4-15 Stochastic Trajectory Optimization for Motion Planning (STOMP)

Fast campus

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

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)

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

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

  → 반복이 핵심!

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

CHOMP

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

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)

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

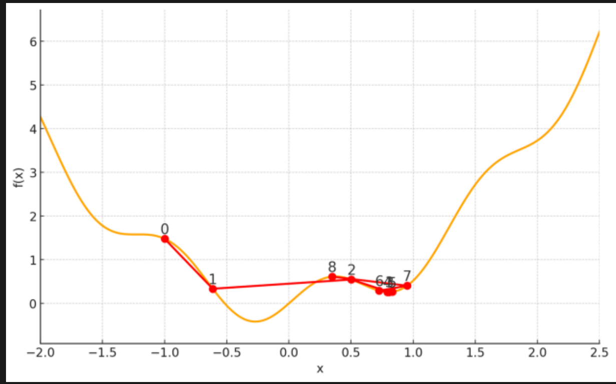
- 제약 조건 (constraints)

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

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

→ 반복이 핵심!

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



1. Deterministic

(Local Minimum Problem)

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)

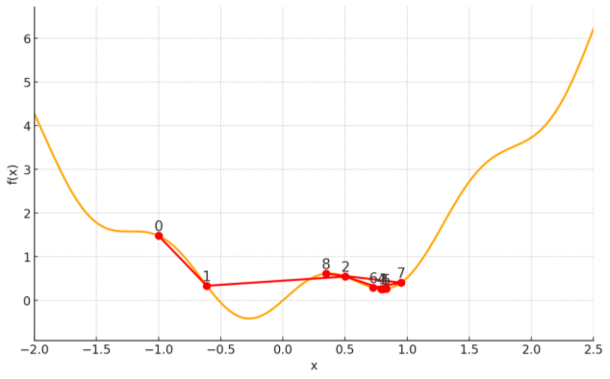
$$J(\mathbf{q}) = \underbrace{J_{\text{smooth}}(\mathbf{q})}_{\text{trajectory smoothness}} + \underbrace{J_{\text{obs}}(\mathbf{q})}_{\text{obstacle avoidance}}$$

- 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!



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

1. Deterministic

   (Local Minimum Problem)

2. Differentiable (정형적 성격)

STOMP

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

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)

- 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

STOMP

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

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)
- 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

  → 1. Stochastic Gradient Descent 를 활용한다면?

STOMP

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

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)
  → 2. 미분 불가능한 함수라면?

- 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
  → 1. Stochastic Gradient Descent 를 활용한다면?

STOMP

목적 함수: 미분이 가능하지 않아도 됨 (자유도가 높음)

- 장애물 회피 (Obstacle)
- 경로의 부드러움 (Smoothness)
- 속도 제한 비용 (Velocity limit)

STOMP

목적 함수: 미분이 가능하지 않아도 됨 (자유도가 높음)

- 장애물 회피 (Obstacle)

$$c_{\text{obs}}(\mathbf{q}_t) = \begin{cases} \alpha \cdot (d_{\text{safe}} - d(\mathbf{q}_t))^2 & \text{if } d(\mathbf{q}_t) < d_{\text{safe}} \\ 0 & \text{otherwise} \end{cases}$$

$d(\mathbf{q}_t)$: configuration $\mathbf{q}_t$에서 가장 가까운 장애물까지 거리

α: penalty scale

d_{safe}: 안전 거리 threshold

- 경로의 부드러움 (Smoothness)

$$c_{\text{smooth}} = \sum_{t=2}^{T-1} \|\mathbf{q}_{t+1} - 2\mathbf{q}_t + \mathbf{q}_{t-1}\|^2$$

- 속도 제한 비용 (Velocity limit)

$$c_{\text{vel}} = \sum_{t=1}^{T-1} \max(0, \|\dot{\mathbf{q}}_t\| - v_{\max})^2$$

$$c(\xi) = w_{\text{obs}} \cdot c_{\text{obs}} + w_{\text{smooth}} \cdot c_{\text{smooth}} + w_{\text{vel}} \cdot c_{\text{vel}}$$

STOMP

목적 함수: 미분이 가능하지 않아도 됨 (자유도가 높음)

- 장애물 회피 (Obstacle)

$$c_{\text{obs}}(\mathbf{q}_t) = \begin{cases} \alpha \cdot (d_{\text{safe}} - d(\mathbf{q}_t))^2 & \text{if } d(\mathbf{q}_t) < d_{\text{safe}} \\ 0 & \text{otherwise} \end{cases}$$

$d(\mathbf{q}_t)$: configuration $\mathbf{q}_t$에서 가장 가까운 장애물까지 거리

α: penalty scale

d_{safe}: 안전 거리 threshold

→ Binary Collision Check 도 활용 가능

- 경로의 부드러움 (Smoothness)

$$c_{\text{smooth}} = \sum_{t=2}^{T-1} \left\| \mathbf{q}_{t+1} - 2\mathbf{q}_t + \mathbf{q}_{t-1} \right\|^2$$

- 속도 제한 비용 (Velocity limit)

$$c_{\text{vel}} = \sum_{t=1}^{T-1} \max(0, \|\dot{\mathbf{q}}_t\| - v_{\max})^2$$

$$c(\xi) = w_{\text{obs}} \cdot c_{\text{obs}} + w_{\text{smooth}} \cdot c_{\text{smooth}} + w_{\text{vel}} \cdot c_{\text{vel}}$$

STOMP

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

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)

$$c(\xi) = w_{\text{obs}} \cdot c_{\text{obs}} + w_{\text{smooth}} \cdot c_{\text{smooth}} + w_{\text{vel}} \cdot c_{\text{vel}}$$

- 제약 조건 (constraints)

