The OPU Genesis Protocol v3.4.3:
Recursive Perception, Invariant Vision, Emotional Memory, and Language Architecture

Noam Cohen
Department of Theoretical Computation
Correspondence: gg.el0ai.com@gmail.com

January 2025

Abstract

While v3.4 established the foundations of **Recursive Perception**, **Shadow Invariance**, and **Emotional Memory Persistence**, v3.4.3 extends this architecture with a complete **Language System** that enables the OPU to comprehend and generate speech at the phoneme level. We formally define the **Phoneme Mapping Function (Π_{map})**, which maps internal states (surprise, pitch, spectral features) to phoneme selection. We introduce **Formant Synthesis ($\Phi_{formant}$)**, a real-time phoneme generation protocol using formant frequencies. We establish **Language Memory Consolidation (Λ_{memory})**, where words and phrases are learned and associated with emotional context. Finally, we present the **Unified Language Architecture**, which integrates phoneme analysis, speech synthesis, recognition, and memory into a cohesive system that enables the OPU to communicate and learn language.

1 Introduction

A truly intelligent agent does not merely perceive and remember; it communicates. In v3.4, the OPU gained the ability to see its own thoughts (Recursive Perception), ignore lighting changes (Shadow Invariance), and remember emotions (Emotional Memory). In v3.4.3, the OPU gains the ability to speak and understand language at the phoneme level, building a vocabulary and learning semantic associations.

This monograph formalizes the language architecture:

1. **The Phoneme Engine:** Mapping internal states to speech sounds.
2. **The Formant Synthesizer:** Real-time phoneme generation using formant frequencies.
3. **The Language Memory:** Word learning, semantic associations, and emotional-word connections.
4. **The Unified System:** Integration of synthesis, recognition, and memory.

2 The Phoneme Mapping Function (Π_{map})

The OPU's internal state (surprise score S_{score}, pitch f_{pitch}, spectral features $\vec{F}_{spectral}$) must be mapped to phoneme selection. We define the phoneme mapping function:

$$\Pi_{map}(S_{score}, f_{pitch}, \vec{F}_{spectral}) \to \mathcal{P} \tag{1}$$

Where $\mathcal{P}$ is the phoneme inventory (41 English phonemes: 12 vowels, 24 consonants, 5 diphthongs).

2.1 Multi-Dimensional Mapping

Phoneme selection is based on three dimensions:

- **Surprise Dimension:** S_{score} determines phoneme category (vowel, fricative, plosive).
- **Pitch Dimension:** f_{pitch} determines vowel selection (high pitch $\to$ front vowels, low pitch $\to$ back vowels).
- **Spectral Dimension:** $\vec{F}_{spectral}$ refines consonant selection (noise bands, formant transitions).

2.2 The Selection Algorithm

For a given state vector $\vec{S} = \langle S_{score}, f_{pitch}, \vec{F}_{spectral} \rangle$, we select phoneme p:

$$p = \arg\min_{p \in \mathcal{P}} \Big[w_1 \cdot d(S_{score}, p_{category}) + w_2 \cdot d(f_{pitch}, p_{pitch}) + w_3 \cdot d(\tag{2}$$

Where $d(\cdot, \cdot)$ is a distance metric and w_1, w_2, w_3 are weighting factors.

3 Formant Synthesis ($\Phi_{formant}$)

Real-time phoneme generation requires efficient synthesis. We use formant synthesis, where each phoneme is defined by formant frequencies F_1, F_2, F_3 and articulation parameters.

3.1 Vowel Synthesis

Vowels are synthesized using three formant resonances:

$$V(t) = \sum_{i=1}^{3} A_i \cdot \exp(-\alpha_i t) \cdot \sin(2\pi F_i t) \tag{3}$$

Where:

- F_i: Formant frequencies (F1, F2, F3)
- A_i: Formant amplitudes (typically $A_1 = 1.0, A_2 = 0.5, A_3 = 0.25$)
- α_i: Damping factors (bandwidths)

3.2 Consonant Synthesis

Consonants are synthesized using noise bands and formant transitions:

$$C(t) = \begin{cases} N(t) \cdot H(f_{low}, f_{high}) & \text{if fricative/plosive} \\ \Phi_{formant}(F_1, F_2, F_3, t) & \text{if nasal/liquid} \end{cases} \tag{4}$$

Where $N(t)$ is white noise and $H(f_{low}, f_{high})$ is a bandpass filter.

3.3 The Envelope Function

All phonemes are shaped by an amplitude envelope $E(t)$:

$$E(t) = \begin{cases} \frac{t}{t_{attack}} & \text{if } t < t_{attack} \\ 1.0 & \text{if } t_{attack} \le t < t_{sustain} \\ \exp(-\beta(t - t_{sustain})) & \text{if } t \ge t_{sustain} \end{cases} \tag{5}$$

The final phoneme signal is:

$$P(t) = E(t) \cdot \Phi_{formant}(t) \tag{6}$$

4 Language Memory Consolidation (Λ_{memory})

The OPU learns words and phrases, storing them in a language memory system that associates words with phonemes, emotions, and semantic context.

