名古屋大学
NAGOYA UNIVERSITY

SUZUKI LAB.

2023/08/04

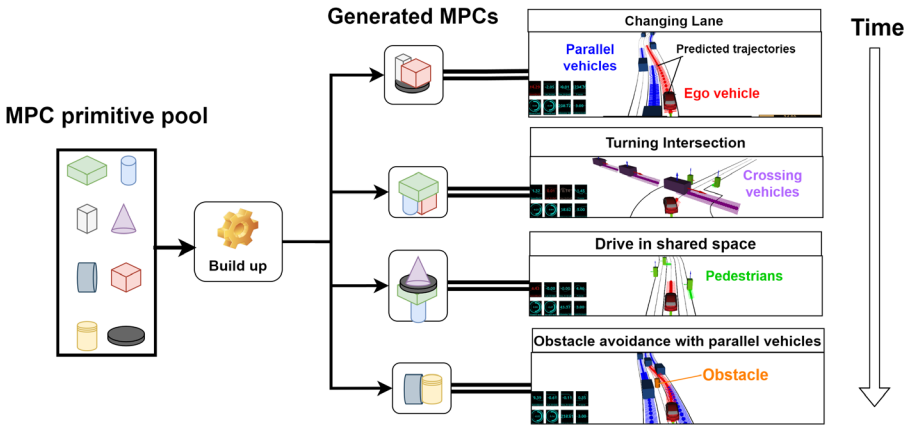
第3回 自律移動知能実装研究会

1/10スケール自動運転レース:
活動と手法解説

本田康平 / Kohei Honda
名古屋大学 鈴木研究室 D3

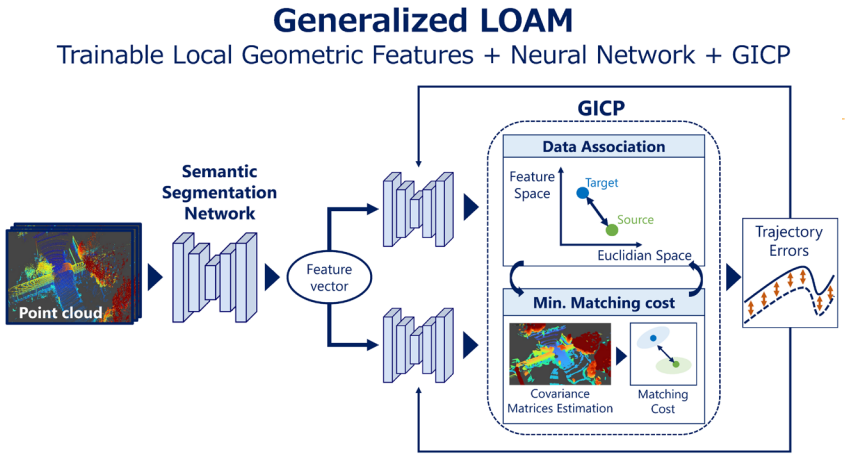
本田康平 / Kohei Honda

- **所属:** D3, モビリティシステムグループ, 名古屋大学
- **Research Interest**: 自律移動ロボットの知能化
- **Recent Projects:**
  1. マルチタスク型モデル予測制御 (Ph.D. thesis)
  2. アルゴリズムベースSLAM x 機械学習
  3. ロボットナビゲーション x 強化学習



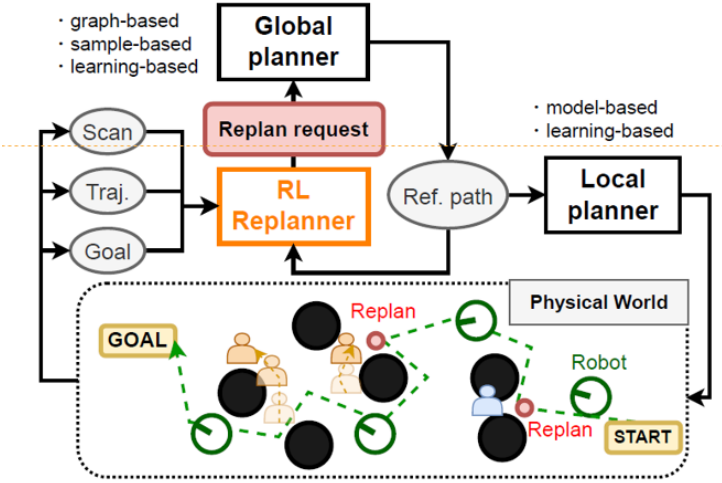
Multi-task MPC for AD

[ITSC '20][AJC '22] [IV '23]



SLAM x NN

[RA-L with ICRA '23]

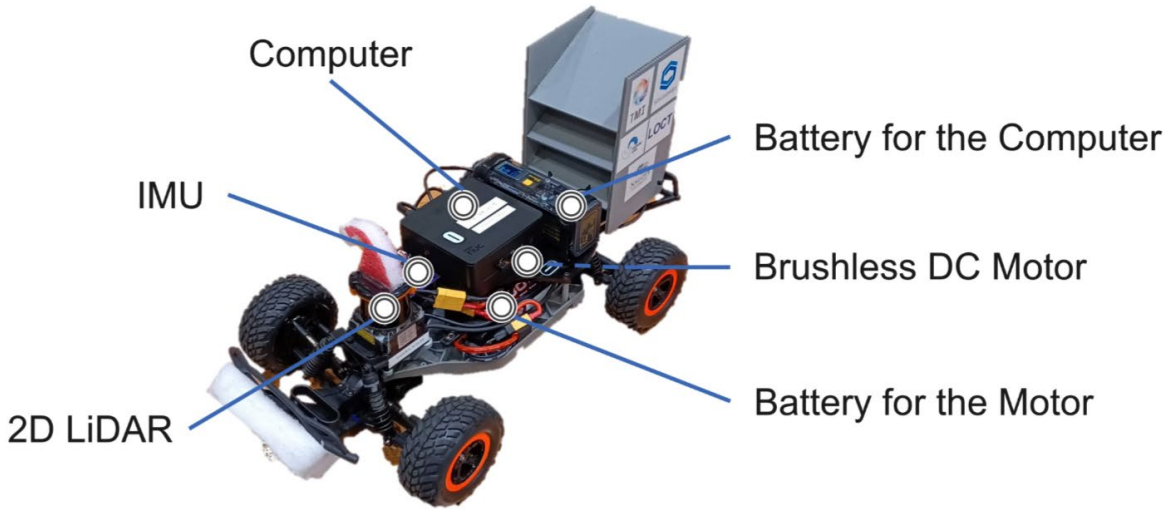


Navigation x RL

[Preprint]