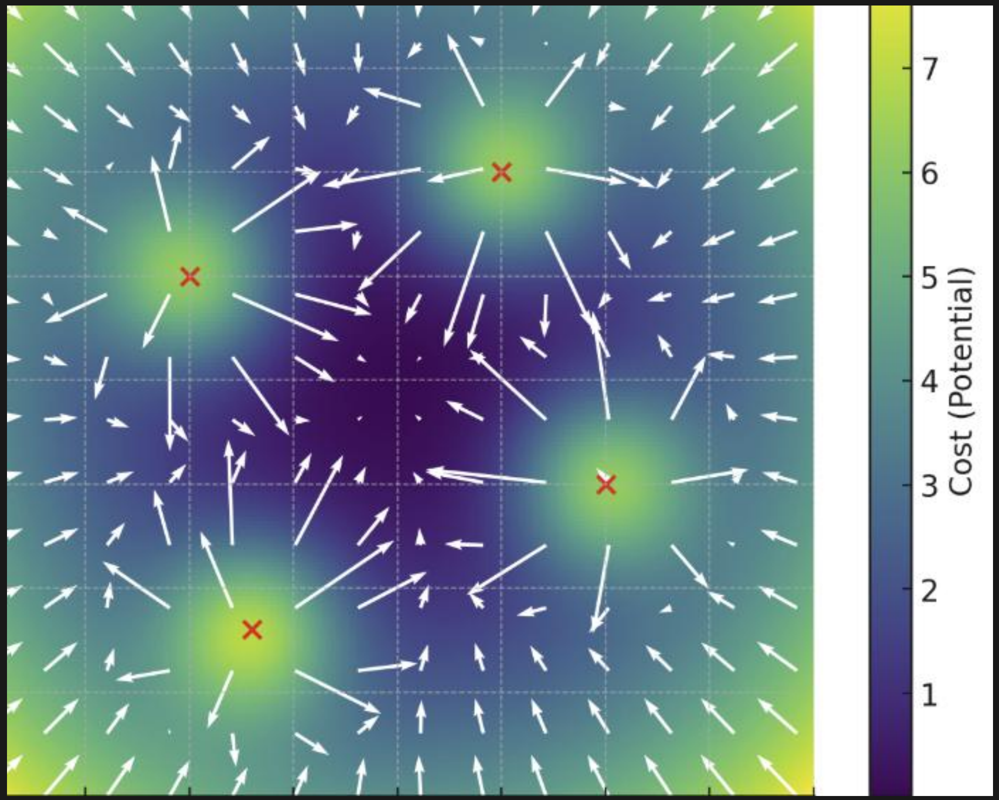
$$\begin{cases} q_0 = q_{\text{start}}, \\ q_{N-1} = q_{\text{goal}}, \end{cases}$$
$$\forall i = 1, \ldots, N-2 : \quad \ell \le q_i \le u,$$
$$\| q_i - q_{i-1} \| \le v_{\max} \Delta t, \quad i = 1, \ldots, N-1,$$
$$\| q_{i+1} - 2\, q_i + q_{i-1} \| \le a_{\max} (\Delta t)^2, \quad i = 1, \ldots, N-2,$$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

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

  → 1. Stochastic Gradient Descent 를 활용한다면?

STOMP

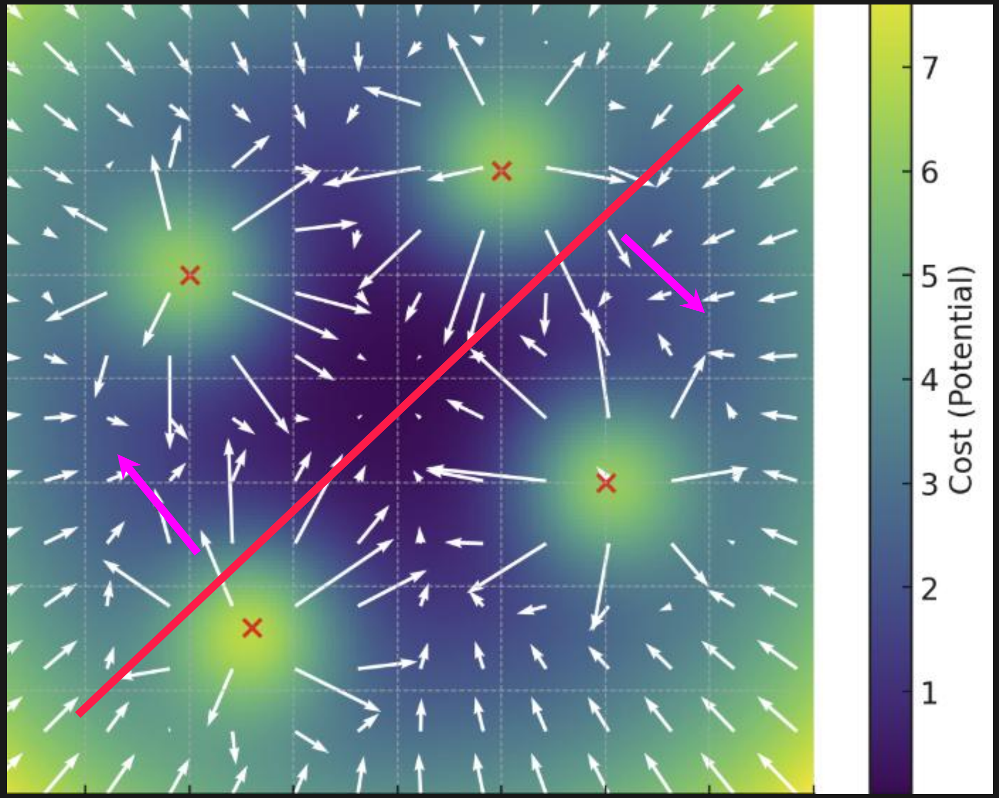
경사 하강법: Deterministic vs. Stochastic



STOMP

경사 하강법: Deterministic vs. Stochastic

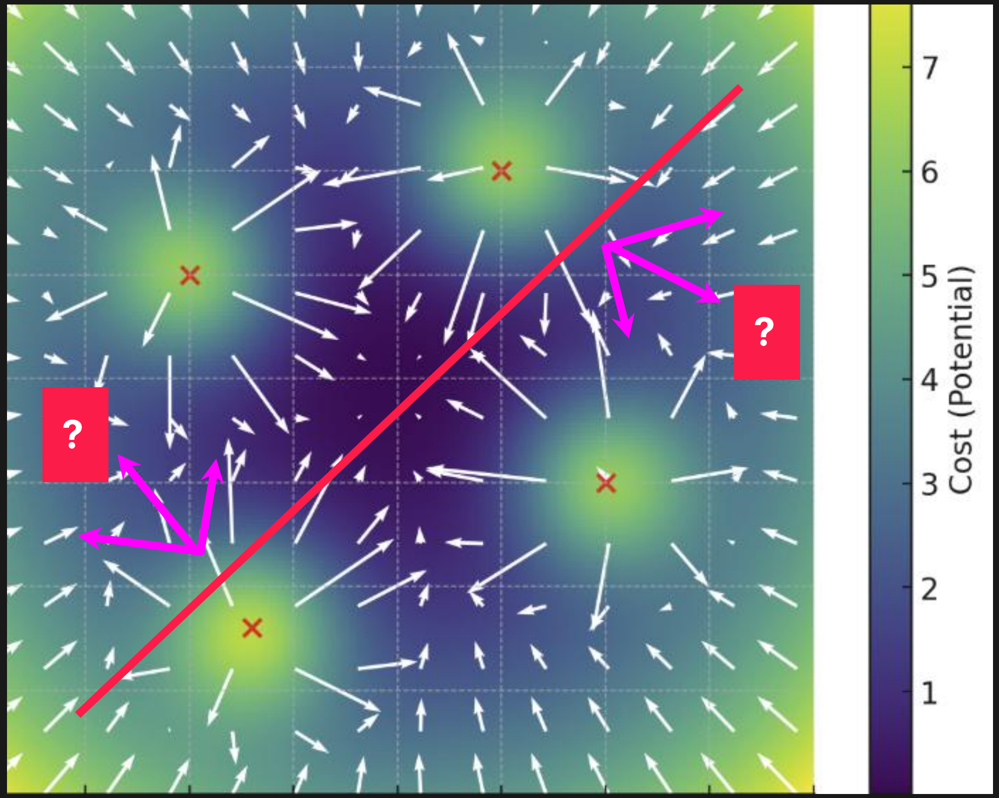
- Deterministic: 수식에 의해 명확하게 업데이트 방향이 결정이 됨



