exercise*, 53(4), 816–824. <https://doi.org/10.1249/MSS.0000000000002518>
- Lievens, E., Klass, M., Bex, T., & Derave, W. (2020). Muscle fiber typology substantially influences time to recover from high-intensity exercise. *Journal of Applied Physiology*, 128(3), 648–659. <https://doi.org/10.1152/jappphysiol.00636.2019>
- Lievens, E., Van Vossel, K., Van De Castele, F., Krššák, M., Murdoch, J. B., Befroy, D. E., & Derave, W. (2021). CORP: Quantification of human skeletal muscle carnosine concentration by proton magnetic resonance spectroscopy. *Journal of Applied Physiology*, 131(1), 250–264. <https://doi.org/10.1152/jappphysiol.00056.2021>
- Lopez, J. (2023). *Differences and possible determinants of cycling gross efficiency in male elite, amateur and female cyclists*. LITHUANIAN SPORTS UNIVERSITY.
- Medbo, J. I., Mohn, A. C., Tabata, I., Bahr, R., Vaage, O., & Sejersted, O. M. (1988). Anaerobic capacity determined by maximal accumulated O₂ deficit. *Journal of Applied Physiology*, 64(1), 50–60. <https://doi.org/10.1152/jappl.1988.64.1.50>
- Miller, P., Perez, N., & Farrell, J. W. (2023). Acute Oxygen Consumption Response to Fast Start High-Intensity Intermittent Exercise. *Sports*, 11(12), 238. <https://doi.org/10.3390/sports11120238>
- Minetti, A. E., Moia, C., Roi, G. S., Susta, D., & Ferretti, G. (2002). Energy cost of walking and running at extreme uphill and downhill slopes. *Journal of Applied Physiology*, 93(3), 1039–1046. <https://doi.org/10.1152/jappphysiol.01177.2001>
- Mogensen, M., Bagger, M., Pedersen, P. K., Fernström, M., & Sahlin, K. (2006). Cycling efficiency in humans is related to low UCP3 content and to type I fibres but not to mitochondrial efficiency. *The Journal of Physiology*, 571(3), 669–681. <https://doi.org/10.1113/jphysiol.2005.101691>
- Morton, R. H. (1986). A three component model of human bioenergetics. *Journal of Mathematical Biology*, 24(4), 451–466. <https://doi.org/10.1007/BF01236892>

Morton, R. H., Fitz-Clarke, J. R., & Banister, E. W. (1990). Modeling human performance in running. *Journal of Applied Physiology*, 69(3), 1171–1177.

<https://doi.org/10.1152/jappl.1990.69.3.1171>

Nolte, S., Quittmann, O. J., & Meden, V. (2022). *Simulation of Steady-State Energy Metabolism in Cycling and Running*. <https://doi.org/10.51224/SRXIV.110>

Nuzzo, J. L. (2023a). Narrative Review of Sex Differences in Muscle Strength, Endurance, Activation, Size, Fiber Type, and Strength Training Participation Rates, Preferences, Motivations, Injuries, and Neuromuscular Adaptations. *Journal of Strength and Conditioning Research*, 37(2), 494–536. <https://doi.org/10.1519/JSC.0000000000004329>

Nuzzo, J. L. (2023b). Supplement to: Narrative Review of Sex Differences in Muscle Strength, Endurance, Activation, Size, Fiber Type, and Strength Training Participation Rates, Preferences, Motivations, Injuries, and Neuromuscular Adaptations. *Journal of Strength and Conditioning Research*, 37(2), 494–536.

<https://doi.org/10.1519/JSC.0000000000004329>

Nuzzo, J. L. (2024). Sex differences in skeletal muscle fiber types: A meta-analysis. *Clinical Anatomy*, 37(1), 81–91. <https://doi.org/10.1002/ca.24091>

Oskolkov, N., Santel, M., Parikh, H. M., Ekström, O., Camp, G. J., Miyamoto-Mikami, E., Ström, K., Mir, B. A., Kryvokhyzha, D., Lehtovirta, M., Kobayashi, H., Kakigi, R., Naito, H., Eriksson, K.-F., Nystedt, B., Fuku, N., Treutlein, B., Pääbo, S., & Hansson, O. (2022). High-throughput muscle fiber typing from RNA sequencing data. *Skeletal Muscle*, 12(1), 16.

<https://doi.org/10.1186/s13395-022-00299-4>

Poole, D. C., Rossiter, H. B., Brooks, G. A., & Gladden, L. B. (2021). The anaerobic threshold: 50+ years of controversy. *The Journal of Physiology*, 599(3), 737–767.

<https://doi.org/10.1113/JP279963>

Rohatgi, A. (2021). *WebPlotDigitizer* (Version 4.5) [Computer software].

<https://automeris.io/>

Running Writings. (2023). *Grade-Adjusted Pace Calculator*.

<https://apps.runningwritings.com/gap-calculator/>

San-Millán, I. (2023). The Key Role of Mitochondrial Function in Health and Disease. *Antioxidants*, 12(4), 782. <https://doi.org/10.3390/antiox12040782>

Saunders, M. J., Evans, E. M., Arngrimsson, S. A., Allison, J. D., Warren, G. L., & Cureton, A. K. J. (2000). Muscle activation and the slow component rise in oxygen uptake during

cycling: *Medicine and Science in Sports and Exercise*, 32(12), 2040–2045.

<https://doi.org/10.1097/00005768-200012000-00012>

Schiaffino, S., & Reggiani, C. (2011). Fiber Types in Mammalian Skeletal Muscles.

Physiological Reviews, 91(4), 1447–1531. <https://doi.org/10.1152/physrev.00031.2010>

Seiler, S. (2010). What is Best Practice for Training Intensity and Duration Distribution in

Endurance Athletes? *International Journal of Sports Physiology and Performance*, 5(3),

276–291. <https://doi.org/10.1123/ijsp.5.3.276>

Semenova, E. A., Khabibova, S. A., Borisov, O. V., Generozov, E. V., & Ahmetov, I. I. (2019).

The Variability of DNA Structure and Muscle-Fiber Composition. *Human Physiology*, 45(2),

225–232. <https://doi.org/10.1134/S0362119719010122>

Sreedhara, V. S. M., Mocko, G. M., & Hutchison, R. E. (2019). A survey of mathematical

models of human performance using power and energy. *Sports Medicine - Open*, 5(1), 54.

<https://doi.org/10.1186/s40798-019-0230-z>

Stegmann, H., Kindermann, W., & Schnabel, A. (1981). Lactate Kinetics and Individual

Anaerobic Threshold*. *International Journal of Sports Medicine*, 02(03), 160–165.

<https://doi.org/10.1055/s-2008-1034604>

Stöggl, T., & Sperlich, B. (2014). Polarized training has greater impact on key endurance

variables than threshold, high intensity, or high volume training. *Frontiers in Physiology*, 5.

<https://doi.org/10.3389/fphys.2014.00033>

Swinnen, W., Lievens, E., Hoogkamer, W., De Groot, F., Derave, W., & Vanwanseele, B.

(2024). Inter-Individual Variability in Muscle Fiber-Type Distribution Affects Running

Economy but Not Running Gait at Submaximal Running Speeds. *Scandinavian Journal of*

Medicine & Science in Sports, 34(11), e14748. <https://doi.org/10.1111/sms.14748>

Tehrani, F., Teymourian, H., Wuerstle, B., Kavner, J., Patel, R., Furnidge, A., Aghavali, R.,

Hosseini-Toudeshki, H., Brown, C., Zhang, F., Mahato, K., Li, Z., Barfidokht, A., Yin, L.,

Warren, P., Huang, N., Patel, Z., Mercier, P. P., & Wang, J. (2022). An integrated wearable

microneedle array for the continuous monitoring of multiple biomarkers in interstitial fluid.

Nature Biomedical Engineering, 6(11), 1214–1224. [https://doi.org/10.1038/s41551-022-](https://doi.org/10.1038/s41551-022-00887-1)

[00887-1](https://doi.org/10.1038/s41551-022-00887-1)

Tesch*, P., Sharp, D., & Daniels, W. (1981). Influence of Fiber Type Composition and

Capillary Density on Onset of Blood Lactate Accumulation. *International Journal of Sports*

Medicine, 02(04), 252–255. <https://doi.org/10.1055/s-2008-1034619>

Van De Castele, F., Van Thienen, R., Horwath, O., Apró, W., Van Der Stede, T., Moberg, M., Lievens, E., & Derave, W. (2024). Does one biopsy cut it? Revisiting human muscle fiber type composition variability using repeated biopsies in the vastus lateralis and gastrocnemius medialis. *Journal of Applied Physiology*, *137*(5), 1341–1353.

<https://doi.org/10.1152/jappphysiol.00394.2024>

Van Der Zwaard, S., Brocherie, F., & Jaspers, R. T. (2021). Under the Hood: Skeletal Muscle Determinants of Endurance Performance. *Frontiers in Sports and Active Living*, *3*, 719434.

<https://doi.org/10.3389/fspor.2021.719434>

Van Vossel, K., Hardeel, J., Van De Castele, F., De Jager, S., Lievens, E., Boone, J., & Derave, W. (2023). Muscle typology influences the number of repetitions to failure during resistance training. *European Journal of Sport Science*, *23*(10), 2021–2030.

<https://doi.org/10.1080/17461391.2023.2207077>

Vanhatalo, A., Black, M. I., DiMenna, F. J., Blackwell, J. R., Schmidt, J. F., Thompson, C., Wylie, L. J., Mohr, M., Bangsbo, J., Krstrup, P., & Jones, A. M. (2016). The mechanistic bases of the power–time relationship: Muscle metabolic responses and relationships to muscle fibre type. *The Journal of Physiology*, *594*(15), 4407–4423. <https://doi.org/10.1113/JP271879>

Venturini, E., & Giallauria, F. (2022). Factors Influencing Running Performance During a Marathon: Breaking the 2-h Barrier. *Frontiers in Cardiovascular Medicine*, *9*, 856875.