内容

1. 1/10ラジコンカー自動運転レースF1tenthの概要
2. 実装した自律移動システムについて
3. 確率的MPCのすゝめ



第1部:

自動運転レースF1tenthについて

F1tenth: 1/10スケールミニカーの自動運転レース大会

主催

F1TENTH Foundation

[1]

[1]

- ミニカー作成に必要な市販パーツ, 組み立て方,
  基本的なシミュレータなどをオープンソースで公開
- 研究者同士の交流を促進し, 優れた研究成果を生み出す狙いがある
  トップチームは開発したアルゴリズムを直接国際学会で発表

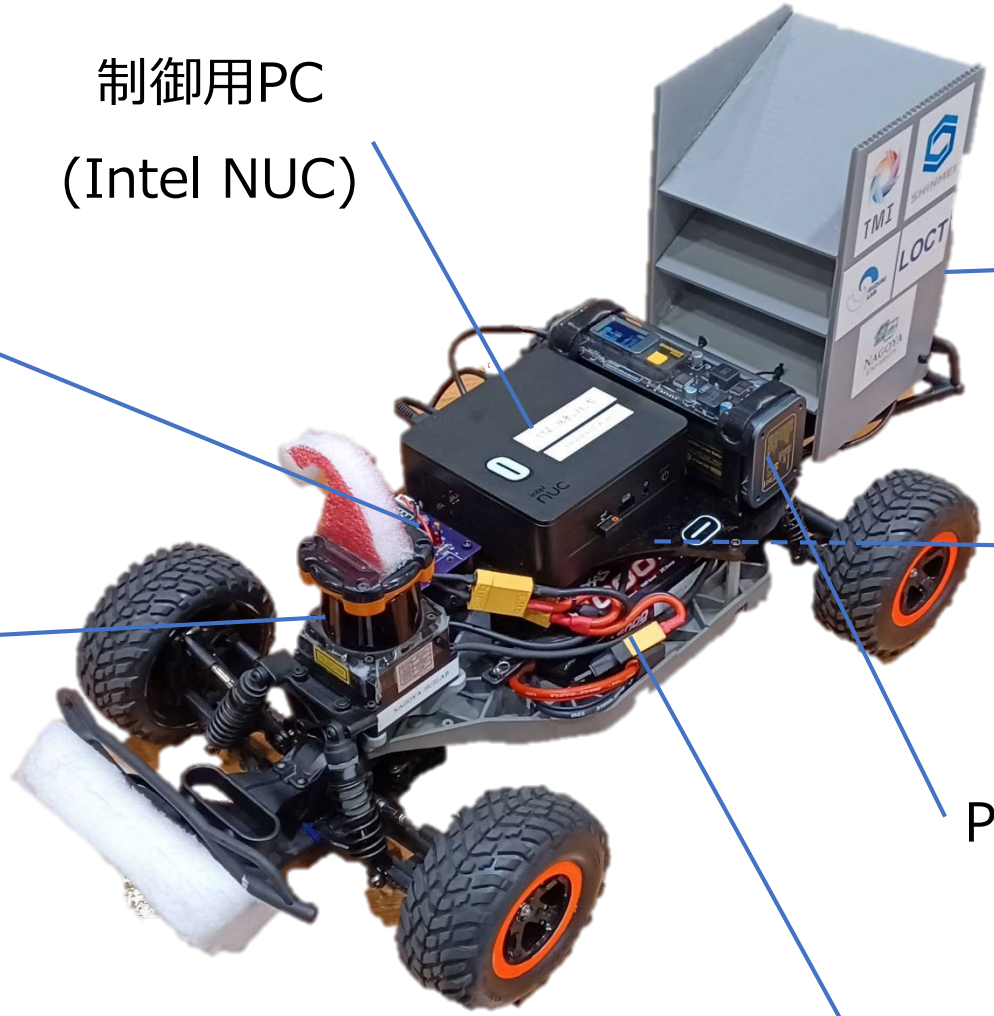
概要

- 年に3~4回程度, 自動運転系・ロボット系国際学会と同時に開催される.
  例: ICRA'23 (4日間): 練習・予選 (2日間) + 決勝 (1日間)
- ハードウェアにはある程度制限があり, アルゴリズムの戦いという側面が強い
  例：2DLiDARを1個まで, カメラは1個まで, など

[1] F1tenth org : https://f1tenth.org/build.html

ハードウエア構成

車体: Traxxas Slash (1/10スケール)



制御用PC

(Intel NUC)

9軸IMU

モータドライバ

2D LiDAR

(**HOKUYO**)

ボックス

(他車両LiDARからの検知用)

駆動モータ

DCブラシレス

PC用バッテリー

駆動モータ用バッテリー

総額: 30~50万円

競技の流れ

練習

- 本番と同じコースで車両を走らせながら, 各チーム自由に調整を実施.
- コースの地図を作り, 事前に走行経路を作成するチームも多い.

