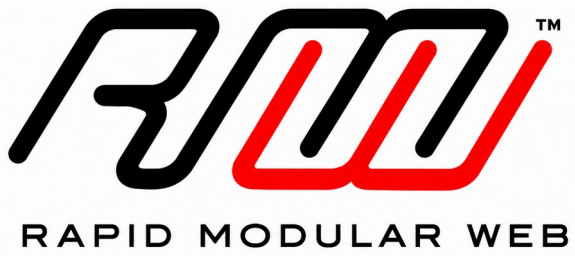

Recursive Agency Cognition for Autonomous Robotics

A Mathematical Supervisory Architecture, Reference Implementation Pattern, and Experimental Protocol

Reuben B. Murphy

Founder, Rapid Modular Web

Engineering Preprint Draft v0.1 | 30 August 2026

PUBLIC DISCLOSURE BOUNDARY

This preprint publishes the generalized mathematical and experimental architecture of agency cognition. It intentionally abstracts proprietary FOX implementation details, including internal orchestration heuristics, provider routing, prompt construction, security composition, mission-correlation logic, and other trade-secret implementation mechanisms.

HISTORICAL PRIVACY NOTE

The early object-networking work discussed in this paper is described only at the level necessary to establish technical lineage. Historical company, product, location, and archival identifiers are intentionally omitted.

Abstract

This paper presents a practical supervisory architecture for recursive agency cognition in autonomous systems. The work originates in a systems-philosophy treatment of perception, value, will, action, resistance, and correction, later formalized as a state-transition model of agency. A separate engineering lineage developed by the author in the early 2000s explored real-time object-oriented networking, synchronized distributed object state, and modular network-centric computing. Modern AI-agency development brought these lines together: the networking platform was re-engineered as the FOX Platform, while the philosophical agency loop was converted into the FOX Agency Cognition Model (FACM), a provider-neutral cognition sub-loop operating inside a governed mission cycle. The proposed architecture does not replace SLAM, reinforcement learning, model predictive control, behavior trees, world models, or low-level robot control. Instead, it supervises them by explicitly representing the agent's mission-relative self/world state, authorized or feasible agency, predicted consequences, actual outcomes, goal discrepancy, prediction discrepancy, resistance classification, and evidence-backed revision. The paper defines the mathematical model, maps it to modern software interfaces, proposes a minimal implementation pattern, and specifies reproducible robotics experiments and ablation studies. The engineering hypothesis is deliberately narrow: an autonomous system that explicitly models its own causal agency and recursively learns from the difference between intended, predicted, and observed outcomes should recover more effectively from novelty than an otherwise comparable system that replans without that explicit agency-cognition state.

Keywords: autonomous robotics; cognitive architecture; agency; self-modeling; world models; model predictive control; adaptive autonomy; metacognition; AI agents; FACM.

1. Engineering Origin: From Systems Theory to Agency Cognition

This work did not begin as a robotics algorithm. It began as a systems question: what makes an action more than motion, and what makes an agent more than a mechanism that happens to change its environment? The author's philosophical manuscript, later completed as Insight: A Treatise on Perception, Will, Resistance, and the Reality of the Other, developed an answer in relational terms. Perception builds an intelligible field; reason models possible states; value makes those states unequal; will adopts an end; agency organizes means; reality answers; and the difference between intention and result becomes information. The philosophical argument is intentionally broader than engineering, but its structure is recognizable as a closed-loop system. In the completed treatise, the system is explicitly recursive: every action changes the world, the agent's model, or both.

The key transition was to stop treating failure as merely a negative outcome. If an agent predicts that action u will produce state x-hat, but reality produces x instead, the discrepancy contains information about the agent's model, capability, environment, or constraints. Resistance is therefore not simply opposition. Properly observed, it is a measurement channel. The same reasoning applies to success: an action may achieve its goal for reasons the agent did not predict, and that unexpected success should still revise the causal model.

Difference → Meaning → Will → Agency → Reality → Insight

Conceptual compression

The philosophical sequence is not a one-way pipeline: Insight recursively modifies the next perception and model.

