Raw Dataset Specification: Pressure & Oscilloscope Streams with File-Level Midpoint Alignment and Configurable Uncertainty

1 Overview: Two Unsynchronized Measurements

The data comprise two streams recorded by different, unsynchronized devices:

- **Pressure stream** (*P-stream*): per-record *global timestamp* followed by a *voltage triple* $(v^{(1)}, v^{(2)}, v^{(3)})$. Each column maps linearly to pressure, but the canonical scalar is *only channel 3*.
- **Oscilloscope stream** (*O-stream*): per *file*, a time series with N_F samples, uniform sampling interval δt_F, and amplitudes (channels) collected as $\boldsymbol{v}_{F,n}$.

Because the devices are not synchronized, each O-stream file F receives a single scalar pressure label by aligning its *file midpoint time* to the *nearest* P-stream global timestamp.

2 Pressure Stream (P-stream)

2.1 Per-Record Structure

Each record consists of:

1. A global *timestamp line* of the form

$$\texttt{M}\underbrace{\texttt{MM}}_{\text{month}}\texttt{-D}\underbrace{\texttt{DD}}_{\text{day}}\texttt{-H}\underbrace{\texttt{HH}}_{\text{hour}}\texttt{-M}\underbrace{\texttt{mm}}_{\text{minute}}\texttt{-S}\underbrace{\texttt{SS}}_{\text{second}}\texttt{-U.}\underbrace{\texttt{uuu}}_{\text{subsecond}},$$

   e.g., `M08-D19-H16-M24-S03-U.128`. Parsing $\tau(\cdot)$ yields absolute time $T^P \in \mathbb{R}$ (seconds).

2. Immediately followed by a *voltage triple*:

$$v^{(1)} \quad v^{(2)} \quad v^{(3)} \qquad \text{(in volts)}.$$

For $k \in \{1, 2, 3\}$, linear calibration is

$$p^{(k)} = \alpha_k \, v^{(k)} + \beta_k.$$

The canonical scalar pressure is *only channel 3*:

$$p = p^{(3)} = \alpha_3 v^{(3)} + \beta_3.$$

We denote the P-stream as $\mathcal{P} = \{(T_m^P, p_m)\}_{m=1}^M$ with $p_m = \alpha_3 v_m^{(3)} + \beta_3$.

3 Oscilloscope Stream (O-stream)

3.1 Per-File Structure and Timing

Each oscilloscope file F carries a session ID and an embedded global timestamp, e.g.,

`itid0notrig3M08-D19-H16-M24-S43-U.788.csv`.

We parse $F \equiv (\texttt{sid}, \texttt{stamp}, \texttt{ext})$ and set $T_F^{\text{start}} = \tau(\texttt{stamp})$. Within the file, we have a uniform series of length N_F with sampling interval δt_F:

$$t_{F,n} = T_F^{\text{start}} + n\,\delta t_F, \qquad n = 0, 1, \ldots, N_F - 1.$$

Let $D_F = N_F \delta t_F$ be the duration and

$$T_F^{\text{mid}} \;=\; T_F^{\text{start}} + \tfrac{1}{2} D_F. \tag{1}$$

4 File-Level Mapping to Pressure (Nearest Timestamp)

For each O-stream file F, map its midpoint time to the nearest P-stream timestamp:

$$T^{P*}(F) \in \arg \min_{T_m^P \in \{T_1^P, \ldots, T_M^P\}} \left| T_F^{\text{mid}} - T_m^P \right|. \tag{2}$$

The associated scalar pressure label is $p^*(F) = p$ evaluated at $T^{P*}(F)$.

Alignment error and acceptance. Define the file-level alignment error

$$E_{\text{align}}(F) \;=\; \left| T_F^{\text{start}} + \tfrac{1}{2} N_F \delta t_F \;-\; T^{P*}(F) \right|. \tag{3}$$

Mapping is *accepted* if $E_{\text{align}}(F) \le O_{\max}$, where $O_{\max} > 0$ is provided in the external configuration file.