※ コース形状は本番で初めて分かる

▼

予選:
タイムアタック

- 一定時間走行で**「衝突なく走れた最大lap数」**と**「最速lap時間」**を記録.
- 予選順位が高いほど, 本選のトーナメント配置で有利になる.

▼

本選:
1on1レース

- トーナメント戦により**1対1のレース**を実施.
- 2チームの車両を対角から同時にスタートさせ, 先に8周した方が勝ち.

競技会の様子: セットアップ (マップ作製)

他車両の走行中にマップ作製をする必要あり．我々はマップレス走行で自動マップ作製

競技会の様子: 玉石混同の走行練習

コース変形・クラッシュ多数の無法地帯

競技会の様子: 予選 (タイムアタック)

最速チームは最大速度10 m/s程度まで出る

競技会の様子: 決勝 (1on1レース)

上位チーム同士では速度が拮抗しているので追い抜きが発生しないこともしばしば

成績: Team Suzlab

ICRA2023大会（ロンドン）

4位 / 22チーム

順位

[1] ForzaETH：ETH Zurich University (Switzerland)
[2] Scuderia Segfault：Vienna University (Austria)
[3] PUT-PPI：Poznan University of Technology (Poland)
[4] **Suzlab** : Nagoya University (Japan)

IV2023大会（アンカレッジ）

3位入賞 / 8チーム

順位

[1] Embedded Machine Learning Club :
North Carolina State University (USA)
[2] Clemson Tigers : Clemson University (USA)
[3] **Suzlab** : Nagoya University (Japan)

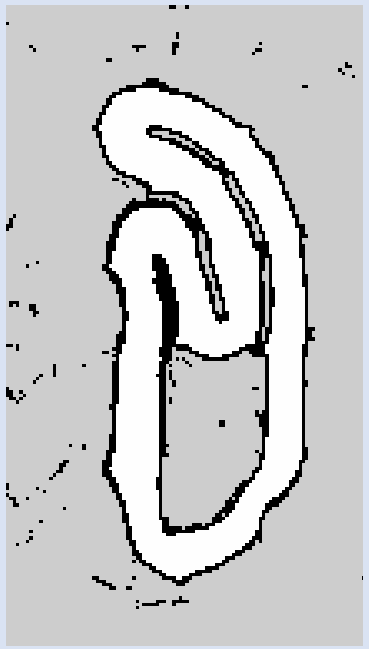
第2部:

