

制御器チューニングにおける LLM エージェント手法の提案と評価

九州工業大学 ○尾郷樹 古賀雅伸

A Proposal and Evaluation of LLM-Based Agent Method for Controller Tuning

Abstract: In this study, we proposed a PID control system whose controller is tuned using an LLM-based agent. Using the benchmark dataset ControlEval for control design and evaluation, we conducted performance assessments in terms of accuracy and computational cost measured by token usage, and discussed the obtained results.

1 はじめに

PID 制御は、その構造的な単純さや、適用範囲の広さから、産業界において最も普及した制御手法の一つとして位置づけられている。実際の運用において、現場で直感的にパラメータをチューニングできる点において、他の制御手法に比較して優れた実用性を有する。しかしながら、PID 制御の制御性能は、チューニングの巧拙に大きく依存するという特性を有しており、現場のエンジニアの経験則的な知識や技能が必要不可欠である。

近年、OpenAI の GPT モデルに代表されるように、LLM (Large Language Model) は目覚ましい発展を遂げている。GPT-4o をはじめとした GPT モデルは、大学学部レベルの制御工学の問題に対して、平均して約 60% の正答率を示している [1]。そのため、従来は人間が担っていた、定性的かつ経験的ともいえるチューニング作業を LLM に代替できる可能性がある。これまで、LLM から望ましい出力を得るための入力 (プロンプト) を設計する技術であるプロンプトエンジニアリングが研究されてきたが、最近では、自律的な AI を目指した LLM エージェント手法が注目されている。そこで本論文では、LLM エージェント手法を用いた制御器チューニング手法の提案および評価を行う。さらに、正答率および使用トークン数の観点から考察を行う。

2 関連研究

2.1 LLM エージェント

LLM エージェント (LLM-based Agent) は、LLM を中核に据えて構築された自律的な AI である [2]。LLM エージェントの基本構成は以下の 4 要素に整理できる。

- プロファイル (Profile)：役割や性格、目標を定義する設定層
- メモリ (Memory)：短期・長期記憶を通じて、過去の情報を保持し学習する機能

- 計画 (Planning)：目標達成のためのステップを生成し、柔軟に修正できる推論機構
- 行動 (Action)：外部ツール・API・環境と連携して実際に行動を実行する層

LLM エージェントの台頭により、LLM は単なるテキスト生成器という従来の枠組みを超え、産業・学術分野の両面での応用が期待されている。

2.2 ControlAgent

工学分野においても、LLM エージェントの応用が進められている。[3, 4] では、複数の LLM エージェントが協働し、それぞれが専門的役割を担うことで、複雑な工学課題を段階的に解決する手法が提案された。人間の専門家が行う分析・設計プロセスを LLM エージェントが模倣することで、高度な工学的タスクを解決できることが示唆されている。

また、[5] では、LLM エージェントを活用した PID 制御系のゲインチューニング手法「ControlAgent」が提案されている。ControlAgent は、複数の LLM エージェントから構成され、制御系設計を自動化するためのフレームワークである。従来、人間が行う逐次的な設計プロセスを模倣することで、完全自動の制御器設計を実現する手法が提案されている。

しかしながら、ControlAgent の問題点は次のように集約できる。第一に、本手法において初期ゲインの決定は LLM に一任されており、具体的なアルゴリズムに基づいた決定法でない。そのため、制御対象の特性に適合しない方向へチューニングが進み、最適解から乖離する可能性がある。第二に、LLM エージェントへの入力がすべて JSON フォーマットで固定されている。ControlAgent の入力 JSON フォーマットを Listing 1 に示す。したがって、LLM 本来の自然言語理解能力を活用した柔軟な制御設計支援や、HITL(Human-in-the-loop) への展開が制限される可能性がある。第三に、本手法は正答率のみに着

目しており、トークン数に関する定量的比較が実施されていない。トークンは、LLM が文章を処理する際の最小単位であり、LLM の使用料金は入力・出力の両方のトークン数に基づいて決定される。そのため、実装時におけるコスト面での評価が必要である。第四に、Python による単一の実装にとどまっており、より高度な制御手法への応用が限定的である。

