4-14 Covariant Hamiltonian Optimization for Motion Planning (CHOMP)

강의 요약

01

최적화 기법 vs. 샘플링 기법

02

최적화 문제 정의

- 목적 함수
- 설계 변수
- 제약 조건

03

최적화 문제 특징

- 문제 정의의 자유도가 높음
- Local minimum problem
- 시작점 위치의 영향
- 경우에 따라서는 솔루션을 찾을 수 없기도 함
- Deterministic & Stochastic Gradient Descent
- 이산 최적화

04

최적화 라이브러리

- High-level: 실제 적용 목적
- Low-level: 연구 목적

최적화 (optimization) - 모션 플래닝

$$\min_{x \in \mathcal{X}} f(x)$$
$$g_i(x) \le 0,\ h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 설계 변수 (design variables)

  궤적 위치 $q(t)$

  궤적 속도 $v(t)$

  제어 입력(가속도) $u(t)$

  총 궤적 실행 시간 T

- 목적 함수 (objective function)

  $$\int_0^T \|u(t)\|^2\, dt$$

- 제약 조건 (constraints)

  $$\begin{aligned} &\dot{q}(t) = v(t), \\ &\dot{v}(t) = u(t), \\ &q(0) = q_{\text{start}}, \quad v(0) = v_{\text{start}}, \\ &q(T) = q_{\text{goal}}, \quad v(T) = v_{\text{goal}}, \\ &q(t) \in \mathcal{C}_{\text{free}} \quad \forall t \in [0, T], \\ &u_{\min} \le u(t) \le u_{\max} \quad \forall t \in [0, T] \end{aligned}$$

최적화 (optimization) - 모션 플래닝

$$\min_{x \in \mathcal{X}} f(x)$$
$$g_i(x) \le 0,\ h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 설계 변수 (design variables)

  궤적 위치 $q(t)$

  궤적 속도 $v(t)$

  제어 입력(가속도) $u(t)$

  총 궤적 실행 시간 T

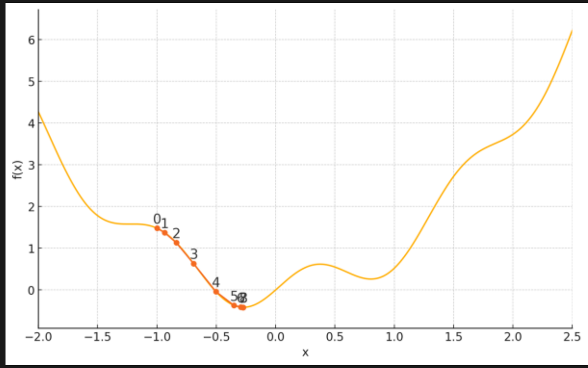
- 목적 함수 (objective function)

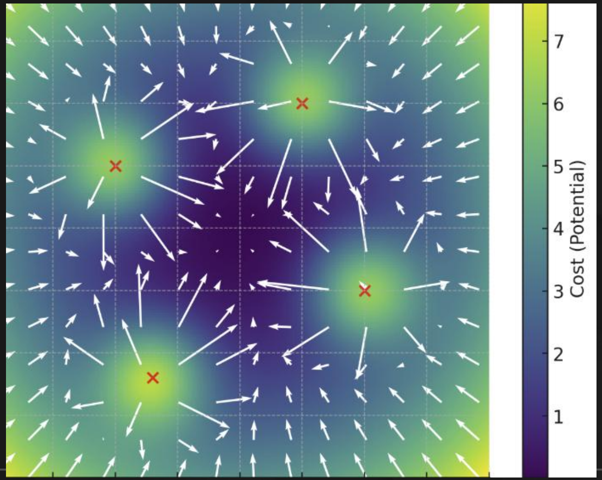
$$\int_0^T \|u(t)\|^2\, dt$$

- 제약 조건 (constraints)

