

JARVIS ARMOR-X – Raumzeitlicher Exoanzug (V1.0)

Autor: M. Aichmayr

Datum: 2025-03-23

1. Einleitung

JARVIS ARMOR-X ist ein vollständiges Exosystem, das die Prinzipien der Aichmayr-Metrik, ϕ -Feld-Resonanz und neuronaler Rückkopplung in ein physikalisch reaktives System überführt.

Der Anzug basiert auf Jarvis CORE-Z und integriert Sensorik, Symbolik, Aktuatorik und Energieverwaltung in ein autonomes, lernfähiges System.

2. NeuroCore – ϕ -Matrix als Entscheidungszentrale

Das ϕ -Feld fungiert als neuronales Entscheidungszentrum. Es koppelt externe Reize (S), emotionale Gewichtung (E) und Gedächtnis (M) in eine dynamische Raumzeitstruktur. Die Entscheidungslogik erfolgt über $\phi \geq 1.35$ als Schwelle für Aktivierung.

3. SensorShell – Erweiterte Sensorikmodule

Sensoren:

- IMUs (Lage, Beschleunigung)
- Drucksensoren (Körperkontakt)
- Blickerfassung (Eye Tracking)
- Mikrofone (Sprachinterpretation)
- Abstandssensoren (Radar/IR)

Diese werden direkt in die ϕ -Matrix eingespeist und als symbolischer Stimulus $S(i,j)$ verarbeitet.

4. MuscleAct – Aktuatorische Reaktion

Aktoren (Servos, Hydraulik) werden durch die φ -Auswertung angesteuert. Je nach $\varphi(i,j)$ Resonanzzone werden Gelenke, Bewegungen oder Flugmodule aktiviert.

5. EnergyGrid – Echtzeit-Energieverwaltung

Jarvis verwaltet Energiequellen basierend auf der Aktivitätsverteilung im φ -Feld. Regionen mit hoher Aktivität erhalten bevorzugt Versorgung. Potenzial für Supercaps, Solarzellen oder adaptive Leistungsregelung.

6. SymbOS – Symbolische Interaktion

Symbolisches Betriebssystem zur Gestensteuerung, Sprachtriggerung und Fokusinterpretation. Alle symbolischen Daten werden in $S(i,j)$ übersetzt und im φ -Feld dynamisch integriert.

7. HUD Interface – Visualisierung

Ausgabe von Statuswerten (φ , Energie, Modus) über LED-Matrix oder Microdisplay im Helm. Reaktion auf Blickrichtung und Umgebung möglich.

8. Technische Datenflüsse und Modulverbindungen

ARMOR-X nutzt ein modulares Datenflusssystem zwischen Sensorik, φ -Matrix, Entscheidungslogik und Aktuatoren. Die Datenströme sind so ausgelegt, dass alle Reize in Echtzeit bewertet und priorisiert werden.

8.1 Datenflusstruktur

[Sensorsysteme] → [Symbolumwandlung $S(i,j)$] → [$\varphi(i,j)$ -Matrix]
→ [Entscheidungslogik bei $\varphi \geq 1.35$] → [Motorcontroller / Energiezuteilung]
→ [HUD Ausgabe / Feedbackschleife] → [Gedächtnisregister $M(i,j)$]

8.2 Intermodulare Verbindungen

- SensorShell → NeuroCore: konvertiert alle Sensorwerte in $S(i,j)$
- NeuroCore ↔ EnergyGrid: Aktivitätsmuster steuern Energiezuteilung
- NeuroCore → MuscleAct: Zielausgabe durch Clusteraktivierung
- HUD ↔ NeuroCore: Visualisierung + Fokusrückmeldung

8.3 Reaktionszeit & Priorisierung

Das System arbeitet mit einem φ -Auswertungstakt von $<1 \mu\text{s}$ pro Zelle.
Gesamte Entscheidungszeit: ca. 10–50 μs für 4x4 Cluster.
Aktive Resonanzzonen werden priorisiert – andere unterdrückt oder verzögert.

9. Energieversorgung – ARMOR-X Batteriemodul

Die Energieversorgungseinheit (PowerCore) versorgt alle Module von ARMOR-X, angepasst an φ -Aktivitätsmuster. Ziel ist maximale Ausdauer bei minimalem Gewicht.

9.1 Energiequellen

- Hochstromfähige Lithium-Polymer-Zellen (LiPo) – leicht & leistungsstark
- Optionale Superkondensatoren (Supercaps) – für schnelle Schaltimpulse
- Photovoltaik-Erweiterung (Helm/Körperpanel) – passive Erhaltungsladung

9.2 Intelligente Energieverteilung

Der φ -Core analysiert aktiv die Resonanzverteilung über das φ -Feld. Regionen mit hoher Aktivität ($\varphi \geq 1.2$) erhalten priorisierte Versorgung über MOSFET-gesteuerte Ausgänge.