Listing 1: Input JSON Format for ControlAgent

```

1 {
2   "num": [
3     225
4   ],
5   "den": [
6     1,
7     54.2
8   ],
9   "gain_margin_min": 2.62870,
10  "phase_margin_min": 56.54215,
11  "settling_time_min": 0.04778,
12  "settling_time_max": 3.64021,
13  "overshoot_max": 10.66329,
14  "steadystate_error_max": 0.0001,
15  "scenario": "other"
16 },

```

3 提案手法

ここでは、2.2 節で前述した ControlAgent の課題を踏まえ、LLM エージェントによりチューニングが行われる PID 制御系を提案する。

3.1 LLM エージェント PID 制御系

本研究で提案する LLM エージェント PID 制御系を図 1 に示す。 r は目標値、 e は偏差、 u は制御入力、 y は制御対象の出力である。図 1 における出力 y および目標値 r は、LLM Agent に入力される。

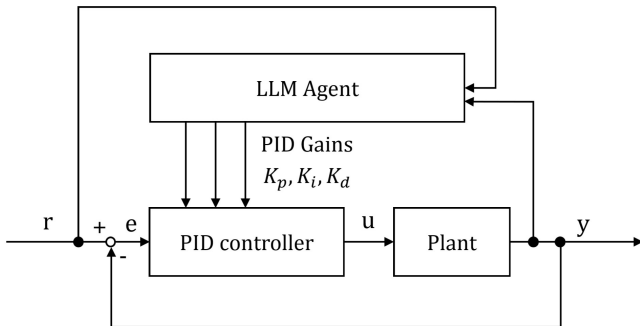


図 1: LLM Agent PID Control System

LLM エージェントの概略図を図 2 に示す。図 2 において、ユーザーによる入力はいずれも自然言語で行われる。自然言語で制御対象および制御目標（設計仕様）を与えると、“Agent: Translator”がこれを解析し、対応する MATLAB コードを自動生成する。そのコードは MATLAB の PID-tune を用いて、初期の PID ゲイン K_p , K_i , K_d を導出する。得られた初期ゲインは、“Agent: Computation, Feedback”に渡される。“Agent: Computation, Feedback”がこれを解析し、各制御性能を解析する MATLAB コードを自動生成し、MATLAB エンジン呼び出すことで各制御性能を計算する。

その後、各ゲインおよび算出された制御性能が“Agent: PID-tuner”へ渡される。“Agent: PID-tuner”は、渡された情報に基づき、新しいゲイン K_{p_new} , K_{i_new} , K_{d_new} を提案する。これらのゲインは、“Agent: Computation, Feedback”によって再び、各制御性能が解析される。このプロセスを反復することにより、パラメータのチューニングが行われる。そして、全ての設計仕様が満たされる、または、指定された繰り返し回数の上限に達すると反復が終了する。本システムの特徴は、自然言語処理と外部ツールである MATLAB を動的に連携できる点にある。設計者の専門的な知識に依存せず、効率的に PID 制御器のチューニングを行える構成となっている。