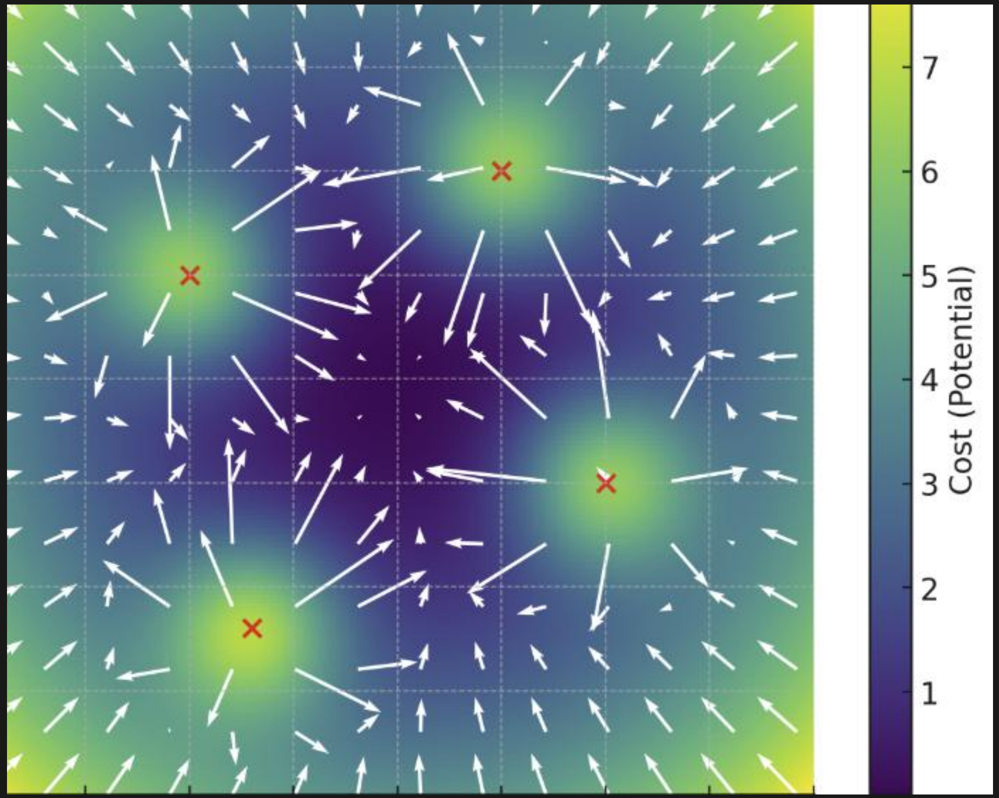
$$\begin{aligned}
&\dot{q}(t) = v(t), \\
&\dot{v}(t) = u(t), \\
&q(0) = q_{\text{start}}, \quad v(0) = v_{\text{start}}, \\
&q(T) = q_{\text{goal}}, \quad v(T) = v_{\text{goal}}, \\
&q(t) \in \mathcal{C}_{\text{free}} \quad \forall t \in [0, T], \\
&u_{\min} \le u(t) \le u_{\max} \quad \forall t \in [0, T]
\end{aligned}$$

- 어떻게 솔루션을 찾을까?
  → 경사 하강법 (gradient descent)
  → 시작점 설정
  → 변수 갱신
  → **반복!**





CHOMP - 개요



CHOMP

$$\min_{x \in \mathcal{X}} f(x)$$
$$g_i(x) \le 0,\ h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 설계 변수 (design variables)

  궤적 위치 $q(t)$

  궤적 속도 $v(t)$

  제어 입력(가속도) $u(t)$

  총 궤적 실행 시간 T

- 목적 함수 (objective function)

  $$\int_0^T \|u(t)\|^2 \, dt$$

- 제약 조건 (constraints)

  $$\begin{aligned}
  &\dot{q}(t) = v(t), \\
  &\dot{v}(t) = u(t), \\
  &q(0) = q_{\text{start}}, \quad v(0) = v_{\text{start}}, \\
  &q(T) = q_{\text{goal}}, \quad v(T) = v_{\text{goal}}, \\
  &q(t) \in \mathcal{C}_{\text{free}} \quad \forall t \in [0, T], \\
  &u_{\min} \le u(t) \le u_{\max} \quad \forall t \in [0, T]
  \end{aligned}$$

CHOMP

$$\min_{x \in \mathcal{X}} f(x)$$
$$g_i(x) \le 0,\ h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 설계 변수 (design variables)

궤적 위치 $q(t)$

궤적 속도 $v(t)$

제어 입력(가속도) $u(t)$

총 궤적 실행 시간 T

$$\mathbf{q} = [\,q_0,\, q_1,\, \dots,\, q_{N-1}\,]^\top, \quad q_i \in \mathbb{R}^d$$

- 목적 함수 (objective function)

$$\int_0^T \|u(t)\|^2 \, dt$$

$$J(\mathbf{q}) = \underbrace{J_{\text{smooth}}(\mathbf{q})}_{\text{trajectory smoothness}} + \underbrace{J_{\text{obs}}(\mathbf{q})}_{\text{obstacle avoidance}}$$

- 제약 조건 (constraints)

$$\begin{aligned}
&\dot{q}(t) = v(t),\\
&\dot{v}(t) = u(t),\\
&q(0) = q_{\text{start}}, \quad v(0) = v_{\text{start}},\\
&q(T) = q_{\text{goal}}, \quad v(T) = v_{\text{goal}},\\
&q(t) \in \mathcal{C}_{\text{free}} \quad \forall t \in [0, T],\\
&u_{\min} \le u(t) \le u_{\max} \quad \forall t \in [0, T]
\end{aligned}$$

