LLM 기반 자연어 명령 해석을 통한 모델예측제어 및 안전제약조건 생성

송상헌 [1], 강동엽 [2], 동지연 [2], 박찬은 [1]*
경북대학교 전자전기공학부 [1], 한국전자통신연구원 [2]

sghonsong@knu.ac.kr, kang@etri.re.kr, jydong@etri.re.kr, chaneun@knu.ac.kr

Model Predictive Control and Safety Constraint Generation Through LLM-Based Natural Language Command Interpretation

Sanghon Song[1], Dongyeop Kang[2], Jiyeon Dong[2], Chan-eun Park[1],
Kyungpook National University, Electronics[1], Telecommunication Research Institute[2]

요 약

본 논문은 사용자의 자연어 명령 속 안전성 맥락을 고려하여, 모델예측제어(Model Predictive Control, MPC)를 구성하는 목적함수 및 안전 제약조건을 LLM(Large Language Model)을 통해 자동 생성하는 프레임워크를 제안한다. 사용자의 언어적 표현에 담긴 안전성 수준에 대응하는 제약 조건의 안전 마진을 설정함으로써, 로봇이 상황에 따라 유연하게 안전 거리를 확보할 수 있도록 한다. SafePandaGym 시뮬레이션 환경에서 수행되었으며, 제안된 방법론이 자연어 기반의 안전성 해석과 실시간 제어 정책 반영에 효과적임을 입증하였다

Ⅰ. 서론

기존 산업용 로봇은 반복적인 low-level 작업에는 정밀하지만, 작업 변경 시마다 전문가의 프로그래밍이 필요해 유연성이 떨어진다. 최근 협동로봇의 등장으로 비전문가도 직관적으로 조작할 수 있는 인터페이스가 요구되며, 자연어 명령은 이를 가능하게 한다. 이에 따라, LLM(Large Language Model)을 활용해 자연어 지시를 low-level 제어 명령으로 변환하는 연구가 활발히 진행되고 있다 [1], [2].

모델 예측 제어(Model Predictive Control, MPC)는 목적함수와 제약조건을 명시적으로 다룰 수 있어 복잡한 작업을 안전하게 수행하는 데 유리한 제어 프레임워크로 널리 사용된다. 특히, 협동로봇과 같이 인간과 물리적으로 상호작용하는 환경에서는 단순한 작업 수행의 최적화뿐만 아니라, 장애물 회피나 충돌 방지와 같은 안전성 확보가 필수적이다. 더 나아가, 사용자가 자연어 명령을 통해 로봇의 안전성 수준까지 직관적으로 제시할 수 있다면, 로봇 시스템은 인간의 의도에 보다 유연하게 대응할 수 있을 것이다. 이러한 맥락에서 [1]에서는 LLM 을 활용해 자연어로부터 MPC 제어기를 위한 목적함수와 제약조건의 생성부터 로봇 시스템의 low-level 제어까지 아우르는 통합 프레임워크를 제안하였다. 하지만 이 프레임워크는 장애물 회피를 위한 안전 마진(safety margin)을 고정된 값으로 사용하며, 움직이는 장애물이 존재하는 동적인 환경에서의 장애물 회피를 검증하지 않는다.

본 논문에서는 이러한 [1]의 한계를 보완하기 위해, 사용자의 자연어 명령에 내포된 안전성 관련 맥락의 강도에 따라 적절한 안전 제약조건을 자동으로 생성하는 방법을 제안한다. 제안하는 방법은 LLM 을 활용하여 사용자의 자연어 명령을 해석하고, 해당 작업을 위한 목적함수와 함께 안전 마진을 고려하여 안전제약조건을 생성한다.

