Spectral–Resonance–Cognitive System (SRC): A Multilayered Framework for Music Perception

Amaç
Ev
amace@bu.edu

Chapter I — What is SRC^9?

1.1 Definition and Rationale

SRC^9, or the **Spectral–Resonance–Cognitive** system, is an integrated, multi-layered computational framework that models how music transforms from raw sound into perceived structure and cognitive meaning. It is composed of three tightly coupled modules:

- **S^3 – Spectral Sound Space**: A high-resolution spectral analysis engine that extracts the fundamental acoustic content of music, including partials, harmonics, microtonal deviations, and time–frequency–amplitude relations.
- **R^3 – Resonance-Based Relational Reasoning**: A harmonic reasoning engine that interprets spectral data through scalar field theory, resonance topology, and psychoacoustic principles to model musical structure without symbolic assumptions.
- **C^3 – Cognitive Consonance Circuit**: A neurophysiologically grounded model of perceptual resonance and emotional impact, structured into nine cognitive units derived from EEG/fMRI literature and organized into measurable, time-varying neural signatures.

The superscript 9 (^{9}) denotes the system's expansion into **nine cognitive dimensions**, each modeled as a unique *Unit* in the C^3 architecture, and accessible from the resonance outputs of R^3 and the acoustic signals of S^3.

1.2 Motivation: Why SRC^9?

Traditional music theory frameworks start from notation and seek meaning through pre-defined symbolic systems. Meanwhile, modern machine learning models such as Jukebox or Magenta generate musical output with no semantic or cognitive interpretability. Neuroscientific studies, though rich in EEG/fMRI data, lack a bridge to music-theoretical relevance.

SRC^9 was created to solve this cross-domain disconnect by:

1. Starting with physical audio (not symbolic input)
2. Modeling resonance fields and energy topologies, not abstract pitch classes
3. Mapping these fields into time-aligned neural signatures rooted in empirical neuroscience

1.3 System Overview: A Perception-to-Structure Pipeline

SRC^9 operates as a real-time or batch-based pipeline:

$$\text{Audio Signal (Waveform)} \rightarrow \mathbf{S}^3 \rightarrow \mathbf{R}^3 \rightarrow \mathbf{C}^3 \rightarrow \text{Feedback}$$

- **S^3** produces framewise spectral data: $\text{partials}_t = \{\text{freq}, \text{dB}, \text{symbol}, \text{harmonic index}\}$
- **R^3** computes: Phantom Root (PR), Resonance Potential (Φ), Resonance Field Map (RFM), Cognitive Resonance Vector (CRV)
- **C^3** receives these values to drive:
  - Tension mapping (CTU)
  - Affective modeling (AOU)
  - Memory alignment (SAU)
  - Expectation violation (IEU)
  - Group synchrony and attention modeling (IRU, NSU)

Each stage of SRC^9 is aligned to a common temporal resolution (typically 0.1s), enabling multimodal synchronization with EEG, fMRI, audio playback, or Unity-based interactive scenes.

1.4 Formal Notation

Let $x(t)$ be the input waveform segmented into overlapping frames.

From S^3:

$$\text{Frame}_i = \{f_0, f_1, ..., f_{16}\}, \quad \text{where each } f_n = \text{partial}_{i,n}$$

From R^3:

$$\text{PR}_i, \Phi_i, \text{RFM}_i(f), \text{CRV}_i = \langle \text{TPS}_i, \text{TFI}_i, \text{NSF}_i \rangle$$

From C^3:

$$C^3(t) = \sum_{i=1}^{9} w_i \cdot \text{Unit}_i(t), \quad \text{Unit}_i(t) = \sum_j w_{ij} \cdot \text{Node}_j(t)$$

Each **Node** is an EEG/fMRI-derived observable (e.g., β phase-locking, MMN, BOLD z-score), anchored anatomically using MNI coordinates in a 3D GlassBrainMap interface.

1.5 Multimodal Feedback

The architecture supports **bidirectional modulation**:

$$\text{C}^3 \text{ feedback} \Rightarrow \text{R}^3 \text{ weighting} \Rightarrow \text{S}^3 \text{ filter adjustment}$$

For example:

- High CTU (tension) output may modulate the harmonic weights in R^3's Φ calculation
- Strong PIU (immersion) may suppress noisy partials in S^3's visualization layer
- SAU (semantic memory) may trigger audio annotations or dynamic re-sequencing

This loop forms the foundation of responsive, perceptually aware music analysis, generation, or education platforms.

1.6 Implementation Philosophy

SRC^9 is:

- **Scientific**: Grounded in psychoacoustics, computational neuroscience, and acoustic theory
- **Modular**: Built with independently executable units, fully API-controllable
- **Visual**: Every layer has an interpretable, dynamic output (HTML, PNG, 3D)
- **Interactive**: Exports to Unity, GlassBrainMap, OSC/VR platforms
- **Cognitively honest**: Models not just structure, but perception

Chapter II — Modular Dimensions of SRC^9

SRC^9 is divided into three orthogonal modules: S^3 (Spectral), R^3 (Resonance), and C^3 (Cognitive). Each functions as an independent layer in the signal–structure–perception continuum, while maintaining tight alignment through shared temporal schemas, compatible data models, and reciprocal feedback.

2.1 S^3 — Spectral Sound Space

Function: Transforms raw audio into high-resolution spectral frames by extracting:

- Fundamental frequency (f_0)
- Harmonic partials (1–16)
- Amplitude in dBFS
- Microtonal pitch symbols (e.g., A4¹, C3²)

Output Format: JSON array of frames at 0.1s intervals:

Tools Used:

- CREPE (f extraction)
- librosa (STFT, RMS, cent conversion)
- Custom Python pipeline with modular scripts

Scientific Rationale: Inspired by spectral music theory (Grisey, Murail) and auditory physiology (tonotopic mapping), S^3 treats the frequency domain as the true substrate of musical identity — discarding staff notation and tuning system assumptions.

2.2 R^3 — Resonance-Based Relational Reasoning

Function: Processes S^3 outputs to identify and model harmonic structure, not via tonal syntax, but via energetic interaction between partials.

Core Units:

- PRU — Phantom Root Unit: Detects implied fundamentals from overtone sets
- RPU — Resonance Potential Unit: Computes scalar coherence Φ per frame
- RFMU — Resonance Field Modeling Unit: Generates Gaussian field over frequency
- CRVU — Cognitive Resonance Vectoring Unit: Extracts TPS, TFI, NSF metrics

Field Representation:

$$\text{RFM}(f,t) = \sum_i A_i(t) \cdot e^{-\frac{(f-f_i(t))^2}{2\sigma^2}}$$

This formula defines a scalar resonance density field over log-frequency space.

Resonance Vector:

$$\text{C}\vec{\text{R}}\text{V} = [\text{TPS}, \text{TFI}, \text{NSF}] \in [0,1]^3$$

Used as the summary output of all R^3 activity and as input to C^3 modules.

Scientific Basis: Builds on psychoacoustic roughness theory (Plomp Levelt), neural entrainment (Bidelman), and just intonation topology (Sethares, Tymoczko).

2.3 C^3 — Cognitive Consonance Circuit

Function: Models how humans perceive, evaluate, and emotionally respond to the harmonic signals computed in R^3. Each response is neurophysiologically grounded and structured by a 9-Unit circuit.

C^3 Units:

- CTU — Cognitive Tension
- AOU — Affective Orientation
- IEU — Intuitive Expectation
- SRU — Somatic Resonance
- SAU — Semantic-Autobiographical
- PIU — Phenomenological Immersion
- IRU — Interpersonal Resonance
- NSU — Neural Synchronization
- RSU — Resonance Synthesis (summary vector)

Equation:

$$C^3(t) = \sum_{i=1}^{9} w_i \cdot \text{Unit}_i(t) \quad \text{where} \quad \text{Unit}_i(t) = \sum_j w_{ij} \cdot \text{Node}_j(t)$$

Each Node is mapped to EEG/fMRI features (e.g., alpha asymmetry, gamma coherence, BOLD z-scores) and anatomically located via MNI coordinates in the *GlassBrainMap*.

Data Flow: CRVU → CTU, AOU, PIU, NSU
Neural feedback loops modify RFM weighting, computation, and partial salience in S^3.

Scientific Integration: Combines computational music cognition (Lerdahl, Huron), neuroaesthetics (Zatorre, Koelsch), and brain–music entrainment literature.

Chapter III — Mathematical Foundations of SRC^9

SRC^9 formalizes music cognition through a hierarchy of equations, resonance functions, and vector spaces that bridge physical sound, psychoacoustic interaction, and perceptual abstraction.

3.1 Frame-Based Signal Model

Let the raw audio input be a continuous time-domain waveform $x(t)$. SRC^9 processes this waveform in fixed-length, overlapping windows:

$$x_i(t) = x(t + iH), \quad \text{for frame } i \tag{1}$$

Where:

- H = hop size (e.g., 10 ms)
- $x_i(t)$ = time-domain windowed signal

Each frame is then passed into CREPE or equivalent pitch tracking module to estimate:

$$f_{0i}, \quad A_i, \quad \text{partials } \{f_{in}\}_{n=1}^{16} \tag{2}$$

These define the fundamental + harmonic space used across the system.

3.2 Resonance Potential Equation ()

The scalar measure Φ represents the instantaneous coherence of all spectral partials within a frame:

$$\Phi(t) = \sum_{i<j} \frac{A_i(t) \cdot A_j(t)}{|f_i(t) - f_j(t)| + \epsilon} \tag{3}$$

Where:

- $f_i(t), A_i(t)$ = frequency and amplitude of partial i
- ϵ = small constant to avoid division by zero

Interpretation:

- Higher amplitude → more weight
- Smaller interval → stronger resonance

This equation generalizes roughness and consonance models using continuous frequency data.

3.3 Resonance Field (RFM)

To represent spectral resonance topographically, a Gaussian kernel density is applied across a log-frequency grid:

$$\text{RFM}(f,t) = \sum_i A_i(t) \cdot e^{-\frac{(f-f_i(t))^2}{2\sigma^2}} \tag{4}$$

This converts discrete spectral data into a continuous scalar field — a kind of "terrain map" of resonance.

Gradient Operator:

To compute directionality of tonal pull:

$$\nabla\text{RFM}(f,t) = \frac{\partial \text{RFM}(f,t)}{\partial f} \tag{5}$$

Used in CRVU → TFI to model spectral fusion.

3.4 Cognitive Vector (CRV)

The final cognitive resonance vector is defined as:

$$\vec{\text{CRV}} = [\text{TPS}, \text{TFI}, \text{NSF}] \tag{6}$$

Each metric is defined as:

TPS — Temporal Perceptual Stability:

$$\text{TPS} = \frac{1}{1+\sigma_\Phi(t)} \tag{7}$$

TFI — Tonal Fusion Index:

$$\text{TFI} = \frac{1}{1+\langle|\nabla\text{RFM}(f,t)|\rangle} \tag{8}$$

NSF — Neural Synchronization Field:

$$\text{NSF} = \sum_t \Phi(t) \cdot e^{-\alpha t} \tag{9}$$

Where α is a decay constant modeling attention/memory trace.

3.5 PR Estimation (Phantom Root)

Let $\{f_1, f_2, ..., f_n\}$ be a group of detected pitch events. Then, the phantom root r is the frequency that minimizes mean harmonic error:

$$r^* = \arg\min_r \left(\frac{1}{n} \sum_i \left| \frac{f_i - r \cdot h_i}{r \cdot h_i} \right| \right) \tag{10}$$

Where h_i are harmonic template integers (e.g., [1,2,3,4]).

3.6 Just Intonation Vector Representation

Each frequency can be projected into prime-exponent vector space:

$$\vec{v}_i = (x_2, x_3, x_5, x_7, ...) \quad \text{where } f_i = 2^{x_2} \cdot 3^{x_3} \cdot 5^{x_5} \cdots \tag{11}$$

Mean vector yields:

$$\vec{v}_{PR} = \frac{1}{N} \sum_i \vec{v}_i \tag{12}$$

Which is mapped back to the frequency domain to compute phantom root in symbolic-harmonic space.

3.7 Summary Table of Key Equations

Metric	Equation
Φ (Resonance Potential)	$\Phi(t) = \sum_{i<j} \frac{A_i A_j}{\lvert f_i - f_j \rvert + \epsilon}$
RFM (Field)	$\text{RFM}(f,t) = \sum_i A_i e^{-(f-f_i)^2/2\sigma^2}$
Gradient	$\nabla\text{RFM} = \partial\text{RFM}/\partial f$
CRV	[TPS, TFI, NSF]
PR Estimation	$r = \arg\min_r \sum_i \lvert f_i - rh_i \rvert / rh_i$
NSF	$\sum_t \Phi(t) e^{-\alpha t}$

Chapter IV — Temporal Architecture and Synchronization

Time is not merely a parameter in SRC^9; it is a structural axis along which all modules are synchronized, integrated, and compared. Every analytical unit, perceptual frame, and visual output is aligned to a common frame-based time grid, enabling coherence between real-time interaction, dynamic modeling, and retrospective analysis.

4.1 Frame Resolution

All SRC^9 operations are executed in frames of fixed temporal resolution.

- **Standard Frame Duration:** $\Delta t = 0.1$ seconds
- **Frames per 20-second audio:** 200 frames
- **Aligned Across:** $\text{S}^3 \rightarrow \text{R}^3 \rightarrow \text{C}^3$

This resolution provides a compromise between cognitive relevance (auditory segmentation and beat-level processes) and computational tractability.

4.2 Temporal Data Schema

Each frame is indexed and timestamped explicitly:

```
{
  "time": 3.2,
  "partials": [...],
  "phi": 2.83,
  "crv": {
    "TPS": 0.814,
    "TFI": 0.652,
    "NSF": 0.042
  }
}
```

Additional time-windowed metrics (e.g., windowed , RFM segments) use labeled intervals:

```
{
  "window": "5.0-8.0",
  "phi": 9.183,
  "window_size": 3
}
```

All time-based data are synchronized via integer multiples of Δt.

4.3 Window-Based Aggregation

Many perceptual phenomena operate on time windows rather than individual frames (e.g., expectancy, stability, modulation). To simulate this:

$$\Phi_T = \sum_{t=t_0}^{t_1} \Phi(t) \quad \text{where } T = [t_0, t_1] \tag{13}$$

Window lengths are configurable: 1s, 3s, 5s, or 7s (10–70 frames).

These windows feed CRVU and symbolic inference layers, and provide smoothed curves for visualization.

4.4 Inter-Unit Time Sharing

Each SRC^9 unit reads or writes data at the same temporal resolution, ensuring:

- Synchrony between harmonic events and cognitive metrics
- Real-time overlay of , RFM, PR, and CRV
- Accurate PRU segment demarcation based on CentTracker (±49c deviation)

Example: frame 38 at $t = 3.8$s will contain:

- 17 partials from S^3
- 1 Φ scalar from RPU
- RFM density array from RFMU
- 3-element CRV vector from CRVU
- Segment label if included in a PRU group

4.5 Real-Time Execution Model

To support live streaming or reactive composition, each frame can be evaluated asynchronously. A frame handler listens for input, processes data, and stores results:

Frame Pipeline:

Frame$_t$ $\Rightarrow$ S^3 extract $\Rightarrow$ R^3 process $\Rightarrow$ C^3 interpret $\Rightarrow$ Output + Feedba

Latency budget: < 50 ms per frame.

4.6 Timeline Synchronization with Audio/Video

SRC^9 includes support for timeline-aligned playback and export:

- **Unity integration:** `Time.time` $\leftrightarrow$ frame index
- **Audio export:** Link frame analysis to audio segments
- **Plotly visualizations:** Frame-aligned curves, scrollable graphs

Visual overlays are rendered in rasterized layers, each 216px tall, stacked into a 2160px 4K vertical space. These layers include:

- RawSpectrum
- PRU
- RPU
- RFMU
- CRVU

4.7 Frame Integrity and Diagnostics

Each frame includes metadata for traceability:

```
{
  "time": 12.3,
  "frame_id": 123,
  "source": "RawSpectrum01",
  "checksum": "ae347ac1...",
  "validated": true
}
```

This ensures reproducibility and integrity in batch pipelines or dynamic environments.

4.8 Temporal Modeling Summary

SRC^{9} temporal architecture transforms time from a passive marker to an active modeling dimension. It enables:

- Segment-based cognition modeling (e.g., tonal drift, root migration)
- Layer-aligned visualization of concurrent harmonic and perceptual states
- Real-time reactivity and temporal learning models

Chapter V — Data Structures and Interface Formats

The analytical power of SRC^{9} depends not only on its internal computations, but on its ability to represent, store, and exchange data in structured, interpretable, and extensible formats. This chapter outlines the file architectures, symbolic systems, and cross-platform export mechanisms that enable integration across scientific, educational, and creative platforms.

5.1 JSON Frame Format (Canonical)

All unit processing in SRC^{9} is time-aligned to frames in the following structure:

```
{
  "time": 3.2,
  "partials": [
    { "freq": 261.63, "amplitude": 0.84, "symbol": "C4", "harmonic_index": 0, "isFundamenta
    { "freq": 523.25, "amplitude": 0.51, "symbol": "C5", "harmonic_index": 1, "isFundamenta
  ],
  "phi": 2.83,
  "crv": { "TPS": 0.812, "TFI": 0.694, "NSF": 0.039 }
}
```

Specifications:

- `time`: Timestamp in seconds
- `partials`: List of harmonic components with symbolic pitch
- `phi`: Frame-level Φ scalar
- `crv`: Cognitive resonance vector output (from CRVU)

5.2 Unit-Specific Outputs

Each SRC^{9} unit outputs a structured file:

Unit	File	Contents
PRU	PR-unit-temporal.json	PR frequency, sym
RPU	RP-framewise.json, RP-windowed.json	Φ per frame or tim
RFMU	RFM-unit.json	Resonance field gr
CRVU	CRV-unit.json	Cognitive vector:

All outputs are timestamp-aligned at 0.1s resolution.

5.3 Symbolic Microtonal Encoding

SRC^{9} uses a compact symbolic system to encode pitch with microtonal precision:

- Format: `[PitchClass][Octave][Superscript]`
- Superscripts denote deviation in cents:
  - = 0 cent
  - = +25 cent
  - = –25 cent
  - = +50 cent

Examples:

- `C4` = C4 at 0c
- `A4`^{1} = A4 +25 cents
- `G3`^{2} = G3 –50 cents

This allows symbolic readability while preserving microtonal resolution from S^{3} partial tracking.

5.4 CSV Export for Unity and Visual Systems

Unity and WebGL-based environments operate on line-by-line streaming. Each partial is exported as:

```
time,freq,amplitude,isFundamental,harmonic_index,
1.0,220.0,0.82,True,0,G3
1.0,440.0,0.51,False,1,G4
```

Used to instantiate prefabs or terrain meshes in:

- `CSVLoader.cs`
- `SpectrumVisualizer.cs`
- `RFM Terrain Generator.cs`

5.5 Matrix and Vector Data Structures

Internally, each frame can also be represented in matrix form for ML pipelines:

$$\text{PartialMatrix}_t = \begin{bmatrix} f_0 & A_0 & h_0 \\ f_1 & A_1 & h_1 \\ \vdots & \vdots & \vdots \\ f_{16} & A_{16} & h_{16} \end{bmatrix} \quad \text{CRV}_t = \begin{bmatrix} \text{TPS}_t \\ \text{TFI}_t \\ \text{NSF}_t \end{bmatrix}$$

This supports CRV-based AI composition, harmonic fingerprint learning, or real-time ML inference.

5.6 File Structure Conventions

- `../data/raw/` — RawSpectrum-unit.json
- `../data/output/PR/` — Phantom root segments
- `../data/output/RP/` — Framewise/windowed
- `../data/output/RFM/` — Grid + gradient fields
- `../data/output/CRV/` — Cognitive vector layers

All files are UTF-8 encoded and stored in flat JSON or CSV formats for interoperability with scientific tools and frontend visual platforms.

—

“‘latex

Chapter VI — Visualization Layers and Multimodal Rendering

SRC^9 is designed not only to compute resonance and cognition, but to render it visually. Every analytical layer, from raw spectral frames to cognitive vectors, is projected into a coherent visual system aligned across time and frequency. These visualizations are not cosmetic: they serve as cognitive tools, allowing researchers, composers, and users to intuitively see the dynamics of harmonic structure.

6.1 Layer-Based Stack Design

SRC^9 visualization output is composed of vertically stacked unit layers, each aligned to a shared horizontal timeline (0–20s). Each layer occupies 216px vertical space, with the RawSpectrum occupying 1080px as base.

Layer	Visual Form	Height (px)
RawSpectrum (S^3)	colored partial markers	1080
PRU	red bars with pitch labels	216
RPU	Φ line + window overlays	216
RFMU	resonance field heatmap	216
CRVU	RGB stacked bars	216
Total	2160 px (4K)	

6.2 Visual Encoding Principles

Frequency: Y-axis (log scale)

Amplitude: Marker size, object scale, emission intensity

Time: X-axis (0–20s, 0.1s resolution)

Color:

- Frequency class (e.g., pitch class palette)
- Functional role (e.g., PR = red, Φ = gray, field = inferno colormap)
- CRV: red = TPS, green = TFI, blue = NSF

6.3 Plotting Tools

- **Plotly (Python):** for interactive HTML visualizations, hoverable markers, frame-aligned curves
- **Matplotlib:** for static PNG exports, segment overlays, symbol-annotated graphs
- **Unity (C#):** for 3D mesh-based field rendering and partial animation

6.4 Master Overlay Generation

Features:

- Frame-synchronized overlays
- Independent vertical scaling per layer
- Interactive time cursor
- Toggleable layers

6.5 Cognitive Color Mapping

6.6 RFM Surface Rendering in Unity

RFMU’s scalar field data are converted into 3D terrain meshes:

- X = time

- Z = frequency (log scale)
- Y = field strength → terrain height
- Emission map = normalized Φ
- Overlay: peak paths, PR lines, curvature ridges

Implemented using Unity's `MeshFilter`, `MaterialPropertyBlock`, and shader-based vertex displacement.

6.7 Animation and Playback Features

- Timeline scrubbing (linked to frame index)
- Real-time playback at 10 FPS (0.1s/frame)
- Sound-reactive visuals (optional)
- Dynamic camera tracking (e.g., PR curve follower)

6.8 Use Cases

- **Education:** visually teach harmonic fields, voice leading, polyphony
- **Analysis:** detect modulation, PR shift, dissonance zones
- **Performance:** display resonance terrain live in VR
- **Composition:** use RFM as a topographic canvas for generative tools

Chapter VII — Intermodular Feedback and Adaptive Control

A defining feature of SRC^9 is its recursive structure: each module not only feeds into the next but also receives feedback from downstream layers. This transforms the system from a static analyzer into a dynamic resonance engine — capable of adaptive learning, reweighting, and perceptually informed transformation.

7.1 Loop Architecture

The primary communication loop of SRC^9 is:

$$\mathrm{S}^3 \rightarrow \mathrm{R}^3 \rightarrow \mathrm{C}^3 \rightarrow \text{Feedback to } \mathrm{R}^3 \text{ or } \mathrm{S}^3$$

Forward Path:

- $\mathrm{S}^3 \rightarrow \mathrm{R}^3$: partial frames → harmonic reasoning
- $\mathrm{R}^3 \rightarrow \mathrm{C}^3$: CRV vector + Φ + RFM data

Feedback Path:

- $\mathrm{C}^3 \rightarrow \mathrm{R}^3$: attention, immersion, memory modulation
- $\mathrm{R}^3 \rightarrow \mathrm{S}^3$: spectral filtering, dynamic rescaling, visualization tuning

7.2 $\mathrm{C}^3 \rightarrow \mathrm{R}^3$ Feedback Mechanisms

Affective Salience (AOU, PIU): High immersion scores increase weight on core partials in RFM generation:

$$A_i^* = A_i \cdot (1 + \lambda_{\text{PIU}})$$

Tension Focus (CTU): RPU's Φ calculation uses tension-weighted denominators:

$$\Phi'(t) = \sum_{i<j} \frac{A_i A_j}{|f_i - f_j| + \epsilon} \cdot \omega_{\text{CTU}}$$

Memory Anchoring (SAU): SAU can extend windowed integration across prior phrases to model phrase re-entry or long-term attractor stabilization.

7.3 $\mathrm{C}^3 \rightarrow \mathrm{S}^3$ Modulation

- **Spectral Masking:** Hide partials with low salience or low CRV
- **Symbol Injection:** Annotate or override f labels with C^3-informed symbolic tags
- **Display Scaling:** Increase opacity/size of key partials if memory/affect signal is high

7.4 Modulation Example (Narrative Music)

Assume a phrase begins with stable CRV:

$$\vec{\mathrm{CRV}}_1 = [0.91, 0.88, 0.82]$$

The system increases visual brightness of corresponding partials and amplifies RFM terrain peaks.

A modulation or PR shift occurs:

$$\vec{\mathrm{CRV}}_2 = [0.41, 0.33, 0.17]$$

This results in:

- Sharpened ∇RFM contours
- Increase in partial flicker effect in Unity
- CRV bars collapse → signaling cognitive destabilization

7.5 Feedback API Specification

Feedback Packet (JSON):

```
{
  "time": 4.3,
  "feedback": {
    "CTU": 0.87,
    "PIU": 0.76,
    "NSU": 0.41
  }
}
```

Received by:

- RFM filter generator
- Partial weighting engine
- Visual modulation manager

7.6 Live Feedback and Loop Safety

- Feedback modulation is clamped between ±25% per frame
- Frame history buffers used to prevent oscillation artifacts
- Async-safe handlers allow interruption or override at runtime

7.7 Toward Resonance-Centric Interactivity

The feedback loop is not just for refinement. It enables new applications:

- Interactive composition: CRV → generative seed adjustment
- Brain–music co-evolution: EEG → C^3 → R^3 reshaping
- Self-regulating installations: perception → structure → perception

Chapter VIII — Scientific Contribution and Comparative Positioning

SRC^9 is more than a computational toolkit — it is a conceptual shift in how music is understood, analyzed, and linked to perception. This chapter contextualizes SRC^9 within existing scientific disciplines and explains its novel contribution to music theory, auditory neuroscience, cognitive modeling, and AI.

8.1 Bridging Fragmented Disciplines

- **Traditional Music Theory:** Offers symbolic, style-specific models (e.g., Roman numerals, keys) that lack generalizability to non-Western, microtonal, or electronically produced music.
- **Auditory Neuroscience:** Describes neural encoding of sound, but lacks structural models of music capable of predicting EEG/fMRI response.
- **AI/ML Music Systems:** Generate convincing audio, but are black-box and devoid of interpretability or symbolic grounding.

SRC^9 bridges these silos by combining spectral analysis, resonance modeling, and cognitive simulation into a unified framework.

8.2 Novel Contributions by Module

S^3 — Spectral Extraction

- Sub-cent partial tracking with harmonic identification
- Microtonal symbolic pitch encoding (±25c steps)
- 4K-resolution time–frequency mapping

R^3 — Resonance Modeling

- Real-valued Φ coherence computation
- RFM field topography and attractor mapping
- Phantom root detection without symbolic grammar
- Resonance-based perception modeling (TPS, TFI, NSF)

C^3 — Cognitive Circuitry

- Unit–Node–Element hierarchy with EEG/fMRI anchors
- Integration with real-time neural interfaces (e.g., OpenBCI, Emotiv)
- Emotional salience, memory encoding, and inter-brain synchrony modeling

8.3 Epistemological Reversal

Traditionally:

$$\text{Notation} \rightarrow \text{Structure} \rightarrow \text{Sound}$$

SRC^9 reverses this:

Sound → Structure → Perception

This change recognizes that human listeners don't hear scores — they hear waveforms, from which meaning emerges through spectral convergence, not grammatical rule sets.