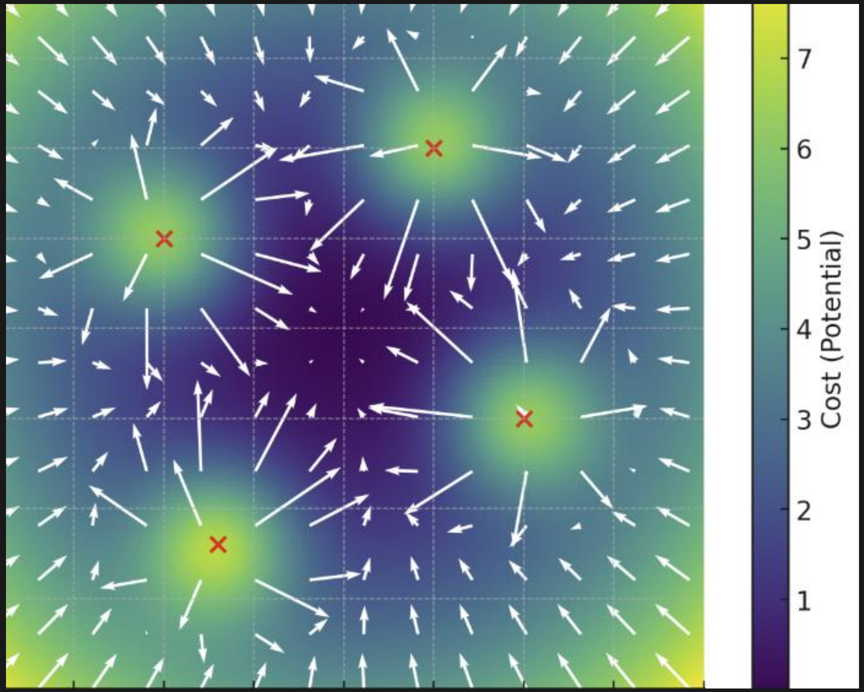
$$\begin{aligned}
&\begin{cases} q_0 = q_{\text{start}},\\ q_{N-1} = q_{\text{goal}}, \end{cases}\\
&\forall i = 1, \dots, N-2: \quad \ell \le q_i \le u,\\
&\|\, q_i - q_{i-1} \| \le v_{\max}\, \Delta t, \quad i = 1, \dots, N-1,\\
&\|\, q_{i+1} - 2\, q_i + q_{i-1} \| \le a_{\max}\, (\Delta t)^2, \quad i = 1, \dots, N-2,\\
&q_i \in C_{\text{free}} \quad \text{for } i = 0, \dots, N-1
\end{aligned}$$

CHOMP

- 설계 변수 (design variables)

$$\mathbf{q} = [\, q_0,\, q_1,\, \ldots,\, q_{N-1} \,]^\top, \quad q_i \in \mathbb{R}^d$$

시작점과 도착점을 연결하는 직선으로 초기화



CHOMP

- 목적 함수 (objective function)

$$J(\mathbf{q}) = \underbrace{J_{\text{smooth}}(\mathbf{q})}_{\text{trajectory smoothness}} + \underbrace{J_{\text{obs}}(\mathbf{q})}_{\text{obstacle avoidance}}$$

- Smoothness Cost: 가속도를 최소화.

- Obstacle Cost: 경로를 장애물로부터 먼 방향으로 밀어내는 역할.

CHOMP

- 목적 함수 (objective function)

$$J(\mathbf{q}) = \underbrace{J_{\text{smooth}}(\mathbf{q})}_{\text{trajectory smoothness}} + \underbrace{J_{\text{obs}}(\mathbf{q})}_{\text{obstacle avoidance}}$$

- Smoothness Cost: 가속도를 최소화.

$$J_{\text{smooth}}(\mathbf{q}) = \sum_{i=1}^{N-2} \|q_{i+1} - 2\,q_i + q_{i-1}\|_W^2$$

- Obstacle Cost: 경로를 장애물로부터 먼 방향으로 밀어내는 역할.

CHOMP

- 목적 함수 (objective function)

$$J(\mathbf{q}) = \underbrace{J_{\text{smooth}}(\mathbf{q})}_{\text{trajectory smoothness}} + \underbrace{J_{\text{obs}}(\mathbf{q})}_{\text{obstacle avoidance}}$$

- Smoothness Cost: 가속도를 최소화.

$$J_{\text{smooth}}(\mathbf{q}) = \sum_{i=1}^{N-2} \| q_{i+1} - 2\, q_i + q_{i-1} \|_W^2$$

- Obstacle Cost: 경로를 장애물로부터 먼 방향으로 밀어내는 역할.