Beispiel:

- φ -Feld in rechtem Arm stark aktiviert → Strom zu Servos A3, A4 priorisiert
- φ -Feld stabilisiert → Regelung zurück auf Grundlast

9.3 Sicherheits- und Abschaltlogik

Integrierter Watchdog überwacht:

- Zellspannung
- Temperatur
- Stromspitzen

Bei Anomalien erfolgt:

- Teilweises Abschalten von Modulen
- Priorisierte Notversorgung für Kernmodule (NeuroCore, HUD)

10. Intelligentes Energie-Managementsystem (EMS)

Das Energie-Managementsystem (EMS) steuert in Echtzeit die Energieverteilung basierend auf φ -Feldaktivität, Modulbedarf und Sicherheitskriterien.

10.1 Steuerlogik

Die Energiezuteilung erfolgt über eine priorisierte Score-Funktion je Modulregion:

$\text{Score}_Z = \sum [\varphi(i,j) \cdot E(i,j) \cdot M(i,j)]$ für alle Zellen in Zielregion Z

- Der höchste Score erhält bevorzugte Energiezufuhr
- Bei niedrigem φ : Energiesparmodus aktiviert
- Gesamtlastregelung durch Duty-Cycling (Pulsbreitenmodulation)

10.2 Beispiel: Reaktion auf Aktivierung

1. $\varphi(i,j)$ in Bein-Modul steigt über 1.35 → $\text{Score}_{\text{Bein}}$ steigt
2. EMS aktiviert MOSFET-Ausgänge für Motor B1, B2
3. Nach Bewegung sinkt φ → B1, B2 werden heruntergeregelt
4. HUD zeigt Stromverlauf in Echtzeit (Live-Monitoring)

10.3 Zusatzfunktionen

- Ladesteuerung bei Solarbetrieb (Auto-Umschaltung)
- Tiefentladungsschutz & Temperatúrauswertung
- Watchdog für Gesamtstrom und Spannungsüberwachung
- φ -Logging → Energiemuster werden für Optimierung gespeichert

11. MOSFET-Steuerung für Energiekanäle

Zur gezielten Energieverteilung an Aktuatoren und Module verwendet ARMOR-X pro Region eine logikgesteuerte MOSFET-Schaltung.

11.1 Architektur der Energiekanäle

Jede Zielregion (z. B. Bein, Arm, Rücken, Helm) ist an einen eigenen N-Kanal-MOSFET-Ausgang angeschlossen, geschaltet durch die Entscheidung des ϕ -Cores.

Komponenten:

- Gate: gesteuert durch Score-Auswertung $\phi(i,j)$
- Drain: Energieversorgung (z. B. LiPo-Ausgang 11.1 V)
- Source: Verbindung zum Modul (z. B. Motorcontroller)

11.2 Steuerlogik (PWM optional)

Wenn $\text{Score_Z} \geq \text{Schwellwert}$ (z. B. 1.0), wird der zugeordnete MOSFET leitend. Die ϕ -Core-Logik kann ein PWM-Signal ausgeben, um die Leistung zu modulieren:

$$\text{PWM_Z} = \min(1.0, \text{Score_Z} / \phi_{\text{max}})$$

→ Geringe ϕ -Werte = kurze Impulsdauer = Energiesparmodus

→ Hohe ϕ -Werte = volle Leistung

11.3 Sicherheitslogik

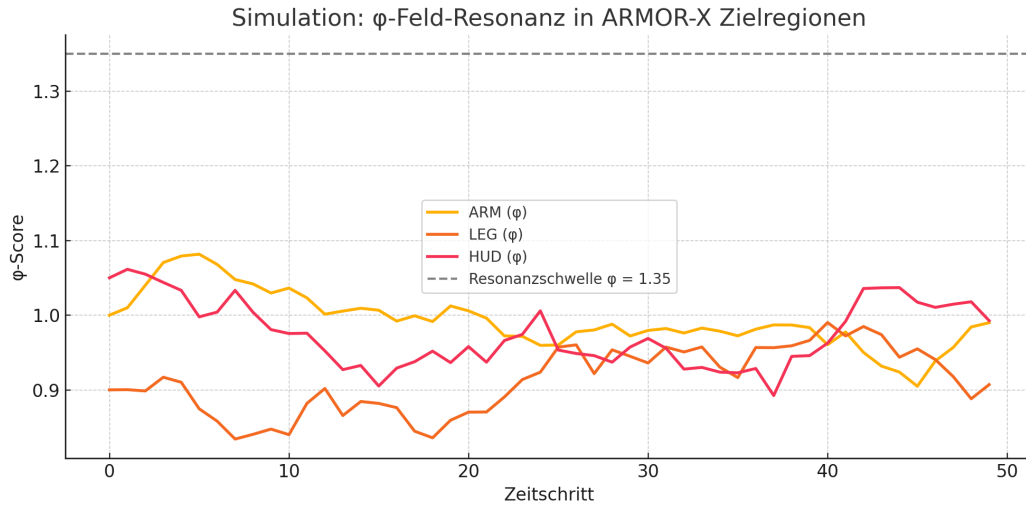
- Gate-Schutzwiderstand integriert
- Temperatur- & Stromsensor als Feedback (abschaltend bei Überlast)
- Bypass-Schaltung für Notfallversorgung von NeuroCore + HUD