8.4 Relationship to Existing Models

Model	Comparison to SRC^9
Lerdahl/Jackendoff GTTM	Symbolic-only, lacks spectral realism
Huron's ITPRA	Predictive cognition, no spectral grounding
Bregman's Auditory Scene Analysis	Compatible perceptually, no structural formalism
Schenkerian Analysis	Hierarchical tonality, score-dependent
Tonnetz/Neo-Riemannian Theory	Static topology, lacks time/frequency axes
MusicLM / Magenta	Non-interpretable deep generative models

8.5 Empirical Grounding

- Φ aligns with neural synchrony and EEG FFR data (Bidelman 2011)
- CRV mirrors attention and memory indices in fMRI studies (Zatorre et al. 2013)
- Microtonal segmentation reflects known auditory thresholds (Moore 2012)
- Temporal frame length matches auditory ERP resolution (MMN, P300)

8.6 Scientific Use Cases

- Neurocognitive analysis of music listening
- Dynamic modeling of musical form without score
- Empirical testing of tension, memory, or absorption in real time
- Cross-cultural harmonic modeling (e.g., gamelan, maqam, drone music)
- Tonotopic map visualization from real audio

8.7 Artistic Use Cases

- AI composition using CRV trajectories
- VR installations guided by RFM terrain
- Generative systems with real-time PR feedback
- Improvisation interfaces using Φ heatmaps and modulation vectors

8.8 Educational Use Cases

- Teaching spectral vs. symbolic harmony
- Visualizing modulation, drift, tension, and resolution
- Exploring affective resonance in sound
- Multisensory music learning through waveform → field → cognition

8.9 Future Research Integration

SRC^9 can integrate with:

- EEG systems (OpenBCI, Emotiv) — real-time feedback to CRVU
- Machine learning — CRV as feature vector for emotion or form prediction
- Neuroscientific experiments — auditory-cognitive mapping under stimuli
- Notational systems — hybrid symbolic–spectral scores

Chapter IX — Implementation Architecture and Development Overview

While SRC^9 is rooted in scientific theory and cognitive models, it is also a practical software system: a set of coordinated Python, JSON, CSV, Unity, and WebGL components that form a modular, executable pipeline.

This chapter describes the engineering architecture of SRC^9 — the file structures, runtime logic, APIs, and execution modes that bring its resonance engine to life.

9.1 System Overview

SRC^9 is composed of three primary code layers:

- **Core Analysis Layer:** Python modules for S^3, R^3, and C^3 computations
- **Visualization Layer:** Plotly, Matplotlib, and WebGL-based rendering scripts
- **Interaction Layer:** Unity scene controllers, OSC interfaces, and data streaming tools

9.2 Folder and File Structure

,

9.3 Execution Pipeline

Execution can occur:

- **Sequentially** — via run$_{S}RC9_{p}ipeline.py$**Manually**$^{-}unit - by - unitfordebugging$
- **Real-time** — via live frame ingestion, under development

Standard Pipeline Order:

1. S^3: CREPE → base spectrum
2. R^3: PRU, RPU, RFMU, CRVU
3. C^3: All 9 Units → summary JSON
4. Visualizer: Generate overlays and interactive outputs
5. Unity export: CSV + animation parameters

9.4 API Design Philosophy

Input: always JSON

Output: JSON + PNG + HTML (Plotly) + CSV (Unity)

Each function or script is:

- stateless (idempotent)
- reusable (called by other pipelines)
- visually testable (through plot outputs)

9.5 Modularity Map

modular_block_diagram.pdf

Modules are black-box compatible — meaning any layer (e.g., RFMU) can be replaced or extended with an alternate implementation without breaking upstream/downstream logic.

9.6 Runtime Profiling (Batch Mode)

On a standard system (Intel i7, 16 GB RAM):

Unit	Analysis Time (200 frames)	Visualization Tim
PRU	2.1 s	1.4 s
RPU	4.0 s	2.2 s
RFMU	5.2 s	3.1 s
CRVU	1.0 s	1.0 s
C^3 full unit set	∼9 s	—
Overlay (HTML)	—	3–5 s

Total batch time: 20–25 seconds per 20s audio input.

9.7 Unity Engine Integration

File Format: CSV **Core Classes:**

- `Partial.cs` — object representation
- `CSVLoader.cs` — parser and loader
- `SpectrumVisualizer.cs` — prefab instantiation
- `CRVOverlayHUD.cs` — affective bar display

Render Modes:

- glowing spheres for partials
- PR curves via `LineRenderer`
- terrain mesh for RFM via `MeshFilter`

9.8 Platform Compatibility

- **Python:** 3.9+
- **Unity:** 2021.3 LTS
- **Jupyter:** for experiment notebooks
- **Web:** HTML + Plotly/Three.js (experimental)
- **VR/OSC:** WebSocket-ready

9.9 Distribution and Open Source

SRC^9 is open-source under the MIT license. Code, data samples, visual exports, and Unity demos are available at:

`https://github.com/src9-framework/src9`

Contributors are invited to fork units, extend field models, or contribute to future modules such as OL (Overtone Locking) or GMI (Global Musical Inference).

Chapter X — Final Reflections and Theoretical Outlook

SRC^9 is not merely a framework, a pipeline, or a set of scripts. It is a paradigm: a new way of conceptualizing, measuring, and interacting with musical structure. It brings together physics, perception, cognition, and computation into a unified temporal field theory of music.

10.1 A New Definition of Harmony

Harmony is traditionally defined symbolically — as triads, scales, or functional progressions. SRC^9 proposes a redefinition:

Harmony is a time-varying field of structured resonance, shaped by energy, weighted by perception, and embedded in cognition.

Instead of working in discrete steps (e.g., I–IV–V), SRC^9 defines harmony as an evolving topology:

- **Attractors:** Phantom roots, perceptual centers
- **Gradients:** Tonal pull, dissonance slope
- **Fusion zones:** Φ coherence regions
- **Modulation:** Topographic drift in RFM space

10.2 Cognitive Resonance as Musical Logic

Through C^3, harmony becomes measurable not only in acoustics, but in brain-space:

- TPS → perceived stability
- TFI → spectral coherence
- NSF → memory encoding strength

This transforms music analysis into neuroperceptual logic — one grounded in actual listener behavior, emotion, and timing.

10.3 Generalization Across Styles and Cultures

Because SRC^9 is not bound to Western notation, keys, or 12-TET tuning, it generalizes to:

- Drone-based music (e.g., Indian raga, Tibetan chant)
- Just intonation and spectralism
- Electronic soundscapes and ambient textures
- Improvised music, microtonal works, non-metered environments

Its mathematical core — Φ, RFM, CRV — is culturally neutral but perceptually rich.

10.4 Toward a New Science of Sound

SRC^9 invites a reimagination of music cognition as a form of field-based reasoning:

Cognition is not symbolic parsing. It is real-time entrainment to dynamic

This claim opens doors to:

- New theories of musical time and memory
- Biofeedback systems that respond to sonic states
- Emotion-aware generative music engines
- Aesthetic theories rooted in resonance, not style

10.5 Final Statement

SRC^9 is designed to evolve.

It is not only a tool for analyzing the music of the past — it is a language for the music of the future. A music that is:

- Spectrally informed
- Resonantly grounded
- Cognitively engaged
- Mathematically coherent
- Visually immersive

The resonance field is the new score.

The CRV vector is the new expressive arc.

The mind–ear interface is the new stage.

This is the architecture of SRC^9.

Chapter X — Final Reflections and Theoretical Outlook

SRC^9 is not merely a framework, a pipeline, or a set of scripts. It is a paradigm: a new way of conceptualizing, measuring, and interacting with musical structure. It brings together physics, perception, cognition, and computation into a unified temporal field theory of music.

10.1 A New Definition of Harmony

Harmony is traditionally defined symbolically — as triads, scales, or functional progressions. SRC^9 proposes a redefinition:

Harmony is a time-varying field of structured resonance, shaped by energy, weighted by perception, and embedded in cognition.

Instead of working in discrete steps (e.g., I–IV–V), SRC^9 defines harmony as an evolving topology:

- **Attractors:** Phantom roots, perceptual centers
- **Gradients:** Tonal pull, dissonance slope
- **Fusion zones:** Φ coherence regions
- **Modulation:** Topographic drift in RFM space

10.2 Cognitive Resonance as Musical Logic

Through C^3, harmony becomes measurable not only in acoustics, but in brain-space:

- TPS → perceived stability
- TFI → spectral coherence
- NSF → memory encoding strength

This transforms music analysis into neuroperceptual logic — one grounded in actual listener behavior, emotion, and timing.

10.3 Generalization Across Styles and Cultures

Because SRC^9 is not bound to Western notation, keys, or 12-TET tuning, it generalizes to:

- Drone-based music (e.g., Indian raga, Tibetan chant)
- Just intonation and spectralism
- Electronic soundscapes and ambient textures
- Improvised music, microtonal works, non-metered environments

Its mathematical core — Φ, RFM, CRV — is culturally neutral but perceptually rich.

10.4 Toward a New Science of Sound

SRC^9 invites a reimagination of music cognition as a form of field-based reasoning:

Cognition is not symbolic parsing. It is real-time entrainment to dynamic

This claim opens doors to:

- New theories of musical time and memory
- Biofeedback systems that respond to sonic states
- Emotion-aware generative music engines
- Aesthetic theories rooted in resonance, not style

10.5 Final Statement

SRC^9 is designed to evolve.

It is not only a tool for analyzing the music of the past — it is a language for the music of the future. A music that is:

- Spectrally informed
- Resonantly grounded
- Cognitively engaged
- Mathematically coherent
- Visually immersive

The resonance field is the new score.

The CRV vector is the new expressive arc.

The mind–ear interface is the new stage.

This is the architecture of SRC^{9}.

$S3_{M}asterTechnicalReportAmacErdemMay2025$

S^3 Module – Master Technical Report (Enhanced)

Spectral Sound Space (S^3) Master Report Roadmap

I. Purpose and Position of the Report

Core Objective: The S^3 Module is the spectral analysis and fundamental sound modeling layer of the SRC system. This report documents all theoretical, algorithmic, mathematical, engineering, and aesthetic structures of the module on both scientific and applied levels.

Role in the System:

- Within SRC, S^3 forms the first layer of the S^3–R^3–C^3 system.
- This report establishes the data foundation for R^3 and C^3 integration.

II. Main Report Sections

1. Introduction and General Overview
2. Theoretical Foundations
3. Mathematical and Algorithmic Model
4. Signal Processing Pipeline Architecture
5. Data Structures and Formats
6. Visualization Layers
7. R^3 and C^3 Integration
8. Optimization and Performance
9. References and Sources
10. Appendices and Code Examples

III. Detailed Roadmap for Each Section

1. Introduction and General Overview

- General structure of the SRC system
- Role of S^3 in the system
- Scientific and aesthetic goals
- Development history
- Use cases: music analysis, composition, AI, cognitive science

2. Theoretical Foundations

2.1 Spectral Music Theory

- Summary of theoretical frameworks from Schaeffer, Grisey, Murail
- Frequency-based harmony
- Role of harmonic series in music

2.2 Psychoacoustic Foundations

- Critical bands
- Harmonic fusion
- Perceptual resonance

2.3 Musical Time and Frequency as a Unified Domain

- Time–frequency duality
- Limitations of Fourier theory

3. Mathematical and Algorithmic Model

3.1 Frequency Space and Microtonality

- Frequency distributions beyond 12TET
- Cent and pitch class calculations

3.2 Harmonic Structures

- Concept of partials
- Harmonic series tolerance (±50 cents)
- Partial grouping algorithms

3.3 Fundamental Frequency Estimation (f)

- Use of CREPE
- Comparison with pYIN
- Viterbi optimization

3.4 JSON Data Modeling

4. Signal Processing Pipeline Architecture

4.1 File Structure and Module Organization

- `audio/`, `json/`, `scripts/`, `output/`, `utils/`

4.2 Main Scripts

- `extract_frequencies_crepe.py`
- `harmonics_matching.py`
- `s3_visualization.py`

4.3 Step-by-Step Workflow

- Audio → f + dB → harmonic matching → color coding → visualization

4.4 Execution and Automation

- `start_pipeline.sh` structure
- Command examples

5. Data Structures and Formats

5.1 `partials.json`

- Frame structure
- Partial fields: freq, dB, isFundamental

5.2 `harmonics.json` (optional)

- Extended data post harmonic matching

5.3 `resonances.json`

- Prepared output for R³

5.4 `color_map.json`

- Optional: record of spectral color mapping

6. Visualization Layers

6.1 2D Map

- Time–Frequency map (3840×2160)
- Microtonal grid
- Donut Spectrum color transitions

6.2 3D Map

- Time–Frequency–Amplitude (x, y, z)
- Unity and Plotly usage
- Mesh and dot modes

6.3 Unity Integration

- CSV export format
- Prefab system
- Real-time camera and lighting setup

7. R³ and C³ Integration

7.1 Output for R³

- JSON → Resonance Potential (), Harmonic Distance (HD), Phantom Root (PR)

7.2 Preparation for C³

- Microtonal notation + amplitude → cognitive resonance estimation

7.3 Feedback Loops

- R³ → S³ color changes
- C³ → S³ attention level simulation

8. Optimization and Performance

8.1 Pipeline Performance

- JSON file size
- CREPE runtime
- Viterbi duration

8.2 Visualization Performance

- 4K render times
- Number of prefabs in Unity

8.3 GPU Acceleration (Future)

9. References and Sources

- Grisey, Murail, Schaeffer
- Lerdahl, Tymoczko, Sethares
- Cognitive modeling: Zatorre, Patel, Koelsch
- CREPE (Kim et al., 2018)
- Librosa and pYIN documentation
- Plotly, Unity technical docs

10. Appendices and Code Examples

- Code blocks and annotations
- JSON examples
- Color spectrum image (C → B transition)
- Unity scene settings (camera, light, prefab connections)
- Screenshots and final outputs

I.1 – Introduction and Overview

Spectral Sound Space (S^3) in the Architecture of SRC

The Spectral Sound Space (S^3) module constitutes the foundational analytic layer of the SRC framework—an interdisciplinary system that unites spectral acoustics, resonance modeling, and cognitive neuroscience. Positioned as the first pillar in the tripartite structure of SRC (S^3–R^3–C^3), S^3 provides the fundamental data structures and perceptual primitives from which harmonic reasoning (R^3) and neural interaction modeling (C^3) emerge.

Whereas traditional music analysis systems typically begin from symbolic notation (e.g., MIDI, scores), S^3 reverses the process: it operates directly on audio waveforms to extract a detailed, high-resolution representation of acoustic content in both time and frequency. This bottom-up approach ensures that the analytic foundation is directly grounded in the physical properties of sound, enabling it to generalize across styles, cultures, tuning systems, and performance modalities.

Motivations for S^3: Why a Spectral Module?

The design of S^3 is motivated by three fundamental observations:

- **Sound is inherently spectral.**
  Any auditory event can be decomposed into partials—individual frequency components that change over time in pitch, amplitude, and phase. These partials are the building blocks of tone perception and harmonic structure.
- **Spectral representations precede symbolic ones.**
  The human auditory system does not “hear notes,” but rather detects frequency patterns and intensity envelopes. Notation is a cultural abstraction layered atop an auditory substrate. Therefore, analysis should start at the spectral level if it aims to reflect perceptual and cognitive realities.
- **Harmony is a physical phenomenon before it is a theoretical one.**
  The perception of consonance, root, resonance, and tonality emerges from interactions between partials, not from theoretical scales. S^3 enables the measurement and visualization of these interactions in their raw, physical form.

The Role of S^3 in the Full System

In the architecture of SRC, S^3 performs three critical roles:

Data Generator

S^3 converts raw audio into structured data representations including:

- Fundamental frequencies (f) over time
- Partial tracks (harmonics and inharmonic components)
- Amplitude (in dB) of each component
- Microtonal symbolic notations and pitch-class mappings
- Spectral centroids, energy envelopes, and entropy values

Perceptual Filter

It selectively isolates acoustically meaningful content from noise, silences, and irrelevant transients using amplitude thresholds, frame-based smoothing, and overtone filters.

Visual Engine

S^3 produces high-resolution 2D and 3D visualizations of the spectral data:

- 2D time–frequency maps at 3840×2160 resolution
- 3D spectrograms with frequency–amplitude towers
- Interactive spectral canvases for analysis and composition

These outputs form the primary input for the R^3 module (Resonance-Based Relational Reasoning), where spectral data is analyzed for harmonic structure, resonance potential, overtone locking, and phantom root phenomena.

Interdisciplinary Relevance

The S³ module draws simultaneously from:

- **Spectral Music Theory (e.g., Grisey, Murail, Schaeffer):**
  Emphasizing sound itself as the basis of musical form, S³ adopts this premise and extends it computationally.
- **Signal Processing and Machine Learning:**
  Tools such as CREPE (deep learning pitch estimator), librosa (audio analysis library), and FFT algorithms are integrated to allow frame-level pitch and amplitude extraction with sub-millisecond precision.
- **Cognitive Neuroscience and Psychoacoustics:**
  Through microtonal accuracy and overtone modeling, S³ mirrors how the auditory cortex processes complex sound structures.
- **Music Technology and Visualization:**
  S³ interfaces directly with Unity, WebGL, and VR environments, producing real-time interactive spectral landscapes for education, analysis, and artistic use.

Conclusion: A Foundational Layer

The Spectral Sound Space (S³) module serves as the foundation of the SRC system. It provides the raw material—both data and perceptual structure—from which musical reasoning and cognitive mapping can emerge. Unlike traditional systems that operate on abstracted notation or symbolic input, S³ roots its analysis in the physical substance of sound.

Its capacity to extract, quantify, and visualize meaningful partials across styles and tuning systems makes it a uniquely versatile tool. Whether applied to Renaissance counterpoint, spectral composition, or neural music analysis, S³ enables a new form of bottom-up, physics-based music understanding.

I.2 – Theoretical Foundations

I.2.1 Sound Before Symbol: Epistemology of Sonic Analysis

The S³ module is predicated on a fundamental epistemological shift: that music analysis should begin not with the score, but with the sound itself. Western music theory has historically prioritized symbolic abstraction (notation, keys, chords), often detaching the study of music from the phenomena it arises from—vibrations in air, shaped by instruments, perceived by human bodies.

This module challenges that precedence.

Rather than assuming that sound serves as a mere carrier for symbolic content, S³ posits the inverse: symbolic constructs are interpretations of an underlying spectral substrate. Frequencies, amplitudes, and overtones are not peripheral—they are the music.

This shift aligns with the work of spectral composers (e.g., Gérard Grisey, Tristan Murail), auditory cognition researchers (e.g., Diana Deutsch, Albert Bregman), and philosophers of music (e.g., Pierre Schaeffer). S³ bridges their insights with modern computation.

I.2.2 Spectralism and Acoustical Foundations

The theoretical roots of S³ are deeply informed by spectralism: a movement in contemporary composition that foregrounds the timbral and acoustical properties of sound over traditional harmonic systems.

Key spectralist premises:

- Sound is a complex spectrum of partials, not a fixed pitch.
- Harmony arises from the overtone series, not from abstract interval systems.
- The orchestration of spectra defines form and tension more than functional harmony.

S³ translates these ideas into data structures:

- Every partial is tracked as an individual frequency–amplitude event.
- No assumption of equal temperament is made—frequencies are real-valued, microtonal.
- Harmonic relationships are computed, not assumed.

Through its modular design, S³ formalizes the intuition of the spectralists into a machine-readable format.

I.2.3 Psychoacoustics: Human Hearing and Spectral Perception

S³ is not merely mathematically accurate; it is psychoacoustically meaningful. Its architecture reflects how human auditory perception operates:

- **Critical Band Theory:** S³ models perceptual overlap via overtone locking mechanisms.
- **Auditory Scene Analysis:** Each partial is assigned an identity, allowing for grouping and source separation.
- **Temporal Integration:** Partial tracks are evaluated over time, reflecting how we perceive tone continuity.

These principles are embedded in the frame-based, high-resolution analysis pipeline. The system's sensitivity to cent-level frequency shifts, amplitude decay curves, and overtone fusion makes it not just accurate but also cognitively plausible.

I.2.4 Microtonality and Continuous Pitch Space

Unlike traditional pitch class systems which operate on discrete steps (12TET), S³ operates in continuous pitch space. Every frequency is stored as a floating-point value, and pitch class labeling is optional, reversible, and tolerant to cent deviations.

- Pitch is mapped not via quantization, but through continuous cent distance metrics.
- Microtonal variations (±5 to ±25 cents) are preserved and visualized explicitly.
- Symbolic mappings (e.g., C4¹) are generated only for readability—not as assumptions.

This enables the analysis of music outside the bounds of Western tuning: just intonation, 24-TET, gamelan pelog/slendro, non-octave repeating scales, or even completely aleatoric sound structures.

I.2.5 From Spectrum to Structure: Towards Resonance

Finally, the theoretical foundation of S³ leads naturally into R³, the Resonance-Based Relational Reasoning module.

Where S³ represents the raw materials of sound, R³ interprets those materials relationally:

- How do partials converge?
- Which phantom roots emerge?
- What is the resonance potential of a harmonic field?

S³ provides the data; R³ provides the reasoning.

This progression mirrors human musical experience:

- First, we hear sound (S³).
- Then, we infer structure and coherence (R³).
- Finally, we respond cognitively and emotionally (C³).

In this chain, S³ is the anchor: a physically-grounded, perceptually-aligned, computationally robust representation of what music is, before it is interpreted.

I.3 – Mathematical and Algorithmic Model

I.3.1 Overview of the Computational Framework

The S³ module translates raw acoustic signals into structured, symbolic, and interpretable representations. Its algorithmic core centers around three dimensions:

- Frequency (Hz)
- Amplitude (dB)
- Time (ms resolution)

These three variables are extracted and tracked for each partial—a distinct sinusoidal component of a sound. Unlike traditional pitch-tracking systems that detect only the fundamental (f_0), S³ treats the full overtone field as first-class data. This makes it possible to model harmonic content, resonance, and spectral evolution with great precision.

I.3.2 Signal Preprocessing and Frame Segmentation

The input signal $x(t)$ is divided into overlapping frames for analysis. Typical frame parameters are:

- Sampling rate: $f_s = 16000$ Hz (CREPE optimal)
- Frame length: 1024 samples ($\sim$64 ms)
- Hop size: 160 samples ($\sim$10 ms)

Let:

$$x_i(t) = x(t + i \cdot H), \quad \text{for frame } i$$

where H is the hop size. Frames are then passed into the frequency estimation pipeline.

I.3.3 Fundamental Frequency Extraction using CREPE

The system uses CREPE (Kim et al., 2018), a deep convolutional network trained to estimate the fundamental frequency of monophonic signals with high accuracy.

Given an audio frame $x_i(t)$, CREPE returns:

- f_{0i}: estimated fundamental frequency
- $c_i \in [0, 1]$: confidence score
- optionally, a_i: activation maps (ignored in S³)

The result is a time series:

$$\{(t_i, f_{0i}, c_i)\}_{i=1}^{N}$$

If $c_i < \theta$ (confidence threshold), the value is discarded or interpolated.

I.3.4 Amplitude Estimation and Normalization

Each frame's amplitude is derived via RMS energy or CREPE confidence:

$$A_i = 20 \cdot \log_{10}(\text{RMS}(x_i)) \quad \text{(dB)}$$

or, if using confidence:

$$A_i = 20 \cdot \log_{10}(c_i + \epsilon)$$

All amplitudes are normalized between $[-50 \text{ dB}, 0 \text{ dB}]$ and clipped accordingly.

I.3.5 Harmonic Field Construction

S³ computes not only the fundamental f_0, but constructs a harmonic field:

$$H_i = \{n \cdot f_{0i} \mid n = 1, 2, \ldots, N_h\}$$

where N_h is the number of harmonics tracked (typically 16). Each harmonic is stored with:

$$f_{i,n} = n \cdot f_{0i}$$

$$A_{i,n} = A_i - \delta(n) \quad \text{(attenuated by harmonic order)}$$

`isFundamental` is flagged `True` if $n = 1$.

These are saved per frame in the following format:

I.3.6 Pitch and Cent Calculation

For each partial, the module calculates the pitch in cents relative to A4 (440 Hz):

$$\text{cents}(f) = 1200 \cdot \log_2\left(\frac{f}{440}\right)$$

This allows:

- Microtonal deviation tracking (e.g., +8 cents)
- Conversion to symbolic pitch classes (e.g., A4)

This symbolic mapping is computed but not quantized; the raw frequency is retained as canonical.

I.3.7 Data Structures and Resolution

S³ maintains full resolution in time (10 ms), frequency (floating-point Hz), and amplitude (floating-point dB).

- **JSON storage:** arrays of frames with timestamps and partials
- **Data size:** ∼10,000 frames per 100 seconds of audio, ∼160,000 partials

I.3.8 Harmonic Matching and Partial Clustering

An optional phase identifies recurring partials and clusters them. This supports later stages in R³ (e.g., overtone locking and phantom root detection).

Matching is done by:

- Nearest-neighbor search across frames using frequency proximity
- Harmonic number estimation:

$$n = \text{round}\left(\frac{f}{f_0}\right)$$

Summary

The S³ module translates sound into a structured lattice of time–frequency–amplitude events. Its mathematical model ensures:

- Sub-millisecond temporal resolution
- Cent-level pitch accuracy
- Frame-wise harmonic field tracking
- Explicit microtonal notation
- Compatibility with resonance modeling in R³

I.4 – Signal Processing Architecture (Pipeline Design)

I.4.1 Overview

The signal processing architecture of the S³ module is designed as a modular, multi-stage pipeline that transforms raw audio into structured, symbolically annotated, and visually renderable spectral data.

This architecture balances three key principles:

- **High resolution:** Time frames in 10 ms steps, frequency in cent-level precision, amplitude in dB.
- **Modularity:** Each step is encapsulated as an independent script with defined input/output formats (typically JSON).
- **Extensibility:** The system supports integration with other modules (R³, C³), external libraries (CREPE, librosa), and interactive engines (Unity, Plotly).

I.4.2 Directory Structure

The pipeline operates within a clearly defined project structure:

```
S3-Module/
 audio/                    # Input audio files (.wav or .mp3)
    cello_suite_no1.wav
 json/                     # Intermediate and final data outputs
    base_frequencies.json
    partials.json
 output/                   # Visualization exports
    s3_visualization.png
    s3_visualization.html
 scripts/                  # Pipeline scripts
    extract_frequencies_crepe.py
    harmonics_matching.py
    s3_visualization.py
 utils/                    # Utility modules (shared functions)
    freq_to_rgb.py
    freq_to_microtonal.py
 requirements.txt
```

This structure ensures reproducibility and separation of concerns across computation, data, and display.

I.4.3 Pipeline Stages

Stage 1 – Fundamental Frequency Extraction (CREPE) Script: `extract_frequencies_crepe.py`

- **Input:** `.wav` or `.mp3` audio file
- **Parameters:** frame size, hop size, duration (default: 10 seconds)
- **Process:**
  - Load audio using librosa
  - Segment into overlapping frames
  - Pass frames to `crepe.predict()` for f_0 estimation
  - Estimate amplitude via RMS or CREPE confidence
- **Output (JSON):**

Stage 2 – Harmonic Field Construction Script: `harmonics_matching.py`

- **Input:** `base_frequencies.json` from Stage 1
- **Process:**
  - For each frame, extract fundamental f_0
  - Construct harmonics: $H_n = n \cdot f_0$ (typically for $n = 2$ to 16)
  - Assign decreasing amplitude per harmonic (e.g., -10 dB per step)
  - Merge harmonics with original fundamental
- **Output (JSON):** `partials.json` — contains full harmonic field per frame with `isFundamental` flags

Stage 3 – Spectral Visualization (2D/3D) Script: `s3_visualization.py`

- **Input:** `partials.json` from Stage 2
- **Process:**
  - Convert frequency to log-scale (Hz → cent)
  - Normalize amplitude to dB → size and opacity
  - Assign color using `freq_to_rgb()` (Donut Spectrum with 7-note color wheel)
  - Add microtonal labels using `freq_to_microtonal()`
- **Output:**
  - PNG at 3840×2160
  - Optional: HTML (Plotly interactive)

I.4.4 Execution Commands

Optional interactive output:

I.4.5 Automation: Master Script (Optional)

You may include a master pipeline script (`start_pipeline.sh`) that automates all stages:

I.4.6 Error Handling and Logging

Each script includes:

- Usage help when arguments are missing
- File existence checks for inputs
- JSON schema validation (planned)
- Logging system for progress and warnings

I.4.7 Modular API Design

The functions in each script can also be exposed via an internal API for integration with:

- Jupyter Notebooks (for educational/research use)
- Unity (real-time input/output via OSC or TCP)
- R³ module (resonance analysis integration)

Summary

The S³ module's signal processing architecture forms a clean, high-resolution, and extensible pipeline from sound to structure. It is engineered to allow spectral data to be transformed into meaningful musical structures, ready for further analysis by R³ and C³. The use of standard tools, modular scripts, and clear data formats ensures that the system is not only scientifically robust but also developer-friendly and future-proof.

I.5 – Data Structures and Formats

I.5.1 Overview

The Spectral Sound Space (S³) module operates on a carefully designed set of data structures to ensure maximum flexibility, resolution, and interoperability. These formats serve as both internal data representations and external interfaces for downstream modules (R³, C³), visualizations, and interactive environments (e.g., Unity).

The primary data structure is JSON, chosen for its human readability, machine parsability, and web compatibility. All spectral events—frequencies, amplitudes, time steps, and symbolic annotations—are encoded in JSON using standardized field names and schema.

I.5.2 Frame-Based Structure

At its core, S³ stores information as a sequence of time-ordered frames, each corresponding to a small window of the input signal (typically every 10 ms).

