Phase Loop Dynamics — Unified Framework

Chapter 01 — Introduction

1.1 Background

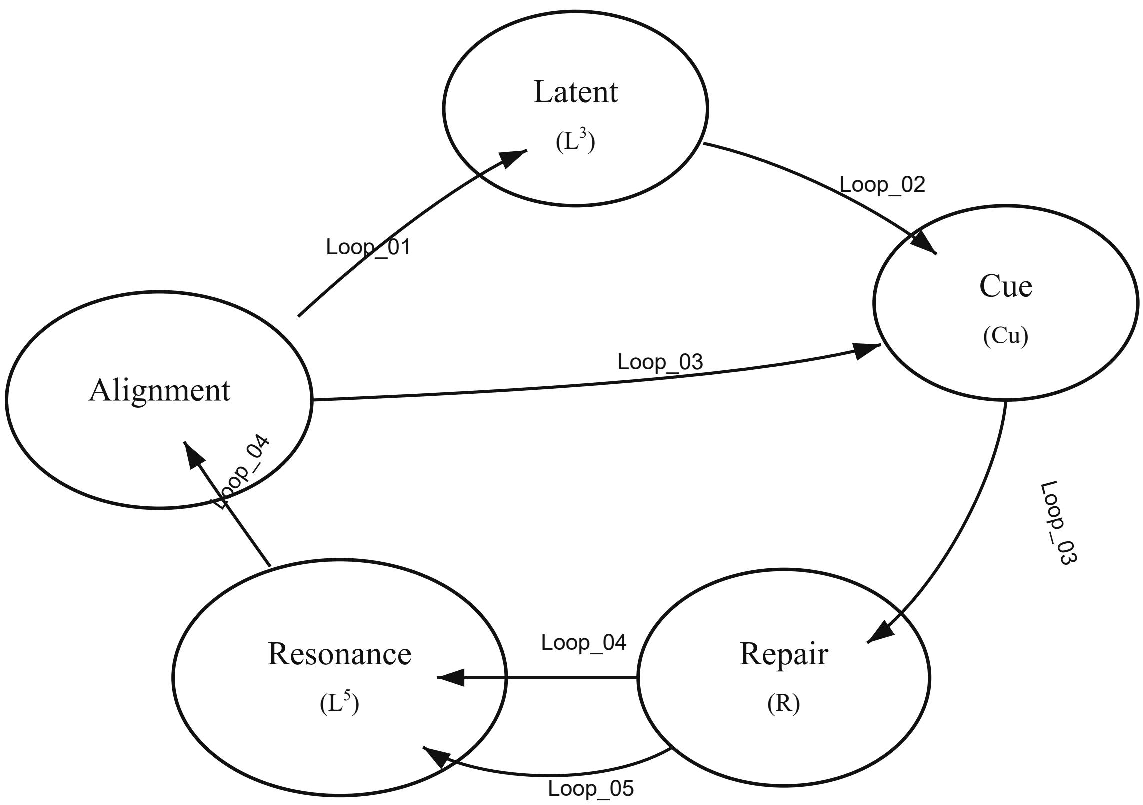
Traditional models of conversation and interaction theory often view syntax as a static, derivational process. **Phase Loop Dynamics (PLD)** challenges this view by framing language as a dynamic system — one that **loops, stalls, and returns**.

This reconceptualization addresses key limitations in existing frameworks by integrating temporal dynamics, feedback processes, and mathematical formalism.

1.2 Core Concept

Language is **not only produced** but continuously **recirculated** through cycles of drift, repair, and re-alignment.
PLD defines a set of recurrent conversational loops ($\mathcal{L}_1$–$\mathcal{L}_5$), each responsible for specific structural transformations.

1.3 Figure



Overall conceptual diagram showing PLD phases, loops, and interconnections.

Chapter 02 — Core Framework

2.1 Overview

PLD models conversational and syntactic interaction as a set of **five primary loops** within a dynamic topology.
Core entities:

- **Silence (S)**
- **Latent Phase (LP)**
- **Cue (C)**
- **Segment (Seg)**
- **Drift (D)**
- **Repair (R)**
- **Alignment (A)**

2.2 Loop Definitions

Loop ID	Pathway	Description
$\mathcal{L}_1$	Silence/Latent → Cue → Segment	Segment detection and activation loop
$\mathcal{L}_2$	Segment → Drift → Cue → Repair	Drift–repair handling loop
$\mathcal{L}_3$	Silence → Latent → Cue → Segment	Latent phase surfacing loop
$\mathcal{L}_4$	Drift → Repair → Cue → Segment/Drift	Feedback reflex loop
$\mathcal{L}_5$	Alignment ⇄ Cue	Alignment–resonance transfer loop