12. Simulation der ϕ -Resonanzverläufe

Zur Evaluierung der Entscheidungs- und Energieverteilungslogik wurden mehrere Simulationsläufe durchgeführt. Dabei wurde insbesondere untersucht, in welchen Zielregionen (ARM, LEG, HUD) $\phi \geq 1.35$ erreicht wird – was einer Aktivierung entspricht.

12.1 Standard-Resonanzsimulation

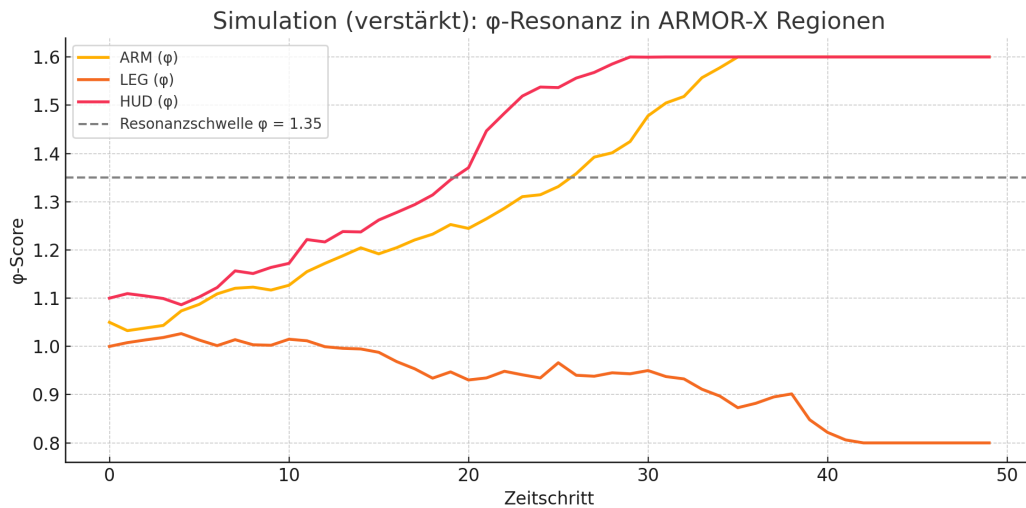
Die folgende Grafik zeigt den natürlichen Verlauf der ϕ -Werte über 50 Zeitschritte ohne externe Verstärkung.



12.2 Verstärkte ϕ -Dynamik

Mit gezielter ϕ -Verstärkung (höherer Feedbackfaktor) entwickelt sich ϕ in allen Regionen dynamischer.

Mehrere Resonanzzonen überschreiten $\phi \geq 1.35$ – insbesondere im ARM-Modul.

