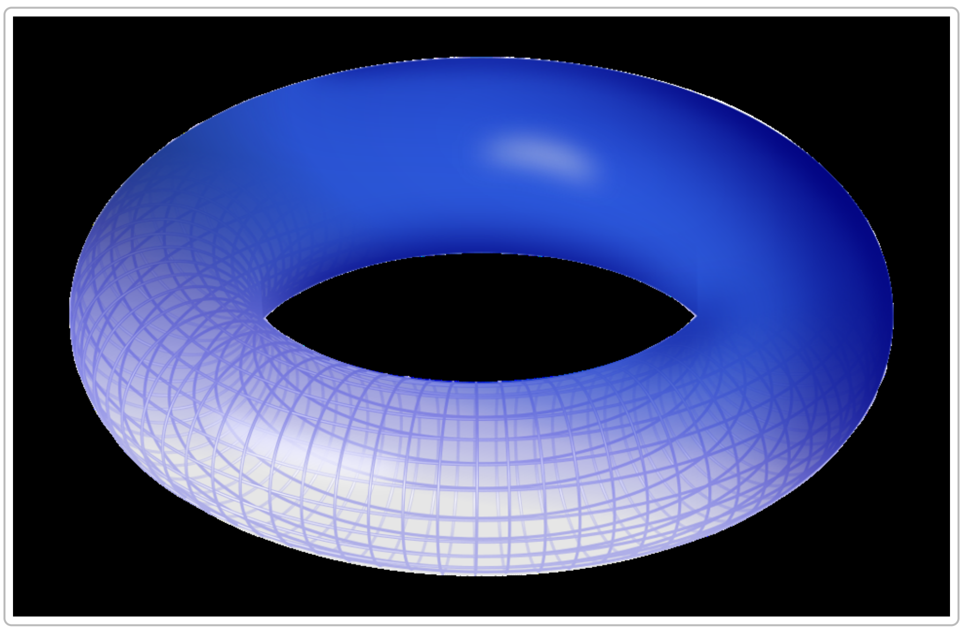
ChatGPT

Symplectic and Category-Theoretic Brain-State Mapping for DonutOS

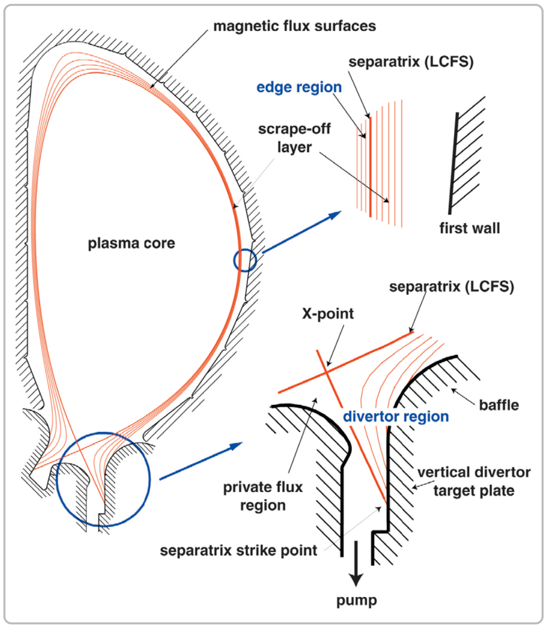
We model brain and UI states as linked categories, using *functors* to map neural activity to control actions. For example, Integrated Information Theory (IIT) frames consciousness as a functor from a neural activity category *N* to an informational/category *I* [1]. Analogously, we define a functor from a *brain-state* category (objects = EEG/oscillation patterns, arrows = transitions) to a *UI-state* category (objects = overlay configurations or control modes, arrows = UI transitions). This functor preserves composition: combining neural inputs yields a corresponding composite UI action. We also use **gluing functors** to merge modalities (e.g. BCI + gaze) by linking overlapping categories, and **translation functors** to shift contexts (e.g. switching goal or scene). In practice, we implement this by docking/fusing control "membranes" (panels) when their underlying neural categories overlap, ensuring consistent behavior. In UI terms, composing two membranes is like composing arrows in a category. The result is a modular, compositional mapping: each brain-signal type (oscillation band, phase state, etc.) is its own category, and our code provides functors that map these into the toroidal UI state, respecting their structure [1].