CHOMP

- 목적 함수 (objective function)

$$J(\mathbf{q}) = \underbrace{J_{\text{smooth}}(\mathbf{q})}_{\text{trajectory smoothness}} + \underbrace{J_{\text{obs}}(\mathbf{q})}_{\text{obstacle avoidance}}$$

- Smoothness Cost: 가속도를 최소화.

$$\begin{aligned} J_{\text{smooth}}(\mathbf{q}) &= \sum_{i=1}^{N-2} \| q_{i+1} - 2\, q_i + q_{i-1} \|_W^2 \\ &= \mathbf{q}^\top (A^\top W A)\, \mathbf{q} \end{aligned}$$

- Obstacle Cost: 경로를 장애물로부터 먼 방향으로 밀어내는 역할.

CHOMP

- 목적 함수 (objective function)

$$J(\mathbf{q}) = \underbrace{J_{\text{smooth}}(\mathbf{q})}_{\text{trajectory smoothness}} + \underbrace{J_{\text{obs}}(\mathbf{q})}_{\text{obstacle avoidance}}$$

- Smoothness Cost: 가속도를 최소화.

$$\begin{aligned} J_{\text{smooth}}(\mathbf{q}) &= \sum_{i=1}^{N-2} \left\| q_{i+1} - 2\,q_i + q_{i-1} \right\|_W^2 \\ &= \mathbf{q}^\top (A^\top W A)\,\mathbf{q} \end{aligned}$$

- Obstacle Cost: 경로를 장애물로부터 먼 방향으로 밀어내는 역할.

$$J_{\text{obs}}(\mathbf{q}) = \sum_{i=0}^{N-1} \sum_{c=1}^{C} U\big(\phi(f_c(q_i))\big)\,\Delta t$$

CHOMP

- 목적 함수 (objective function)

$$J(\mathbf{q}) = \underbrace{J_{\text{smooth}}(\mathbf{q})}_{\text{trajectory smoothness}} + \underbrace{J_{\text{obs}}(\mathbf{q})}_{\text{obstacle avoidance}}$$

- Smoothness Cost: 가속도를 최소화.

$$\begin{aligned} J_{\text{smooth}}(\mathbf{q}) &= \sum_{i=1}^{N-2} \|q_{i+1} - 2\,q_i + q_{i-1}\|_W^2 \\ &= \mathbf{q}^\top (A^\top W A)\,\mathbf{q} \end{aligned}$$

- Obstacle Cost: 경로를 장애물로부터 먼 방향으로 밀어내는 역할.

$$J_{\text{obs}}(\mathbf{q}) = \sum_{i=0}^{N-1} \sum_{c=1}^{C} U\big(\phi(f_c(q_i))\big)\,\Delta t$$

$\phi(x)$ Signed Distance Function (SDF)
→ 장애물과 가장 가까운 거리를 출력
→ Work space 에서 적용

CHOMP

- 목적 함수 (objective function)

$$J(\mathbf{q}) = \underbrace{J_{\text{smooth}}(\mathbf{q})}_{\text{trajectory smoothness}} + \underbrace{J_{\text{obs}}(\mathbf{q})}_{\text{obstacle avoidance}}$$

- Smoothness Cost: 가속도를 최소화.

$$J_{\text{smooth}}(\mathbf{q}) = \sum_{i=1}^{N-2} \|q_{i+1} - 2\,q_i + q_{i-1}\|_W^2 = \mathbf{q}^\top (A^\top W A)\,\mathbf{q}$$

- Obstacle Cost: 경로를 장애물로부터 먼 방향으로 밀어내는 역할.

$$J_{\text{obs}}(\mathbf{q}) = \sum_{i=0}^{N-1} \sum_{c=1}^{C} U\big(\phi(f_c(q_i))\big)\,\Delta t$$

$\phi(x)$ Signed Distance Function (SDF)
→ 장애물과 가장 가까운 거리를 출력
→ Work space 에서 적용

$f_c(q_i)$ Forward Kinematics
→ C-Space 에서 Work Space 로 변환

$$x_{i,c} = f_c(q_i), \quad c = 1, \ldots, C$$

CHOMP

- 목적 함수 (objective function)

$$J(\mathbf{q}) = \underbrace{J_{\text{smooth}}(\mathbf{q})}_{\text{trajectory smoothness}} + \underbrace{J_{\text{obs}}(\mathbf{q})}_{\text{obstacle avoidance}}$$

- Smoothness Cost: 가속도를 최소화.

