VIRTUAL ENGINE TEST BENCH v1.1 FRAMEWORK

Contents

[1) Introduction 1](#_Toc211346257)

[1.1) Motivation 1](#_Toc211346258)

[2) Framework Overview 2](#_Toc211346259)

[2.1) Main Script 2](#_Toc211346260)

[2.2) Engine Database 2](#_Toc211346261)

[2.3) Fuel Database 2](#_Toc211346262)

[2.4) Engine Module 3](#_Toc211346263)

[2.5) Calibration Module 3](#_Toc211346264)

[2.6) Testing Module 3](#_Toc211346265)

[2.7) Reporting Module 3](#_Toc211346266)

[3) Validation and Results 3](#_Toc211346267)

[4) Discussion and Future Work 3](#_Toc211346268)

[5) Version Updates 3](#_Toc211346269)

# Introduction

The Virtual Engine Test Bench (VETB) is a Python-based simulation framework designed to replicate the workflow of an internal-combustion-engine (ICE) dynamometer test environment. Its goal is to bridge the gap between simplified theoretical models and the experimental calibration process typically conducted on an engine test cell.

Rather than acting as a black-box performance simulator, the VETB is built as a physically bounded, modular system, where every variable has a clear thermodynamic origin and can be traced through the same reasoning used by a calibration engineer.

The framework emulates the standard steps of a dyno test: selecting operating points (engine speed and load), defining air and fuel conditions, running virtual cycles, and recording performance outputs such as torque, power, IMEP, BMEP, and emissions.

This allows the user to analyse how changes in engine geometry, combustion parameters, or fuel properties affect overall performance — without the need for costly or time-consuming physical testing.

## Motivation

Modern engine development increasingly relies on **model-based calibration** and **virtual testing** to reduce experimental effort and accelerate iteration cycles. However, most existing tools either sacrifice interpretability (black-box ML approaches) or demand high computational cost (3-D CFD and detailed kinetics). The VETB sits deliberately in between these two extremes: it prioritizes speed, clarity, and tunability while preserving the physical essence of combustion, gas exchange, and mechanical losses.

At the same time, the architecture has been designed to **accommodate hybrid modelling approaches**, where **data-driven or machine-learning layers** can be trained to correct or enhance physically derived quantities such as volumetric efficiency, flame speed, or heat-release rate. This mirrors the philosophy of projects like **DT-HATS**, which combine physics-based digital twins with data-driven surrogate models to achieve both accuracy and generality.  
By exposing clean interfaces between modules, the VETB can later be expanded into a **grey-box or black-box-augmented digital twin**, capable of learning from test data while maintaining a physically interpretable core.

The long-term vision is therefore to evolve this framework into a **virtual calibration environment** — a digital twin capable of reproducing engine behaviour across fuels, operating conditions, and even degradation states.  
In this sense, the VETB is not merely a research tool but a **learning and exploration platform**: it teaches the logic behind test-bench decisions, the influence of fuel chemistry on burn phasing, and the thermodynamic trade-offs that define modern powertrain design.

# Framework Overview

### 2.1) Main Script

### 2.2) Engine Database

### 2.3) Fuel Database

The fuel database has been expanded from v1.0, solely gasoline, and restructured in v1.1 to enable multi-fuel operation and compositional blending. It now acts as a centralized physical property library for all fuels used within the combustion model, including gasoline, alcohols, hydrogen, ammonia, methane and selected mixtures. Each entry defines the essential thermochemical parameters required by both the laminar flame-speed and energy-release sub-models.

All fuels are stored as instances of the *FuelSpec* dataclass. Each object represents a single fuel or blend, containing its chemical, thermophysical and combustion-specific properties. Contained properties are:

* LHV (MJ / kg): lower heating value of the fuel (energy released per unit mass of fuel burned).
* O2required / kg: mass of oxygen required per kilogram of fuel for stoichiometric combustion.
* AFRstoich: stoichiometric air–fuel ratio derived from oxygen demand and air composition.
* MW (g / mol): molecular weight.
* Ρ(kg/m3): gas density at 1 atm and 20 °C (mainly used for comparison).
* flam\_limits\_volpct – lower and upper flammability limits (volumetric %).
* SL(300K)(m / s): reference laminar flame speed at 300 K and 1 atm.
* Tautoignition(K): autoignition temperature.
* γ (T): optional temperature-dependent function for the unburned-gas specific-heat ratio (γ).

| **Fuel** | **LHV (MJ/kg)** | **AFRₛₜₒᵢcʰ** | **S\_L (m/s)** | **Tₐᵤₜₒ (K)** | **Notes** |
| --- | --- | --- | --- | --- | --- |
| Gasoline | 43.0 | 14.7 | 0.37 | 803 | baseline reference fuel |
| Ethanol | 26.8 | 9.0 | 0.45 | 690 | oxygenated; faster laminar flame |
| Methanol | 19.9 | 6.4 | 0.48 | 760 | high latent heat, slower burn rate per energy |
| Hydrogen | 120.0 | 34.3 | 2.5 | 858 | high diffusivity, wide flammability limits |
| Ammonia | 18.6 | 6.1 | 0.07 | 924 | low reactivity, slow flame propagation |
| Methane | 50.0 | 17.2 | 0.38 | 813 | gaseous benchmark fuel |

#### 2.3.1) Blending Framework

To enable multi-fuel and transitional operation, three blending utilities were implemented.

a) Gasoline–Ethanol and Gasoline–Methanol Blends

* Input: volume fraction of oxygenate (E% or M%).
* Converts volume % → mass % using liquid densities (0.74 kg/L for gasoline, 0.789 / 0.792 kg/L for ethanol/methanol).
* Blends key properties linearly by mass:
  + LHV, O₂ requirement, AFRₛₜₒᵢcʰ, laminar flame speed, molecular weight, autoignition temperature.
* Returns a new FuelSpec instance (e.g. *E10 (Gasoline–Ethanol)*).

This allows continuous control over fuel composition for studying the effect of increasing alcohol content on combustion efficiency or knock behaviour.

b) H₂–NH₃ Blends

Hydrogen–ammonia mixtures are defined by either mass fraction or energy fraction of H₂.

### 2.4) Engine Module

#### 2.4.1) Combustion

#### 2.4.2) Emissions

### 2.5) Calibration Module

### 2.6) Testing Module

### 2.7) Reporting Module

# 3) Validation and Results

# 4) Discussion and Future Work

# 5) Version Updates