GlobalLogic®
A Hitachi Group Company

Agentic AI's Coming of Age

From Rocket Landings to Intelligent Enterprises: Understanding the Complexity of Agentic AI

Contents

The pursuit of the stars has always been a testament to human ingenuity and the relentless desire to push beyond known boundaries.

As SpaceX's latest Starship spacecraft soared into the skies and returned with precision, it wasn't just a milestone in aerospace engineering—it was a vivid illustration of advanced control systems operating at the pinnacle of complexity.

This achievement shows that an autonomous system can successfully control a sophisticated system and steer it to defined goals by combining key aspects of the agentic AI: autonomy, goal-oriented behavior, awareness of the environment, and decision-making power.

In the case of the Starship, the two other elements of the agentic AI—adaptability and the ability to act upon the environment to effect change—are concentrated within the control center and the development teams.

Can we replicate this impact, reliability, and precision with a more down-to-earth application of GenAI-enabled agentic AI to enterprise systems?

What challenges arose in the production implementations, and what lessons can we learn and apply from the control theory to LLM-powered agentic AI systems?

Contents

A Short Recap of the Control Theory

Control theory brought humanity into the industrial age, where complex dynamic systems with intrinsic machines and mechanisms inside could be effectively self-controlled through a feedback loop.

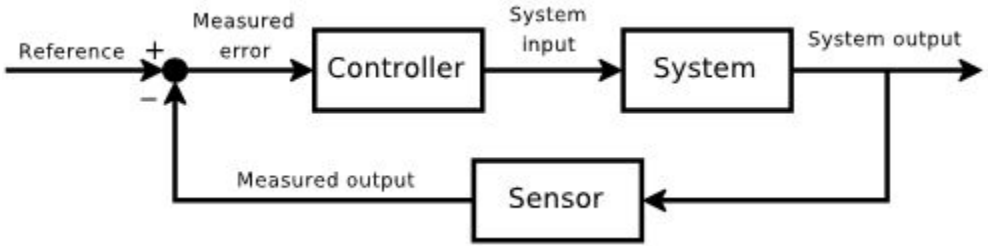
It started with windmill controls analyzed by James Clerk Maxwell in 1868 (a system that controls the fuel intake to maintain the constant speed of the rotation) and eventually resulted in control systems for spaceflight, nuclear stations, and country-wide electric grids.

At the heart of any classical autonomous system lies control theory—a mathematical framework that governs how systems respond to inputs to achieve desired outputs.

Control theory provides the tools to design systems that can maintain stability, optimize performance, and adapt to changes in their environment.

It is the backbone of engineering disciplines ranging from aerospace to robotics, enabling the precise maneuvering of spacecraft, the stabilization of drones, and the regulation of industrial processes.

In the most primitive form, control theory guides the design of the systems such as seen in the image below. The desired state at any given moment (e.g., parameters of the landing for a spacecraft) is continuously measured by Sensors, the desired state is compared with the planned, calculating the error to be transformed into the corrective actions generated by the Controller used as an input to the System. With the right design, the rocket will land where and how it is supposed to land–most of the time.



The key principles of control theory include:

Mathematical Modeling

Control systems are built upon precise mathematical models that describe the dynamics of the system.

These models, often expressed as differential equations, enable engineers to predict how a system will respond to various inputs.

Feedback Loops

A fundamental component of control systems, feedback loops allow a system to adjust its behavior based on its output.

By comparing the desired outcome (setpoint) with the actual output, the system can minimize errors through continuous adjustments.

Controllability, Stability, Observability, and Robustness

Control theory emphasizes designing systems that remain stable, monitored, and controlled. under a variety of conditions.

Robust control systems can handle uncertainties and external disturbances without deviating from their intended performance.

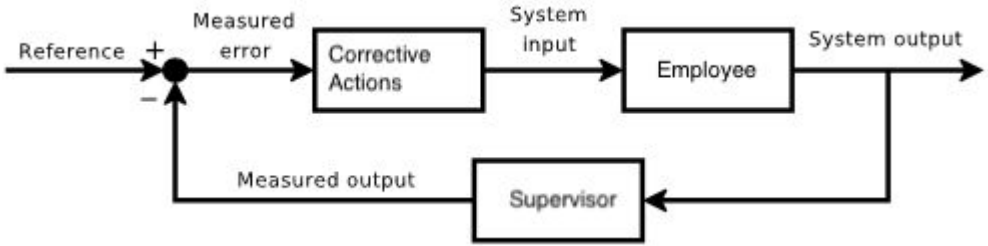
Predictability and Determinism

Traditional control systems operate under deterministic principles, where the same input will consistently produce the same output.