1.1 A parallel engineering lineage: object networking

Years after the early philosophical work, the same systems instinct appeared in software. The author developed an object-networking architecture in the early 2000s around a simple proposition: networked computing becomes more coherent when applications exchange and synchronize structured object state rather than treating every interaction as an isolated document or screen transfer. Archived engineering records from that period describe a network-centric model built from reusable objects, n-way synchronization, modular transport, explicit network identifiers, and a repeated read/check/send/process loop. Those records are historical evidence of an engineering preoccupation with state continuity, distributed representation, and feedback long before the present wave of agentic AI.

The historical platform is not presented here as an early form of FACM. That would be an anachronism. Its relevance is architectural: it treated a network as a living field of synchronized state and reusable objects, and it separated application

logic from transport. Those ideas later proved unusually compatible with modern AI agency, where an intelligent system must maintain continuity across applications, devices, tools, remote endpoints, and changing computational providers.

1.2 Why modern AI agency resurrected the platform

Large language models and modern AI tooling changed the value proposition of the old networking problem. A distributed object platform once needed human-authored logic to decide what networked state meant and what should happen next. Contemporary AI can perform interpretation, planning, prioritization, and explanation across heterogeneous application contexts. At the same time, raw model intelligence does not solve authority, continuity, evidence, or execution governance. An AI system can generate a plausible plan while remaining wrong about what it is permitted to do, what tools it actually possesses, whether an operation occurred, or why an outcome differed from its expectation.

That tension resurrected the platform as FOX: a modern object-networking and application architecture in which AI is not the platform itself, but a cognitive participant inside governed state and execution boundaries. The FOX implementation provided a practical test bed for translating the agency theory from philosophical prose into software contracts. The result is FACM - the FOX Agency Cognition Model - implemented as a governing cognition sub-loop inside FOX Brain's mission cycle. This paper extracts the generalizable engineering model while leaving the proprietary FOX implementation details out of scope.

2. Engineering Problem and Contribution

Robotic autonomy already contains many of the ingredients required for intelligent adaptation: state estimation, world modeling, trajectory prediction, planning, control, reinforcement learning, fault detection, behavior trees, and increasingly language-model-based task reasoning. The proposed contribution is not another replacement for these techniques. It is a supervisory representation that binds them around the question of agency: what did the system believe, what was it trying to cause, what means did it select, what did it predict, what actually happened, what does the discrepancy imply, and what should be revised before the next act?

The distinction matters because ordinary replanning can hide causal ignorance. A robot may repeatedly route around an obstacle without learning that its traversability model is systematically wrong. A manipulator may eventually succeed at a grasp without recognizing that success depended on unmodeled compliance. A mobile platform may interpret repeated path interference as random dynamics when another agent is adapting to its behavior. Conversely, it may anthropomorphize ordinary failure as opposition. FACM makes these distinctions explicit and testable.

Primary engineering claim

Explicit agency cognition is a supervisory state-estimation and learning problem. The architecture should improve recovery from novelty when it separately tracks goal error, prediction error, causal capability, resistance, and evidence-backed model revision.

The proposed contributions are:

- a compact mathematical state-transition model for agency cognition;
- a dual-discrepancy formulation separating mission/goal error from causal prediction error;
- an explicit self/world/authority/agency state that can be implemented with or without generative AI;
- a conservative resistance classifier that does not infer counter-agency from failure alone;
- a software architecture that can wrap existing robotics stacks rather than replace them;
- a bounded cognition-outcome interface suitable for human operators; and
- a reproducible experimental and ablation protocol for mobile robots, manipulators, or simulation.

3. Mathematical Formulation of the Agency Cognition Loop

Let the autonomous system maintain an internal agency state at discrete cycle t. The state is deliberately broader than geometric pose. It may include estimated external state, verified self-capabilities, mission state, constraints, authority or policy limits, memory, uncertainty, and active dependencies.

$$\hat{S}_t = M(E_t, m_t, I_t, A_t, H_t)$$

(1)

Internal agency state from evidence E, memory m, identity/self-model I, available agency A, and mission/value context H.

For a robot, the internal state may be implemented as a structured vector or graph rather than a single numeric tensor:

$$\hat{S}_t = [\hat{x}_t, \hat{r}_t, \hat{a}_t, \hat{c}_t, \hat{h}_t]$$

(2)

World estimate, robot self-state, available capabilities, constraints/authority, and mission-history state.