5 Pressure Error from Mapping Mismatch

Let $p(t)$ denote the scalar pressure trajectory from channel 3. We estimate its local derivative at the matched time using a configurable window size W (from the configuration file) around $T^{P*}(F)$.

Derivative estimator (configurable window W). Define the slope estimator at $T^{P*}(F)$ as

$$\widehat{\dot{p}}_W\big(T^{P*}(F)\big) \;\equiv\; \mathrm{DerivEst}\big(\mathcal{P};\, T^{P*}(F),\, W\big)\,, \tag{4}$$

where DerivEst is a user-selected method (e.g., central difference, local linear regression) over the window of temporal width W centered at $T^{P*}(F)$.

Uncertainty multiplier. Let $\kappa \ge 0$ be a user-provided multiplier that scales the bound conservatively.

Pressure uncertainty bound. The induced pressure inaccuracy for file F due to alignment error (3) is

$$|\Delta P(F)| \;\le\; \kappa \left| \widehat{\dot{p}}_W\big(T^{P*}(F)\big) \right| \cdot E_{\text{align}}(F). \tag{5}$$

Equation (4) ("Eq. 4") explicitly states that the derivative is estimated using the input window W. *Reference macro:* Eq. (4).

6 Relational Materialization for ML

For downstream ML, a single tall table is sufficient; a normalized mapping is optional.

Option A: Single Tall Table

$$\texttt{Signals}\Big(\underbrace{\texttt{sid}}_{\text{session/run ID}},\ \underbrace{\texttt{file\_stamp}}_{T_F^{\text{start}}},\ \underbrace{\texttt{idx}}_{n},\ \underbrace{\texttt{t\_abs}}_{t_{F,n}},\ \boldsymbol{v}\ \text{(channels)},\ \underbrace{\texttt{N}}_{N_F},\ \underbrace{\texttt{delta\_t}}_{\delta t_F},\ \underbrace{\texttt{duration}}_{D_F},\ \underbrace{\texttt{t\_mid}}_{T_F^{\text{mid}}},\ \underbrace{\texttt{p\_time}}_{T^{P*}(F)},\ \underbrace{\texttt{p}}_{p^*(F)=p^{(3)}},\ \texttt{O\_}$$

Primary key: $(\texttt{sid}, \texttt{file\_stamp}, \texttt{idx})$.

Option B: Normalized Mapping

$$\texttt{OscFiles}(\texttt{sid},\ \texttt{file\_stamp},\ N,\ \delta t,\ \texttt{duration},\ \texttt{t\_mid})$$

$$\texttt{File2PressureMap}(\texttt{sid},\ \texttt{file\_stamp},\ \texttt{p\_time} = T^{P*},\ \texttt{p} = p^*,\ \texttt{O\_max},\ \texttt{W},\ \texttt{kappa},\ \texttt{E\_align},\ \texttt{dP\_bound})$$

$$\texttt{Signals}(\texttt{sid},\ \texttt{file\_stamp},\ \texttt{idx},\ \texttt{t\_abs},\ \boldsymbol{v})$$

7 External Configuration File: `dataset_config.yml`

All missing inputs and policy choices are centralized in a YAML file that ships with the dataset/repo.

7.1 YAML Schema (math semantics)

Key	Type	Meaning / Math Symbol
`calibration.alpha`	list[float] (len 3)	Per-channel slopes $(\alpha_1, \alpha_2, \alpha_3)$.
`calibration.beta`	list[float] (len 3)	Per-channel intercepts $(\beta_1, \beta_2, \beta_3)$.
`pressure.scalar_channel`	int	Channel index used for scalar pressure (must be 3)
`units.pressure`	str	Pressure unit (e.g., `mmHg`).
`units.voltage`	str	Voltage unit (e.g., `V`).
`timestamp.format`	str	Stamp grammar/pattern for parser $\tau(\cdot)$.
`timestamp.subsec_unit`	str	Subsecond unit for `U.` field (e.g., `ms` or `us`).
`timestamp.timezone`	str	Time zone assumption for stamps.
`mapping.tie_breaker`	enum	Nearest-timestamp tie policy (`earliest`|`latest`).
`mapping.O_max`	float (s)	Max acceptable alignment error $O_{\max}$ (seconds).
`derivative.method`	enum	DerivEst type (`central_diff`|`local_linear`|`savgo`
`derivative.W`	float (s)	Window width W used in Eq. (4).
`uncertainty.kappa`	float	Multiplier κ used in Eq. (5).
`quality.min_records_in_W`	int	Minimum P-stream points inside W to accept slo estimate.
`quality.reject_if_Ealign_gt_Omax`	bool	If true, discard file when $E_{\text{align}} > O_{\max}$.
`defaults.year_fallback`	int	Year to use if missing on lines (overridden by f name).