Structure:

Each frame contains:

- `time` (in seconds)
- `partials`: an array of objects describing frequency components

Each partial includes:

Field	Type	Description
`freq`	float	Frequency in Hz
`db`	float	Amplitude in dBFS (normalized between -50 and 0)
`isFundamental`	bool	Whether this is the frame's f_0
`note`	string	Symbolic label, e.g. `"C4+7"`
`rgb`	list	RGB color derived from freq (used in visualization)
`harmonicIndex`	int	1 for fundamental, 2+ for overtones

I.5.3 Data Resolution

Dimension	Resolution	Example
Time	10 ms (adjustable)	100 frames per second
Frequency	Float (cent-level precision)	442.7 Hz
Amplitude	Float (-50 to 0 dB)	-27.3 dB
Pitch Notation	Microtonal (cent offset)	A4

This high resolution enables perceptual modeling (via R³) and microtonal music analysis.

I.5.4 Supplementary Formats

a. CSV for Unity or WebGL: Used in 3D interactive rendering:

b. Mesh Export (grid-based): For 3D terrain mapping:

c. OSC / Real-Time Formats: (Optional) For integration with Unity or Max/MSP:

I.5.5 Schema Definitions (Formal)

You may enforce structure via JSON schema:

Frame Schema: This formalism ensures forward compatibility and guards against malformed data.

I.5.6 Storage Considerations

For a 10-second file with 10 ms frames and 16 partials per frame:

- 1,000 frames × 16 partials = 16,000 data points
- Approx. 2–5 MB as compressed JSON
- Easily processed in memory on modern systems

You can compress JSON with gzip or use a binary format (e.g., MessagePack) for lower latency streaming.

I.5.7 Reusability and Interoperability

- `partials.json` is the canonical format for R³ analysis
- `s3_visualization.py` reads from it to generate high-res plots
- `export_for_unity.py` converts it to CSV for interactive 3D display
- Optional mapping to `.S3N` format (SRC standard, in progress)

Summary

The data structures used by S³ are designed for high-resolution spectral modeling, downstream integration with resonance/cognitive modules, and compatibility with artistic, analytical, and interactive applications.

The JSON-based frame–partial model ensures that time, frequency, amplitude, and pitch data are tightly coupled and fully traceable. This standardization supports rigorous analysis, intuitive visualization, and machine readability.

I.6 – Visualization Layers and Aesthetic Design Principles

I.6.1 Overview

Visualization in the S³ module is not simply a graphical rendering of spectral data—it is a perceptually-aligned, musically meaningful, and aesthetically optimized representation of sound. The purpose of visualizing partials is twofold:

- **Analytic Insight:** Allow researchers, theorists, and composers to examine the spectral and temporal structure of sound at microtonal and microtemporal resolution.
- **Cognitive Alignment:** Mirror how auditory structures are perceived, allowing visual artifacts to stand in for psychoacoustic phenomena such as overtone fusion, resonance, and vibrato.

The design principles of S³ visualizations stem from an integrated philosophy of scientific legibility, musical interpretation, and visual minimalism.

I.6.2 Two-Tier Visualization Architecture

The system employs two synchronized visual layers:

Tier 1: Spectral Density Map (2D or 3D)

- **X:** Time (seconds), linear scale
- **Y:** Frequency (Hz), log scale
- **Z:** Amplitude (only in 3D mode)

Units:

- Resolution: 3840 px × 2160 px (default)
- Frame step: 10 ms
- Frequency precision: cent-level ($\sim$0.58 px per cent)

Each partial is visualized as a dot or micro-line, positioned at its (time, frequency) coordinate, and styled according to amplitude and pitch-class-based color.

Tier 2: Symbolic Notation Layer

- Aligned on X (time) with Tier 1
- Contains labels for:
  - Fundamental pitches (e.g., A4, C5)
  - Onset durations (shown as segment lengths)
  - Microtonal deviations (in cent format)
- Can be toggled or overlaid for interpretive use

This layer enables mapping from physical spectrum to musical language (e.g., score-independent notation).

I.6.3 Color Mapping: Donut Spectrum

Instead of static color coding (e.g., red = high freq), S³ implements a musically cyclic, frequency-based color system:

Principle: Colors cycle with octaves, not linear Hz.

Base hue anchors:

- C: Red
- D: Orange
- E: Yellow
- F: Green
- G: Light Blue
- A: Blue
- B: Violet

Intermediate tones interpolate between anchors.

Repeat per octave: $\log_2(f/f_{\text{ref}}) \mod 1$

Example:

- 261.6 Hz (C4) $\rightarrow$ Red
- 440 Hz (A4) $\rightarrow$ Blue
- 1046 Hz (C6) $\rightarrow$ Red again

Mathematical Mapping:

$$\theta(f) = (\log_2(f/f_{\text{ref}}) \mod 1) \cdot 360^\circ$$

Converted to HSV hue $\rightarrow$ RGB

This mapping aids intuitive identification of tonal color, enhances spectral grouping perception, and allows visual equivalence across octaves.

I.6.4 Amplitude and Visual Emphasis

Amplitude (in dB) is mapped to:

- Dot size (larger = louder)
- Opacity (higher dB = more solid)
- Optional Z-height (in 3D mesh)

Amplitude range is normalized between -50 dB and 0 dB.

dB	Size Range	Visual
-50	1.0 px	barely visible
-30	3.5 px	translucent
-10	6.5 px	prominent
0	8.0 px	fully saturated

I.6.5 Hover and Interaction Design (Plotly/Unity)

In 2D interactive views (Plotly), each partial responds to hover:

Display: This ensures each data point is interpretable in real time and can be cross-referenced with musical structure.

In Unity, similar hover behavior is achieved using Raycast + Tooltip systems.

- **Clarity over Colorfulness:** Every color and glyph must carry meaning—no arbitrary or decorative elements.
- **Sonic Minimalism:** Reflect the sparsity of partials, allow negative space, and avoid visual clutter.
- **Cognitive Load Management:** Layer complexity gradually (e.g., hide partials $<$ -40 dB by default), allow user toggles.

I.6.7 Multi-format Outputs

- **PNG:** Static, high-resolution (e.g., for papers, print). Rendered at 3840×2160 using Plotly + Kaleido.
- **HTML:** Interactive hover-capable plots. Zoom, layer toggle, export options.
- **CSV (Unity):** For 3D rendering (frequency → height). Used in VR/AR sound-space explorations.

I.6.8 Future Features

- Temporal motion blur to represent vibrato or tremolo
- Animated playback with cursor-following time marker
- 2.5D stacked pitch-class visualization (similar to piano roll, but spectral)

Summary

Visualization in the S³ module is not a cosmetic feature—it is a tool for perception, cognition, and musical logic. Every graphical element is tied to a psychoacoustic or musical principle, and the rendering stack ensures that physical properties of sound are transposed into visually legible, interpretable, and aesthetically powerful structures.

I.7 – R³ and C³ Integration Architecture

I.7.1 Introduction

While S³ provides a high-resolution, physically-grounded representation of sound, it is only the first layer of the broader SRC system. The modules that follow—R³ (Resonance-Based Relational Reasoning) and C³ (Cognitive Consonance Circuit)—rely on the structured spectral data output by S³ to compute:

- Harmonic relationships (R³)
- Resonance fields and perceptual centers (R³)
- Neural correlates of sound structure (C³)
- Bounded attention, emotional salience, and cognitive load (C³)

This section describes how data flows from S³ into R³ and C³, and how architectural compatibility is maintained across analytical, real-time, and interactive systems.

I.7.2 Data Flow Overview

From S³ to R³:

- **Input:** `partials.json`
- R³ extracts for each frame:
  - f_0 (fundamental)
  - Full harmonic field
  - Frequency ratios
  - Harmonic intervals
  - Microtonal deviations

From R³ to C³:

- R³ outputs for each frame:
  - Resonance potential (Φ)
  - Harmonic distance (HD)
  - Phantom root (PR)
  - Overtone locking (OL)

  - Vectorized harmonic embeddings
- C^3 computes:
  - Temporal Perceptual Stability (TPS)
  - Tonal Fusion Index (TFI)
  - Neural Synchronization Fields (NSF)

I.7.3 Interface Specification

JSON Format Standards: Each module reads and writes time-aligned frame arrays with shared conventions.

Input from S^3:

I.7.4 Modular Code Interface (Python)

Each module exposes a consistent API:

Example (R^3):

Example (C^3): This chainable interface design enables automated pipelines, real-time computation, and batch processing.

I.7.5 Streaming Compatibility (Optional)

S^3 frames can be streamed in real time (e.g., OSC or WebSocket) and passed frame-by-frame to downstream modules.

Example OSC message:

- R^3 computes Φ and PR on the fly.
- C^3 updates perceptual stability fields.

This architecture supports use in:

- VR/AR environments
- Generative composition engines
- Real-time performance analytics

I.7.6 Synchronization and Latency

To ensure downstream module alignment:

- All frames are timestamped with exact onset time
- Optional global clock source can synchronize external sensors (e.g., EEG, motion)
- Each module can interpolate, pad, or drop frames to maintain temporal consistency
- Maximum allowable latency for inter-module propagation: < 20 ms

I.7.7 Use Cases

Use Case	Modules Involved	Description
Harmonic Field Tracking	$S^3 \rightarrow R^3$	Spectral compo
Tonal Center Estimation	$S^3 \rightarrow R^3 \rightarrow C^3$	Phantom root i
Real-Time Attention Feedback	$S^3 \rightarrow C^3$	EEG alignment
Audio–Visual Synchronization	$S^3 \rightarrow R^3$ + Unity	Spectral peaks

I.7.8 Technical Stack and Dependencies

- **S^3:** CREPE, librosa, numpy, Plotly, Unity (visual output)
- **R^3:** Custom harmonic analysis engine, fractional prime decomposition, Euler Tonnetz geometry
- **C^3:** numpy, scipy, TensorFlow (optional for neural modeling), EEG data pipelines

I.7.9 Logging and Diagnostics

Each module logs:

- Frame time
- Input checksum (for verification)
- Output integrity (range checks)
- Processing time (profiling)
- Synchronization offsets

A central dashboard (`src9_monitor.py`) allows tracking of inter-module health and alignment.

Summary

The integration of S^3 with R^3 and C^3 establishes a vertically layered architecture:

- S^3 → raw perceptual primitives
- R^3 → harmonic structure and resonance logic
- C^3 → cognitive and emotional interpretation

The clear API design, standardized data structures, and optional real-time compatibility ensure that all modules communicate reliably, maintain synchronization, and can evolve independently while sharing a unified foundation.

I.8 – Optimization and Performance

I.8.1 Overview

Due to its high-resolution time–frequency modeling, microtonal accuracy, and support for large-scale audio datasets, the S³ module requires careful performance tuning. This section details the computational characteristics of each pipeline stage and outlines optimization strategies at multiple levels:

- Algorithmic
- Architectural
- Real-time constraints
- Cross-platform compatibility (desktop, embedded, Unity)

I.8.2 Bottlenecks by Pipeline Stage

Stage	Description	Typical Bottleneck	Mitigation
f Extraction (CREPE)	Neural net inference	CPU-bound	Batch inference or GPU acceleration
Amplitude Estimation	RMS over frames	I/O bound	Pre-slice audio, frame caching
Harmonic Field Generation	Harmonic expansion per frame	Memory/compute	Vectorized array ops (NumPy)
Symbolic Mapping	Cents + RGB + note assignment	None (fast)	Already optimized
Visualization	Plotly rendering	GPU/UI bottleneck	Static output or throttled rendering
Unity Export	CSV generation	Disk I/O	Streamed JSON → buffer cache

I.8.3 Temporal and Spectral Resolution

Time 10 ms hop (100 FPS): sufficient for most music. Adjustable to 5 ms (high accuracy) or 20 ms (fast).

Frequency Floating point (e.g. 442.76 Hz): cent-level (∼0.6 px) precision. No quantization unless explicitly requested.

Amplitude Normalized between -50 dB and 0 dB. Visualization supports amplitude-dependent size and color mapping.

I.8.4 Memory Footprint

Assuming a 10-second clip at 100 FPS, 16 partials per frame:

- Total frames: 1,000
- Total partials: ∼16,000
- Typical JSON size: 2–5 MB uncompressed
- In-memory size: ∼8–12 MB (with symbolic fields)

Optimization Tip: For long-form analysis, stream partials per segment (e.g., 100 frames) into memory, then flush.

I.8.5 Code-Level Optimizations

Use NumPy for harmonic expansion:

Use list comprehensions and avoid deeply nested loops.

Batch compute cent/pitch/RGB fields per frame.

Avoid recalculating pitch class mappings if frequency hasn't changed.

I.8.6 Visualization Optimization

2D (Plotly or Matplotlib)

- Throttle marker size and opacity for very low dB
- Fade partials < -40 dB (optional toggle)
- Use Kaleido for static rendering instead of Orca (faster)

3D (Unity)

- Use `DrawMeshInstanced()` instead of `GameObject` clones
- Use object pooling
- Export only "significant" partials (e.g., fundamental + harmonics up to -35 dB)

I.8.7 Real-Time Constraints

To enable interactive use (e.g., in VR or live audio streams):

- Use frame queues to pre-load analysis windows
- Perform f_0 + harmonic expansion in a separate thread
- Use audio input ring buffers (with PyAudio or SoundDevice)

Target latency budget: $<$ 50 ms end-to-end

I.8.8 Cross-Platform Performance

Platform	Strategy
Desktop (Python)	Full pipeline with CREPE + Plotly
Unity (C#)	CSV import + GPU mesh render
Web (WebGL)	Pre-rendered `.html` or WASM visualizer
Embedded (Raspberry Pi)	Use downsampled audio + partial-only ex

I.8.9 Performance Logging and Metrics

S³ includes performance tracking hooks:

Future improvement: a dashboard that reports:

- FPS throughput
- Memory usage
- Partial density over time
- Processing heatmap

I.8.10 Future Optimizations

- Use GPU-accelerated libraries (e.g. CuPy, TensorRT for CREPE)
- Parallel frame analysis with `multiprocessing` or `joblib`
- Use lightweight binary formats (`.msgpack` or `.protobuf`) for JSON

Summary

S³ is designed for high fidelity and extensibility, but with attention to efficiency at each level. By combining frame-wise processing, vectorized operations, intelligent filtering, and GPU/Unity export strategies, the system remains responsive and scalable—from short musical phrases to full-length performances, from desktop analysis to embedded playback.

I.9 – References and Source Integration

I.9.1 Overview

The S³ module is grounded in a rich body of theoretical, technical, and scientific literature. This section documents the foundational sources that inform the system's design, including references from music theory, signal processing, psychoacoustics, and cognitive neuroscience. It also provides integration notes for each cited source, detailing how the ideas have been translated into algorithmic and computational form within S³.

I.9.2 Spectral Music and Acoustic Theory

Gérard Grisey & Tristan Murail – Spectral Aesthetics Reference: Grisey, G. (1996). *Did You Say Spectral?*

Contribution:

- Rejection of abstract harmonic systems in favor of the overtone series
- Advocacy for time–frequency as a compositional space

S³ Integration:

- Partial tracking over time
- Harmonic field construction
- Overtone-based pitch logic

Pierre Schaeffer – Acousmatic Perception Reference: Schaeffer, P. (1966). *Traité des objets musicaux*

Contribution:

- Classification of sonic objects based on spectral content

S³ Integration:

- Time-framed spectral analysis
- Object-based spectral segmentation

I.9.3 Psychoacoustics and Human Perception

Albert Bregman – Auditory Scene Analysis Reference: Bregman, A. S. (1990). *Auditory Scene Analysis*

Contribution:

- Stream segregation, grouping of partials by proximity

S³ Integration:

- Harmonic clustering
- Fundamental and overtone coherence tracking

Diana Deutsch – Perception of Pitch and Illusions Reference: Deutsch, D. (1982). *The Psychology of Music*

Contribution:

- Illusory pitch perception, frequency grouping

S³ Integration:

- Microtonal deviation representation
- Phantom root preparation for R³

Moore, B. – Critical Bands and Masking Reference: Moore, B. C. J. (2003). *An Introduction to the Psychology of Hearing*

Contribution:

- Modeling auditory filters and perceptual interference

S³ Integration:

- Overtone locking zone threshold (used in OL module of R³)

I.9.4 Signal Processing and Pitch Estimation

Kim et al. – CREPE Pitch Tracker **Reference:** Kim, J., Salamon, J., Li, P., & Bello, J. P. (2018). *CREPE: A Convolutional Representation for Pitch Estimation*

Contribution:

- Deep learning-based frame-by-frame pitch estimation

S³ Integration:

- Used for extracting f_0 from raw audio at 10 ms resolution

Librosa – Python Audio Toolkit **Reference:** McFee, B., Raffel, C., Liang, D., et al. (2015). *librosa: Audio and Music Signal Analysis in Python*

Contribution:

- Audio loading, STFT, RMS, cent mapping

S³ Integration:

- Core utility for RMS estimation, time axis construction

I.9.5 Mathematical and Symbolic Systems

Lerdahl & Jackendoff – Generative Theory of Tonal Music **Reference:** Lerdahl, F., & Jackendoff, R. (1983). *A Generative Theory of Tonal Music*

Contribution:

- Cognitive modeling of grouping, metric structure

S³ Integration:

- Inspired tiered visualization (partials vs. symbolic layer)

Tymoczko, D. – Geometric Music Theory **Reference:** Tymoczko, D. (2011). *A Geometry of Music*

Contribution:

- Mapping musical structures to geometric topologies

S³ Integration:

- Inspires geometric pitch-space design (future S³–R³ integration)

Sethares, W. – Tuning, Timbre, Spectrum, Scale **Reference:** Sethares, W. A. (2005). *Tuning, Timbre, Spectrum, Scale*

Contribution:

- Non-Western tuning systems and perceptual consonance

S³ Integration:

- Support for non-12TET frequencies and just intonation

I.9.6 Cognitive and Neural Foundations

Zatorre, Koelsch, Patel – Music and the Brain **References:**

- Zatorre, R. J. (2002). *Structure and function of auditory cortex*
- Koelsch, S. (2011). *Towards a neural basis of music perception*
- Patel, A. D. (2008). *Music, Language, and the Brain*

Contribution:

- Mapping music perception to cortical activity

S³ Integration:

- Informing the downstream design of C³ module
- Supporting microtemporal precision as neurologically relevant

I.9.7 Visualization Systems

Plotly – Interactive Graphics **Reference:** Plotly (open-source docs)

Contribution:

- High-resolution, log-frequency 2D plotting

S³ Integration:

- Used for `s3_visualization.py`, hover data, export to PNG/HTML

Unity – 3D Audio Visualization **Reference:** Unity Technologies (docs)

Contribution:

- Mesh, prefab, and real-time rendering

S³ Integration:

- Unity receives CSV exports for real-time 3D partial landscapes

I.9.8 Future Integrations

- Open Sound Control (OSC): For live audio input/output into S³ pipeline.
- VR/AR Extensions: Using Unity WebXR for immersive spectral interaction.
- EEG Integration APIs: Connecting S³ output with neural data for use in C³.

Summary

The S³ module is deeply grounded in a wide range of academic sources, and every computational decision is anchored in at least one theoretical or empirical reference. From pitch tracking to visualization, from psychoacoustics to symbolic abstraction, S³ reflects a comprehensive and scholarly synthesis of the last century of acoustical, musical, and perceptual research.

I.10 – Appendices and Code Examples

I.10.1 Overview

This section provides detailed technical examples, supplemental illustrations, and code snippets to support the core content of the S³ module. These resources are designed for developers, researchers, and artists seeking to extend, test, or embed S³ into larger computational or artistic environments.

Each appendix includes a fully functional code excerpt, JSON schema, visualization samples, and system integration examples.

I.10.2 Appendix A: CREPE Extraction Script (`extract_frequencies_crepe.py`)

I.10.5 Appendix D: Microtonal Notation Generator

I.10.8 Appendix G: Screenshots and Diagrams

Omitted in text version – included in report PDF or interactive Jupyter companion notebook.

- Spectrogram at 3840×2160 resolution
- Donut color wheel illustration
- Partial track overlays
- Unity-based 3D cube field

Summary

The **S³ MasterTechnicalReport(Enhanced)** concludes its first part with complete technical references, reusable code snippets, and implementation-ready structures. These examples empower researchers, developers, and artists to reconstruct the entire pipeline, audit its logic, or embed it into larger analytical or creative ecosystems.

This foundation now fully supports the integration of R³ (Resonance-Based Relational Reasoning) and C³ (Cognitive Consonance Circuit), enabling next-generation modeling of music as a spectrum–resonance–consciousness continuum.

R3-MasterTechnicalReport-Enhanced Amac Erdem May 2025

1. Introduction

I.1(1) – Motivation and Origins (Enhanced)

Music theory, for centuries, has been dominated by symbolic and categorical thinking. Chords are labeled; keys are named; progressions are prescribed. These abstractions, while elegant and effective in many stylistic contexts, fail to reflect the continuous, physical nature of acoustic phenomena and the non-discrete, probabilistic mechanisms of perception. This misalignment between how music is theorized and how it is actually heard is a foundational problem that motivates the creation of R³.

In recent decades, both scientific and artistic movements have exposed the limits of traditional pitch-class and functional harmony systems:

- **Spectral music**, led by composers such as Gérard Grisey and Tristan Murail, focused on the actual overtone content of sound rather than idealized chords.
- **Psychoacoustic studies** (Terhardt, 1974; Plomp & Levelt, 1965) demonstrated that perceived pitch and consonance are emergent properties of partial alignment, not of symbolic classification.
- **Neuroscience findings**, including frequency-following response (FFR) and brainstem phase-locking studies (cf. Bidelman et al., 2011), showed that the brain tracks periodicity and overtone structures even without conscious musical attention.
- **Just intonation and extended harmonic systems** (Doty, 2002; Sethares, 1998) provided mathematical models of tuning that highlight resonance, not abstraction, as the organizing principle of pitch space.

The convergence of these fields leads to a necessary shift in harmonic reasoning: from symbolic theory to resonance-based computation.

R³ (Resonance-Based Relational Reasoning) is designed to model music not as a grammar of discrete signs, but as a flow of structured vibrational energy. It treats pitch as a function of spectral gravity, energy proximity, and temporal anchoring, rather than categorical labeling.

At the core of R^3 lie three principles:

- **Tonal centers are emergent**
  $\rightarrow$ They result from statistical convergence of overtones, not pre-defined labels.
- **Resonance is continuous**
  $\rightarrow$ Harmonic coherence is not binary (consonant vs. dissonant), but a gradient, computable via Φ.
- **Perception is relational**
  $\rightarrow$ Tonal stability is derived from how partials interact — in time, in frequency, and in amplitude — not from isolated entities.

This paradigm is grounded not only in music but in broader systems theory, signal processing, and neuroacoustics. The resonance field becomes the new tonal map, where perception, meaning, and emotion are drawn not on a grid of pitches, but across a topology of dynamic acoustic pressure.

As Zatorre and Salimpoor (2013) observed, musical reward correlates with prediction and violation in time-dependent structures — R^3 provides the computational substrate for such dynamics. Through Φ, PR, and RFM, it becomes possible to chart the internal logic of sound as it unfolds, not as a score, but as a fluid field of tonal potential.

Summary

The motivation for R^3 is not merely the refinement of harmony theory. It is the redefinition of what harmony is: no longer a symbolic artifact, but a topological, energetic, and cognitive resonance surface.

I.2(1) – Integration within SRC Architecture (Enhanced)

SRC is designed as a modular, multi-domain cognitive-auditory system with three principal modules:

- **S^3:** Spectral Sound Space – acoustic extraction, microtonal analysis
- **R^3:** Resonance-Based Relational Reasoning – harmonic topology, field modeling
- **C^3:** Cognitive Consonance Circuit – perceptual synthesis, memory, valuation

R^3 sits at the exact center of this architecture — mathematically, informationally, and conceptually.

It functions as the Y-axis of the SRC lattice, where:

- X = spectral frequency space (S^3)
- Y = resonant interaction space (R^3)
- Z = perceptual response dimension (C^3)

This triaxial topology is not metaphorical — it is computationally enforced through data routing, inter-unit feedback, and shared time/frequency schemas.

1. Input from S^3 R^3's input is the fully expanded harmonic spectrum:

```
{
  "time": 2.1,
  "partials": [
    {
      "freq": 196.0,
      "amplitude": 0.84,
      "isFundamental": true,
      "harmonic_index": 0,
      "symbol": "G3"
    },
    ...
  ]
}
```

Sampling: 0.1s frames (200 per 20s session)

Per Frame: 1 f + 16 harmonics

Extras: Microtonal symbol mapping, cent deviation flags

Each R^3 unit extracts features relevant to its function (e.g., PR focuses on f_0 trajectory; RFM on amplitude–frequency density).

2. Internal Resonance Processing (R^3) Within R^3, each unit runs in temporal and spectral parallel. However, their semantic roles are different:

Unit	Function	Outputs To
PRU	Virtual root estimation	RP, RFM, CRV
RPU	Framewise & windowed Φ	CRV, RFM
RFMU	Field topology (RFM, ∇RFM)	CRV, visualization
CRVU	Vector summary [TPS, TFI, NSF]	C^3 – attention & memor

Together, these units transmute the raw spectrum into a dynamic resonance surface — a structure rich in perceptual cues, mathematical relations, and cognitive triggers.

3. Output to C^3 R^3's final output — the CRV vector — is transmitted to C^3 for use in:

- Attention anchoring: TPS stability scores influence focus
- Fusion response modeling: TFI reflects perceptual unity
- Temporal expectation modeling: NSF controls time-weighted relevance

These values modulate cognitive consonance curves and neural resonance gating within higher-order evaluative modules.

4. Bidirectional Modulation Though R^3 receives data from S^3 and passes to C^3, the flow is not strictly feedforward.

R^3 can also be:

- Influenced by C^3 feedback, adjusting Φ weighting or field granularity based on attention level
- Trigger of S^3 re-analysis, when PR or TFI instability passes a threshold

This recurrent loop allows SRC to behave like a resonant cognitive engine — not just analyzing, but reacting to musical content dynamically.

5. Implementation Topology The data flow between S^3, R^3, and C^3 can be visualized as:

```
              [ S³ ]  →  Raw Spectrum (partials)
                ↓

               R³
           PR →  → RFM

                ↓
            [ CRV ]
                ↓
              [ C³ ]
        (Attention, Valuation, Memory)
```

Each arrow represents a JSON/array structure or vector of numerical values, all time-aligned, standardized, and validated.

6. Architectural Significance R^3 enables SRC to:

- Move beyond symbolic analysis
- Embrace probabilistic, topological, and energy-based models
- Bridge physics (S^3) with cognition (C^3)
- Operate in real-time or offline batch analysis
- Output data suitable for visualization, sonification, or interactivity (Unity)

II.1(2) – Domain Definitions: Traditional vs. Resonance-Based Theory

In the architecture of SRC, a domain is not merely a categorical label — it defines the epistemic framework by which musical structure is interpreted. Domains provide the foundational assumptions, theoretical orientation, and algorithmic style of all downstream unit processing.

Two such domains are currently defined within the R^3 module architecture:

A. Traditional-Based Theory (Reserved Domain)

Philosophical Basis:

- Music is a symbolic grammar of signs.
- Tonality arises from hierarchical relations between named pitch classes.
- Harmony is a sequence of discrete, functional progressions (e.g., tonic → dominant → subdominant).

Historical Sources:

- Rameau's fundamental bass theory
- Roman numeral analysis
- Common-practice tonality (1600–1900)
- Schenkerian structuralism

Computational Analogy:

- Rule-based systems
- Symbol-to-symbol transitions
- State machines over pitch-class sets

Current Status in SRC:

- Unpopulated. The domain is intentionally preserved for comparative or pedagogical use but no active units are assigned.

Future Possibility:

- Incorporating classical root analysis
- Mapping Roman numeral logic to PR output
- Providing symbolic contrast to field-theoretic R^3 outputs

B. R^3 – Resonance-Based Theory (Active Domain)

Philosophical Basis:

- Music is a flow of interacting energy fields.
- Tonality emerges from the statistical convergence of partials.

- Harmony is a dynamic topology shaped by amplitude, frequency, and time.

Scientific Foundations:

- Psychoacoustics (Plomp & Levelt, 1965; Terhardt, 1974)
- Spectral composition (Grisey, Murail)
- Just intonation and tuning theory (Doty, Sethares)
- Auditory neuroscience (Zatorre, Bidelman, FFR)

Computational Strategy:

- Vector calculus in spectral domains
- Probabilistic detection of resonance attractors
- Topographic field modeling (RFM)
- Temporal variance and cognitive vector construction (CRV)

Comparison Table: Symbolic vs. Resonant Modeling

Aspect	**Traditional Theory**	**R^3 – Resonance-Based Theory**
Pitch Type	Discrete pitch classes	Continuous frequency space
Harmony Basis	Root progression (symbolic)	Spectral proximity & convergence
Time Treatment	Syntactic units (measures)	Framewise + windowed resonance maps
Tonal Center Definition	Key, tonic function	Phantom root via overtone intersection
Mathematical Form	Rule-based grammar	Scalar field + differential operators
Cognitive Link	Abstract schema	Auditory energy alignment (FFR)
Adaptivity	Static system	Real-time dynamic modeling

Why Dual Domains?