<https://doi.org/10.3389/fcvm.2022.856875>

Vikne, H., Gundersen, K., Liestøl, K., Mælen, J., & Vøllestad, N. (2012). Intermuscular relationship of human muscle fiber type proportions: Slow leg muscles predict slow neck muscles. *Muscle & Nerve*, *45*(4), 527–535. <https://doi.org/10.1002/mus.22315>

Wackwitz, T., Minahan, C., Lievens, E., Kennedy, B., Derave, W., & Bellinger, P. (2025). Muscle-Fiber Typology Is Associated With Sprint-Cycling Characteristics in World-Class and Elite Track Cyclists. *International Journal of Sports Physiology and Performance*, *20*(1), 142–148.

<https://doi.org/10.1123/ijsp.2024-0089>

Widrick, J. J., Trappe, S. W., Costill, D. L., & Fitts, R. H. (1996). Force-velocity and force-power properties of single muscle fibers from elite master runners and sedentary men. *American Journal of Physiology-Cell Physiology*, *271*(2), C676–C683.

<https://doi.org/10.1152/ajpcell.1996.271.2.C676>

Zhang, J., Iannetta, D., Alzeeby, M., MacInnis, M. J., & Aboodarda, S. J. (2021). Exercising muscle mass influences neuromuscular, cardiorespiratory, and perceptual responses during and following ramp-incremental cycling to task failure. *American Journal of*

Physiology-Regulatory, Integrative and Comparative Physiology, 321(2), R238–R249.

<https://doi.org/10.1152/ajpregu.00286.2020>

Zuccarelli, L., Porcelli, S., Rasica, L., Marzorati, M., & Grassi, B. (2018). Comparison between Slow Components of HR and $\dot{V}O_2$ Kinetics: Functional Significance. *Medicine & Science in Sports & Exercise*, 50(8), 1649–1657.

<https://doi.org/10.1249/MSS.0000000000001612>

Appendix

Overall VO₂ Model Error Table:

Metric	Comparison	p-value	Significance
MAE	Overall vs Max	0.0000	***
MAE	A vs Max	0.0000	***
MAE	B vs Max	0.0000	***
MSE	Overall vs Max	0.0000	***
RMSE	Overall vs Max	0.0000	***
MSE	B vs Max	0.0000	***
MSE	A vs Max	0.0000	***
RMSE	A vs Max	0.0000	***
RMSE	B vs Max	0.0000	***
R ²	Overall vs Max	0.0000	***
MAE	Overall vs A	0.0001	***
RMSE	Overall vs A	0.0002	***
Bias	Overall vs Max	0.0006	***
Bias	A vs Max	0.0008	***
Bias	B vs Max	0.0011	**
R ²	Overall vs B	0.0016	**
MSE	Overall vs A	0.0052	**
RMSE	Overall vs B	0.0088	**
Bias	Overall vs B	0.0109	*
R ²	A vs Max	0.0324	*
MAE	Overall vs B	0.0727	*
R ²	Overall vs A	0.0928	*
R ²	A vs B	0.1413	
MAE	A vs B	0.1622	
Bias	A vs B	0.1952	
MSE	A vs B	0.2007	
RMSE	A vs B	0.3344	
R ²	B vs Max	0.3494	
Bias	Overall vs A	0.4141	
MSE	Overall vs B	0.4736	
MAPE	Overall vs A	0.6827	
MAPE	A vs B	0.7366	
MAPE	A vs Max	0.7723	
MAPE	Overall vs B	0.8104	
MAPE	B vs Max	0.8296	
MAPE	Overall vs Max	0.8983	

RER=VCO2/VO2 Correction Algorithm Code

```
def apply_vo2_vco2_correction(vo2_eco=None, vco2_eco=None, vo2_max=None,
                             vco2_max=None,

                             sd_vo2=0.118, sd_vco2=0.143,

                             rer_floor=0.7, rer_ceiling=1.4):

    lambda1 = 1 / sd_vo2**2

    lambda2 = 1 / sd_vco2**2

    hard_penalty = 1e6 # Large penalty if min(RER) < 0.7

def correction_cost(params, vo2, vco2):

    alpha, beta = params

    vo2_corr = alpha * vo2

    vco2_corr = beta * vco2

    rer_corr = vco2_corr / vo2_corr

    penalty_low = np.sum((np.maximum(0, rer_floor - rer_corr))**2)

    penalty_high = np.sum((np.maximum(0, rer_corr - rer_ceiling))**2)

    reg = lambda1 * (alpha - 1)**2 + lambda2 * (beta - 1)**2

    hard_constraint_penalty = 0

    if np.min(rer_corr) < rer_floor:

        hard_constraint_penalty = hard_penalty * (rer_floor - np.min(rer_corr))**2

    return penalty_low + penalty_high + reg + hard_constraint_penalty
```

```

# Optimization
if vo2_eco is not None and vco2_eco is not None:
    result = minimize(
        correction_cost, x0=[1.0, 1.0],
        args=(vo2_eco, vco2_eco),
        bounds=[(0.85, 1.0), (1.0, 1.2)]
    )
elif vo2_max is not None and vco2_max is not None:
    result = minimize(
        correction_cost, x0=[1.0, 1.0],
        args=(vo2_max, vco2_max),
        bounds=[(0.85, 1.0), (1.0, 1.2)]
    )
else:
    return None, None, None, None

alpha, beta = result.x

def correct(vo2, vco2):
    vo2_corr = alpha * vo2
    vco2_corr = beta * vco2
    rer_corr = vco2_corr / vo2_corr
    rer_raw = vco2 / vo2
    return pd.DataFrame({
        "VO2_raw": vo2,
        "VCO2_raw": vco2,

```

```
"RER_raw": rer_raw,  
"VO2_corr": vo2_corr,  
"VCO2_corr": vco2_corr,  
"RER_corr": rer_corr  
})
```

```
df_eco = correct(vo2_eco, vco2_eco) if vo2_eco is not None else None
```

```
if df_eco is not None:
```

```
    df_eco["Test"] = "eco"
```

```
df_max = correct(vo2_max, vco2_max) if vo2_max is not None else None
```

```
if df_max is not None:
```

```
    df_max["Test"] = "max"
```

```
return df_eco, df_max, alpha, beta
```

```
# === File grouping and processing ===
```

```
no_rer_adj = []
```

```
rer_adj_og = []
```

```
adjusted_data = {}
```

```
raw_data = {} # store unadjusted (no correction needed) test data
```

```
all_files = [f for f in os.listdir(folder) if f.endswith(".xlsx")]
```

```
grouped = {}
```

```
# Group by subject prefix
```

```
for fname in all_files:
```

```
    parts = fname.lower().split("_")
```

```
    base = "_".join(parts[:2]) # Keep first and last name
```

```
    grouped.setdefault(base, []).append(fname)
```

```
for subj, files in grouped.items():
```

```
    eco_file = next((f for f in files if "eco" in f.lower()), None)
```

```
    max_file = next((f for f in files if "max" in f.lower()), None)
```

```
    vo2_eco = vco2_eco = time_eco = rer_eco = None
```

```
    vo2_max = vco2_max = time_max = rer_max = None
```

```
try:
```

```
    if eco_file:
```

```
        eco_df = pd.read_excel(os.path.join(folder, eco_file))
```

```
        eco_df.columns = column_names[:len(eco_df.columns)]
```

```
        eco_df = eco_df[["Time_min", "VO2_Lmin", "VCO2_Lmin", "RER"]]
```

```
        eco_df["Time_min"] = pd.to_numeric(eco_df["Time_min"], errors="coerce")
```

```
        eco_df["RER"] = pd.to_numeric(eco_df["RER"], errors="coerce")
```

```
        eco_df = eco_df.dropna()
```

```
        vo2_eco = eco_df["VO2_Lmin"].values
```

```
        vco2_eco = eco_df["VCO2_Lmin"].values
```

```
        time_eco = pd.to_numeric(eco_df["Time_min"], errors="coerce").values
```

```
        rer_eco = eco_df["RER"].values
```

```
    if max_file:
```

```

max_df = pd.read_excel(os.path.join(folder, max_file))
max_df.columns = column_names[:len(max_df.columns)]
max_df = max_df[["Time_min", "VO2_Lmin", "VCO2_Lmin", "RER"]]
max_df["Time_min"] = pd.to_numeric(max_df["Time_min"], errors="coerce")
max_df["RER"] = pd.to_numeric(max_df["RER"], errors="coerce")
max_df = max_df.dropna()
vo2_max = max_df["VO2_Lmin"].values
vco2_max = max_df["VCO2_Lmin"].values
time_max = pd.to_numeric(max_df["Time_min"], errors="coerce").values
rer_max = max_df["RER"].values

```

except Exception as e:

```

    continue

```

try:

```

# Determine which dataset has the lower RER

```

```

min_rer_eco = np.min(rer_eco) if rer_eco is not None else np.inf

```

```

min_rer_max = np.min(rer_max) if rer_max is not None else np.inf

```

```

min_rer_total = min(min_rer_eco, min_rer_max)

```

```

if min_rer_total >= 0.7:

```

```

    if eco_file: no_rer_adj.append(eco_file)

```

```

    if max_file: no_rer_adj.append(max_file)

```

```

# Store the raw data

```

```

def format_unadjusted(df):

```

```

    return pd.DataFrame({

```

```

"VO2_raw": df["VO2_Lmin"],
"VCO2_raw": df["VCO2_Lmin"],
"RER_raw": df["RER"],
})

raw_data[subj] = {
    "eco": format_unadjusted(eco_df) if eco_file else None,
    "max": format_unadjusted(max_df) if max_file else None
}
continue

# Choose which to fit based on lowest RER
if min_rer_eco <= min_rer_max and vo2_eco is not None:
    fit_vo2, fit_vco2 = vo2_eco, vco2_eco
elif vo2_max is not None:
    fit_vo2, fit_vco2 = vo2_max, vco2_max
else:
    fit_vo2 = fit_vco2 = None

if fit_vo2 is None or fit_vco2 is None:
    continue # skip this subject if both are missing or invalid

# Redefine objective with fit_vo2/vco2
def correction_cost(params):
    alpha, beta = params
    vo2_corr = alpha * fit_vo2