$$\begin{aligned} J_{\text{smooth}}(\mathbf{q}) &= \sum_{i=1}^{N-2} \| q_{i+1} - 2\, q_i + q_{i-1} \|_W^2 \\ &= \mathbf{q}^\top (A^\top W A)\, \mathbf{q} \end{aligned}$$

- Obstacle Cost: 경로를 장애물로부터 먼 방향으로 밀어내는 역할.

$$J_{\text{obs}}(\mathbf{q}) = \sum_{i=0}^{N-1} \sum_{c=1}^{C} U(\phi(f_c(q_i)))\, \Delta t$$

$\phi(x)$ Signed Distance Function (SDF)
→ 장애물과 가장 가까운 거리를 출력
→ Work space 에서 적용

$f_c(q_i)$ Forward Kinematics
→ C-Space 에서 Work Space 로 변환

$$x_{i,c} = f_c(q_i), \quad c = 1, \ldots, C$$

$$U(\phi) = \begin{cases} 0, & \phi > d_{\text{inflation}}, \\ \frac{1}{2}\left(d_{\text{inflation}} - \phi\right)^2, & 0 \le \phi \le d_{\text{inflation}}, \\ -\frac{1}{2}\left(\phi + d_{\text{in}}\right)^2 + \text{const}, & \phi < 0, \end{cases}$$

CHOMP

- 제약 조건 (constraints)

$$\begin{cases} q_0 = q_{\text{start}}, \\ q_{N-1} = q_{\text{goal}}, \end{cases}$$
$$\forall i = 1, \ldots, N-2: \quad \ell \le q_i \le u,$$
$$\| q_i - q_{i-1} \| \le v_{\max} \Delta t, \quad i = 1, \ldots, N-1,$$
$$\| q_{i+1} - 2\, q_i + q_{i-1} \| \le a_{\max} (\Delta t)^2, \quad i = 1, \ldots, N-2,$$
$$q_i \in C_{\text{free}} \quad \text{for } i = 0, \ldots, N-1$$

CHOMP

$$\min_{x \in \mathcal{X}} f(x)$$
$$g_i(x) \le 0,\ h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 설계 변수 (design variables)

$$\mathbf{q} = [\, q_0,\ q_1,\ \dots,\ q_{N-1} \,]^\top, \quad q_i \in \mathbb{R}^d$$

- 목적 함수 (objective function)

$$J(\mathbf{q}) \ = \ \underbrace{J_{\text{smooth}}(\mathbf{q})}_{\text{trajectory smoothness}} \ + \ \underbrace{J_{\text{obs}}(\mathbf{q})}_{\text{obstacle avoidance}}$$

- 제약 조건 (constraints)

$$\begin{cases} q_0 = q_{\text{start}}, \\ q_{N-1} = q_{\text{goal}}, \end{cases}$$
$$\forall\, i = 1, \dots, N-2: \quad \ell \le q_i \le u,$$
$$\|\, q_i - q_{i-1} \| \le v_{\max}\, \Delta t, \quad i = 1, \dots, N-1,$$
$$\|\, q_{i+1} - 2\, q_i + q_{i-1} \| \le a_{\max}\, (\Delta t)^2, \quad i = 1, \dots, N-2,$$
$$q_i \in C_{\text{free}} \quad \text{for } i = 0, \dots, N-1$$

CHOMP

$$\min_{x \in \mathcal{X}} f(x)$$
$$g_i(x) \le 0,\; h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 설계 변수 (design variables)

$$\mathbf{q} = [\, q_0,\, q_1,\, \ldots,\, q_{N-1} \,]^\top, \quad q_i \in \mathbb{R}^d$$

- 목적 함수 (objective function)

$$J(\mathbf{q}) \;=\; \underbrace{J_{\text{smooth}}(\mathbf{q})}_{\text{trajectory smoothness}} \;+\; \underbrace{J_{\text{obs}}(\mathbf{q})}_{\text{obstacle avoidance}}$$

- 제약 조건 (constraints)

$$\begin{cases} q_0 = q_{\text{start}}, \\ q_{N-1} = q_{\text{goal}}, \end{cases}$$
$$\forall\, i = 1, \ldots, N-2 : \quad \ell \le q_i \le u,$$
$$\|\, q_i - q_{i-1} \| \le v_{\max}\, \Delta t, \quad i = 1, \ldots, N-1,$$
$$\|\, q_{i+1} - 2\, q_i + q_{i-1} \| \le a_{\max}\, (\Delta t)^2, \quad i = 1, \ldots, N-2,$$
$$q_i \in C_{\text{free}} \quad \text{for } i = 0, \ldots, N-1$$