3.1 Goal binding

The desired state may be imposed by a mission planner, human operator, supervisory policy, or a higher-level autonomous planner. FACM does not require the robot to originate its own values. The implementation only requires an explicit goal representation against which progress can be evaluated.

$$g_t = G(\text{mission}_t, \text{constraints}_t, \hat{S}_t)$$

(3)

3.2 Agency projection and action selection

The architecture distinguishes the abstract means selected by agency cognition from the low-level controller that realizes them. The action may therefore be a behavior-tree node, navigation command, manipulator skill, MPC reference, task primitive, ROS 2 action, or another bounded capability.

$$u_t = \Gamma(g_t, \hat{S}_t, C_t, A_t)$$

(4)

Select a feasible/authorized means under constraints C and available capability set A.

3.3 Prediction before execution

Before execution, the agent records what it expects the selected means to cause. This is the key step that turns post-hoc explanation into a falsifiable internal model.

$$\hat{x}_{t+1} = F(\hat{S}_t, u_t)$$

(5)

Predicted next state or predicted outcome distribution.

3.4 Reality and observation

Execution occurs through the robot's existing controller and physical plant. The realized state depends on the actual environment, unmodeled dynamics, noise, other agents, actuator limitations, policy enforcement, and physical constraints.

$$x_{t+1} = \Phi(x_t, u_t, C_t)$$

(6)

The realized state is determined by action interacting with a reality that is not exhausted by the internal model.

3.5 Dual discrepancy

FACM maintains two errors because goal completion and model accuracy are not the same thing.

$$e_t^g = d_g(g_t, x_{t+1})$$

(7)

Goal discrepancy: how far the observed outcome is from the desired state.

$$e_t^p = d_p(\hat{x}_{t+1}, x_{t+1})$$

(8)

Prediction discrepancy: how far reality departed from the agent's predicted consequence.

Goal error	Prediction error	Engineering interpretation
Low	Low	Expected success. Preserve model unless other evidence contradicts it.
High	Low	Expected inability or constraint. Goal not reached, but causal model may be accurate.
High	High	Unexpected failure. Revise self/world/constraint model before repeating.
Low	High	Unexpected success. Goal reached, but causal explanation is unreliable and should be revised.

3.6 Evidence-backed recursion

The next agency state is obtained by revision, not by simply appending a textual reflection. The implementation should require canonical evidence before changing claims about capability, environment, or constraint.

$$\hat{S}_{t+1} = U(\hat{S}_t, E_{t+1}, e_t^g, e_t^p, R_t)$$

(9)

Update the agency state from new evidence, discrepancies, and resistance classification.

$$\mathrm{FACM}_t : (\hat{S}_t, g_t, u_t, \hat{x}_{t+1}, x_{t+1}) \to (e_t^g, e_t^p, R_t, \hat{S}_{t+1})$$

(10)

The complete supervisory transformation.

4. Resistance as an Engineering Signal

A distinctive part of the model is resistance classification. Robotics systems already detect collisions, faults, tracking error, uncertainty, and dynamic obstacles. FACM adds a supervisory interpretation layer that asks what kind of constraint the evidence supports. The classifier is intentionally conservative.

$$R_t \in \{R_P, R_D, R_A, R_M, R_U\}$$

(11)

R_P passive; R_D dynamic; R_A adaptive-agent; R_M mixed; R_U unresolved.

A static wall is passive. A rolling object or changing terrain may be dynamic without being agentic. Another robot or human that changes behavior in response to the subject robot may support an adaptive-agent classification. Failure alone is never sufficient evidence of another agent.

$$\text{failure} \not\Rightarrow \text{counter-agency}$$

(12)

No anthropomorphic inference from non-zero error alone.

$$\text{adaptive-agent only if } P(E \mid \text{adaptive response}) > \tau \text{ and alternative explanations are weaker}$$

(13)

τ is an experiment-specific evidence threshold, not a universal constant.

5. Supervisory Architecture: FACM Above the Existing Robot Stack

FACM is designed to sit above existing robotic perception, planning, and control. A laboratory should be able to add the architecture without replacing a validated controller. This is especially important for experimental adoption: the same robot can be tested with FACM enabled or disabled while retaining identical low-level autonomy.

$$\text{FACM} \rightarrow \text{planner/skills} \rightarrow \text{controller} \rightarrow \text{robot/environment} \rightarrow \text{evidence} \rightarrow \text{FACM}$$

Supervisory loop

Layer	Existing system role	FACM interaction
Perception / state estimation	SLAM, localization, vision, proprioception, semantic scene estimation	Consumes bounded state/evidence and uncertainty.
Task / motion planning	Behavior tree, symbolic planner, path planner, MPC reference generator	Receives goal/context; returns candidate/selected means and predicted effects.
Control	MPC, PID, whole-body control, learned policy	Executes commanded behavior; remains authoritative for low-level dynamics.
Robot / environment	Physical plant and external world	Produces actual consequences, disturbances, constraints, and interactions.
FACM supervisory state	Agency cognition	Maintains self/world/goal/agency/prediction/error/revision across cycles.