7.2 YAML Example (to include in repo as `dataset_config.yml`)

```
calibration:
  alpha: [a1, a2, a3]        # slopes for channels 1..3
  beta:  [b1, b2, b3]        # intercepts for channels 1..3
pressure:
  scalar_channel: 3          # must be 3
```

```
units:
  pressure: "mmHg"
  voltage: "V"
timestamp:
  format: "M%02d-D%02d-H%02d-M%02d-S%02d-U.%03d"
  subsec_unit: "ms"           # or "us"
  timezone: "UTC"
mapping:
  tie_breaker: "earliest"     # or "latest"
  O_max: 0.250                # seconds (example)
derivative:
  method: "central_diff"      # or "local_linear", "savgol"
  W: 2.0                      # seconds; window centered at T^{P*}(F)
uncertainty:
  kappa: 1.0                  # conservative multiplier
quality:
  min_records_in_W: 3         # require >=3 P-stream points in window
  reject_if_Ealign_gt_Omax: true
defaults:
  year_fallback: 2025
```


Semantics.

- Scalar pressure is $p = \alpha_3 v^{(3)} + \beta_3$; channels 1,2 may be retained as auxiliary.
- Alignment follows Eq. (2); error via Eq. (3); acceptance threshold $O_{\max}$ from the YAML.
- The slope estimator in Eq. (4) uses the window W and method specified in `derivative`.
- The uncertainty bound in Eq. (5) uses κ from `uncertainty`.
- If `reject_if_Ealign_gt_Omax` is true and $E_{\text{align}}(F) > O_{\max}$, the file F is excluded or flagged.

8 Integrity Constraints & Sanity Checks

1. **Units:** voltages in V; pressures in mmHg; (α_k, β_k) must be unit-consistent.
2. **Scalar channel:** the only scalar pressure used downstream is $p^{(3)}$.
3. **O-stream timing:** $t_{F,n} = T_F^{\text{start}} + n\delta t_F$; $D_F = N_F \delta t_F$; T_F^{mid} as in Eq. (1).
4. **Mapping rule:** Eq. (2); error Eq. (3); accept iff $E_{\text{align}}(F) \le O_{\max}$.
5. **Derivative window:** Eq. (4) uses window W from the YAML config.
6. **Uncertainty bound:** Eq. (5) with multiplier κ; store `dP_bound`.
7. **Determinism:** fixed tie-breaking policy for nearest timestamp (YAML `mapping.tie_breaker`).

9 Minimal Examples

Pressure record (P-stream)

```
M08-D19-H16-M24-S03-U.128
2.427  1.092  6.266
```

Parsed at $T^P = \tau$(“M08-D19-H16-M24-S03-U.128”) with $p = \alpha_3 \cdot 6.266 + \beta_3$.

Oscilloscope file (O-stream)

`itid0notrig3M08-D19-H16-M24-S43-U.788.csv`

Set $T_F^{\text{start}} = \tau$(“M08-D19-H16-M24-S43-U.788”), compute $D_F = N_F \delta t_F$, T_F^{mid} via Eq. (1), find $T^{P*}(F)$ by Eq. (2), $E_{\text{align}}(F)$ by Eq. (3), accept if $\le O_{\max}$, and set $p^*(F) = p$ at $T^{P*}(F)$ with uncertainty bound Eq. (5).