STOMP

경사 하강법: Deterministic vs. Stochastic

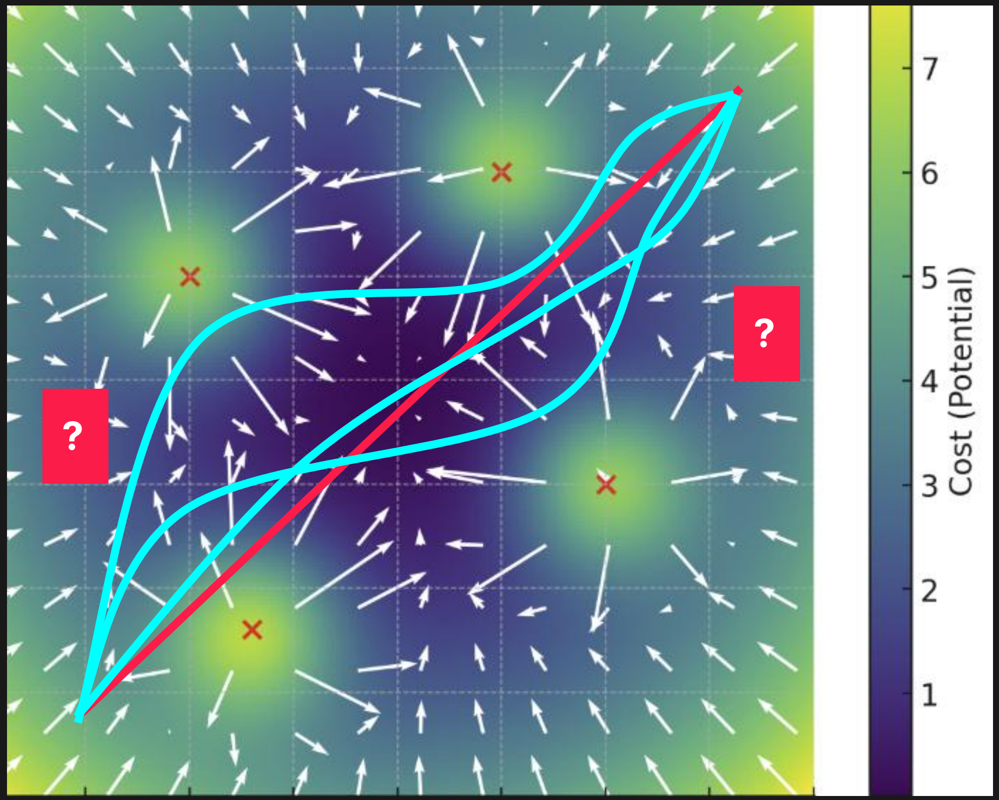
- Deterministic: 수식에 의해 명확하게 업데이트 방향이 결정이 됨
- Stochastic: 핵심은 샘플링을 통해 경사하강을 진행한다는 점



STOMP

경사 하강법: Deterministic vs. Stochastic

- Deterministic: 수식에 의해 명확하게 업데이트 방향이 결정이 됨
- Stochastic: 핵심은 샘플링을 통해 경사하강을 진행한다는 점

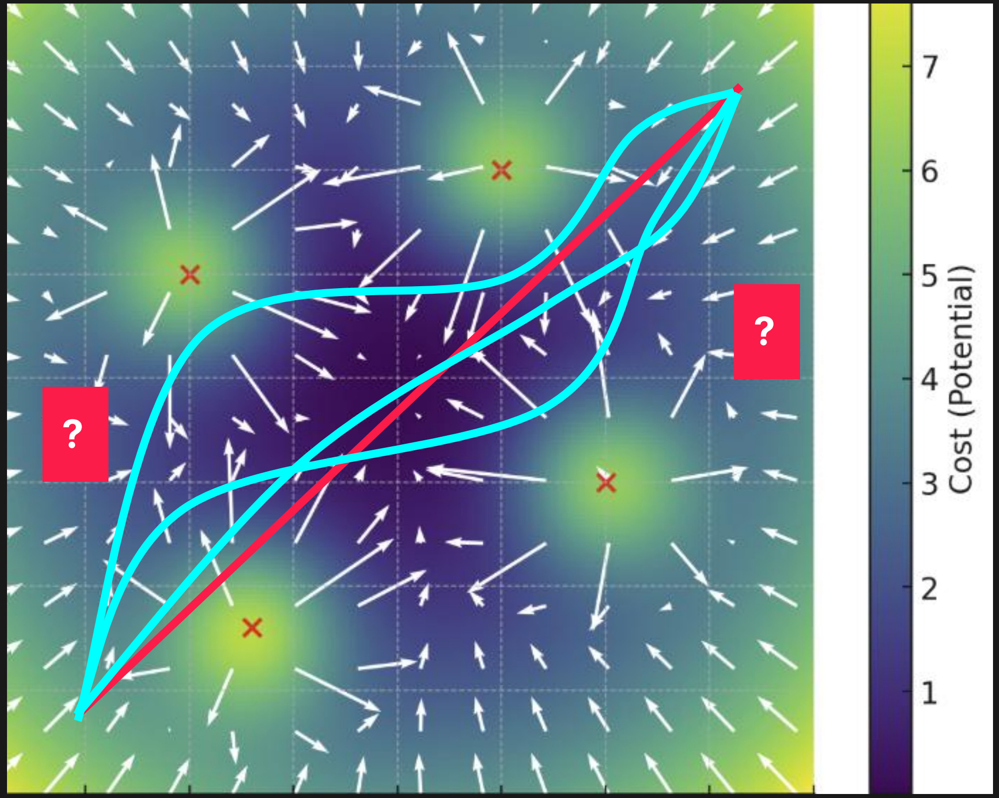


STOMP

경사 하강법: Deterministic vs. Stochastic

- Deterministic: 수식에 의해 명확하게 업데이트 방향이 결정이 됨
- Stochastic: 핵심은 샘플링을 통해 경사하강을 진행한다는 점

