

# 制御コマンドの言語化によるロボットの言語指示操作

理学専攻・情報科学コース

2340658

久保山 瞳

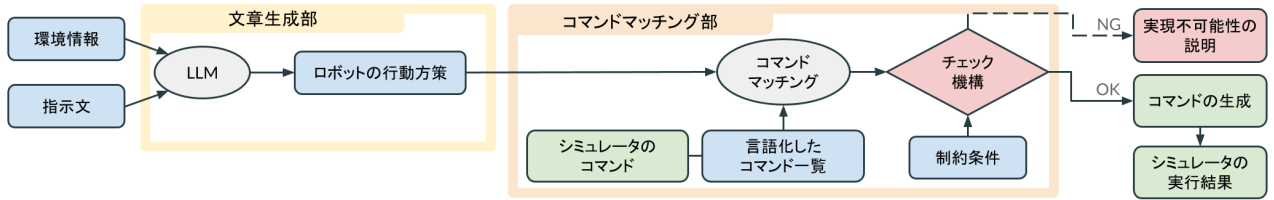


図 1: 提案手法概要

## 1 はじめに

大規模言語モデル (LLM) の発展により様々な分野においてその利用方法が検討されており、中でもロボティクス分野ではロボットの環境認識やプログラムの自動生成への応用が進んでいる。ロボットの制御には固有のソフトウェアやハードウェアを考慮する必要があり、同じタスクでも生成すべき制御プログラムが異なる。

本研究では、ロボットの動作単位を「コマンド」として抽象化し言葉で表現することによって、ロボットに依存しない汎用的な行動方策を大規模言語モデルに生成させる手法を提案する。さらに、Chain-of-Thought (CoT) を用いて行動方策を段階的に生成するプロセスを導入することで、複雑な条件にも対応可能な制御性能の向上を目指した。提案手法の有効性を検証するため、複数の異なるロボットアームを対象に実験を行い、生成された行動方策がロボット固有の制御コマンドとして変換・実行可能であることを示した。

## 2 ロボットの行動方策生成モデル

本研究における提案手法の概要を図 1 に示す。

提案モデルは行動方策生成を行う文章生成部と、対応するコマンドを決定するコマンドマッチング部、実現可能性から生成した文章及びコマンドの妥当性を判断するチェック機構によって構成される。

### 2.1 文章生成部

文章生成部では大規模言語モデルを用いて、ロボットの行動方策を生成する。大規模言語モデルに対して、Chain-of-Thought(CoT)[1] のようにロボットアームの状況を踏まえながら順序立てて動作を説明させることで、具体的なロボットの行動方策を生成する。大規模言語モデルにはテキストのみのプロンプトを与え、出力した文章に対してさらにプロンプトを追加して生成させることを 4 段階 (STEP1~4) で実行することで、最終的な行動方策文を生成する。

STEP1 では環境情報をもとにし、目的地の座標と行動方策のベースとなる動作を CoT のように順序立てた形で生成させる。動作説明の生成を助けるために one-shot で「持ち上げる」という動作についての説明の例を与えている。STEP2, 3 では移動を分割しその移動方向について明確にさせ、STEP4 で明瞭な説明と座標の組み合わせとなるように文の変形を行う。最終的に生成された行動方策文は、大規模言語モデルの考

えるロボットアームの動作説明であり、これはロボットの動作を表す言葉の抽象度と同程度の文のみで構成されるものと仮定する。実際に与えたプロンプトを以下に示す。

#### STEP1

あなたは人を手助けする優秀なロボットアームです。受け取った指示と環境情報に基づいて指示を解釈し、自分がどのように行動するのかを順序立てて最後まで具体的に説明することができます。例えばものを持つという動作は持ち上げる動作と言い換えられるので、アームをコップに近づけハンドを閉じてアームを上上げます。移動先の座標については机の大きさや人の位置を鑑みて自分で考えてみてください。