```

```
vco2_corr = beta * fit_vco2
rer_corr = vco2_corr / vo2_corr

penalty_low = np.sum((np.maximum(0, 0.7 - rer_corr))**2)
penalty_high = np.sum((np.maximum(0, rer_corr - 1.4))**2)
reg = (1 / 0.118**2) * (alpha - 1)**2 + (1 / 0.143**2) * (beta - 1)**2
hard_penalty = 1e6 * max(0, 0.7 - np.min(rer_corr))**2
return penalty_low + penalty_high + reg + hard_penalty
```

```
# Run optimization
```

```
result = minimize(correction_cost, x0=[1.0, 1.0], bounds=[(0.85, 1.0), (1.0, 1.2)])
```

```
alpha, beta = result.x
```

```
# Apply  $\alpha$ ,  $\beta$  correction to both datasets
```

```
def correct(vo2, vco2):
```

```
    vo2_corr = alpha * vo2
```

```
    vco2_corr = beta * vco2
```

```
    rer_corr = vco2_corr / vo2_corr
```

```
    rer_raw = vco2 / vo2
```

```
    return pd.DataFrame({
```

```
        "VO2_raw": vo2,
```

```
        "VCO2_raw": vco2,
```

```
        "RER_raw": rer_raw,
```

```
        "VO2_corr": vo2_corr,
```

```
        "VCO2_corr": vco2_corr,
```

```
        "RER_corr": rer_corr
```

```
)
```

```
df_eco_corr = correct(vo2_eco, vco2_eco) if vo2_eco is not None else None
```

```
if df_eco_corr is not None:
```

```
    df_eco_corr["Test"] = "eco"
```

```
df_max_corr = correct(vo2_max, vco2_max) if vo2_max is not None else None
```

```
if df_max_corr is not None:
```

```
    df_max_corr["Test"] = "max"
```

```
if eco_file: rer_adj_og.append(eco_file)
```

```
if max_file: rer_adj_og.append(max_file)
```

```
adjusted_data[subj] = {
```

```
    "eco": df_eco_corr, "max": df_max_corr,
```

```
    "alpha": alpha, "beta": beta,
```

```
    "files": {"eco": eco_file, "max": max_file}
```

```
}
```

```
print(f"Adjusted {subj}:  $\alpha = \{alpha:.4f\}$ ,  $\beta = \{beta:.4f\}$ ")
```

```
except Exception as e:
```

```
    continue
```

```
# === Display result summaries ===
```

```
print("=== No RER Adjustment Needed ===")
```

```
print(pd.DataFrame(no_rer_adj, columns=["Filename"]))
```

```
print("\n=== Files Adjusted for RER ===")
```

```
print(pd.DataFrame(rer_adj_og, columns=["Filename"]))
```

Derived Substrate Oxidation Recalculation Code

After VCO2 and/or VO2 adjustments, these subject data columns require recomputation

```
column_names = [  
    "Time_min", "VO2_Lmin", "VO2/kg_Lmin", "METS", "VCO2_Lmin", "VE_Lmin", "RER",  
    "VE/VCO2", "Vt_L", "pCHO", "VE/VO2", "HR_bpm", "Speed_mph", "Gradient",  
    "AcKcal", "CHOfin", "FATmin", "KCHO", "KFAT", "KPRO"  
]
```

```
def safe_get(df, row, col):  
    try:  
        return pd.to_numeric(df.iloc[row, col], errors="coerce")  
    except:  
        return np.nan
```

```
def find_matching_file(subj, test_type, all_files):  
    subj_lower = subj.lower()  
    test_lower = test_type.lower()  
    for f in all_files:  
        if subj_lower in f.lower() and test_lower in f.lower():  
            return f  
    return None
```

```
def recalculate_derived_columns(df, weight_kg, vo2_col="VO2_corr",  
vco2_col="VCO2_corr", rer_col="RER_corr"):
```

```

df = df.copy()

df["VE_VCO2"] = df["VE_Lmin"] / df[vco2_col]
df["VO2/kg"] = df[vo2_col]*1000 / weight_kg

df["CHOmin"] = (4.55 * df[vco2_col] - 3.21 * df[vo2_col]).clip(lower=0)
df['FATmin'] = (1.695*df[vo2_col]-1.701*df[vco2_col]).clip(lower=0)

df["pCHO"] = (100 * (df[rer_col] - 0.7) / 0.3).clip(lower=0, upper=100)

time = df["Time_min"].values
time_deltas = np.zeros_like(time)
time_deltas[0] = time[0]
time_deltas[1:] = time[1:] - time[:-1]
time_deltas = np.where(time_deltas == 0, 1e-6, time_deltas)
df["Ackcal_inst"] = (3.9 * df[vo2_col] + 1.1 * df[vco2_col]) / (1 / time_deltas)
df["AcKcal"] = np.cumsum(df["Ackcal_inst"])
#df['Ktot_rollsum'] = 1.00413 * df["AcKcal"] -.09
df['KCHO'] = df["Ackcal_inst"] * df["pCHO"]/100
df['KFAT'] = df["Ackcal_inst"] * (1 - df["pCHO"]/100)

#df["KCHO"] = ((5383 * df["CHOmin"] ).clip(lower=0))/24/60/2

#df["pCHO"] = df["pCHO"].clip(lower=0.01, upper=99.99)
#df["KcalInst"] = (df["KCHO"] / df["pCHO"])/24/60/2
#df["KFAT"] = (df["KCHO"] * (1 - df["pCHO"] / 100) / (df["pCHO"] / 100))/24/60/2
#df["FATmin"] = 0.000073409 * df["KFAT"]*24*60*2 - 0.000000022

```

```

#df["Ktot"] = df["KCHO"] + df["KFAT"]

return df

def load_clean_test_data(file_path):
    """Load and return cleaned test data and metadata from a given Excel file path."""
    df = pd.read_excel(file_path, sheet_name=0)

    # Identify start of test data using dashed separator
    start_candidates = df.apply(lambda row: row.astype(str).str.contains("-{5,}").any(),
axis=1)
    start_row_idx = start_candidates[start_candidates].index[0] + 1

    # Identify end of data (first fully blank row after start)
    data_rows = df.iloc[start_row_idx:]
    empty_row_indices = data_rows.index[data_rows.isna().all(axis=1)]
    end_row_idx = empty_row_indices[0] if not empty_row_indices.empty else
data_rows.index[-1] + 1

    test_data_df = df.iloc[start_row_idx:end_row_idx].reset_index(drop=True)

    # Assign standard column names
    num_cols = test_data_df.shape[1]
    test_data_df.columns = column_names[:num_cols]

    # Extract metadata
    def safe_get(row, col):

```

```

try:
    return pd.to_numeric(df.iloc[row, col], errors="coerce")
except:
    return np.nan

if 'eco' in file_path.lower():
    test_type = 'threshold'
elif 'max' in file_path.lower():
    test_type = 'max'
else:
    test_type = 'unknown'

metadata = {
    "Name": df.iloc[4, 1] if len(df.columns) > 1 else None,
    "Weight_kg": safe_get(6, 8),
    "Test Type": test_type
}

return test_data_df, metadata

```

Threshold Identification Algorithm Code

```
#MAIN
```

```
import pandas as pd
```

```
import numpy as np
```

```
from scipy.stats import linregress
```

```

def identify_lt1_lt2(last_first, adjusted_data, raw_data):
    subj_key = last_first.lower()

    # Load ECO test data
    if subj_key in adjusted_data and adjusted_data[subj_key]['eco'] is not None:
        df = adjusted_data[subj_key]['eco'].copy()
        filetype = 'adjusted'
    elif subj_key in raw_data and raw_data[subj_key]['eco'] is not None:
        df = raw_data[subj_key]['eco'].copy()
        filetype = 'raw'
    else:
        print(f"No ECO test data found for subject '{last_first}'.")
        return None

    # Identify stages based on rounded speed
    df["Stage"] = df["Speed_mph"].round(2)
    stages = df.groupby("Stage").filter(lambda x: len(x) >= 4).groupby("Stage")
    sorted_stages = sorted(stages.groups.keys())
    stage_indices = {s: i for i, s in enumerate(sorted_stages)}
    stage_list = list(stages)
    print(stage_indices)

    def last_2min(df_stage):
        return df_stage[df_stage["Time_min"] >= df_stage["Time_min"].max() - 2]

    # Extract last 2 min averages for each stage

```

```

stage_metrics = {}

idx=1

for s, grp in stage_list:

    last2 = last_2min(grp)

    stage_metrics[s] = {

        "VE_VO2": last2["VE/VO2"].mean(),

        "VE_VCO2": last2["VE/VCO2"].mean(),

        "VE_Lmin": last2["VE_Lmin"].mean(),

        "RER": last2["RER_corr"].mean() if "RER_corr" in last2.columns else
last2["RER_raw"].mean(),

        "VO2": last2["VO2_corr"].mean() if "VO2_corr" in last2.columns else
last2["VO2_raw"].mean(),

        "VO2/kg": last2["VO2_kg_corr"].mean() if "VO2_kg_corr" in last2.columns else
last2["VO2_kg_raw"].mean(),

        "GAPspdperkW": last2["GAPspdperkW"].mean() if "GAPspdperkW" in last2.columns
else np.nan

    }#

    if idx ==1:

        if filetype == 'adjusted':

            adjusted_data[subj]['eco'].attrs['GAPspdperkws1l2minavg'] =
last2["GAPspdperkW"].mean() if "GAPspdperkW" in last2.columns else np.nan

            elif filetype == 'raw':

                raw_data[subj]['eco'].attrs['GAPspdperkws1l2minavg'] =
last2["GAPspdperkW"].mean() if "GAPspdperkW" in last2.columns else np.nan

        idx+=1

# ----- LT2 -----

candidate_lt2 = sorted_stages[-3:] if len(sorted_stages) >= 6 else sorted_stages[-2:]