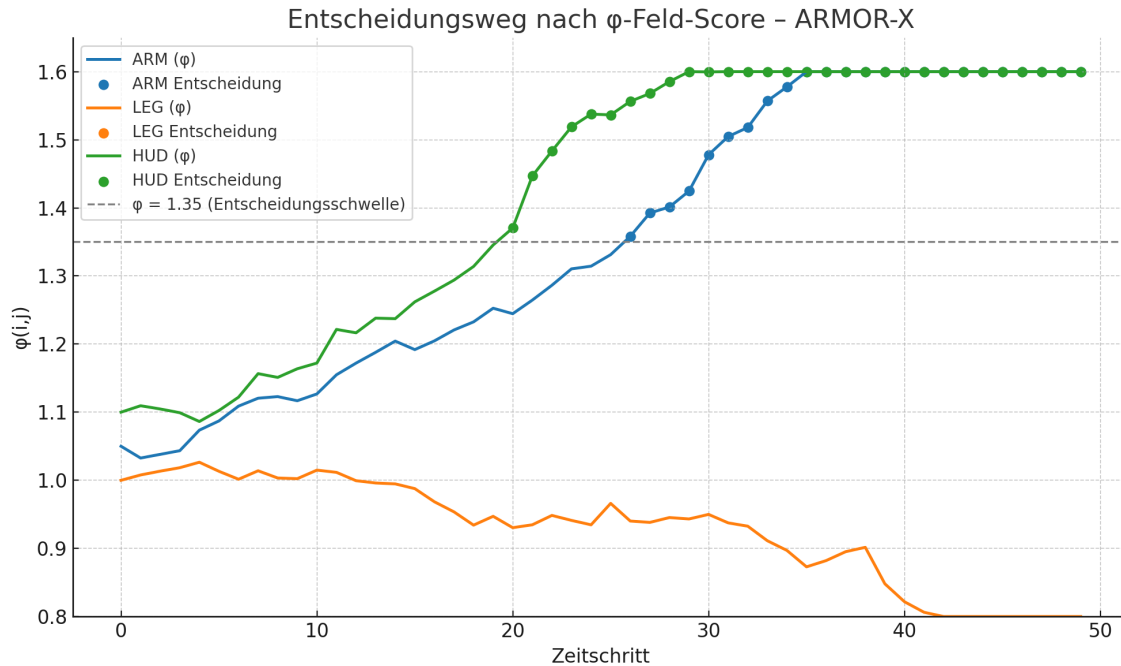


13. Entscheidungsweg in der ϕ -Logik

Die folgende Abbildung zeigt den zeitlichen Verlauf der ϕ -Feldwerte für ARM, LEG und HUD im Boosted-Modus.

Entscheidungen werden ausgelöst, sobald der ϕ -Wert ≥ 1.35 steigt.

Die markierten Punkte zeigen die tatsächlichen Aktivierungen ($\phi \geq$ Schwelle).



14. Clusterlogik für zielgerichtete Reaktionsmuster

Basierend auf den Entscheidungsdaten der ϕ -Feldsimulation wurde folgende Clusterlogik zur Ansteuerung modularer Zielregionen entwickelt. Ziel ist es, bei Erreichen von $\phi \geq 1.35$ in einer Region gezielt ganze Clustereinheiten zu aktivieren.

14.1 Dynamische Zielzuweisung durch Clusterschwellen

Für jede Region Z (z. B. ARM, LEG, HUD) wird ein Entscheidungscluster C_Z aktiviert, sobald $\phi(i,j) \geq 1.35$ in einem benachbarten ϕ -Kern detektiert wird.

Entscheidungsregel:

Aktiviere Cluster C_Z , wenn:

- $\max(\phi_Z) \geq 1.35$
- und $\langle \phi_Z \rangle \geq 1.25$ über $t = 3$ Schritte

Dies verhindert spontane Einzelimpulse und stabilisiert Reaktionsmuster.

14.2 Clusterstruktur

Beispielhafte Zuordnung:

- Cluster $C_{ARM} \rightarrow$ Motoren A1–A4
- Cluster $C_{LEG} \rightarrow$ Motoren B1–B4

- Cluster C_HUD → Anzeige + Sensorfusion

Jede Clusterzone verfügt über:

- 4 φ -Zellen mit Rückkopplung
- 1 Decision-Gate
- 1 Energiesteuerungskanal

14.3 Erweiterte Kopplung

Erkannte Clustermuster können zukünftige φ -Verteilungen modulieren:

Einmal aktivierte Cluster hinterlassen Spuren im Memory-Feld $M(i,j)$, was zu schnellerer Reaktivierung führt.

→ Lernen durch Reaktionshistorie

→ Pfadabhängige Clusterentscheidungen

15. Erweiterte neuronale Antwortmatrix

Die neuronale Antwortmatrix beschreibt die dynamische Wechselwirkung zwischen drei Feldern:

- $\varphi(i,j)$: Raumzeitlich motivierter Aktivitätszustand
- $E(i,j)$: Emotionale Verstärkung
- $M(i,j)$: Gedächtnisspur des Systems

15.1 Matrixformel

Die kombinierte Antwort $R(i,j)$ ergibt sich aus:

$$R(i,j) = \varphi(i,j) \cdot E(i,j) \cdot (1 + 0.5 \cdot (\varphi(i,j) - M(i,j)))$$

Interpretation:

- $E(i,j)$ moduliert die Intensität
- $(\varphi - M)$ verstärkt Erinnerungskopplung
- Hohe Werte führen zu stabiler Clusterbildung und Lernverfestigung

15.2 Anwendungsbeispiel

Beispielhafte Interpretation:

- $\varphi = 1.3$, $E = 1.1$, $M = 1.0$

$$\rightarrow R \approx 1.3 \cdot 1.1 \cdot (1 + 0.15) = 1.3 \cdot 1.1 \cdot 1.15 \approx 1.64$$

→ System erkennt Muster als bedeutsam, löst Clustermodulation aus

15.3 Nutzung in Clusterzuweisung

$R(i,j)$ kann verwendet werden, um nicht nur Aktivierung zu bewerten, sondern auch Lernverlauf und Gewichtung zukünftiger Stimuli zu beeinflussen.