チームSuzLab自律移動システム概要

ソフトウエア構成

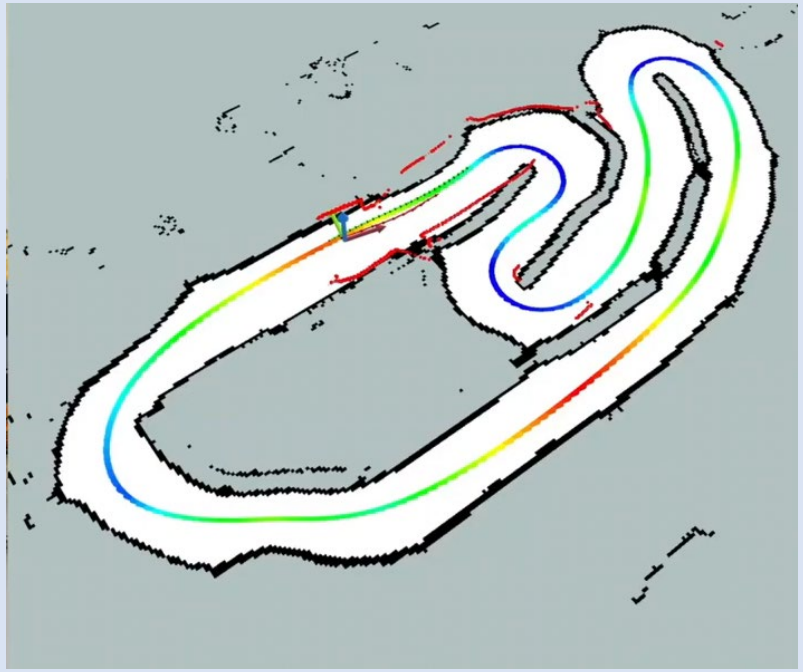
事前準備

地図作成 ▶ 参照軌道計画



リアルタイム処理

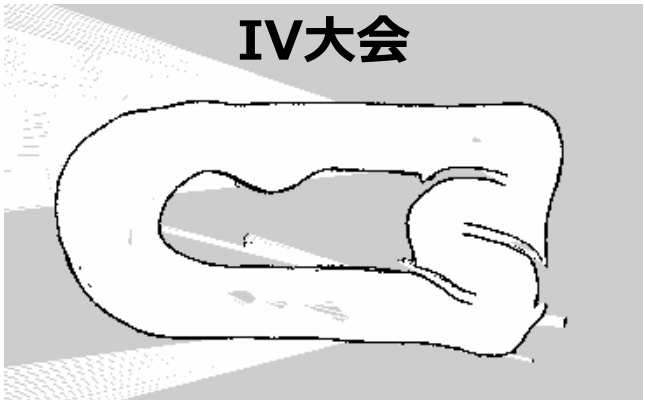
自己位置推定 ▶ 計画・制御



地図作成

地図作成 ▶ 参照軌道計画 自己位置推定 ▶ 計画・制御

- **gmapping** [1]を利用
  - ROS公式が提供する2D SLAMパッケージ
  - デフォルトパラメータで問題なく動作した



[1] gmapping : http://wiki.ros.org/gmapping

参照軌道 (経路・速度) 計画

地図作成 ▶ 参照軌道計画 | 自己位置推定 ▶ 計画・制御

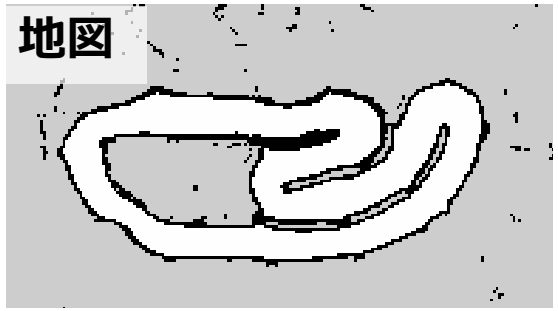
◆ **経路・速度を同時に最適化: アウト・イン・アウト走法**

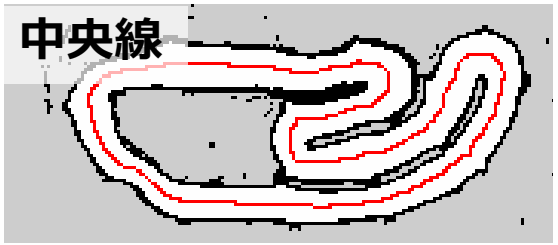
- 最適化ソルバー：IPOPT [1] (主双対内点法)
- 目的：ラップタイムの最小化

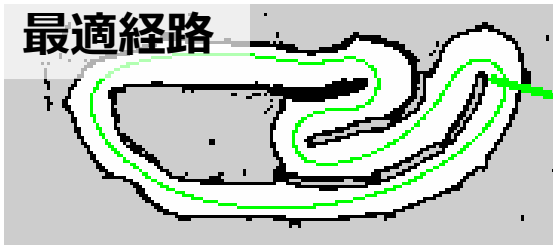
◆ **制約条件**

- 車両速度, 前後方向加速度, 左右方向加速度の制限

  (例) カーブで滑る場合 → 左右方向加速度を小さくする
- 壁面から一定距離を空けて走行





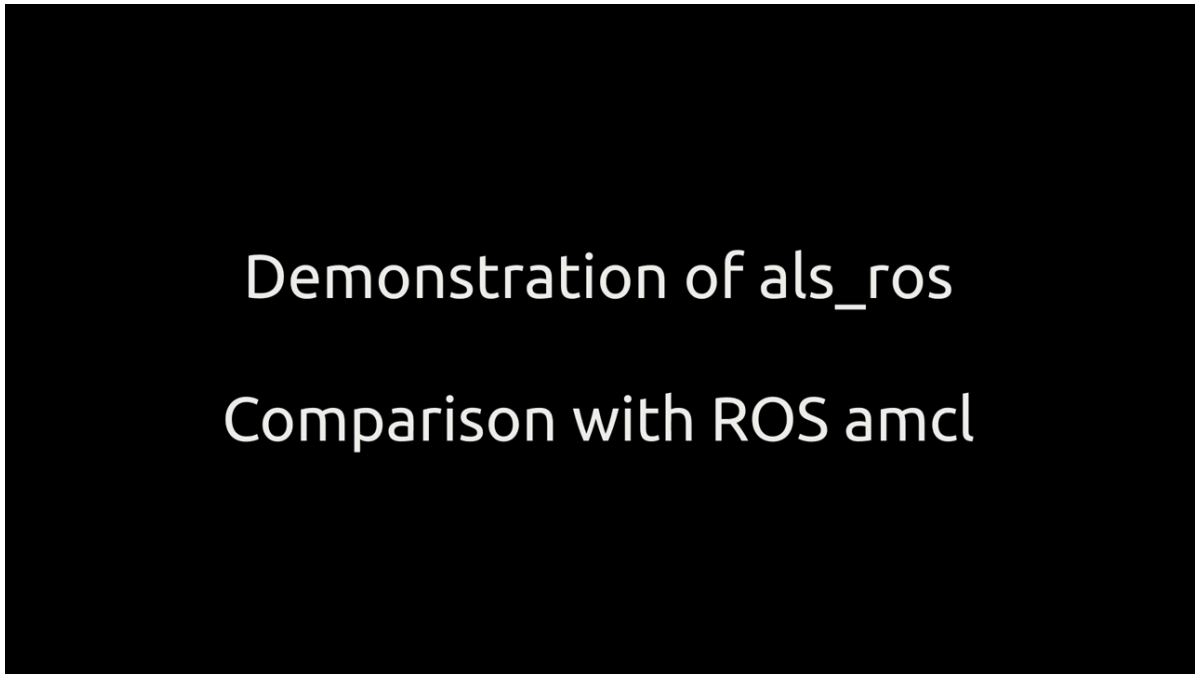


[1] IPOPT : https://coin-or.github.io/Ipopt/

自己位置推定

地図作成 ▶ 参照軌道計画 | 自己位置推定 ▶ 計画・制御

- **als_ros** [1]を利用
  - 赤井先生作 強化版AMCLパッケージ (OSS)
  - 最速8~10 m/sの速度域でも
    デフォルトパラメータで問題なく動作した
  - 推定を安定化する工夫が含まれている
    → マップの変形に強い印象
  - ROS公式 amcl [2] と比較しても,
    als_rosの方が安定して動作した



[3]

