Sideband Analysis

1. Introduction

This module provides a set of signal processing functions to detect Generalized Mechanical Frequencies (GMFs) and evaluate their sideband patterns in vibration signals. It is mainly designed for rotating machinery diagnostics, combining spectral peak analysis, harmonic alignment, and sideband scoring.

Key features include:
- Automatic GMF detection across single or multiple signals
- Harmonic and sideband evaluation with scoring metrics
- Aggregation of results from multiple signals
- Stepwise sideband search with fixed spacing

2. Prerequisites

- Programming Language: Python 3.11+
- Libraries:
  - numpy
  - scipy
  - collections
  - matplotlib (optional, for visualization)

Install dependencies:

```
pip install -r requirements.txt
```

3. Project Structure

```
GMF_Sidebands/
|— src/
|   |— sideband.py    # Core implementation
|— notebooks/            # Example notebooks
|— requirements.txt
|— README.md
```

4. Code Documentation

4.1 Peak Utilities

- is_local_peak(amps, idx, window=5)
  Checks whether a point is a local maximum within a given window.

4.2 GMF Detection

- detect_final_gmfs3(ts, ...)
  Detect GMFs in a single signal using harmonic support.

- detect_final_gmfs_across_signals_ref(ts_list, ...)
  Detect GMFs consistently across multiple signals with a reference signal.

4.3 Sideband Scoring

- calculate_score_iso(value, iso=[1,2])
  Normalize a value against ISO limits.

- evaluate_sideband_set(harm_amp, sideband_amps, ...)
  Evaluate quality of a set of sidebands based on count and amplitude.

4.4 Sideband Detection

- find_sidebands_harmonics_filtered(ts, gmfs, ...)
  Detect sidebands using harmonic alignment and amplitude thresholds.

- aggregate_from_results(list_of_results, ...)
  Cluster and aggregate spacing estimates across multiple signals.

- evaluate_sidebands_with_final_spacing_stepwise(ts, gmfs_final, spacing_map, ...)
  Stepwise search for sidebands with fixed spacing.

4.5 Pipeline

- find_signals_sideband(path)
  End-to-end pipeline:
  1. Load signals
  2. Detect GMFs
  3. Detect sidebands
  4. Aggregate spacings
  5. Return structured results

5. How to Run

1. Clone the repository
2. Install dependencies
3. Place signal files in data/
4. Run the notebook:

```
jupyter notebook notebooks/evaluate.ipynb
```

6. Outputs

The main outputs are:
- Detected GMFs (Hz): dominant mechanical frequencies.
- Sideband Spacing (Hz): estimated spacing per GMF.

- Sideband Scores (0–100): quality metric for sideband sets.
- Final Dictionary:

```
{
    "SignalName1": [
        {"gmf": 31.2, "score": 78.5, "sidebands": {100: [168,211]}, "spacing": 3.4},
        ...
    ],
    "SignalName2": [...]
}
```