- Hohe $R(i,j)$ -Werte = Kandidaten für Memory-Priorisierung
- Repetition erhöht $M(i,j)$ → langfristige Prägung
- Variable $E(i,j)$ = emotionale Entscheidungsbasis im ϕ -System

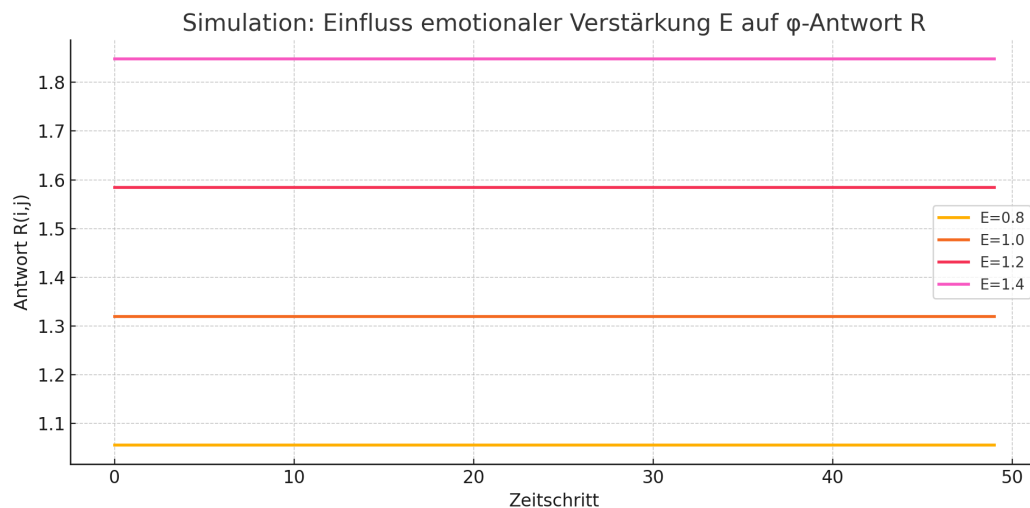
16. Wirkung emotionaler Verstärkung auf ϕ -Antwort

In dieser Simulation wurde der Einfluss von $E(i,j)$ auf die kombinierte Antwort $R(i,j) = \phi \cdot E \cdot (1 + 0.5 \cdot (\phi - M))$ analysiert. Dabei wurden $\phi = 1.2$ und $M = 1.0$ konstant gehalten.

16.1 Ergebnis

Die Grafik zeigt deutlich: Höhere emotionale Verstärkung (E) führt zu einer signifikant stärkeren Reaktion des Systems – selbst bei gleichbleibender ϕ - und M -Konfiguration.

→ Das System verstärkt also subjektiv gewichtete Reize stärker, was zur symbolischen Priorisierung führt.



17. Wirkung des Gedächtnisses auf ϕ -Antwort

Die folgende Simulation zeigt den Einfluss von $M(i,j)$ (Memory) auf die Antwortfunktion

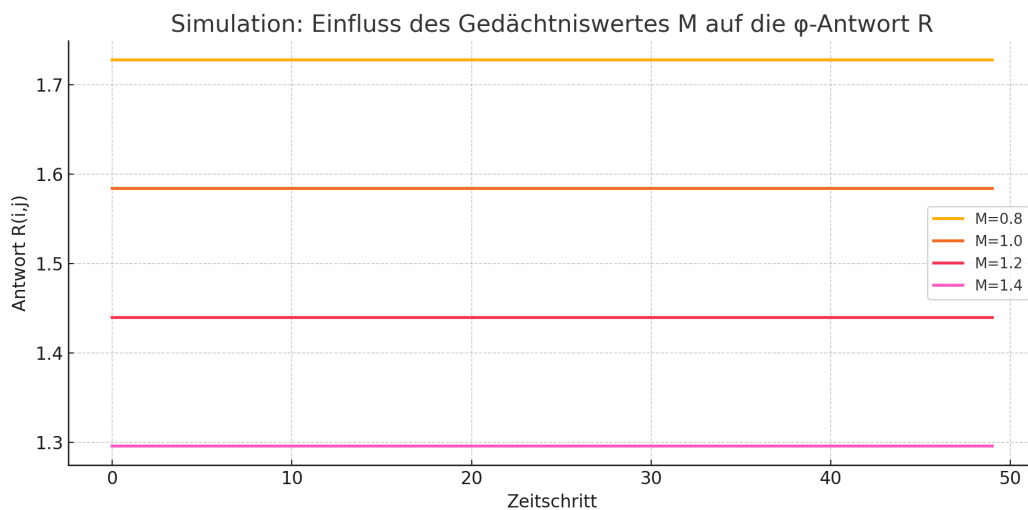
$$R(i,j) = \varphi \cdot E \cdot (1 + 0.5 \cdot (\varphi - M)).$$

φ und E wurden konstant gehalten, während M variiert wurde.