This predictability is crucial for critical applications like spaceflight, where reliability is non-negotiable.

Contents

An enterprise is also an ecosystem of multiple subsystems. Employees who perform tasks, along with company policies and objectives, constitute a complex system. In the paradigm of control theory, the **System** is an employee performing a task, the **Controller** is the corrective actions to improve the employee's performance, and the **Sensor** is the supervisor.



The employee can be tasked to produce an analytics report, and the supervisor would read the report and measure the "error" against the "perfect" report that the company usually sends to its customers. The Controller role can be performed either by the supervisor (that's what we call micromanagement), or the employee. A more seasoned or advanced employee can detect and sense the deviation using only the initially stated goals and without the need of the supervisor.

How can we replicate this goal pursuit with an agentic AI?

Contents

Let's examine what makes a GenAI-powered agentic system different from a classical, non-GenAI agentic system, and explore what properties of LLMs make the above control theory principles so hard to implement.

As we venture into the realm of agentic AI and generative AI systems powered by LLMs, we encounter a paradigm shift.

Unlike traditional control systems, LLM-based systems exhibit a level of complexity and unpredictability that challenges conventional engineering approaches. LLM-based systems also exhibit properties that make them different from traditional software systems, even those of high complexity.

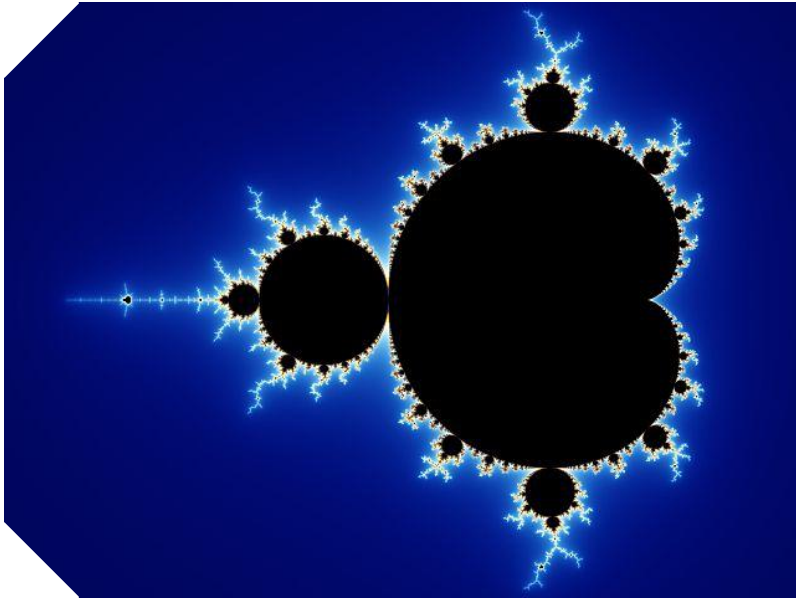
The following are some of the key challenges.

1. Computational Irreducibility

Computational irreducibility implies that the only way to determine the outcome of a process is to perform each computational step; there are no shortcuts or simplified models that can predict the end state without executing the full computation.

LLMs, like GPT-4, are so complex and their internal workings so intricate that we cannot predict their exact outputs without actually running them.

Their behavior cannot be encapsulated in simplified mathematical models akin to those used in control theory, where controllers can be designed to anticipate system responses to inputs and disturbances.



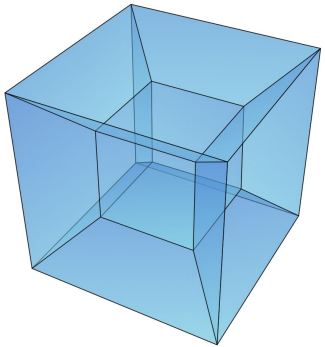
2. Nonlinearity and High Dimensionality

LLMs operate in high-dimensional vector spaces, processing inputs through multiple nonlinear transformations. This makes their internal decision-making processes opaque and their outputs sensitive to minute changes in input.

LLMs have high dimensionality of the inputs and outputs.