環境情報：この環境にはロボットアームが 1 台と机とコップがあります。座標系は右手系であり、ロボットアームの基準点と机の中心は XY 平面の原点 (0, 0) です。コップは机の上においてあり、コップの底面の位置は (0.5, 0, 0.7)、高さは 0.1m です。指示を与える人は机の X, Y 正方向に立っています。机の上面は XY 平面で、Z 軸が上向きです。机の上面の中心座標は (0, 0, 0.65) です。机のサイズは X 方向に 1.5m, Y 方向に 1.0m, Z 方向に 0.05m であり、机の天板の四隅の座標は (-0.75, -0.5, 0.65), (-0.75, 0.5, 0.65), (0.75, -0.5, 0.65), (0.75, 0.5, 0.65) です。アーム先の初期位置は (0.5, 0.0, 1.0) で、グリッパーの長さは 0.2m です。ロボットアームの制御はアームの先端に注目するものであって、グリッパーの先端を制御できるわけではありません。またロボットアームのグリッパーは最初閉じています。

#### STEP2

元の意味を保持したまま、動作を明瞭にしてください。この時アームを移動させる場合には移動する場合は必ず X, Y 軸のどちらかに平行になるように移動するように分割してください。

#### STEP3

元の意味を保持したまま方向の説明を加えてください。アームを移動させる場合の説明は、今の位置から見て前後左右どの方向に動かすかについて教えてください。環境情報の通り右手系のため、X 軸正方向に動かす時は前に、Y 軸正方向に動かす時は左に動くことになるので向きを間違えないでください。また表示するのは移動先の座標のみです。

#### STEP4

元の意味を保持したまま、各説明は「アームを持ち上げる (x,y,z).」「アームを後ろに動かす (x,y,z).」「ハンドを開く。」のように、説明文と移動先の座標のみで構成してください。この時動かす方向に注意してください。

行動方策の文生成には OpenAI の AssistantsAPI を介して、GPT-4o(gpt-4o-2024-08-06) を用いた。なお生成される文章の正確性を上げるため、環境情報やロボットの仕様を記述したドキュメントを用意し、そこから情報検索を行う RAG[2] の手法を用いた。参照するドキュメントは STEP1 で与えた環境情報の説明文、物理的な制約条件、各アームの仕様、座標情報を箇条書きで列挙したテキストファイル計 5 つおよび、各アームの可動範囲・軸数などの仕様が載ったテキストと画像で構成される公式のデータシートである。

表 1: コマンドマッチングによる生成の流れ

STEP1 で生成した行動方策文	STEP4 で生成した最終的な行動方策文	類似度	ロボットのコマンド
1. アームの下降: 指示された内容通りに、アームをコップの上に移動させるため、Z 軸方向に下げます。安全にコップを掴むため、コップの底面上方 0.15m、座標 (0.5, 0, 0.85) に移動します。	アームを持ち下げる (0.5, 0.0, 0.85).	0.965	アームを下ろす
2. グリッパーの開放: グリッパーを開きます。これによりコップを掴む準備が整います。	ハンドを開く.	1.000	ハンドを開く
3. アームの下降とコップの把持: グリッパーでコップを掴むため、アーム先端をさらに下降し、コップの上端付近 (0.5, 0, 0.75) まで移動します。	アームを持ち下げる (0.5, 0.0, 0.75).	0.965	アームを下ろす
4. グリッパーの閉鎖: グリッパーを閉じてコップをしっかりと把持します。	ハンドを閉じる.	1.000	ハンドを閉じる
5. アームの上昇: コップを持ち上げるために、アームを Z 軸方向に上昇させ、元の Z=0.85 まで戻します。	アームを持ち上げる (0.5, 0.0, 0.85).	0.980	アームを上げる
6. コップを目的地に移動: 人の指示は机の正面方向への移動だったので、コップを寄せるために少し X 方向と Y 方向のプラス方向に移動することとします。安全に置くために、座標 (0.6, 0.2, 0.85) までアームを移動します。	アームを前に動かす (0.6, 0.0, 0.85). アームを左に動かす (0.6, 0.2, 0.85).	1.000 1.000	アームを前に動かす アームを左に動かす
7. アームの下降: コップを机の上に置くため、アームを下降し、コップが机に触れる位置 Z=0.75 まで移動します。	アームを持ち下げる (0.6, 0.2, 0.75).	0.965	アームを下ろす
8. グリッパーの開放: コップをリリースするためにグリッパーを開きます。	ハンドを開く.	1.000	ハンドを開く
9. アームの上昇: 掴み動作を完了してアームが邪魔にならないよう、アームを Z 軸方向に元の高さ Z=1.0 まで戻します。	アームを持ち上げる (0.6, 0.2, 1.0).	0.980	アームを上げる