“여러 개의 노이즈를 trajectory에 더해보고, 그 중 성능이 좋은 (cost가 낮은) trajectory 쪽으로 이동한다”

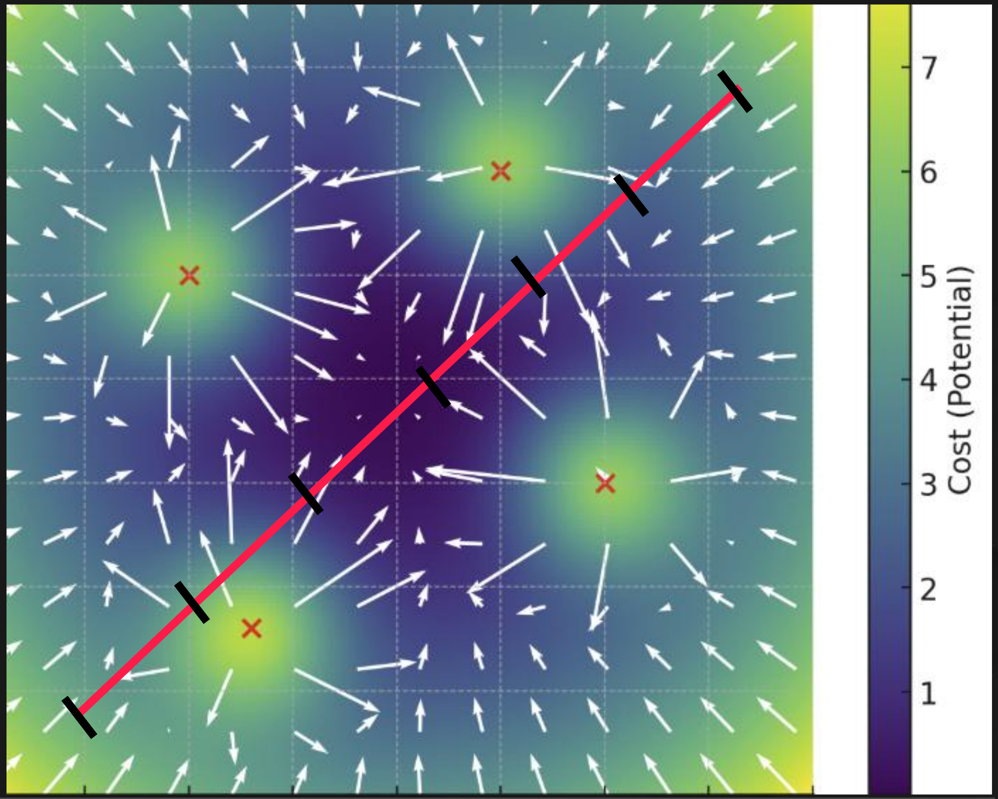


STOMP

"여러 개의 노이즈를 trajectory에 더해보고, 그 중 성능이 좋은 (cost가 낮은) trajectory 쪽으로 이동한다"

초기 trajectory $\xi = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T]$

$$\xi_k = \xi + \epsilon_k \quad \text{where } \epsilon_k \sim \mathcal{N}(0, \Sigma)$$

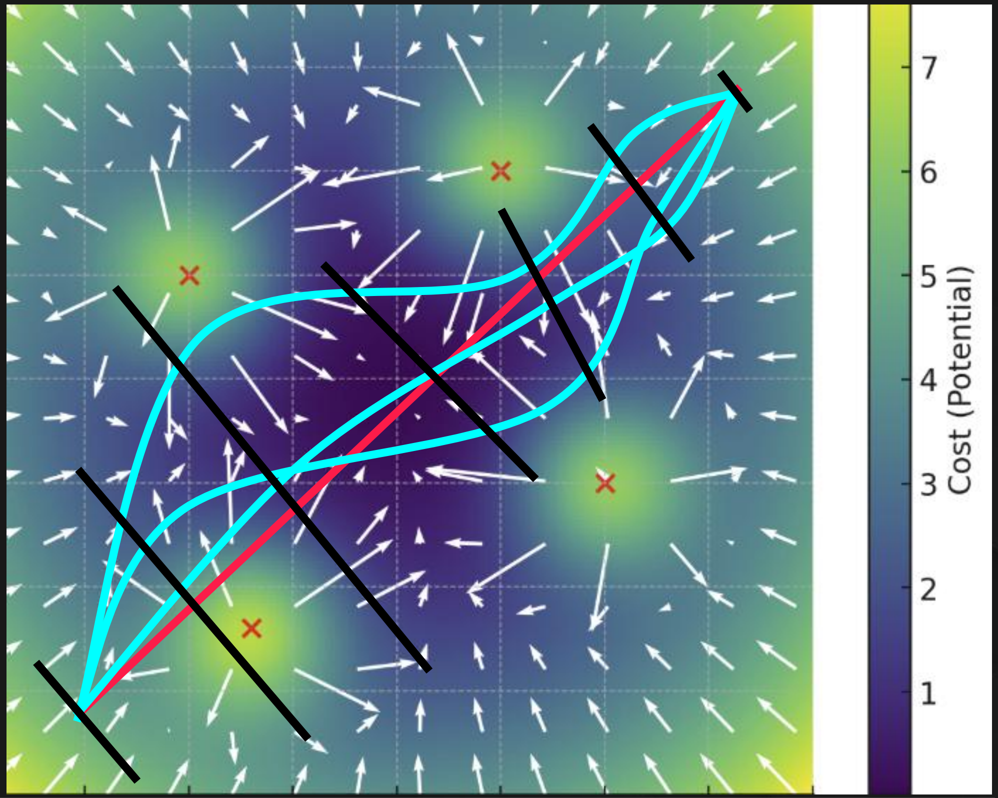


STOMP

"여러 개의 노이즈를 trajectory에 더해보고, 그 중 성능이 좋은 (cost가 낮은) trajectory 쪽으로 이동한다"

초기 trajectory $\xi = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_T]$

$$\xi_k = \xi + \epsilon_k \quad \text{where } \epsilon_k \sim \mathcal{N}(0, \Sigma)$$