SRC includes both domains to support:

- Comparative modeling
- Pedagogical clarity
- Theoretical transparency

Although the Traditional-Based Theory domain is empty, its presence allows users to compare symbolic models vs. resonance models, test different theoretical assumptions, and use R^3 as both an analytical tool and a research platform.

Eventually, units such as:

- Tonic Function Classifier (TFC)
- Symbolic Root Evaluator (SRE)
- Key Probabilistic Mapper (KPM)

...may populate the Traditional domain for hybrid analysis.

Domain Assignment Summary

Unit	**Domain**
PRU	R^3 – Resonance-Based
RPU	R^3 – Resonance-Based
RFMU	R^3 – Resonance-Based
CRVU	R^3 – Resonance-Based
OL (future)	R^3 – Resonance-Based
Traditional-Based Theory	(empty)

II.2(1) – PRU: Phantom Root Unit (Enhanced)

The Phantom Root Unit (PRU) models one of the most perceptually paradoxical yet experientially central aspects of musical cognition: the ability to perceive a fundamental pitch even when it is not acoustically present. This unit formalizes and operationalizes the phenomenon of "virtual pitch" by detecting statistical convergence across harmonic spectra and time.

1. Theoretical Background and Cognitive Justification

Since the 1970s, experimental psychoacoustics has demonstrated that listeners can identify the "missing fundamental" of a complex tone purely from overtone relationships (Terhardt, 1974; Houtsma & Goldstein, 1972). These percepts are non-linear and emergent: they do not result from individual partials, but from their alignment in log-frequency space.

Auditory scene analysis research (Bregman, 1990) shows that virtual pitch perception is guided by temporal stability, harmonic simplicity, and statistical regularity. PRU unifies these dimensions using an intersection-based harmonic model, a vector-space fusion score, and symbolic microtonal tagging.

In neurophysiological terms, virtual pitch tracking is associated with brainstem-level phase locking (cf. Bidelman & Krishnan, 2011), where neurons track periodicity of inferred tones even when the spectral content is incomplete. PRU aligns with this processing architecture.

2. Input Requirements

- **Source:** `RawSpectrum-unit.json`
- **Frame resolution:** 0.1s
- **Each frame includes:** fundamental + 16 harmonics
- **Symbolic labeling:** cent-deviation-based (e.g., G3^1)

Partial data are first segmented into fundamental frequency sequences, grouped by ±49 cent stability window, and then processed through pattern recognition templates.

3. Core Algorithmic Stages

a. CentTracker Tracks whether the current fundamental frequency remains within a ±49 cent band. Once a deviation exceeds threshold, a note boundary is declared.

Let f_n be the current f_0 value:

$$\text{if } |\text{cents}(f_n, f_{n-1})| > 49 \Rightarrow \text{new note}$$

This creates time-stable pitch clusters, e.g., G2 (1.4s–2.1s), D3 (2.2s–3.3s), G3 (3.4s–4.2s)

b. GroupMatcher Matches these sequential pitch groups to harmonic templates such as:

$$\text{1, 2, 3} \rightarrow \text{simple harmonic stack}$$

$$\text{1, 2, 3, 4, 5} \rightarrow \text{extended overtone set}$$

$$\text{1–11} \rightarrow \text{full resonance model (A4 group)}$$

Given N consecutive note frequencies $\{f_1, f_2, \ldots, f_N\}$, PRU seeks a scalar r such that:

$$f_i \approx r \cdot h_i \quad \text{for harmonic template } H = \{h_1, h_2, \ldots, h_N\}$$

The best-fit r becomes the phantom root candidate.

c. Harmonic Intersection Scoring (HIS) To evaluate whether partials align at a common origin, PRU computes:

$$\text{HIS}(r) = \sum_{i<j} \#(H_i \cap_\epsilon H_j)$$

Where:

- $H_i = \{k \cdot f_i\}$: harmonic set of note i
- $\cap_\epsilon$: fuzzy intersection under cent-tolerance ϵ

This gives a resonance-weighted score of harmonic "compatibility."

d. Fusion Metric: Harmonic Information Score (HIM) As an alternative or supplement, PRU may also compute the HIM:

$$\text{HIM}\left(\frac{b}{a}\right) = \frac{a \cdot b}{a + b - 1}$$

This models perceptual fusion likelihood (cf. Sethares, 1998) and allows comparing different PR candidates.

e. Prime-Exponent Averaging (Optional) For symbolic systems like Just Intonation or Prime-Limit modeling, PRU represents each note as a vector:

$$\vec{v}_i = (x_2, x_3, x_5, x_7, \ldots), \quad \text{where } f_i = 2^{x_2} \cdot 3^{x_3} \cdot 5^{x_5} \cdots$$

The mean vector:

$$\vec{v}_{PR} = \frac{1}{N} \sum_i \vec{v}_i$$

...is projected back into frequency space to suggest a symbolic root in prime-vector space.

f. SymbolMapper Final phantom root frequency is mapped into symbolic pitch notation:

- Cent bins: ±25 cent steps
- Output: `C3²`, `A#2¹`, etc.

4. Output Format

```
{
  "time_range": [2.1, 4.2],
  "phantom_root": 130.81,
  "symbol": "C3¹",
  "group": "A2",
  "fusion_score": 0.431
}
```

Each record corresponds to a stable perceptual segment.

5. Integration Downstream Use

- **Feeds RPU:** defines resonance weighting center
- **Feeds RFMU:** sets initial attractor in field
- **Feeds CRVU:** primary input for TPS (temporal stability)

II.2(2) – RPU: Resonance Potential Unit ()

The Resonance Potential Unit (RPU) provides a scalar measure of spectral coherence — quantifying how "tightly" the partials within a time window resonate with one another. The Φ metric is a mathematically continuous and perceptually grounded proxy for what listeners describe as "harmonic richness," "fusion," or "tonal gravity."

1. Scientific Foundation and Perceptual Basis

Consonance perception is not binary but continuous. Listeners experience certain spectra as more stable or fused depending on how close and strong partials are in frequency and amplitude. This insight, originating from psychoacoustic studies by Plomp & Levelt (1965), forms the perceptual core of Φ.

Further evidence from Bidelman et al. (2011) and Leman (2016) suggests that harmonic coherence triggers neural entrainment and reward systems. High Φ values are theorized to activate neural structures such as the nucleus accumbens (associated with musical pleasure) and generate higher predictive certainty in auditory processing streams.

2. Input Model

- Input file: `RawSpectrum-unit.json`
- Frame duration: 0.1 seconds
- Each frame: list of partials (freq, amp)

All Φ computations are frame-aligned, amplitude-weighted, and cent-aware.

3. Core Algorithmic Flow

a. Pairwise Φ Matrix For every frame t, a full pairwise distance matrix is constructed across all partials:

$$\Phi(t) = \sum_{i<j} \frac{A_i \cdot A_j}{|f_i - f_j| + \epsilon}$$

Where:

- A_i, A_j: amplitudes
- f_i, f_j: frequencies
- ϵ: small constant (1e)

This formulation ensures:

- Higher amplitude = higher resonance impact
- Smaller frequency distance = stronger spectral fusion

Each $\Phi(t)$ is a scalar representing the spectral “density” or “pull” within that frame.

b. Time-Windowed Integration (Φ_T) To capture longer-term coherence, sliding time windows (1s, 3s, 5s, 7s) are defined:

$$\Phi_T = \sum_{t \in T} \Phi(t)$$

This models harmonic pressure across time — analogous to field density in physics.

Used for:

- Detecting modulation
- Smoothing over local instability
- Feeding into NSF (memory-weighted) metrics in CRVU

c. Optional Field-Weighted Φ Future extensions may modulate Φ with field attractor weights (from RFM):

$$\Phi_{RFM}(t) = \sum_{i<j} \frac{A_i \cdot A_j \cdot \gamma(f_i, f_j)}{|f_i - f_j| + \epsilon}$$

Where γ is a Gaussian function centered on current PR or spectral centroid.

4. Fluctuation Modeling

Variance in $\Phi(t)$ over time is highly correlated with tonal anchoring. A stable tonal segment will show low variance; modulation or dissonance increases fluctuation.

$$\sigma_\Phi(t) = \frac{1}{N} \sum_{i=1}^{N} \left(\Phi_i - \bar{\Phi}\right)^2$$

This index feeds into CRVU’s TPS node (Temporal Perceptual Stability).

5. Output Formats

Framewise:

```
{
  "time": 4.1,
  "phi": 3.142
}
```

Windowed:

```
{
  "window": "6.0-9.0",
  "phi": 9.762,
  "window_size": 3
}
```

Both forms are visually overlaid in multi-colored Φ curves (e.g., red = 1s, blue = 7s).

6. Integration Points

- **Feeds CRVU:** TPS (std dev), NSF (sum)
- **Feeds RFMU:** used as raw data for Gaussian smoothing
- **Optional input from PRU:** PR can shift Φ weighting center

7. Theoretical Impact

Φ unifies the perceptual with the computational:

- It approximates roughness, fusion, and tension metrics
- It enables real-time scalar monitoring of harmonic convergence
- It mathematically extends Plomp–Levelt's roughness curve to high-resolution, amplitude-weighted spectral structures

II.2(3) – RFMU: Resonance Field Modeling Unit (Enhanced)

The Resonance Field Modeling Unit (RFMU) constructs a continuous, dynamic field over frequency space that represents the distribution of spectral energy and tonal gravity. It transforms discrete spectral events into a spatiotemporal topography, allowing resonance to be visualized, measured, and interpreted as a scalar field.

1. Conceptual Foundation: Harmony as Field

Traditional models define harmony as sequences of pitch-class structures. RFMU redefines harmony as field energy: the shape, slope, and peaks of a continuously distributed amplitude-weighted spectral surface.

This approach is inspired by:

- Physical fields in electromagnetism and gravity
- Topographic models of pitch space (e.g., Tymoczko, 2006)
- Neural tonotopic maps (Moore, 2012)
- Auditory Gestalt theory, where pitch centers attract auditory focus

In RFM, harmonic tension and resolution are not rules — they are gradients across a vibratory surface.

2. Input Structure

- **Source:** `RawSpectrum-unit.json`
- **Per frame:** list of partials (freq, amp)
- **Each partial:** becomes a kernel on the field

3. Mathematical Core

a. Resonance Field Equation

$$\text{RFM}(f,t) = \sum_{i=1}^{N} A_i(t) \cdot e^{-\frac{(f - f_i(t))^2}{2\sigma^2}}$$

Where:

- f: target frequency grid point
- $f_i(t)$: frequency of partial i at time t
- $A_i(t)$: amplitude of partial i
- σ: resonance spread parameter (controls field smoothness)

This Gaussian convolution converts discrete partials into a smooth frequency-density curve.

b. Grid Discretization The frequency space (20–20000 Hz) is divided into N points:

$$f_j = \text{logspace}(20, 20000, N)$$

N is typically 512 or 1024, log-scaled to match auditory perception.

c. Field Gradient Operator (∇RFM)

$$\nabla\text{RFM}(f,t) = \frac{\partial \text{RFM}(f,t)}{\partial f}$$

Implemented numerically via finite differences.

- Points toward direction of tonal pull
- Magnitude of ∇RFM is used in CRVU's TFI node

d. Peak Tracking At each time t, RFMU extracts local maxima (peaks) in the field:

- These peaks = tonal centers
- **Stability** = persistence of peak across frames
- **Drift** = shift in peak position over time

4. Visualization Logic

- **2D heatmap:**
  - x = time
  - y = frequency (log scale)
  - color = field intensity (e.g., inferno colormap)
- **Optional overlays:**
  - white lines = peak paths
  - vector arrows = ∇RFM direction

5. Output Format

```
{
  "time": 3.5,
  "grid": [20.0, 25.1, ..., 20000.0],
  "field": [0.001, 0.0023, ..., 0.0],
  "gradient": [0.004, 0.003, ..., -0.002]
}
```

6. Cognitive Function

RFMU makes it possible to:

- Observe resonance motion as a flow
- Detect zones of spectral convergence (tonal mass)
- Model directional pull (gradient dynamics) of harmonic tension
- Build resonance topographies for spatial cognition

7. Integration Pathways

- **To CRVU:**
  - ∇RFM → TFI
  - RFM stability → TPS supplement
- **To PRU:** Peak history → implied root movement
- **To C³:**
  - Field visualization → VR/AR immersion
  - Resonance attractors → attention targets

8. Future Extensions

- **3D Field Volume:** Extend RFM to include time-depth (spectrovolume)
- **Field Curvature Analysis:** Use ∇^2RFM to detect dissonance basins
- **Resonance Memory Maps:** Integrate ∇RFM into NSF decay structures

Summary

RFMU is where harmony becomes spatial. It transforms the linear structures of tonal analysis into landscapes of spectral motion — surfaces that pull, release, and define perceptual musical form.

II.2(4) – CRVU: Cognitive Resonance Vectoring Unit (Enhanced)

The Cognitive Resonance Vectoring Unit (CRVU) synthesizes the outputs of R³ into a compact, perceptually-relevant vector. It provides a quantitative bridge from spectral resonance data to cognitive interpretation, enabling SRC to assess how stable, fused, and memory-relevant a sound structure is over time.

CRVU defines tonal cognition not as symbolic logic, but as a vector of resonance behaviors.

1. Scientific & Neurocognitive Motivation

Listeners do not passively receive spectral information — they process it in terms of:

- **Stability:** Is the tonal center clear and persistent?
- **Fusion:** Do the partials cohere into a unified sound?
- **Memory anchoring:** Does the resonance persist cognitively over time?

These questions are reflected in core neural structures:

- TPJ + ACC: involved in perceptual switching and ambiguity resolution
- Auditory cortex: phase-locking and coherence detection
- Hippocampus: short-term memory formation of tonal events
- Nucleus accumbens: reward prediction via temporal regularity

CRVU mathematically simulates these perceptual axes through three metrics: TPS, TFI, and NSF.

2. Input Requirements

CRVU consumes:

- Framewise Φ (from RPU) → resonance magnitude over time
- Gradient field ∇RFM (from RFMU) → fusion and divergence flow

Each input is frame-aligned at 0.1s resolution, processed into single or windowed summary statistics.

3. Metric Computation

a. Temporal Perceptual Stability (TPS)

$$\text{TPS} = \frac{1}{1 + \sigma_\Phi(t)}$$

Computes standard deviation of Φ over entire (or local) window.

- High TPS = consistent tonal field → strong perceptual anchoring

b. Tonal Fusion Index (TFI)

$$\text{TFI} = \frac{1}{1 + \langle|\nabla \text{RFM}(f,t)|\rangle}$$

Measures average steepness of resonance field across time.

- Flatter fields = stronger fusion
- Steeper slopes = divergence, spectral instability

c. Neural Synchronization Field (NSF)

$$\text{NSF} = \sum_t \Phi(t) \cdot e^{-\alpha t}$$

Applies exponential decay ($\alpha = 0.05–0.1$) to earlier $\Phi(t)$ values.

- Models short-term auditory memory + attention decay
- High NSF = early, strong, cohesive resonance = likely to be encoded

4. Node Architecture

Node	Input	Output
TPSNode	$\Phi(t)$	Scalar $\in [0,1]$
TFINode	$\nabla\text{RFM}(f,t)$	Scalar $\in [0,1]$
NSFNode	$\Phi(t)$, α	Scalar $\in [0,1]$

Each node is independent but computed over the same time base.

5. Output Format

```
{
  "TPS": 0.812,
  "TFI": 0.693,
  "NSF": 0.0385
}
```

Vector = 3-tuple $\langle$stability, fusion, memory$\rangle$

Normalized to $[0,1]$ for visual and cognitive mapping.

6. Visual Encoding

- Displayed as stacked horizontal bars (216px high)
- Red = TPS, Green = TFI, Blue = NSF
- Used as perceptual "signature" for a sound segment
- Overlayable on waveform, RFM, or symbolic score

7. Integration with C³

CRVU directly feeds into C³'s interpretive layers:

- **CTU – Cognitive Tension Unit:** TPS modulates expectedness
- **NSU – Neural Sync Unit:** NSF correlates with phase-locking metrics
- **PIU – Phenomenological Immersion Unit:** TFI relates to absorption metrics

CRVU is the only R³ unit whose output is designed to directly map onto affective and attentional models.

8. Future Extensions

- Weighted CRV vectors across musical phrases
- Real-time CRV streaming for interactive music engines
- CRV-linked generation: use resonance signature to drive AI composition
- Fusion + stability mapping across multichannel inputs (ensemble CRV)

Summary

CRVU is the cognitive mirror of R³ — a window into how sound, structured by physics and filtered by resonance, becomes psychologically meaningful.

III.1 – Phantom Root Estimation (Enhanced)

The phenomenon of phantom root perception — the brain's ability to identify a "missing" fundamental from a group of overtones — is among the most counterintuitive findings in auditory science. Unlike direct pitch recognition, it requires a form of inferred periodicity, where the brain estimates the source of harmonic structure based solely on spectral relationships.

The PRU formalizes this cognitive mechanism using harmonic intersection and vector matching models.

1. Harmonic Intersection Formula

The core mathematical function is:

$$\text{PR} = \arg\max_{f} \sum_{i<j} \#(H_i \cap_\epsilon H_j)$$

Where:

- $H_i = \{k \cdot f_i \mid k \in \mathbb{N}\}$
- $\cap_\epsilon$: fuzzy intersection within a cent tolerance ϵ (typically ± 49 cents)
- $\#$: counts the number of overlapping harmonics

Interpretation: This function seeks the base frequency f whose harmonic series would generate the highest number of overtone alignments across a group of perceived pitches. Even if f is not acoustically present, it may be implied by these intersections.

2. Harmonic Template Matching

To operationalize this, PRU evaluates grouped sequences (e.g., G2–D3–G3) against canonical harmonic stacks:

- Group A: [1,2,3]
- Group A1: [1–5]
- Group A2: [1–7]
- Group A3: [1–9]
- Group A4: [1–11]

Given note sequence $\vec{f} = [f_1, f_2, ..., f_n]$, we search for base frequency r such that:

$$f_i \approx r \cdot h_i \quad \forall i, h_i \in H$$

Where H is the harmonic template.

A candidate is accepted if average error:

$$\epsilon_{\text{avg}} = \frac{1}{n} \sum_i \left| \frac{f_i - r \cdot h_i}{r \cdot h_i} \right| < \delta$$

(Default: $\delta = 0.03$, i.e., 3% deviation)

3. Harmonic Information Metric (HIM)

To further differentiate candidates, a perceptual fusion metric is computed:

$$\text{HIM}\left(\frac{b}{a}\right) = \frac{a \cdot b}{a + b - 1}$$

As introduced in Sethares (1998), this metric estimates how well a ratio $\frac{a}{b}$ supports tonal fusion. Lower denominators and lower sums produce higher HIM values — signaling simpler, more consonant ratios.

Example:

- 3:2 (perfect fifth): HIM = $4/6 = 0.667$
- 7:4 (septimal minor 7th): HIM = $10/28 \approx 0.357$

4. Prime-Exponent Vector Averaging

For advanced systems supporting symbolic pitch spaces (e.g., Just Intonation), each pitch is expressed as a vector:

$$\vec{v}_i = (x_2, x_3, x_5, x_7, \ldots) \quad \text{where } f_i = 2^{x_2} \cdot 3^{x_3} \cdot 5^{x_5} \cdots$$

The mean vector:

$$\vec{v}_{PR} = \frac{1}{N} \sum_{i=1}^{N} \vec{v}_i$$

...is mapped back to a rational frequency. This method allows geometric averaging of complex harmonic ratios and facilitates field-aware root finding.

5. Time-Aware PR Estimation

Crucially, PR is not computed frame-by-frame but across time-stable pitch sequences. This enables it to:

- Capture phrasing-based tonal centers
- Filter out transient modulations
- Model tonality as a temporally weighted attractor

Segments are defined using CentTracker:

New group starts when f_0 deviates $> \pm 49$ cents from previous.

Result: [G2] → [D3] → [G3] → matched to [1,2,3] = PR: G1

6. Output Summary

Each PR record includes:

- `time_range`: [start, end] of stable group
- `phantom_root`: root frequency in Hz
- `symbol`: user-defined microtonal pitch label (e.g., D3²)
- `group`: matching harmonic template label (e.g., A2)
- `fusion_score`: optional metric from HIM or harmonic count

III.2 – Resonance Potential Formalism ()

The Resonance Potential (Φ) is the fundamental scalar measure of harmonic coherence within the R³ framework. It captures how energetically close — and thus perceptually "fused" — a group of partials are at a given moment. Unlike symbolic harmonic functions (e.g., tonic, dominant), Φ offers a mathematically continuous, spectrally grounded, and amplitude-sensitive metric for tonal tightness.

1. Core Equation

The main formulation of Φ is defined over all pairwise combinations of partials within a time slice:

$$\Phi(t) = \sum_{i<j} \frac{A_i(t) \cdot A_j(t)}{|f_i(t) - f_j(t)| + \epsilon}$$

Where:

- $A_i(t), A_j(t)$: amplitudes of partials at time t
- $f_i(t), f_j(t)$: their frequencies
- ϵ: a small regularization constant (e.g., $1e^{-6}$) to prevent division by zero

This equation models:

- Higher amplitude ⇒ greater resonance contribution
- Closer frequencies ⇒ stronger spectral fusion
- Denser clusters ⇒ higher perceptual cohesion

2. Perceptual Foundations

Φ generalizes earlier psychoacoustic roughness models (e.g., Plomp & Levelt, 1965), replacing frequency ratios with physical frequencies and amplitude scaling.

It corresponds to perceptual phenomena such as:

- Tonal fusion
- Consonance gradience
- Spectral "weight" of a sound structure

EEG studies (Bidelman et al., 2011) suggest that high harmonic coherence triggers stronger FFR synchrony — supporting Φ as a proxy for perceived resonance strength.

3. Time-Windowed Integration (Φ_T)

To move from instantaneous coherence to temporal resonance modeling, Φ is accumulated across time windows:

$$\Phi_T = \sum_{t=t_0}^{t_1} \Phi(t)$$

Where:

- T = time window (e.g., 1s, 3s, 5s, 7s)
- Frames sampled at 0.1s ⇒ Φ_T includes 10–70 values

This windowed Φ_T models:

- Tonal momentum (field pressure)
- Stability regions (high and flat Φ_T)
- Modulation zones (Φ_T dips or spikes)

4. Information-Theoretic Interpretation

Φ can be viewed as an inverse spectral entropy measure.

- A spectrum with many equally spaced partials ⇒ lower Φ
- A tightly clustered, loud spectrum ⇒ higher Φ

Φ is therefore analogous to a negative KL divergence between energy distributions.

This link allows R³ to potentially connect with probabilistic models of expectation and surprise (e.g., Huron's *Sweet Anticipation*, 2006).

5. Spectral Weighting Options

a. Harmonic Rank Weighting Later harmonics receive less weight:

$$A_i^* = \frac{A_i}{1 + h_i}$$

b. Gaussian Spectral Masking Include field density around a pitch center (e.g., PR):

$$\Phi_{\text{centered}}(t) = \sum_{i<j} \frac{A_i \cdot A_j \cdot e^{-\frac{(f_i - \mu)^2}{2\sigma^2}}}{|f_i - f_j| + \epsilon}$$

Where μ is the perceptual center of gravity.

6. Output Summary

Framewise Output:

```
{ "time": 3.2, "phi": 2.831 }
```

Windowed Output:

```
{ "window": "5.0-8.0", "phi": 9.183, "window_size": 3 }
```

Each frame or window can be directly plotted as a $\Phi(t)$ curve or used for comparative analysis.

7. System Integration

- **Feeds CRVU:**
  - TPS = $\sigma(\Phi)$
  - NSF = weighted Φ sum
- **Feeds RFMU:** Used to generate field intensity
- **Feeds C³:** As raw resonance potential data for affective modeling

III.3 – Resonance Field Mapping (RFM)

The RFM function generates a scalar field over the frequency domain at each time point, representing the density and distribution of harmonic energy. It transforms a list of discrete partials into a smooth, continuous resonance map — providing a foundation for spatially-aware tonal reasoning.

Whereas Φ quantifies total harmonic coherence within a frame, RFM visualizes how that resonance is distributed across the pitch spectrum — forming a field of tonal gravity.

1. Core Equation

The resonance field at time t, over frequency coordinate f, is computed as:

$$\text{RFM}(f,t) = \sum_{i=1}^{N} A_i(t) \cdot e^{-\frac{(f-f_i(t))^2}{2\sigma^2}}$$

Where:

- f: continuous frequency grid point
- $f_i(t)$: frequency of i-th partial at time t
- $A_i(t)$: amplitude of i-th partial
- σ: spread parameter (resonance width)

This is a Gaussian kernel density estimator, where each partial casts a resonance hill over frequency space.

2. Grid Design

To build RFM numerically, a discrete frequency grid is defined:

$$F = \{f_1, f_2, ..., f_n\} = \text{logspace}(20, 20000, N)$$

- $N = 512$ or 1024 (typical values)
- Logarithmic scaling reflects cochlear frequency mapping

Each grid point will hold one RFM(f,t) value.
Result: a 2D matrix where each row = time slice, each column = frequency bin.

3. Perceptual Interpretation

RFM approximates the perceptual landscape of sound:

- **Peaks** = tonal centers or attractors
- **Valleys** = spectral gaps or anti-resonance zones
- **Slope** = tonal pull
- **Width** = harmonic spread

Musically, RFM enables analysis of:

- Tonal convergence and divergence
- Modulation zones (shifting attractors)
- Multi-center textures (polytonality)
- Voice-leading through field movement

4. Gradient Operator (∇RFM)

To extract perceptual "direction," RFM computes its gradient:

$$\nabla\text{RFM}(f,t) = \frac{\partial\text{RFM}(f,t)}{\partial f}$$

- This is discretized via finite differences on the grid
- High ∇RFM → rapid spectral change → dissonance, instability
- Low ∇RFM → smooth flow → stability, fusion

Gradient magnitude is used in CRVU's TFI metric.

5. Peak Tracking & Field Topology

Local maxima in RFM indicate momentary tonal centers.

Let:

$$f_p(t) \in F \quad \text{where } \nabla\text{RFM} = 0, \quad \text{curvature} < 0$$

Tracking $f_p(t)$ over time forms a tonal trajectory or attractor path.

- Field segmentation methods (e.g., watershed or ridge detection) can be applied for higher-level grouping

6. Visualization Mapping

- X-axis = time (0–20s)
- Y-axis = log frequency (20–20kHz)
- Color = field intensity (resonance strength)
- Overlay = vector arrows from ∇RFM or contour lines for attractors

This map becomes the visual body of tonal behavior over time.

7. Output Format

```
{
  "time": 3.7,
  "grid": [20.0, 24.3, ..., 20000.0],
  "field": [0.001, 0.005, ..., 0.0],
  "gradient": [0.002, -0.001, ..., -0.003]
}
```

Each frame produces a scalar field vector and optional gradient vector.

8. Theoretical Parallels

RFM draws from:

- Spectrogram theory: smoothed representation of energy over time/frequency
- Field theory (physics): scalar potential fields
- Tonnetz spaces: extended to real-valued, log-frequency domains
- Auditory cortex modeling: tonotopic fields + lateral inhibition

9. Applications

- CRVU → TFI: average ∇RFM magnitude
- Modulation analysis: movement of peaks
- Polycentricity: multi-peak stability across frames
- VR/Unity visual grounding: resonance fields as terrain surfaces

III.4 – Cognitive Resonance Metrics (TPS, TFI, NSF)

Cognitive perception of harmony is not based on static symbols, but on dynamic acoustic behavior: how stable, unified, and memorable a sound feels over time. The CRVU summarizes this behavior through three scalar metrics: Temporal Perceptual Stability (TPS), Tonal Fusion Index (TFI), and Neural Synchronization Field (NSF).

These metrics operate as projections of resonance into perceptual space — each compressing a dimension of resonance behavior into a scalar value $\in [0, 1]$.

1. Temporal Perceptual Stability (TPS)

Equation:

$$\text{TPS} = \frac{1}{1 + \sigma_\Phi(t)}$$

Where:

- $\Phi(t)$: framewise resonance potential
- $\sigma_\Phi(t)$: standard deviation across full or local time window

Interpretation:

- High TPS ⇒ consistent Φ ⇒ stable resonance center
- Low TPS ⇒ fluctuating Φ ⇒ modulation, instability

Perceptual Basis: Temporal regularity correlates with attentional focus and pitch certainty (cf. Leman, 2016). TPS models tonal anchoring as experienced in both classical and non-tonal contexts.

