Article

Not peer-reviewed version

A Carrier Manifolds Framework for AGI

Xiaolin Ge *

Posted Date: 17 May 2024

doi: 10.20944/preprints202405.1164.v1

Keywords: Artificial General Intelligence; hybrid systems; logic programming; Large Language Models (LLM)

Article

A Carrier Manifolds Framework for AGI

Xiaolin Ge

Hybrid Algorithms, LLC, Phoenix, Arizona USA; xiaolin.ge@hybridalgorithms.ai

Abstract: This paper integrates carrier manifolds from control theory and hybrid systems modeling to develop a novel mathematical framework for Artificial General Intelligence (AGI). Carrier manifolds represent the state space of complex AGI systems, while hybrid systems model discrete and continuous dynamics together. The proposed architecture embeds carrier manifold into GPTProX, a hybrid AI solver combining conversational AI, logic programming, and optimization.

Keywords: artificial general intelligence; hybrid systems; logic programming; Large Language Models (LLM)

---

1. Introduction

Artificial General Intelligence (AGI) represents a pinnacle in our quest to advance artificial intelligence, with the goal to mirror or exceed human cognitive capabilities. The evolution of Large Language Models (LLM) like GPT has demonstrated exceptional text generation abilities, yet these models frequently lack an explicit mathematical foundation that directly correlates with their output generation mechanisms. This ambiguity is particularly concerning when these models generate content that is unpredictable or misaligned with reality, known as "hallucination." As AI becomes more integral to both critical and routine applications, establishing a robust, mathematically sound framework is essential not just for enhancing dependability but also for fostering transparency and trustworthiness in AI technologies.

The advancement of various technologies has historically been rooted in solid mathematical foundations, from classical mechanics to contemporary computer science. These frameworks offer profound insights into the mechanisms of technology, serving as a roadmap for innovation, problem-solving, and advancement. In contrast, the mechanisms of sophisticated neural networks, particularly those powering LLMs, often remain enigmatic, with inputs processed into outputs through intricate and opaque methods. This lack of clarity is manageable when AI functions in an assistive capacity but becomes untenable as AI assumes more autonomous roles, where hallucinatory or non-transparent outputs are unacceptable.

Introducing carrier manifolds and hybrid systems to AGI aims to bridge this knowledge gap, providing a structured mathematical perspective to scrutinize and comprehend the inner workings of AGI systems. Borrowed from control theory and dynamical systems, carrier manifolds offer a comprehensive framework to represent the state of dynamic, complex systems like AGI. A carrier manifold can encapsulate the diverse states and interactions within an AGI system, affording a unified mathematical structure to monitor and dissect its evolution over time.

Hybrid systems meld discrete decision-making processes with continuous dynamics, presenting an apt model for AGI systems that interact with a world comprising both digital and physical aspects. Positioning AGI within the hybrid systems framework allows us to leverage a vast repository of mathematical theories and tools for designing, analyzing, and fine-tuning these intricate systems. This methodology not only bolsters our grasp and governance of AGI systems but also lays down a structured approach to assure their reliability, safety, and congruence with human ethics and objectives.

Embedding carrier manifolds and hybrid systems at the core of AGI development transcends academic pursuit; it is a critical strategy to ensure that as AI systems gain autonomy, their behavior remains predictable, interpretable, and in alignment with our goals. This mathematical foundation acts as a navigational tool in the AGI expedition, ensuring that as we venture into the novel realms of intelligence, we retain clarity, control, and confidence in the technologies we craft.

 doi:10.20944/preprints202405.1164.v1

This research is ongoing, and details are extensive. For in-depth information, please refer to the provided references due to page limitations.