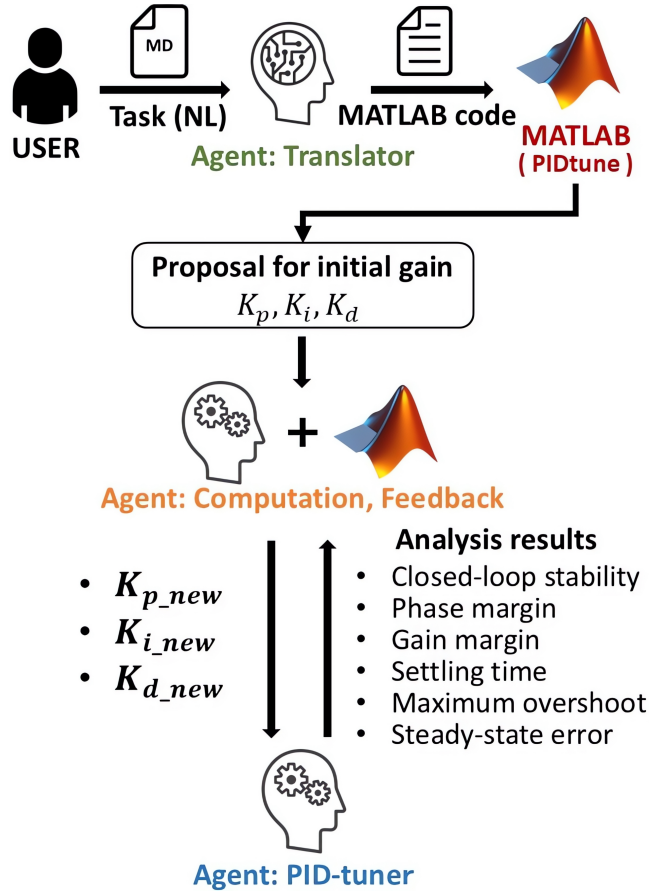


図 2: Proposed LLM Agent Workflow

4 提案手法の性能評価

4.1 評価手法

提案手法の性能評価および既存手法との比較について述べる．評価には制御設計評価用ベンチマークデータセットである ControlEval[5] を用いる．ControlEval は、一次および二次の安定・不安定系、時間遅れを伴う一次システム、高次の安定・不安定系の合計 10 種類の制御器設計のタスクからなる．各タイプごとに 50 種類のプラントと設計仕様が設定され、合計 500 の制御タスクから構成されている．各タスクの設計仕様には、閉ループ安定性・位相余裕・ゲイン余裕・整定時間・最大オーバーシュート量・定常偏差が含まれており、これらの指標に基づいて設計された制御器の性能が総合的に評価される．設計タスクの一例を図 3 に示す．

Design a controller for
$G(s) = \frac{10s^2 + 20s + 45}{s^5 + 40.8s^4 + 383.92s^3 + 2010s^2 + 6597s + 6750}$
Requirements:
1. Stability: The closed-loop system should be stable.
2. Phase margin: Larger than 67.94 degree
3. Settling time: Between 0.06 s and 8.24 s
4. Gain margin: Larger than 7.29 dB
5. Overshoot: Less than 3.29
6. Steady-state error: Less than 0.0001

図 3: An Example of a ControlEval Design Task

4.2 実験結果

従来手法・ControlAgent・提案手法に対して、ControlEval の高次システム（Higher-order System）を用いた評価実験を行った．実験結果を表 1 に示す．なお、ControlAgent に対して再現実験を行ったところ、当該論文記載の結果と大きく異なる結果が得られたため、再現実験結果を併せて記載している．実験に設定した LLM モデルおよびハイパーパラメータ、PIDtune のオプションは表 2 の通りである．本実験では、出力のランダム性を抑制するため、temperature=0 に設定している．また、本論文での評価指標である Accuracy Rate (ACC) は、全問題に対する正答数の割合である．ControlEval の計 6 個の設計仕様を全て満たす PID ゲインが得られた場合を正解とし、1 つでも設計仕様を満たさない場合を不正解とする．

表 1 より、提案手法は安定な高次システムにおいて、ControlAgent と比較して正答率が 28.0%向上している．

4.3 トークン数の比較

高次システムに対して、各手法における総トークン数および 1 タスクあたりの平均トークン数を比較した結果を表 3 に示す．表 3 より、提案手法は ControlAgent と比較してトークン数が約 56.0%減少したことが分かる．

表 3: Token Count Comparison for Higher-order System

Method	Total Tokens	Tokens / Task
ControlAgent	2,714,975	54,300
Proposed Method	1,195,853	23,917

また、表 1 の安定な高次システムにおいて、PIDtune だけでは設計仕様を満たすゲインを導出できなかった問題に対して、初期ゲインの提案に PIDtune を用いた場合と用いなかった場合の使用トークン数を比較する．比較結果を表 4 に示す．

表 4: Comparison of Token Counts with and without PIDtune

PIDtune	Tokens	Total Tasks	Tokens / Task
Enabled	117,983	15	9,832
Disabled	322,492	15	26,874

表 4 より、初期ゲイン提案に PIDtune を用いることで、トークン数は約 63.4%減少したことが分かる．

また、図 3 で示したタスクにおいて、提案手法を適用した際のステップ応答を図 4 に示す．図 4 において、青線が PIDtune によって導出された初期ゲイン、破線が “Agent:PIDtuner” によってチューニングされた各ゲイン、赤線が最終的に設計仕様を満たした PID ゲインでの応答波形である．計 9 回のイテレーションで、反復回数を重ねるごとにオーバーシュートが小さくなり、振動が減少していることが分かる．