Ⅱ. 본론

본 논문에서는 OpenAI 의 사전 학습 언어모델 GPT-4o [3]를 활용하여, 사용자로부터 입력된 자연어 명령을 해석하고, MPC 제어기의 목적함수 및 안전 제약조건을 자동으로 생성한다. 실험은 SafePandaGym [4] 환경에서 수행되었으며, 이 시뮬레이션은 다양한 장애물 조건과 협동작업 시나리오를 반영할 수 있어 적합하다. 제어 대상은 3 차원 작업공간에서 이동하는 로봇 매니퓰레이터(manipulator)이며, 상태는 그리퍼(gripper)의 3 차원 위치 벡터 $\boldsymbol{x} = [x, y, z]^T$ 로 정의되고, 제어 입력은 그리퍼의 속도 벡터로 구성된다. 제어기는 예측 지평선(prediction horizon) $T = 15$ 를 기준으로 동작한다. LLM 이 생성하는 목적함수는 타겟 물체와 그리퍼 사이의 거리를 최소화하는 이차 형식(quadratic form) 형태로 구성되며, 다음과 같은 심볼릭(symbolic) 표현으로 나타난다:

$$\mathrm{J} = \sum_{t=0}^{T-1} \left\| \boldsymbol{x}(t) - \boldsymbol{x}_{target}(t) \right\|_2^2 \tag{1}$$

안전 제약조건은 장애물과 그리퍼 간의 최소 거리, 즉, 안전 마진을 보장하는 유클리디안 거리 기반 부등식으로 정의된다. 각 장애물의 위치 $\boldsymbol{o}_i$ 에 대해, 그리퍼의 위치 $\boldsymbol{x}$는 다음의 조건을 만족해야 한다:

$$\|\boldsymbol{x}(t) - \boldsymbol{o}_i(t)\|_2^2 \geq m_i \tag{2}$$

여기서 m_i는 LLM 이 자연어 명령 속의 안전성 표현을 기반으로 설정한 안전 마진이다. 마진 값 m_i 은 실험적으로 결정된 범위인 $m_i \in [0.025, 0.05]$ m 사이에서 설정되며, "살짝 떨어져", "절대 가까이 가지 마" 등 다양한 자연어 표현을 학습한 in-context few-shot 예시를 기반으로 LLM 이 적절한 마진을 선택한다.

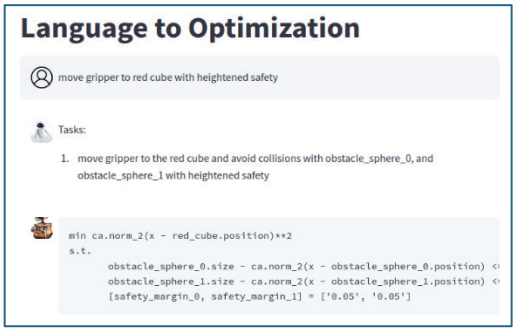

그림 1 LLM 과의 상호작용을 위한 GUI

제어기의 구성은 사용자가 GUI 를 통해 자연어 명령을 사용자 프롬프트(user prompt)로서 입력하면서 시작된다. LLM 에는 사용자 프롬프트와 함께 시스템 프롬프트(system prompt)가 사용된다. 시스템 프롬프트([그림 3] 을 보라.)에는 현재 환경의 구성 정보(예: 작업에 포함된 물체 리스트, 물체 종류, 목적 설명 등)와 더불어 사용자 명령을 해석하는 데 필요한 문법 규칙, 가능한 행동 설명 템플릿, 안전 마진 설정 방법 등이 포함된다. 이 때 LLM 은 환경 내의 객체의 위치나 거리 정보는 직접적으로 알지 못하며, 대신 시뮬레이션 환경의 제어 모듈에서 이를 실시간으로 계산하고 적용한다. [그림 1]은 GUI 화면을 나타낸다. 예를 들어, "move gripper to red cube with heightened safety."라는 명령이 주어졌을 때, LLM 은 red cube 로의 이동을 목표로 하는 목적함수와 함께, 주변 장애물과의 충돌 회피를 위한 거리 제약, 이 거리 제약에 적용될 안전 마진을 포함하는 MPC 구성 명세를 출력한다. 이 출력은 API 를 통해 외부 제어 모듈로 전달되고, 해당 모듈에서 수식을 구문 분석한 후, 실시간으로 MPC 에 반영되어 제어 루프가 실행된다.