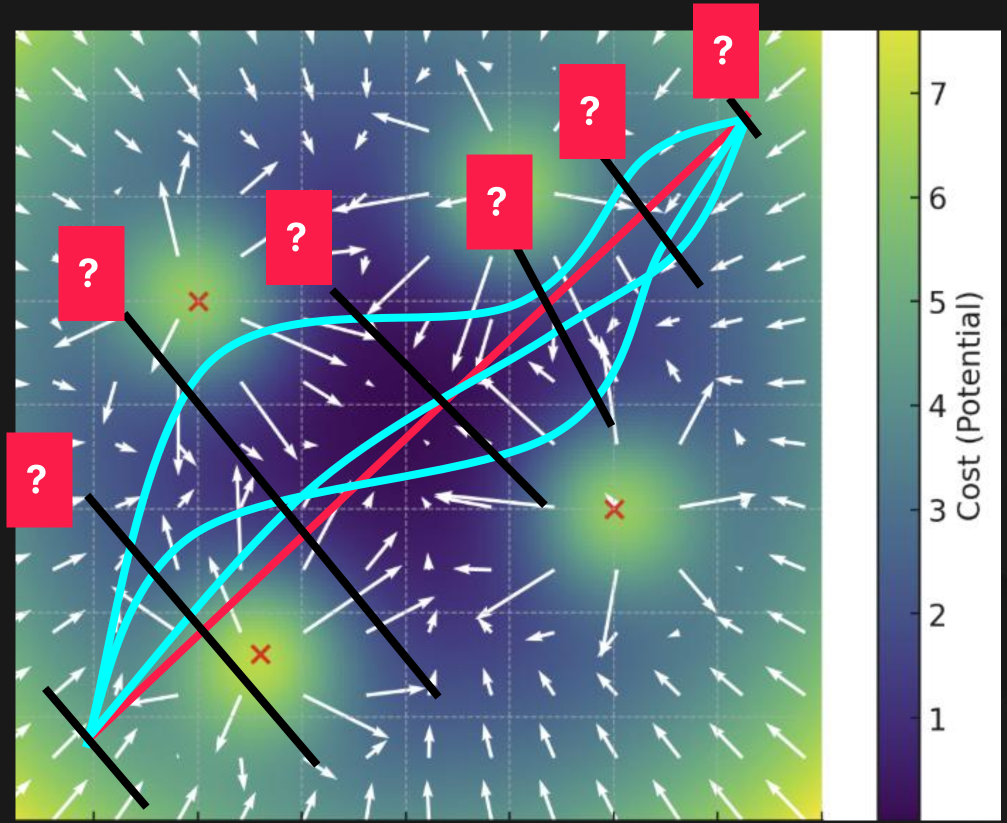


STOMP

"여러 개의 노이즈를 trajectory에 더해보고, 그 중 성능이 좋은 (cost가 낮은) trajectory 쪽으로 이동한다"

초기 trajectory $\xi = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T]$

$$\xi_k = \xi + \epsilon_k \quad \text{where } \epsilon_k \sim \mathcal{N}(0, \Sigma)$$

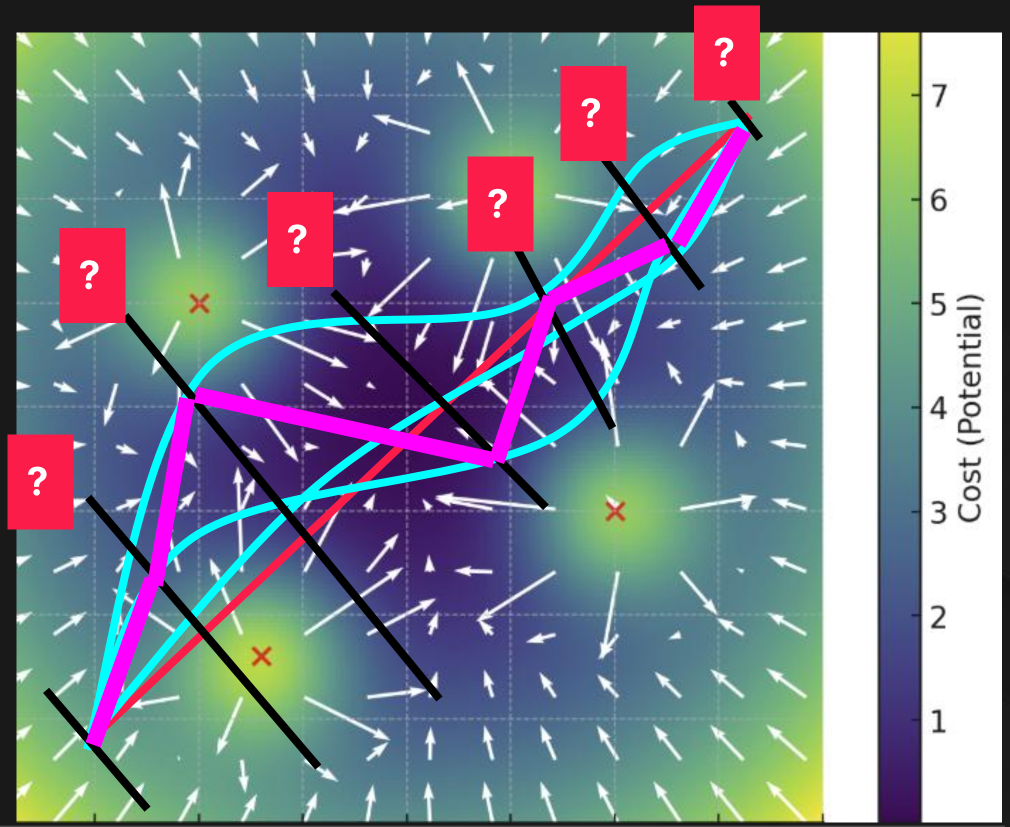


STOMP

"여러 개의 노이즈를 trajectory에 더해보고, 그 중 성능이 좋은 (cost가 낮은) trajectory 쪽으로 이동한다"

초기 trajectory $\xi = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_T]$

$$\xi_k = \xi + \epsilon_k \quad \text{where } \epsilon_k \sim \mathcal{N}(0, \Sigma)$$

