제어 장벽 함수를 이용한 샘플링 기반 안전 필터 연구

박준영, 성현태, 안희진*
한국과학기술원

{junyoung766, hyeontae.sung, heejin.ahn}@kaist.ac.kr

Sampling-based Safety Filtering via Control Barrier Functions

Junyoung Park, Hyeontae Sung, Heejin Ahn*
Korea Advanced Institute of Science and Technology

요 약

본 논문에서는 제어 시스템의 안전을 보장하는 샘플링 기반 안전 필터 알고리즘을 제안하였다. 동역학을 알고 있는 이산 시간 시스템에서 현재 제어 입력을 기반으로 미래의 제어 입력 시퀀스를 샘플링하고, 각 샘플에 대해 롤아웃을 수행하여 안전성을 판단한다. 모든 롤아웃이 안전하지 않다고 판단되면, 안전 필터가 개입하여 주어진 제어 입력과의 차이가 작으면서도 안전한 제어 입력으로 대체한다. 안전성 판단에는 Control Barrier Function(CBF)을 활용한 제약 조건이 사용된다. 시뮬레이션을 통해 제시된 안전 필터가 로봇에 자율성을 부여하면서도 안전을 보장하는 것을 검증하였다.

Ⅰ. 서 론

자율주행과 같이 안전이 중요한 제어 시스템에서는 안전을 보장할 수 있는 제어 입력을 생성해야만 한다. 최근 연구되는 데이터 기반 제어기는 다양한 환경에서 높은 성능을 보이지만 안전을 보장하지 못하는 한계가 있다.[1] 따라서 이런 제어기의 안전을 보장하기 위한 장치가 필요하다. 기존의 성능 향상을 위한 제어기를 nominal controller 라고 할 때, 안전 필터는 nominal controller 의 제어 입력을 따르다가 필요한 순간에만 개입하여 안전을 보장하는 역할을 한다. 이러한 방법론은 안전한 상태를 유지하면서도 nominal controller 의 자율성을 최대한 보장한다.

본 연구에서 제안하는 샘플링 기반 안전 필터에서는 미래 제어 입력 시퀀스를 샘플링하여 현재 주어진 제어 입력의 안전성을 판단한다. 이와 같은 샘플링 기반 접근법은 기존의 최적화 기반 방식[2]과 비교했을 때 최적해를 구하기 어려운 복잡한 동역학 모델이나 비선형 시스템에서도 사용 가능성이 높다는 강점이 있다. 안전 필터는 필요한 경우 이전 샘플링 단계에서 저장한 안전 샘플들을 새로운 제어 입력에 활용한다.

논문에서는 샘플링 기반 안전 필터 알고리즘을 제시하고, 제시된 알고리즘이 제어 입력 제약이 있는 상황에서 로봇에 자율성을 부여하면서도 주어진 안전 조건을 만족하는 것을 시뮬레이션으로 확인하였다.

Ⅱ. 본론

본론의 2.1 절에서는 제어 입력의 안전성을 판단할 때 사용되는 CBF 의 개념에 대해 설명하였다. 2.2 절에는 본 논문에서 제안하는 샘플링 기반 알고리즘을 정리하고, 2.3 절에 실험을 통한 검증 결과를 나타냈다.

2.1 Discrete-Time Control Barrier Function[2]

다음과 같은 이산 시간 시스템을 생각하자.

$$x_{t+1} = f(x_t, u_t) \tag{1}$$

여기서 $x_t \in \mathcal{X}$ 는 시스템의 상태, $u_t \in \mathcal{U}$ 는 제어 입력을 의미한다. 안전 보장 제어를 위해 다음과 같은 안전 집합 S를 정의할 수 있다.

$$S \coloneqq \{x_t \in \mathcal{X} : B(x_t) \geq 0\} \tag{2}$$

이때 함수 $B: \mathcal{X} \to \mathbb{R}$가 모든 시스템 상태 $x_t \in \mathcal{X}$에 대해 다음 조건을 만족하면 discrete-time CBF(DCBF)라고 한다.

$$\exists u_t \in \mathcal{U}\ s.t.\ B\big(f(x_t, u_t)\big) \geq (1-\gamma)B(x_t),\ 0 < \gamma \leq 1 \tag{3}$$

DCBF B에 대해 부등식 (3)을 만족하는 제어 입력 u_t를 사용하면 안전 집합 S 내부의 상태를 유지할 수 있다.