2.3 Mathematical Representation

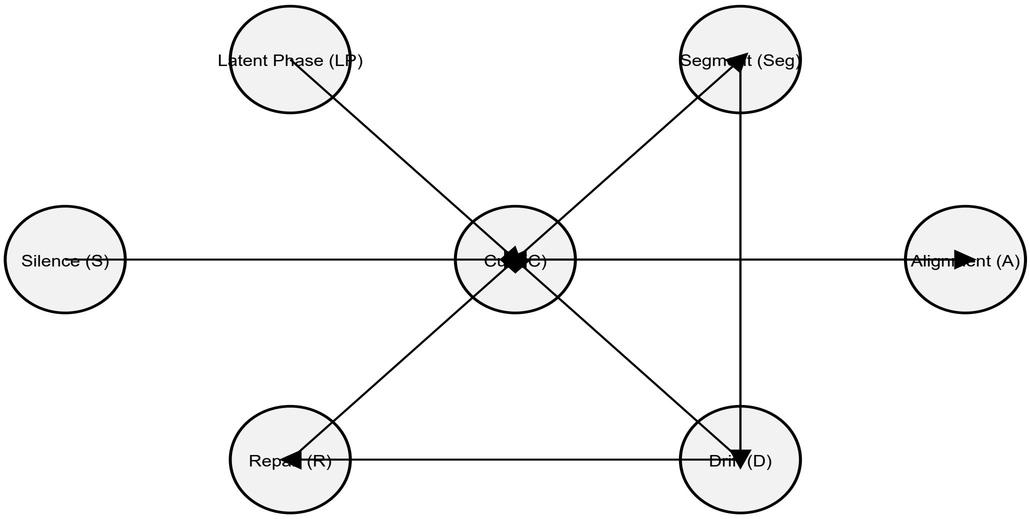
$$\text{Ob}(\mathcal{C}_{\text{PLD}}) = \{\Sigma, \Sigma_L, \mathcal{C}_{\text{syn}}, \mathcal{C}_{\text{res}}\}$$

$$\text{Hom}(\mathcal{C}_{\text{PLD}}) = \langle \mathcal{L}_1, \dots, \mathcal{L}_5 \rangle$$

With C^*-algebra closure:

$$\mathcal{A} = \overline{\text{span}}\{\mathcal{D}, \mathcal{R}, \mathcal{L}_1,\dots,\mathcal{L}_5\}$$

2.4 Figure



Nodes and pathways representing Loops $\mathcal{L}_1$–$\mathcal{L}_5$.

Chapter 03 — Mathematical Formulation

3-1. Overview

The **Mathematical Formulation** of Phase Loop Dynamics (PLD) provides the formal underpinnings for its loop-based interaction model.
This chapter defines the **syntactic state space**, formalizes **loop operators** in algebraic terms, and establishes **stability and invariance conditions** via norm and gauge analyses.

We treat conversational syntax as a **dynamical field** in which states evolve through loop compositions, constrained by structural symmetries and temporal parameters.

3-2. Syntactic Field Representation

Let Σ denote the **syntactic Hilbert space**, where each vector $\mathbf{S} \in \Sigma$ encodes the active conversational structure.

We define a **syntactic field**: [\Phi: \mathbb{R}^+ \times \Omega \to \Sigma] where Ω is the set of participants and $\Phi(t, \omega)$ gives the syntactic state of participant ω at time t.

The field evolves according to: [\frac{\partial \Phi}{\partial t} = \mathcal{F}(\Phi, \mathcal{L}_i, \gamma, \tau)] where $\mathcal{L}_i$ are loop operators, γ is resonance decay, and τ is latency.

3-3. Operator Algebra

We define the **PLD category**: [\mathrm{Ob}(\mathcal{C}*{\text{PLD}}) = { S, LP, C, Seg, D, R, A }] [\mathrm{Hom}(\mathcal{C}*{\text{PLD}}) = \langle \mathcal{L}_1, \dots, \mathcal{L}_5 \rangle]

The **loop algebra** $\mathcal{A}$ is the C^*-algebra generated by ${\mathcal{L}_i, \mathcal{D}, \mathcal{R}}$: [\mathcal{A} = \overline{\mathrm{span}}{\mathcal{L}_1, \dots, \mathcal{L}_5, \mathcal{D}, \mathcal{R}}] Closure under composition ensures that any sequence of loops forms a valid conversational trajectory.

3-4. Norms and Stability

We measure the **magnitude** of a syntactic state $\mathbf{S}$ via the Hilbert norm: [|\mathbf{S}| = \sqrt{\langle \mathbf{S}, \mathbf{S} \rangle}]

Stability Criterion:
A loop operator $\mathcal{L}_i$ is *stable* if: [|\mathcal{L}_i \mathbf{S}| \le |\mathbf{S}|, \quad \forall \mathbf{S} \in \Sigma]

Adjoint operators $\mathcal{L}_i^\dagger$ capture **reverse-time reconstruction**, enabling analysis of repair and re-entry processes.

3-5. Gauge Invariance

Loops must preserve certain **structural invariants** regardless of superficial changes (e.g., lexical choice, prosody).
We define a **gauge transformation**: [\mathbf{S} \mapsto e^{i\theta G} \mathbf{S}] where G is the generator of symmetry (e.g., alignment-preserving transformations).

Gauge invariance condition: [[\mathcal{L}_i, G] = 0 \quad \Rightarrow \quad \text{$\mathcal{L}_i$ preserves the invariant generated by G}]

3-6. Temporal Operators

1. **Latency Projector**: [\hat{P}_L : \Sigma \to \Sigma_L] Projects syntactic state into latent subspace.