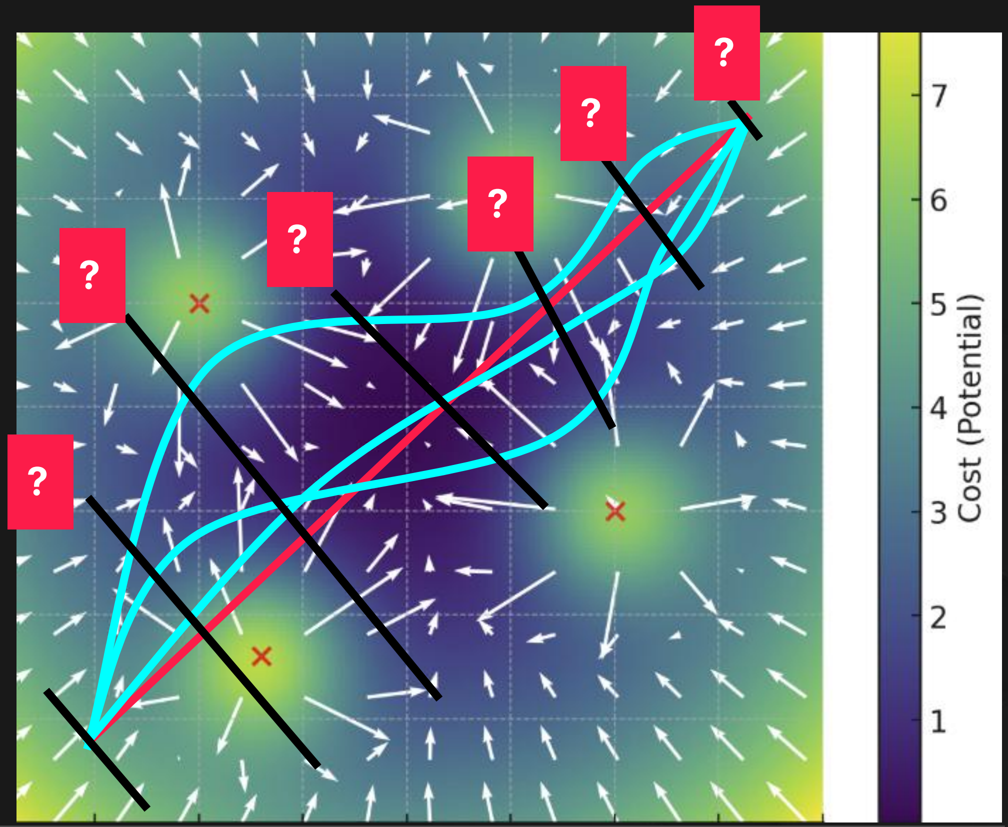


STOMP

"여러 개의 노이즈를 trajectory에 더해보고, 그 중 성능이 좋은 (cost가 낮은) trajectory 쪽으로 이동한다"

초기 trajectory $\xi = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T]$

$$\xi_k = \xi + \epsilon_k \quad \text{where } \epsilon_k \sim \mathcal{N}(0, \Sigma)$$



STOMP

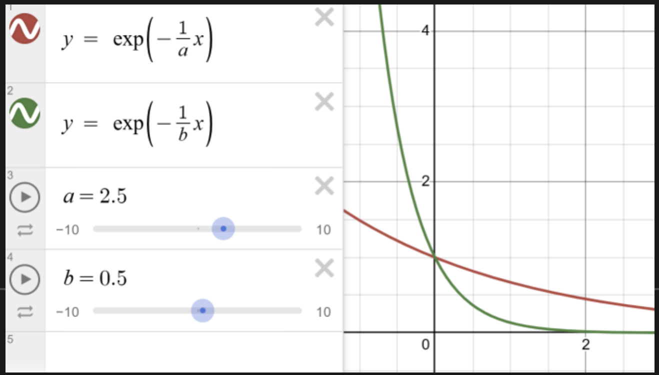
"여러 개의 노이즈를 trajectory에 더해보고, 그 중 성능이 좋은 (cost가 낮은) trajectory 쪽으로 이동한다"

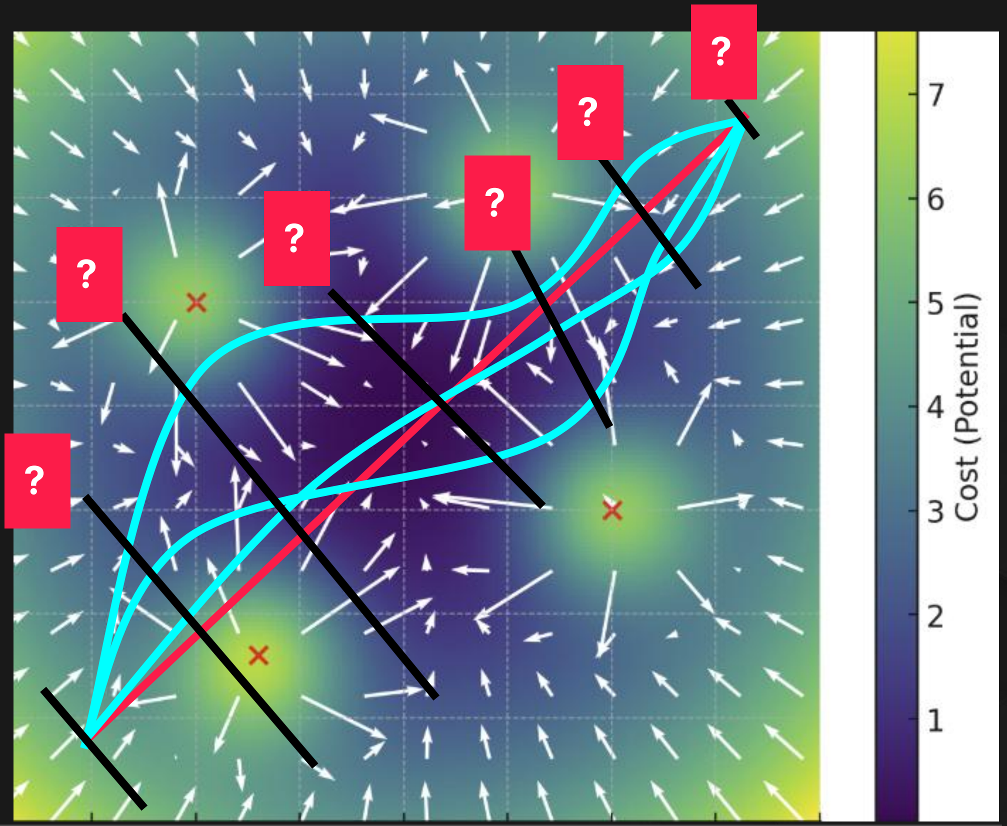
초기 trajectory $\xi = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T]$

$$\xi_k = \xi + \epsilon_k \quad \text{where } \epsilon_k \sim \mathcal{N}(0, \Sigma)$$

$$w_t^{(k)} = \frac{\exp\left(-\frac{1}{\lambda} c_t^{(k)}\right)}{\sum_{k=1}^{K} \exp\left(-\frac{1}{\lambda} c_t^{(k)}\right)}$$

각 trajectory 마다,
timestep 마다 계산





STOMP

"여러 개의 노이즈를 trajectory에 더해보고, 그 중 성능이 좋은 (cost가 낮은) trajectory 쪽으로 이동한다"