10 Adapters: Mapping $(t, v, \delta t)$ to ML-Ready Vectors

Each adapter $\mathcal{A}$ maps the oscilloscope file F with time series $\{(t_{F,n}, v_{F,n})\}_{n=0}^{N_F-1}$ to a fixed-length vector

$$\boldsymbol{x}_F \in \mathbb{R}^L \quad \text{(independent of } N_F \text{ and invariant to time shifts).}$$

Adapters may depend on an estimated fundamental period T_0 (or $f_0 = 1/T_0$) and configuration parameters supplied in `dataset_config.yml`.

10.1 Adapter API

Given F, configuration Γ, and (optionally) a prior T_0,

$$\boldsymbol{x}_F,\ \mathcal{Q}_F \ = \ \mathcal{A}(F; \Gamma, T_0)\,,$$

where $\mathcal{Q}_F$ are quality diagnostics (e.g., cycles used, SNR gain, detection confidence). Each adapter must:

1. produce a fixed length L (set in Γ),
2. be (approximately) invariant to global time shift of $v_{F,\cdot}$,
3. remain comparable across files with different pulse counts (N_F varies).

10.2 Cycle/Period Detection Options (shared prelude)

Before any adapter, we may estimate T_0 using one (or combine several) of:

- **Autocorrelation peak:** $\hat{T}_0 \in \arg\max_{\tau \in \mathcal{T}} \sum_n v_{F,n}\, v_{F,n+\tau}$.
- **Cepstrum:** $\hat{T}_0$ at the dominant quefrency peak of $\mathcal{C}(q) = \mathcal{F}^{-1}\{\log|\mathcal{F}\{v\}|^2\}$.
- **Spectral peak:** $\hat{f}_0 = $ arg max peak in $|\mathcal{F}\{v\}|$; $T_0 = 1/\hat{f}_0$.
- **Fiducial peaks:** robust peak picking with amplitude threshold + refractory period near $\hat{T}_0$.

Quality gates (min peak ratio, min SNR) guard against spurious periodicity.

10.3 Adapter A: Phase-Bin Cycle-Synchronous Averaging (PB-CSA)

Idea: Fold samples by *phase* then average across cycles (super-res in phase).

1. Estimate $\hat{T}_0$; define phase $\theta_n = 2\pi\,(t_{F,n} \bmod \hat{T}_0)/\hat{T}_0$.
2. Choose phase bins $\Theta_k = [2\pi(k-1)/K,\, 2\pi k/K)$, $k = 1..K$.
3. Aggregate $x_k = \mathrm{Aggr}\{\, v_{F,n} :\ \theta_n \in \Theta_k \,\}$ using *mean/median/trimmed-mean*.
4. **Shift invariance**: rotate so the maximal bin is centered (circular shift) or anchor to a fiducial (e.g., rising-edge phase).

Output: $\boldsymbol{x}_F \in \mathbb{R}^K$ (set $L = K$). *Notes*: Large K gives phase super-resolution by multi-cycle dithering. Require min counts per bin; adapt K if needed. Cycle-synchronous averaging is a classical approach for periodic signal enhancement [1].

10.4 Adapter B: Peak-Locked Segmentation & Time Normalization (PLSTN)

Idea: Cut windows around detected peaks, normalize each cycle to M samples, then robust-average.

1. Detect peak indices $\{n_j\}$ with refractory $\approx \hat{T}_0$.
2. For each cycle window $W_j = [n_j - w_-,\, n_j + w_+]$, resample to M points via bandlimited (sinc) or cubic interpolation.
3. Robust-average across cycles (median / Huber) to get $\boldsymbol{x}_F \in \mathbb{R}^M$.

Shift invariance: windows are peak-centered. *Pros*: faithful time-domain shape. *Cons*: sensitive to missed/false peaks. *Notes*: Resampling each cycle to a fixed length can compensate for slight period variations (e.g., due to speed fluctuations) [2].