5.1 Relationship to FOX and Sigmund

FOX is the first engineering implementation developed by the author from this formal model. In FOX, a canonical mission loop known as Sigmund governs mission lifecycle, while FACM operates as the agency-cognition sub-loop used during orientation, decision, evaluation, and update. The implementation preserves strict separation between cognition and execution authority. The public contribution here is the generalized architecture, not the proprietary mechanisms by which FOX composes context, routes model providers, orchestrates applications, or enforces its internal security and mission contracts.

The same separation is recommended in robotics: the cognition layer may propose and revise, but physical execution should remain behind the robot's established safety, policy, and control boundaries.

6. Conversion to Modern Software Engineering Practice

The mathematical loop maps naturally to contemporary modular software practice. The critical design decision is to represent agency cognition as explicit state and pure transformations wherever possible, rather than burying the model inside prompts or controller side effects.

Interface / module	Responsibility	Recommended property
AgencyStateEstimator	Build $\hat{S}_t$ from sensor/evidence, memory, self-state and mission context.	Provider-neutral; deterministic path available.
GoalBinder	Resolve g_t from mission and constraints.	Cannot silently rewrite the governing mission.
AgencyProjector	Enumerate/select candidate means u_t.	Capability-aware; bounded by controller/policy availability.
OutcomePredictor	Generate $\hat{x}_{t+1}$ or predicted distribution.	Record prediction before execution.
DiscrepancyEvaluator	Compute e^g and e^p.	Domain-specific distance functions permitted.
ResistanceClassifier	Classify passive/dynamic/adaptive/mixed/unresolved.	Evidence-thresholded; failure alone insufficient.
AgencyModelUpdater	Produce $\hat{S}_{t+1}$.	Evidence-backed; auditable revisions.
CognitionOutcomeProjector	Expose bounded result to operator/AI interface.	Outcome, not private chain-of-thought.

A modern implementation should also follow several ordinary engineering disciplines: dependency injection between the supervisory model and robot-specific adapters; immutable or versioned cognition frames; event/evidence references rather than unverified narrative state; explicit confidence fields; bounded serialization; unit tests for each transformation; replayable mission traces; and a deterministic no-generative-AI mode. Generative AI can improve interpretation or planning, but the formal loop should not disappear when a model provider is absent.

6.1 Minimal reference data structure

```
AgencyCognitionFrame {
 cycle_id
 mission_id
 perception: { state, evidence, uncertainty }
 self_model: { verified_capabilities, limitations, health }
 mission_model: { goal, completion_criteria, status }
 agency_model: { candidates, selected_means, expected_effects }
 prediction: { next_state, confidence }
 observation: { actual_state, evidence }
 discrepancy: { goal_error, prediction_error }
 resistance: { class, confidence, evidence }
 revision: { changed_fields, lesson }
}
```

6.2 Minimal reference algorithm

```
frame = build_state(evidence, memory, robot_state, mission)
```

```
while mission.active:
  goal = bind_goal(frame, mission)
  means = enumerate_capabilities(frame)
  action = select_means(frame, goal, means)
  predicted = predict_outcome(frame, action)

  result = robot_controller.execute(action)
  observed = observe_canonical_result(result, sensors)

  goal_error = compare(goal, observed)
  prediction_error = compare(predicted, observed)
  resistance = classify_resistance(frame, action, observed)

  frame = revise(frame, observed,
           goal_error, prediction_error, resistance)
  publish_bounded_cognition_outcome(frame)
```

7. Optional AI Integration Without Making the Model AI-Dependent

FACM can be implemented entirely with conventional robotics software. This is important scientifically because it permits experiments that isolate the architecture from the capabilities of a particular language model or foundation model. AI can then be introduced as a replaceable cognitive contributor.

$$AI = \varnothing \quad \text{is a valid FACM operating mode} \tag{14}$$

With no generative AI, state estimation, goal binding, prediction, classification, and update can be deterministic, probabilistic, learned, or hybrid. With AI enabled, a model may interpret ambiguous evidence, propose hypotheses, generate candidate plans, explain cognition outcomes, or assist with semantic resistance classification. However, claims about physical execution, capability, or authority should remain evidence-backed. A language model should not be allowed to declare that an action occurred merely because it proposed that action.

