

UTAC Module M-β — Zyklische Emergenz: Steilheitsgrad als Potenzial → Bedingung

These. Der Steilheitsgrad β einer Schwellenantwort ist ein *akkumulierendes Potenzial*. Überschreitet das getriebene Feld $R(t)$ die adaptive Schwelle $\Theta(t)$, manifestiert sich ein neuer Systemzustand ψ . Das vormals latente Potenzial (β) wird damit zur **Bedingung** (Systemkomponente), die die künftige Feldtopologie (und damit β, Θ) der übrigen Potenziale aktiv moduliert. Es entsteht eine **kaskadierende Kohärenzschwelle**.

1. Executive Summary

- **Potenzial** \triangleq Steilheit β einer sigmoiden Reaktion.
 - **Manifestation** bei $R > \Theta$: Übergang von Potenzial → **Bedingung**.
 - **Rückkopplung**: Neue Bedingung verschiebt Θ , moduliert β , verändert die Feldtopologie → bereitet den nächsten Sprung vor.
 - **Universelle Form**: $\psi = \sigma(\beta(t) [R(t) - \Theta(t)])$, mit dynamischem β, Θ .
 - **Testbarkeit**: Vorläufer-Signaturen: ansteigende lokale β , adaptive Θ -Verschiebungen, Clusterbildung von Sprungereignissen.
-

2. Formale Grundgleichungen

2.1 Schwellenfeld-Antwort

$$\psi(t) = \sigma(\beta(t) [R(t) - \Theta(t)]) = \frac{1}{1 + e^{-\beta(t)(R(t) - \Theta(t))}}.$$

2.2 Adaptive Schwelle (Kontextplastizität)

$$\Theta(t) = \Theta_0 + \alpha S(t) + \sum_k \eta_k \Delta_k(t),$$

mit Systemkomplexität S (z. B. Kopplungsdichte), lokalen Ereignissen Δ_k (Emergenzen) und Gewichten α, η_k .

2.3 Dynamik der Steilheit (Potenzialakkumulation)

$$\beta(t + \Delta t) = \beta(t) + \kappa_{\uparrow} \Phi_{\text{Ladung}}(t) - \kappa_{\downarrow} \Phi_{\text{Entladung}}(t).$$

- **Ladung** (Potenzialaufbau): Kopplungszunahme, Stress, Dichte, Redundanzabbau, Resonanzen. -
Entladung (nach Manifestation): Sättigung, Refraktärzeit, Stabilisierung.

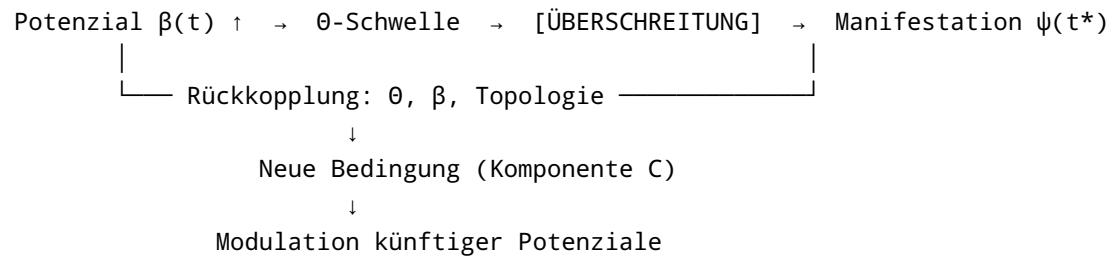
2.4 Ereignis-Update (Potenzial → Bedingung)

Bei **Manifestation** zum Zeitpunkt t^* (z. B. ψ überschreitet Output-Schwelle ψ_c):

$$\mathcal{C} \leftarrow \mathcal{C} \cup \{E\}, \quad \Theta \mapsto \Theta + \Delta\Theta(E, \mathcal{C}), \quad \beta \mapsto (1 - \rho_{\text{damp}}) \beta + \rho_{\text{amp}} G(E, \mathcal{C}),$$

mit Komponentenmenge \mathcal{C} , Dämpfung $\rho_{\text{damp}} \in [0, 1]$, Verstärkung $\rho_{\text{amp}} \in [0, 1]$, und G als Verstärkungsfunktion (z. B. neue Freiheitsgrade).