10.5 Adapter C: Harmonic Magnitude Vector (HMV)

Idea: Represent the signal by magnitudes at the first H harmonics of $\hat{f}_0$.

$$X(k) \;=\; \left| \sum_n v_{F,n} e^{-i2\pi k \hat{f}_0 t_{F,n}} \right|, \quad k = 1..H, \quad \boldsymbol{x}_F = [X(1), \ldots, X(H)].$$

Shift invariance: magnitudes remove phase. Optionally normalize by DC or ℓ_2 norm. *Pros*: compact, robust to misalignment. *Cons*: loses waveform phase/shape details. *Notes*: Using a limited set of harmonic amplitudes is essentially a truncated Fourier series representation of the periodic signal [3].

10.6 Adapter D: Cepstral Envelope Coefficients (CEC)

Idea: Use low-quefrency cepstral coefficients to capture periodic envelope.

$$\boldsymbol{x}_F = \big[c_1, \ldots, c_Q\big], \quad \text{where } c_q \text{ are first } Q \text{ coefficients of } \mathcal{C}(q).$$

Shift invariance: by construction (depends on magnitude spectrum). *Pros*: noise-robust; detects periodicity. *Cons*: less interpretable in time domain. *Notes*: Cepstral analysis was introduced for echo and pitch detection in acoustics [4], and low-quefrency coefficients (similar to MFCCs) are widely used to represent spectral envelope.

10.7 Adapter E: DTW-Template Average (DTW-TA)

Idea: Align each detected cycle to a reference waveform via constrained DTW, resample to M, robust-average.

1. Build a provisional template (median of a few cycles or from PB-CSA).
2. Constrained DTW (Sakoe–Chiba band) to warp each cycle to template length M.
3. Robust-average the aligned cycles $\Rightarrow \boldsymbol{x}_F \in \mathbb{R}^M$.

Shift invariance: implicit via alignment. *Pros*: handles cycle length drift/shape variability. *Cons*: heavier compute; risk of over-warping noise. *Notes*: Dynamic time warping is a classic method for sequence alignment (originating in speech recognition) [5], and can be leveraged to compute an average waveform across cycles.

10.8 Adapter F: Matched-Template Projection (MTP)

Idea: If a known canonical periodic pattern $z(t)$ exists (identical across files), estimate shift/scale and project onto z.

$$(\hat{a}, \hat{\tau}) = \arg\min_{a,\tau} \sum_n \big(v_{F,n} - a\, z(t_{F,n} - \tau)\big)^2,$$
$$\boldsymbol{x}_F = \big[\, \hat{a}, \quad \underbrace{\langle v_F, z_k \rangle}_{\text{optional multi-basis}} \quad , \ \dots \big].$$

Shift invariance: encode only $\hat{a}$ or coefficients in the pattern basis; drop $\hat{\tau}$ for invariance. *Pros*: very compact; high SNR when template is accurate. *Cons*: requires a reliable $z(t)$. *Notes*: This approach parallels the matched filter concept in signal processing [6], where a known template is used to detect and quantify a signal of interest.

10.9 Boosting Resolution and Robustness (shared tricks)

- **Interpolation**: prefer bandlimited/sinc (best fidelity), cubic as lighter fallback.
- **Robust aggregation**: median or trimmed-mean (e.g., 10%) to suppress outlier cycles.
- **Adaptive resolution**: choose K or M so that expected samples/bin $\geq c_{\min}$; otherwise reduce K or merge bins.
- **Normalization (optional)**: RMS or peak normalization per cycle when amplitude should not encode label; leave in absolute units when it should.
- **SNR gain**: cycle averaging boosts SNR by $\approx \sqrt{J_{\text{eff}}}$ where J_{eff} is the number of non-rejected cycles.