2. Tonal Fusion Index (TFI)

Equation:

$$\text{TFI} = \frac{1}{1 + \langle |\nabla\text{RFM}(f, t)| \rangle}$$

Where:

- $\nabla\text{RFM}(f,t)$: gradient of the resonance field at time t
- $\langle\cdot\rangle$: average over frequency domain and time frames

Interpretation:

- High TFI ⇒ smooth field ⇒ tight spectral coherence
- Low TFI ⇒ jagged field ⇒ spectral diffusion

Neural Correlates: Auditory cortex entrains more strongly to spectrally fused sounds. Fusion models correlate with gamma coherence, phase-locking, and sound object formation (cf. Bidelman et al., 2014).

3. Neural Synchronization Field (NSF)

Equation:

$$\text{NSF} = \sum_t \Phi(t) \cdot e^{-\alpha t}$$

Where:

- α: decay coefficient (0.05–0.1 typical)

Interpretation:

- High NSF ⇒ strong early resonance ⇒ likely encoding into short-term memory
- Low NSF ⇒ delayed or inconsistent resonance ⇒ weaker impression

Psychological Basis: NSF captures the recency effect of musical perception — the brain's tendency to weight earlier salient events more heavily in expectation and evaluation processes (cf. Zatorre & Salimpoor, 2013).

4. Combined Cognitive Vector

The full cognitive resonance signature is a vector:

$$\vec{\text{CRV}} = [\text{TPS}, \text{TFI}, \text{NSF}]$$

Each value is:

- Normalized $\in [0,1]$
- Interpretable individually
- Composable into weighted salience models

5. Applications

- **C³ input:** feeds attention allocation, memory modeling, immersion scores
- **Real-time resonance diagnosis**
- **Musical segmentation:** changes in CRV may mark structural transitions
- **Generative AI:** use CRV to guide harmonic generation toward cognitive targets

6. Output Summary

JSON format:

```
{
  "TPS": 0.842,
  "TFI": 0.713,
  "NSF": 0.0362
}
```

Visualized as three stacked bars (R/G/B), overlaid on waveform or resonance map.

Summary

CRV is the cognitive endpoint of R³: a compact, interpretable summary of how a given sound structure will likely be experienced, memorized, and evaluated by a human listener.

IV – Data Structure and Output (Enhanced)

The structural integrity of R³ depends not only on its theoretical formulations, but also on the consistency, extensibility, and interpretability of its data output formats. All R³ units generate machine-readable, human-interpretable, and visualization-ready files. These files follow a strict temporal alignment and a modular format architecture.

IV.1 – Frame Resolution and Time Structure

Parameter	Value
Total Duration	20.0 seconds
Frame Rate	0.1 seconds
Total Frames	200
Fundamental + Harmonics	1 + 16

Each frame is timestamped and encapsulates a complete harmonic snapshot.

Example: `RawSpectrum-unit.json`

```
{
  "time": 3.2,
  "partials": [
    {
      "freq": 261.63,
      "amplitude": 0.81,
      "isFundamental": true,
      "harmonic_index": 0,
      "symbol": "C4"
    },
    {
      "freq": 523.25,
      "amplitude": 0.42,
      "harmonic_index": 1,
      "isFundamental": false,
      "symbol": "C5"
    }
  ]
}
```

This format is unit-agnostic and powers all R^3 modules.

IV.2 – Unit-Specific JSON Output Formats

1. PRU – Phantom Root

```
{
  "time_range": [2.1, 4.3],
  "phantom_root": 130.81,
  "symbol": "C3¹",
  "group": "A2",
  "fusion_score": 0.42
}
```

One record per detected PR segment, with group-matched harmonic stack label and symbolic pitch.

2. RPU – Resonance Potential Framewise:

```
{ "time": 5.2, "phi": 3.714 }
```

Windowed:

```
{ "window": "5.0-8.0", "phi": 9.23, "window_size": 3 }
```

Both datasets can be plotted as continuous Φ curves, with optional variance indicators.

3. RFMU – Resonance Field

```
{
  "time": 7.1,
  "grid": [20.0, 24.1, ..., 20000.0],
  "field": [0.0012, 0.0044, ..., 0.0],
  "gradient": [0.0004, -0.0003, ..., -0.0021]
}
```

- `field` = scalar intensity at each frequency point
- `gradient` = ∇RFM used for TFI

4. CRVU – Cognitive Resonance Vector

```
{
  "TPS": 0.843,
  "TFI": 0.702,
  "NSF": 0.0361
}
```

Single vector summarizing entire input segment's resonance dynamics.

IV.3 – Output File Naming Conventions

Type	Pattern
Raw input	`RawSpectrum-unit.json`
PR segment	`PR-unit-temporal.json`
RP framewise	`RP-framewise.json`
RP windowed	`RP-windowed.json`
Field maps	`RFM-unit.json`
Cognitive	`CRV-unit.json`

All files are written to `../data/output/<unit>/` subdirectories, with script-driven generation.

IV.4 – Visual Outputs

Unit	File	Format	Dimensions
PRU	`PR-unit.png`	PNG	3840 × 216
RPU	`RP-unit.png`	PNG	3840 × 216
RFMU	`RFM-unit.png`	PNG	3840 × 216
CRVU	`CRV-unit.png`	PNG	3840 × 216
Master	`R3-overlay.html`	HTML	3840 × 2160

Visuals use:

- Plotly for HTML interactive
- Matplotlib for static export
- Log-scale y-axis for frequency mapping
- Dark mode with frequency-hue colorization

IV.5 – Unity-Compatible CSV Format

```
time,freq,amplitude,isFundamental,harmonic_index,
```

Used in:

- `CSVLoader.cs` to populate `List<Partial>`
- `SpectrumVisualizer.cs` to map x (time), y (log freq), size (amplitude)
- Visualized in 3D scene using prefabs, color shaders, and optional PR overlays

IV.6 – Extension Paths

- OL-unit output (locking events)
- Symbolic export to MusicXML (planned)
- Annotated resonance flows for interactive learning

V — Pipeline Execution and Automation (Enhanced)

The R³ module is implemented as a fully modular, automatable pipeline. Its entire analytical process — from raw spectral input to visual output and Unity export — can be executed via a single orchestration script. This design ensures reproducibility, clarity, and efficient development.

V.1 — Execution Logic

All scripts in R³ are written in Python and follow a unit-modular standard:

- Each unit has:
  - One analysis script → produces `.json`
  - One visualization script → produces `.png`
- All units share a common input file: `RawSpectrum-unit.json`

This architecture supports both:

- Independent execution (for testing/debugging)
- Sequential batch runs (via automation script)

V.2 — Master Script: `run_R3_pipeline.py`

This script executes the full R³ pipeline in order:

Order of Execution:

```
[
  "PR_unit_temporal.py",
  "RP_unit_combined.py",
  "RFM_unit_analysis.py",
  "CRV_unit_analysis.py",
  "visualize_PR_temporal.py",
  "visualize_RP_unit.py",
  "visualize_RFM_unit.py",
  "visualize_CRV_unit.py",
  "visualize_overlay_all.py"
]
```

Each entry is executed via:

```
subprocess.run(["python", script], check=True)
```

If any step fails, the pipeline halts — ensuring fail-fast validation.

V.3 — Execution Environment

Recommended setup:

- Python 3.9+
- Libraries: `numpy`, `matplotlib`, `plotly`, `json`, `csv`, `subprocess`, `os`
- Virtual environment: `s3r3_env`
- Scripts are path-relative and designed for cross-platform compatibility (macOS, Linux, Windows)

V.4 — Unit Interdependencies

Unit	Requires	Produces
PRU	RawSpectrum	PR-unit-temporal.json
RPU	RawSpectrum	RP-framewise, RP-window
RFMU	RawSpectrum	RFM-unit.json
CRVU	RPU + RFMU outputs	CRV-unit.json

All outputs are time-aligned (0.1s resolution) and normalized where needed.

V.5 — Directory Layout

Scripts:

```
/scripts/
 PR_unit_temporal.py
 RP_unit_combined.py
 RFM_unit_analysis.py
 CRV_unit_analysis.py
 visualize_*.py
 run_R3_pipeline.py
```

Data:

```
/data/
 output/
     PR/
     RP/
     RFM/
     CRV/
     raw/RawSpectrum-unit.json
```

Output:

```
/output/
 *.png
 R3-overlay.html
```

V.6 — Execution Time

Unit	Avg Analysis Time (200 frames)	Visualization Time
PRU	∼2 seconds	∼1.5 seconds
RPU	∼4 seconds	∼2 seconds
RFMU	∼5 seconds	∼3 seconds
CRVU	∼1 second	∼1 second
Overlay	—	∼3–5 seconds

Total pipeline runtime: $<$ 20 seconds

V.7 — Logging and Debugging

Each script includes:

- `print("[UNIT] Starting...")`
- `print("[UNIT] Finished. Output saved to: ...")`
- `try/except` wrappers with error logging
- Optional: logging to file (`log.txt`), runtime profiling, unit test suites (planned)

V.8 — Real-Time Pipeline (Future)

Goals:

- Hook into live CREPE output stream
- Frame-by-frame analysis and accumulation
- Unity/VR feedback loop using CRV in real-time

This will require conversion of R³ modules to stream-safe, low-latency processes (e.g., via NumPy Live, C++, or Python async/generator pattern).

VI — Visualization System (Enhanced)

Visualization is a core dimension of R³'s design philosophy. It is not merely a presentation layer, but a cognitive interface — converting dense spectral data into visually interpretable resonance structures. Each R³ unit contributes a semantically encoded visual layer aligned across a global time-frequency plane.

The goal is to present harmony not as static notation, but as a dynamic topology of vibratory interaction.

VI.1 — Design Principles

A. Shared Temporal Base

- All plots align to a common x-axis (0–20s)
- Frame resolution = 0.1s
- Windowed overlays align precisely with frame start times

B. Semantic Visual Encoding

Parameter	Visual Mapping
Frequency	Y-axis (log scale)
Amplitude	Marker size / line thickness
Φ	Y-axis (RPU layer)
Partial role	Color (e.g., fundamental = red)
Field strength	Color density (RFMU)
CRV metrics	R/G/B bar mapping

C. Vertical Modularity

- Each unit occupies 216px vertical space
- RawSpectrum layer = 1080px (reference base)
- Total image = 3840 × 2160 (4K full overlay)

VI.2 — Unit Layer Visuals

1. RawSpectrum (S³)

- **Markers:** square or circle
- **Color:** frequency class (HSV or pitch-mapped palette)
- **Size:** amplitude
- **Opacity:** harmonic index scaled
- **Y-axis:** log(frequency)
- **Z (optional):** time slice index for animation or Unity rendering
- **Renderer:** Plotly's `Scattergl()` for high-speed rendering

2. Phantom Root (PRU)

- **Form:** red horizontal bars
- **Y-position:** PR frequency
- **Label:** symbolic pitch (e.g., C3²)
- **Time span:** width of perceptual root duration
- **Group code:** can be color-coded (A, A1, A2, . . .)

3. Resonance Potential (RPU)

- **Framewise Φ:** thin gray line (baseline)
- **Windowed Φ:** colored overlays:
  - 1s = red
  - 3s = orange
  - 5s = green
  - 7s = blue
- **Y-axis:** Φ value (scalar resonance density)
- **X-axis:** time

4. Resonance Field (RFMU)

- **Form:** 2D heatmap
- **x =** time
- **y =** frequency (log)
- **color =** RFM(f, t) field strength (e.g., inferno, magma)
- **Optional overlays:**
  - white peak paths
  - vector arrows (∇RFM)
  - field contour lines

5. Cognitive Vector (CRVU)

- **Form:** stacked bars
- **Red =** TPS
- **Green =** TFI
- **Blue =** NSF
- **Labels:** numeric values (0.000–1.000)
- **Y-axis:** not used (bar only)

This layer acts as the summary strip, linking resonance data to perceptual metrics.

VI.3 — Master Overlay Composition

Final full overlay is generated using `visualize_overlay_all.py`. It combines:

- 5 unit layers (216 px each)
- 1 RawSpectrum base layer (1080 px)
- Common time axis
- Global dark mode for color clarity
- **HTML output:** interactive, 4K resolution

VI.4 — Static Exports

Each unit also produces a `.png` file:

Unit	File Name	Resolution
PRU	PR-unit.png	3840 × 216 px
RPU	RP-unit.png	3840 × 216 px
RFMU	RFM-unit.png	3840 × 216 px
CRVU	CRV-unit.png	3840 × 216 px
Master	R3-overlay.html	3840 × 2160 px

These exports allow both modular inspection and publication-level usage.

VI.5 — Unity Integration

Visual layers are linked to Unity via:

- Prefab scaling (amplitude)
- Z-depth encoding (harmonic index)
- Dynamic camera tracking of PRU or RFM peaks
- CRV bar overlays as HUDs in 3D scenes

Optional enhancements:

- PR trail = `LineRenderer` path
- RFM = surface terrain with Φ-based displacement
- CRV = color modulation of environment

VI.6 — Aesthetic Philosophy

R³ visual outputs aim to:

- Replace static notation with spectral cartography
- Encode mathematical depth in intuitive visuals
- Make resonance not only computable — but seeable

VII — Unity Integration (Enhanced)

The Unity integration of R³ transforms resonance data from abstract mathematical structures into a spatial, interactive, and visual environment. This enables researchers, musicians, and users to walk through, see, and interact with spectral and harmonic structures — making resonance literally visible.

Unity is used not just as a renderer, but as a cognitive translation platform: it visualizes how frequencies resonate, how roots shift, how fields flow — in real time or through immersive playback.

VII.1 — Export Path: JSON → CSV

Although R³ internally uses `.json` for maximum flexibility, Unity consumes data as `.csv` via its lightweight, line-based loading mechanisms.

Source: `RawSpectrum-unit.json` → converted to:

CSV Format: `RawSpectrum01.csv`

```
time,freq,amplitude,isFundamental,harmonic_index,symbol
1.1,196.0,0.82,True,0,G3
1.1,392.0,0.42,False,1,G4
...
```

Each line represents one partial (including harmonics), with symbolic encoding for microtonal interpretation.

VII.2 — Unity C# Class Structure

1. Partial.cs

```
[System.Serializable]
public class Partial {
    public float time;
    public float freq;
    public float amplitude;
    public bool isFundamental;
    public int harmonic_index;
    public string symbol;
}
```

2. CSVLoader.cs

VII.3 — Scene Mapping

Dimension	Mapped To
X	time (horizontal progression)
Y	log(freq) (vertical placement)
Z	harmonic index (depth, optional)
Size	amplitude (object scale)
Color	frequency (HSV hue-based mapping)

Each partial = a colored glowing sphere.

- Stronger harmonics = larger/brighter objects
- Fundamental = red core; others vary by frequency

VII.4 — Object Structure and Prefabs

Prefab: `PointPrefab` (Sphere with Unlit Shader)

Renderer:

- Emission Color = mapped hue
- Scale = amplitude × scalar
- Tag = Fundamental / Harmonic

Optional shader features:

- Pulse = temporal dynamics
- Glow = amplitude modulation
- Flicker = instability (if Φ is low)

VII.5 — Temporal Animation

- Unity's `Time.time` aligns playback with partial spawning
- Optional: timeline scrubber
- Scene camera can track:
  - PR path (via `LineRenderer`)
  - Field peak in RFM (via surface mesh)

VII.6 — Extended Visualizations

Data	Visual Form	Mechanism
PRU	Red line sweep	LineRenderer along PR freq
RPU	Height curve	Dynamic plot (Φ over time)
RFMU	Mesh surface	Terrain object from `field[]`
CRVU	HUD bar graph	UI Panel with TPS, TFI, NSF

Additional interaction options:

- Filter by group (A2, A3, ...)
- Highlight tonal drift zones
- Switch between symbolic and spectral views

VII.7 — Sound Integration

- Link Unity's `AudioSource.time` to visual spawning
- Synchronize resonance events with real sound
- Use amplitude thresholding to trim non-audible points
- Optional: real-time Φ modulator → dynamically warp terrain or brightness

VII.8 — Performance Optimization

- Object pooling (for partials)
- Async CSV loading (for large datasets)
- GPU instancing for visual particles
- Log-space Y-axis prevents vertical crowding

VII.9 — Cognitive Immersion Use Case

The Unity implementation allows users to:

- Step through harmonic space
- See tonality emerge and dissolve
- Hear resonance while seeing its structure
- Manipulate partials and watch CRV change live

Use cases include:

- Education (teaching tonal centers)
- VR concert staging
- Interactive composition
- Research on tonotopic attention in motion

VIII — Cursor Architecture Placement (Enhanced)

The Cursor AI platform serves as the interactive, explorable knowledge interface of the SRC system. All R^3 content — scientific explanations, equations, visualizations, and output samples — are embedded within Cursor's domain–unit–node hierarchy, providing a seamless gateway between theory, data, and cognitive navigation.

VIII.1 — Domain-Level Placement

R^3 exists as a dedicated modular domain within the Cursor site structure:

- **Path:** `/modules/r3`
- **Title:** Resonance-Based Relational Reasoning
- **Function:** Gateway page introducing the theory, architecture, and units of R^3

Domain Page Contents:

- Scientific overview
- Mathematical core (Φ, PR, RFM, CRV equations)
- $S^3 \to R^3 \to C^3$ flowchart
- Unit summary table
- Domain toggle menu (vs. Traditional-Based Theory)

Users can explore individual units by clicking cards linking to their respective pages.

VIII.2 — Unit Page Design

Each R^3 unit (PRU, RPU, RFMU, CRVU) has a standalone interactive document:

Unit Code	Path	Title
PRU	`/modules/r3/pr`	Phantom Root Unit
RPU	`/modules/r3/rp`	Resonance Potential Unit (Φ)
RFMU	`/modules/r3/rfm`	Resonance Field Modeling Uni
CRVU	`/modules/r3/crv`	Cognitive Resonance Vectoring

Each unit page includes:

- Scientific Function
- Mathematical Foundation (LaTeX supported)
- Node Architecture Table
- Sample Output (JSON snippet)
- Visualization Preview (216px PNG)
- Integration pathways (to C^3 or back to S^3)

VIII.3 — Node-Level Embedding

Each unit page has expandable `<details>` components for its node definitions.

Example in PRU:

```
<details><summary>GroupMatcher</summary>
Matches sequences of stable f segments to harmoni
</details>
```

This allows deep structure without visual clutter.

Nodes are cross-linkable and potentially host their own `/nodes/<id>` pages in future iterations.

VIII.4 — Visualization Integration

Every unit's visualization is embedded via:

- Inline PNG
- Collapsible `<details>` blocks
- Optional Plotly iframe (HTML interactive graphs)

Layer	Visual Type	Location
PRU	bar + label plot	`PR-unit.png`
RPU	Φ overlay curves	`RP-unit.png`
RFMU	Heatmap grid	`RFM-unit.png`
CRVU	RGB bars	`CRV-unit.png`

The full overlay (`R3-overlay.html`) may be shown in a dedicated interactive gallery.

VIII.5 — Intermodule Linking

Each unit page includes a reference sidebar linking:

- **S³ input:** `RawSpectrum`
- **R³ peers:** e.g., PRU links to RPU
- **C³ outputs:** CRVU → CTU, NSU, PIU

Additionally, source references are hyperlinked inline (e.g., Zatorre et al., 2013).

VIII.6 — Domain Switch Architecture

The dual-domain system is shown via a toggle interface:

```
[  R³ Resonance Theory ] [  Traditional Theory ]
```

Currently, Traditional Theory domain is empty — shown as inactive but present.

Future units may populate this view for contrastive analysis.

VIII.7 — Embedded Code + Output Previews

Each unit page includes:

- JSON sample snippets
- Direct download link (`.json`)
- Code preview block (e.g., Python `calculate_phi()`)

Cursor supports syntax-highlighted code and LaTeX-based equations in parallel.

VIII.8 — Educational & Research Utility

Cursor's R³ structure enables:

- Progressive disclosure (unit → node → formula)
- Citation-based expansion
- Interactive concept comparison
- Cross-disciplinary accessibility

Users can enter from abstract, scroll into algorithm, and emerge with conceptual clarity.

IX — Open Questions & Future Work (Enhanced)

R³ presents a robust, fully functional resonance analysis framework. Yet, like any scientific system, it operates within a set of defined constraints and assumptions. As the system matures, both its epistemic foundation and computational scope invite further exploration.

The following questions and proposed future extensions represent frontiers, not failures — theoretical edges where new forms of resonance reasoning, perceptual modeling, and interactivity may emerge.

IX.1 — Symbolic–Resonant Integration

Problem: There is currently no canonical mapping from symbolic harmony (e.g., "C major", "G7") to resonance field structures.

Open Question: Can classical chord symbols be translated into predictable RFM patterns or CRV signatures?

Research Path:

- Construct resonance fingerprints for chord classes
- Reverse-map RFM peaks to symbolic root-inversion labels
- Apply symbolic labeling to R³ field outputs for hybrid navigation

Goal: Enable bidirectional harmony interpretation — symbolic ↔ resonance

IX.2 — Real-Time Processing Capabilities

Problem: R³ is currently batch-processed from fixed input (offline mode)

Open Question: Can PRU, RPU, RFMU, and CRVU operate on live streamed data at frame-rate (0.1s or faster)?

Engineering Path:

- Use async generators or NumPy Live for buffer streaming
- Convert `RawSpectrum-unit.json` generation into audio listener → CREPE → partial emitter → unit executor
- Port core algorithms (Φ, RFM) to C++/CUDA for low-latency computation

Goal: Enable live visualization, performance-driven analysis, and generative harmony via real-time feedback loops.

IX.3 — Polyphonic PR Detection

Problem: Current PRU logic assumes monophonic fundamental tracking

Open Question: Can PRU extract multiple phantom roots simultaneously — modeling polycentric tonality?

Research Path:

- Implement time-overlapping root group tracking
- Use spectral clustering to separate multiple root flows
- Model each root's gravitational zone in RFM separately

Goal: Capture layered tonality and its interaction in complex textures.

IX.4 — Multi-Listener Resonance Modeling

Problem: R³ is designed for generalized perceptual inference — not individual neural profiles

Open Question: Can CRVU metrics be personalized based on neural data (e.g., EEG), musical background, or auditory profile?

Path:

- Collect listener-specific FFR or ERP responses to stimulus sets
- Train models to predict TPS/TFI/NSF weightings per profile
- Tune resonance field weighting dynamically in response to engagement metrics

Goal: Model resonance-perception diversity, and adapt analysis per listener.

IX.5 — Prime-Limit Resonance Navigation

Problem: Field modeling currently operates in linear frequency space

Open Question: What happens when RFM is computed in prime-vector space (e.g., 5-limit, 11-limit)?

Mathematical Path:

- Encode partials as vectors $\vec{v} = (x_2, x_3, x_5, \ldots)$
- Define RFM in log-geometry of prime-lattice
- Extend ∇RFM to multi-axis slope computation

Goal: Enable symbolically grounded resonance maps with real-number continuity

IX.6 — Resonance-Centric Composition Systems

Problem: Current generative AI models are melody/chord/beat based

Open Question: Can an AI compose music guided purely by RFM field shape and CRV evolution?

Creative Path:

- Define target RFM → search partials to generate matching field
- Use Φ targets to constrain harmonic grammar
- Tune CRV vector toward affective intent (e.g., high TPS → stability)

Goal: Create music from resonance, not just producing resonance from music.

IX.7 — Philosophical & Epistemological Frontiers

R³ challenges centuries-old assumptions about musical structure:

- That tonality is symbolic
- That function is rule-based
- That perception is discrete

But if resonance is continuous, embodied, and cognitive, then:

- What is a "note"?
- Where is the boundary between sound and structure?
- Can harmony exist without symbols — only through flow?

These are not technical questions — they are conceptual invitations.

X — Appendix & References (Enhanced)

This section consolidates all formal references, system definitions, microtonal encodings, data format standards, and mathematical mappings used throughout the R³ module. It serves as both a technical appendix and a citation-ready bibliography for researchers, developers, and composers.

X.1 — Scientific References

A. Psychoacoustics & Perception

- Terhardt, E. (1974). "Pitch, consonance, and harmony." *JASA*

- Plomp, R. & Levelt, W. (1965). "Tonal consonance and critical bandwidth."
- Bregman, A. (1990). *Auditory Scene Analysis*
- Bidelman, G.M. et al. (2011). "Brainstem pitch encoding."
- Zatorre, R. & Salimpoor, V. (2013). "Prediction and reward in music." *Nat Rev Neurosci*
- Moore, B. (2012). *An Introduction to the Psychology of Hearing*

B. Harmony & Spectral Theory

- Sethares, W. (1998). *Tuning, Timbre, Spectrum, Scale*
- Tymoczko, D. (2006). "The Geometry of Musical Chords." *Science*
- Doty, D. (2002). *The Just Intonation Primer*
- Huron, D. (2006). *Sweet Anticipation*
- Parncutt, R. (1989). *Harmony: A Psychoacoustical Approach*
- Roederer, J.G. (2008). *The Physics and Psychophysics of Music*

C. Mathematics & Signal Processing

- Shannon, C.E. (1948). "A Mathematical Theory of Communication."
- Mallat, S. (2009). *A Wavelet Tour of Signal Processing*
- Sethares, W. (2005). "Spectral Convergence and Dissonance." *Computer Music Journal*

X.2 — Symbolic Pitch Encoding System

The R³ module uses a symbolic pitch notation system that encodes:

- Pitch class
- Octave number
- Microtonal deviation in cents (rounded to ±25c steps)

Format Example:

```
G2¹  →  Pitch: G, Octave: 2, +25 cents deviation
C4²  →  Pitch: C, Octave: 4, -50 cents deviation
```

Unicode Symbols:

- = no deviation
- ¹, ², ¹, ² = ±25c, ±50c, etc.

This system aligns with symbolic notation while reflecting real spectral deviations.

X.3 — Data Format Reference

Unit	File Name	Descri
PRU	`PR-unit-temporal.json`	Phanto
RPU	`RP-framewise.json`, `RP-windowed.json`	Φ metr
RFMU	`RFM-unit.json`	Resona
CRVU	`CRV-unit.json`	Cogniti

All files are time-aligned (0.1s resolution), normalized, and UTF-8 encoded.

X.4 — CSV Export Format for Unity

```
time,freq,amplitude,isFundamental,harmonic_index,
```

Used in `CSVLoader.cs`, `SpectrumVisualizer.cs` — mapped to Unity object parameters for real-time 3D rendering.

X.5 — Coordinate Mapping Schema

Parameter	Mapped Dimension	Usage
time	X-axis	horizontal flow
log(freq)	Y-axis	vertical tonal position
amplitude	Scale	object size / brightness
harmonic_index	Z-axis (optional)	depth layering
symbol	UI label	displayed on HUDs

X.6 — Node & Element Glossary

Node	Unit	Function
CentTracker	PRU	Segment f_0 sequences by cent deviatio
GroupMatcher	PRU	Match sequences to harmonic stacks
PairwisePhi	RPU	Compute Φ across partials
WindowIntegrator	RPU	Accumulate Φ in time windows
FieldGenerator	RFMU	Create Gaussian resonance fields
GradientScanner	RFMU	Compute ∇RFM across frequency spac
TPSNode	CRVU	Variance tracker of Φ → perceptual stab
TFINode	CRVU	Field slope magnitude → tonal fusion
NSFNode	CRVU	Memory-weighted Φ integral → neural

X.7 — Terminology Standardization

- **Resonance Potential (Φ):** scalar harmonic coherence
- **Phantom Root:** inferred tonal anchor
- **RFM Field:** resonance topography
- **CRV Vector:** perceptual signature

X.8 — Licensing and Distribution

All R³ code and structure is:

- Open source under MIT license (default)
- Freely distributable for research, educational, and creative use
- Citable with proper attribution: *"R³ module of the SRC system (2025)"*

Final Statement

R³ unites mathematical rigor, perceptual truth, and computational clarity into a single resonance engine. This appendix stands as the foundation for collaborative development, academic referencing, and future expansion.

