DDP with CBF Constraints

Wonyoung Park (University of Seoul)
Date: January 26, 2023

I. ABSTRACTION

본 문서는 Differential Dynamic Programming(이하 DDP)를 사용하여 자동차의 path planning 진행 사항에 대해 기록하고 있다. DDP를 이용하여 경로를 만들되, Control Barrier Function(이하 CBF)를 이용하여 constraint를 구성하고자 한다.

II. 사전 지식

A. Differential Dynamic Programming

DDP에 대해서 다룰 때, discrete-time system을 고려한다. 따라서 모델의 kinematics는 식(1)와 같이 함수 f와 g로 구성된다.

$$x_{i+1} = f(x_i) + g(x_i)u_i \tag{1}$$

DDP의 total cost는

$$J(x_0, U) = \sum_{i=0}^{N-1} \ell(x_i, u_i) + \ell_f(x_N) \tag{2}$$

식(2)와 같다. 여기서 $U \triangleq \{u_0, u_1, u_2 ..., u_{N-1}\}$이고, N은 총 길이, ell_f는 final cost를 의미한다. 이 상황에서 최소화를 하고자 하는 것은 $U^\star \triangleq \arg\min_U J(x_0, U)$이다. i번째 경우일 때 Value는 $V_i(x) \triangleq \min_{U_i} J_i(x, U_i)$이다. 따라서 (3)를 만족하게 된다.

$$V(x) = \min_u [\ell(x, u) + V'(f(x) + g(x)u)] \tag{3}$$

DDP는 iteration을 여러번 진행하며 (3)를 backward pass에서 최소화하고, forward pass를 진행한다.

1) Backward Pass: Q-function이 δx,δu만큼의 변화를 가지면 $Q(x+\delta x, u+\delta u) = Q(x,u) + \hat{Q}(\delta x, \delta u) + \epsilon$와 같다.

$$\begin{aligned} Q(\delta x, \delta u) = Q_x \delta x + Q_u \delta u + \frac{1}{2}\delta x^\top Q_{xx} \delta x \\ + \delta x^\top Q_{xu} \delta u + \frac{1}{2} \delta u^\top Q_{uu} \delta u \end{aligned} \tag{4}$$

여기서 Q는 δ의 변화에 따른 second-order approximation 다.

$$\begin{aligned} Q(\delta x, \delta u) = \ell(x + \delta x, u + \delta u) \\ + V'(f(x+\delta x) + g(x+\delta x)(u + \delta u)) \end{aligned} \tag{5}$$

(3)를 Taylor expansion하면 (5)과 같다. Q-function의 pseudo-Hamiltonian을 구해보면 다음과 같다.

$$\begin{aligned} Q_x &= \ell_x + f_x^\top V_x' \\ Q_u &= \ell_u + f_u^\top V_x' \\ Q_{xx} &= \ell_{xx} + f_x^\top V_{xx}' f_x + V_x' f_{xx} \\ Q_{ux} &= \ell_{ux} + f_u^\top V_{xx} f_x + V_x' f_{ux} \end{aligned}$$

$$Q_{uu} = \ell_{uu} + f_u^\top V_{xx}' f_u + V_x' f_{uu} \tag{6}$$

Q에 대한 minimization을 진행하면

$$\delta u^\star = \arg\min_{\delta u} Q(\delta x, \delta u) = k + K\delta x \tag{7}$$

가 만족된다. 추가적인 constarints가 존재하지 않는다면 locally-linear한 상황에서

$$k \triangleq -Q_{uu}^{-1} Q_u, K \triangleq -Q_{uu}^{-1} Q_{ux} \tag{8}$$

를 만족한다. k와 K를 구한 것을 이용하여 V에 대한 variation을 구할 수 있다.

$$\begin{aligned} \Delta V &= -\frac{1}{2} k^\top Q_{uu} k \\ V_x &= Q_x - K^\top Q_{uu} k \\ V_{xx} &= Q_{xx} - K^\top Q_{uu} K \end{aligned} \tag{9}$$

수식 8는 constraints가 없는 경우이다. constraints가 존재하는 경우 k와 K에 대한 다른 방법이 사용된다. backward pass에서는 δx에 대한 부분을 알 수 없기 때문에, 0으로 두고 δu에 대한 부분을 구한다. 그러면 수식 7와 수식 4의 δx=0을 넣고 quadratic problem(QP)을 풀 수 있다. 그 결과

$$\begin{aligned} k = \arg\min_{\delta u} \frac{1}{2} \delta u^\top Q_{uu} \delta u \\ subject\ to\ \underline{b} \le u + \delta u \le \bar{b} \end{aligned}$$

QP를 풀 때 projected newtons's method를 사용한다. 이 과정에서 clamp가 안 된 free dimension이 나오게 된다. free dimension을 이용하며 $Q_{uu,f}$를 구하여 $K_f = -Q_{uu,f} Q_{ux}$를 통해 K를 구한다.