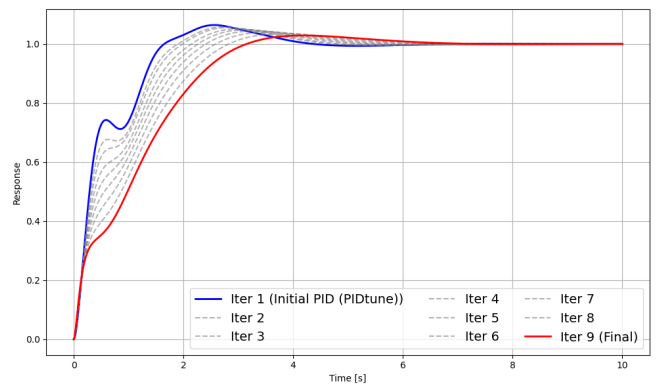


図 4: Step Response of Proposed Method

表 1: Accuracy Rate (ACC, %) of baseline and Proposed Methods on ControlEval

Method	Type	Higher-order (Stable)	Higher-order (Unstable)
LLM only[5]	Zero-shot	4.0	0.0
	Zero-shot CoT	16.0	0.0
	Few-shot	18.4	5.6
	Few-shot CoT	22.4	10.4
MATLAB only	PIDtune[5]	48.0	60.0
	PIDtune (re-experiment)	52.0	0.0
Conventional LLM-agent	ControlAgent[5]	72.0	61.6
	ControlAgent (re-experiment)	68.0	4.0
Proposed LLM-agent	Proposed Method	96.0	0.0

表 2: Configuration of Hyperparameters and PIDtune Options

	Hyperparameters / Options
GPT-4o	model = gpt-4o-2024-0806, temperature = 0, maxtokens = 1024
PIDtune	PhaseMargin: ‘Task Value’, DesignFocus: ‘reference-tracking’

5 考察

表 3,4 に示すように, 初期ゲイン推定に PIDtune を活用した場合, 使用トークン数は約 63.4%削減された. これは, LLM が初期段階から適切なゲイン候補を得ることで, 逐次的に再提案を行う必要が減少したためであると考えられる. そのため, 本手法は計算・通信コストの観点でも効率的であることが分かる. 外部ツールとの連携, すなわち LLM が得意とする自然言語理解と, 数値解析に特化した従来ツールの長所を統合することが, リソース効率の向上に寄与していると考えられる.

一方で, LLM エージェントの推論過程は依然としてブラックボックス的であり, 得られるゲインには一定のランダム性がある. また, ControlEval 上では良好な結果が得られたものの, 実際の物理システムや非線形・時変系への適用可能性についてはさらなる検証が必要である. さらに, 一定のトークン削減効果は確認されたものの, モデルや推論時間の最適化を含めた総合的な計算コスト評価が今後の課題である.

6 おわりに

本論文では, 制御設計用ベンチマークデータセット ControlEval を用いて, 制御器チューニングにおける LLM エージェント手法の提案と評価を行った. PIDtune 等の既存のオートチューナーを組み合わせることによって, 一定の正答率の向上および, 使用トークン数の減少が見られた. さらに, 提案手法は MATLAB との動的な連携が可能であるため, より高度な制御手法への応用が期待で

きる. 今後は, LLM を既存の制御手法と適切に組み合わせ, 短所を補完しながら制御システムの構築を目指すアプローチが重要となる.

参考文献

- [1] Itsuki Ogo and Masanobu Koga. Can ChatGPT pass modern control theory exam? In *ICCAS 2024, The 24th International Conference on Control, Automation and Systems*, Jeju, Korea, 2024.
- [2] L. Wang, C. Ma, X. Feng, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18:186345, 2024.
- [3] S. Hong et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*, August 2024.
- [4] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.
- [5] Xingang Guo, Darioush Keivan, Usman Syed, Lianhui Qin, Huan Zhang, Geir Dullerud, Peter Seiler, and Bin Hu. Controlagent: Automating control system design via novel integration of llm agents and domain expertise. *arXiv preprint arXiv:2410.19811*, 2024.