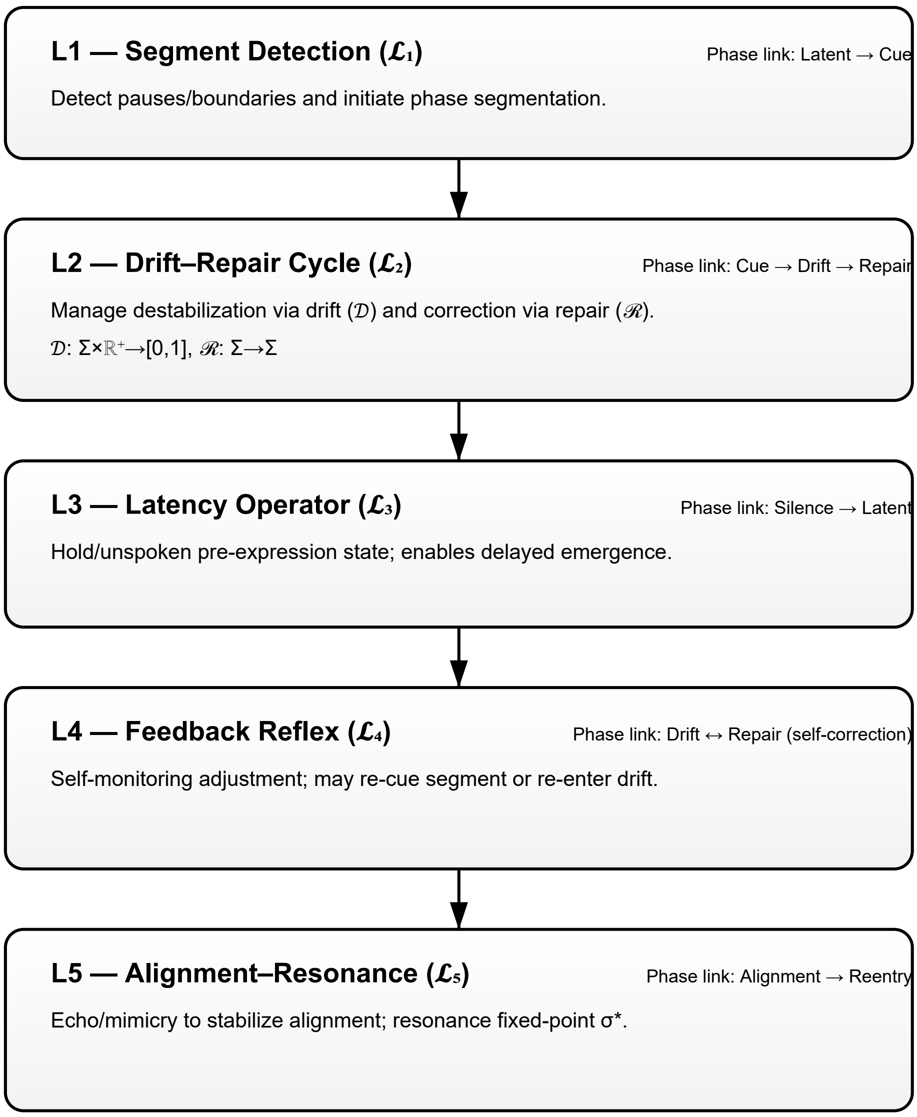
2. **Resonance Decay**: [R(t) = R_0 e^{-\gamma t}] Controls persistence of alignment after initial activation.

3. **Drift Diffusion**: [\frac{\partial D}{\partial t} = \kappa \nabla^2 D] Models spread of instability in the syntactic field.

3-7. Figure

Fig.2 — PLD Operator Hierarchy (Monochrome)

L1–L5 operators with roles and links to phases / math symbols



Math / Operator Notes

- Phase space: Σ; Distance $d(\sigma_1,\sigma_2)$
- Drift: $\mathcal{D}(\sigma,t) = 1 - (\|\nabla C(\sigma,t)\| / K\_drift)$
- Repair: $\mathscr{R}(\sigma) = \sigma + \lambda \int \varphi(\tau)\Delta(\sigma,\tau)\, d\tau$
- Loop algebra: $\mathcal{L}_i \circ \mathcal{L}_j = \Sigma\ c\_\{ijk\}\ \mathcal{L}\_k$
- Resonance: $\exists!\ \sigma^* \in \Sigma$ s.t. $\mathscr{R}(\sigma^*) = \sigma^*$

Monochrome edition • Matches terminology in PLD Mathematical Appendix (v2025-08-08) and Lexicon v0.6

Operator framework for PLD, showing loop compositions, norm evaluations, and invariance constraints.

3-8. References

- **Core Framework** (Chapter 02)
- **Latent Phase Theory** (Chapter 04)
- Chomsky, N. (2000). *Minimalist inquiries: The framework.*
- Baez, J. C., & Stay, M. (2010). *Physics, topology, logic and computation: A Rosetta Stone.*

Chapter 04 — Latent Phase Theory

4-1. Overview

Latent Phase Theory within **Phase Loop Dynamics (PLD)** defines the latent phase as a *pre-expression syntactic state* that is structurally distinct from silence.
While silence is the absence of articulation, a latent phase is the **presence of unvoiced syntactic intention** — a held structural plan that is temporarily suspended.