- 어떻게 솔루션을 찾을까?
  → 경사 하강법 (gradient descent)
  → 시작점 설정
  → 변수 갱신
  → **반복!**

CHOMP

- 솔루션을 찾는 방법: 경사 하강법 (Gradient Descent) 기법을 응용

$$\nabla J(\mathbf{q}) \;=\; \nabla J_{\text{smooth}}(\mathbf{q}) \;+\; \nabla J_{\text{obs}}(\mathbf{q})$$

CHOMP

- 솔루션을 찾는 방법: 경사 하강법 (Gradient Descent) 기법을 응용

$$\nabla J(\mathbf{q}) \;=\; \nabla J_{\text{smooth}}(\mathbf{q}) \;+\; \nabla J_{\text{obs}}(\mathbf{q})$$

기본 경사 하강법: $\mathbf{q} \leftarrow \mathbf{q} - \alpha \nabla J(\mathbf{q})$

CHOMP

- 솔루션을 찾는 방법: 경사 하강법 (Gradient Descent) 기법을 응용

$$\nabla J(\mathbf{q}) \;=\; \nabla J_{\text{smooth}}(\mathbf{q}) \;+\; \nabla J_{\text{obs}}(\mathbf{q})$$

기본 경사 하강법: $\mathbf{q} \leftarrow \mathbf{q} - \alpha \nabla J(\mathbf{q})$

→ 기본 경사 하강법의 문제점: 간혹 안정적인 업데이트 X, Local minimum

CHOMP

- 솔루션을 찾는 방법: 경사 하강법 (Gradient Descent) 기법을 응용

$$\nabla J(\mathbf{q}) \ = \ \nabla J_{\text{smooth}}(\mathbf{q}) \ + \ \nabla J_{\text{obs}}(\mathbf{q})$$

기본 경사 하강법: $\mathbf{q} \leftarrow \mathbf{q} - \alpha \nabla J(\mathbf{q})$

→ 기본 경사 하강법의 문제점: 간혹 안정적인 업데이트 X, Local minimum

→ 모션의 안정적인 업데이트를 위해 Smoothness term 을 활용하여 경사하강

CHOMP

- 솔루션을 찾는 방법: 경사 하강법 (Gradient Descent) 기법을 응용

$$\nabla J(\mathbf{q}) \;=\; \nabla J_{\text{smooth}}(\mathbf{q}) \;+\; \nabla J_{\text{obs}}(\mathbf{q})$$

기본 경사 하강법: $$\mathbf{q} \leftarrow \mathbf{q} - \alpha \nabla J(\mathbf{q})$$

→ 기본 경사 하강법의 문제점: 간혹 안정적인 업데이트 X, Local minimum

→ 모션의 안정적인 업데이트를 위해 Smoothness term 을 활용하여 경사하강

$$J_{\text{smooth}}(\mathbf{q}) \;=\; \sum_{i=1}^{N-2} \|q_{i+1} - 2\,q_i + q_{i-1}\|_W^2 \;=\; \mathbf{q}^\top (A^\top W A)\,\mathbf{q}$$

$$\mathbf{q} \leftarrow \mathbf{q} - \alpha\, M^{-1}\, \nabla J$$

CHOMP

- 솔루션을 찾는 방법: 경사 하강법 (Gradient Descent) 기법을 응용

$$\nabla J(\mathbf{q}) \;=\; \nabla J_{\text{smooth}}(\mathbf{q}) \;+\; \nabla J_{\text{obs}}(\mathbf{q})$$