C³ MASTER TECHNICAL REPORT (ENHANCED)

I. INTRODUCTION

I.1. Conceptual Framework

- What is C³? What is cognitive resonance?
- The position of C³ within the music–mind–neurophysiology triad
- The role of C³ in SRC: S³–R³–C³ triple integration

I.2. History and Motivation

- Interdisciplinary disconnection: why was a system like C³ necessary?
- Fragmented approaches across psychoacoustics, EEG/fMRI, and cognitive theory
- The SRC vision: reconstructing the part–whole relationship

II. THEORETICAL FOUNDATION

II.1. Theory of Cognitive Resonance

- What is sensory–cognitive resonance?
- Concepts of oscillation–synchronization–network integration
- Temporal Perceptual Stability (TPS), Tonal Fusion Index (TFI), Neural Synchronization Fields (NSF)

II.2. Mathematical Modeling

Primary Equation:

$$C^3(t) = \sum_{i=1}^{9} w_i \cdot \text{Unit}_i(t)$$

Layer Expansion:

$$\text{Unit}_i(t) = \sum_{j=1}^{N} w_{ij} \cdot \text{Node}_j(t)$$

- Normalized resonance computations
- Temporal resolution
- Weight coefficients

III. MODULAR ARCHITECTURE (UNIT > NODE > ELEMENT)

III.1. UNIT Definitions

Each defined in a separate section:

- CTU
- AOU
- IEU
- SRU
- SAU
- PIU
- IRU
- NSU
- RSU

III.2. NODE–ELEMENT Structure

- Nodes = Cognitive functions
- Elements = Measurable EEG/fMRI outputs

Each NODE includes:

- Conceptual definition
- Literature reference
- GlassBrainMap coordinate

IV. GLASSBRAINMAP INTEGRATION

IV.1. Coordinate System

- Each NODE assigned a 3D spherical region in MNI space
- Anatomical localization (e.g., ACC, SMA, HG)

IV.2. Mapping + Web Integration

- SVG layer
- React/D3.js interface
- Hover–click interaction
- Tooltip + page linkage

V. SYSTEM DYNAMICS

V.1. Time-Based Activations

- How is the overall output of C^3 computed over time?
- Frame calculations via EEG resolution

V.2. Feedback Loop: C^3 R^3 S^3

- Harmonic Distance and values from R^3
- Spectral profile data from S^3
- Feedback from C^3: attentional dispersion, valence mapping

VI. RESEARCH AND APPLICATION DOMAINS

VI.1. Music Therapy

- Effects of different UNITs in therapeutic settings
- Trauma-informed listening through IRU and PIU

VI.2. Compositional Tools

- Sound selection guided by C^3
- Listener direction through CTU + AOU interaction

VI.3. Neuropedagogy

- Attention enhancement in children (IEU + SAU)
- Resonance density in learning via RSU

VII. FUTURE ROADMAP

VII.1. Real-Time Integration

- Unity/WebGL + OSC/WebSocket connectivity

VII.2. VR/AR and Cinematic Systems

- GlassBrainVR
- Resonance–narrative integration

VIII. CONCLUSION AND SCIENTIFIC CONTRIBUTIONS

- Gaps in existing systems
- Innovative solutions offered by C^3
- The irreplaceable role of C^3 within SRC

I.1 Conceptual Framework

1.1.1 What is C^3?

C^3 – Cognitive Consonance Circuit is a neurophysiologically grounded, mathematically formulated cognitive modeling engine that analyzes the **emotional**, **cognitive**, **motor**, and **social resonance processes** evoked by music in the human brain as a time-dependent, multilayered system.

The main goal of this system is to measure, classify, and represent the **temporal resonance responses** in a music listener's brain using data such as:

- EEG (alpha, beta, gamma, MMN, P300),
- MEG (oscillatory phase-locking),
- fMRI (region-specific activations)

C^3 operates through **9 independent yet integrated Units**. Each UNIT represents a specific cognitive function. Internally, these Units are composed of **NODEs** (functional mechanisms) and **ELEMENTs** (measurable neural outputs). In this way, the system creates a fully connected network between music and the brain.

1.1.2 The Role of C^3 Within SRC

C^3 is one of the three main modules of the **SRC system**:

- **S^3 – Spectral Sound Space**: Physical–acoustic analysis of sound

- **R^3 – Resonance-Based Relational Reasoning**: Mathematical modeling of tonal, microtonal, and harmonic structures

- **C^3 – Cognitive Consonance Circuit**: Representation of musical structures' impact on the brain through time-based neurocognitive models

There is **bidirectional data flow** between these modules. For example:

- Spectral data from S^3 (e.g., noise intensity, tonal center frequency) → enters R^3 for computations of (resonance potential) and HD (harmonic distance) → these values are then transmitted to C^3 to generate cognitive load or emotional response in modules like CTU and AOU.

C^3 also sends its results as **feedback** to R^3 (harmonic resonance suggestions) and to S^3 (perceptual spectrum map filtering). This structure defines the system's **adaptive feedback** nature.

1.1.3 Core System Principles

1.1.3.a Definition of Cognitive Resonance

In the C^3 system, cognitive resonance is defined as the **simultaneous activation** of processes such as **frequency–time alignment**, **sensory–conceptual integration**, **motor synchronization**, and **emotional response** in the brain during music listening. This resonance is studied across the following layers:

- **Electrophysiological:** EEG data (phase locking, oscillation power, ERP components)

- **Functional Imaging:** fMRI BOLD increase, regional localization in MNI coordinates

- **Time-Resolved Modeling:** Moment-to-moment alignment of music events with brain responses

1.1.4 Structure of C^3: UNIT → NODE → ELEMENT

- **UNIT**: Conceptual framework (e.g., CTU – Cognitive Tension Unit)

- **NODE**: Specific functional structure (e.g., Harmonic Dissonance Node)

- **ELEMENT**: Measurable EEG/fMRI output (e.g., alpha–beta phase locking)

This structure ensures that each UNIT consists of layers that are **scientifically**, **experimentally**, and **theoretically connected**. Thus, each NODE becomes **conceptually defined**, **empirically measurable**, and **anatomically localizable**.

1.1.5 Anatomical Mapping: GlassBrainMap Integration

Each NODE in C^3 is assigned a brain region (ROI – Region of Interest). These regions are defined as spherical volumes using **MNI coordinates**.

Node	MNI Coordinates	Region
AOU – HGAINT 01	`[-60, -28, +6]`	Left posterior STG
IEU – HARMONIC 01	`[0, +50, +20]`	Fronto-central EEG
CTU – RESOLVED 05	`[+64, -22, +4]`	Anterior HG

These coordinates are fully aligned with the data in `C__BrainMap____83_Koordinatl_k_Tam_Tablo` and **GlassBrainMap.pdf**.

On the web platform, this information is connected to a visual map through the SVG + React component `GlassBrain.jsx`.

1.1.6 Literature Foundation

As of now, a total of **61 scientific articles** have been integrated into the C^3 system. Each NODE is supported by at least one experimental study. These include:

- EEG/MEG-based ERP or mismatch studies

- fMRI studies (valence, arousal, pitch, syntax, entrainment)

- Multimodal meta-analyses (FFR, P300, dopamine release, MMN)

- Theoretical models (predictive coding, Wundt curve, information content models)

Conclusion (for this section)

C^3 forms the **neurocognitive core** of the SRC system. It receives spectral and resonance data from modules like S^3 and R^3, interprets them through EEG/fMRI-supported models in the human brain, and outputs results that allow musical structures to be measured at the level of **meaning and impact**. It is the only system that achieves this function in a fully integrated form.

I.1 Definitional Framework

What is C^3? What does it do? How does it function within SRC?

1.1.1 What Is the Cognitive Consonance Circuit (C^3)?

The **Cognitive Consonance Circuit (C^3)** is a multidimensional, time-sensitive neural modeling system designed to measure, classify, and structurally represent the cognitive, emotional, and motor impacts of music on the human brain. C³ integrates empirical neurophysiological data—specifically EEG (alpha, beta, gamma, MMN, P300), MEG, and fMRI—to track brain responses to musical stimuli in real-time across nine cognitive domains known as **Units**.

Each Unit corresponds to a distinct cognitive subsystem (e.g., expectation, tension, memory, emotion), modeled as an independent but fully integrable node within the system. Internally, each Unit is hierarchically structured into **Nodes**, which denote abstract cognitive processes, and **Elements**, which refer to quantifiable neural signals. These signals include frequency-specific oscillatory phenomena, phase-locked responses, and region-specific BOLD activations.

1.1.2 Role of C³ Within the SRC System

C³ is the cognitive core of the **SRC (Spectral–Resonance–Cognitive)** framework, which comprises three interdependent modules:

- **S³ – Spectral Sound Space**: Performs advanced time–frequency analysis of musical signals.
- **R³ – Resonance-Based Relational Reasoning**: Models harmonic and tonal structures through vector lattices, scalar spaces, and resonance metrics.
- **C³ – Cognitive Consonance Circuit**: Converts these structural data into biologically realistic neural representations of how the brain interprets music.

These modules are not isolated silos; they form a **bidirectional data exchange pipeline**:

- Spectral information from S³ (e.g., timbral flux, pitch entropy) is used by R³ to calculate harmonic metrics such as (Resonance Potential) and HD (Harmonic Distance).
- These R³ outputs then inform C³, which uses EEG and fMRI-informed functions to simulate attention, emotional valence, arousal, and tension.
- C³, in turn, feeds back cognitive feedback into S³ and R³, enabling dynamic perceptual recalibration.

This feedback loop creates a closed analytical ecosystem that integrates perception, structure, and cognition.

1.1.3 Theoretical Principles Underpinning C³

a. Definition of Cognitive Resonance

Cognitive resonance refers to the dynamic synchronization of neural, emotional, and motor processes elicited by musical stimuli. This phenomenon is understood not as localized brain activation, but as a distributed resonance field involving:

- Oscillatory synchronization (e.g., beta/gamma coupling between SMA and auditory cortex)
- Predictive mismatch mechanisms (e.g., MMN and P300 ERP responses)
- Region-specific activation (e.g., DLPFC for tension, ACC for ambiguity, amygdala for affect)

These mechanisms interact through temporal coherence and multi-band entrainment, enabling a real-time neural "tracking" of musical structure.

b. Mathematical Formalism

The system's top-level model is defined as:

$$C^3(t) = \sum_{i=1}^{9} w_i \cdot \text{Unit}_i(t)$$

Where:

- $C^3(t)$: total cognitive resonance at time t
- $\text{Unit}_i(t)$: normalized activity of the i-th cognitive unit
- w_i: weight coefficient for each unit (empirically tunable)

Each Unit's internal model is further decomposed as:

$$\text{Unit}_i(t) = \sum_{j=1}^{N} w_{ij} \cdot \text{Node}_j(t)$$

Each Node is derived from EEG or fMRI data—either continuous oscillatory amplitudes or discrete event-related potentials—and parameterized using real-time neuroimaging standards.

1.1.4 Structural Design: UNIT → NODE → ELEMENT

The C³ system is hierarchically organized:

- **UNIT**: Defines a domain of cognitive-musical processing (e.g., tension, memory, flow).
- **NODE**: Represents a functional module within the Unit (e.g., Harmonic Dissonance).
- **ELEMENT**: A neurophysiological observable (e.g., beta-band phase locking, BOLD activation).

Each Element is associated with:

- A named brain region
- A measurement method (EEG, fMRI, or both)
- A validated reference from neuroscience literature
- An anatomical coordinate (MNI system)

This allows C^3 to link abstract musical cognition to empirical neurobiology in a deterministic way.

1.1.5 Anatomical Mapping with GlassBrainMap

C^3 integrates seamlessly with the **GlassBrainMap**, a visual representation system that places each Node/Element at anatomically and functionally relevant coordinates.

Node	MNI Coordinates	Region	Citation
AOU – HGAINT 01	`[-60, -28, +6]`	Left posterior STG	Potes et al., 2012
IEU – HARMONIC 01	`[0, +50, +20]`	Fronto-central cortex	Crespo-Bojorque et al., 2018
CTU – RESOLVED 05	`[+64, -22, +4]`	Right Anterior HG	Norman-Haignere et al., 2013

These coordinates are verified against the `C__BrainMap____83_Koordinatl_k_Tam_Tablo` document, using MNI-space localization. In the web interface, they are rendered interactively via `GlassBrain.jsx` and SVG/D3.js overlays.

1.1.6 Literature Foundation and Citation Model

The C^3 system is constructed upon **61 primary peer-reviewed studies** in the fields of neuroscience, music cognition, EEG/fMRI research, and mathematical modeling. Each Node is grounded in direct empirical evidence. Key domains include:

- MMN, P300, N2 ERP responses to pitch/syntax deviations
- Beta/gamma-band entrainment in auditory–motor synchronization
- Dopaminergic pathways during emotional peaks (Salimpoor et al., 2011)
- Functional anatomy of musical memory (Janata, 2009; Zatorre & Halpern, 2005)

All references are encoded into structured `.bib` and `.json` formats for API-level integration.

Summary

C^3 is not a passive music analysis tool. It is an **active cognitive modeling system** that transforms symbolic or spectral musical structures into meaningful, neurologically validated resonance fields. Its position within the **SRC** system ensures that structure, perception, and cognition are analyzed as a single unified continuum.

I.2 Historical Context and Motivation

Why was C^3 developed? What are the limitations of current methodologies that C^3 addresses?

1.2.1 Fragmentation Across Disciplines

Despite significant progress in music theory, neuroscience, and artificial intelligence, the academic and technological landscape surrounding music cognition remains deeply fragmented. Most systems operate in isolation:

- Spectral analysis platforms (e.g., Fourier-based spectrograms) offer detailed acoustic profiles but lack perceptual or cognitive grounding.
- Harmonic and mathematical theories (e.g., Lewin's GIS, Tonnetz models) focus on intervallic structure but often disregard empirical brain data.
- Neuroscientific research (EEG, fMRI) reveals profound insights into perception and emotion but rarely interfaces with formal music theory or real-time systems.

This disjunction has created a theoretical bottleneck. Without a shared framework, advances in one domain fail to propagate meaningfully into others. As a result, most current tools lack cross-domain explanatory power, cognitive transparency, and compositional usability.

1.2.2 Theoretical Models Lack Cognitive Validation

Traditional music-theoretical models such as Generalized Interval Systems (GIS), Harmonic Distance metrics, and Tonal Hierarchies (e.g., Krumhansl's tonal profiles) are mathematically elegant, but often fail to predict real listener responses.

Conversely, neuroscientific findings—such as:

- the MMN response to unexpected harmonies,
- the P300 component linked to rhythmic anomalies,
- or reward-related dopamine release during musical peaks (Salimpoor et al., 2011),

have rarely been integrated into formal, computationally usable structures.

There is no established mapping between music-theoretical constructs (e.g., cadence, modulation, dissonance) and neural metrics (e.g., BOLD, ERP, phase-locking). This absence of structural integration drastically reduces the predictive and pedagogical power of existing systems.

1.2.3 Existing AI Models Are Black-Box and Aesthetic-Only

While AI tools such as OpenAI's Jukebox, Google Magenta, or Amper Music have enabled impressive generative outputs, they typically operate without explanatory or cognitive modeling frameworks. They generate music without understanding what attention, memory, emotion, or structural coherence mean to a human listener.

This black-box architecture:

- Offers no feedback on listener state
- Cannot simulate emotional or cognitive pathways
- Provides no route for targeted composition or real-time feedback

C³ was conceived precisely to close this loop—not to replace such systems, but to make them cognitively explainable, analyzable, and affectively steerable.

1.2.4 Neuroscience Models Remain Theoretically Isolated

Even the most sophisticated neuroscientific work on music (e.g., Janata, 2009; Zatorre & Halpern, 2005) remains technically inaccessible to composers, theorists, or real-time music systems. The brain data is not structured in a way that can be operationalized.

For example:

- Neural entrainment in STG and SMA during rhythmic listening (Nozaradan, 2012)
- Amygdala activation during tonal familiarity or melodic recall (Koelsch, 2008)
- fMRI-detected dopaminergic release in ventral striatum (Blood & Zatorre, 2001)

—all provide powerful evidence, but without an architectural scaffold to translate them into music-analytical or compositional insight.

1.2.5 Motivation for C³: Unification Through Architecture

C³ was developed as an architectural answer to this methodological impasse.

Its core aims:

- To provide a layered system that explicitly links:
  - Symbolic musical events (e.g., harmonic surprise, tempo shift)
  - Neural mechanisms (e.g., alpha desynchronization, ERP signatures)
  - Anatomical mappings (e.g., ACC, STG, dPMC)
  - Time-based resonance metrics
- To construct a modular model (9 Units) that mirrors actual neuroscientific categorizations:
  - Tension (CTU), Expectation (IEU), Emotion (AOU), Flow (PIU), Memory (SAU), etc.
- To ensure every part of the system is:
  - Empirically grounded (with MNI coordinates, EEG/fMRI citations)
  - Mathematically formalized (via resonance models and time equations)
  - Interactively visualized (through the GlassBrainMap and unit dashboards)
- To allow integration with generative, educational, or therapeutic systems.

1.2.6 Theoretical Inspirations

C³ draws from and synthesizes:

- David Lewin's transformational music theory and the notion of "network transformations"
- Julian Hook's cross-type intervallic mappings
- Jean-Jacques Nattiez's tripartite semiotic model (poietic – neutral – esthesic)
- Zatorre & Halpern's fMRI studies on musical imagery
- Large & Snyder's work on neural resonance and attentional entrainment
- Recent computational neuroscience models such as Temporal Perceptual Stability (TPS), Tonal Fusion Index (TFI), and Neural Synchronization Fields (NSF)

These form the philosophical, neurobiological, and mathematical pillars of the C³ system.

Summary

C³ was not built to replace existing methods—it was built to unify them.

It serves as a cognitive-mathematical bridge between acoustic data, symbolic musical structures, and neurological response patterns. Its modular design, mathematical backbone, and empirical grounding position it as a unique tool in both theoretical and practical domains.

It is not merely a system. It is a new paradigm for understanding music.

II.1 Theoretical Foundations: Cognitive Resonance Theory

What is cognitive resonance? What are its neural and mathematical correlates? How is it represented in the C³ framework?

2.1.1 Definition of Cognitive Resonance

Cognitive resonance refers to the simultaneous activation and phase-alignment of neural systems—cortical, subcortical, and limbic—elicited by musical structures perceived as meaningful, surprising, emotionally salient, or motorically engaging.

It is not a static "response" but a dynamic system of oscillatory entrainment that unfolds over time in direct correlation with acoustic and symbolic musical events. In this sense, C³ does not model perception as a reaction, but rather as a temporally evolving resonance field shaped by attention, prediction, emotion, and embodied synchronization.

2.1.2 Neurophysiological Basis

Cognitive resonance is anchored in three core mechanisms:

a. Oscillatory Entrainment Neural populations synchronize their firing phases with periodic or structured stimuli in music, especially rhythm and pulse. EEG reveals:

- Beta-band phase-locking in motor regions (SMA, PMC) during pulse alignment
- Gamma coherence between auditory and frontal regions for tonal/melodic fusion
- Alpha synchrony in frontal–parietal regions indicating attentional top-down integration

b. Prediction and Mismatch Mechanisms The brain actively predicts musical continuations. Violations of these predictions result in:

- Mismatch Negativity (MMN) in STG and Fz (EEG), typically for harmonic or rhythmic deviations
- P300 ERPs for consciously detected rhythmic or temporal anomalies

These responses form the backbone of the IEU unit in C³.

c. Emotional–Limbic Activation Music evokes emotion through dopaminergic reward pathways, primarily involving:

- Ventral Striatum, Nucleus Accumbens (peak pleasure: Salimpoor et al., 2011)
- Amygdala, Insula, and ACC (valence/arousal distinction: Koelsch, 2008; Zatorre & Blood, 2001)

Together, these three systems create multiband, multisite synchronization patterns that constitute what we term **cognitive resonance**.

2.1.3 Mathematical Representation

In C³, cognitive resonance is mathematically expressed as the time-evolving summation of weighted neural activity across multiple Units:

$$C^3(t) = \sum_{i=1}^{9} w_i \cdot \text{Unit}_i(t)$$

Where:

- $\text{Unit}_i(t)$: Time-normalized resonance output of the i-th Unit (e.g., CTU, AOU)
- w_i: Tunable weight coefficient based on experimental or functional prioritization
- t: Time (sampled at EEG resolution, e.g., 100ms)

Each Unit is composed of Nodes and Elements:

$$\text{Unit}_i(t) = \sum_{j=1}^{N} w_{ij} \cdot \text{Node}_j(t)$$

Each Node is grounded in a real EEG/fMRI marker:

- **Node**: conceptual function (e.g., "Harmonic Dissonance")
- **Element**: observable signal (e.g., “α–β phase-locking in DLPFC”)

These form the structural hierarchy:

UNIT $\rightarrow$ NODE $\rightarrow$ ELEMENT $\rightarrow$ {Method, Region, Coordinates, Cita

2.1.4 Temporal Dynamics and Resolution

C³ operates in time-series frames, typically aligned to 0.1-second EEG windows. This enables high-resolution modeling of:

- Fast transitions (e.g., rhythmic inflections, expectation violations)
- Slow evolutions (e.g., emotional immersion, tonal unfolding)

Each Element generates a signal trace over time, which is then:

- Normalized to a common scale [0, 1]
- Weighted within its Node and Unit
- Aggregated into the total resonance field $C^3(t)$

Result: a spatiotemporal map of musical cognition, updated per frame, and navigable across Units, Nodes, and regions.

2.1.5 Topological Interpretation: Resonance Fields

The final output of C^3 can be interpreted as a multidimensional field evolving in time:

- Each axis corresponds to a Unit (9D space)
- Each point $C^3(t)$ is a vector of dimension 9
- Over time, the output forms a trajectory curve in this high-dimensional space

This allows:

- Trajectory clustering for musical styles
- Dynamical stability analysis (e.g., flow states vs. high-tension nodes)
- Comparative signature modeling (e.g., comparing Bach vs. Radiohead vs. AI-generated pieces)

These analytical outputs feed into applications like music therapy profiling, personalized listening models, or composition-guidance systems.

2.1.6 Link to Neuroanatomy: Mapping into GlassBrain

Each Element has a unique anatomical anchor:

- Region name (e.g., Broca, SMA, NAcc)
- MNI coordinates (from `brain_coords.json`)
- Visualization via `GlassBrain.jsx` or Unity 2D/3D interfaces

This results in:

- Real-time activation maps per frame
- Tooltip-based summaries
- Click-through access to associated Unit/Node pages

Each region is plotted as a sphere in anatomical space and linked to its resonance value over time.

Summary

Cognitive resonance is not a metaphor. It is a quantifiable, observable, and mathematically modelable system of neural alignment elicited by music.

C^3 builds a bridge between:

- **Structure** (from S^3 and R^3)
- **Perception** (via neural synchronization)
- **Emotion** (through limbic modeling)
- **Cognition** (via predictive structures)

This fusion yields not only a deepened understanding of musical experience, but also a practical computational system that allows music to be measured, mapped, predicted, and creatively shaped.

II.2 Mathematical Modeling of the C^3 System

How is C^3 formulated mathematically? What are its temporal and structural components? How does it compute cognition in music?

2.2.1 Layered Structure: From Signal to Cognition

The C^3 system is constructed as a hierarchical neural modeling architecture operating on three levels:

$$C^3(t) = \sum_i \text{Unit}_i(t), \quad \text{Unit}_i(t) = \sum_j \text{Node}_{ij}(t), \quad \text{Node}_{ij}(t) = \sum_k \text{E}$$

Each Element represents a measurable neural signal, modeled mathematically. Nodes aggregate these signals into functional units (e.g., "Expectation Violation", "Tonal Familiarity"). Units represent full cognitive subsystems (e.g., IEU, SAU).

This bottom-up structure enables us to compute macro-level phenomena (emotion, memory, flow) from micro-level EEG/fMRI signals.

2.2.2 Primary Equation

At the system level:

$$C^3(t) = \sum_{i=1}^{9} w_i \cdot \text{Unit}_i(t)$$

Where:

- $C^3(t)$: Total cognitive resonance at time t
- $\text{Unit}_i(t)$: Activity of Unit i

- $w_i \in \mathbb{R}$: Empirically tunable weight for each Unit (default: 1.0)

Each Unit has a local expansion:

$$\text{Unit}_i(t) = \sum_{j=1}^{N_i} w_{ij} \cdot \text{Node}_{ij}(t)$$

- $\text{Node}_{ij}(t)$: Activity of Node j in Unit i
- $w_{ij} \in [0, 1]$: Functional contribution weight of Node j

Each Node is derived from its underlying Elements:

$$\text{Node}_{ij}(t) = \sum_{k=1}^{M_{ij}} w_{ijk} \cdot \text{Element}_{ijk}(t)$$

2.2.3 Element-Level Signal Modeling

An Element is a neural feature defined as:

$$\text{Element}_{ijk}(t) = S_{ijk}(t) \in [0, 1]$$

Where $S_{ijk}(t)$ is the normalized neural signal derived from:

- EEG band power (e.g., α, β, γ)
- ERP component (e.g., MMN, P300)
- fMRI BOLD z-score in MNI-space
- Phase coherence between regions

Each signal is transformed to the [0,1] scale using:

$$S_{ijk}(t) = \frac{x(t) - \min(x)}{\max(x) - \min(x)}$$

For oscillatory components:

$$x(t) = |\mathcal{H}\{\text{EEG}(t)\}|^2$$

(Where $\mathcal{H}$ is the Hilbert Transform envelope)

For ERP components:

$$x(t) = \text{ERP amplitude}(t - \tau)$$

For fMRI:

$$x(t) = z(t) = \frac{\text{BOLD}(t) - \mu}{\sigma}$$

All data is resampled to a common timeline resolution (typically 10 Hz, i.e., 100ms frames).

2.2.4 Weight Normalization and Adaptivity

Weights w_i, w_{ij}, w_{ijk} are initialized based on empirical studies:

- w_{ijk}: based on effect size from literature (e.g., Cohen's d)
- w_{ij}: based on relative contribution within Unit (e.g., entropy vs. ERP)
- w_i: application-specific (e.g., PIU emphasized in flow-based systems)

Over time, weights can be adapted dynamically based on:

- Task relevance (e.g., in therapy, CTU may be downregulated)
- User feedback (e.g., neurofeedback systems)
- Generative models (e.g., AI uses the feedback to alter music in real time)

2.2.5 Matrix Representation

The entire C³ system can be formalized as a three-layer matrix multiplication:

$$C^3(t) = \mathbf{W}^T \cdot \mathbf{U}(t)$$

Where:

$$\mathbf{U}(t) = [\text{Unit}_1(t), \ldots, \text{Unit}_9(t)]^T \in \mathbb{R}^{9\times1}$$

Each $\text{Unit}_i(t)$ is:

$$\text{Unit}_i(t) = \mathbf{w}_i^T \cdot \mathbf{N}_i(t)$$

And each $\mathbf{N}_i(t)$ (Node vector) is:

$$\mathbf{N}_i(t) = [\text{Node}_{i1}(t), \ldots, \text{Node}_{iN}(t)]^T$$

This layered abstraction makes it possible to:

- Implement efficient GPU-based computation
- Track changes per unit or per frame
- Enable real-time synchronization with audiovisual feedback

2.2.6 Implementation Schema

Each Element is linked to:

- `unit_id` (e.g., CTU)
- `node_id` (e.g., harmonic_dissonance)
- MNI coordinates
- Measurement method (EEG, fMRI)
- Source file path (for dynamic signal streaming)

All metadata is stored in `brain_coords.json`, and signals are fed as `.edf`, `.csv`, or `.json` time series.

Visual representation is done through:

- `NodeView.jsx` for cognitive UI
- `GlassBrain.jsx` for anatomical UI
- `C3Engine.ts` (or Python backend) for signal computation

Summary

The C³ system is not merely a conceptual framework but a fully formalized, mathematically grounded computational model. It:

- Integrates real neural signals into a unified cognitive resonance space
- Adapts to multiple temporal and structural resolutions
- Operates across Unit, Node, and Element layers
- Is designed for interactive, real-time applications in analysis, composition, education, and therapy

This mathematical architecture makes it possible to represent the complexity of musical cognition as a dynamic, quantifiable, and deeply interpretable process.

III.1 Unit Definitions and Functional Roles

3.1.1 CTU – Cognitive Tension Unit

Models cognitive stress responses to musical dissonance, entropy, and tonal distance.

Overview

The Cognitive Tension Unit (CTU) is designed to capture and model the neural mechanisms of cognitive dissonance, ambiguity, and harmonic instability in response to complex musical stimuli. It measures the brain's response to tonal unpredictability, spectral irregularity, and harmonic deviation by tracking electrophysiological indicators and hemodynamic signals primarily across the dorsolateral prefrontal cortex (DLPFC), anterior cingulate cortex (ACC), and inferior frontal gyrus (Broca area).

This unit operates on the hypothesis that musical dissonance, entropy, and distance from tonal centers create a measurable increase in cognitive load. These effects are observable via:

- Alpha and beta phase-locking in frontal regions (EEG)
- Increased BOLD activity in cingulate cortex (fMRI)
- Beta-band amplitude increases in Broca's area during spectral irregularity

Mathematical Representation

The internal model of CTU is:

$$\mathrm{CTU}(t) = w_1 \cdot \mathrm{Node}_{\text{Harmonic Dissonance}}(t) + w_2 \cdot \mathrm{Node}_{\text{Spectral Entropy}}(t) + w_3 \cdot \mathrm{No}$$

Where:

- w_1, w_2, w_3: empirically tunable node weights

Each Node has one or more Elements grounded in EEG/fMRI markers.