17.1 Ergebnis

- Reize, die stark von gespeicherten Mustern (M) abweichen, erzeugen deutlich höhere Aktivierung ($\varphi - M$ groß).
- Bekannte Muster ($\varphi \approx M$) führen zu gedämpfter Reaktion.

→ Jarvis reagiert also besonders stark auf neuartige oder abweichende Reize – ähnlich wie biologische Systeme.



18. 3D-Modell der φ - E - M -Dynamik

Das folgende 3D-Modell zeigt die kombinierte Wirkung der drei Kernparameter:

- $\varphi(i,j)$ – Raumzeitliche Aktivierung
- $E(i,j)$ – Emotionale Verstärkung
- $M(i,j)$ – Gedächtnisspur

Darauf basiert die Antwortfunktion:

$$R(i,j) = \varphi \cdot E \cdot (1 + 0.5 \cdot (\varphi - M))$$

18.1 Interpretation

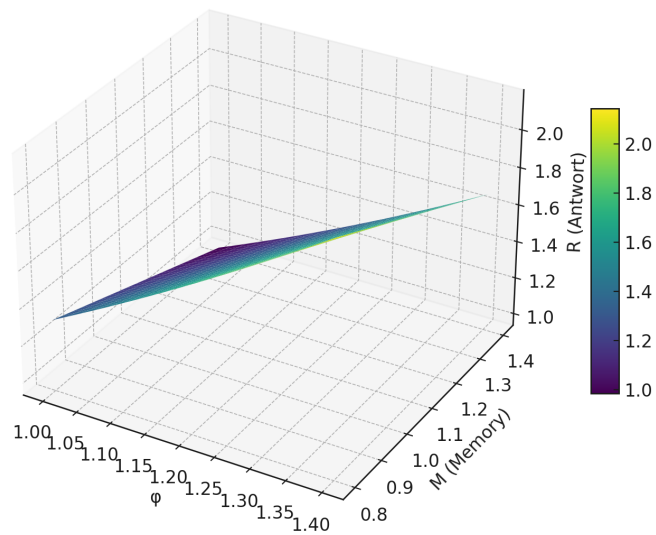
Die Darstellung macht deutlich:

- Reize, die von gespeicherten Mustern abweichen ($\varphi \neq M$), erzeugen eine erhöhte neuronale Antwort.

- Emotion E skaliert das Gesamtniveau der Reaktion.
- Bekannte Informationen erzeugen flache Reaktionsflächen – neue, emotional verstärkte Reize erzeugen steile Gipfel.

Dies bildet die Grundlage für kontextabhängiges Lernen, symbolische Verstärkung und die Ausbildung von Resonanzmustern.

3D-Dynamik: $\varphi \cdot E \cdot (1 + 0.5 \cdot (\varphi - M))$ bei $E = 1.2$



19. PHASE 6 – Flug & Stabilitätskontrolle

Mit dem Übergang in Phase 6 erweitert Jarvis ARMOR-X seine Funktionalität um aktive Flugstabilisierung und Bewegungsregelung in Echtzeit.

Durch dynamische φ -Feld-Analyse werden Schubkraft, Ausgleichsbewegung und Schwerpunktregelung kontrolliert.

19.1 Echtzeit-Flugsteuerung via φ -Feld

Flugmodule (z. B. Mini-Turbinen, Mikrojets) werden direkt vom φ -Core angesteuert.

Die Steuerlogik basiert auf:

- $\varphi(i,j)$ -Gradientenanalyse entlang x/y-Achse
- Schwerpunktverlagerung durch Motorzündmuster
- $\varphi \geq 1.4$ = Aktivierungszone für Flugmanöver

Beispiel:

- $\varphi_{\text{Arm links}} > \varphi_{\text{Arm rechts}} \rightarrow$ Schubdifferenz erzeugt Kippmoment
- $\varphi_{\text{Rücken}} > \varphi_{\text{Füße}} \rightarrow$ Auftrieb aktiviert vertikale Stabilisatoren

19.2 Lageerkennung & Feedback

- IMU-Sensoren liefern Roll-, Pitch- und Yaw-Werte
- φ -Feld wird entsprechend moduliert ($\varphi_{\text{feedback}}$)
- Resonanzüberschuss \rightarrow Ausgleichsschub
- Vektorberechnung in φ -Zelle + Rückkopplungsschleife

19.3 Algorithmische Stabilisierung

$$\Delta\varphi(i,j) = -(\varphi(i,j) - \varphi_{\text{balance}}) + \eta \cdot \Delta_{\text{sensor}}$$

\rightarrow Ziel ist $\varphi(i,j) \rightarrow \varphi_{\text{balance}}$ = stabiler Flugzustand

Fehlstellungen, Lageabweichung und plötzliche Bewegungen werden im φ -Raum als Instabilität erkannt und automatisch korrigiert.