10.10 Choosing an Adapter (summary)

Adapter	Output	Shift Inv.	Pros	Cons
PB-CSA	K phase bins	via circular align	super-res phase; fast	needs good T_0
PLSTN	M samples	peak-centered	faithful waveform	peak sensitivity
HMV	H mags	by magnitude	compact; robust	loses phase/shape
CEC	Q coeffs	yes	noise-robust	less interpretable
DTW-TA	M samples	via warping	handles drift	compute heavy
MTP	few scalars	by design	very compact	needs template

10.11 Quality Diagnostics (recommended to store)

For each F, store: cycles detected/used (J, J_{eff}), detection SNR, min counts/bin (PB-CSA), rejected cycle ratio, interpolation method, and an instability flag (e.g., period variance over windows).

11 YAML Extensions: Adapter Selection & Hyperparameters

Key	Type	Meaning
`adapter.name`	enum	`pb_csa` | `plstn` | `hmv` | `cec` | `dtw_ta` | `mtp`.
`adapter.output_length`	int	L (i.e., K or M or H or Q).
`adapter.period_est.method`	enum	`autocorr` | `cepstrum` | `spectral` | `peaks` | `hybrid`.
`adapter.period_est.search_range`	[float,float] (s)	plausible T_0 bounds.
`adapter.pb_csa.K`	int	# phase bins (if `pb_csa`).
`adapter.pb_csa.aggregator`	enum	`mean` | `median` | `trimmed_mean`.
`adapter.pb_csa.min_bin_count`	int	minimum samples per phase bin.
`adapter.plstn.M`	int	resampled points per cycle (if `plstn` or `dtw_ta`).
`adapter.plstn.window`	[int,int]	w_-, w_+ samples around peak.
`adapter.plstn.interp`	enum	`sinc` | `cubic`.
`adapter.hmv.H`	int	# harmonics.
`adapter.cec.Q`	int	# low-quefrency coeffs.
`adapter.dtw_ta.band`	float	DTW Sakoe–Chiba band (as fraction of M).
`adapter.mtp.template_path`	str	path to canonical $z(t)$.
`adapter.normalization`	enum	`none` | `rms` | `peak`.
`adapter.reject_outlier_cycles`	bool	robust averaging on/off.
`adapter.trim_fraction`	float	for trimmed means (e.g., 0.1).
`adapter.seed`	int	deterministic tie-breaks and sampling.

Notes.

- Period/phase estimates may be re-used across adapters to save compute.
- When signal is multi-channel, apply adapters per channel or on a fused channel (e.g., principal component), then concatenate.
- Store diagnostics with each $\boldsymbol{x}_F$ to enable downstream quality-aware ML (e.g., weighting by J_{eff}).

12 Second-Layer Transform Adapters

In addition to the cycle-synchronous adapters (A–F), we define a second layer of classical transform-based adapters. These operate directly on a fixed-length time series (either raw or first-layer adapted) to produce a fixed-length vector in $\mathbb{R}^L$. Each transform is well-established in acoustic signal processing and designed to be invariant to global time shifts.

Notation (additive). Let M_{s} denote the input time-sample length provided to any second-layer transform; for MFCC we retain C_{MFCC} coefficients (typically 12–13).

12.1 Adapter G: Fourier Transform Spectrum (FTS)

Idea: Represent the signal by its frequency content. For a time series $v_n,\ n = 0, \ldots, M-1$, the discrete Fourier transform (DFT) is

$$X(k) = \sum_{n=0}^{M-1} v_n\, e^{-i2\pi kn/M}, \qquad k = 0, \ldots, M-1.$$

We take the magnitude spectrum $|X(k)|$ as features. **Shift invariance:** A circular shift adds only a phase factor, leaving $|X(k)|$ unchanged. **Output:** $\boldsymbol{x}_F = \big[|X(0)|, \ldots, |X(M/2)|\big] \in \mathbb{R}^{M/2+1}$ for real signals. **Pros:** Captures full spectral content, widely used in acoustics. **Cons:** High dimensionality; assumes stationarity over the segment. *Reference:* [3].