2. Hybrid Systems and Carrier Manifolds

The concept of a "world model" has emerged as a novel and somewhat nebulous idea, spurred by the advancements in LLM generative AI and the existing gaps in our theoretical grasp of its essence. We propose an approach grounded in energy principles, utilizing the Lagrangian formulation. The 1994 introduction of carrier manifolds has revolutionized their application in multi-agent systems, permitting the use of infinitesimal flow generators as control agents. This approach is in harmony with the Hybrid System Lagrangian framework, underscoring its potential in the evolution of artificial general intelligence (AGI). Within this paradigm, AGI systems are viewed as physical systems whose behavior is determined by the equilibrium between kinetic and potential energies, providing a methodical means to explore and delineate the complex interactions and trade-offs vital for achieving general intelligence. Employing Lagrangian principles, renowned for their capacity to handle constraints and conservation laws, offers a compelling strategy for developing AGI structures adept at managing the intricate state spaces and the multifaceted optimization challenges characteristic of general intelligence.

2.1. Assumptions and Definitions

In our 1994 paper [1], Wolf Kohn and Anil Nerode initially conceived the innovative idea of a 'carrier manifold', which we further elaborated upon. This concept plays a pivotal role in the hybrid systems models and the autonomous agents model for hybrid distributed autonomous control. The carrier manifold essentially encapsulates the essence of a distributed agent autonomous hybrid system. It is within the framework of this manifold that the cost functions requiring ϵ-optimization through measured value control gain their significance. Our paper expanded on how this manifold serves as the foundational structure enabling precise control and optimization within these complex systems.

- Given spaces: goal space G, sensor data space S, controller state space X, control action space C, and interagent message space J.
- Each agent state p is associated with a quintuple (g, s, x, a, j), where:
  - g: current goal
  - s: current sensor data
  - x: current joint state of all agent controllers
  - a: current control action (changes the control law applied to the agent's controller)
  - j: current messages to and from other agents
- Each coordinate of (g, s, x, a, j) is a point in a fixed Euclidean space of finite dimension.
- Interagent messages (j) are represented as Lagrangian terms added to an agent's Lagrangian before minimization using calculus of variations.
- The total number of coordinates in (g, s, x, a, j) is k, associating each agent state with local coordinates in a k-dimensional Euclidean space.

2.2. Manifold Structure

In the context of defining a manifold M, one is provided with a countable collection U_j of open sets where the union of all U_j is equal to M (referred to as charts). Alongside, there exists a set of local coordinate mappings φ_j: $U_j \to V_j$ that homeomorphically project each U_j onto an open V_j in R^n. The construction of these coordinate mappings mandates that the aggregation of local neighborhoods V_j to construct the manifold ensures compatibility. This is reflected in the requirement that for any non-empty intersection $U_i \cap U_j$, the corresponding mapping function between these intersecting sets should belong to a designated class of smooth functions, such as infinitely differentiable functions for a differentiable manifold. Moreover, for these mappings to preserve the structure of open sets in M's finer topology when related to U_j, they need to ensure that the way points are connected remains consistent across different coordinate systems. This requirement is logical as it ensures the preservation of 'closeness' information when transitioning between local coordinate systems. However, achieving this necessitates first the establishment of the appropriate finer topology on M and then the careful crafting of charts and coordinate maps to reflect this structure, a process that is not inherently straightforward.

- The set M of agent states has at least the structure of a manifold, and in examples, it can be designed as a differentiable manifold.
- M is equipped with two topologies:
  1. Standard Hausdorff topology τ_1
  2. Subtopology θ of τ_1, reflecting the ability of a digital control program to distinguish between continuously varying points in M (called the "small topology")

When M is not naturally presented as a subset of k-dimensional Euclidean space, the language of manifolds becomes necessary. Otherwise, agent states can be identified with points in $\mathbb{E}^k$, using the usual $\mathbb{E}^k$-topology as the standard topology and a subtopology of the standard topology on $\mathbb{E}^k$ as the small topology θ.

3. Solution Generation through Convex Analysis

We introduce an algorithm that computes chattering approximations to relaxed optimal controls over Euclidean regions. This algorithm can be extended to carrier manifolds of a wide range of multiple agent autonomous distributed control systems, but for the sake of brevity, we focus on the Euclidean region case.

3.1. Relaxed Optimality

Instead of seeking an exact optimal solution, we relax the optimality requirement to account for the inherent error tolerance (ε) in engineering problems. Our goal is to construct and implement a control function that brings the integral of the Lagrangian along the trajectory within ε of its minimum value.