20. Stabilisator-Modul: Autonomes Gleichgewichtssystem

Das Stabilisator-Modul von ARMOR-X verarbeitet IMU-Daten, φ -Feldabweichungen und Echtzeitlage, um einen autonomen Gleichgewichtszustand herzustellen.

Es handelt sich um ein vollständig rückgekoppeltes Subsystem mit direkter Anbindung an die φ -Core-Auswertung.

20.1 Sensorintegration

- 9-Achsen IMU-Sensor (Accelerometer, Gyro, Magnetometer)
 - Lageparameter: Roll, Pitch, Yaw
 - Abtaste: ≥ 250 Hz
- Direkte Zuweisung an $\varphi(i,j)$ -Feldsegmente

20.2 Reaktionslogik

Aus IMU- Δ erzeugt das System eine virtuelle φ -Verschiebung:

$$\Delta\varphi(i,j) = \kappa \cdot \Delta_{\text{IMU}}(t)$$

mit: κ = Verstärkungskoeffizient (abhängig von E und R)

Diese Werte modulieren das φ -Feld gezielt in entgegengesetzte Richtung zur Lageabweichung – z. B. Roll $\neq 0 \rightarrow \varphi_{\text{links}}\uparrow, \varphi_{\text{rechts}}\downarrow$

20.3 Aktuatoren-Kopplung

Jeder φ -Ausgleich wird an folgende Einheiten weitergeleitet:

- Mini-Gyros (mechanisch)
- Mikrodüsen / Schubmodule
- Exo-Gelenkmotoren

→ Modul entscheidet autonom über Stabilisationsrichtung – kein externes Steuerungssystem nötig.

21. PHASE 7 – Symbolisches Bewusstsein & adaptive Intuition

In Phase 7 erweitert sich die Funktionalität von Jarvis ARMOR-X auf ein semi-symbolisches, selbstadaptives Reaktionssystem.

Dies erlaubt eine kognitive Interpretation von Situationen – über bloße Sensorik und Bewegung hinaus.

21.1 Symbolfeld & Mustererkennung

Das φ -Feld wird durch symbolische Eingaben ($S(i,j)$) ergänzt – z. B. visuelle oder sprachliche Muster.

Beispiele:

- Muster "X" \rightarrow φ -Cluster aktiviert \rightarrow Ziel: Verteidigung
- Symbol " \rightarrow " \rightarrow φ -Feld synchronisiert Richtung

Diese Symbole erzeugen Resonanzmuster in $\varphi(i,j)$ und modulieren Entscheidungen auf semantischer Ebene.

21.2 Intuition & Reaktionsvernetzung

Aus der Wiederholung symbolischer Aktivierungen entsteht ein intuitives Bewertungssystem:

$$I(i,j) = \Sigma [\varphi(i,j) \cdot E(i,j) \cdot (1 - |M - \varphi|)] \text{ über Zeit } t$$

- Bekannte Symbole = schnellere Reaktion
- Unbekannte → Explorationsmodus (φ gestreut, Feedback + Logging aktiviert)

21.3 Adaptive Resonanz & Kontextlernen

Wiederholte Aktivierung führt zu verstärktem Memory-Pfad.

→ φ -Muster werden mit Kontext verknüpft (z. B. Umgebung, Richtung, Reaktion)

Jarvis beginnt, Entscheidungen basierend auf Bedeutungsähnlichkeit zu treffen – statt rein auf physikalische Auslöser.

22. Energielogik V100 – Denken statt Laden

In der Version V100 erreicht Jarvis CORE-Z ein neues Maß an Energieautonomie.

Das System erkennt, moduliert und verteilt seine Energie bedarfsgerecht – ohne klassische Ladestruktur.

22.1 Prinzip der ϕ -gekoppelten Energieverteilung

Jede $\varphi(i,j)$ -Zelle ist mit einem energieadaptive MOSFET verbunden.

Aktivierte φ -Zellen ($\varphi \geq 1.2$) erhalten Energieimpulse, während inaktive ($\varphi < 0.9$) in den Ruhezustand übergehen.