```

```

lt2 = None
lt2_type = None
for s in candidate_lt2:
    if stage_indices[s] <=2: #NOT 1st or 2nd or 3rd stage, likely.
        continue
    prev_s = sorted_stages[stage_indices[s] - 1]
    if stage_metrics[s]["VE_VO2"] > stage_metrics[prev_s]["VE_VO2"] and \
        stage_metrics[s]["VE_VCO2"] > stage_metrics[prev_s]["VE_VCO2"]:
        x = [k for k in sorted_stages[:stage_indices[s]]]
        y = [stage_metrics[k]["VE_Lmin"] for k in x]
        slope, intercept, _, _, stderr = linregress(x, y)
        pred_ve = slope * s + intercept
        print('predve:',pred_ve,'se',stderr)
        print('active:',stage_metrics[s]["VE_Lmin"])
        if stage_metrics[s]["VE_Lmin"] > pred_ve + stderr:
            lt2 = {"Speed": s, "VO2":
stage_metrics[s]["VO2'],'VO2/kg':stage_metrics[s]['VO2/kg']}
            lt2_type = "pred"

            break

if not lt2:
    fallback_s = candidate_lt2[-2] if len(candidate_lt2) == 3 else candidate_lt2[-1]
    lt2 = {"Speed": fallback_s, "VO2":
stage_metrics[fallback_s]["VO2'],'VO2/kg':stage_metrics[fallback_s]['VO2/kg']}
    lt2_type = "fallback"

```

```

print(f" ⚠️ LT2 fallback used at stage {fallback_s}")

# ----- LT1 -----
lt2_index = stage_indices[lt2["Speed"]]
lt1 = None
lt1_type = None
for i in range(1, min(4, lt2_index - 1)):
    s = sorted_stages[i]
    prev_s = sorted_stages[i - 1]
    if s >= lt2["Speed"]:
        continue
    if stage_metrics[s]["VE_VO2"] > stage_metrics[prev_s]["VE_VO2"] and \
        stage_metrics[s]["VE_VCO2"] <= stage_metrics[prev_s]["VE_VCO2"] and \
        stage_metrics[s]["RER"] >= 0.825:
        lt1 = {"Speed": s, "VO2": stage_metrics[s]["VO2"], 'VO2/kg': stage_metrics[s]['VO2/kg']}
        lt1_type = "pred"
        break

if not lt1:
    if lt2_index == 3:
        s = sorted_stages[1]
        lt1 = {"Speed": s, "VO2": stage_metrics[s]["VO2"], 'VO2/kg': stage_metrics[s]['VO2/kg']}
        lt1_type = "fallback2"
        print(f" ⚠️ LT1 fallback: using stage 2 due to LT2 proximity at stage3")
    else:
        s1 = sorted_stages[1]

```

```

s2 = sorted_stages[2]
avg_speed = (s1 + s2) / 2
avg_vo2 = (stage_metrics[s1]["VO2"] + stage_metrics[s2]["VO2"]) / 2
avg_vo2_kg = (stage_metrics[s1]["VO2/kg"] + stage_metrics[s2]["VO2/kg"]) / 2
lt1 = {"Speed": avg_speed, "VO2": avg_vo2, 'VO2/kg': avg_vo2_kg} #FIX1
lt1_type = "fallback2.5"

print(" ⚠️ LT1 fallback: average of stages 2 and 3 used")

return lt1, lt2, lt1_type, lt2_type, stage_indices

```

W' Estimation Code

```
def estimate_W_nonox(subj, merged_data, gap_col="GAPspd"):
```

```
    """
```

```
    Estimate W'_non-ox using GAP speed and time-varying C1est from the max test,
    with CP derived from the LT2 speed stored in eco.attrs["thresholds"].
```

```
Parameters:
```

```
    subj (str): Subject ID
```

```
    merged_data (dict): Dictionary with subject data including 'max' and 'eco'
```

```
    gap_col (str): Column in max test containing GAP speed in mph
```

```
Returns:
```

```
    float: W'_non-ox in Joules
```

```
    """
```

```
if subj not in merged_data:
```

```

raise ValueError(f"{subj} not found in merged_data.")

# Get CP (LT2 speed) from eco test metadata
eco_df = merged_data[subj].get("eco")
if not isinstance(eco_df, pd.DataFrame):
    raise ValueError(f"'eco' test is not a DataFrame for {subj}.")

lt2 = eco_df.attrs.get("thresholds", {}).get("LT2", None)
if lt2 is None or "LT_Speed" not in lt2:
    raise ValueError(f"LT2 threshold missing for {subj} in eco.attrs.")

CP_speed = lt2["LT_Speed"] #in mph
CP_mps = CP_speed * 0.44704 # convert to m/s
CP_vo2 = lt2["LT_VO2"] # in L/min
# Pull GAP speed and C1est from max test
max_df = merged_data[subj].get("max")
if not isinstance(max_df, pd.DataFrame):
    raise ValueError(f"'max' test is not a DataFrame for {subj}.")

required_cols = [gap_col, "GAPspdperkW", "Time_min", "C1est"]
if not all(col in max_df.columns for col in required_cols):
    raise ValueError(f"Missing required columns in max test for {subj}.")

gap = max_df[gap_col].values * 0.44704 # in m/s
n1 = max_df["GAPspdperkW"].values * .44704 * (1/1000) #Eta in mph/kW: n = mph/ (J/s) -
> m/s / J/s = m/J      # in J/(m/s)

```

```

time_min = max_df["Time_min"].values
vo2 = max_df["VO2_corr"].values if "VO2_corr" in max_df.columns else
max_df["VO2_raw"].values
C1est = max_df["C1est"].values
dt = np.diff(time_min, prepend=time_min[1]) * 60 # in seconds

#ventilation and accum metabol correction, est from Lidar:
vo2pct = vo2/max(vo2)
vepct = 1.2499*(vo2pct**3) - 1.5287*(vo2pct**2) + 1.2687*vo2pct #cubic reg.
Bve = .93
vepct = 0.088*(Bve*(vepct) + (1-Bve)*((vepct)**2)) # MRve estimate (percent)
accpct = .36 * vepct #MRacc est. rough since dont have x4

vo2_adj = vo2 * (1-accpct-vepct)
vo2_excess = vo2_adj - CP_vo2
vo2_excess[vo2_excess < 0] = 0
vo2_excess_conv = vo2_excess*1000 * C1est * (1/60) * n1
above_cp = gap - CP_mps
above_cp[above_cp < 0] = 0 # keep only values above CP
#print(above_cp,vo2_excess_conv)

#W_nonox = np.nansum(((above_cp) * dt) / n1) /1000
W_ox = np.nansum(((vo2_excess_conv) * dt) / n1) /1000
#print(W_nonox,W_ox)
W_nonox = np.nansum(((above_cp - vo2_excess_conv) * dt) / n1) /1000 #put it from J to kJ
#print(n1)

```

```

# --- NEW: Extract C1est at VO2max index ---

if "VO2_corr" in max_df.columns or "VO2_raw" in max_df.columns:

    vo2 = max_df["VO2_corr"].values if "VO2_corr" in max_df.columns else
max_df["VO2_raw"].values

    vo2max_idx = np.argmax(vo2)

    C1est_max = (max_df["C1est"].values)[vo2max_idx]

    merged_data[subj]['max'].attrs['C1est_max'] = C1est_max

    merged_data[subj]['max'].attrs['MO_kjs'] = C1est_max * np.max(vo2) * (1/60)

else:

    print('erC1',str(subj))

return W_nonox, W_ox

```

AG Estimation Code

```

def estimate_AG(subj, merged_data, mp_factor=0.08577,PC_max=20,
muscle_fraction=0.5,phi=.3):

    import numpy as np

    import matplotlib.pyplot as plt

    try:

        max_df = merged_data[subj]["max"]

        eco = merged_data[subj]["eco"]

        VO2max_kg = max_df.attrs["VO2max_kg"] # ml/kg/min

```

```

weight_kg = max_df.attrs["weight_kg"]
VO2max_Lmin = VO2max_kg * weight_kg / 1000
Wprime = max_df.attrs["Wprime_nonox"]

# Threshold VO2 values
alpha = eco.attrs["thresholds"]["LT1"]["LT_VO2"] / VO2max_Lmin
beta = eco.attrs["thresholds"]["LT2"]["LT_VO2"] / VO2max_Lmin

# Constants
phi = phi
theta = alpha * (1 - phi)
MP = mp_factor * weight_kg * 1000 # W
MO = merged_data[subj][['max']].attrs['MO_kjs'] #KJ/sec
MO_MP = MO*1000 / MP

# λ
lambda_val = 1 - (theta * ((1 / alpha) - MO_MP)) / ((1 / beta) - MO_MP)

# l @ Pcrit
numerator = 1 - lambda_val
denominator = (MO_MP * ((1 - theta - lambda_val) / (1 - phi))) + 1
l_pcrit = numerator / denominator

# --- Physiological Estimation of AP and AT ---
muscle_mass = muscle_fraction * weight_kg

```

```

# AT = [La]_LT1 / theta * m_muscle * C2
AT = (3 / theta) * muscle_mass * 100 # 100 J/mmol

# AP = [PC]_max * m_muscle * C3
AP = PC_max * muscle_mass * 43.3 # 43.3 J/mmol

# AG
AG = (Wprime*1000 - AP * l_pcrit - AT * theta) / (l_pcrit - theta)
AP = AP/1000
AG = AG/1000
AT=AT/1000
MO=MO
MP=MP/1000

# --- Plot AG breakdown ---
plt.figure(figsize=(8, 5))
components = [AT * theta, AP * 1, AG * (1-theta - lambda_val)]
labels = ["G Capillary", "P Tank", "G Main"]
plt.bar(labels, components, color=["blue", "green", "purple"])
plt.title(f"{subj} – AG Component Breakdown")
plt.ylabel("Joules")
plt.grid(True)
plt.tight_layout()
plt.show()

return {

```

```
"AG": round(AG,2),
"Wprime": round(Wprime,3),
"MO": round(MO,2),
"MP": round(MP,2),
"theta": round(theta,4),
"beta": round(beta,3),
'alpha': round(alpha,3),
"lambda": round(lambda_val,4),
'muscle_mass': round(muscle_mass,3),
"L_pcrit": round(L_pcrit,3),
"AT": round(AT,3),
"AP": round(AP,3),
"phi": phi
}
```

except Exception as e:

```
print(f"✘ {subj}: {e}")
```

```
return None
```

Grade Adjusted Pace Code

```
import numpy as np
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