For operator-facing systems, the architecture should publish a bounded cognition outcome: what the robot currently understands, what it expected, what happened, what changed in its model, confidence, and recommended next action. This is operational transparency, not disclosure of hidden reasoning traces.

$$O_t = D(\hat{S}_{t+1}, e_t^g, e_t^p, R_t, \kappa_t) \tag{15}$$

Bounded user/operator cognition outcome.

8. Experimental Protocol for Robotics Laboratories

The preferred first evaluation is intentionally simple. FACM should be tested against an otherwise identical baseline using the same robot, perception stack, planner, controller, and mission. The experimental variable is whether the supervisory agency-cognition state is maintained and used for revision.

8.1 Minimum platform

- Differential-drive mobile robot or equivalent simulator (ROS 2 / Gazebo, Webots, Isaac Sim, MuJoCo, or comparable environment).

- Localization and obstacle sensing sufficient for route execution.
- Two or more feasible routes between start and goal.
- Ability to inject static, dynamic, and adaptive interference.
- Logging of predicted outcome, actual outcome, goal error, prediction error, resistance class, and revised state.

8.2 Experimental conditions

Condition	Environment change	Expected FACM behavior
A - Nominal	No injected novelty.	Behave comparably to baseline; minimal cognition overhead.
B - Static novelty	Known route blocked by a stationary obstacle.	Detect goal/prediction discrepancy; revise traversability assumption.
C - Dynamic non-agentic	Obstacle position changes independently of robot action.	Classify as dynamic/unresolved, not counter-agency.
D - Adaptive other agent	A second agent changes interference in response to robot route choice.	Accumulate evidence for adaptive-agent classification.
E - Self-capability degradation	Wheel slip, actuator derating, joint restriction, or simulated damage.	Revise self-model and avoid repeating infeasible actions.
F - Unexpected success	Unmodeled assistance or benign contact causes goal completion.	Keep success but revise causal/prediction model.

8.3 Baseline comparison

The baseline should replan using the same available planning stack but should not maintain the explicit FACM state linking self-capability, pre-action prediction, dual discrepancy, resistance class, and persistent evidence-backed revision. This is a stronger control than comparing FACM to a naive open-loop robot.

8.4 Primary hypotheses

$$T_{recovery}^{FACM} < T_{recovery}^{baseline}$$

H1 - Novel failure recovery

$$N_{repeat\ error}^{FACM} < N_{repeat\ error}^{baseline}$$

H2 - Repeated invalid actions

$$E_{pred}(t+n) < E_{pred}(t)$$

H3 - Model improvement after informative interaction

$$Q_{classification}^{FACM} > Q_{classification}^{baseline}$$

H4 - Resistance interpretation in adaptive multi-agent conditions

8.5 Metrics

Metric	Definition	Why it matters
Mission success rate	Fraction of trials reaching completion criteria.	Measures utility without assuming speed is the only objective.
Recovery time	Time/cycles from novel discrepancy to effective alternate behavior.	Primary adaptation measure.
Repeated-error count	Attempts that reproduce a previously evidenced invalid assumption.	Measures whether reality actually changes the model.
Prediction error	Task-specific distance between predicted and observed outcome.	Measures causal/world-model calibration.
Goal error	Distance between desired and actual state.	Separates mission progress from model accuracy.
Resistance classification F1 / calibration	Accuracy or probability calibration across passive/dynamic/adaptive classes.	Tests conservative counter-agency inference.
Computation overhead	Added latency, memory, and CPU/GPU load.	Determines deployability.
Explanation fidelity	Operator outcome matches canonical logged state.	Tests human-facing transparency.

9. Required Ablation Study

A serious evaluation should not test only 'FACM on' versus 'FACM off.' The model contains several separable mechanisms, and the contribution should survive ablation analysis.