We begin by convexifying the Lagrangian $L(z,u,t)$ with respect to u, obtaining $L^{**}(z,u,t)$. We have developed a constructive version of the Ekeland-Temam [2] convex analysis proof for the existence of measure-valued (weak) optimal solutions. This constructive proof is used to produce an algorithm that computes ε-optimal chattering solutions to the original non-convexified problem.

When the problem is autonomous (i.e., no dependence on t), controls can be taken as functions of state (Bellman Dynamic Programming), resulting in a finite control automaton. This automaton issues a chattering control to be used for a specified interval Δt, after which another control must be computed. The chattering control is a finite sequence of successive control values c_i, each operative for a specified portion α_i of the time interval Δt.

3.2. The Algorithm

The algorithm is a variant of real linear programming with a new wrinkle. It simultaneously computes a basis for an initially undetermined space and solves a linear programming problem in that space. This is necessary because we aim to compute one of the absolute minima as a convex combination of local minima. We must get sufficiently close to enough local minima and express a global minimum as a convex combination of these local minima to guarantee that the combination is a control yielding an ε-optimal trajectory.

To ensure that the needed time derivative $\dot{z}(t)$ of $z(t)$ lies in the convex hull of the selected w_i's, we use a domain decomposition into tubes. This approach provides a universally correct algorithm.

In [1,2], We presented the α-Finder Algorithm, which introduces new variables β_i and uses the simplex method to solve a linear program [2] with $n+2$ equations, 2^n variables, and constraints. The algorithm iteratively adds new tubes and updates the solution until a feasible solution is found or the termination condition is met.

 doi:10.20944/preprints202405.1164.v1

Once a solution $\alpha_1, \ldots, \alpha_{n+1}$ is found, the desired $\dot{z}(t)$ is constructed on the interval $[t_0, t_0 + \Delta]$ by dividing the interval into subintervals and defining $\dot{z}(t) = w_i$ for t in the i-th subinterval. The procedure is repeated for subsequent intervals, extending $z(t)$ and $\dot{z}(t)$ to the whole interval.

4. GPTProX

We summerize our pending patent GPTProX [3], a software system of GPT and Prolog integration.

GPTProX's architecture is an advanced integration of conversational AI with logic programming, mathematical optimization, and enterprise data systems, designed to deliver precise and validated solutions across various industries. This innovative system is engineered to overcome the limitations of conventional AI, particularly in scenarios where accuracy and reliability are paramount.

At the core of GPTProX is its unique ability to parse responses from a large language model like ChatGPT into logic programming predicates, a process that allows the system to validate responses against a comprehensive knowledge base. This knowledge base is constructed using both manual encoding by domain experts and automated extraction from structured data sources, ensuring that the system's responses are not only contextually relevant but also grounded in verifiable facts.

GPTProX is adept at interfacing with a range of data systems, including financial databases, GIS systems, and Wikipedia. This capability allows it to enrich its responses with up-to-date, real-world information, making it particularly valuable in industries such as energy, finance, transportation, manufacturing, and retail. For instance, in the energy sector, GPTProX can leverage utility grid data to offer actionable insights for outage diagnostics and restoration strategies.

The system's mathematical optimization component is built on a robust framework that supports a variety of problem-solving templates, including linear programming, dynamic programming, and optimal control. When confronted with a query that requires optimization, GPTProX can generate and execute the appropriate mathematical models, ensuring that the solutions it provides are not only contextually accurate but also optimized for the given parameters.

5. Carrier Manifolds Based AGI Implementation

5.1. Architecture of AGI

The architecture of our AGI system employs an integrated approach that merges the precision of formal logic with machine learning's adaptability, embodied in a sophisticated "world model". This model comprises several key components that interact dynamically to create a robust framework for AGI.

5.2. Logic Programming Database

At the core of the architecture is a logic programming database, designated for storing AGI objects. This database utilizes first-order logic to articulate and manipulate the intricate relationships and characteristics of these objects. Given its intrinsic suitability for symbolic reasoning, logic programming forms an ideal foundation for the AGI's knowledge base. Within this framework, the entities and their interactions are delineated through predicates and rules, enabling elaborate reasoning and inference processes.