2) Forward Pass: Backward pass를 통해서 k,K의 값을 구하였고, 이를 이용하여 $\delta u^\star$를 구하였다. 이 δu를 통해서 u에 대한 update를 진행한다.

$$\begin{aligned} x_{0\_update} &= x_0 \\ u_{i\_update} &= u_i + \delta u \\ x_{i+1\_update} &= f(x_{i\_update}) + g(x_{i\_update}) u_{i\_update} \end{aligned} \tag{10}$$

수식 (10)에서의 δu는 $\alpha k_i + K_i(x_{i_update} - x_i)$이다. α는 1로 시작해서 점점 줄어드는 값이다.

B. Control Barrier Function

안전 영역 $\mathcal{C}$이 있다고 하고, 안전 영역의 구역을 다음과 같이 정의 가능하다.

$$\mathcal{C} \coloneqq \{x \in \mathbb{R} : h(x) \ge 0\} \tag{11}$$

$h(x) \geq 0$을 만족하는 조건을 가지는 h(x)와 함께, 장벽 함수를 구현해야 한다. 크게 2가지의 장벽 함수를 고려했는데, 장벽 함수의 형태는 다음과 같다.

$$b_1 = -log(\frac{h(x)}{1+h(x)}), b_2(x) = \frac{1}{h(x)} \tag{12}$$

위와 같이 barrier function을 설정하면, h(x)가 안전 영역 $\mathcal{C}$ 안에 있는 경우에는 항상 양수의 값을 가진다. 다만, 안전 영역의 경계로 다가갈수록 함수의 값이 커지게 된다.

1) Reciprocal Barrier Function:

$$\frac{1}{\alpha_1(h(x))} \leq b(x) \leq \frac{1}{\alpha_2(h(x))}, L_f(b(x)) \leq \alpha_3(h(x)) \tag{13}$$

α는 $class - \mathcal{K}$를 의미한다. 수식 13은 reciprocal barrier function이라고 부른다. reciprocal barrier function의 경우 $h(x)$가 경계에서는 $L_f h(x) \geq 0$ 을 만족해야 한다. 그래야 h(x)가 안전 영역에 머무를 수 있다.

2) Zeroing Barrier Function:

$$L_f h(x) \geq -\alpha(h(x)) \tag{14}$$

다음과 같은 h(x)를 zero barrier function이라고 한다.

3) reciprocal control barrier function:

$$\frac{1}{\alpha_1 h(x)} \leq b(x) \leq \frac{1}{\alpha_2 h(x)},$$
$$\inf_{u \in U}[L_f b(x) + L_g b(x) u] \leq \alpha_3(h(x)) \tag{15}$$

4) zeroing control barrier function :

$$\sup_{u \in U}[L_f h(x) + L_g h(x) u] \geq -\alpha(h(x)) \tag{16}$$

C. 문제 설정

현재 ackerman steering으로 움직이는 바퀴 4개 달린 자동차의 kinematics를 기준으로 하고 있다. $x = \begin{bmatrix} x \\ y \\ \theta \\ \phi \end{bmatrix}, u = \begin{bmatrix} v \\ \dot{\phi} \end{bmatrix}$ 일 때, 자동차의 kinematics는 다음과 같다.

$$\dot{x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x + \begin{bmatrix} dt * cos(\theta) & 0 \\ dt * sin(\theta) & 0 \\ 0 & \frac{dt * tan(\phi)}{L} \\ 0 & dt \end{bmatrix} u \tag{17}$$

여기서 L은 자동차의 앞바퀴에서 뒷바퀴까지의 거리를 의미한다.
위와 같은 kinematics를 통해 $x^{dot} = f(x) + g(x)u$를 구성했다. 또한, 수식 (4)을 완성시켰다.
여기까지의 과정으로 DDP를 이용한 경로 생성이 가능하다. 시작점을 출발점으로 하고 Car cost function의 x에 원하는 목적지인 $\hat{x}$를 추가하여 $x - \hat{x}$와 같이 구성하면, 출발점에서 목적지까지 이동하는 경로를 생성 가능하다.
이제 추가적으로 구성을 해야 하는 부분은 장애물의 회피에 관한 부분이다. 장애물은 Control Barrier Function의 형태를 가지며 DDP에서 quadratic problem을 풀 때 내부 constraint로 들어갈 예정이다. 또한, Constrait의 형태는 inequation의 형태를 가지고 있다.