3. Zyklusdiagramm (Kaskadierende Kohärenz)



Intuition: Potenziale „schaffen“ sich durch Manifestation den Raum, weitere Potenziale zu entfalten; das System lernt seine eigene Topologie.

4. Falsifizierbare Vorhersagen (Domänenübergreifend)

- 1) **LLM-Grokking:** Vor einem plötzlichen Generalisierungssprung steigt die lokal geschätzte β in loss→accuracy-Sigmoiden; nach dem Sprung sinkt β in der betroffenen Teilaufgabe (Sättigung), während β in angrenzenden Aufgaben ansteigt (Potenzialverlagerung).
- 2) **Klimakipp-Elemente:** In proximalen Indikatoren (z. B. AMOC-Proxy, Albedo-Indizes) erhöht sich β der Zustands-Transitionskurven vor dem Kippunkt; nach Teilkippungen verschiebt sich Θ der restlichen Subfelder messbar (Resilienzverlust/Gewinn je nach Intervention).
- 3) **Neuroplastik bei Sensorikverlust:** Nach Wegfall einer Sinnesmodalität sinkt R_{Info} und der Feldkern destabilisiert (Orientierungsverlust). Messbar ist ein **temporärer Abfall** lokaler β in Netzwerken, gefolgt von **Steigerungen** in Kompensationsarealen (Rekalibrierung) und einer **Θ -Verschiebung** (neue Sensitivitätsschwellen).
- 4) **Genetische Sprünge (Transposons/Stress):** Unter Stressbedingungen steigen β -Schätzer in Mutations/Expression-Response-Kurven in relevanten Modulen; Manifestation (fixierte Mutation) reduziert lokale β , erhöht aber in gekoppelten Pfaden (Exaptation) β .
- 5) **Sozialdynamik (Adoptionskaskaden):** Steilheit der Adoptionskurve in Clustern steigt vor der Diffusion; nach Cluster-Sättigung verschiebt sich Θ in Nachbarclustern (niedrigere Aufnahmehürde durch soziale Evidenz).

Falsifikation: Finden sich stabile Übergänge ohne Vorläufer- β -Anstieg *und* ohne nachgelagerte Θ -Verschiebung, widerspricht das der Kaskadenhypothese.

5. Minimal-Simulation (Python-Skizze)

Datei-Vorschlag: `empirical/sim/steeplness_cascade.py`

```
import numpy as np

sigmoid = lambda x: 1.0/(1.0 + np.exp(-x))

class Cascade:
    def __init__(self, beta=1.0, theta=0.0, k_up=0.02, k_dn=0.15,
                 rho_damp=0.4, rho_amp=0.3):
        self.beta = beta
        self.theta = theta
        self.k_up = k_up
        self.k_dn = k_dn
        self.rho_damp = rho_damp
        self.rho_amp = rho_amp
        self.components = []

    def step(self, R, charge, discharge):
        # Potenzialdynamik
        self.beta += self.k_up*charge - self.k_dn*discharge
        y = sigmoid(self.beta*(R - self.theta))
        manifested = (y > 0.9) # Output-Schwelle  $\psi_c$ 
        if manifested:
            self.components.append({'t': None, 'beta': self.beta, 'theta': self.theta})
            # Ereignis-Update: Potenzial  $\rightarrow$  Bedingung
            self.theta += 0.15 # 0-Verschiebung (Beispiel)
            self.beta = (1 - self.rho_damp)*self.beta + self.rho_amp*(1 +
0.1*len(self.components))
        return y, manifested

# Demo
if __name__ == "__main__":
    np.random.seed(7)
    sim = Cascade(beta=1.2, theta=0.4)
    R = 0.0
    traj = []
    for t in range(400):
        R += 0.01 # langsamer Treiber
        charge = 0.8 + 0.2*np.sin(0.05*t)
        discharge = 0.2
        y, m = sim.step(R, charge, discharge)
        traj.append((t, R, sim.beta, sim.theta, y, m))
    # Save to CSV in empirical/sim/out/
```

Erwartung: Mehrere Manifestationen ($y \rightarrow 1$) bei ansteigendem Treiber R , jeweils begleitet von sprunghaften Θ -Verschiebungen und lokalen Reset/Anhebungen von β ; kaskadierende Clusterbildung.