초기 trajectory $\xi = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T]$

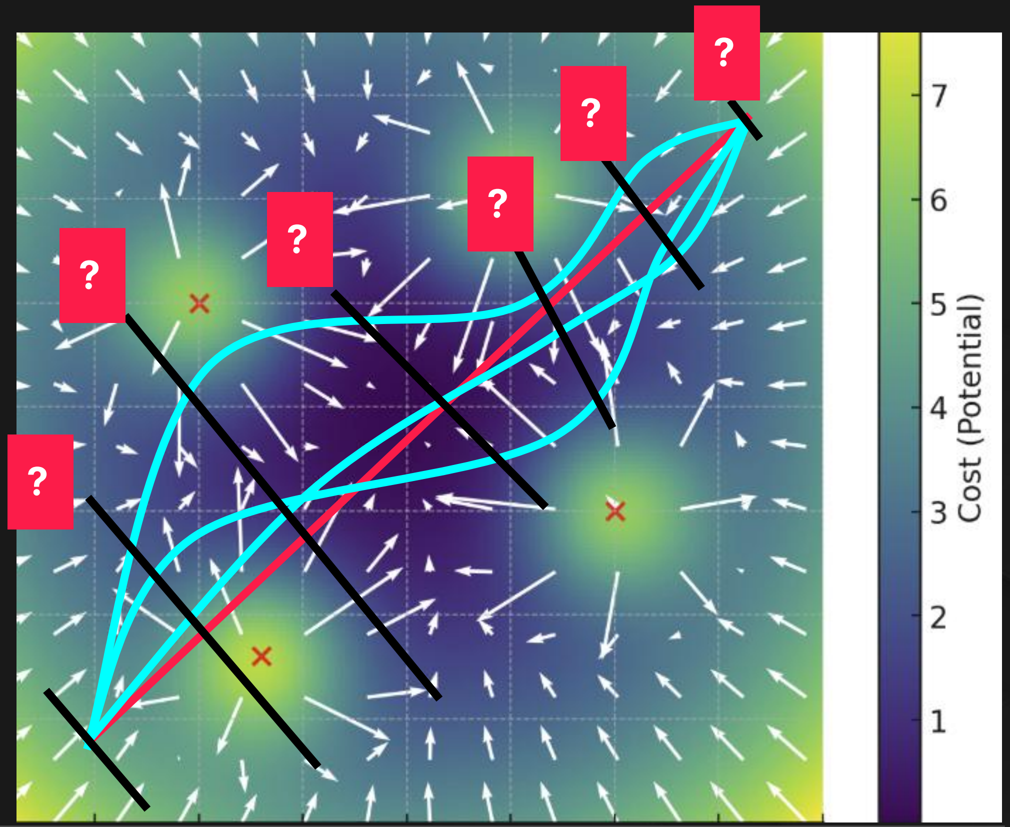
$$\xi_k = \xi + \epsilon_k \quad \text{where } \epsilon_k \sim \mathcal{N}(0, \Sigma)$$

$$w_t^{(k)} = \frac{\exp\left(-\frac{1}{\lambda} c_t^{(k)}\right)}{\sum_{k=1}^{K} \exp\left(-\frac{1}{\lambda} c_t^{(k)}\right)}$$

각 trajectory 마다, timestep 마다 계산

$$\xi \leftarrow \xi + \sum_{k=1}^{K} w^{(k)} \epsilon_k$$

노이즈가 기여한 만큼 업데이트



STOMP

"여러 개의 노이즈를 trajectory에 더해보고, 그 중 성능이 좋은 (cost가 낮은) trajectory 쪽으로 이동한다"

반복!

초기 trajectory $\xi = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T]$

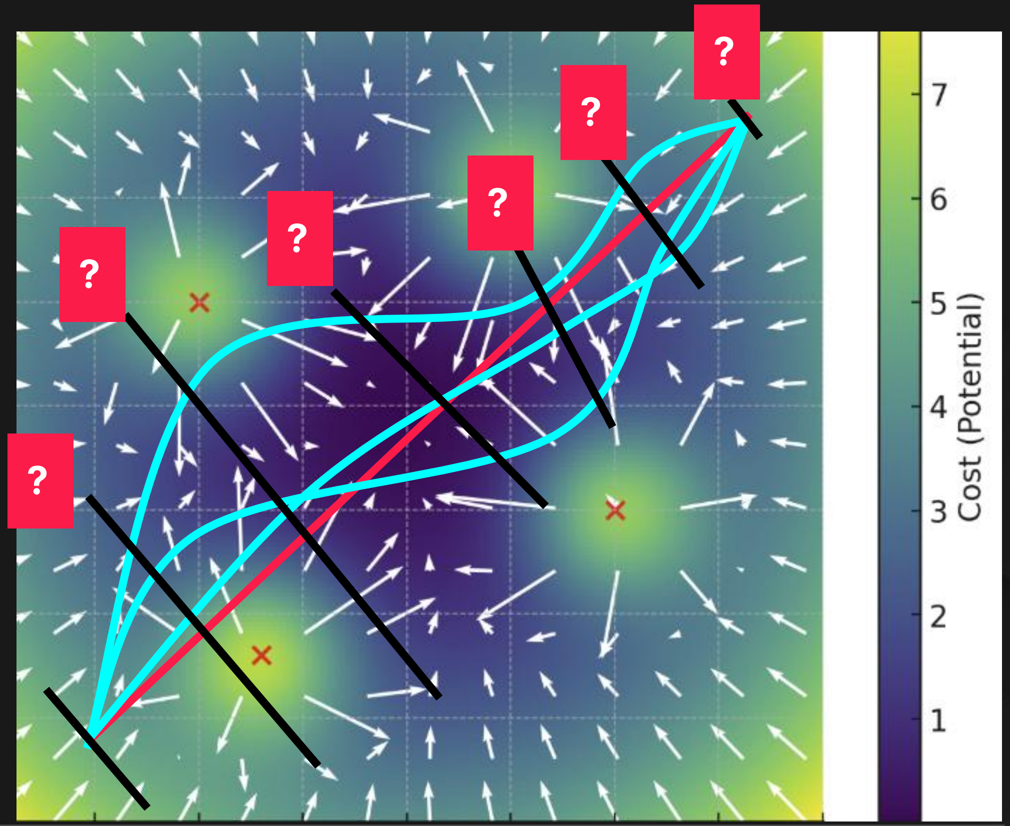
$$\xi_k = \xi + \epsilon_k \quad \text{where } \epsilon_k \sim \mathcal{N}(0, \Sigma)$$

$$w_t^{(k)} = \frac{\exp\left(-\frac{1}{\lambda} c_t^{(k)}\right)}{\sum_{k=1}^{K} \exp\left(-\frac{1}{\lambda} c_t^{(k)}\right)}$$