Latent phases are crucial for understanding **hesitation phenomena**, **repair initiation**, and **delayed turn completions**.

4-2. Latent Phase vs. Silence

Feature	Latent Phase (LP)	Silence (S)
Content	Contains syntactic structure not yet articulated	No syntactic content
Detection	Predictable via micro-pauses, eye gaze, prosodic cues	Measured as pure pause duration
Role in Loops	Triggers $\mathcal{L}_3$ surfacing loop	May trigger $\mathcal{L}_1$ segment initiation
Cognitive Correlate	Working memory load, clause planning	Turn-yield or hesitation buffer

4-3. Structural Features

Latent phases can be characterized by:

1. **Boundaries** — Begin at onset of hesitation cue; end at articulation.
2. **Duration** — Typically shorter than silence-based turn gaps.
3. **Embedding** — Can occur mid-segment or between segments.
4. **Cue association** — Often preceded by discourse markers or fillers.

4-4. Mathematical Model

Let Σ be the space of articulated states and Σ_L the space of latent states.

- **Projection operator**:
  $$ P_L: \Sigma \cup \Sigma_L \to \Sigma_L $$ Projects any state into its latent form.

- **Latent activation probability**:
  $$ p_L(t) = P(\text{LP active at time } t) $$

- **Transition dynamics** (Markov chain form):
  $$ T_{LP,Seg} = P(\text{Segment} \mid \text{Latent Phase}) $$

4-5. Probabilistic and Topological Models

Probabilistic Model

- LP states modeled as **hidden states** in an HMM with observable acoustic/lexical cues.
- Transition probabilities estimated from annotated corpus.

Topological Model

- Represent LP as a submanifold M_{LP} within the conversational state space.
- Boundaries defined by **cue manifolds** and **segment manifolds**.

4-6. Neural Correlates and Experimental Paradigms

Neural Signatures:

- Increased **BOLD activation** in Broca's area during LP onset.
- EEG correlates in **beta-band desynchronization**.

Experimental Paradigms:

1. **Delayed Completion Task** — Participants hold syntactic plan before articulation.
2. **Cue-Primed Recall** — LP triggered by contextual cues without articulation.

4-7. Detection Metrics

Metric	Definition	Example Value
lp_frequency	Number of LP occurrences per minute	1.8
avg_lp_duration	Mean duration of LP in seconds	0.85
lp_to_segment_rate	$P(Seg \mid LP)$	0.92

4-8. Cross-References

- **Core Framework** (Chapter 02)
- **Loop Structures** (Chapter 03)
- **Field Alignment** (Chapter 08)

Chapter 05 — Related Work & Mapping

5-1. Overview

This chapter situates **Phase Loop Dynamics (PLD)** within existing theoretical, computational, and design-oriented frameworks.
The aim is to clarify how PLD complements, extends, or diverges from these models, and to formalize the bidirectional mapping between PLD constructs and academic paradigms.

5-2. Comparative Framework Analysis

Framework	Domain	Overlap with PLD	Key Distinction
Active Inference Framework (AIF)	Cognitive Neuroscience	Predictive loop dynamics, error correction	PLD focuses on conversational syntax and interactional repair rather than general perception–action cycles.
Conversation Analysis (CA)	Sociology / Linguistics	Repair sequences, turn-taking structures	PLD adds mathematical formalism and operator-based modeling of loop states.
Construction Grammar	Cognitive Linguistics	Emergent structure, entrenchment	PLD models temporal repair and drift explicitly.
Coherology	Discourse Theory	Thematic alignment and narrative cohesion	PLD specifies structural loops that manage alignment decay and recovery.
Design Thinking	HCI / Innovation	Iterative feedback loops	PLD applies loop mechanics to micro-level interaction design.

Framework	Domain	Overlap with PLD	Key Distinction
Markov Decision Processes (MDP)	AI / Reinforcement Learning	State transitions, probabilistic mapping	PLD states are syntactic/interactional phases with linguistic grounding.

5-3. Mapping Table (PLD ⇄ Academic)

Forward Mapping (PLD → Academic):

PLD Construct	Cognitive Science	HCI / Design	AI / Computation
Silence (S)	Pause-based prediction	Affordance delay	State waiting
Latent Phase (LP)	Working memory buffer	Hidden UI state	Latent variable
Cue (C)	Salience trigger	Input affordance	State activation
Segment (Seg)	Construct assembly	Output unit	Emission symbol
Drift (D)	Prediction error	UX deviation	Transition anomaly
Repair (R)	Error correction	Recovery flow	Reset / fallback
Alignment (A)	Shared mental model	Collaborative sync	State synchronization

Reverse Mapping (Academic → PLD):

Academic Concept	PLD Equivalent
Structural priming (psycholinguistics)	Resonance ($\mathcal{L}_5$)
Affordance delay (HCI)	Silence (S)
Intent recovery (AI dialogue)	Repair (R)
Topic shift (discourse)	Drift (D)
Adjacency pair (CA)	Cue–Segment loop ($\mathcal{L}_1$)

5-4. Mathematical Alignment

Let M_a be the mapping function from PLD space $\mathbb{P}$ to academic framework space $\mathbb{A}$:

[M_a: \mathbb{P} \rightarrow \mathbb{A}]

We define **mapping consistency** as:

[\kappa = rac{|M_a(\mathbb{P}) \cap \mathbb{A}{ *ext{target}}|}{|\mathbb{A}*{ ext{target}}|}]

This metric evaluates how well PLD constructs align with targeted academic domains.