5.3. Mathematical Formulation and Representation

The integration of mathematical formulations alongside graphic representations into the AGI model boosts its expressiveness and clarity. Mathematical models offer a structured methodology to capture and scrutinize the dynamics of AGI objects. Concurrently, graphic representations furnish intuitive visual insights into the objects and their interplay, assisting in both comprehension and system debugging.

Each AGI object is linked with a carrier manifold, providing an advanced depiction of the object's state and evolution. This manifold offers a structured context where object states can progress and be methodically examined. It also aids in modeling the intricate interdependencies and interactions among various objects, laying a solid groundwork for the AGI's "world model".

5.4. Continuous Enhancement

An API interface connected to a large language model (LLM), such as GPT, acts as a dynamic "model adaptation" engine. This component perpetually refines and updates the world model by assimilating new data, formulating predictions, or suggesting amendments based on the LLM's outputs. The LLM processes natural language inputs, extracts pertinent information, and generates responses that are reintegrated into the logic programming database, facilitating the model's evolution and adaptation.

Designed for incessant self-improvement, the system evolves through constant interaction with the LLM, which refines its outputs based on ongoing feedback. This cyclical process ensures that the AGI system progressively enhances its models and strategies, adapting to new data and learning from its interactions.

The synergy of these elements equips the AGI system to navigate the complexities of artificial general intelligence. With a strong foundation in logic programming, advanced modeling through carrier manifolds, and continual learning via an LLM, the system is poised for sophisticated reasoning and dynamic interaction with its environment, marking a stride in the development of robust and adaptive AGI.

6. Learning and Adaptation in AGI Models

A pivotal aspect of advancing AGI systems is their ability to learn and adapt dynamically. The integration of GPTProX's three-pronged training architecture offers a robust framework for enhancing the learning capabilities of carrier manifolds based AGI systems. This section elucidates how Inductive Logic Programming (ILP), SpaCy [4] for text-to-logic conversion, and LLM finetuning synergistically empower AGI systems to refine their knowledge and adapt to new information.

6.1. Online Learning of Behavior Functions and Carrier Manifolds Representations

The AGI system employs ILP to extrapolate behavior functions and refine carrier manifolds representations continually. This process enables the system to:

- Derive new logical rules from incoming data, enhancing the system's understanding of complex dependencies and scenarios.
- Integrate and adapt to new information dynamically, ensuring that the carrier manifold reflects the most current knowledge.
- Utilize induced rules for improved query answering and insight generation, bolstering the AGI's reasoning capabilities.

6.2. Utilizing ChatGPT's Learning Capabilities for Refining Carrier Manifolds

By harnessing ChatGPT's advanced learning mechanisms, the AGI system can:

- Parse natural language inputs with high accuracy using SpaCy, converting text into structured logic that informs the carrier manifold.
- Fine-tune its domain-specific knowledge through LLM customization, enabling more precise and context-relevant responses.
- Continually evolve its knowledge base, ensuring that the carrier manifold representation stays relevant and comprehensive.

6.3. Incorporating Feedback and Corrections from Human Experts

Human expertise plays a crucial role in the learning loop, allowing for:

- Validation and refinement of the AGI's knowledge and logic rules derived through ILP.

 doi:10.20944/preprints202405.1164.v1

- Enhancement of the system's adaptability by incorporating expert feedback into the learning process, ensuring that the AGI's outputs align with real-world expectations and expertise.

6.4. Transfer Learning Across Agents for Efficient Knowledge Sharing

Transfer learning is employed to disseminate knowledge across various agents within the AGI system, facilitating:

- Rapid assimilation of domain-specific expertise from one agent to another, enhancing collective intelligence.
- Efficient sharing of learned behaviors and manifold representations, reducing redundancy and accelerating the learning process across the AGI ecosystem.

