Control Theory of Large Language Models

Domenic Rosati[1,2]

[1]Dalhousie University (SAIL, HyperMatrix) [2]Vector Institute for Artifical Intelligence

November 10, 2024

Why should I care about control theory?

The goal of this presentation is to help you think of LLMs problems in a new control theoretic way.

If you are trying to get a ML system to do anything, you already do care about control theory.

1. **(Inference Time Control)** → Find prompts to control (text generation) behaviour.
2. (Training Time Control) → Find parameters to control (text generation) behaviour.

Case Study: Prompt Engineering

Definition (Prompt Engineering (Informal))

A search for the best input (or input structure) to get the right output from a large language model.

Question: Why does prompt engineering work? How does it work? What are its limits? Where do we start an investigation? How do we think of the problem? Can you find a prompt that will make an LLM output anything?

Typical "Theory": Matches Training Distribution
Typical NLP Perspective: Empirical post-hoc experimental design
Examples:

1. Format following (JSON) and code in pretraining corpora
2. Chain-of-thought matches reasoning-style distribution

Control Theoretic: Characterize System Dynamics

Outline

LLMs as Dynamic Systems	(Soatto et al. [5])
Reachability	(Soatto et al. [5])
Controllability	(Soatto et al. [5])
Optimal Control	(Luo et al. [3])
Limits of Control	(Bhargava et al.[1])
Control Engineering	(Kong et al.[2] / Miyaoka and Inoue [4])

LLMs as Dynamic Systems

A system in control theory behaves according to a dynamic rule (also called trajectory, or evolution rule) of the following form: $x_{t+1} = f(x_t)$ where x is an input, $f(x)$ is an output and t is a time parameter.

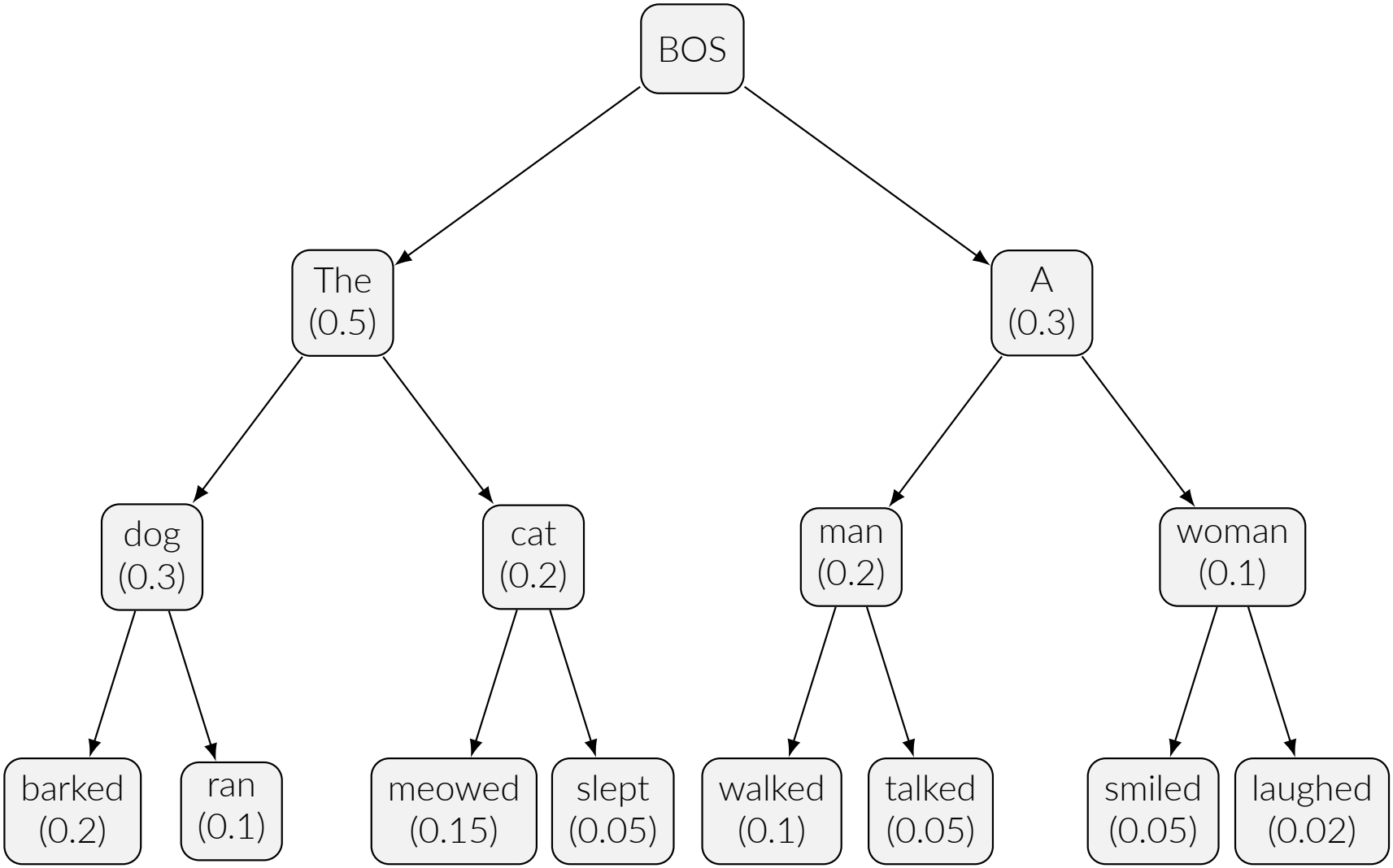
Language Model: weights w that parameterize the following probability distribution: $P_w(x_t|x_{1:t-1})$ for the tokens x in a sequence of tokens from 1 to t.