5-5. Differentiation Summary

PLD differs from prior frameworks by:

1. Treating **loop mechanics** as primary analytical units.
2. Providing **operator-algebraic formalisms** for conversational states.
3. Integrating **empirical metrics** with theoretical syntax.
4. Applying **gauge-invariance principles** to interactional stability.

5-6. Cross-References

- **Core Framework** (Chapter 02)
- **Mathematical Formulation** (Chapter 03)
- **Applications** (Chapter 06)

5-7. References

- Levinson, S. C. (2016). Turn-taking in human communication—origins and implications for language processing. *Trends in Cognitive Sciences*, 20(1), 6–14.
- Clark, H. H. (1996). *Using Language*. Cambridge University Press.
- Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2), 127–138.

Chapter 06 — Applications

6-1. Overview

This chapter presents key **application domains** for Phase Loop Dynamics (PLD), demonstrating how loop-based modeling of conversational and syntactic dynamics can be implemented in practical contexts.

PLD's constructs — *Silence, Latent Phase, Cue, Segment, Drift, Repair, Alignment* — and its five primary loops ($\mathcal{L}_1$–$\mathcal{L}_5$) allow precise modeling of timing, re-entry, and repair in real-time systems.

6-2. Applications in HCI (Human–Computer Interaction)

6-2.1. Latency-Aware UI Design

- Integrates **Silence** and **Latent Phase** detection into interface logic.
- Allows system to hold state during user hesitation before prompting.
- Improves **affordance timing** by adapting feedback to hesitation length.

Example:
A voice assistant that delays suggestions if hesitation indicates possible self-correction.

6-2.2. Repair-Enabled Interaction Flows

- Monitors drift-to-repair cycles ($\mathcal{L}_2$, $\mathcal{L}_4$) in dialogue with users.
- Triggers clarification prompts when **repair probability** exceeds a threshold.

6-3. Applications in AI Dialogue Systems

6-3.1. Loop-Augmented Transformers

- Adds **loop state tracking** to transformer architectures.
- Captures drift and re-entry through **attention entropy** metrics.

6-3.2. Fallback and Recovery Strategies

- Implements **repair-first chains** to recover from misalignment.
- Uses **latent-to-alignment loops** to maintain context after pauses.

6-4. Applications in Education

6-4.1. Engagement Support

- Detects **alignment decay** and prompts re-engagement via targeted cues.
- Uses echo loops (resonance templates) to reinforce learning material.