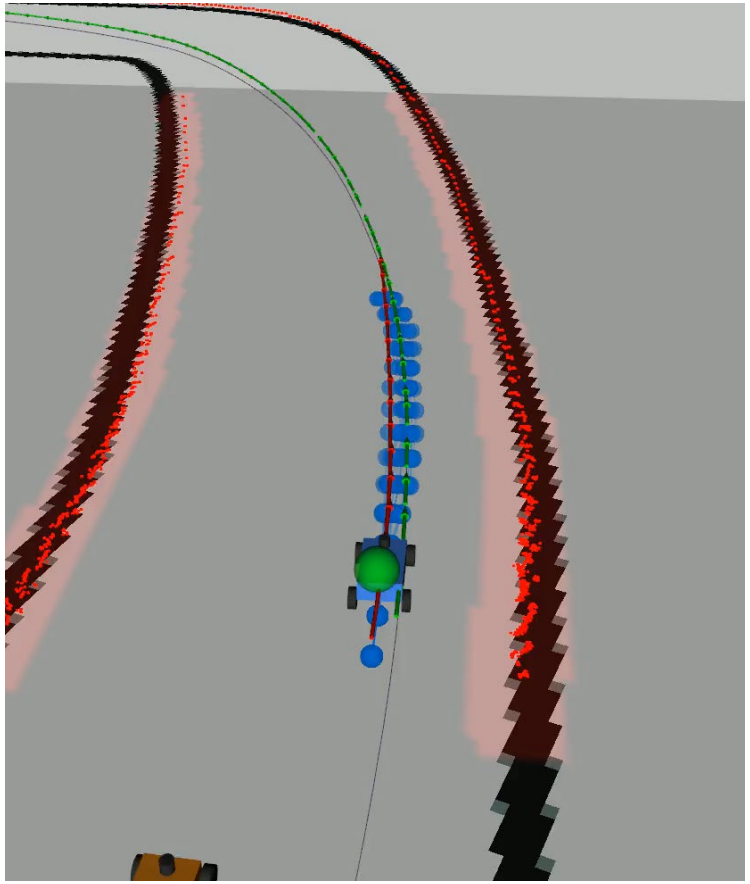
[1] als_ros : https://github.com/NaokiAkai/als_ros
[2] amcl : http://wiki.ros.org/amcl
[3] comparison between als_ros and ros amcl : https://www.youtube.com/watch?v=wsoXvUgJvWk

局所計画・制御

地図作成 ▶ 参照軌道計画 | 自己位置推定 ▶ 計画・制御

◆ **Model Predictive Path-Integral control (MPPI) [1]**を実装

- 車両の**未来状態を予測**し，
  実時間で計算した最適挙動をもとに制御入力を決定
- 障害物との衝突にペナルティをかけることで，
  対戦相手を追い越す挙動も実現可能



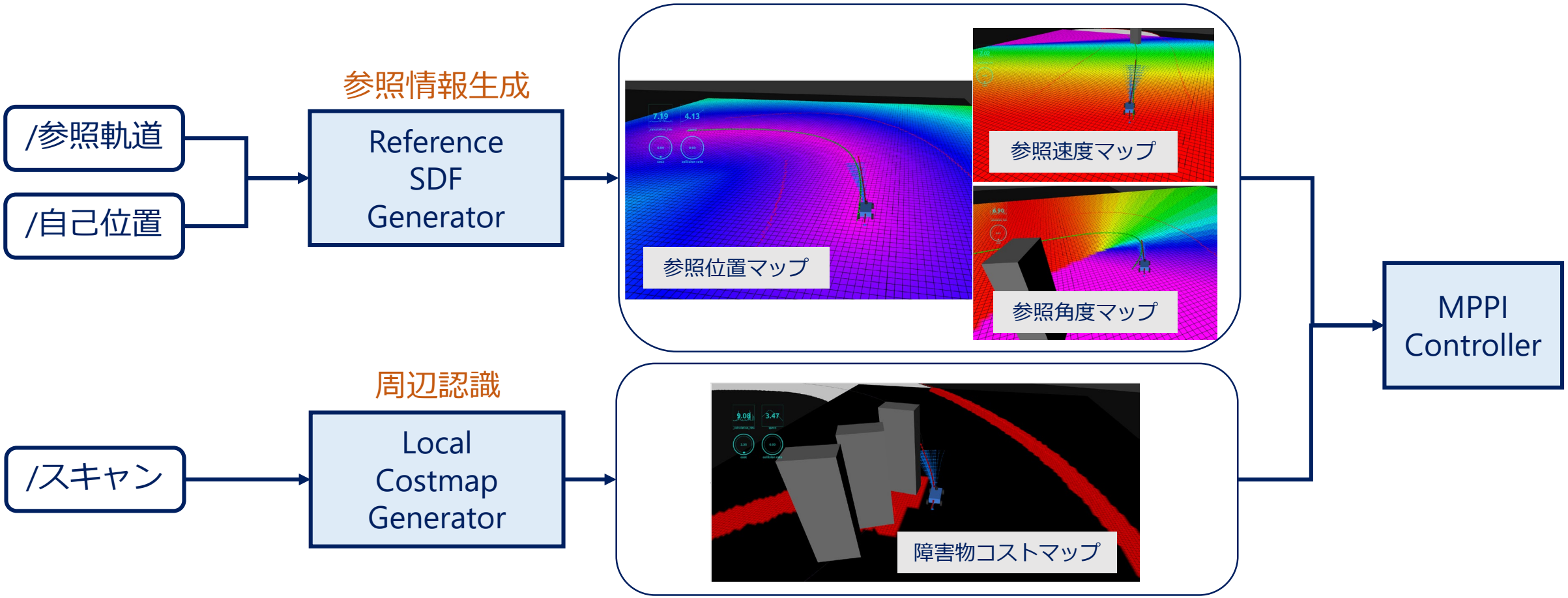
第3部にて詳細を説明

[1] G. Williams+, Information Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving, IEEE T-RO, 2018

局所計画・制御: 高速化のための工夫

地図作成 ▶ 参照軌道計画 | 自己位置推定 ▶ 計画・制御

★ MPPIの用いる外部情報はすべてグリッドマップ化 → 近傍探索高速化



知見・改良すべき点

- 車両制御の安定性自体は, 上位チームと遜色無かった
- 走行タイム短縮のためには車両の**加減速性能向上**が必須
  - 特にコースの直線部分において, 車両の加減速性能がタイムに直接現れる
  - **車両の軽量化・低重心化, モータ出力強化, ステアサーボ応答の高速化**, などが重要
- **コースの滑りやすさ**が時々刻々と変化するため, パラメータ調整が難しい
  - 傾向として, 時間が経つごとに滑りやすくなった
- **マップの形状は頻繁に変化**する
  - 他車両が頻繁に壁へ衝突するため
  - 練習中, 実際に自己位置推定機能が悪化 → マップを更新して改善