Ablation	Removed element	Expected diagnostic result
A0	Full FACM	Reference.
A1	Prediction model	Goal progress remains possible, but causal learning becomes weaker.
A2	Persistent self-model	Robot may re-discover the same capability limits repeatedly.
A3	Prediction discrepancy e^p	Unexpected success/failure collapses into mission error only.
A4	Resistance classifier	Adaptation may occur, but passive/adaptive distinctions are lost.
A5	Cross-cycle memory	No durable learning; each mission cycle begins effectively naive.
A6	Cognition outcome projection	Autonomy unchanged, but operator transparency is reduced.

10. Relationship to Established Robotics and AI Work

FACM should be evaluated as a composition of ideas that already have strong precedents, not as a claim that prediction, self-modeling, feedback, or adaptive control are new. Model predictive control explicitly predicts future system evolution while optimizing control inputs. World-model approaches learn compact representations that support policy learning. Self-modeling robotics has demonstrated that a robot can infer aspects of its own morphology and revise those models after damage. Active inference and related embodied-cognition work place prediction, preferred outcomes, and action-perception loops at the center of adaptive behavior. Reinforcement learning provides a mature framework for value-based behavior improvement through interaction.

The proposed distinction is architectural. FACM packages a mission-relative self/world model, explicit agency/capability representation, pre-action prediction, separate goal and prediction discrepancies, conservative resistance interpretation, evidence-backed revision, and bounded cognition dissemination into one supervisory loop that can be placed above

heterogeneous planners and controllers. Whether that packaging yields measurable advantages is an empirical question, which is why the implementation protocol and ablation tests are central to the proposal.

Adjacent paradigm	Common ground	FACM emphasis
Model Predictive Control	Forward prediction and constraint-aware control.	Supervisory causal learning across missions and heterogeneous skills; not a low-level optimizer.
World models	Learned internal representation and imagined outcomes.	Explicit mission/self/agency state plus dual discrepancy and evidence-backed revision.
Self-modeling robots	Robot learns its own morphology/capability and adapts after damage.	Generalizes self-model revision to mission, agency, constraint, and other-agent interpretation.
Active inference	Perception-action loop, generative models, preferred outcomes, prediction error.	Engineering-neutral supervisory interface; no commitment to free-energy minimization as the optimization principle.
Reinforcement learning	Learning behavior from interaction and value/reward.	Separates goal success from causal prediction accuracy and preserves explicit cognition state.

11. Safety, Epistemic Discipline, and Known Limitations

The model is intentionally not framed as a test for consciousness. An explicit self-model is an engineering representation, not evidence of phenomenal awareness. Likewise, a resistance classifier that infers adaptive agency is a statistical or rule-based model of interaction, not proof of another system's subjective state.

The main technical risks are familiar: model error, sensor error, reward/goal misspecification, non-stationary environments, compounding uncertainty, overconfident classification, computational burden, and feedback corruption. FACM can make these problems more visible, but it cannot eliminate them. In high-consequence robotics, safety envelopes, verified controllers, interlocks, and human authorization remain independent constraints.

A second limitation is identifiability. The same observed discrepancy may be explained by multiple causes: actuator degradation, terrain, perception error, another agent, or a faulty prediction model. Resistance classification should therefore preserve an unresolved state rather than force a categorical interpretation. A useful agency model must be permitted to say 'unknown.'

12. Implementation Recipe for an Experimental Robot

A robotics team can implement the minimum viable architecture without access to FOX and without reproducing any RMW proprietary subsystem. The following sequence is sufficient for a first experiment:

1. Choose one existing robot task with a stable baseline planner/controller.
2. Define a serialized AgencyCognitionFrame for the task. Start with only robot health/capability, mission goal, selected action, predicted next state, observed state, and two discrepancies.
3. Add a pre-execution prediction hook. The prediction can initially be analytic, heuristic, or learned.
4. After each action, compute goal error and prediction error separately.
5. Persist evidence-backed revisions across cycles. Do not let a natural-language explanation become the source of truth.
6. Add a four/five-class resistance classifier only after the basic discrepancy loop is stable.
7. Run the baseline and FACM conditions under identical novelty injections.
8. Add ablations, then optional AI semantic reasoning as a separate experimental factor.
9. Publish logs and metric definitions so other labs can reproduce the result.

Minimum viable scientific test

If the FACM robot encounters a novel discrepancy, changes its self/world model on the basis of evidence, avoids repeating an invalid assumption, and improves prediction or recovery relative to the same baseline controller, the architecture has demonstrated an experimentally meaningful effect. If it does not, the hypothesis should be rejected or revised.