Symplectic Geometry on the Toroidal Manifold



We embed neural oscillatory states onto a 2D torus (a phase-space manifold) and apply **symplectic maps** (canonical transforms) to manipulate them. By definition, a symplectomorphism is a diffeomorphism of a symplectic manifold (like our torus) that preserves the symplectic form (phase-space "area") [2]. Concretely, if ω is the torus's area form, then a map f satisfies $f^\omega=\omega$. *This means our UI rotations or shear transforms (e.g. twisting the torus, tilting overlay grids) are volume-preserving: they maintain invariant integrals (analogous to conserved quantities in Hamiltonian flows). In practice, we implement rotations* *or canonical transformations on the torus so that angles remain orthogonal and areas are preserved, ensuring that oscillation-mode interactions stay consistent. Equations like $f*^\omega=\omega$ guarantee that sequential mappings commute properly. The UI implication is that multi-dimensional oscillator data can be combined and unfolded on the torus without distortion – for example, composite rotations of the torus correspond exactly to composing the underlying neural phase shifts. (In code, each rotation in ${x,p}$ coordinates is implemented as a symplectic matrix or Three.js transform that preserves the torus's intrinsic metric.) This preserves the integrity of "phase volume" when mapping between brain and display, as in classical Hamiltonian dynamics [2].

Plasma/Flux-Surface Analogies



We borrow from magnetohydrodynamics (MHD) by treating neural fields like plasmas confined on nested toroidal "flux surfaces" [3]. In a tokamak, magnetic field lines lie on nested closed tori (flux surfaces) and key quantities (e.g. magnetic flux) are conserved on each surface [3]. We use this as a metaphor: imagine brainwave activity flows on concentric toroidal shells. Critical brain regions or synchronized subnetworks then correspond to different flux layers. When these layers reconnect or break (magnetic reconnection), we interpret it as dynamic network reconfiguration – e.g. the UI can reroute an overlay's connections or spawn a new panel when neural patterns shift. Indeed, electrophysiological modeling suggests that the brain's ion-rich extracellular fluid can exhibit magnetofluid dynamics, generating "MHD-like" forces and reconnection phenomena that quickly spread activity [4]. Thus, we visualize data flows as magnetic flux lines on the torus: zones of high coherence act like strong field regions, and when the "plasma" (neural activity) reorganizes, it can jump between layers. Practically, we overlay simulated flux lines or vector fields on the torus (e.g. using Three.js tubes on surfaces) to represent information pathways. UI actions like dragging a "solar gate" panel correspond to shifting or reconnecting these flux surfaces metaphorically. These flux-surface overlays and the concept of separatrices/X-points guide how we update panels: for example, crossing a separatrix (phase boundary) might toggle a membrane's state (edge region vs. core region) in the UI. In summary, the toroidal flux metaphor provides an intuitive, structure-preserving template for visualizing and controlling complex brain-state topologies [3] [4].

Phase Alignment and Boundary Gating

Phase coherence acts as an active constraint on our mappings. Neuroscience shows that inter-regional phase synchronization enables long-range communication, whereas desynchronization weakens it [5]. We exploit this: when two oscillations are **phase-locked**, we allow a mapping or transform (holonomy transfer) to proceed, possibly with full strength; when **out-of-phase**, we attenuate or block the mapping. For instance, if a user's frontal and parietal rhythms are strongly phase-aligned (high phase-lock value), we interpret that as readiness to apply a sensitive holonomy-based control (like a fine rotation or tool activation). If the phase lock is low, we hold the UI steady (embedding a "boundary" or dead-zone) until coherence returns [5]. This implements a weak-control gating: EEG phase-lock metrics (computed continuously) drive a "lock-meter" in the UI; high lock "unlocks" membrane actions, low lock "locks" them. Conceptually, phase acts like a bulk-vs-boundary indicator: coherent brain regions become the "bulk" (open to transformation), while incoherent ones form a "boundary layer" that constrains updates. We display this to the user via explicit indicators (e.g. color-coded halos or a boundary status strip) so the user knows when controls are active. This also serves as a safety clamp: if data is ambiguous (low phase synchrony), mappings are disabled to avoid erratic UI changes. (This idea echoes the "communication through coherence" principle [5] in a control context.)