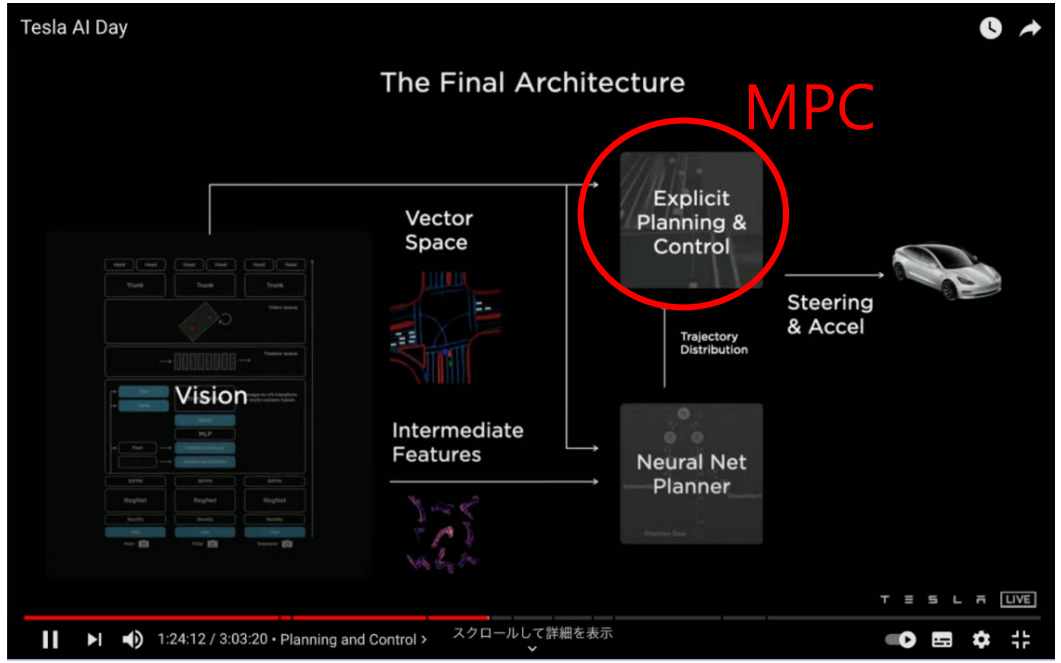
第3部:

確率的MPCのすゝめ

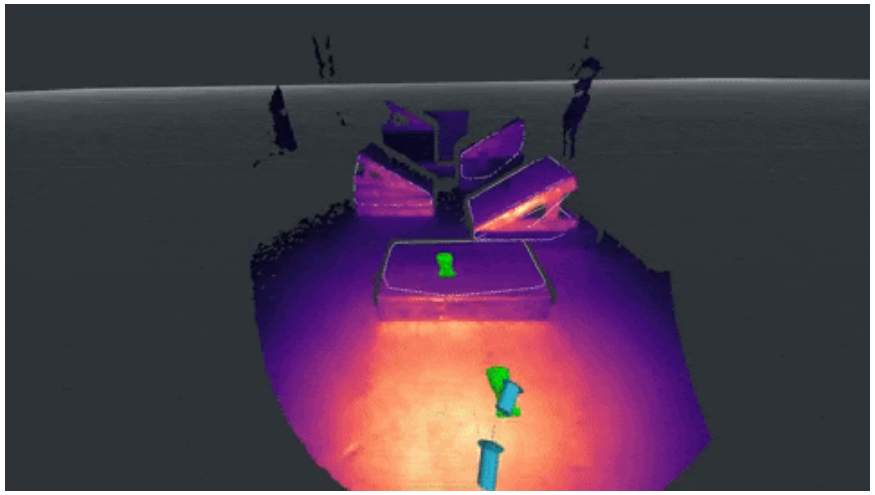
モビリティを支えるModel Predictive Control (MPC)

ロケットの自動着陸制御 (SpaceX)

自動運転のローカルプランニング (Tesla)



ヒューマノイドロボット(Atlas)の制御 (Boston Dynamics)

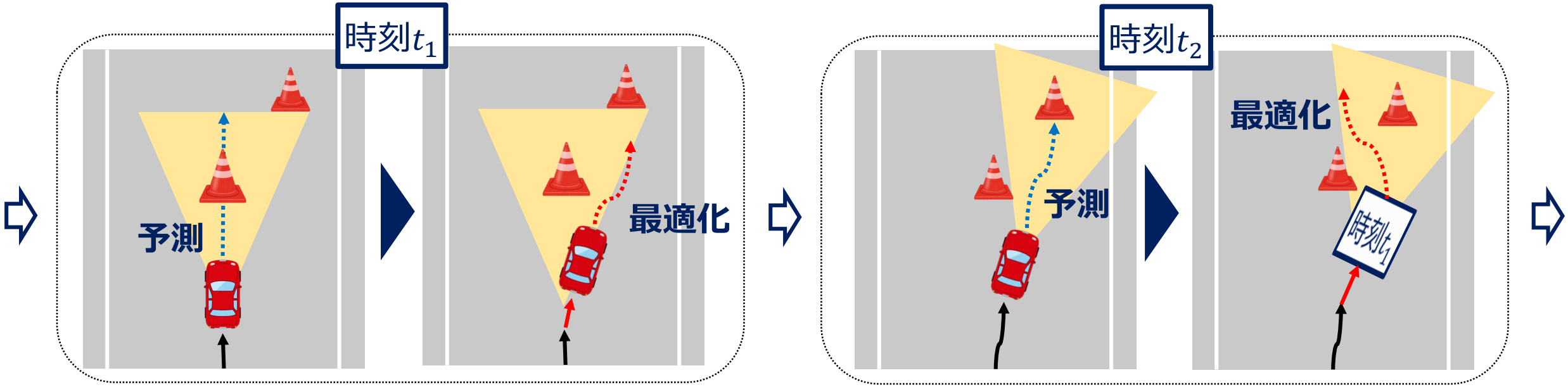


高精度・高速度・高自由度な制御

→ MPC!