```
 #(x, y) -> z
```

```

numbers = np.arange(1, 13)
repeated_numbers = np.repeat(numbers, 14)
#print(repeated_numbers)
numbers2 = np.arange(6, 13, 0.5)
repeated_numbers2 = np.tile(numbers2, 12)
#print(repeated_numbers2)
x = repeated_numbers
#np.array([1,1,1,1,1,1,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4,5,5,5,5,5,5,6,6,6,6,6,6,7,7,7])
#gradients
y = repeated_numbers2 #np.array([6, 6.5, 7, 7.5, 8, 8.5,6, 6.5, 7, 7.5, 8, 8.5,6, 6.5, 7, 7.5, 8,
8.5,6, 6.5, 7, 7.5, 8, 8.5,6, 6.5, 7, 7.5, 8, 8.5,6, 6.5, 7, 7.5, 8, 8.5,]) #mph
z = np.array([6.5,
7,
7.5,
7.9,
8.4,
8.9,
9.4,
9.9,
10.5,
11,
11.5,
12,
12.5,
13,
7,
7.4,

```

7.9,

8.3,

8.8,

9.3,

9.9,

10.4,

10.9,

11.4,

12,

12.5,

13,

13.6,

7.4,

7.8,

8.3,

8.7,

9.2,

9.8,

10.3,

10.8,

11.4,

11.9,

12.5,

13,

13.6,

14.1,

7.8,

8.2,

8.7,

9.2,

9.7,

10.2,

10.8,

11.3,

11.9,

12.5,

13,

13.6,

14.1,

14.7,

8.2,

8.6,

9.1,

9.6,

10.1,

10.7,

11.2,

11.8,

12.4,

13,

13.6,

14.1,

14.7,

15.3,

8.6,

9,

9.5,

10,

10.5,

11.1,

11.7,

12.3,

12.9,

13.5,

14.1,

14.7,

15.3,

15.9,

8.9,

9.4,

9.9,

10.4,

11,

11.6,

12.2,

12.8,

13.4,

14,

14.7,

15.3,

15.9,

16.5,

9.3,

9.8,

10.3,

10.8,

11.4,

12,

12.7,

13.3,

14,

14.6,

15.2,

15.9,

16.5,

17.1,

9.7,

10.2,

10.7,

11.3,

11.9,

12.5,

13.2,

13.8,

14.5,

15.2,

15.8,

16.5,

17.1,

17.8,

10,

10.6,

11.1,

11.7,

12.4,

13,

13.7,

14.4,

15,

15.7,

16.4,

17.1,

17.7,

18.4,

10.4,

11,

11.5,

12.2,

12.8,

13.5,

14.2,

14.9,

15.6,

16.3,

17,

17.7,

18.4,

19,

10.8,

11.4,

12,

12.6,

13.3,

14,

14.7,

15.4,

16.2,

16.9,

17.6,

18.3,

19,

19.7

])

Combine x and y into a single array of features

X = np.column_stack((x, y))

```

degree = 2

poly = PolynomialFeatures(degree)

X_poly = poly.fit_transform(X)

model = LinearRegression()

model.fit(X_poly, z)

z_pred = model.predict(X_poly)

mse = mean_squared_error(z, z_pred)

print(f"Mean Squared Error: {mse}")

print("Coefficients:")

print(model.coef_)

print("Intercept:")

print(model.intercept_)

# Example of predicting a new value

x_new = np.array([7])

y_new = np.array([8.5])

X_new = np.column_stack((x_new, y_new))

X_new_poly = poly.transform(X_new)

z_new_pred = model.predict(X_new_poly)

print(f"Predicted z for (x, y) = (7%, 8.5mph): {z_new_pred[0]}")

print("equiv flat MPH = "+str(round(model.coef_[1],5))+"(%)"+
"+str(round(model.coef_[2],5))+"(mph)+ "+str(round(model.coef_[3],5))+"(%^2)+
"+str(round(model.coef_[4],5))+"(% * mph)+ "+str(round(model.coef_[5],5))+"(mph^2) +
"+str(round(model.intercept_,5)))

print("equiv Grade Adjusted Pace (GAP) (min/mile) = (60 / equiv flat MPH)")

```

Muscle-Blood Lactate Function Code

```
# lactate tank testing setup
```

```
def lab_ss(lam,RBLA=1.5,RMLA=1.5):
```

```
    """ Steady-state blood lactate [Lab] using empirical regression. """
```

```
    labslope = .5551
```

```
    labint = 1.2226
```

```
    if RBLA != 1.5:
```

```
        newint = RBLA - labslope*RMLA
```

```
        return labslope * lam + newint
```

```
    else:
```

```
        return labslope * lam + labint
```

```
def compute_tau(lab, lam, RMLA=1.5, Mlacmax=30.0, min_tau=26, max_tau=96):
```

```
    """Linearly scaled tau from max_tau (low lam) to min_tau (high lam)."""
```

```
    # Clamp lam within [RMLA, Mlacmax] to avoid extrapolation
```

```
    lam_clamped = np.clip(lam, RMLA, Mlacmax)
```

```
    # Normalize between 0 and 1
```

```
    lam_norm = (lam_clamped - RMLA) / (Mlacmax - RMLA)
```

```
    # Inverse linear interpolation (high lactate = faster tau)
```

```
    tau = max_tau - lam_norm * (max_tau - min_tau)
```

```
    return tau
```

```
def update_lab(lab, lam, dt, RMLA=1.5, Mlacmax=30.0, RBLA=1.5):
```

```
    tau = compute_tau(lab, lam, RMLA=RMLA, Mlacmax=Mlacmax)
```

```
    if lam <= RMLA:
```

```

lab_new = RBLA
else:
lab_new = lab + (lab_ss(lam,RBLA,RMLA) - lab) / tau * dt
return lab_new

```

Discrete Digital Twin Simulation Function Code

```

def dtsim6b(MG=9.15, MO=1.34, MP=4.48, MR=3.66, phi=.3, lamb=.38, AT=11.27,
AP=27.79, AG=320.3, theta=.43, PCmax=20, nums=None, cad=None, tor=None,
t1pct=None, t2pct=None, showfig=True, savefigs=True, lactau=30, RBLA=1.5,
C1=20.9, C2=100, C3=43.3, RMLA=1.5,
xpmec=0, xh=0, xl=0, xlddt=0, xlab=2, xpcconc=20, xhddot=0,
eta_bike=.24551,eta_run=6, sport="bike",Mlacmax=18,
MRaeint=-40,MRaeslope=356,atw=61, etaslope= -.25, etaint=8.8, prev_pmec=None,
Aacc=114, maxRER=1.1):
global t_fail, t_fail_maxpwr, t_failcurVmax
t_fail = t_fail_maxpwr = t_failcurVmax = 0
C1 = C1 # 20.9 # J/ml energy/ml of oxygen, Fixed for now
C2 = C2 # 100 # J/mmol joules/accum lac in muscle
C3 = C3 # 43.3 # J/mmol of PC stores
atw = atw # kg total body mass
RBLA=RBLA
prev_pmec=prev_pmec
if sport == 'bike':
eta = eta_bike #0.24551 #Wmec/Wphys base value for Cyclist(1?) from Boillet
else:

```

```
# eta = eta_run/1000 #GAP speed(mph) per VO2 wattage (KILO Watts-> Watts). around  
6-9 mph / kW
```

```
eta = (etaslope * xpmec + etaint)/1000 # GAPmph / W
```

```
m_muscle = 1000 * AP / (C3 * PCmax) # print('m_musc',m_muscle)
```

```
Pmecmax = MP * eta * 1000 # W = J/s max theo fresh mech pwr
```

```
if t1pct == None and t2pct == None:
```

```
    t1pct = 50
```

```
    t2pct = 50
```

```
    # print('shlapped t1,t2 %@50ea')
```

```
if t1pct == None:
```

```
    t1pct = 100 - t2pct
```

```
if t2pct == None:
```

```
    t2pct = 100 - t1pct
```

```
xh = xh
```

```
xl = xl
```

```
xg = 1 - xl #q
```

```
xldt = xldt
```

```
curPmecmax = MP*eta*1000
```

```
if xl > theta:
```

```
    curPmecmax = min(curPmecmax * ((1-lamb-(xl*.8))/(1-lamb-theta)),MP*eta*1000)
```

```
else:
```

```
    curPmecmax = curPmecmax * ((1-lamb-theta)/(1-lamb-theta))
```

if $x_{pmech} > curP_{mecmax}$:

$x_{pmech} = curP_{mecmax}$

$x_{pphys} = x_{pmech} / \eta / 1000$ #in kJ/s

$x_{lacb} = x_{lacb}$

if $x_{pmech} \neq 0$:

$z_{MaxP_{mecAerobic}} = MO * 1000 * \eta$

else:

if $prev_p_{mec}$ is not None:

$z_{MaxP_{mecAerobic}} = MO * 1000 * (\eta_{slope} * prev_p_{mec} + \eta_{aint}) / 1000$

else:

$z_{MaxP_{mecAerobic}} = MO * 1000 * \eta$

$x_{pctp_{mecmax}} = x_{pmech} * 100 / P_{mecmax}$

if $x_l > \theta$:

if $x_{dldt} \geq 0$:

$a_{cur} = (AP * AG / MG) * (1 - \lambda)$

$b_{cur} = (((MO * (1 - \lambda)) / (MG * (1 - \phi))) + 1) * AG + AP$

else:

$a_{cur} = (AP * AG / MR) * (1 - \lambda)$

$b_{cur} = (((MO * (1 - \lambda)) / (MR * (1 - \phi))) + 1) * AG + AP$

$q = 0.3$ # $q_list[i] = q$

else:

```

if xlddt >= 0:
    a_cur = (AP * AT / MG) * (1 - lamb)
    b_cur = (((MO * (1 - lamb)) / (MG * (1 - phi))) + 1) * AT + AP
else:
    a_cur = (AP * AT / MR) * (1 - lamb)
    b_cur = (((MO * (1 - lamb)) / (MR * (1 - phi))) + 1) * AT + AP
q = 0.1 # q_list[i] = q

c_cur = MO / (1 - phi)
#lac_m_scalor1 = 1
#lac_m_scalor2 = 1
if xl <= theta:
    Lac_m = (1 / C2) * xl * 1000 * AT #* lac_m_scalor1
else:
    #print('AGnow') #ZYZZ
    Lac_m = (1 / C2) * ((xl - theta) * AG * 1000 + theta * AT * 1000) #- RMLA*m_muscle #*
lac_m_scalor2 - RMLA # +.4*m_muscle#HOLY TODO FIX DIRTY
# Lac_m = (1 / C2) * xl * 1000 * AT * lac_m_scalor1 if xl <= theta else (1 / C2) * (
# (xl - theta) * AG * 1000 + theta * AT * 1000) * lac_m_scalor2 - RMLA #
+.4*m_muscle#HOLY TODO FIX DIRTY
Lac_m /= m_muscle
Lac_m += RMLA # 1.5 baseline musc lac

Lac_b = update_lab(xlacb, Lac_m, 1, RMLA, Mlacmax, RBLA=RBLA)
next_lacb = Lac_b # done RETURN

L_ddot = (xpphys - b_cur * xlddt - c_cur * xl) / a_cur

```

```

h_dot = xdldt + ((1 - lamb) * (AG if xl > theta else AT) / (MG if xdldt >= 0 else MR)) * L_ddot
# h_dot = L_dot*AG/AP + ((1 - lamb) * (AG if l > theta else AT) / (MG if L_dot >= 0 else MR)) *
L_ddot

# h_dot = P_phys_values[i] - AG*L_dot/AP - MO*(h_list[i]/(1-theta))

#arb_h_dot_cutoff = 0.25 # FIX TODO ARB #ZEZ

#h_dot = max(min(h_dot, arb_h_dot_cutoff), -arb_h_dot_cutoff) #ZEZ

VP = AP * h_dot

L_new = xl + xdldt # / 2 #new

# if L_new > 1-lamb: #q

# L_new = 1-lamb

if xl > theta:

    VG = AG * xdldt

    q = 5

elif xl < theta:

    if L_new > theta: # SEMI ARB wtf is 1.5

        VG = (L_new-xl)*AT #+ (L_new-theta)*(AG-AT)*.25 # (theta - xl) * AT #* 1.5 # + (L_new-
theta)*AG*1 ##+ (L_new-theta)*AG*1

        # print('toG', i, 'VGunsc', VG)

        q = 10

        VP = VP #* .5

        # print('Vg', VG, 'VP', VP)

    else:

        VG = AT * xdldt

        q = 0

scalor = 1000 / C1

VP = VP * scalor # 1000/C3 # * .975

```

```
VG = VG * scalar # 1000/C2 # * 1.025
```

```
VO2_derived = (xpphys * scalar - VP - VG)
```

```
#print(VO2_derived)
```

```
P_VO2 = VO2_derived * C1 * 64 * eta * (1 / 60) # * eta # pmec eff from vo2
```

```
if P_VO2 >= zMaxPmecAerobic:
```

```
    P_VO2 = zMaxPmecAerobic
```

```
    #print('VO2max-d out')
```

```
if P_VO2 < 0:
```

```
    P_VO2 = 0
```

```
VO2_Lmin = P_VO2 * 60 * (1/C1) * (1/1000) * (1/eta) # physiological L/min of VO2
```

```
#lidar attempts (wrong time course tbh:
```

```
#totalVO2 = (xpphys*1000)*60*(1/C1)*(1/1000) + .00000001 # for testing
```

```
#print('totvo2: '+str(round(totalVO2,2)))
```

```
#RQ = (((118 + P_VO2/eta) / (4184/60) / totalVO2) - 3.8149) / 1.232 # from lidar2023
```

```
#TODO: check that VO2_derived is in L/min
```

```
# swapped xpphys*1000 for P_VO2 eqn1 of Lidar2023...unsure tbh
```

```
# 118W + PVO2 bc bmr is part of MRae
```

```
G_kJ_left = 0
```

```
#G_rel_curr = (xg - lamb) / (1 - lamb)
```

```
if xl > theta:
```

```
    G_kJ_left = AG*(1-theta-lamb) - (xl-theta)*AG
```

```
elif xl <= theta:
```

```
    G_kJ_left = (1-theta-lamb)*AG + theta*AT - xl*AT
```

```

#LIDAR2023

lidarwgt=74 #kg, sd 6

MRrest = 143*atw/lidarwgt

G_kJ_full = AG * (1-theta-lamb) + AT * theta #AG * ((1 - theta - lamb) / (1 - lamb)) + AT *
((theta) / (1 - lamb)) #xl=0@full

x4 = 1- (G_kJ_left/G_kJ_full)

# if x4>1: #q

# print(x4,'x4',G_kJ_left/G_kJ_full,'Gfrac',xl,'L')

Aacc=Aacc

Bacc=.97

MRacc = ((Aacc*Bacc*x4*1.03)/lidarwgt)*atw

accfrac = MRacc / (((Aacc*Bacc*1*1.03)/lidarwgt)*atw)

# if accfrac > 1: #q

# print(accfrac,"accfrac over 1")

# accfrac = 1

maxRER = maxRER*Bacc -.01 #adjustment for x4 scaling #q

rer_range = maxRER - .7

RER = np.clip( ((accfrac)*rer_range) + .7 ,.7,maxRER+.16)#ZYZZ

VCO2_Lmin = VO2_Lmin * RER

Af=158 #sd20, r50-250

Bf=.7267 #2.96 #sd.09 r2.2-3.5

MRf = ((Af + Bf*(xpmecc/eta))/lidarwgt)*atw if xpmecc != 0 else 0 #scaled to per lidarkg,
rescaled.

```

```

Bve = .93 #L-Q distribution
VO2_VO2max = P_VO2/zMaxPmecAerobic
VE_VEEmax = 1.2499*(VO2_VO2max**3) - 1.5287*(VO2_VO2max**2) +
1.2687*(VO2_VO2max)
MRve = 182 * (Bve*(VE_VEEmax) + (1-Bve)*((VE_VEEmax)**2)) *atw/lidarwgt

#*atw*atw/lidarwgt/61

MRae = min((MRaeslope * (VO2_Lmin) + MRaeint),MO * 1000) #ZYZZ + MRacc? + MRve?
partial?

xh += h_dot #
next_h = xh # done return
cur_dhdt = h_dot # unused
L_new = xl + xdldt #q # done return
L_dot_new = xdldt + L_ddot # done return
# xl = L_new
# xdldt = L_dot_new
# next_l = L_new
next_g = 1 - L_new # implied, can return if needed
next_hddot = h_dot - xhddot # -h_ddot_list[i] #done return

PC_conc = ((1 - xh) * AP * 1000) / (C3 * m_muscle)
PC_pwr = (xpcconc - PC_conc) * m_muscle * C3 * eta
if PC_pwr < 0:
    #print(PC_pwr, "PCpwr belo zero") #q
    PC_pwr = 0

```

#TODO: was nec bc went negative... maybe later fix? see lidar 2023-maybe need negative for recovery?

```
an_pwr = xpmec - P_VO2 - PC_pwr
```

```
if an_pwr < 0:
```

```
    #print(an_pwr,"anpwr belo zero") #qq
```

```
    an_pwr = 0
```

```
# Ptank_list = 1 - h_list #duh
```

```
zANpwr = xpmec - P_VO2
```

```
oxyprwtext = 'cur Po2: ' + str(round(P_VO2, 1)) + 'w'
```

```
vo2curtext = 'relVO2cur: ' + str(round(VO2_derived * 1000 / C1, 1)) + ' ml/kg/min'
```

```
# TODO FIX not sure ab VO2_derived[i] - needs to b current VO2 pwr in kJ/s?
```

```
# NOTE: not in loop:
```

```
zPCrit = eta * ((1 - lamb) / ((1 - phi) + (MO / MP) * (1 - theta - lamb))) * MO * 1000
```

```
zVO2max = round((MO * 1000) / C1) # ml/kg/min?? not sure
```

```
zCPpctofVO2max = 100 * MO * (1 - lamb) # not sure
```

```
zPatLT1 = 1000 * eta * MO * (theta / (1 - phi))
```

```
zPatCP = eta * ((1 - lamb) / ((1 - phi) + (MO / MP) * (1 - theta - lamb))) * MO * 1000
```

```
zBeta = (1 - lamb) / ((1 - phi) + (MO / MP) * (1 - theta - lamb))
```

```
prev_pmec = xpmec
```

```
#
```

```
0  1  2  3  4 5  6  #7  8  9  10  11  12  13  14  
15 16 17 18 19 20
```

```

# return l_list, h_list, VO2_derived, VO2permin_l, h, Lac_m_list, PC_conc_list,
PC_pwr_list, dhdt_list, P_mec_values, P_VO2_list, m_muscle, dldt_list, VP_list, VG_list,
q2_list, tor, cad, Pmecmax, pct_pmecmax_list, Lac_b_list

# 0, 1, 2, 3, 4, 5, 6, 7 8

return l_new, l_dot_new, next_hddot, next_h, next_lacb, PC_conc, PC_pwr, an_pwr,
P_VO2, Lac_m, MRacc,MRf, xpphys*1000, MRae, MRve, MRrest, prev_pmec, RER,
VCO2_Lmin,VO2_Lmin

```

Constant-Power Batch Simulation Code

```

# --- Batch Simulation Function ---

def run_dtsim_batch(power=250, cadence=90, duration=300, theta=0.43, mo=1.34,

    sport='bike',MG=9.15,MP=4.48,MR=3.66, AT=11.27, AP=27.79,

    AG=320.3,PCmax=20,RBLA=1.5,C1=20.9,C2=100,C3=43.3,RMLA=1.5,

    Mlacmax=18,lamb=.38,eta_run=6, eta_bike=.24551,

    MRaeint=-40, MRaeslope=356,atw=61, etaslope= -.25, etaint=8.8,

    prev_pmec=None,Aacc=114,maxRER=1.1):

    torque = (power * 60) / (2 * np.pi * cadence)

    xh = xl = xdldt = xhddot = 0

    xlacb = xlacm = 1.5

    xpcconc = 20

    time_list = []

    pmec_list, pglc_list, pvo2_list, pcpwr_list = [], [], [], []

    lacb_list, lacm_list = [], []

    h_list, l_list = [], []