기본 경사 하강법: $\mathbf{q} \leftarrow \mathbf{q} - \alpha \nabla J(\mathbf{q})$

→ 기본 경사 하강법의 문제점: 간혹 안정적인 업데이트 X, Local minimum

→ 모션의 안정적인 업데이트를 위해 Smoothness term 을 활용하여 경사하강

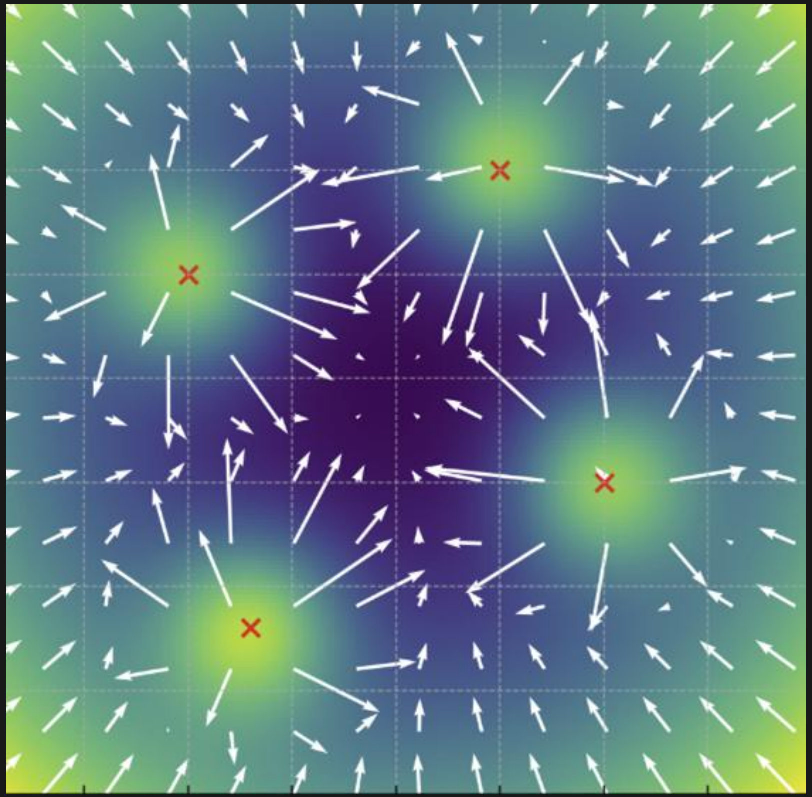
$$\begin{aligned} J_{\text{smooth}}(\mathbf{q}) \;&=\; \sum_{i=1}^{N-2} \| q_{i+1} - 2\, q_i + q_{i-1} \|_W^2 \\ &=\; \mathbf{q}^\top (A^\top W A)\, \mathbf{q} \end{aligned}$$

$$\mathbf{q} \leftarrow \mathbf{q} - \alpha\, M^{-1}\, \nabla J$$

→ 반복! $\mathbf{q}^{(k+1)} \;=\; \mathbf{q}^{(k)} \;-\; \alpha\, M^{-1} \left[\nabla J_{\text{smooth}}\left(\mathbf{q}^{(k)}\right) \;+\; \nabla J_{\text{obs}}\left(\mathbf{q}^{(k)}\right) \right]$

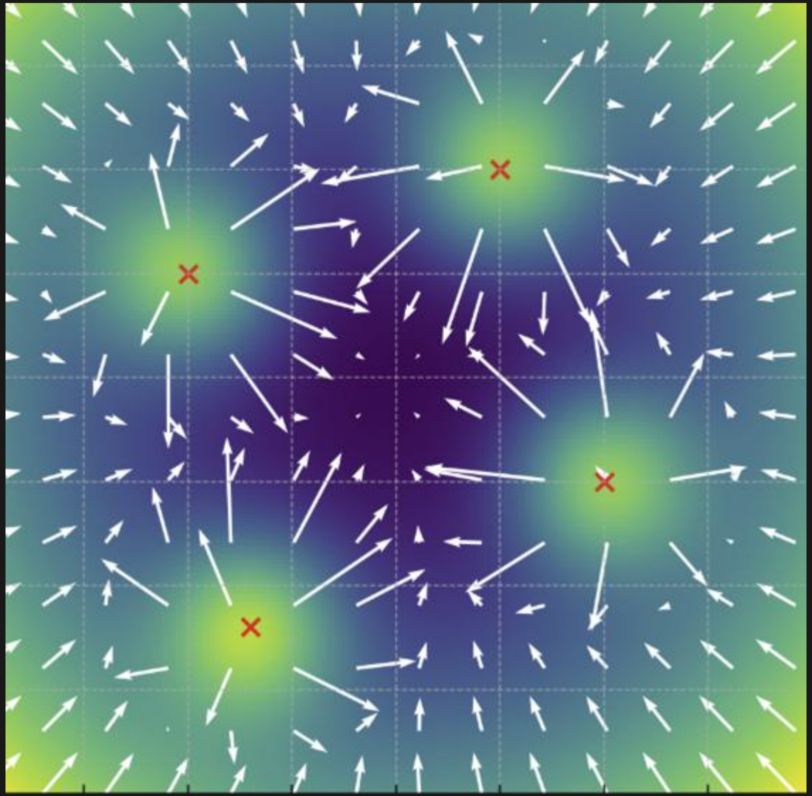
CHOMP

C-Space (Joint Space)

