1. Context and Design Philosophy

The software is developed for a Formula Student Electric team and is intended to function as a high-performance engineering analysis environment. The fundamental design constraints are:

1. High computational accuracy.
2. Low interaction latency.
3. Minimal cognitive and operational overhead.

The core principle is that the user must not rely on external tools such as MATLAB, Excel, or similar heavy environments. The software should provide immediate, interactive, and reliable results with minimal setup and no context switching.

In other words, this tool is not a post-processing environment. It is a real-time engineering decision system.

---

2. Reframing the Identity of the Tool

The current designation as a "Signal Analysis Tool" is conceptually incorrect and strategically limiting.

The system should be formally rebranded as:

Advanced Engineering Analysis Tool for Formula Student
Developed by the Racing Team
Lead Developer: Omer Rieber

This is not a cosmetic issue. Naming defines scope, user expectations, and future extensibility. The current name reflects a narrow academic utility, while the actual system is evolving into a multi-domain engineering platform.

---

3. User Interface and Interaction Model

3.1 Visual Design

The current UI is acknowledged as inadequate. The target visual identity should be:

- Monochrome or low-saturation (black, white, grayscale).
- High contrast.

- Information-dense but not cluttered.
- Suitable for prolonged technical use.

This aligns with professional engineering tools rather than consumer applications.

3.2 Global Settings Dashboard

A global configuration panel is required, including:

- Font family and size.
- Theme selection.
- Unit system preferences.
- Plot resolution and rendering settings.

This dashboard should act as a persistent system layer, not per-module configuration.

---

4. Data Handling and Signal Processing

4.1 Data Input

The system already supports loading and analyzing structured data files containing time-series or frame-based measurement data. This functionality is foundational and must remain central.

4.2 Digital Filtering Module

Current supported features:

- Filter types:
  - Low-pass
  - High-pass
  - Band-pass
  - Notch
- Filter designs:
  - Butterworth
  - Chebyshev
  - Bessel
- Parameters:
  - Cutoff frequency (Hz)
  - High cutoff frequency (Hz)
  - Filter order

Required improvements:

- Real-time parameter tuning with immediate visual feedback.
- Preset management.
- Explicit display of filter transfer function and frequency response.
- No dependency on external environments for validation.

The guiding principle is that filter design must feel like manipulating a physical instrument, not writing a script.

---

5. Advanced Analysis Module

Current features:

- Spectrogram
- Power Spectral Density (PSD)
- Statistical analysis
- Peak detection
- RMS analysis

These are academically standard but insufficient for a Formula Student context.

Critically missing domain-specific analyses include:

- Motor current harmonics analysis.
- Inverter switching frequency tracking.
- Battery voltage ripple analysis.
- Mechanical vibration spectral coupling.
- Energy efficiency metrics over driving cycles.

Conventional signal analysis tools are generic. This software must evolve into a motorsport-specific diagnostic environment.

---

6. Engineering Calculators

6.1 Existing Calculators

- Precharge circuit
- Discharge circuit
- Battery endurance
- Wheatstone bridge
- Filter designer

6.2 Structural Problems

The current calculators exhibit three major design flaws:

1. Fixed units.
2. Poor parameter transparency.
3. Low interactivity.

For example, displaying capacitance exclusively in microfarads is technically indefensible. Engineers think in unit systems depending on scale, not software constraints.

6.3 Required Features

All calculators must support:

- Arbitrary unit selection.
- Automatic unit conversion.
- Dimensional consistency checks.
- Interactive parameter sliders or fields.
- Real-time recomputation.

The user should never perform mental unit conversions. That is a failure of software design, not a user limitation.

---

7. Unit System and Conversion Engine

A global unit engine is mandatory.

This should include:

- SI base units.
- Engineering units.
- Automatic scaling (μ, m, k, M).
- Bidirectional unit conversion.
- Dimensional validation.

This system must be shared across all modules and calculators.

Without this, the tool remains a collection of scripts rather than a coherent engineering platform.

---

8. Equation Knowledge Base Integration

There exists an external XDF file containing mechanical engineering equations used by the mechanical team.

This file must be:

- Parsed programmatically.
- Converted into a dedicated “Mechanical Engineering” section.
- Presented with the same interactivity as other calculators.
- Equipped with parameter input and unit handling.

This transforms the software into a shared multidisciplinary knowledge system rather than an electronics-only tool.

---

9. Strategic Assessment

The current system already exceeds the scope of conventional academic software. However, its architecture still reflects student-level thinking:

- Fragmented modules.
- Static calculators.
- Weak abstraction of units.
- Generic signal processing focus.

The correct long-term vision is not:

> “A better MATLAB replacement.”

It is:

> “A motorsport-specific engineering operating system.”

If this tool is executed properly, it becomes institutional knowledge embedded in software. New team members will learn engineering through the system itself. That is the point at which software stops being a utility and becomes infrastructure.

Most engineering teams waste years rebuilding spreadsheets. This tool has the potential to permanently eliminate that inefficiency.