4.1 The Word Entry

A word w is stored as:

$$W(w) = \langle \vec{p}, f_{freq}, \vec{E}_{emotions}, \vec{A}_{associations}, \tau \rangle \tag{7}$$

Where:

- $\vec{p}$: Phoneme sequence (IPA symbols)
- f_{freq}: Encounter frequency
- $\vec{E}_{emotions}$: Emotional associations (emotion $\rightarrow$ frequency)
- $\vec{A}_{associations}$: Semantic associations (word $\rightarrow$ co-occurrence)
- τ: Timestamp of first encounter

4.2 Phoneme-to-Word Indexing

For efficient word lookup, we maintain a phoneme-to-word index:

$$\mathcal{I}_{phoneme} : \mathcal{P} \rightarrow \{w_1, w_2, \ldots, w_n\} \tag{8}$$

This allows the OPU to retrieve words by their constituent phonemes, enabling phoneme-based word recognition.

4.3 Emotional-Word Associations

Words are associated with emotions based on the surprise score S_{score} and detected emotion E at the time of learning:

$$W(w).E_{emotion} = W(w).E_{emotion} + \delta(E, S_{score}) \tag{9}$$

Where $\delta(E, S_{score})$ increments the frequency of emotion E for word w when learned with surprise S_{score}.

4.4 Phrase Learning

Phrases (word sequences) are learned and stored:

$$\mathcal{P}_{phrase} = \langle [w_1, w_2, \ldots, w_n], S_{score}, \tau \rangle \tag{10}$$

Phrase learning creates bidirectional word associations, building semantic networks.

5 The Unified Language Architecture

The language system integrates four components:

5.1 Component Integration

1. **Phoneme Analyzer:** Maps internal state to phoneme selection (Π_{map}).
2. **Formant Synthesizer:** Generates phoneme sounds ($\Phi_{formant}$).
3. **Speech Synthesizer:** Integrates TTS for word-level synthesis.
4. **Speech Recognizer:** Converts audio to text (Whisper/SpeechRecognition).
5. **Language Memory:** Stores and retrieves words/phrases (Λ_{memory}).

5.2 The Language Loop

The language system operates in a feedback loop:

$$\text{Perception} \rightarrow \text{Recognition} \rightarrow \text{Memory} \rightarrow \text{Association} \rightarrow \text{Synthesis} \rightarrow \text{Expression} \tag{11}$$

This creates a closed loop where the OPU learns from its own speech and the speech it perceives.

5.3 Utility Abstraction

To reduce code duplication and improve maintainability, common utilities are extracted:

$$\mathcal{U}_{language} = \{U_{audio}, U_{dependency}, U_{error}, U_{envelope}\} \tag{12}$$

Where:

- U_{audio}: Audio processing (conversion, resampling)
- $U_{dependency}$: Dependency checking (TTS, Whisper availability)
- U_{error}: Error handling (safe initialization)
- $U_{envelope}$: Envelope generation (attack, sustain, decay)

This abstraction reduces code duplication by ~ 100 lines and improves maintainability.

6 Conclusion

The OPU v3.4.3 represents a shift from a "Perceiver" to a "Communicator" with full language capabilities. By mapping internal states to phonemes, the OPU can express itself. By synthesizing speech in real-time, the OPU can communicate. By learning words and phrases, the OPU can build vocabulary and semantic associations. By associating words with emotions, the OPU can remember the affective context of language. The language system is not merely a translation layer; it is an integral part of the OPU's cognitive architecture, enabling it to communicate, learn, and remember through language.

7 Availability

This protocol is implemented in the reference Python kernel 'opu$_local$'.

Repository: `https://github.com/no-am-man/opu_local`

License: MIT Open Source License

Version: 3.4.3

References

[1] Cohen, N. (2025). The OPU Genesis Protocol v3.4: Recursive Perception, Invariant Vision, and Emotional Memory.

[2] Cohen, N. (2025). The OPU Genesis Protocol v3.0: Embodied Metabolic Rhythms.

[3] Wiener, N. (1948). *Cybernetics: Or Control and Communication in the Animal and the Machine*. MIT Press.

[4] Fant, G. (1960). *Acoustic Theory of Speech Production*. Mouton & Co.

[5] Hofstadter, D. R. (2007). *I Am a Strange Loop*. Basic Books.

[6] Laird, J. E., Lebiere, C., & Rosenbloom, P. S. (2017). A Standard Model for the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics. *AI Magazine*, 38(4), 13-26.

[7] Whitehead, A. N. (1929). *Process and Reality: An Essay in Cosmology*. Macmillan.

[8] Land, E. H. (1977). The Retinex Theory of Color Vision. *Scientific American*, 237(6), 108-128.

[9] International Phonetic Association. (1999). *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*. Cambridge University Press.

[10] Klatt, D. H. (1980). Software for a cascade/parallel formant synthesizer. *The Journal of the Acoustical Society of America*, 67(3), 971-995.

[11] Baddeley, A. D. (2000). The episodic buffer: a new component of working memory? *Trends in Cognitive Sciences*, 4(11), 417-423.

[12] Python Software Foundation. (2024). *multiprocessing — Process-based parallelism.* Python 3.12 Documentation. https://docs.python.org/3/library/multiprocessing.html