For Llama 3.1, the embedding vector size is 4096, where the token vocabulary size is 128,256. A simple input sentence is about 100 tokens, and a megaprompt that includes not only the actual input size, but also all of the supporting system prompt content can have 2,048 or more tokens.

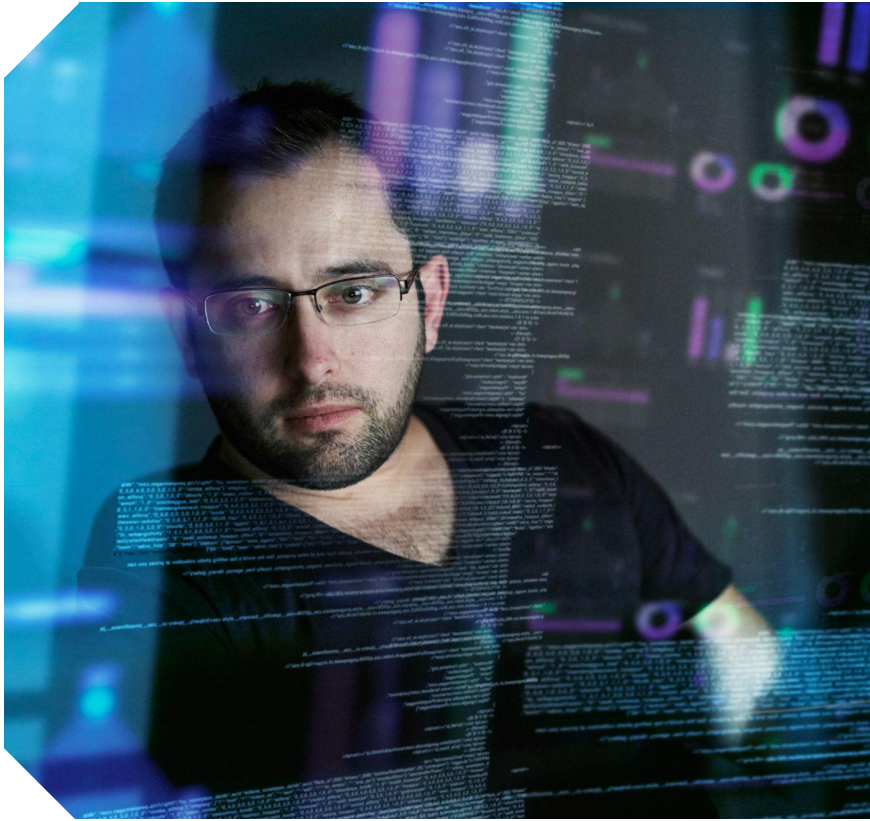
This results in the staggering 8 million dimensions for the input. For agentic components, such as text-to-SQL, it is easy to imagine the difficulty of testing and optimizing the performance of this single component of a more complex, AI agentic system that needs to query the data.



3. Separation of Code and Data (or Lack Thereof)

Most computer systems have a clear separation of code and data, as per Von Neumann Architecture. This is an extension of the classical dynamic control systems, where input data from sensors and the control algorithms are equivalent to the input and the code in software systems, accordingly. That is, unless you program in Lisp.

LLM-based systems, unfortunately, mix the code and the input data. In a typical RAG system, the retrieved data is formatted and added to the prompt. The prompt of an LLM system is effectively the code, and mixing the data and the code introduces a host of testing, reliability, and security issues. This mixing of ever-growing data sets with the prompts introduces variability that is difficult to model and predict, in addition to the intrinsic input dimensionality problem of LLMs described previously.



4. Stochastic Behavior

LLMs may produce different outputs for the same input due to factors like randomness in sampling methods during generation. This stochastic nature contrasts with the deterministic outputs expected from traditional control systems, as well as from traditional software systems.

It is no surprise that this sensitivity of software systems to random data and code changes resulted in implementing error correction in the hardware (ECC memory used in servers versus regular, non-correcting memory), as well as using distributed and redundant computation algorithms accounting for these issues.

5. Feedback Loop Problem and System Stability

In the traditional systems, including the latest SpaceX Starship, feedback loops are highly effective due to the predictability of system responses. Adjustments made based on feedback can be reliably calculated to bring the system closer to the desired state, which means a successful atmospheric reentry and landing.

For LLM-based systems, feedback loops may be less effective because the system's response to feedback is not easily predictable. The lack of a simplified model (see irreducibility above) makes it challenging to design feedback mechanisms that can consistently guide the system toward desired outcomes.