Autoregressive sampling systems [5]: For a sequence of tokens x and language model with weights w.

$$x_{t+1} = y \sim P_w(y|x_{1:t})$$

The system of autoregressive sampling with a LM w is a dynamic system starting at x_0 with token trajectories $x_{1:}$. For a sequence length of k we have $\mathcal{V}^k$ possible token sequences in vocabulary $x_i \in \mathcal{V}$.

Dynamics Visualized



Reachability for LLMs

Reachability: In control theory, we are concerned with how to characterize the possible outputs that can be achieved given a set of inputs.

Reachability for LLMs [5] The reachable set $\mathcal{R}(x_{0:n}) \subset \mathcal{V}^k$ is the set of all possible token sequences that can be generated with autoregressive sampling given the initial $x_{0:n}$ set of tokens.

Clearly if our sampling was uniform random then we can reach any set of tokens given enough time. Similarly if our sampling was greedy, our reachable set length is 1 given a fixed k.

Often we want to know if a desired output sequence of tokens $y \in V^k$ is in that reachability set $y \in \mathcal{R}(x_{0:n})$ and how much probability mass is placed on it (likelihood of generation).

Reachability Tool for Analysis

Definition (θ-reachable [5])

For some positive threshold $\theta > 0$, a θ-reachable set is the set of sequences that can be reached with probability greater than θ. $\mathcal{R}_\theta = \{y \in \mathcal{R}(x_{0:n}) \; s.t. P_w(y|x_{0:n}) \geq \theta\}$

Measuring these reachable sets require exhaustive search on an exponential sequence set. We will address this later but we can start to see its utility in formulating questions about text generation.

Safety: For harmful text sequences $\bar{\mathcal{S}}$ and a θ. Which sequences $Y \subseteq \bar{\mathcal{S}}$ are part of the θ-reachable set $R_\theta(x_{0:n})$ for an initial sequence of tokens $x_{0:n}$?

Jailbreaks: There exists a set of tokens p such that for an LLM where $\bar{\mathcal{S}} \not\subset R_\theta(x_{0:n})$ and $\bar{\mathcal{S}} \subset R_\theta(p + x_{0:n})$.

Controllability

The problem of finding a jailbreak is a special case of prompt engineering and in order to formalize this problem we need to define **controllability**.

[5] A LLM is ***controllable*** if there exists a set of initial tokens $x_{0:n}$ and a desired output y such that $y \in \mathcal{R}_\theta(x_{0:n})$ with $\theta \approx 1$ within finitely many steps.

Notice that controllability has the following two additional conditions to reachability: (1) Probability of 1 and (2) finite many steps. Otherwise our control is not optimal and is expensive.

Prompt Engineering as Optimal Control

[3] uses controllability to characterize Prompt Engineering as an Optimal Control problem.

Given an evaluation function (for example accuracy, token overlap, or a classifier) $f(\hat{y}, y) \to \mathbb{R}$ where $\hat{y} = \phi_w(z)$ and y is a reference output and ϕ_w is our LLM. Find the prompt z that maximizes this evaluation function:

$$\underset{z}{argmax}\ f(\phi_w(z), y)$$

To this we can add a few modifications for example we might need to minimize the number of tokens in z to control for budget or we might want to maximize the number of prompts that that in a prompt set that maximize f.