MPCとは？

少し先の未来までの挙動をモデルを用いて**予測&実時間最適化**

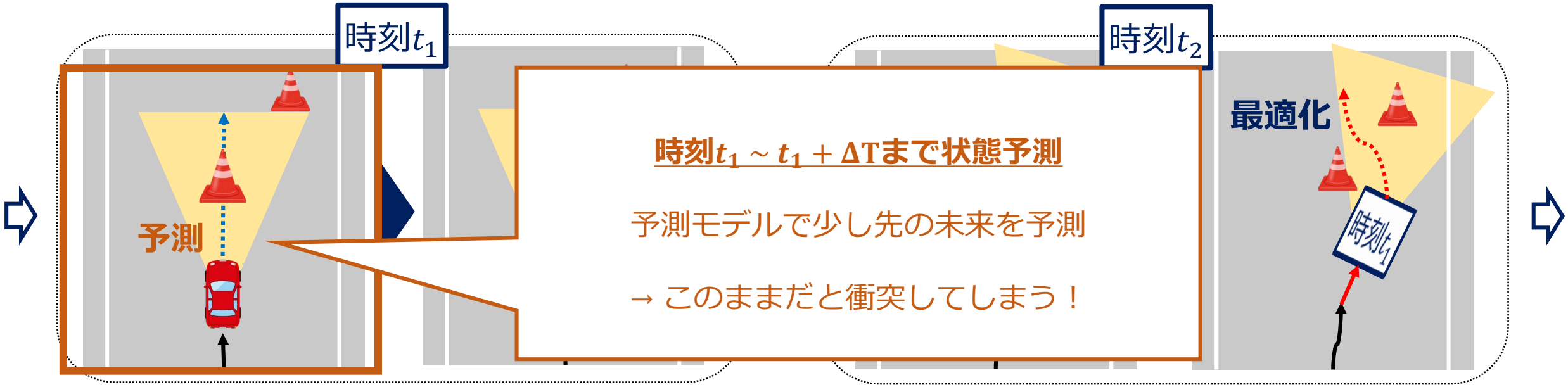


繰り返し

1. 状態の観測
2. 数理モデルで有限時間先まで予測
3. 最適化問題を解く
4. 最適制御入力を印加

MPCとは？

少し先の未来までの挙動をモデルを用いて**予測&実時間最適化**

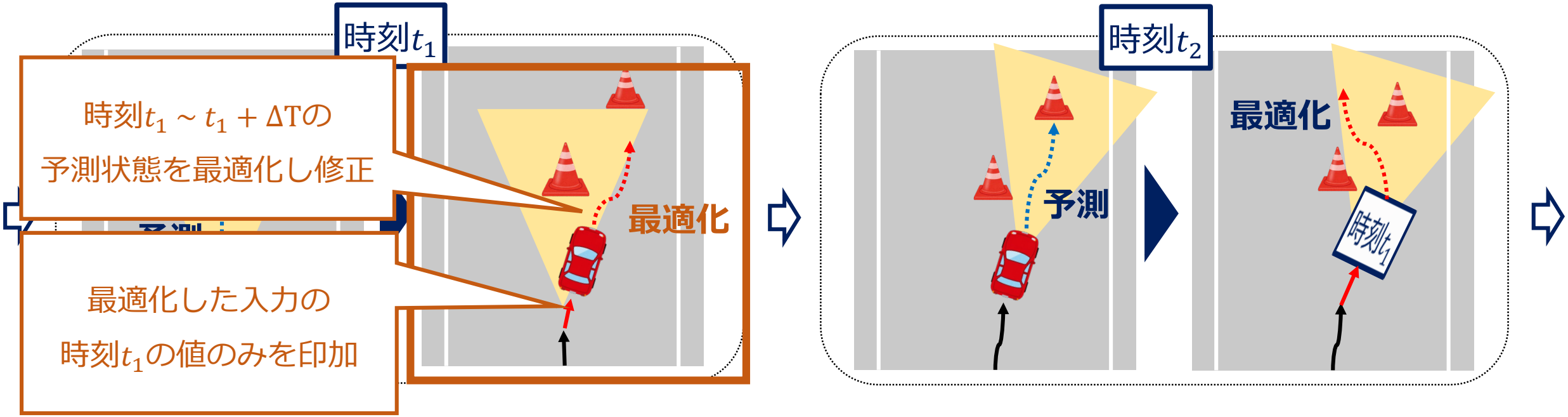


繰り返し

1. 状態の観測
2. 数理モデルで有限時間先まで予測
3. 最適化問題を解く
4. 最適制御入力を印加

MPCとは？

少し先の未来までの挙動をモデルを用いて**予測&実時間最適化**

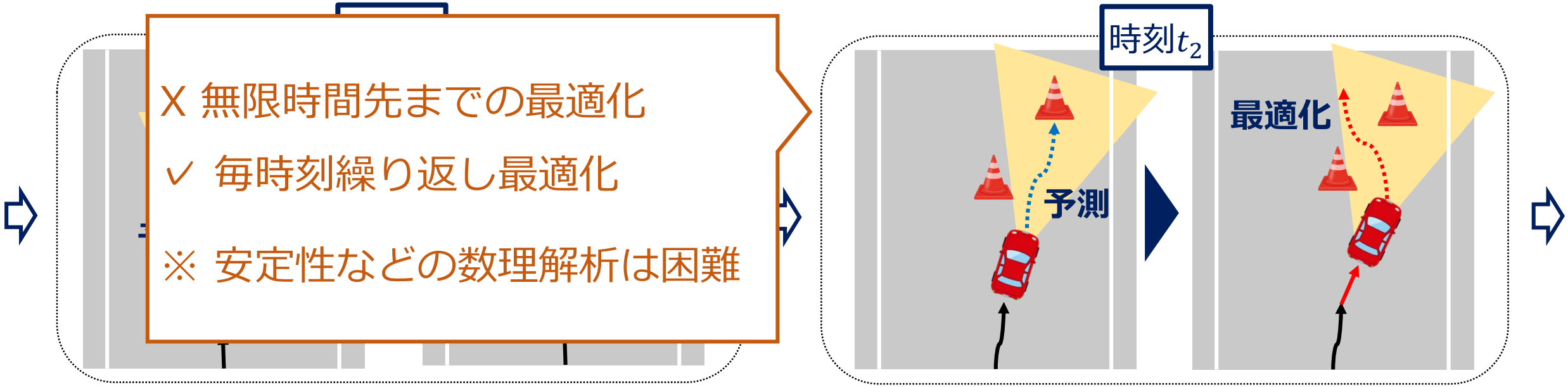