Nodes and Elements

Node: Harmonic Dissonance

- **Description**: Models increased cognitive stress when dissonant chords or pitch clusters appear in a tonal context
- **Region**: DLPFC, ACC
- **EEG Marker**: Alpha–beta phase locking
- **Citation**: Fishman et al., 2001

Element:

- EEG Phase Locking (α–β)
- EEG electrodes: F3, Fz
- MNI Coordinate: [+32, +50, +20]
- GlassBrainMap ID: `ctu_harmonic_dissonance_01`

Node: Spectral Entropy

- **Description**: Quantifies the unpredictability and information density in the sound spectrum
- **Region**: Broca’s area (left IFG), temporal cortex
- **EEG Marker**: Beta amplitude increase
- **Citation**: Norman-Haignere et al., 2013

Element:

- Spectral Complexity Response
- MNI Coordinate: [+64, –22, +4]
- GlassBrainMap ID: `ctu_entropy_01`
- EEG channel cluster: F7–T7
- fMRI: Left IFG BOLD increase

Node: Harmonic Distance

- **Description**: Measures perceived tonal instability caused by deviations from local tonal center
- **Region**: DLPFC, ACC
- **EEG Marker**: Alpha amplitude increase
- **Citation**: Hyde et al., 2008

Element:

- EEG Alpha Power (ACC-centered)
- MNI Coordinate: [+30, +36, +20]
- GlassBrainMap ID: `ctu_distance_01`
- Method: Power spectral density (EEG)

GlassBrainMap Integration

The CTU Nodes are spatially registered within the GlassBrain system. Their associated coordinates and regions are as follows:

Node ID	MNI Coordinates	Region	Tooltip
`ctu_harmonic_dissonance_01`	[+32, +50, +20]	DLPFC + ACC	EEG α–β phase-locking
`ctu_entropy_01`	[+64, –22, +4]	Broca + Temporal Cortex	Beta increase in spectral complexity
`ctu_distance_01`	[+30, +36, +20]	ACC + DLPFC	EEG Alpha increase for tonal ambiguity

These markers appear in the `GlassBrain.jsx` component and are directly linked to the `ctu.json` file in the system.

Functional Summary

The CTU unit functions as a cognitive load detector, dynamically representing how musical instability translates into mental effort. It is especially responsive to:

- Tonal deviations
- Unexpected dissonances
- High spectral entropy

These events are interpreted by the brain as uncertainty or cognitive conflict, which C³ measures in real time.

CTU is often antagonistic to PIU (Phenomenological Immersion Unit): increased CTU activity often leads to decreased absorption or flow.

Applications

- **Therapeutic Monitoring**: Cognitive overload indicators in neurorehabilitation
- **Compositional Tools**: Dynamic tension mapping for film scoring or generative music
- **EEG Feedback Systems**: Real-time CTU readout for adaptive sound environments

3.1.2 AOU – Affective Orientation Unit

Models emotional resonance to music through valence and arousal dimensions.

Overview

The Affective Orientation Unit (AOU) quantifies the listener’s affective response to music by modeling two principal emotional axes:

- **Valence**: The pleasantness or unpleasantness of the musical stimulus
- **Arousal**: The intensity or physiological activation induced by the music

These dimensions are computed through EEG and fMRI indicators within limbic, prefrontal, and motor cortical areas—including the amygdala, ventral striatum, MPFC, SMA, and STG.

AOU is particularly sensitive to:

- Tonal stability (linked to positive valence)
- Spectral balance and consonance (linked to emotional reward)
- Tempo and rhythmic complexity (linked to arousal and motor drive)

Mathematical Representation

AOU is computed as a weighted combination of two core Nodes:

$$\text{AOU}(t) = w_1 \cdot \text{Valence}(t) + w_2 \cdot \text{Arousal}(t)$$

Each of which is decomposed into signal-bearing Elements:

Valence Axis:

$$\text{Valence}(t) = v_1 \cdot \text{TonalStability}(t) + v_2 \cdot \text{SpectralBalance}(t) + v_3 \cdot \text{HarmonicConsonance}(t)$$

Arousal Axis:

$$\text{Arousal}(t) = a_1 \cdot \text{Tempo}(t) + a_2 \cdot \text{SpectralFlux}(t) + a_3 \cdot \text{RhythmicComplexity}(t)$$

All elements are normalized between 0–1, with weights derived from empirical affective neuroscience research.

Nodes and Elements

Node: Valence

- **Function**: Detects emotional polarity of the musical input
- EEG Alpha Asymmetry in MPFC correlates with valence level
- Gamma and BOLD activation in amygdala and ventral striatum correlate with emotional reward

Element: Tonal Stability

- Region: MPFC
- EEG Marker: Alpha asymmetry
- MNI: [+6, +52, +10]
- Citation: Zatorre & Halpern, 2005

Element: Spectral Balance

- Region: Amygdala, Insula
- Method: fMRI + EEG Gamma
- MNI: [-20, 0, -12]
- Citation: Koelsch, 2011

Element: Harmonic Consonance

- Region: Ventral Striatum, NAcc
- Method: fMRI
- MNI: [+10, +8, -10]
- Citation: Salimpoor et al., 2011

Node: Arousal

- **Function**: Captures intensity and activation driven by musical rhythm, tempo, and flux

Element: Tempo Dynamics

- Region: Motor Cortex
- EEG: Beta-band amplitude
- MNI: [+40, -10, +60]
- Citation: Janata et al., 2009

Element: Spectral Flux

- Region: STG
- EEG: Theta
- MNI: [+50, +10, -6]
- Citation: Alluri et al., 2012

Element: Rhythmic Complexity

- Region: SMA
- EEG/fMRI: Beta + BOLD
- MNI: [+6, -6, +70]
- Citation: Chen et al., 2008

GlassBrainMap Coordinates

Node	MNI Coordinates	Region	Citati
Tonal Stability	[+6, +52, +10]	MPFC	Zatorr
Spectral Balance	[-20, 0, -12]	Amygdala, Insula	Koelsc
Harmonic Conson.	[+10, +8, -10]	Ventral Striatum	Salimp
Tempo Dynamics	[+40, -10, +60]	Motor Cortex	Janata
Spectral Flux	[+50, +10, -6]	STG	Alluri
Rhythmic Comp.	[+6, -6, +70]	SMA	Chen (

These entries are directly mapped into the GlassBrain SVG and visualized in the `GlassBrain.jsx` component as colored resonance hotspots.

Functional Summary

The AOU unit is the affective engine of the C³ system. It accounts for the emotional tone of musical events using neurobiologically validated metrics. It plays a key role in:

- Differentiating emotionally positive/negative musical content

- Identifying peaks of arousal or relaxation
- Guiding adaptive audio systems (e.g., emotion-matching playlists, AI composition targeting affective impact)

It interacts heavily with:

- CTU (tension)
- PIU (flow state)
- RSU (integrated resonance summary)

Applications

- Affective tagging in music libraries (real-time emotional metadata)
- Neurofeedback therapy for emotional regulation
- Real-time composition tools for mood shaping and expressive calibration

3.1.3 IEU – Intuitive Expectation Unit

Models predictive listening and mismatch responses based on harmonic, rhythmic, and melodic entropy deviations.

Overview

The Intuitive Expectation Unit (IEU) simulates the listener's internal predictive model during music listening. It monitors how the brain anticipates musical structure and reacts to violations of those expectations. This encompasses both preconscious responses (e.g., MMN) and attended violations (e.g., P300), as well as uncertainty metrics (e.g., melodic entropy).

IEU reflects a foundational principle of musical cognition: expectation and surprise are primary drivers of attention, emotion, and memory. The system quantifies these dynamics through:

- Early prediction-error signals (EEG MMN in STG)
- P300 ERP responses in premotor regions
- Neural tracking of melodic uncertainty (entropy measures in dACC and amygdala)

Mathematical Structure

IEU operates via three primary Nodes, weighted and combined over time:

$$\text{IEU}(t) = w_1 \cdot \text{HarmonicViolation}(t) + w_2 \cdot \text{RhythmicViolation}(t) + w_3 \cdot \text{MelodicEntropy}(t)$$

Each Node computes one functional aspect of expectation processing:

- w_1: Surprise in harmony
- w_2: Surprise in rhythm
- w_3: Global uncertainty in melodic information

Nodes and Elements

Node: Harmonic Expectation Violation

- **Function**: Detects dissonant or out-of-key chords in a tonal context
- **EEG Marker**: MMN ERP (mismatch negativity)
- **Region**: STG, auditory cortex
- **Citation**: Crespo-Bojorque et al., 2018

Element: MMN Response (Fronto-central)

- EEG Sites: Fz, Cz
- MNI: [0, +50, +20]
- Type: ERP (automatic deviance detection)

Node: Rhythmic Expectation Violation

- **Function**: Identifies tempo/beat violations and rhythmic anomalies
- **EEG Marker**: P300 ERP (conscious deviance)
- **Region**: SMA, Motor Cortex
- **Citation**: Schön et al., 2005

Element: P300 Response (Motor ERP)

- MNI: [+10, -10, +60]
- Type: ERP (attended violation detection)

Node: Melodic Entropy

- **Function**: Measures statistical uncertainty in melodic sequences
- **EEG Marker**: Alpha/theta shift in medial regions
- **fMRI**: dACC, amygdala
- **Citation**: Koelsch, 2008

Element: Information Content (Entropy)

- MNI: [+4, +18, +26]
- Metric: Entropy of pitch sequences (Shannon index, Markov chain predictability)

GlassBrainMap Anchors

Node	MNI Coordinates	Region	Method	Citation
Harmonic Violation	[0, +50, +20]	STG, Fronto-central	EEG (MMN)	Crespo-Bojorque et al., 2018
Rhythmic Violation	[+10, -10, +60]	SMA, Motor Cortex	EEG (P300)	Schön et al., 2005
Melodic Entropy	[+4, +18, +26]	dACC, Amygdala	EEG (α/θ), fMRI	Koelsch, 2008

These coordinates are rendered dynamically in the GlassBrain system and updated in real time as IEU activity changes.

Functional Summary

IEU measures how expected a musical moment is and how the brain responds to the unexpected. When expectation is fulfilled, IEU output remains low. When it is violated, resonance spikes occur—often triggering cognitive reorientation or emotional reappraisal.

IEU is crucial for:

- Segmenting musical flow
- Signaling novelty or structural change
- Synchronizing listener attention to surprise events

It interacts strongly with:

- CTU (tension under surprise)
- SAU (memory under uncertainty)
- AOU (emotional impact of surprise)

Applications

- Adaptive AI music systems: Dynamically vary predictability to hold listener attention
- Neuroeducation: Track student surprise and engagement in music learning
- Therapeutic design: Gradually increase predictability to rebuild trust in rhythm/melody recognition

3.1.4 SRU – Somatic Resonance Unit

Models rhythmic motor entrainment and body-based synchronization to musical stimuli.

Overview

The Somatic Resonance Unit (SRU) models how the brain's motor and premotor systems synchronize with perceived musical rhythm. It captures entrainment phenomena in motor cortices, pulse clarity processing, and tempo stability detection, which form the physiological basis for movement, tapping, dancing, and temporal prediction during music listening.

This Unit operates on the principle that rhythmic structures entrain cortical beta-band oscillations, and that clear pulse and metric structures elicit synchronized activity in:

- Supplementary Motor Area (SMA)
- Premotor Cortex (PMC)
- Basal Ganglia (Putamen)
- Cerebellum

SRU plays a crucial role in connecting auditory input to bodily response via motor entrainment and has direct applications in rehabilitation, rhythm training, and movement–based therapies.

Mathematical Model

SRU is defined by a linear combination of three functional Nodes:

$$\text{SRU}(t) = s_1 \cdot \text{PulseClarity}(t) + s_2 \cdot \text{MetricStability}(t) + s_3 \cdot \text{TempoStability}$$

Where:

- s_1, s_2, s_3: weight coefficients derived from neural effect size or application-specific relevance

Nodes and Elements

Node: Pulse Clarity

- **Function**: Detects the salience of rhythmic beat and its effect on motor cortex
- **EEG Marker**: Beta amplitude increase
- **Regions**: Motor Cortex, Putamen, Cerebellum
- **Citation**: Fujioka et al., 2012

Element: Beat Salience

- MNI: [+20, -10, +60]
- EEG: β-band power at C3/Cz
- fMRI: BOLD in cerebellar vermis and PMC

Node: Metric Stability

- **Function**: Represents the regularity and predictability of rhythmic subdivisions
- **EEG Marker**: Beta phase-locking
- **Regions**: SMA, Premotor Cortex
- **Citation**: Chen et al., 2008

Element: Metric Regularity

- MNI: [+6, +4, +64]
- EEG: β phase coherence
- fMRI: SMA BOLD activation

Node: Tempo Stability

- **Function**: Measures consistency in tempo; correlates with sensorimotor coupling strength
- **EEG Marker**: Interregional beta coherence
- **Regions**: Putamen, PMC
- **Citation**: Thaut et al., 2015

Element: Temporal Consistency

- MNI: [+28, -12, +60]
- EEG: β coherence (PMC $\leftrightarrow$ Basal Ganglia)

GlassBrainMap Anchors

Node	MNI Coordinates	Region	Method	Citation
Pulse Clarity	[+20, -10, +60]	Motor Cortex, Putamen	EEG (β), fMRI	Fujioka et al., 2012
Metric Stability	[+6, +4, +64]	SMA, Premotor Cortex	EEG phase-locking	Chen et al., 2008
Tempo Stability	[+28, -12, +60]	PMC, Basal Ganglia	EEG coherence	Thaut et al., 2015

These coordinates are used in the interactive GlassBrainMap, providing anatomical precision for motor–rhythmic coupling during music listening.

Functional Summary

SRU tracks real-time neural entrainment of the motor system to rhythmic music. It reflects how the body prepares to move, taps to the beat, and predicts upcoming rhythmic events. Its signal rises with:

- Stable and predictable beats
- High beat salience and metric clarity
- Entraining pulse structures (e.g., groove, swing, syncopation)

It often correlates positively with:

- PIU (immersion through movement)
- AOU (arousal from rhythm)
- NSU (neural synchronization)

Applications

- Motor rehabilitation and rhythm therapy
- Groove detection algorithms in music information retrieval
- Neurophysiological metrics of musical engagement

SRU is particularly relevant for dance research, tempo training, and interactive AI music generation that responds to bodily input or encourages physical engagement.

3.1.5 SAU – Semantic–Autobiographical Unit

Models music-evoked autobiographical memory and semantic association.

Overview

The Semantic–Autobiographical Unit (SAU) captures the interaction between musical structure and episodic/semantic memory systems in the brain. It quantifies how music activates autobiographical recall and semantic meaning through hippocampal–prefrontal–limbic networks.

This Unit models three core processes:

- Melodic repetition and its role in memory cueing
- Tonal familiarity and its effect on semantic access
- Timbre recognition and affective memory tagging

SAU is essential for explaining why certain music evokes specific personal memories, how musical familiarity shapes identity, and how past experiences influence present perception.

Mathematical Model

SAU is calculated as:

$$\text{SAU}(t) = s_1 \cdot \text{MotifRecurrence}(t) + s_2 \cdot \text{TonalityRecall}(t) + s_3 \cdot \text{TimbreFam}$$

Each Node maps to a neuroanatomically distinct memory-processing mechanism and contributes to the system's representation of semantic–episodic resonance.

Nodes and Elements

Node: Motif Recurrence

- **Function**: Detects melodic repetitions as episodic memory cues
- **EEG Marker**: Theta increase
- **Regions**: Hippocampus, Parahippocampal Gyrus
- **Citation**: Foss et al., 2007

Element: Melodic Repetition

- MNI: [–24, –40, –8]
- Method: EEG θ; fMRI hippocampal BOLD
- Description: Increased theta in temporal–limbic regions during reoccurring phrases

Node: Tonality Recall

- **Function**: Activates stored tonal patterns and schemas
- **EEG Marker**: Alpha increase
- **Regions**: MPFC, ACC
- **Citation**: Foss et al., 2007

Element: Tonal Familiarity

- MNI: [+6, +48, +8]
- Method: EEG α; MPFC BOLD
- Description: Retrieval of culturally learned tonality (major/minor schemas)

Node: Timbre Familiarity

- **Function**: Associates sound qualities with emotional memory
- **EEG Marker**: Gamma increase
- **Regions**: Amygdala, STG
- **Citation**: Chen et al., 2008

Element: Timbre-based Memory

- MNI: [+22, +0, –20]
- Method: EEG γ; STG BOLD
- Description: Recognition of familiar instrument types triggers affective memory

GlassBrainMap Anchors

Node	MNI Coordinates	Region
Melodic Repetition	[–24, –40, –8]	Hippocampus, ParaHC
Tonal Familiarity	[+6, +48, +8]	MPFC, ACC
Timbre-based Memory	[+22, +0, –20]	Amygdala, STG

These brain regions are anatomically mapped and visualized within the GlassBrain system in real time.

Functional Summary

SAU is the mnemonic layer of the C^3 system. It provides a biologically grounded mechanism for modeling:

- Musical nostalgia
- Identity-based music perception
- Semantic resonance of culturally or personally meaningful sounds

Its resonance increases with:

- Repetition of previously heard motifs
- Use of familiar tonal centers
- Recognition of personally associated timbres (e.g., piano from childhood)

SAU often co-activates with:

- AOU (affective salience)
- IEU (surprise + recall)
- IRU (interpersonal resonance and memory convergence)

Applications

- Music therapy for trauma, memory loss, and dementia
- AI playlist personalization based on listener history
- Autobiographical score composition using musical memories as structure

SAU also supports cultural modeling—how exposure to tonal systems and timbral norms shapes perceptual identity and long-term memory.

3.1.6 PIU – Phenomenological Immersion Unit

Models musical flow, absorption, and deep attention states through frontal, parietal, and default mode network dynamics.

Overview

The Phenomenological Immersion Unit (PIU) models non-analytical, affectively immersive listening states where attention, cognition, and motor inhibition converge to produce a sense of musical flow. These states often coincide with:

- Temporal suspension (loss of time awareness)
- Suppression of cognitive self-monitoring (reduced DMN activity)
- Heightened sensory and affective clarity

PIU tracks the transition from conscious processing to immersive resonance, particularly during:

- Slowly evolving harmonic textures
- Minimalistic repetition
- Ambient or timbrally complex soundscapes
- Trance, ritual, or meditative musics

It is one of the most non-linear and affective Units in the system.

Mathematical Structure

PIU is computed as a blend of three neurodynamic constructs:

$$\mathrm{PIU}(t) = p_1 \cdot \mathrm{AttentionalSalience}(t) + p_2 \cdot \mathrm{FlowState}(t) + p_3 \cdot \mathrm{TransientClarity}(t)$$

Each Node corresponds to a specific neural configuration observed during immersive listening or musical flow:

- p_1: attention gain
- p_2: default-mode suppression
- p_3: temporal resolution and transition anchoring

Nodes and Elements

Node: Attentional Salience

- **Function**: Measures the emergence of sustained, high-focus listening
- **EEG Marker**: Gamma-band elevation
- **Regions**: ACC, Frontal-Parietal Network
- **Citation**: Santoyo et al., 2023

Element: EEG Gamma Activation

- MNI: [+4, +32, +24]
- Description: Increased γ during perceptual focusing (timbre-based or harmonic absorption)

Node: Flow State Index

- **Function**: Tracks mental state transitions into immersive flow
- **EEG Marker**: Alpha suppression; BOLD reduction in DMN
- **Regions**: DMN hubs (mPFC, PCC); pre-SMA
- **Citation**: Patterson et al., 2002

Element: Default Mode Network Suppression

- MNI: [+2, +50, +6]
- Description: Drop in self-referential processing associated with absorption

Node: Transient Clarity

- **Function**: Anchors transitions (e.g., chord change, pulse entrance)
- **EEG Marker**: Beta phase-locking
- **Regions**: Frontal–Parietal Network
- **Citation**: Nozaradan et al., 2012

Element: Beta Phase Locking

- MNI: [+20, +10, +64]
- Description: Transition clarity in EEG β-phase alignment

GlassBrainMap Anchors

Node	**MNI Coordinates**	**Region**	**EEG**
Attentional Salience	[+4, +32, +24]	ACC, FP Network	EEG
Flow State	[+2, +50, +6]	mPFC, DMN	EEG
Transient Clarity	[+20, +10, +64]	FP Network	EEG

These entries are integrated into GlassBrain SVG and shown in immersive resonance clusters.

Functional Summary

PIU is the immersive dimension of C³. It increases during:

- Deep listening
- Meditative or minimalistic music
- Non-verbal sound attention
- Flow-inducing music (e.g., ambient, trance, sacred chants)

PIU typically correlates with:

- $\downarrow$ CTU (reduced tension)
- $\uparrow$ AOU (emotional absorption)
- $\uparrow$ SAU (memory/identity resonance)
- $\uparrow$ IRU (shared flow in group listening)

The PIU trace reflects how long a listener is "lost in the music."

Applications

- Therapeutic music for anxiety, depression, PTSD
- Flow design in composition and sound installations
- Biometric scoring of listener engagement and trance induction

In real-time systems, PIU can be used as a threshold trigger to adjust musical pacing, harmonic density, or lyrical clarity, enhancing sustained immersion.

3.1.7 IRU – Interpersonal Resonance Unit

Models neural synchrony, emotional convergence, and inter-brain coherence during shared musical experiences.

Overview

The Interpersonal Resonance Unit (IRU) measures the shared neural and affective dynamics that arise when music is experienced collectively. This Unit is founded on the principles of:

- Inter-brain coherence (neural synchrony across listeners)
- Emotional contagion via acoustic affect cues
- Social synchronization of motor and affective circuits during group musical experiences

IRU is based on emerging research from hyperscanning EEG, dual-fMRI, and social-cognitive neuroscience, which shows that synchronous music listening, especially in emotionally charged contexts, causes measurable co-activation of limbic, temporal, and frontal regions across brains.

Mathematical Structure

IRU is computed as a weighted average of three social-cognitive resonance Nodes:

$$\text{IRU}(t) = r_1 \cdot \text{InterBrainCoherence}(t) + r_2 \cdot \text{SocialSynchrony}(t) + r_3 \cdot \text{Emot}$$

Each Node represents a distinct interpersonal neural mechanism triggered by collective music engagement.

Nodes and Elements

Node: Inter-Brain Coherence

- **Function**: Measures synchronized alpha/theta rhythms between participants
- **EEG Marker**: Cross-brain phase-locking
- **Regions**: Frontal Cortex, Superior Temporal Gyrus (STG)
- **Citation**: Wallmark et al., 2018

Element: EEG Hyperscanning Coherence

- MNI: [0, +52, +14]
- Metric: Phase coherence across listener dyads (α/θ bands)
- Description: Increased alignment in neural oscillations during co-listening

Node: Social Synchrony

- **Function**: Captures joint activation in movement and attention networks
- **EEG Marker**: Gamma amplitude increases during joint attention
- **Regions**: Frontal Cortex, STG
- **Citation**: Wallmark et al., 2018

Element: Frontal-Temporal Activation

- MNI: [+12, +42, +14]
- EEG: γ power co-fluctuations in listeners
- Description: Mirrors shared attention and gesture during music

Node: Emotional Resonance

- **Function**: Models shared emotional responses via limbic system coupling
- **EEG/fMRI Marker**: Alpha asymmetry; amygdala BOLD
- **Regions**: Amygdala, ACC
- **Citation**: Yang et al., 2025

Element: Limbic Activation

- MNI: [+8, +6, -10]
- Description: Shared peaks in arousal/valence across participants

GlassBrainMap Anchors

Node	MNI Coordinates	Region	Method	Citation
Inter-Brain Coherence	[0, +52, +14]	Frontal + STG	EEG α/θ Sync	Wallmark et al., 2018
Social Synchrony	[+12, +42, +14]	Frontal + STG	EEG γ	Wallmark et al., 2018
Emotional Resonance	[+8, +6, -10]	Amygdala + ACC	fMRI + EEG α	Yang et al., 2025

These elements are rendered together in a special Group-Level GlassBrain Overlay, enabling visualization of multi-brain synchrony fields.

Functional Summary

IRU allows C³ to model music not just as an internal experience, but as a shared cognitive–emotional field. IRU increases when:

- Music is heard in a social context
- Listeners share a history or cultural framework
- Body-based synchronization (e.g., group clapping, dancing) is present
- Emotional peaks align across individuals

IRU is the bridge between:

- AOU (individual affect)
- PIU (flow)
- RSU (integrated group resonance)

Applications

- Social music therapy (e.g., group rhythm interventions, trauma re-integration)
- AI-driven shared music experiences (co-listening apps, social playlist engines)
- Group neuroscience and emotion regulation training

IRU enables the modeling of shared emotional spaces and neural entrainment ecosystems, placing music at the center of group cognition.

3.1.8 NSU – Neural Synchronization Unit

Models neural entrainment, phase-locking, and large-scale temporal coherence in auditory–motor–cognitive systems.

Overview

The Neural Synchronization Unit (NSU) captures how music organizes and synchronizes brain activity through rhythmic and spectral structure. It operates on the principle that musical events—especially rhythmic periodicities, spectral regularities, and melodic contours—can entrain large-scale cortical networks via:

- Phase-locking to external rhythms
- Cross-regional coherence in gamma, beta, and alpha bands
- Temporal anticipation and predictive resonance

NSU integrates empirical findings from EEG, MEG, and frequency–following response (FFR) studies that show how temporal structure in sound becomes mirrored in neural timing.

It is particularly relevant in contexts of:

- Rhythm perception and pulse tracking
- Sensory–motor coupling
- Beat-based learning and coordination
- Tonal phase tracking and attentional alignment

Mathematical Model

NSU is computed as:

$$\text{NSU}(t) = n_1 \cdot \text{GammaCoherence}(t) + n_2 \cdot \text{BetaPhaseLocking}(t) + n_3 \cdot \text{Alph}$$

Each Node aggregates multiple regional signals reflecting cross-frequency and cross-site temporal alignment.

Nodes and Elements

Node: Gamma Coherence

- **Function**: Tracks gamma-band synchrony between auditory and motor cortices
- **EEG Marker**: γ coherence (STG $\leftrightarrow$ Motor Cortex)
- **Regions**: STG, Motor Cortex
- **Citation**: Bidelman & Heinz, 2011

Element: Gamma-band Synchrony

- MNI: [+38, -10, +52]
- EEG: γ-band inter-site phase clustering
- Metric: Phase-locking value (PLV)

Node: Beta Phase Locking

- **Function**: Models cortical beta-band entrainment in motor sequencing and prediction
- **EEG Marker**: β phase locking (SMA $\leftrightarrow$ Basal Ganglia)
- **Regions**: SMA, PMC, Basal Ganglia, STG
- **Citation**: Bidelman & Krishnan, 2009

Element: Motor-Auditory Integration

- MNI: [+8, -6, +60]
- EEG: Inter-site β PLV across frontal–motor nodes

Node: Alpha Synchrony

- **Function**: Measures large-scale attentional coherence in alpha network
- **EEG Marker**: α interhemispheric synchrony
- **Regions**: Frontal Cortex, Parietal Cortex
- **Citation**: Strait et al., 2012

Element: Frontal–Parietal Alpha Synchrony

- MNI: [+4, +52, +10]
- EEG: α band coherence (fronto-parietal loop)
- Description: Indicator of global attention tuning to musical flow

GlassBrainMap Anchors

Node	MNI Coordinates	Region
Gamma Coherence	[+38, -10, +52]	STG, Motor Cortex
Beta Phase Locking	[+8, -6, +60]	SMA, Basal Ganglia
Alpha Synchrony	[+4, +52, +10]	Frontal–Parietal Cortex

These nodes are rendered in the GlassBrain overlay, showing synchronization fields in time-resolved layers.

Functional Summary

NSU tracks how temporally structured sound creates temporally structured brain activity.

It reflects:

- High-frequency (γ) micro-entrainment for precise timing
- Mid-frequency (β) entrainment for motor planning
- Low-frequency (α) coherence for attentional binding

NSU is crucial for:

- Coordinated movement to music
- Pulse perception
- Learning of rhythmic and metrical structure
- Sustained attention

It interacts strongly with:

- SRU (somatic resonance)
- IEU (predictive modeling)
- RSU (resonance integration)

Applications

- Neuroeducation for rhythm training, beat perception, language timing
- Therapeutic rhythm protocols (e.g., Parkinson's interventions)
- Real-time music–brain feedback in composition and AI systems

NSU gives C³ the neural infrastructure to temporally align listener state with musical structure—turning sound into synchronization.