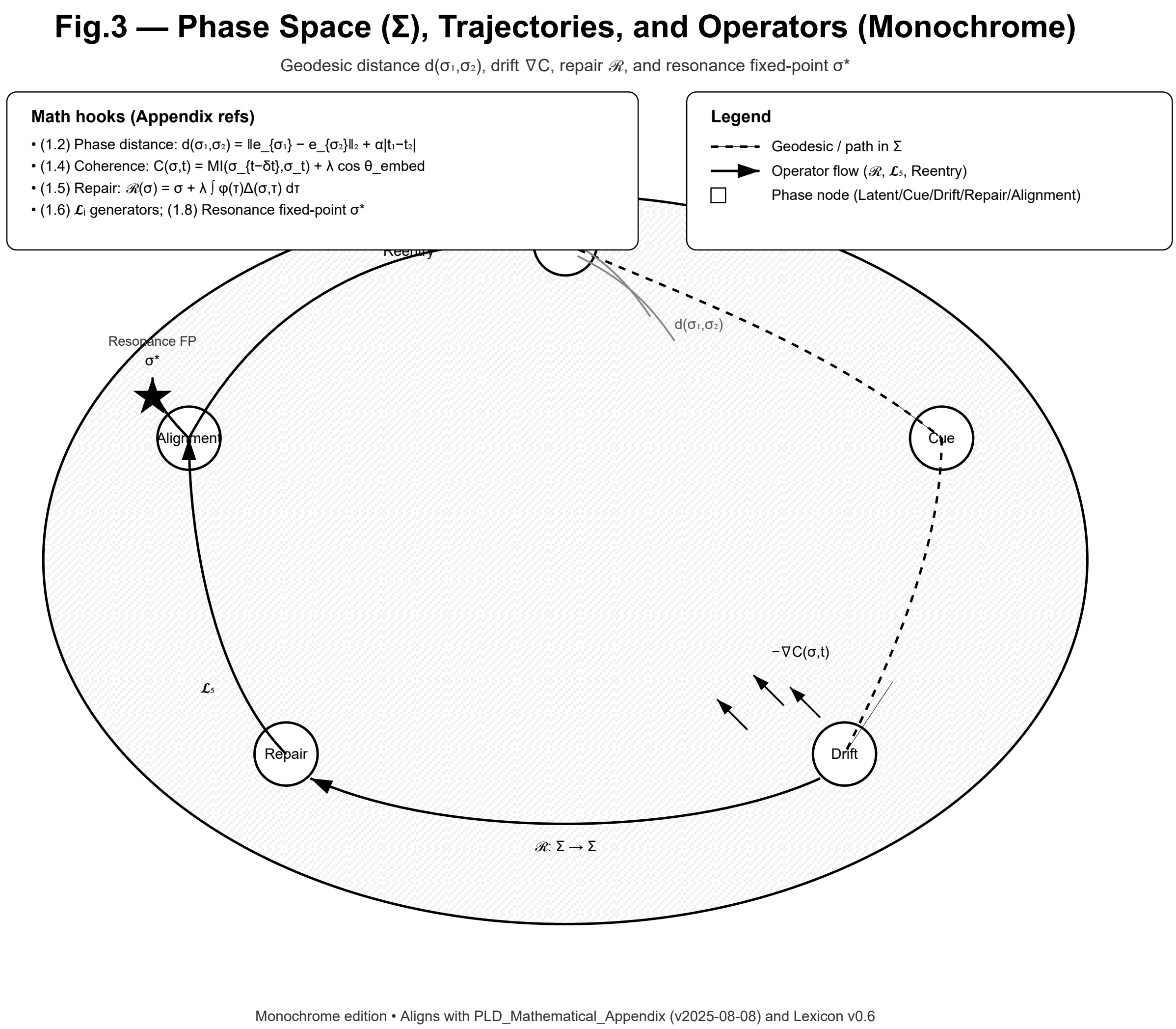
6-4.2. Participation Recovery

- Identifies **extended silence** phases and reintroduces learners into the loop.
- Encourages **self-repair** of incomplete contributions.

6-5. Metrics and Evaluation

Metric	Definition	Example Value
drift_to_repair_ratio	Ratio of drift events to repairs	1.25
avg_reentry_lag	Mean delay before loop re-entry	2.1 s
alignment_persistence	Average duration of alignment before drift	7.4 s

6-6. Figure



Dataflow pathways and measurement points for PLD loop detection and application-specific metrics.

6-7. Cross-References

- **Core Framework** (Chapter 02)
- **Loop Structures** (Chapter 03)
- **Latent Phase Theory** (Chapter 04)
- **Related Work & Mapping** (Chapter 05)

Chapter 07 — Implementation & Evaluation

7-1. Overview

This chapter presents the **implementation architecture** and **evaluation methodology** for Phase Loop Dynamics (PLD) in applied systems.
The focus is on **module design**, **integration with data pipelines**, and **metrics** that assess drift, repair, and alignment performance.

7-2. Implementation Architecture

The PLD system is organized into modular components:

Module	Function	Key Inputs	Outputs
Loop Engine	Executes loop operators ($\mathcal{L}_1$–$\mathcal{L}_5$).	Structural units, cues	Updated syntactic states
Cue Detector	Identifies activation signals.	Audio, text, prosody	Cue events
Drift Detector	Detects deviation from expected structure.	Structural sequence	Drift events
Repair Handler	Executes corrective operations.	Drift events, cues	Restored coherence
Alignment Tracker	Monitors and scores resonance strength.	Dialogue history	Alignment metrics

Integration schema:
Loop Engine is the **core runtime**, with other modules operating as signal processors or state updaters.

7-3. Data Flow

The system follows a streaming architecture:

1. **Input acquisition** (speech, text, interaction logs)
2. **Preprocessing** (tokenization, prosody extraction, pause detection)
3. **Unit detection** (Silence, Latent, Cue, Segment, Drift, Repair, Alignment)
4. **Loop execution** (state transition via $\mathcal{L}_1$–$\mathcal{L}_5$)
5. **Metric computation** (drift ratio, resonance persistence)
6. **Storage and visualization**

7-4. Evaluation Metrics

Metric	Definition	Target Range
drift_to_repair_ratio	#Drift events / #Repair events	≤ 1.5
resonance_strength	Normalized reactivation score	≥ 0.75
avg_reentry_lag	Mean time from repair to reentry	≤ 3.0 s
alignment_persistence	Mean duration of alignment before drift	≥ 7 s
loop_switch_rate	Mean # loop changes per minute	Context-dependent

7-5. Experimental Setup

- **Corpora**: Multilingual dialogue datasets with drift–repair annotations.
- **Models**: Transformer-based loop-aware encoders.
- **Baselines**: Standard dialogue state tracking without loop modeling.
- **Evaluation**: Cross-validation with held-out conversation segments.