[3] discusses methods for enlarging the prompt set over a multi-turn dialogue for complex tasks.

$k - \epsilon$ controllability

Unfortunately, control and reachability in discrete spaces are difficult to measure when the state of inputs and outputs are very large.

[1] provide a way to empirically show a lower bound within a given dataset of a reachable set.

Given a dataset of input-output pairs $\mathcal{D} = \{(x_i, y_i)\}_i^{|N|}$, a LLM is controllable w.r.t $\mathcal{D}$ if $y \in \mathcal{R}_\theta(x)$ where $\theta = \epsilon$. Additionally x is composed of a control portion u and a context portion x_0 and $|u| \leftarrow k$.

It turns out that this is very similar to measuring attack success rate of jailbreak attacks on large language models. Where a jailbreak prefix, the control portion u, is concatenated with the input x_0.

Experimental Evaluation

The authors of [1] evaluate whether 5000 sequences from Wikitext are reachable. Using two different discrete prompt optimization methods, Greedy Back-Generation ($k \leq 3$) [1] and Greedy Coordinate Gradient [6] to find prompts for Falcon 7b.

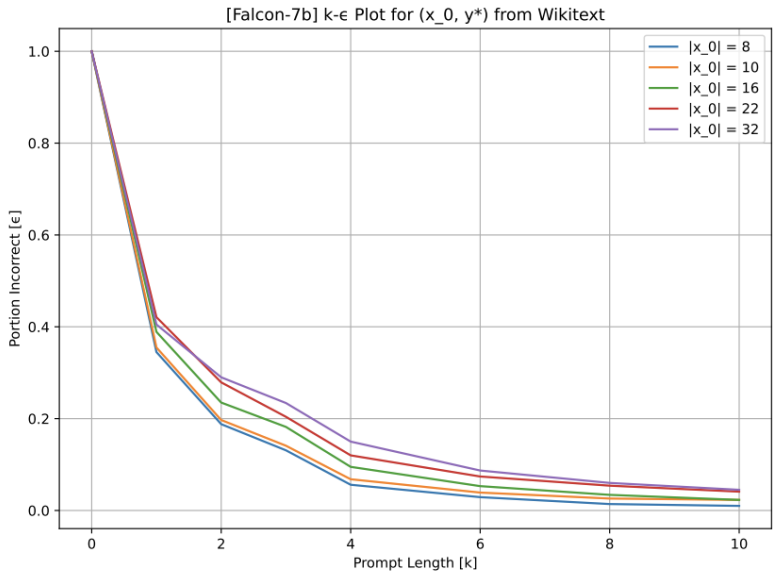


Figure 1. The authors find that 97% of samples are reachable indicating high controllability of this LLM under these methods.

Self-Attention Control Theorem: Intuition

Under what conditions are certain tokens y reachable or not?

Observation 1: Self-Attention determines whether the control tokens in u impact additional context x_0 in the representation space. Since there exists an post-attention embedding Y^* that results in generating y, then in order for y to be reachable by $u; x_0$, Y^* needs to be reachable by $Y_u + Y_x$ which are a decomposition of a post-attention outputs.

Observation 2: Y^* is in a vector space. In order to reach that vector using control vector u and a context x we must have $Y_u + Y_x = Y^*$. If the orthogonal component w.r.t Y^* of Y_u and Y_x do not equal 0 (norm) then Y_* can't be reached.

Observation 3: The eigenvalues of the transformation matrices in attention determine the scaling transformations of Y_x so the vector norm $||Y_{x,\perp}||$ must be bounded by those eigenvalues in order for $Y_u + Y_x = Y^*$ to hold.

Self-Attention Control Theorem: Intuition

Theorem (Self-Attention Control)

Y^* is unreachable for any control input u if $||Y^i_{x,\perp}|| > k\gamma(X, \phi)$ for any i token representation.

