

Extended Geometric–Informational Multi-Agent System with Hybrid Frames and Complex Dynamics

Jarra Brezina

April 2025

1 Mathematical Formulation

We model the system as a hybrid dynamical system on a state space $X = M \times S$, where M is a (possibly time-varying) manifold representing the geometric frame, and S is a discrete “mode” set indexing frame configurations. Each mode $s \in S$ has its own manifold M_s (with its own topology, metric, and field variables). Agents evolve continuously on M_s via smooth dynamics $\dot{x} = f_s(x, u, t)$ until an event triggers a jump to a new mode $s' \neq s$. At that instant, the underlying manifold switches to $M_{s'}$ and a reset map $x^+ = \Phi_{s \rightarrow s'}(x^-)$ reassigns agent states to the new geometry. For example, if a hole is created or filled, Φ might project each point to its nearest counterpart on the new manifold. This hybrid structure is analogous to piecewise-smooth or switching flows on manifolds. Metrics and fields are mode-dependent. In mode s , define a Riemannian metric g_s and any relevant fields (potentials, information functions) on M_s . A jump may switch to $(M_{s'}, g_{s'}, \phi_{s'})$, altering distances and field values. Agents must then update their state according to $\Phi_{s \rightarrow s'}$. For instance, a cost function $V_s(x)$ on M_s might be reinitialized to $V_{s'}(\Phi(x))$ on $M_{s'}$.

We also permit complex-valued states. Formally, each agent state x_i may lie in C^n (or on a complex manifold), and dynamics are given by complex-valued differential equations. Equivalently, split $x = a + ib$ into two real layers (a, b) with coupled dynamics. Utility functions, coupling matrices, or fields may be complex-valued. This generalizes real flows to complex flows (using Wirtinger calculus for differentiation). For example, an agent’s evolution could follow $\dot{z} = F(z, u)$ with $z \in C$, capturing both amplitude and phase. Complex manifolds (e.g. Riemann surfaces) may serve as configuration spaces for complex states.

2 Agents and Complex States

Complex variables can encode dual-layer information. An agent might interpret $z = x + iy$ as two linked real quantities (e.g. value x and phase y). In applications

this might mean amplitude/phase of a signal, or a primary value and a secondary “polarity.” Agents process such states via complex algebra: they split real/imag parts or operate on the complex number directly. For instance, an agent update rule could use complex backpropagation (as in CVNNs) to adjust z_i . Complex utilities or couplings can then naturally handle phenomena like interference or oscillatory synchronization.

Recent work on complex-valued MAS shows these ideas in practice. For example, directed complex-valued multi-agent systems (DCVMASs) have been studied in containment/consensus problems with time delays and attacks. There, each agent’s state is complex and control laws are designed so that all agents converge (in the complex plane) to a convex hull of leaders. Agents effectively treat real and imaginary parts as intertwined dynamics, using observer states and stability analysis for complex switched systems. Similarly, complex-valued neural networks have been successful in “wave-typed” and frequency-domain tasks. In our context, each agent could run a small CVNN to process inputs on $(\Re z, \Im z)$, ensuring that phase information is properly learned and propagated. Thus, agents may interpret complex states in various ways: • As phase-coded variables, where $\arg(z)$ carries a phase and $|z|$ carries an amplitude or reliability. • As two-layer states, with $\Re(z)$ and $\Im(z)$ acting like parallel real-valued states with coupling. • As quantum-like amplitudes, where z might even represent a probability amplitude in a simplified quantum model of decision-making. In all cases, agents apply dynamics respecting complex differentiability (e.g. using Wirtinger calculus) so that the flow on C^n or on a complex manifold is well-defined.

3 Modeling Discontinuous Transitions

Sudden changes in the frame are modeled as hybrid jumps. At certain discrete events (time triggers or threshold crossings), the mode switches from s to s' , instantly changing the manifold $M_s \rightarrow M_{s'}$. This can represent topological changes (creating a hole, splitting or merging regions) or abrupt geometric changes (metric discontinuity, obstacle appearance). Mathematically, we treat such events as instantaneous transitions with a reset mapping $\Phi_{s \rightarrow s'} : M_s \rightarrow M_{s'}$. For example, if a hole is removed, Φ may smoothly map points from the boundary of the hole onto a new interior region; if a region disconnects, Φ may assign agents to the nearest remaining component. These jumps impact agents’ value functions and links. A value function $V_s(x)$ defined on M_s may become discontinuous under a jump. One approach is to “carry over” the function: define $V_{s'}(\Phi(x)) = V_s(x)$ to maintain continuity. In optimal control terms, we impose boundary conditions on value functions at events. Agents then recompute gradients or policies using the new $V_{s'}$. Informational links (agent communication or network weights) may also rewire: if connectivity depended on geodesic distance in M_s , a new $M_{s'}$ implies recalculated distances and hence updated adjacency. In effect, the network graph is redefined at the jump, akin to switching topologies in consensus algorithms. Formally, this is handled by

switched system theory on manifolds. The continuous evolution on each mode s is governed by differential equations on M_s , while the jump map handles discrete reconfiguration. These are similar to “reset maps” in hybrid automata or “jump conditions” in impulsive systems. If the manifold changes topology (e.g. genus changes), one may use algebraic topology to update global invariants. If only the Riemannian metric changes, the jump is a rescaling $g_s \rightarrow g_{s'}$ so that distances abruptly shift.