D. 알고리즘 설명 및 시도

장애물을 나타내는 h(x)의 경우는 제일 기본적인 형태로 $h(x) = (x - xc)^2 + (y - yc)^2 - r^2$이다. 여기서 (xc,yc)는 장애물의 중심 (x,y)를 의미하고 r은 장애물의 반지름을 의미한다. h(x)의 기본형을 저렇게 잡아둔 것이지, 항상 저 형태로 유지되고 있지는 않다. -가 곱해지거나, $\log|h(x)|$가 취해지는 등 다양한 형태를 시도 중이다.

b(x)의 경우는 $-\log(\frac{h(x)}{1+h(x)})$를 사용하고 있다.

1) case 1:

$$h(x) = (x - xc)^2 + (y - yc)^2 - r^2$$
$$\text{or}$$
$$h(x) = r^2 - (x - xc)^2 - (y - yc)^2$$
$$B(x) = -\log\left(\frac{h(x)}{1 + h(x)}\right)$$

2) case 2:

$$\alpha = \arccos\left(\frac{xc(xc - x) + yc(yc - y)}{\sqrt{xc^2 + yc^2}\sqrt{(xc - x)^2 + (yc - y)^2}}\right)$$
$$h(x) = \frac{H_0}{(x - xc)^2 + (y - yc)^2 - r^2} + \frac{H_1}{180 - \theta - \alpha - \phi}$$
$$B(x) = -\log\left(\frac{h(x)}{1 + h(x)}\right)$$

이게 ϕ, θ를 고려하는 방식이었는데, 일단은 폐기

3) case 3:

$$h(x) = \frac{H_0}{(x - xc)^2 + (y - yc)^2 - r^2}$$
$$B(x) = -\log\left(\frac{h(x)}{1 + h(x)}\right)$$

4) case 4: 위의 case들에서 r을 $r + extra_size$로 변경 $extra_size$를 통해서 r보다 큰 bumper를 만들려고 시도

5) case 5:

$$h(x) = (r^2 - (x - xc)^2 - (y - yc)^2)u((r + 5)^2 - (x - xc)^2 - (y - yc)^2$$
$$B(x) = -\log\left(\frac{h(x)}{1 + h(x)}\right)$$

Matlab 상에서 heaviside를 통해서 $u(t)$를 구성, 5가 r에 더해진 이유는 변화가 생기는 지점을 조절하고 싶어서 넣었다.

6) case 6:

$$h(x) = \log\left\|(x - xc)^2 + (y - yc)^2 - r^2\right\|$$
$$B(x) = -\log\left(\frac{h(x)}{1 + h(x)}\right) \text{or} B(x) = \frac{1}{h(x)}$$

E. Theorem 환경 사용

연구실에서 수행하는 연구로 논문을 작성할 때, 경우에 따라 definition, lemma, theorem 등을 문서에 추가할 필요가 있습니다. 이 경우 IEEE 템플릿에서 공식 제공하거나 `amsthm` package를 통해 `newtheorem`으로 정의된 definition, lemma, theorem 함수 등을 아래와 같이 활용할 수 있습니다.

Definition 1: Blabla... □

Lemma 1: Blabla... □

Theorem 1: Blabla... □

Proof: Blabla... ■

수식과 마찬가지로 `label` 함수를 이용하여 Theorem 1과 같이 상호 참조도 가능합니다.

F. Reference 작성

LaTex에서는 `thebibliography`라는 함수 혹은 `bib` 파일을 이용하여 reference를 관리합니다. Reference 작성법은 별도로 연구실 Wiki를 통해 소개하고, 추후 템플릿에도 요약하여 정리해놓도록 하겠습니다. Rerefenrece 작성 예시는 본 문서의 제일 마지막을 참고해주세요.

G. 유용한 참고 자료

아래 리스트에서는 LaTex 사용과 관련하여 유용한 참고 자료들을 소개합니다.

- **Overleaf documentation** : https://www.overleaf.com/learn
- **LaTex 관련 Youtube 영상** (KAIST 수리과학과) : https://www.youtube.com/watch?v=ysnKC1jEC1s
- **Detexify** : https://detexify.kirelabs.org/classify.html
- **KTUG** (한국 텍 사용자 그룹) :
- **LaTex - Beginner's Guide** : http://static.latexstudio.net/wp-content/uploads/2015/03/LaTeX_Beginners_Guide.pdf
- **IEEE Transactions Template 사용법** : https://www.cs.cmu.edu/~steffan/personal/tmp/IEEEtran_HOWTO.pdf
- (이후 업데이트 될 예정입니다.)

REFERENCES

[1] *IEEE Journal Paper Template*, IEEE, https://www.overleaf.com/latex/templates/ieee-journal-paper-template/jbbbdkztwxrd

[2] *LaTex*, Wikipedia, https://en.wikipedia.org/wiki/LaTeX