Implementation Steps and UI Patterns

1. **Define Categories & Functors:** Identify the relevant neural categories (oscillator phases, amplitude envelopes, etc.) and UI target categories (e.g. menus, overlays). Implement functors mapping signals to controls, ensuring composition works: e.g. composing theta+gamma patterns maps to composed UI state. Use the category-theoretic insight that merging categories is like "gluing" data pipelines, so panels that share functions are fused smoothly.

2. **Embed on a Toroidal Phase Space:** Convert multi-band brain signals to points or regions on a torus via phase-to-angle embedding. Use symplectic parameterization (action-angle coordinates) so that rotational updates preserve structure. For example, map phase offsets to rotations around one toroidal cycle, and slower-phase trends to the other. In code, this means representing each oscillation as an angle on S^1 and combining them into a T^2 state, then applying structure-preserving (area-preserving) transforms.

3. **Apply Symplectic (Canonical) Transforms:** Implement UI controls (like rotating the torus or sliding an overlay) as symplectomorphisms. Because $f^*\omega=\omega$, repeated operations remain commutative and reversible. For example, dragging a torus rotation handle corresponds to a Hamiltonian flow on phase space, so that undoing the drag naturally returns the system. This also ensures that if two controls act simultaneously, their effects compose predictably.

4. **Model Information Flow as Flux Surfaces:** Overlay nested contours or vector fields on the torus representing "flux surfaces" of activity (using [58†embed_image] as inspiration). Interpret strongly-connected neural groups as plasmatic flows along these surfaces. Use these flows to drive UI layout: e.g. if activity concentrates on an inner flux surface, display an inner panel; if reconnection occurs (activity jumps surfaces), spawn or merge panels accordingly. This can be done by sampling the EEG topography and drawing tubes on the torus mesh, with reconnection events (sudden changes in cross-surface connectivity) triggering animated transitions between UI panels.

5. **Gate by Phase Coherence:** Continuously compute phase-locking values. Use them to *weight* control mappings: e.g. apply a rotation only if coherence > threshold. Implement UI elements like a “bindu window” that only activates certain tools when phase-lock conditions are met. For example, only apply a “holonomy transfer” (parallel transport of UI context around the torus) when a user’s EEG exhibits stable phase alignment [5]. Show this in the interface as enabled/disabled icons or a phase-lock meter, so users see the boundary conditions in real time.

6. **UI Overlays and Feedback:** Ground the above in concrete patterns: for instance, when phase coherence is high, highlight the torus surface or UI membrane associated with that signal (e.g. lighting up a “coherence knob” or permitting focus). When signals from different categories must be “glued” (e.g. merging BCI intent with gaze intent), animate the docking of panels to reflect the categorical composition. Always visualize the constraints: display boundary strips or glow effects when a mapping is gated. For group sync, use a shared lock meter that accumulates phase alignment across users, unlocking collaborative controls only when group coherence is strong.

In summary, we treat brain oscillations and their phase relations as variables in a symplectic, category-theoretic mapping system. The implementation follows a pipeline: embed signals on a torus, apply structure-preserving transforms, use plasma/flux metaphors for flow, and gate actions by phase coherence. Each step is annotated with underlying theory: e.g. structure-preserving (symplectic) ensures consistency [2], and phase-synchrony gating ensures effective communication [5]. This yields a research-grounded yet practical guide for DonutOS overlays, blending abstract mappings with clear UI hooks.

References: Category-theoretic composition [1]; symplectic (phase-space) maps [2]; nested toroidal flux surfaces [3]; MHD plasma-like brain energy with reconnection [4]; phase synchronization for communication [5].

---

[1] Mathematics and the Brain: A Category Theoretical Approach to Go Beyond the Neural Correlates of Consciousness
https://www.mdpi.com/1099-4300/21/12/1234

[2] Symplectomorphism - Wikipedia
https://en.wikipedia.org/wiki/Symplectomorphism

[3] suli.pppl.gov
https://suli.pppl.gov/2023/course/Moser_SULI_tokamak_2023.pdf

[4] Light and Brain: A Clinical Case Depicting the Effects of Light on Brainwaves and Possible Presence of Plasma-Like Brain Energy.[v1] | Preprints.org
https://www.preprints.org/manuscript/202401.1288

[5] (PDF) Temporal concentration and phase synchronization in phase-amplitude coupling
https://www.researchgate.net/publication/396354433_Temporal_concentration_and_phase_synchronization_in_phase-amplitude_coupling