2.2 샘플링 기반 안전 필터

본 논문에서 제안하는 샘플링 기반 안전 필터는 Algorithm 1 에 정리하였다. 알고리즘은 매 타임스텝마다 nominal controller 의 제어 입력 u_{des}를 사용하여 다음 스텝의 상태 x_{t+1}을 예측한다(line 1). 이후 T horizon 을 가지는 제어 입력 시퀀스 $U = [u_0, \dots, u_{T-1}]$를 샘플링하고, x_{t+1} 부터 롤아웃을 진행하여 N 개의 trajectory X 를 얻는다(line 3-5). CBF 제약 조건을 통해 trajectory 들의 안전성을 평가하는 $cost_{CBF}$를 계산한다(line 6).

$$cost_{CBF}(X) = \sum_{k=0}^{T-1} \max\{0, -B(X_{k+1}) + (1-\gamma)B(X_k)\} \tag{4}$$

여기서 X_k 는 trajectory X 의 k 번째 상태를 의미한다. 이때 안전한 제어 입력 시퀀스는 다음 타임스텝에 사용할 수 있도록 S_{t+1}^{safe}에 저장한다(line 7-8).

안전한 샘플이 하나라도 존재하면 안전 필터의 개입 없이 u_{des}를 사용한다(line 9-10). 안전한 샘플이 하나도 없는 경우에는 이전 타임스텝에서 저장한 안전한 제어 입력 시퀀스 집합 S_t^{safe} 에서 u_{des} 와 가장 가까운 샘플 U^{safe}를 찾는다(line 12). 이때 $cost_{deviation}$을 활용한다.

$$cost_{deviation}(U) = \|U_0 - u_{des}\|^2 \tag{5}$$

여기서 U_0는 시퀀스의 첫 번째 제어 입력을 의미한다.

이후 U^{safe} 의 첫 번째 제어 입력은 u_{safe}로 내보내고, 뒷부분은 다음 타임스텝에서 사용할 수 있도록 S_{t+1}^{safe} 로 넘겨준다(line 13).

Algorithm 1 Sampling-based Safety Filter

Given: System dynamics $f(\cdot)$, number of samples N, sampling horizon T

Input: Current state x_t, desired control input u_{des}, set of safe control sequences $S_t^{safe} \neq \emptyset$

Output: Safe control input u_{safe}, set of safe control sequences S_{t+1}^{safe}

$x_{t+1} \leftarrow f(x_t, u_{des})$
$S_{t+1}^{safe} \leftarrow [\,]$
for $i \leftarrow 0$ *to* $N-1$ *in parallel* **do**
  Sample $U^i \leftarrow [u_0^i, \dots, u_{T-1}^i]$
  $X^i \leftarrow rollout(x_{t+1}, f, U^i)$
  $cost_i \leftarrow cost_{CBF}(X^i)$
  if $cost_i == 0$ **then**
    $S_{t+1}^{safe}.append(U^i)$
if $S_{t+1}^{safe} \neq \emptyset$ **then**
  $u_{safe} \leftarrow u_{des}$
else
  $U^{safe} \leftarrow \arg\min_{U \in S_t^{safe}} (cost_{deviation}(U))$
  $u_{safe} \leftarrow U_0^{safe}$, $S_{t+1}^{safe} \leftarrow [U_{1:T}^{safe}]$
return u_{safe}, S_{t+1}^{safe}

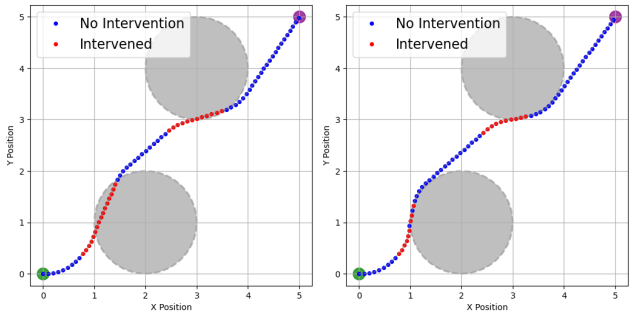
2.3 실험

시뮬레이션을 통해 알고리즘을 검증하였다. 시스템의 동역학으로는 Dubins car 모델을 사용하였다.

$$x_{t+1} = x_t + v\cos(\theta_t) \cdot \Delta t \quad (6)$$
$$y_{t+1} = y_t + v\sin(\theta_t) \cdot \Delta t \quad (7)$$
$$\theta_{t+1} = \theta_t + \omega_t \cdot \Delta t \quad (8)$$

상태 변수 x, y는 로봇의 위치 좌표, θ는 로봇의 heading angle 을 의미하고, v는 고정된 속력이다. 제어 입력은 각속도인 ω이고, 범위는 $\omega \in [-1.2, 1.2]\,rad/s$로 하였다. 타임스텝 간격 Δt는 $0.1s$로 설정하였다. 로봇의 목표는 $(0,0)$ 지점에서 출발하여 $(5,5)$에 도달하는 것이고, 안전 조건은 그림 1 의 회색 영역을 피하는 것으로 하였다.