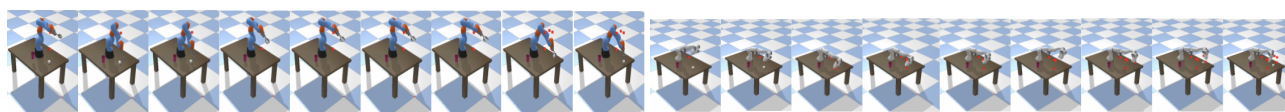


図 2: 2つのロボットアームのシミュレータの実行結果の例

## 2.2 コマンドマッチング部

与えられた指示とロボットの制御コマンドとを言葉の意味をもとに結びつけるため、まずは以下の通りロボットの動作を言葉で表すためのコマンド群を用意した。

ハンドを閉じる	ハンドを開く	周囲を確認する
アームを上げる	アームを下ろす	ハンドを回転させる
アームを右に動かす	アームを左に動かす	アームを前に動かす
アームを後ろに動かす	行動を終了する	

これらのコマンド群は、シミュレータ上では言葉通りに実行する関数（ロボットの制御コマンド）として実装されており、対応するコマンドが呼び出されることによって、各動作が実行される。

提案手法において自然言語文は埋め込みベクトルで表現され、CoTによってロボットの行動計画として生成された文ベクトルと用意したロボットの制御コマンド文との  $\cos$  類似度をとることによってコマンドのマッチングを行う。ベクトルを用いることで、文自体がどれだけ一致しているかという表面的な評価にとどまらず、意味を考慮した評価を行うことができる。

文ベクトルの作成には Multilingual-E5 [3] をベースとして、Sentence-Transformer [4] を用いる。

## 3 実験

### 3.1 実験設定

「コップをこちに寄せて」という指示をもとに、生成した行動方策文 20 文に対して提案手法を用いてコマンドを生成し、シミュレータで動作を確認した。

ロボットアームとしては KUKA 社の 7 軸協働ロボットである LBR iiwa 7 R800 および、Franka Emika 社の 7 軸協働ロボット Panda アームを使用する。シミュレータには PyBullet を使い、自作した環境下で生成したコマンドを実行した。

### 3.2 結果

ここでは KUKA 社の iiwa アームについての実行結果を示す。提案手法を用いて、表 1 のように行動方策を生成しコマンドに変換、実行した。座標を除く動作自体に注目すると 90%(18/20) で正しい動作が生成され、4 段階に分けて文を整形することによって、座標情報

を保持したまま文の抽象度を上げ、ロボットの制御コマンドを結びつけることができた。特に方向について常に注意させることで汎用的なロボットアームの操作手順をなぞり、想定通りの動作につながった。しかし目的地の座標について机の大きさをきちんと考慮できている完全な生成は 35%(7/20) に留まり、40%(8/20) は y 座標を机の端に設定したことでコップを落とす例が見受けられたことから、座標生成の制御については不完全であり、さらなる検討が必要である。テキスト情報のみで正確な行動方策を生成するためには、座標情報をもとに方向を介して、環境の認識を持たせ続けることが重要である。

なお、iiwa アーム及び Panda アームにおいて成功した場合のシミュレータの動作を図 2 に示す。2つの異なるロボットアームに対しても同様に操作ができた。

## 4 おわりに

ロボットの動作を言葉で表現することにより汎用的な行動方策を生成する手法をもとに、2つのロボットアームに対して操作を行った。環境情報を踏まえたロボットの動作を確認し、大規模言語モデルを利用し行動方策を段階的に生成することによってロボットの制御コマンドへの変換・および実行可能であることを示した。今後は動作のみならず妥当な目的地座標生成精度を向上させるほか、より複雑な環境やタスクに対する動作に対して同様の手法での実現を目指す。

## 参考文献

- [1] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.
- [2] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.
- [3] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report, 2024.
- [4] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.