12.2 Adapter H: Hilbert Transform Envelope (HTE)

Idea: Form the analytic signal $a(t) = v(t) + i\mathcal{H}\{v(t)\}$ using the Hilbert transform $\mathcal{H}\{v\}$, and extract the amplitude envelope

$$e(t) = |a(t)|, \qquad \phi(t) = \arg(a(t)).$$

We use sampled $e(t)$ or its summary statistics as the feature vector. **Shift invariance:** The envelope is insensitive to carrier phase and global shifts. **Output:** $\boldsymbol{x}_F = [e_0, \ldots, e_{M-1}]$ or reduced summary. **Pros:** Captures amplitude modulation patterns critical in acoustics. **Cons:** Loses fine timing; may need subband envelopes. *Reference:* [7].

Shift-invariance enforcement (additive). Either (i) circularly rotate e so that $\arg\max_n e_n$ is at a fixed index (e.g., 0) before sampling, or (ii) compute the DCT of $\log(e)$ and keep its first Q_e coefficients as the feature. Both produce fixed-length, shift-invariant vectors.

12.3 Adapter I: Wavelet Coefficient Vector (WCV)

Idea: Apply the discrete wavelet transform (DWT) to obtain approximation coefficients A_J and detail coefficients D_j across scales $j = 1, \ldots, J$. Collect either the raw coefficients or subband energies:

$$E_j = \sum_n |D_j[n]|^2, \qquad \boldsymbol{x}_F = [E_1, \ldots, E_J, \|A_J\|_2^2].$$

Shift invariance: Standard decimated DWT is shift-variant; using stationary wavelet transform (SWT) or aggregated energies yields approximate invariance. **Output:** $\boldsymbol{x}_F \in \mathbb{R}^{J+1}$. **Pros:** Multi-resolution time-frequency analysis; robust to nonstationarity. **Cons:** Choice of wavelet family/levels impacts results. *Reference:* [8].

Shift-invariance note (additive). We use stationary (undecimated) wavelets or subband-energy pooling to achieve approximate shift invariance.

12.4 Adapter J: Mel-Frequency Cepstral Coefficients (MFCC)

Idea: Compute log-energy in mel-spaced frequency bands and apply discrete cosine transform (DCT). For B mel filters:

1. Compute band energies E_k, $k = 1..B$.
2. $Y_k = \ln E_k$.
3. Apply DCT: $c_m = \sum_{k=1}^{B} Y_k \cos\big[\frac{\pi m}{B}(k - 0.5)\big]$, $m = 1..M$.

Shift invariance: Based on spectrum magnitude only, hence invariant to global shifts. **Output:** $\boldsymbol{x}_F = [c_1, \ldots, c_M] \in \mathbb{R}^M$. **Pros:** Compact, perceptually motivated, proven in speech/audio. **Cons:** Discards pitch and fine harmonic detail. *Reference:* [9].

References

References

[1] S. Braun, *The Extraction of Periodic Waveforms by Time Domain Averaging*, Acustica **32**(2), 69–77 (1975).

[2] P. D. McFadden, *Interpolation techniques for time domain averaging of gear vibration*, Mechanical Systems and Signal Processing **3**(1), 87–97 (1989).

[3] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall (1989).

[4] A. M. Noll, *Cepstrum pitch determination*, J. Acoust. Soc. Am. **41**(2), 293–309 (1967).

[5] H. Sakoe and S. Chiba, *Dynamic programming algorithm optimization for spoken word recognition*, IEEE Trans. Acoust. Speech Signal Process. **26**(1), 43–49 (1978).

[6] G. L. Turin, *An introduction to matched filters*, IRE Trans. Information Theory **6**(3), 311–329 (1960).

[7] S. O. Sadjadi and J. H. L. Hansen, *Hilbert envelope based features for robust speaker identification under reverberant mismatched conditions*, Proc. IEEE ICASSP, 5448–5451 (2011).

[8] G. Tzanetakis, G. Essl, and P. Cook, *Audio analysis using the discrete wavelet transform*, Proc. WSES Int. Conf. Acoustics & Music Theory & Applications, Skiathos, Greece (2001).

[9] S. B. Davis and P. Mermelstein, *Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*, IEEE Trans. Acoust. Speech Signal Process. **28**(4), 357–366 (1980).