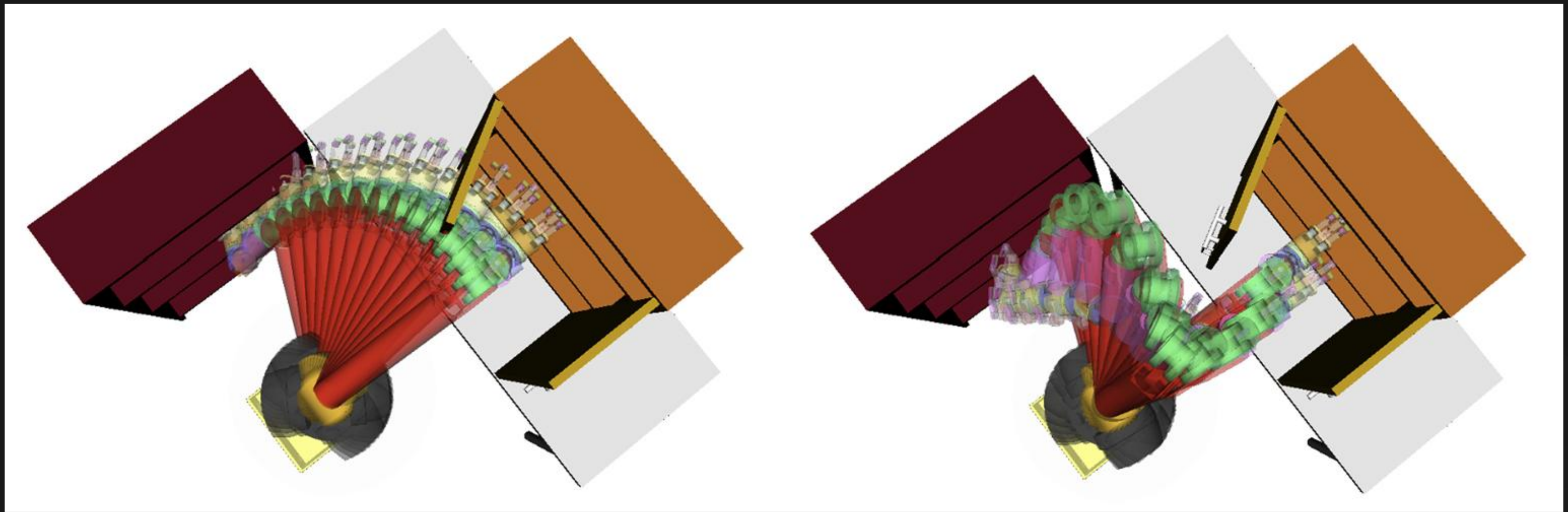


CHOMP

C-Space (Joint Space)



Work Space (Task Space)



Nathan et al. CHOMP: Gradient Optimization Techniques for Efficient Motion Planning

CHOMP

$$\min_{x \in \mathcal{X}} f(x)$$
$$g_i(x) \le 0,\ h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 설계 변수 (design variables)

$$\mathbf{q} = \left[\, q_0,\ q_1,\ \ldots,\ q_{N-1} \,\right]^{\top}, \quad q_i \in \mathbb{R}^d$$

- 목적 함수 (objective function)

$$J(\mathbf{q}) \;=\; \underbrace{J_{\text{smooth}}(\mathbf{q})}_{\text{trajectory smoothness}} \;+\; \underbrace{J_{\text{obs}}(\mathbf{q})}_{\text{obstacle avoidance}}$$

- 제약 조건 (constraints)

$$\begin{cases} q_0 \;=\; q_{\text{start}}, \\ q_{N-1} \;=\; q_{\text{goal}}, \end{cases}$$
$$\forall\, i = 1, \ldots, N-2 : \quad \ell \;\le\; q_i \;\le\; u,$$
$$\|\, q_i - q_{i-1} \| \;\le\; v_{\max}\, \Delta t, \quad i = 1, \ldots, N-1,$$
$$\|\, q_{i+1} - 2\, q_i + q_{i-1} \| \;\le\; a_{\max}\, (\Delta t)^2, \quad i = 1, \ldots, N-2,$$
$$q_i \;\in\; C_{\text{free}} \quad \text{for } i = 0, \ldots, N-1$$

- 솔루션을 찾는 방법: 경사 하강법 (Gradient Descent) 기법을 응용

  → 반복이 핵심!

$$\mathbf{q}^{(k+1)} \;=\; \mathbf{q}^{(k)} \;-\; \alpha\, M^{-1} \left[\nabla J_{\text{smooth}}\left(\mathbf{q}^{(k)}\right) \;+\; \nabla J_{\text{obs}}\left(\mathbf{q}^{(k)}\right) \right]$$

강의 요약

01

문제 정의

- 목적 함수
  → Smooth + obs.
- 설계 변수
  → Traj.
- 제약 조건
  → 시작점, range

02

Gradient Descent

- Configuration 의 관계를 고려한 (Covariance) 업데이트
- Smoothness 에서 차용한 M matrix

03

직관적 이해

- 초기값을 당기고 펴는 과정
- 조금씩 찾아가는 과정