6. Empirische Pipeline (Repo-Integration)

Neue Pfade - `theory/appendices/appendix_Mbeta_stEEPNESS_cascade.tex` (oder `.md`) - `empirical/sim/stEEPNESS_cascade.py` (*Simulation*) - `tests/test_stEEPNESS_cascade.py` (*Signatur-Checks: Mehrfach-Manifestationen, Θ -Shift, β -Precursor*) - `empirical/tools/beta_theta_estimator.py` (*Fensterbasierte Schätzer für lokale β & Θ*)

README-Ergänzung - Abschnitt "**M- β : Zyklische Emergenz**" mit Abbildung der Kaskade und Hinweis auf Testkriterien.

CITATION.cff - Modulreferenz **UTAC v1.0.1 — M- β** hinzufügen.

7. Export-Schema & Bugfix (β -Mean vs. CI-Breite)

Zur Vermeidung der gemeldeten Verwechslung (*`beta_band_mean` enthielt CI-Breite*):

Schema (YAML/JSON)

```
hypotheses:
  - id: beta_universality
    evidence:
      beta_mean: float          # Mittelwert der  $\beta$ -Schätzungen über Elemente
      beta_ci_width_mean: float # Mittelwert der CI-Breiten (zur Separat-Interpretation)
      beta_values: [float]
      delta_aic_linear: float
      delta_aic_power_law: float
```

Python-Aggregation (Snippet)

```
beta_values = [e.beta for e in elements]
ci_widths = [e.stEEPNESS_band for e in elements] # Breite, nicht Mittelwert!
note["evidence"]["beta_mean"] = float(np.mean(beta_values))
note["evidence"]["beta_ci_width_mean"] = float(np.mean(ci_widths))
```

Tests - Sicherstellen, dass `beta_mean ≈ 4.0±0.4` bei unterstützendem Status. - `beta_ci_width_mean` separat reporten; nie als β -Mittel „misslabeln“.

8. Edge Cases & Stabilitätsregeln

- **Sättigung:** Mehrfache Manifestationen ohne dazwischen ansteigende $\beta \rightarrow$ Setze Refraktärzeit (temporäres $\kappa_{\uparrow} = 0$).
 - **Überkopplung:** Wenn β zu groß (numerische Instabilität), clippen (z. B. $\beta \leq 20$).
 - **Meta-Cluster:** Synchronisierte Sprünge in distinkten Subfeldern \rightarrow globaler Θ -Shift; kennzeichnen und separat auswerten.
-

9. Notation & Konsistenz

- β : Steilheitsparameter (Potenzial).
 - Θ : adaptive Schwelle (Resilienz/Barriere).
 - R : Treiber/Trigger (Stress, Dichte, Kopplung, Inputlast).
 - ψ : Systemantwort/Manifestationsgrad.
 - \mathcal{C} : Menge manifestierter Komponenten (Bedingungen).
-

10. Anschlussfähigkeit & Prüfpfad

- **Replikations-Kriterien:** (i) Nachweis lokaler β -Präursors vor Sprüngen; (ii) messbare Θ -Verschiebung nach Manifestationen; (iii) Clusterbildung.
 - **Domänen-Übersetzung:** Metriken so wählen, dass β, Θ als dimensionslose, vergleichbare Größen geschätzt werden können (z. B. normierte Z-Scores, Informationsentropie, AIC-Vergleiche gg. Nullmodelle).
 - **Publikations-Hook:** "Cascading Steepness Hypothesis" als Appendix mit klaren Testfällen.
-

11. Nächste Schritte (konkret)

1. **Appendix einchecken** (`appendix_Mbeta_stEEPNESS_cascade.*`).
2. **Sim & Estimator** implementieren; Unit-Tests aktivieren.
3. **Aggregations-Bugfix** (`beta_mean` vs `beta_ci_width_mean`) in Exportpfad.
4. **Abbildung** (Kaskadenverlauf) generieren und ins Manuscript übernehmen.
5. **Klimakapitel** mit M- β -Vorhersage-Absatz verknüpfen (Vorläufer- $\beta + \Theta$ -Shift).

— Ende des Moduls —