Contents

Practical Recommendations

As enterprises seek to harness the power of LLMs to enhance productivity and innovation, they must grapple with the challenges posed by computational irreducibility and unpredictability. What follows are our **top nine recommendations** for building a functional agentic AI.

1. Use Domain-Specific Languages (DSLs)

For inner workings, human language is not the best. DSLs are best suited to minimize dimensionality while preserving the richness of the actions and environment.

Structured JSON can be the simplest way, or it can be more sophisticated DSLs. Limiting the token generation to DSL-compliant output allows LLM output to be more reliable without affecting performance.

2. Use Natural Language for Interactions with Humans Only

LLMs perform best when communicating with humans and understanding their intent. LLMs are also great at translating structured data to unstructured and back.

For communication between agents, it is best to use non-human, structured data exchanges.

3. Fuse Traditional AI/ML with GenAI

LLMs, while slowly improving, are not created for complex reasoning.

Classical data-driven machine learning, combined with mathematically precise logical computations, such as predicate logic, dynamic programming, or a variety of other methods, adds the necessary grounding to the stochastic nature of LLMs.

4. Prioritize Safety and Reliability

The lack of predictability raises concerns about the safety of agentic AI systems.

Implementing robust safety protocols and fail-safes becomes crucial. Specific tolerance to wrong agentic decisions must be nuanced and use case-specific.

5. Ensure Ethical Alignment

Aligning the goals and actions of agentic AI with human values is more challenging when the AI's decision-making process is not fully transparent or predictable.

LLMs in their current form lack comprehensive ways to interpret and understand the internal workings, so adding guardrails on the input and the output (decisions) of the agentic system can address ethical concerns without jeopardizing the efficiency of the inner operations.

6. Introduce Effective Feedback Loops

Embracing adaptive control mechanisms that can adjust to the dynamic behavior of LLMs, and learning from their outputs to improve system performance over time, is critical for robust AI agents.

LLM-as-a-judge is an excellent way to create feedback loops for LLM-powered agentic flows.

7. Implement Adaptive Human-in-the-Loop (HITL)

For critical agentic decisions with significant consequences, humans must validate the AI agent's decision.

In the SpaceX example, the decision not to guide the booster back to the launchpad and land it in the ocean was made by a human, given the telemetry and AI-powered analysis.

8. Implement Comprehensive Observability

Even when HITL is not needed, full journaling of agentic decisions must be performed for several reasons: debugging any issues, improving agentic quality, and, coincidentally, addressing the key requirements of AI governance.

9. Embrace Modular Architecture

Designing agentic AI systems with a modular architecture enhances scalability, maintainability, and reliability.

By structuring the system into distinct agents with well-defined interfaces, each architecture component can perform specific tasks independently while communicating seamlessly with others.

Conclusion

This new frontier demands a collaborative effort between disciplines—melding the rigor of control theory and the best practices of software engineering with the advances of GenAI. By understanding the limitations and potentials of each approach, we can design agentic systems that are not only powerful but also safe, reliable, and aligned with human values.

We help enterprises harness the potential of GenAI with solutions that are:

Responsible

AI you can trust—ethical, transparent, and aligned with your governance and security needs.

Reliable

AI that performs consistently, delivering accurate results even as conditions change.

Reusable

Scalable AI with modular, reusable frameworks that save time and drive faster results across your business.

Learn more: globallogic.com/genai

About the Author

Yuriy Yuzifovich, Chief Technology Officer, AI

Yuriy Yuzifovich serves as GlobalLogic's CTO, AI, driving the company's AI vision, strategy, and growth roadmap while incubating deep-tech capabilities. With 25 years of experience, Yuriy brings expertise in technology consulting, AI, and data-driven product development.

Before GlobalLogic, he led Alibaba Cloud's efforts to transform data silos into actionable insights, developing advanced cybersecurity solutions. At Akamai, he built AI systems to process telecom data, enhancing security for major ISPs.

Yuriy holds a master's in computer science and an MBA in quantitative finance from Washington University in St. Louis. Based in Los Gatos, he enjoys hiking and fishing in his free time.

GlobalLogic, a Hitachi company, is a trusted digital engineering partner to the world's largest and most forward-thinking companies.

We helped create some of the most innovative and widely used digital products and experiences. Today we're helping clients transform businesses and redefine industries through intelligent products, platforms, and services.