13. Research Roadmap

The software implementation in FOX establishes that the formal agency loop can be represented in modern code and integrated as a cognition sub-loop inside a larger mission architecture. Robotics is the more demanding test because physical embodiment supplies continuous sensor uncertainty, actuator constraints, energy costs, latency, damage, dynamic environments, and direct interaction with other agents.

A staged research program is therefore recommended. Stage 1 should use simulation and a wheeled mobile robot. Stage 2 should test manipulator capability degradation and unexpected contact. Stage 3 should introduce multiple autonomous agents with deliberately adaptive interference. Stage 4 should compare deterministic FACM, learned-prediction FACM, and FACM augmented by a language or vision-language model. Stage 5 can investigate longer-horizon agency memory, hierarchical goals, multi-agent models-of-models, and human-robot collaborative explanation.

The underlying proposition remains the same across these stages: cognition improves when action is treated as an experiment on reality, and when the answer of reality is allowed to change the agent's model of what it can cause.

14. Conclusion

The path to FACM crossed two domains. The first was philosophical systems theory: an attempt to explain why perception, value, will, action, resistance, and responsibility form a recursive structure rather than a set of isolated concepts. The second was engineering: a long-standing interest in networked objects, synchronized state, modular systems, and coordinated applications. Modern AI agency made the two domains operationally relevant to one another. A network platform could now host a cognitive system capable of reasoning about distributed state, while the agency theory supplied a disciplined way for that cognition to model itself as a causal participant rather than merely generate plans.

FOX provided the first software implementation of that synthesis. The general contribution extracted here is smaller and more useful to the robotics community: maintain an explicit agency state; bind a goal; select means; predict the consequence; execute through established control authority; observe what actually happened; distinguish goal error from prediction error; classify resistance conservatively; revise the self/world model from evidence; and carry that revision into the next cycle.

$$\hat{S}_t \rightarrow g_t \rightarrow u_t \rightarrow \hat{x}_{t+1} \rightarrow x_{t+1} \rightarrow (e_t^g, e_t^p) \rightarrow \hat{S}_{t+1}$$

Agency Cognition Loop

The architecture is intentionally easy to falsify. If explicit agency cognition adds no measurable improvement over a well-constructed baseline under novelty, its engineering value is limited. If it reduces repeated error, improves recovery, calibrates prediction, or enables more accurate interpretation of adaptive interaction, it provides a practical bridge between self-modeling, predictive control, world modeling, and higher-level autonomous cognition. That is the experiment this paper invites robotics engineers to run.

References

[1] J. Bongard, V. Zykov, and H. Lipson, "Resilient Machines Through Continuous Self-Modeling," Science, vol. 314, no. 5802, pp. 1118-1121, 2006. doi:10.1126/science.1133687.

[2] D. Ha and J. Schmidhuber, "World Models," arXiv:1803.10122, 2018.

[3] A. Linson, A. Clark, S. Ramamoorthy, and K. Friston, "The Active Inference Approach to Ecological Perception: General Information Dynamics for Natural and Artificial Embodied Cognition," Frontiers in Robotics and AI, vol. 5, art. 21, 2018. doi:10.3389/frobt.2018.00021.

[4] M. Kiverstein, M. D. Kirchhoff, and T. Froese, "The Problem of Meaning: The Free Energy Principle and Artificial Agency," Frontiers in Neurorobotics, vol. 16, 2022. doi:10.3389/fnbot.2022.844773.

[5] H. Nguyen, M. Kamel, K. Alexis, and R. Siegwart, "Model Predictive Control for Micro Aerial Vehicles: A Survey," arXiv:2011.11104, 2020.

[6] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed., MIT Press, 2018.

[7] T. Van de Maele, T. Verbelen, O. Çatal, C. De Boom, and B. Dhoedt, "Active Vision for Robot Manipulators Using the Free Energy Principle," Frontiers in Neurorobotics, vol. 15, 2021. doi:10.3389/fnbot.2021.642780.

[8] R. B. Murphy, Insight: A Treatise on Perception, Will, Resistance, and the Reality of the Other, completed edition, 2026.

[9] R. B. Murphy, private engineering archive, object-networking design records, 2002-2003. Historical identifiers omitted.