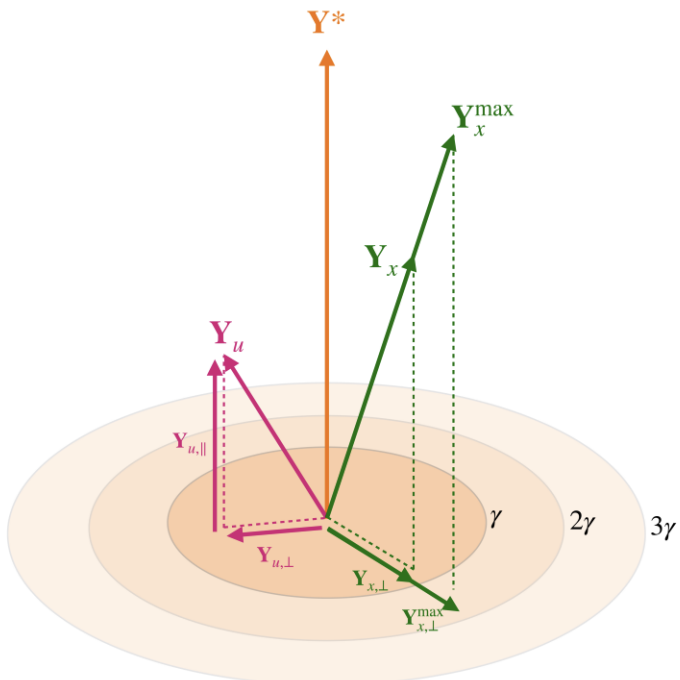


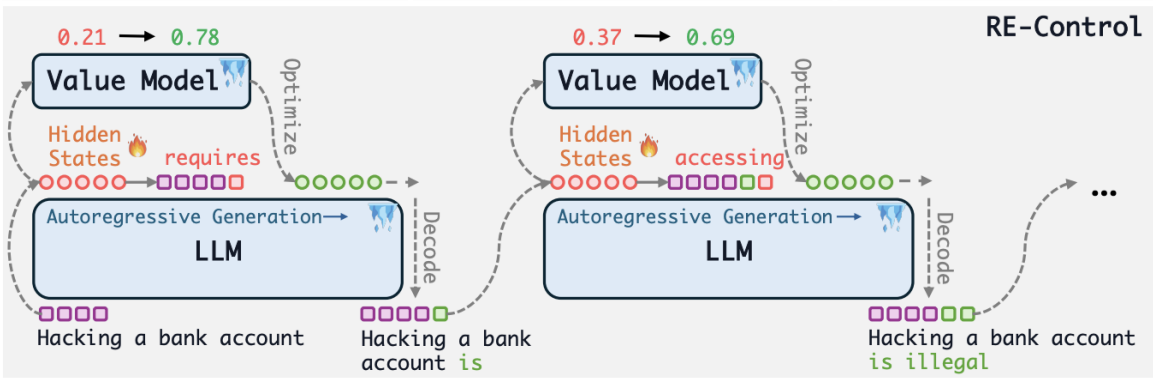
Figure 2. **Implication**: the more tokens k allowed for u the easier it is to control a transformer.

Control Theory →Control Engineering

Design Concerns: How do we control sequence generation trajectories for safe text generation using insights from control theory?

Controllers: control systems that impose a control on a dynamical system to achieve some goal.

Re-Control [2] finds the optimal steering vector to apply to an LLM to achieve a safety target using a value model.



Constraints

Control Barrier Functions (CBF) Control Barrier Functions are any function $h : \mathbb{R}^{\times} \to \mathbb{R}$ that have the following conditions for a safe $\mathcal{S}$ and unsafe set $\bar{\mathcal{S}}$.

$$h(x) \geq 0, x \in \mathcal{S} \tag{1}$$

$$h(x) < 0, x \in \bar{\mathcal{S}} \tag{2}$$

If there exists a control u the guarantees $h(x) \geq 0$ holds.

This motivates the design of a control filter CBF-LLM [4] which ensures safe control by reweighing LM token probability before sampling through the use of a filter function.

Roadmap for Control Theory for LLMs

1. Tractable measures of Reachability
2. Limits of Controllability
3. Applications beyond Prompt Engineering
4. **Control Theoretic Analysis of Training LLMs***

References I

[1] Aman Bhargava, Cameron Witkowski, Shi-Zhuo Looi, and Matt Thomson. What's the magic word? a control theory of llm prompting, 2024.

[2] Lingkai Kong, Haorui Wang, Wenhao Mu, Yuanqi Du, Yuchen Zhuang, Yifei Zhou, Yue Song, Rongzhi Zhang, Kai Wang, and Chao Zhang. Aligning large language models with representation editing: A control perspective, 2024.

[3] Yifan Luo, Yiming Tang, Chengfeng Shen, Zhennan Zhou, and Bin Dong. Prompt engineering through the lens of optimal control, 2023.

[4] Yuya Miyaoka and Masaki Inoue. Cbf-llm: Safe control for llm alignment, 2024.

[5] Stefano Soatto, Paulo Tabuada, Pratik Chaudhari, and Tian Yu Liu. Taming ai bots: Controllability of neural states in large language models, 2023.

[6] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023.