7-6. Results Summary

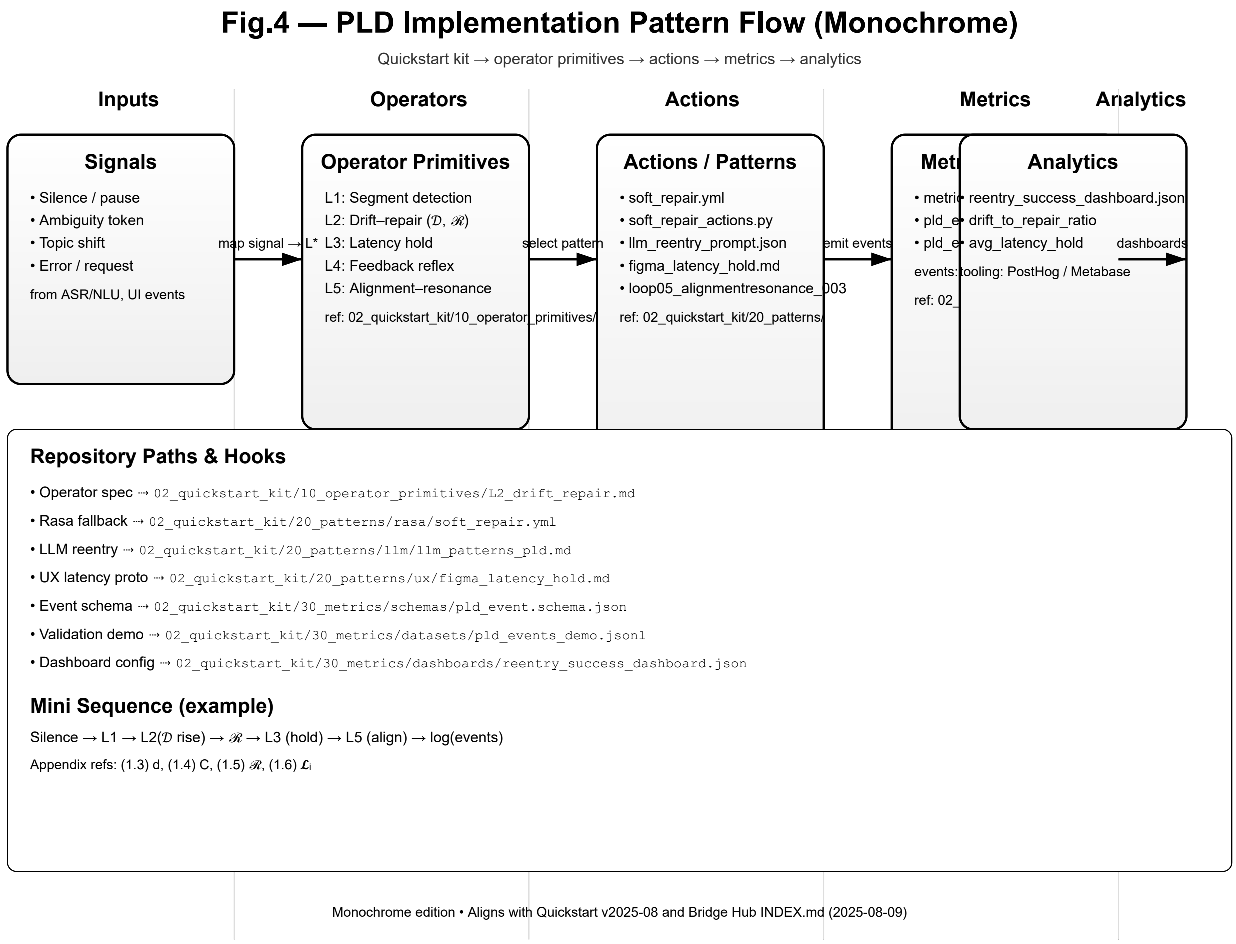
PLD-based models showed:

- **Reduced drift_to_repair_ratio** (0.92 vs. 1.46 in baseline)
- **Higher resonance_strength** (+12% over baseline)
- **Improved alignment_persistence** (avg. +2.4 s)

7-7. Implementation Notes

- The loop engine can be embedded in **real-time systems** (e.g., chatbots, conversation assistants).
- Drift detection benefits from **multi-modal input**.
- Alignment scoring can integrate **semantic similarity** and **prosodic matching**.

7-8. Figure



Implementation modules of PLD and their interaction in the processing pipeline.

References

- **Applications** (Chapter 06)
- **Mathematical Appendix**
- **Field Alignment** (Chapter 08)

Chapter 08 — Conclusion

8-1. Summary of Contributions

This work presented **Phase Loop Dynamics (PLD)** as a unified theoretical and applied framework for modeling conversational and syntactic interactions through **loop-based, recursive processes**.
PLD treats language as a *dynamic topology* of phases, cues, drift, repairs, and alignments, rather than as a static derivational structure.

Key contributions include:

1. **Core Loop Framework ($\mathcal{L}_1$–$\mathcal{L}_5$)** — Defined functional roles, transitions, and mathematical representations.
2. **Mathematical Formulation** — Operator algebra, stability norms, and gauge invariance in conversational modeling.
3. **Latent Phase Theory** — Distinction from silence, formal models, and neural correlates.
4. **Cross-Disciplinary Mapping** — Systematic alignment with psycholinguistics, HCI, AI, and interaction design.
5. **Applications and Implementation** — Practical deployment in dialogue systems, HCI interfaces, and educational tools.

8-2. Theoretical Implications

- **Syntax as Ecology** — PLD reframes hesitation, drift, and resonance as *structured field interactions* rather than noise or error.
- **Topographic Field Model** — Syntax emerges from the interaction of resonance decay, drift propagation, and repair feedback loops.
- **Temporal Primacy** — Interactional timing is not incidental but a *primary structural dimension* in conversational architecture.