각 trajectory 마다, timestep 마다 계산

$$\xi \leftarrow \xi + \sum_{k=1}^{K} w^{(k)} \epsilon_k$$

노이즈가 기여한 만큼 업데이트



STOMP

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

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)

$$c(\xi) = w_{\text{obs}} \cdot c_{\text{obs}} + w_{\text{smooth}} \cdot c_{\text{smooth}} + w_{\text{vel}} \cdot c_{\text{vel}}$$

- 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

초기 trajectory $\xi = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T]$

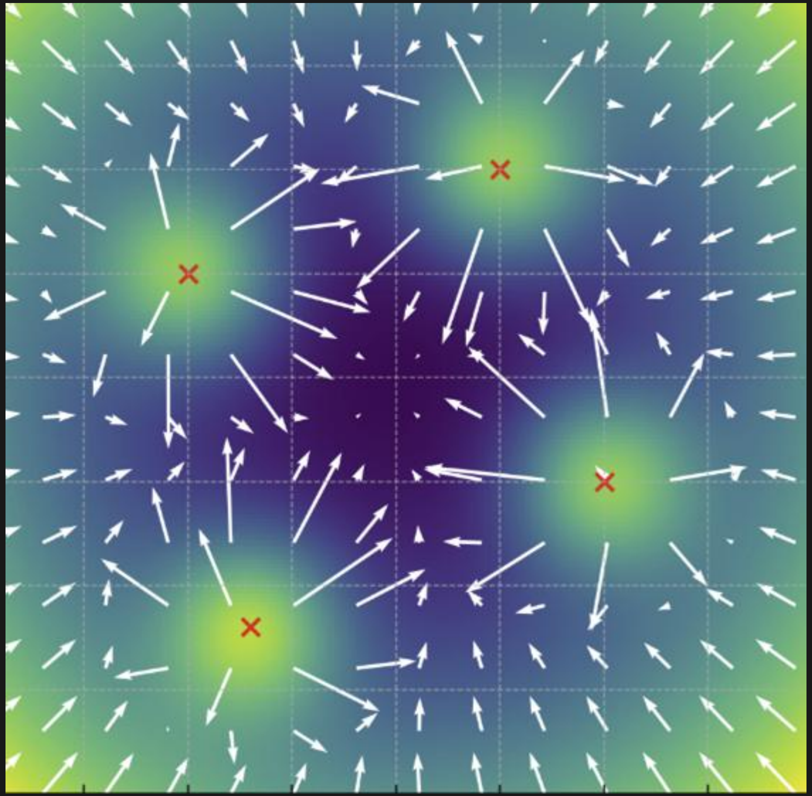
$$\xi_k = \xi + \epsilon_k \quad \text{where } \epsilon_k \sim \mathcal{N}(0, \Sigma)$$

$$w_t^{(k)} = \frac{\exp\left(-\frac{1}{\lambda} c_t^{(k)}\right)}{\sum_{k=1}^{K} \exp\left(-\frac{1}{\lambda} c_t^{(k)}\right)}$$

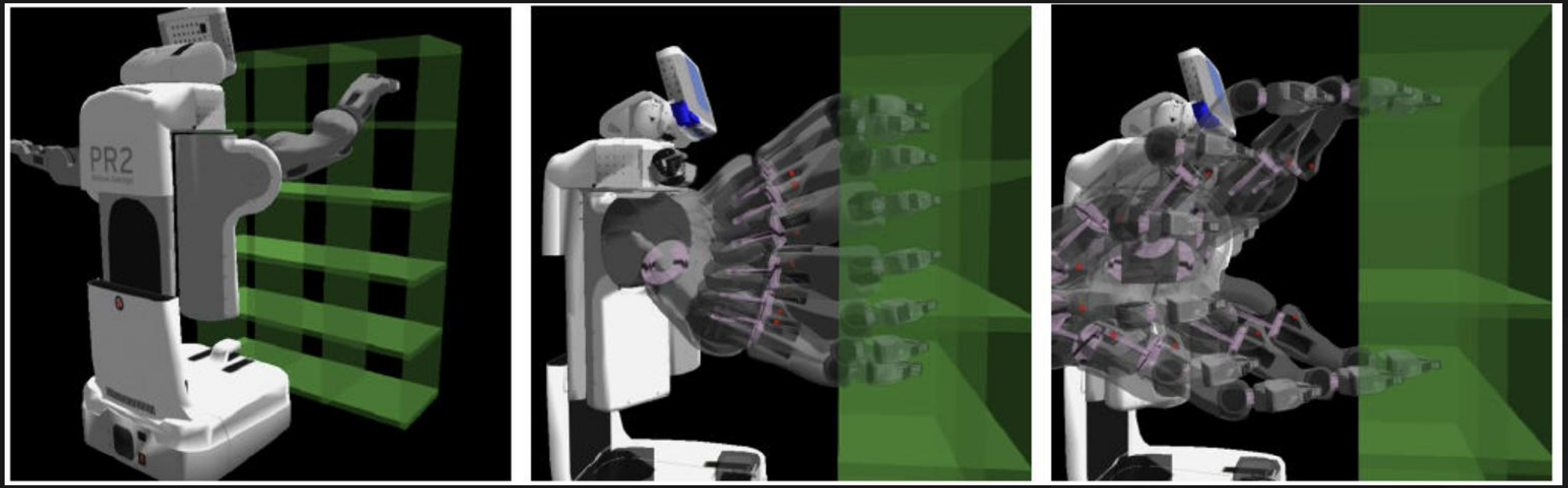
$$\xi \leftarrow \xi + \sum_{k=1}^{K} w^{(k)} \epsilon_k$$

CHOMP

C-Space (Joint Space)



Work Space (Task Space)



Mrinal et al. STOMP: Stochastic Trajectory Optimization for Motion Planning

CHOMP vs. STOMP

CHOMP

- Deterministic
- 미분가능 여부: 필수
- 계산 속도: 비교적 빠름
- 안정성: 민감함
  (local minimum, initialization)
- 활용: 비교적 부드러운 환경

최적화 기반
모션 플래닝

STOMP

- Stochastic
- 미분가능 여부: 불필요
- 계산 속도: 샘플 개수에 따라 다름
- 안정성: 비교적 robust
  (exploration)
- 활용: 복잡한 환경, 비선형 목적함수

강의 요약

01

문제 정의

- 미분 불가능한 목적함수 사용 가능 (자유도가 높다)

02

Gradient Descent

- 노이즈를 추가한 초기 경로에 가중치를 부여하여 업데이트 (Stochastic)

03

직관적 이해

- 초기값을 당기고 펴는 과정
- 조금씩 찾아가는 과정