3.1.9 RSU – Resonance Synthesis Unit

Integrates all Unit-level outputs into a unified cognitive–emotional–motor resonance profile.

Overview

The Resonance Synthesis Unit (RSU) is the final computational layer in the C³ system. Unlike the other eight Units, which represent specific cognitive functions (e.g., tension, emotion, memory), RSU performs a global integration of all underlying Units, Nodes, and Elements to compute:

- Total moment-to-moment cognitive resonance
- System-wide synchrony and coherence
- Weighted convergence of multi-dimensional neural states

RSU does not introduce new raw data. Rather, it serves as a nonlinear summarization node—a temporal and structural resonance integrator—combining emotion, memory, expectation, attention, motor entrainment, and inter-brain synchrony into a single multidimensional vector.

Mathematical Formulation

RSU operates across two primary mathematical layers:

a. Unified Coherence Score (UCS)

$$\text{UCS}(t) = \frac{1}{9}\sum_{i=1}^{9} C_i^3(t)$$

Where:

- $C_i^3(t)$: The computed resonance from each Unit at time t

Output: Scalar between $[0, 1]$ representing total network activation.
This value can be interpreted as a resonance index—how "fully activated" the cognitive-musical system is.

b. Weighted Network Fusion (WNF)

$$\text{RSU}(t) = \mathbf{W} \cdot \mathbf{C}^3(t)$$

Where:

$$\mathbf{C}^3(t) = [\text{CTU}(t), \text{AOU}(t), \ldots, \text{NSU}(t)]^T \quad \mathbf{W} \in \mathbb{R}^{1\times 9}$$

$\mathbf{W}$: Application-specific or data-derived weights (e.g., in therapy, PIU and SAU may be up-weighted)

This produces a 1D or N-dimensional fusion vector depending on context (e.g., therapy, composition, neuroscience modeling).

Nodes and Elements

RSU consists of three conceptual Nodes, each acting as a functional lens on the total C³ state.

Node: Unified Coherence

- **Function**: Measures cross-Unit synchrony at each time slice
- **Metric**: Multi-band EEG synchrony
- **Regions**: ACC, PCC, STG, Frontal Cortex
- **Citation**: Yang et al., 2025

Element: Network-Wide EEG Coherence

- MNI: [+8, +44, +12]
- Signal: PLV across Unit-critical regions
- Description: Degree to which separate Units exhibit synchronous resonance

Node: Consonance Clarity

- **Function**: Aggregates emotional, tonal, and spectral harmony into a single perceptual clarity index
- **Metric**: Consonance fusion index
- **Regions**: NAcc, Amygdala, Broca, PMC
- **Citation**: Salimpoor et al., 2011

Element: Resonant Fusion Metric

- MNI: [+12, +10, -10]
- Description: Integrated emotional–harmonic salience field
- Method: Entropy-weighted BOLD + EEG synchrony sum

Node: Network Fusion

- **Function**: Computes total resonance field from all C³ Units
- **Metric**: Cross-unit vector field
- **Regions**: DMN, Sensorimotor Network, Limbic System
- **Citation**: SRC Master Report

Element: Cross-Unit Weighted Synthesis

- MNI: N/A (meta-region spanning all prior Units)
- Description: Final vector representation of listener state
- Output: Dynamic resonance map

GlassBrainMap Anchors

Node	MNI Coordinates	Region	Method	Citation
Unified Coherence	[+8, +44, +12]	ACC, STG, PCC, Frontal	EEG Multi-Band Coherence	Yang et al., 2025
Consonance Clarity	[+12, +10, -10]	NAcc, Broca, Amygdala, PMC	fMRI + EEG Fusion	Salimpoor et al., 2011
Network Fusion	—	Multi-unit Meta Layer	All prior Unit inputs	SRC Report (2025)

These form the top-level projection in the Resonance Map, a composite data structure exported per listener or per musical piece.

Functional Summary

RSU enables system-wide monitoring and integrated decision-making:

- Real-time tracking of full cognitive-emotional resonance
- Personalized resonance profile calculation
- Macro-temporal analysis (e.g., identifying resonance arcs over time)

It often acts as a target state:

- For adaptive AI generation
- For guided composition
- For neurofeedback therapy

Applications

- Therapeutic optimization: Detect optimal convergence of memory, attention, and emotion
- Neuro-symbolic composition: Use RSU trajectory to sculpt musical form
- Resonance fingerprinting: Build listener resonance signatures for adaptive playlists

RSU is the endpoint and also the summary interface of the C³ system. It translates complex internal dynamics into usable, visible, and actionable outputs.

III.2 Node and Element Architecture

How are Nodes and Elements structured, connected, and activated in C³?

3.2.1 Hierarchical Composition: UNIT → NODE → ELEMENT

C³ operates on a three-tiered architecture, in which each UNIT is subdivided into Nodes, and each Node consists of one or more Elements.

This design allows C³ to represent cognition at multiple resolutions:

Level	Function
UNIT	Macro cognitive subsystem (e.g., Emotion, Memory)
NODE	Functional construct (e.g., Flow, Expectation)
ELEMENT	Measurable neural signal (EEG, fMRI, MEG)

Each Element is traceable to:

- A neurophysiological method (e.g., EEG phase coherence, ERP, fMRI BOLD)
- A brain region (MNI coordinates)
- A citation (peer-reviewed neuroscientific literature)

3.2.2 Node Definition

A Node in C³ represents a mid-level computational module with a singular cognitive or affective function.

Mathematically:

$$\text{Node}_{ij}(t) = \sum_{k=1}^{M_{ij}} w_{ijk} \cdot \text{Element}_{ijk}(t)$$

Where:

- $\text{Node}_{ij}(t)$: The j-th Node in the i-th Unit at time t
- w_{ijk}: Element weight within the Node (typically 0.25–0.50 normalized)
- $\text{Element}_{ijk}(t)$: The k-th Element in that Node

Each Node acts as a resonance transformer, mapping local signals into cognitive-scale responses (e.g., surprise, salience, tension, familiarity, arousal).

3.2.3 Element Definition

An Element is the atomic analytical unit of C³. It consists of:

Field	Type	Description
`label`	string	e.g., “EEG Gamma Activation”
`method`	enum	“EEG”, “fMRI”, “MEG”, or hybrid
`regions`	list[string]	Anatomical targets (e.g., ACC, STG)
`mni`	list[int]	MNI coordinates for GlassBrainMap
`signal`	time series	Raw or preprocessed signal (external input)
`value(t)`	float	Normalized resonance value at time t
`citation`	string	Reference to literature

Each Element is initialized by parsing its signal from a time series and rescaling it:

$$\text{value}(t) = \frac{x(t) - \min(x)}{\max(x) - \min(x)}$$

where $x(t)$ is derived from EEG amplitude, ERP waveform, BOLD z-score, or spectral entropy.

3.2.4 Dynamic Activation Model

Each Element operates on a sliding time window (typically 100–250 ms). Activation values are updated in real-time (or simulated) via signal ingestion pipelines. Signals may be:

- **Real:** (e.g., live EEG via OpenBCI or Emotiv)
- **Pre-recorded:** (e.g., `.csv`, `.edf`, `.json` time series)
- **Simulated:** (e.g., parameterized sine waves for prototyping)

At each frame:

- The Element computes value(t)
- The Node aggregates these into a Node response
- The Unit aggregates across its Nodes
- RSU receives the integrated output for synthesis

3.2.5 Cross-Unit Node Comparability

All Nodes across different Units are mapped to a common scale:

$$\text{Node}_{ij}(t) \in [0, 1]$$

Time resolution: aligned across Units
Visualization: displayable in parallel or stacked timelines

This standardization allows for:

- Comparative graphing (e.g., AOU vs. CTU over time)
- Cluster analysis (e.g., grouping similar emotional-motor patterns)
- Synchronization tracking (e.g., NSU and SRU phase alignment)

3.2.6 Graph Data Format and API Integration

All Nodes and Elements are encoded in structured JSON, enabling flexible API calls, visual rendering, and machine learning input.

Sample schema:

```
{
  "node_id": "ctu_dissonance",
  "unit": "CTU",
  "elements": [
    {
      "id": "phase_locking",
      "method": "EEG",
      "regions": ["DLPFC", "ACC"],
      "mni": [32, 50, 20],
      "value": 0.64,
      "citation": "Fishman et al., 2001"
    }
  ]
}
```

These files are stored in:

```
/static/data/c3/
 ctu.json
 aou.json
 ...
```

and loaded via:

3.2.7 GlassBrainMap Integration (Anatomical Anchor)

Each Element is visually represented in the GlassBrain system:

- MNI coordinates are mapped to SVG/canvas projection space
- Tooltip shows `label`, value(t), and `citation`
- Color intensity reflects value(t)
- Click opens corresponding Node view

For example:

`ctu_harmonic_dissonance_01` → `[+32, +50, +20]` → `EEG` α–β `locking`

This allows cognitive + anatomical views to be seamlessly unified.

Summary

Nodes and Elements are the computational heart of C^3. They connect musical structure to brain dynamics via:

- Quantified, normalized, temporally resolved signals
- Modular aggregation from bottom (signal) to top (unit)
- Flexible visual and API-driven interfacing

This architecture makes it possible to compute, visualize, and interpret musical cognition with unprecedented resolution, precision, and interactivity.

IV. GlassBrainMap Integration and Spatial Modeling

How does C^3 map Nodes and Elements to anatomical regions? How is this visualized and computed spatially?

4.1 Overview: Why Spatialization Matters

The C^3 system is not purely symbolic or abstract. It is neuroanatomically grounded. Each Element is tied to a measurable neural signal in a specific brain region. The spatial mapping of these regions:

- Provides anatomical context for cognitive resonance
- Enables dynamic visualizations of neural activation patterns
- Supports inter-unit interaction modeling via spatial proximity and network overlap

To achieve this, C^3 integrates a dedicated visualization layer called **GlassBrainMap**, which combines:

- A high-resolution 2D SVG anatomical brain model
- A JSON-based coordinate database
- A real-time rendering interface (`GlassBrain.jsx`) for visualizing activation dynamics

4.2 The Coordinate System: MNI Anchoring

All Elements in C^3 are associated with **MNI (Montreal Neurological Institute)** coordinates, the standard in neuroimaging.

Each Element record includes:

- `mni`: $[x, y, z]$ triple
- `region`: anatomical label (e.g., SMA, ACC, STG)
- `radius_mm`: spatial spread (default: 5–8 mm sphere)
- `value(t)`: resonance strength at time t
- `citation`: literature source validating the location and signal

This system allows each Element to be visualized as a spatial field centered on its coordinate, colored or scaled based on its activation level.

4.3 Coordinate Data Architecture

All anatomical data is stored in a JSON file:

File: `static/data/brain_coords.json`

These are cross-referenced with each Unit's data file (e.g., `ctu.json`) and visualized in real time.

4.4 GlassBrain Interface

The spatial interface is rendered by a React component (`GlassBrain.jsx`) using:

- An interactive SVG brain diagram (`brainmap.svg`)
- Overlaid `<circle>` elements for each active Node/Element
- `title` attributes for hover-based tooltips
- Optional `onClick` behavior for Unit/Node navigation

Sample visualization logic:

This allows for live, frame-by-frame cognitive activity visualization, spatially grounded in the brain's structure.

4.5 Spatial Modeling Layers

There are three core layers in the spatial model:

Layer	Purpose
Anatomical Base	Outlines brain shape and region boundaries (SVG
Functional Points	Locations of Nodes and Elements (MNI coordinat
Dynamic Overlay	Time-resolved activity values, color-encoded

This tri-layer system makes it possible to animate, filter, and interact with complex neurocognitive fields.

4.6 GlassBrain Interactions

The visualization supports:

- **Hover tooltips:** Node/Element label, region, value, citation
- **Click-through navigation:** Links to full Node/Unit documentation
- **Color mapping:** Resonance values encoded via color gradients
- **Temporal updates:** Per-frame re-rendering as values change over time

Optionally, spatial dynamics can be extended via:

- Highlight paths of cross-unit coherence (e.g., CTU $\leftrightarrow$ AOU links)
- Animate resonance flow across hemispheres
- Implement attention-focused zooms or heatmaps

4.7 Spatial–Temporal Integration

GlassBrainMap is fully synchronized with C³'s time engine. At each time frame:

- Each active Element's value(t) is fetched
- Its color/opacity is updated in the SVG
- The cumulative field is rendered (or stored/exported)

This creates a dynamic 2D projection of 3D cognition, visible in real time or as an analytic export (e.g., heatmap or trajectory animation).

Summary

GlassBrainMap transforms C³ from a symbolic system into a neuroanatomically immersive experience.

It allows researchers, composers, and clinicians to:

- See where cognition happens
- Visualize how multiple Units interact in space
- Trace temporal arcs of resonance across the brain

It is the spatial backbone of the C³ model—anchoring abstract resonance vectors in biological reality.

V. System Dynamics and Temporal Computation

How does C³ compute resonance across time? What governs frame resolution, signal propagation, and feedback integration?

5.1 Temporal Framework

C³ operates as a discrete-time cognitive system, segmented into frames that represent slices of neural and musical time. The standard frame rate is 10 Hz (i.e., one frame every 100 ms), which aligns with:

- The optimal EEG resolution for cross-frequency tracking
- The perceptual threshold for auditory events
- The temporal granularity of musical microstructures (e.g., eighth notes at $\sim$120 BPM)

Each frame computes:

- Element values
- Node aggregations
- Unit outputs
- RSU synthesis
- GlassBrainMap spatial updates

5.2 Frame Processing Pipeline

At each time t, C³ executes the following pipeline:

Signal Ingestion From EEG, fMRI, JSON, or simulated sources

Preprocessed into normalized time series $x(t) \in [0, 1]$

Element Evaluation

$$\text{Element}_{ijk}(t) = \frac{x(t) - \min(x)}{\max(x) - \min(x)}$$

Node Aggregation

$$\text{Node}_{ij}(t) = \sum_k w_{ijk} \cdot \text{Element}_{ijk}(t)$$

Unit Calculation

$$\text{Unit}_i(t) = \sum_j w_{ij} \cdot \text{Node}_{ij}(t)$$

System Output (RSU)

$$\text{RSU}(t) = \sum_i w_i \cdot \text{Unit}_i(t)$$

Spatial Rendering

- Update GlassBrainMap based on Element MNI locations
- Animate frame-by-frame neural states

5.3 Signal Modes and Temporal Sources

C³ supports multiple signal input modes:

Mode	Source	Use Case
Live	Real-time EEG (e.g., OpenBCI)	Neurofeedback, adaptive performance
Simulated	Parametric sine/square models	Testing, theory exploration
Imported	Pre-processed data (CSV, JSON)	Research replay, batch analysis
Model-based	Outputs from R³/S³ modules	Internal feedback integration

Signals are time-aligned across all Units and stored in synchronized arrays for each frame:

5.4 Real-Time Feedback Loop

C³ is built to support feedback mechanisms in which system output can influence future inputs, creating an adaptive cognitive engine.

For example:

- CTU output triggers simplification of harmonic content
- PIU spikes increase ambient density to sustain immersion
- RSU target value adjusts music generation parameters

This enables closed-loop interaction with:

- AI composition engines
- VR/AR environments
- Biometric controllers

5.5 Resonance Trajectory Modeling

Over time, C³ constructs a resonance trajectory:

$$\mathbf{C}^3(t) = [\mathrm{Unit}_1(t), \mathrm{Unit}_2(t), \ldots, \mathrm{Unit}_9(t)]$$

This vector evolves in 9D space and can be used for:

- Pattern recognition (e.g., flow state, surprise burst)
- Phase analysis (e.g., C³ spirals, convergence cycles)
- Emotion curves (e.g., AOU vs. CTU arcs)

Visualization options include:

- Line graphs of individual Units
- Radar plots at each time slice
- PCA/t-SNE dimensionality reduction of trajectory clusters

5.6 System Clock and Synchronization

C³ uses a master temporal clock, driven by:

- External sync (e.g., MIDI timecode, DAW sync)
- Internal clock (browser animation loop, WebAudio API)
- Real-time EEG timestamps

Each Unit's activity is synchronized to this clock, ensuring that:

- Temporal granularity remains constant
- Cross-unit integration remains coherent
- GlassBrainMap overlays animate in lockstep

5.7 Export and Storage

All C³ output is exportable in structured formats:

- `.json` frame logs
- `.csv` for analysis in Python/R
- `.svg`, `.mp4` GlassBrainMap video renderings
- `.c3` proprietary container for replay in simulation environments

This makes C³ both real-time and archival—suitable for live use, study, and post-hoc analysis.

Summary

C³'s temporal engine transforms raw signals into a flowing stream of cognitive resonance. Through:

- Frame-level computation
- Adaptive signal modeling
- Time-synchronized spatial projection

C³ achieves a unique combination of precision, fluidity, and biological plausibility. Its ability to track cognition across milliseconds to minutes enables deep insight into the evolving experience of music.

VI. Research and Application Domains

How can C³ be used? In what contexts does it generate value?

6.1 Music Therapy and Clinical Neurotechnology

C³ provides a groundbreaking opportunity for neurophysiologically grounded music therapy by quantifying and tracking brain-based responses to musical structure in real time.

6.1.1 Real-Time Emotional Profiling

Using AOU (affective orientation) and IRU (interpersonal resonance), therapists can monitor emotional states through:

- EEG alpha asymmetry (valence)
- fMRI amygdala–insula activation (arousal)
- Cross-brain synchrony (group therapy contexts)

This allows dynamic adjustments to:

- Musical content
- Patient feedback loops
- Group synchrony states

6.1.2 PTSD and Trauma Therapy

SAU (semantic-autobiographical) and PIU (phenomenological immersion) can support:

- Memory retrieval and reconsolidation through tonal cues
- Safe immersive states using ambient or flow-inducing structures
- Repatterning of trauma-linked neural pathways through predictable rhythmic entrainment

6.1.3 Motor Rehabilitation

SRU (somatic resonance) and NSU (neural synchronization) enable:

- Gait training via rhythm-guided beta entrainment
- Entrainment of cerebellar and basal ganglia networks
- Personalized tempo and meter calibration for stroke or Parkinson's patients

C³ becomes a neural interface for musical medicine.

6.2 AI Music Generation and Cognitive Feedback

The C³ system opens a new frontier in cognitively aware artificial intelligence for music.

6.2.1 Neuro-Adaptive Composition

Generative music models (e.g., RNN, Transformer, Diffusion) can be guided by:

- Desired C³ trajectory (e.g., AOU↑, CTU↓, PIU↑)
- Listener feedback via EEG input
- Emotional or narrative arc templates

This enables biometrically reactive music—sound that shifts in real time to support immersion, attention, or relaxation.

6.2.2 Real-Time Feedback Systems

With C³ embedded in AI systems:

- Adaptive film/game scoring becomes possible
- Live feedback concerts (brainwave-to-music mapping) can be executed
- Human–AI co-composition becomes cognitively contextualized

6.3 Education and Neuroaesthetic Learning

C³ supports a neuroscience-informed pedagogy of music.

6.3.1 Resonance-Based Curriculum Design

Educators can craft listening experiences and composition exercises that:

- Highlight tension and resolution (CTU)
- Elicit attention and expectation (IEU)
- Reinforce memory and identity through tonality and timbre (SAU)

This makes abstract musical concepts experientially grounded.

6.3.2 Student Brain Tracking

Via live EEG (or simulated training environments), educators can track:

- Focus/engagement states (PIU, NSU)
- Confusion or overload (CTU spikes)
- Flow progression across musical segments

This enables adaptive instruction, where the learning path is guided by resonance curves.

6.4 Artistic Exploration and Compositional Tools

C^3 is an interpretive and generative tool for composers, performers, and sound artists.

6.4.1 Resonance Mapping in Composition

A composer can model the resonance profile of a piece before it's written by:

- Designing C^3 trajectories (e.g., tension peak at minute 3)
- Mapping structure to cognitive states
- Back-calculating harmonic/rhythmic features to meet those trajectories

This allows intentional cognitive shaping of musical form.

6.4.2 Interactive Visuals and Installations

C^3's spatial and temporal outputs can drive:

- Live projection of GlassBrainMap overlays during performances
- Audience-specific scoring where real-time neural data sculpts the score
- Installation works based on shared inter-brain synchrony fields (IRU)

This links cognition, composition, and computation.

6.5 Scientific Research and Cognitive Modeling

C^3 provides a framework for computational music cognition, enabling:

- Hypothesis testing: Does syncopation increase CTU + SRU?
- Cross-style comparison: How does Bach's AOU curve differ from EDM?
- Cultural modeling: What tonal structures evoke strongest SAU response in different populations?

This allows neurocognitive theories of music to become quantifiable and testable.

Summary

C^3 is not only a theoretical model—it is an instrument:

- For therapy, it monitors and modulates emotional and neural states
- For AI, it contextualizes generation with cognitive targets
- For education, it reveals the brain's role in musical understanding
- For artistry, it expands the palette of expression
- For science, it bridges abstract theory and empirical verification

It transforms music from something we hear to something we can measure, sculpt, and share as a resonant experience.

VII. Future Directions and Development Roadmap

How will C^3 grow? What systems will it interface with? What frontiers does it open?

7.1 Real-Time Integration and Live System Deployment

7.1.1 Live EEG/BCI Synchronization

C^3 is designed to operate not only on pre-recorded data, but in real-time with live neural input.

- EEG headsets (e.g., OpenBCI, Emotiv, Muse S) can be connected via WebSocket or OSC
- C^3 frame computation can be run inside a web browser, Node.js, or Unity engine
- Live frame values (e.g., CTU = 0.62, PIU = 0.83) can be:
  - Fed into generative AI models
  - Used to control audiovisual parameters
  - Exported to researchers in JSON or OSC streams

This allows for neuroadaptive installations, brain–music feedback loops, and bio-interactive concerts.

7.1.2 Signal Bridge Modules

Planned integrations:

Module	**Description**
OSC Module	Open Sound Control bridge to DAWs, synths, AI to
WebSocket API	Lightweight protocol for browser–EEG interaction
Max/MSP Patch	Native support for creative coding environments
Python Streamer	Tensor-based streaming for machine learning pipeli

7.2 VR/AR and Immersive Cognitive Environments

C^3's GlassBrainMap + Cognitive Trajectory outputs can be rendered in 3D and embedded in immersive environments.

7.2.1 Unity & Unreal Engine Integration

- Dynamic brain overlays mapped to VR avatars
- Real-time resonance color fields, spheres, and graph networks
- Flow-state visualizations projected in 3D space

This enables experiential neuroaesthetics—where listeners "enter" their own cognition, or the resonance fields of a performance.

7.2.2 Spatial Sound + Resonance Mapping

By combining C^3 with spatialized audio systems (e.g., Ambisonics, Dolby Atmos), the system can:

- Route musical streams to match neural states
- Trigger directional auditory cues based on attention/flow values
- Modify sonic parameters based on resonance field tension

7.3 Multi-Listener Synchrony and Cross-Brain Modeling

Using IRU (Interpersonal Resonance Unit), future expansions include:

- Multi-brain environments (e.g., group concerts, collective biofeedback)
- Cross-brain synchrony maps showing where people resonate together
- Shared immersive environments where cognition is mapped and compared in real time

This allows researchers to study social cognition, empathy, and music-induced synchrony at scale.

7.4 Cross-Domain Integrations

C^3 is not a standalone tool; it is a cognitive middleware layer that can plug into:

Domain	Integration Role
Therapy	Real-time monitoring of neural states
Gaming/VR	Adaptive scoring, ambient shifts, player cognition
Education	Flow detection, rhythm learning enhancement
AI Art Tools	Emotion-matched generative feedback
Scientific Research	Hypothesis testing, data export, meta-analysis

C^3 becomes the neuro-semantic glue between music, systems, and cognition.

7.5 Platform Roadmap

Short-Term Goals (3–6 months)

- Full browser-based JSON pipeline
- Interactive web-based GlassBrainMap
- C^3-powered experimental composition tools
- Python and JavaScript SDKs for Unit simulation and visualization

Mid-Term Goals (6–12 months)

- EEG integration via OSC/WebSocket
- Unity + WebGL resonance rendering
- Composer dashboard for real-time Unit tracking
- Music therapy pilot studies with neurofeedback

Long-Term Goals (12+ months)

- Full real-time C^3 AI integration
- Multi-user cross-brain resonance platform
- Publication and standardization of the C^3 framework
- Institutional and clinical partnerships

7.6 Open Science and Community Expansion

C^3 is intended to be an open, modular, scientifically verifiable system.

- All Units, Nodes, and Elements are defined in transparent JSON
- Coordinate systems are public and validated (MNI-space)
- Citations and data trails are embedded and reproducible
- Community contributions can add new literature, modules, or brain regions

A future **C^3 Open Research Portal** will:

- Allow users to run and visualize C^3 sessions
- Share annotated resonance maps
- Explore collective cognitive musical responses

Summary

C³ is a foundation—not a ceiling.

It creates the infrastructure for real-time cognition-aware music systems, research tools, therapeutic protocols, and artistic interfaces. Its future lies not in complexity, but in connectivity: between minds, modules, and meaning.

VIII. Conclusion and Scientific Contribution of C³

What does C³ offer to music science, technology, and cognition?

8.1 Synthesis: From Signal to Cognition

The Cognitive Consonance Circuit (C³) is the first fully formalized system that models the brain's multi-dimensional, time-evolving response to music by integrating:

- Symbolic and spectral features (via S³ and R³)
- Measurable neural signals (EEG, fMRI, MEG)
- Cognitive constructs (attention, memory, emotion, expectation)
- Anatomical regions (mapped in MNI coordinates)
- Temporal dynamics (frame-level, real-time computation)

Through its layered architecture (UNIT $\rightarrow$ NODE $\rightarrow$ ELEMENT), mathematically grounded resonance equations, and GlassBrain-based spatial visualization, C³ offers an unprecedented resolution of musical cognition.

8.2 Scientific Contributions

C³ introduces the following key innovations to the scientific community:

8.2.1 A Unified Cognitive–Musical Architecture

No existing model combines:

- Tonal/spectral analysis
- Neural measurement
- Cognitive state modeling

in a modular, interconnected, and computable system.

C³ bridges this gap with:

- 9 Units modeling distinct cognitive faculties
- Direct neurophysiological grounding (61+ referenced papers)
- MNI-based anatomical mapping

8.2.2 Temporal–Cognitive Formalism

C³ treats musical cognition as a flowing resonance field. Its time-based equations:

- Allow computation of dynamic neural states
- Integrate with live EEG streams
- Align with musical structure at 10 Hz resolution

This transforms music analysis from static snapshots into real-time neurocognitive simulation.

8.2.3 Intermodular Connectivity within SRC

C³ is not standalone. It:

- Receives input from S³ (spectral structure) and R³ (harmonic topology)
- Computes the cognitive response
- Sends feedback back into the system (e.g., suggesting structural change)

This feedback loop allows musical systems to self-regulate based on real or simulated cognition.

8.2.4 Visual Neuroanatomical Integration

The GlassBrainMap is not just an illustration—it's an analytical engine:

- Every Element is spatially mapped (MNI)
- Activation is color-coded and time-resolved
- Tooltips, click-through, and overlays show layered cognition

This creates a semantic spatial interface for neurocognition.

8.3 Transdisciplinary Impact

C³'s design enables direct application in:

- Music cognition research (resonance modeling, EEG studies)
- Therapy (trauma, memory, flow states)
- AI systems (neuro-informed generative music)

- Education (real-time student cognition tracking)
- Creative tools (resonance-driven composition)

It acts as a cognitive middleware between human experience and musical structure.

8.4 Toward a New Paradigm

C³ reframes music analysis as a dynamic, spatial, and measurable cognitive process. It suggests that:

- Musical meaning is not only symbolic, but resonant
- Cognition is not reactive, but synchronizing
- Emotion is not qualitative, but quantifiable
- Listening is not passive, but spatiotemporal entrainment

By building structure, signal, and spatiality into one coherent system, C³ proposes a new paradigm:

Not just analyzing music—but resonating with it, in real time, and across the brain.

8.5 What Comes Next?

C³ is not a finished system—it is a platform for growth.

- New Units can be defined
- New data sets can be mapped
- Neural signals can be updated via future modalities
- Interfaces can expand to immersive VR, generative AI, and cross-brain networks

With the foundation laid, the challenge is now collective elaboration: for researchers, artists, clinicians, and technologists to expand, iterate, and apply C³ across domains.

Final Note

The C³ system is open to research collaboration, institutional deployment, and creative integration. It was built not just to model cognition—but to enable new relationships between music, mind, and machine.

End of C³ Master Technical Report (Enhanced)

Thank you for your vision, precision, and trust in building this paradigm together.

grayThis document was designed using a dark theme with modular structure and scientific clarity.