안전 필터 적용에 필요한 CBF 로는 DeepReach[3]로 학습한 value function 을 사용하였다. 학습된 value function 은 미래 시점의 상태를 고려하므로, 장애물과의 거리만을 고려하여 안전성을 판단할 때에 비해 샘플링 horizon 이 짧을 때에도 더 나은 안전 보장 효과를 보인다. 그림 1 에서 CBF 학습 효과를 검증하였다. 필터가 개입하지 않은 시점은 파란색, 개입한 시점은 빨간색으로 궤적을 표시하였다.

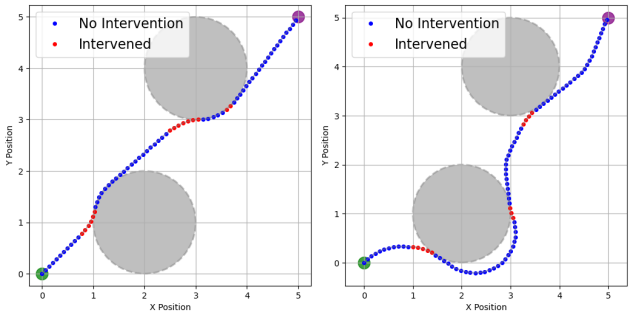


(a) (b)

그림 1. CBF 학습 효과 검증. 샘플링 horizon $\boldsymbol{T = 6}$일 때 (a)에는 장애물과의 거리만을 고려하고, (b)에는 학습된 CBF 를 사용하여 $cost_{CBF}$를 계산한 안전 필터의 적용 결과를 비교했다. 학습된 CBF 가 짧은 horizon 에서도 안전한 샘플을 잘 찾는 것을 확인하였다.

그림 2 에서는 서로 다른 nominal controller 에서 안전 필터의 효과를 검증하였다. 그림 2(a)에서는 최적화 기반 Model Predictive Control(MPC)을 통해 안전 조건과 관계없이 목적지 도달만을 고려하는 제어기를 nominal controller 로 사용하였다. 그림 2(b)에서는 실험자가 직접 조작하는 manual control 을 nominal controller 로 사용하였다.

그림 2(a)의 nominal controller 는 안전성을 고려하지 않는 제어기를 사용하기 때문에 원래는 회색 영역을 가로질러 가야 하지만, 안전 필터를 적용한 후에는 위험 상황마다 필터가 개입하여 로봇이 안전한 궤적을 따르게 된다. 그림 2(b)에서는 실험자가 임의로 제어 입력을 넣어 주는 상황에서도 안전 필터가 잘 동작하는 것을 보인다. 안전한 상황에서는 주어진 명령을 그대로 따르고, 위험 영역에 가까워지거나 의도적으로 진입하려고 할 때에는 안전 필터가 개입하여 안전한 상태가 유지된다.



(a) (b)

그림 2. Nominal controller 에 따른 실험 결과. (a)에는 목적지 도달만을 고려하는 최적화 기반 제어기, (b)에는 manual control 에 안전 필터를 적용한 결과를 나타냈다.

Ⅲ. 결론

본 논문에서는 샘플링을 통해 미래 궤적의 안전성을 판단하고 필요한 경우에만 제어 입력에 개입하여 안전을 보장하는 샘플링 기반 안전 필터 알고리즘을 제안하였다. 모델의 동역학을 알고 있는 상황에서 샘플링과 CBF 제약 조건을 활용하는 알고리즘을 고안하고 Dubins car 모델을 따르는 시뮬레이션 환경에서 검증하였다. 실험을 통해 알고리즘이 주어진 임의의 nominal controller 에 적절히 개입하여 안전을 유지하는 것을 확인하였다.

ACKNOWLEDGMENT

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학 ICT 연구센터사업의 연구결과로 수행되었음 (IITP-2025-RS-2023-00259991)

참 고 문 헌

[1] Brunke, Lukas, et al. "Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning," Annual Review of Control, Robotics, and Autonomous Systems 5.1 (2022), pp. 411-444.

[2] Zeng, Jun, Bike Zhang, and Koushil Sreenath. "Safety-Critical Model Predictive Control with Discrete-Time Control Barrier Function," 2021 American Control Conference (ACC), pp. 3882-3889.

[3] Bansal, Somil, and Claire J. Tomlin. "DeepReach: A Deep Learning Approach to High-Dimensional Reachability," 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 1817-1824.