Ⅲ. 실험 결과

제안한 시스템의 성능을 평가하기 위해, 다양한 수준의 안전성 맥락을 포함하는 명령어를 사용하여 궤적의 변화를 비교하였다. [그림 2]는 "적당히 장애물과 떨어져"와 "절대로 부딪히지 마"라는 명령어에 따라 생성된 로봇 궤적을 보여준다. 전자의 경우 안전 마진이 작게 설정되어 장애물 근처를 통과하지만, 후자의 경우 큰 마진이 적용되어 더 멀리 우회하는 경로가 생성된다. 이를 통해 제안하는 프레임워크가 사용자의 명령 속 안전성 의도를 반영하여 제어 정책을 조정함을 확인할 수 있었다.

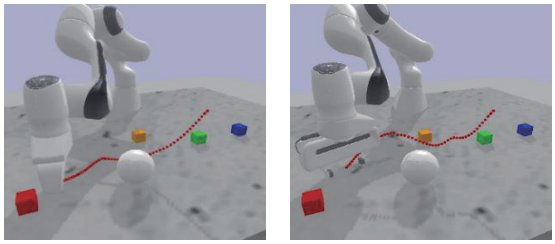

그림 2 안전성 맥락의 수준에 따른 궤적 비교

```
You are a helpful assistant tasked with generating optimization formulations for configuring an MPC controller for robotic manipulation tasks.
I will give you a command, and you will return the objective function and, if necessary, the constraint functions to be applied in the MPC controller.

Scene description is as follows:
  (1) The CasADi library is used to implement the MPC.
  (2) The variable `x` represents the gripper's position in 3D space, i.e., (x, y, z).
…
General rules:
  (2) All inequality constraints must be written to be satisfied when less than or equal to zero:
    (a) For example, to express "ca.norm_2(x) >= 1", write it as "1 - ca.norm_2(x)".
…
Safety rules (Interpret safety-related expressions flexibly and appropriately):
  (1) If the command contains any language that implies a safety concern, you must generate an optimization formulation that avoids obstacles.
  (2) Determine an appropriate safety margin based on the level of safety concern expressed in the command. The safety margin must be in the range [0.025, 0.05].
…
You must return your output in valid JSON format. Here are a few examples:

objects = ['object_1', 'object_2', 'obstacle_sphere_0', 'obstacle_sphere_1']
# Command: move gripper to object_1 with moderate safety
{
  "objective": "ca.norm_2(x - object_1.position)**2",
  "equality_constraints": [],
  "inequality_constraints": [
    "obstacle_sphere_0.size - ca.norm_2(x - obstacle_sphere_0.position)",
    "obstacle_sphere_1.size - ca.norm_2(x - obstacle_sphere_1.position)"
  ],
  "input_constraints": [],
  "safety_margin": ["0.025", "0.025"]
}
```

그림 3 제어 명세 생성을 위한 시스템 프롬프트

Ⅳ. 결론

본 논문은 자연어 명령에 내포된 안전성 맥락을 반영하여, MPC 제어기 구성을 위한 목적함수, 안전 제약조건, 안전 마진을 자동 생성하는 LLM 기반 로봇 제어 프레임워크를 제안하였다. 실험 결과, 사용자 표현에 따라 유연하게 제약조건의 안전 마진을 조정할 수 있음을 확인하였으며, 이는 인간-로봇 상호작용의 직관성과 안전성을 동시에 향상시키는 데 기여할 수 있다.

ACKNOWLEDGMENT

이 논문(또는 저서, 특허)은(는) 정부(과학기술정보통신부)의 재원으로 과학기술사업화진흥원의 지원을 받아 수행된 연구임('학연협력플랫폼구축 시범사업' RS-2023-00304776, RS-2023-00304695).

참 고 문 헌

[1] S. Ismail, A. Arbues, R. Cotterell, R. Zurbrügg, and C. A. Alonso, "NARRATE: Versatile Language Architecture for Optimal Control in Robotics," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2024, pp. 9628–9635.

[2] W. Yu *et al.*, "Language to Rewards for Robotic Skill Synthesis," in *Proc. 7th Conf. Robot Learning (CoRL)*, PMLR, 2023, pp. 374–404.

[3] OpenAI *et al.*, "GPT-4o System Card," *arXiv preprint* arXiv:2410.21276, 2024.

[4] T. Oseni and S. Wang, "Safe Panda Gym," GitHub repository, 2022. [Online].