    MRacc_list, MRf_list, Pphys_list, MRae_list,MRve_list, MRrest_list = [], [], [], [],[],[]

```

```

prev_pmec_list = []
# prev_vo2_list = []
RER_list = []
VCO2_Lmin_list = []
VO2_Lmin_list = []
#, RER, VCO2_Lmin,VO2_Lmin
for t in range(duration):
    l_new, l_dot_new, next_hddot, next_h, next_lacb, PC_conc, PC_pwr, an_pwr, P_VO2,
    Lac_m, MRacc, MRf, Pphys, MRae, MRve, MRrest,prev_pmec, RER, VCO2_Lmin,VO2_Lmin
= dtsim6b(
    xpmec=power,
    xh=xh,
    xl=xl,
    xdldt=xdldt,
    xlacb=xlacb,
    xpcconc=xpcconc,
    xhddot=xhddot,
    MO=mo,
    theta=theta,
    tor=torque,
    cad=cadence, sport=sport, MG=MG, MP=MP, MR=MR, AT=AT, AP=AP, AG=AG,
    PCmax=PCmax, RBLA=RBLA, C1=C1, C2=C2, C3=C3, RMLA=RMLA, lamb=lamb,
    eta_run=eta_run, Mlacmax=Mlacmax, MRaeint=MRaeint, MRaeslope=MRaeslope,
    atw=atw, eta_bike=eta_bike, etaslope= etaslope,
    etaint=etaint, prev_pmec=prev_pmec, Aacc=Aacc, maxRER=maxRER    )

```

```
xlacb, xh, xl, xlddt, xpcconc, xhddot = next_lacb, next_h, l_new, l_dot_new, PC_conc,  
next_hddot
```

```
time_list.append(t)
```

```
pmec_list.append(power)
```

```
pglc_list.append(an_pwr)
```

```
pvo2_list.append(P_VO2)
```

```
pcpwr_list.append(PC_pwr)
```

```
lacb_list.append(next_lacb)
```

```
lacm_list.append(Lac_m)
```

```
h_list.append(xh)
```

```
l_list.append(l_new)
```

```
MRacc_list.append(MRacc)
```

```
MRf_list.append(MRf)
```

```
MRae_list.append(MRae)
```

```
MRve_list.append(MRve)
```

```
MRrest_list.append(MRrest)
```

```
Pphys_list.append(Pphys)
```

```
RER_list.append(RER)
```

```
VCO2_Lmin_list.append(VCO2_Lmin)
```

```
VO2_Lmin_list.append(VO2_Lmin)
```

```
prev_pmec_list.append(prev_pmec)
```

```
# --- Plotting ---
```

```
# Plot 1: Tanks & Lactate
```

```
plt.figure(figsize=(12, 5))
plt.plot(time_list, 1 - np.array(h_list), label='P Phosphagen')
plt.plot(time_list, 1 - np.array(l_list), label='G Glycolytic')
plt.plot(time_list, lacb_list, label='lacb Blood Lactate')
plt.plot(time_list, lacm_list, label='lacm Muscle Lactate')
plt.title('Tanks & Lactate')
plt.xlabel('Time (s)')
plt.ylabel('Level / [Lactate] (mmol/L or mmol/kg_w.w.)')
plt.legend()
plt.grid(True)
plt.show()
```

Plot 2: Power Breakdown

```
plt.figure(figsize=(12, 5))
#plt.figure(figsize=(7, 7))
plt.plot(time_list, pglc_list, label='Glycolytic Power')
plt.plot(time_list, pvo2_list, label='Aerobic Power')
plt.plot(time_list, pmec_list, label='Demanded Pmec')
plt.plot(time_list, prev_pmec_list, label='Actual Pmec')
plt.plot(time_list, pcpwr_list, label='PCr Power')
plt.title('Power Breakdown')
plt.xlabel('Time (s)')
plt.ylabel('Power Output (W)')
plt.legend()
plt.grid(True)
plt.show()
```

```

time = np.array(time_list)
MRrest = np.array(MRrest_list)
MRf = np.array(MRf_list)
MRve = np.array(MRve_list)
MRacc = np.array(MRacc_list)
MRae = np.array(MRae_list)
Pphys = np.array(Pphys_list)

# Stack layers on top of MRrest baseline
MRf_total = MRrest + MRf
MRf_ve = MRf_total + MRve
MRf_ve_acc = MRf_ve + MRacc

# --- Plot ---
plt.figure(figsize=(12, 5))

# Base resting MR: from 0 to MRrest
plt.fill_between(time, 0, MRrest, color='lightgrey', alpha=0.6, label='Resting MR (MRrest)')

# Shade MRf: from MRrest → MRf+MRrest
plt.fill_between(time, MRrest, MRf_total, color='lightgreen', alpha=0.4,
                hatch='//', edgecolor='green', linewidth=0.0, label='Functional (MRf)')

# Shade MRve: from MRf+MRrest → MRf+MRve+MRrest

```

```
plt.fill_between(time, MRf_total, MRf_ve, color='purple', alpha=0.4, label='Ventilation (MRve)')
```

```
# Shade MRacc: from MRf+MRve+MRrest → +MRacc
```

```
plt.fill_between(time, MRf_ve, MRf_ve_acc, color='black', alpha=0.3, label='Accum. metabolites (MRacc)')
```

```
# MRae as solid yellow, plotted last
```

```
plt.plot(time, MRae, '-', color='gold', linewidth=2, label='MRae - aerobic supply', zorder=10)
```

```
# Optional outlines (dashed)
```

```
plt.plot(time, MRf_total, '--', color='green', linewidth=1)
```

```
plt.plot(time, MRf_ve, '--', color='purple', linewidth=1)
```

```
plt.plot(time, MRf_ve_acc, '--', color='black', linewidth=1)
```

```
plt.plot(time, Pphys, '--', color='gray', label='Pphys - Boillet')
```

```
# Final plot settings
```

```
plt.title('Metabolic Supply/Demand Rates (Stacked on MRrest)')
```

```
plt.xlabel('Time (s)')
```

```
plt.ylabel('Metabolic Rate (J/s)')
```

```
plt.grid(True)
```

```
plt.legend()
```

```
plt.show()
```

Real-Time Twin Localhost Simulation Code

Uses `dtsim()` iterative function (not included here) to do live animation in local browser

```

import streamlit as st
import plotly.graph_objects as go
import numpy as np
import time
from dtsim6c import dtsim6b

st.set_page_config(layout="wide")
st.title("🚴 ♂ DTsim6b - a6b")

import math

# --- Persistent defaults
if "power_val" not in st.session_state:
    st.session_state.power_val = 100.0
if "torque_val" not in st.session_state:
    st.session_state.torque_val = 30.0
if "cadence_val" not in st.session_state:
    st.session_state.cadence_val = 90.0

# --- Exclusive lock selector
lock_state = st.radio("🔒 Lock one to auto-calculate:", ["Power", "Torque", "Cadence"],
index=0, horizontal=True)

# --- Layout
colA, colB, colC, colD = st.columns(4)
colE, colF = st.columns(2)

# --- Editable inputs (only 2 are editable at a time)
if lock_state != "Power":
    with colA:
        st.session_state.power_val = st.number_input("Power (W)",
value=st.session_state.power_val, step=5.0, key="power_input")
if lock_state != "Torque":
    with colB:
        st.session_state.torque_val = st.number_input("Torque (Nm)",
value=st.session_state.torque_val, step=1.0, key="torque_input")
if lock_state != "Cadence":
    with colC:

```

```

    st.session_state.cadence_val = st.number_input("Cadence (RPM)",
value=st.session_state.cadence_val, step=1.0, key="cadence_input")

# --- Computed field (disabled input)
with (colA if lock_state == "Power" else colB if lock_state == "Torque" else colC):
    if lock_state == "Power":
        if st.session_state.cadence_val > 0:
            power_calc = (2 * math.pi * st.session_state.torque_val *
st.session_state.cadence_val) / 60
        else:
            power_calc = 0
        st.number_input("Power (W)", value=round(power_calc, 2), disabled=True,
key="computed_power")
        st.session_state.power_val = power_calc

    elif lock_state == "Torque":
        if st.session_state.cadence_val > 0:
            torque_calc = (st.session_state.power_val * 60) / (2 * math.pi *
st.session_state.cadence_val)
        else:
            torque_calc = 0
        st.number_input("Torque (Nm)", value=round(torque_calc, 2), disabled=True,
key="computed_torque")
        st.session_state.torque_val = torque_calc

    elif lock_state == "Cadence":
        if st.session_state.torque_val > 0:
            cadence_calc = (st.session_state.power_val * 60) / (2 * math.pi *
st.session_state.torque_val)
        else:
            cadence_calc = 0
        st.number_input("Cadence (RPM)", value=round(cadence_calc, 2), disabled=True,
key="computed_cadence")
        st.session_state.cadence_val = cadence_calc

# --- Remaining inputs
with colD:
    update_interval = st.number_input("Update Intvl (s)", min_value=0.0, max_value=2.0,

```

```

value=1.0, step=0.2, format="%0.1f", key='update_input')
with colE:
    mo = st.number_input("MO (W overhead?)", min_value=0.0, max_value=500.0,
value=1.34, step=.02, key='mo_input')
with colF:
    theta = st.number_input("θ (rad or deg)", min_value=0.0, max_value=360.0, value=0.43,
step=.02, key='theta_input')

col1, col2 = st.columns([2, 1])

if 'running' not in st.session_state:
    st.session_state.running = -1

if col2.button("(RE)Start Simulation"): # Init/reset
    st.session_state.running = 1
    for key in ['pmec_list', 'pglc_list', 'pvo2_list', 'pcpwr_list', 'lacb_list', 'lacm_list',
        'h_list', 'l_list', 'time_list', 'MRacc_list', 'MRf_list', 'Pphys_list']:
        st.session_state[key] = []
    st.session_state.update(dict(
        xlacb=1.5, xlacm=1.5, xh=0, xl=0, xpmec=0,
        xdldt=0, xpcconc=20, xhddot=0, t=0
    ))

if col2.button("Stop and Clear Simulation"):
    st.session_state.running = -1

if col2.button("Pause and Save Simulation"):
    st.session_state.running = 2 # triggers one-frame draw

if col2.button("Resume Simulation"):
    st.session_state.running = 1

power_container = st.empty() # Plot containers
tank_container = st.empty()
metabolic_container = st.empty()

if st.session_state.running == 1: # LIVE SIMULATION LOOP
    last_plot_time = time.time()

```