7. Testing and Evaluation

Rigorous testing and evaluation of carrier manifolds based AGI systems is critical before real-world deployment. Comprehensive benchmark tasks and simulated environments must be developed to quantitatively and qualitatively measure the AGI's cognitive performance across domains. Comparisons against human baselines and other AGI approaches are needed to validate effectiveness. Specific evaluations should assess learning of manifold representations, knowledge transfer, ethical constraint adherence, transparency, and general intelligence traits like curiosity and self-awareness. Interpretability analyses can diagnose vulnerabilities before consequential deployments. Establishing clear performance thresholds on these criteria is vital for responsibly transitioning such AGI systems from research to impactful applications.

8. Ethical Considerations and Societal Impact

The carrier manifold framework introduces mathematical transparency into AGI systems by explicitly modeling the state representations and dynamics of AI agents on manifold structures. This transparency is crucial for ensuring AGI alignment with human values, enabling scrutiny of an AI's reasoning process and allowing course-corrections for undesirable behaviors. The hybrid systems modeling further allows implementing ethical constraints and secure overrides at key decision points. While promoting interpretability, this transparency also creates potential vulnerabilities that require balanced frameworks managing transparency versus security risks. As AGI capabilities grow, interdisciplinary analysis of socioeconomic impacts through ethical, legal, and policy lenses becomes imperative to uphold equality, justice and human autonomy. Ultimately, integrating ethical architectures like carrier manifolds is vital for developing reliable, value-aligned AGI beneficial to humanity's future, but must be complemented by proactive governance co-evolving with the technology itself.

9. Conclusion and Future Directions

The integration of carrier manifolds and hybrid systems modeling provides a novel and mathematically grounded framework for advancing Artificial General Intelligence. By representing the complex state space of AGI systems through carrier manifolds, this approach allows for a structured and transparent understanding of an AI's inner workings and decision-making processes.

The proposed architecture, combining logic programming, optimized solution generation, and large language model integration, lays a robust foundation for developing AGI systems that are adaptable, reliable and aligned with human objectives. The synergistic combination of these components enables continuous learning, refinement of world models, and coherent reasoning across diverse domains.

While the concepts and methodologies outlined represent promising strides, their full potential can only be realized through extensive research, implementation, and empirical evaluation. Future efforts focus on constructing benchmark tasks, test environments, and quantitative/qualitative measures to rigorously assess the cognitive performance of AGI systems built on this framework. Comprehensive

 doi:10.20944/preprints202405.1164.v1

comparisons against human cognition and other AGI approaches will be instrumental in validating its effectiveness.

Transparency, interpretability, and human control over autonomous AGI decisions remain paramount concerns that must be meticulously addressed. Developing clear guidelines, risk mitigation strategies, and ethical frameworks for the responsible development and deployment of such powerful AI technologies is an imperative that cannot be overstated.

Looking ahead, integrating insights from neuroscience, cognitive science, and other disciplines could potentially enrich and expand the capabilities of carrier manifolds based AGI systems. Exploring novel mathematical formalisms, optimization techniques, and machine learning architectures to further elevate the reasoning and learning abilities of these AI agents is an exciting avenue for future research.

Ultimately, the proposed framework represents a conceptual advancement towards engineering reliable, ethical and broadly capable artificial general intelligence. However, the road ahead remains arduous, necessitating a concerted, multidisciplinary effort to navigate the intricate challenges and profound implications that AGI entails for humanity's future.

References

1. Kohn, W.; Nerode, A.; Remmel, J.B.; Ge, X. Multiple agent hybrid control: carrier manifolds and chattering approximations to optimal control. Proceedings of 1994 33rd IEEE Conference on Decision and Control, 1994, Vol. 4, pp. 4221–4227. doi:10.1109/CDC.1994.411614.
2. Ge, X.; Nerode, A.; Kohn, W.; James, J. Distributed intelligent control theory of hybrid systems. Hybrid Systems III. IEEE, 1994, pp. 76–100. doi:10.1109/AIHAS.1994.390507.
3. Ge, X. GPTProX, A Hybrid Enterprise AI Solver. Patent number 63539102. Pending, filing date: 09/19/2023, 2023. USPTO.
4. Honnibal, M.; Montani, I. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. open source software package available: https://spacy.io/.