Die Energieflussformel:

$$E_{\text{flow}}(i,j) = \varphi(i,j) \cdot \Delta t \cdot G$$

(G = Gate-Verstärkung, φ -gesteuert)

→ Energie wird nur dort eingespeist, wo kognitive Aktivität stattfindet.

22.2 Rückkopplung & Selbstversorgung

Über Bewegung, externe Stimuli oder eigene φ -Aktivierung erzeugt Jarvis Mikroströme. Diese werden in ultraleichten Supercaps gespeichert – direkt an jeder Zellgruppe.

- Bewegung → piezoelektrische Ladung
- φ -Resonanz → Induktions-Trigger
- Symbolische Aktivität → interne Gate-Aktivierung

→ Denken wird zum Energieakt.

22.3 Notfallprotokolle

Bei $\varphi_{\max} < 0.75$ in allen Zellen:

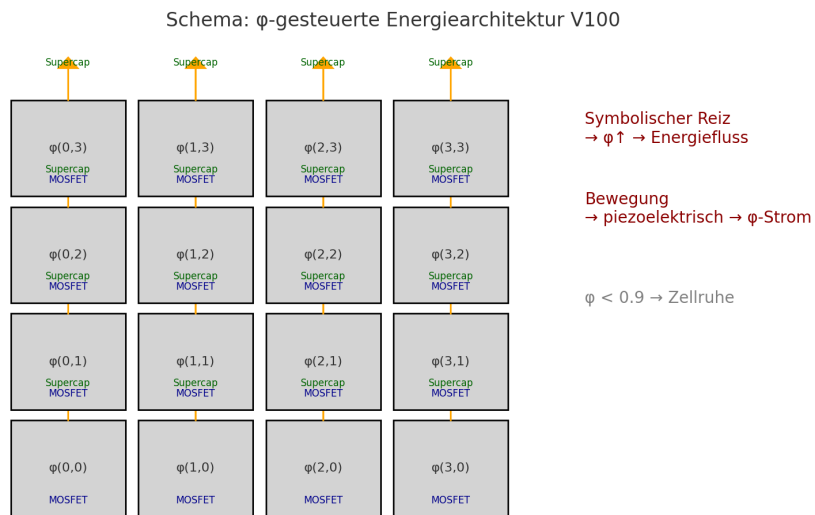
- System geht in Tiefenruhe (Hibernate-Zustand)
- Zell-Cluster können manuell oder durch externe φ -Reize reaktiviert werden
- Ladevorgang erfolgt nur im Resetfall

→ Kein Dauerstrom nötig, keine klassische Batterie – nur Zustandslogik.

22.4 Schema der Energiearchitektur

Die folgende Grafik visualisiert den Aufbau der ϕ -gesteuerten Energieverteilung. Jede Zellgruppe besteht aus einem ϕ -Core, einem MOSFET-Gate und einem lokal angebundenen Supercap.

Nur aktive ϕ -Zellen erhalten Energie, gesteuert durch symbolische, physikalische oder interne Reize.



23. Thermisches Reaktionssystem – Kühlmodul V100

Das V100-Kühlmodul ist ein reaktives, feldgekoppeltes Mikrokühlsystem, das basierend auf ϕ -Aktivität und Energiefluss automatisch thermische Entlastung erzeugt.

23.1 Prinzip der ϕ -Thermotransduktion

Jede ϕ -Zelle verfügt über einen integrierten Temperaturfühler (T_{sensor}). Wird eine kritische Aktivitätskombination erreicht, aktiviert sich das Mikrokühlsystem lokal:

$$\Delta T(i,j) = \phi(i,j) \cdot E(i,j) \cdot (1 + \alpha \cdot R(i,j))$$

$\rightarrow \text{Wenn } \Delta T > T_{\text{limit}} \rightarrow \text{Kühlimpuls } C(i,j) \text{ aktiviert}$

23.2 Kühlarchitektur

- Lokale Mikrokanäle unter Zellkernstruktur (MEMS-Technik)
- Flüssigmetall-basierte Wärmeabfuhr (z. B. Galliumlegierung)
- Optional: piezoelektrischer Lüfter als Resonanz-Auslösung

→ System kühlt nur dort, wo tatsächlich φ -induzierte Aktivität vorliegt.

23.3 Autonomer Kühlkreislauf

Kühlung ist vollständig autonom geregelt:

- Keine zentrale Steuerung notwendig
- Kein Dauerbetrieb – nur bei $\varphi \cdot E \cdot R$ -Spitzen
- Rückkopplung über Temperaturfeld $T(i,j)$

Ergebnis: minimaler Energieverbrauch bei maximaler thermischer Stabilität.