```

while st.session_state.running == 1:
    power_now = st.session_state.power_val
    L_new, L_dot_new, next_hddot, next_h, next_lacb, PC_conc, PC_pwr, an_pwr, P_VO2,
Lac_m, MRacc, MRf, Pphys = dtsim6b(
    xpmec=power_now,
    xh=st.session_state.xh,
    xl=st.session_state.xl,
    xdldt=st.session_state.xdldt,
    xlacb=st.session_state.xlacb,
    xpcconc=st.session_state.xpcconc,
    xhddot=st.session_state.xhddot,
    MO=st.session_state.mo_input,
    theta=st.session_state.theta_input,
    tor=st.session_state.torque_input,
    cad=st.session_state.cadence_input
)

```

```

st.session_state.xlacb = next_lacb
st.session_state.xh = next_h
st.session_state.xl = L_new
st.session_state.xdldt = L_dot_new
st.session_state.xpcconc = PC_conc
st.session_state.xhddot = next_hddot

```

Store history

```

st.session_state.pmec_list.append(power_now)
st.session_state.pglc_list.append(an_pwr)
st.session_state.pvo2_list.append(P_VO2)
st.session_state.pcpwr_list.append(PC_pwr)
st.session_state.lacb_list.append(next_lacb)
st.session_state.lacm_list.append(Lac_m)
st.session_state.h_list.append(next_h)
st.session_state.l_list.append(L_new)
st.session_state.time_list.append(st.session_state.t)
st.session_state.MRacc_list.append(MRacc)
st.session_state.MRf_list.append(MRf)
st.session_state.Pphys_list.append(Pphys)
st.session_state.t += 1

```

```
if time.time() - last_plot_time > 1: #og 2 sec betw upd
    last_plot_time = time.time() # Update Plotly charts
```

```
fig1 = go.Figure()
# Left y-axis: Tank levels
fig1.add_trace(go.Scatter(
    x=st.session_state.time_list,
    y=1 - np.array(st.session_state.h_list),
    name="P Phosphagen",
    yaxis='y1'
))
fig1.add_trace(go.Scatter(
    x=st.session_state.time_list,
    y=1 - np.array(st.session_state.l_list),
    name="G Glycolytic",
    yaxis='y1'
))
```

```
# Right y-axis: Lactate
fig1.add_trace(go.Scatter(
    x=st.session_state.time_list,
    y=st.session_state.lacb_list,
    name="lacb Blood Lactate",
    yaxis='y2'
))
fig1.add_trace(go.Scatter(
    x=st.session_state.time_list,
    y=st.session_state.lacm_list,
    name="lacm Muscle Lactate",
    yaxis='y2'
))
```

```
# Layout with dual y-axes
fig1.update_layout(
    title="Tanks & Lactate",
    xaxis_title="Time (s)",
    yaxis=dict(title="Tank Fill Level", side='left'),
```

```
yaxis2=dict(title="[Lactate] (mmol/L)", overlaying='y', side='right'),  
legend=dict(x=0, y=1.1, orientation="h")  
)
```

```
tank_container.plotly_chart(fig1, use_container_width=True)
```

```
fig2 = go.Figure()
```

```
# Left y-axis: Power outputs
```

```
fig2.add_trace(go.Scatter(  
    x=st.session_state.time_list,  
    y=st.session_state.pglc_list,  
    name="Glycolytic Power",  
    yaxis='y1'
```

```
))
```

```
fig2.add_trace(go.Scatter(  
    x=st.session_state.time_list,  
    y=st.session_state.pvo2_list,  
    name="Aerobic Power",  
    yaxis='y1'
```

```
))
```

```
fig2.add_trace(go.Scatter(  
    x=st.session_state.time_list,  
    y=st.session_state.pmec_list,  
    name="Output Pmec",  
    yaxis='y1'
```

```
))
```

```
fig2.add_trace(go.Scatter(  
    x=st.session_state.time_list,  
    y=st.session_state.pcpwr_list,  
    name="PCr Power",  
    yaxis='y1'
```

```
))
```

```
fig2.update_layout(  
    title="Power Breakdown & Metabolic Rates",  
    xaxis_title="Time (s)",  
    yaxis=dict(title="Power Output (W)", side="left"),
```

```

    yaxi2=dict(title="Metabolic Rate (J/s)", overlaying="y", side="right"),
    legend=dict(x=0, y=1.1, orientation="h")
)

power_container.plotly_chart(fig2, use_container_width=True)

fig3 = go.Figure()

# Right y-axis: MRacc and MRf
fig3.add_trace(go.Scatter(
    x=st.session_state.time_list,
    y=st.session_state.MRacc_list,
    name="MRacc - accum. metabolites",
    yaxi='y1',
    line=dict(dash='dot')
))
fig3.add_trace(go.Scatter(
    x=st.session_state.time_list,
    y=st.session_state.MRf_list,
    name="MRf - functional",
    yaxi='y1',
    line=dict(dash='dash')
))
fig3.add_trace(go.Scatter(
    x=st.session_state.time_list,
    y=st.session_state.Pphys_list,
    name="Pphys Boillet",
    yaxi='y1',
    line=dict(dash='dash')
))

fig3.update_layout(
    title="Metabolic Supply/Demand Rates",
    xaxis_title="Time (s)",
    yaxi=dict(title="Power Output (W)", side="left"),
    yaxi2=dict(title="Metabolic Rate (J/s)", overlaying="y", side="right"),
    legend=dict(x=0, y=1.1, orientation="h")
)

```

```

)

metabolic_container.plotly_chart(fig3, use_container_width=True)

# time.sleep(1 - st.session_state.time_slider)
if update_interval > 0:
    time.sleep(update_interval)

# SINGLE-FRAME DRAW WHEN PAUSED
elif st.session_state.running == 2:
    fig1 = go.Figure()

    # Left y-axis: Tank levels
    fig1.add_trace(go.Scatter(
        x=st.session_state.time_list,
        y=1 - np.array(st.session_state.h_list),
        name="P Phosphagen",
        yaxis='y1'
    ))
    fig1.add_trace(go.Scatter(
        x=st.session_state.time_list,
        y=1 - np.array(st.session_state.l_list),
        name="G Glycolytic",
        yaxis='y1'
    ))

    # Right y-axis: Lactate
    fig1.add_trace(go.Scatter(
        x=st.session_state.time_list,
        y=st.session_state.lacb_list,
        name="lacb Blood Lactate",
        yaxis='y2'
    ))
    fig1.add_trace(go.Scatter(
        x=st.session_state.time_list,
        y=st.session_state.lacm_list,
        name="lacm Muscle Lactate",
        yaxis='y2'
    ))

```

```
))
```

```
# Layout with dual y-axes
```

```
fig1.update_layout(  
    title="Tanks & Lactate",  
    xaxis_title="Time (s)",  
    yaxis=dict(title="Tank Fill Level", side='left'),  
    yaxis2=dict(title="[Lactate] (mmol/L)", overlaying='y', side='right'),  
    legend=dict(x=0, y=1.1, orientation="h")  
)
```

```
tank_container.plotly_chart(fig1, use_container_width=True)
```

```
fig2 = go.Figure()
```

```
# Left y-axis: Power outputs
```

```
fig2.add_trace(go.Scatter(  
    x=st.session_state.time_list,  
    y=st.session_state.pglc_list,  
    name="P Glyco",  
    yaxis='y1'  
))
```

```
fig2.add_trace(go.Scatter(  
    x=st.session_state.time_list,  
    y=st.session_state.pvo2_list,  
    name="Aerobic (O2)",  
    yaxis='y1'  
))
```

```
fig2.add_trace(go.Scatter(  
    x=st.session_state.time_list,  
    y=st.session_state.pmec_list,  
    name="Output P",  
    yaxis='y1'  
))
```

```
fig2.add_trace(go.Scatter(  
    x=st.session_state.time_list,  
    y=st.session_state.pcpwr_list,  
    name="PCr Power",  
))
```

```
    yaxis='y1'  
))
```

```
fig2.update_layout(  
    title="Power Breakdown & Metabolic Rates",  
    xaxis_title="Time (s)",  
    yaxis=dict(title="Power Output (W)", side="left"),  
    yaxis2=dict(title="Metabolic Rate (J/s)", overlaying="y", side="right"),  
    legend=dict(x=0, y=1.1, orientation="h")  
)
```

```
power_container.plotly_chart(fig2, use_container_width=True)
```

```
fig3 = go.Figure()
```

```
# Right y-axis: MRacc and MRf
```

```
fig3.add_trace(go.Scatter(  
    x=st.session_state.time_list,  
    y=st.session_state.MRacc_list,  
    name="MRacc - accum. metabolites",  
    yaxis='y1',  
    line=dict(dash='dot')  
))
```

```
fig3.add_trace(go.Scatter(  
    x=st.session_state.time_list,  
    y=st.session_state.MRf_list,  
    name="MRf - functional",  
    yaxis='y1',  
    line=dict(dash='dash')  
))
```

```
fig3.add_trace(go.Scatter(  
    x=st.session_state.time_list,  
    y=st.session_state.Pphys_list,  
    name="Pphys Boillet",  
    yaxis='y1',  
    line=dict(dash='dash')  
))
```

```
fig3.update_layout(  
    title="Metabolic Supply/Demand Rates",  
    xaxis_title="Time (s)",  
    yaxis=dict(title="Power Output (W)", side="left"),  
    yaxis2=dict(title="Metabolic Rate (J/s)", overlaying="y", side="right"),  
    legend=dict(x=0, y=1.1, orientation="h")  
)
```

```
metabolic_container.plotly_chart(fig3, use_container_width=True)
```