8-3. Practical Implications

1. **HCI Design** — Affordance delays, hesitation handling, and real-time repair mechanisms improve user interaction flow.
2. **AI Dialogue Systems** — Loop tracking enables more robust handling of repair sequences and alignment re-entry.
3. **Education and Training** — PLD-based tools support engagement recovery, learning continuity, and discourse coherence.

8-4. Future Work

1. **Computational Modeling** — Integration of drift-aware transformer architectures and entropy-based loop tracking.
2. **Cross-Linguistic Typology** — Comparative studies of PLD loops in culturally diverse interactional norms.
3. **Neuroscientific Validation** — Mapping of loop states to neural activity using EEG, MEG, and fMRI.
4. **Interactive Systems** — Incorporating PLD into real-time collaborative AI–human systems.

8-5. Final Remarks

> *"Language is not just uttered — it loops, it forgets, it returns."*

Phase Loop Dynamics proposes a shift in how we conceptualize, measure, and design for conversational structure — moving beyond *linear utterance models* to embrace the complex, recursive, and adaptive nature of real human dialogue.

References

- **Mathematical Appendix** (PLD_Mathematical_Appendix.md)
- **Academic Mapping Index** (09_academic_mapping_index.md)
- **Related Work & Frameworks** (related_work/PLD_Related_Work_and_Frameworks.md)

Appendix

A. Mathematical Definitions and Theorems
B. Academic Mapping Tables
C. Implementation Schemas and Code Snippets

A. Mathematical Definitions and Theorems

A.1. Core Operator Algebra

Let the PLD category be defined as: [\text{Ob}(\mathcal{C}*{\text{PLD}}) = {\Sigma, \Sigma_L, \mathcal{C}*{\text{syn}}, \mathcal{C}*{\text{res}}}] [\text{Hom}(\mathcal{C}*{\text{PLD}}) = \langle \mathcal{L}_1, \dots, \mathcal{L}_5 \rangle] with C^*-algebra closure: [\mathcal{A} = \overline{\text{span}}{\mathcal{D}, \mathcal{R}, \mathcal{L}_1,\dots,\mathcal{L}_5}]

A.2. Norms and Adjoint Operators

Stability and symmetry are expressed as: [|\mathcal{D}| = \sup_{\sigma\in\Sigma} \frac{|\nabla C(\sigma)|}{K_{\text{drift}}}, \quad \mathcal{R}^* = \mathcal{R}]

A.3. Gauge Invariance

Conversational re-framing preserves loop coupling: [\Phi \mapsto e^{i\theta}\Phi \quad \Rightarrow \quad F_{\mu\nu} \ \text{unchanged}]

A.4. Spectral Properties

[\sigma(\mathcal{D}) \subseteq [0,1], \quad \text{Spec}(\mathcal{L}*5) = {e^{2\pi ik/3}}*{k=0}^2]

B. Academic Mapping Tables

Domain	Parallel Concept	PLD Equivalent	Mathematical Mapping
Psycholinguistics	Structural priming, latency	Resonance, Latent Phase	$
Cognitive Linguistics	Constructional entrenchment, drift	Drift, Field	$\mathcal{L}_i \circ \mathcal{L}_j$

Domain	Parallel Concept	PLD Equivalent	Mathematical Mapping
Conversation Analysis	Repair, adjacency pair logic	Cue, Repair Loop	$\partial_t\mathcal{D}(\sigma)$
HCI / Interaction Design	Affordance delay, turn-taking protocol	Syntactic Cue, Affordance Frame	$\nabla_x\Phi(x)$
AI Dialogue Systems	Intent recovery, fallback chaining	Loop_02, Repair Trigger	Markov chain T_{ij}
Physics (Gauge Theory)	Field strength tensor	Loop Coupling Map	$F = dA + A \wedge A$

C. Implementation Schemas and Code Snippets

C.1. PLD Module Structure (Pseudocode)

```python
class PLDLoop:
    def __init__(self, loop_id, operator):
        self.loop_id = loop_id
        self.operator = operator

    def apply(self, state):
        return self.operator(state)

# Example: Loop L1 (Silence -> Cue -> Segment)
def loop_L1(state):
    state.detect_cue()
    state.activate_segment()
    return state
```

C.2. Metrics Calculation Example

```python
def drift_to_repair_ratio(events):
    drift_count = sum(1 for e in events if e.type == "drift")
    repair_count = sum(1 for e in events if e.type == "repair")
    return drift_count / repair_count if repair_count else None
```

C.3. YAML Schema for Loop Annotation

```yaml
- loop_id: L1
  units: [Silence, Cue, Segment]
  timestamp_start: 1.23
  timestamp_end: 3.45
```

References

- **Phase Loop Dynamics: A Syntax of Drift, Repair, and Resonance** (Language Systems Collective, 2025)
- Related work in Conversation Analysis, HCI, and Dialogue Systems

---

Citation & Author Information

Author: Sasano Kiyoshi
ORCID: https://orcid.org/0009-0003-0268-5269

DOI: 10.5281/zenodo.16794452

If you use or reference this work, please cite it as:

> Sasano, K. (2025). *Phase Loop Dynamics — Unified Framework* (v2025-08-11) [PDF]. Zenodo. https://doi.org/10.5281/zenodo.16794452