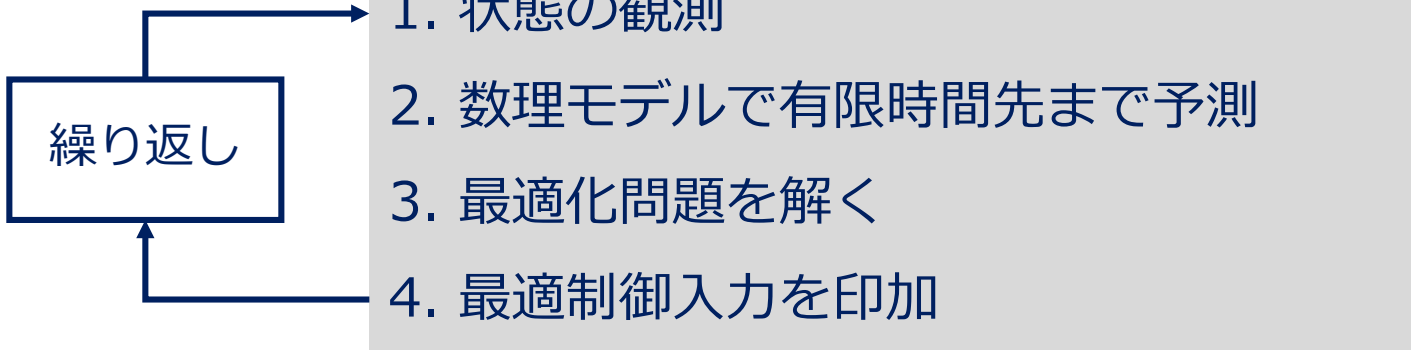
繰り返し

1. 状態の観測
2. 数理モデルで有限時間先まで予測
3. 最適化問題を解く
4. 最適制御入力を印加

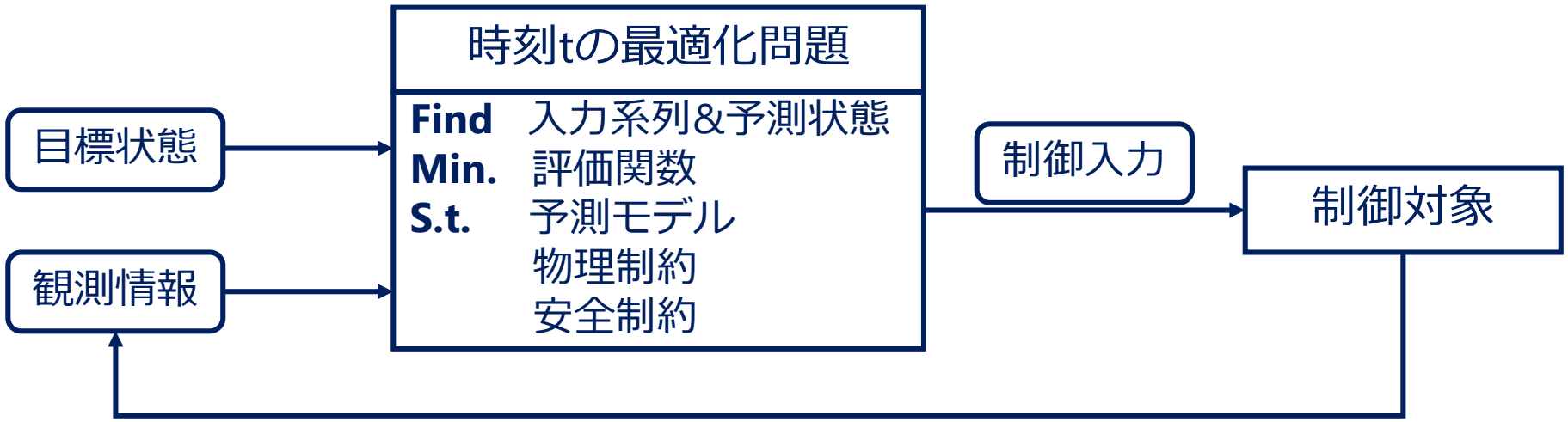
MPCとは？

少し先の未来までの挙動をモデルを用いて**予測&実時間最適化**





MPCの設計要素



◆ 【設計要素 1】：最適化問題をどうやって設計するか？

※ ”予測” = 満たすべき状態遷移条件 = 制約条件として扱う

今日の内容 ⇨ ◆ 【設計要素 2】：最適化計算をどうやって実時間で解くか

障害物回避最適化問題の雰囲気

(目的) ロボットが障害物にぶつからずにゴールする

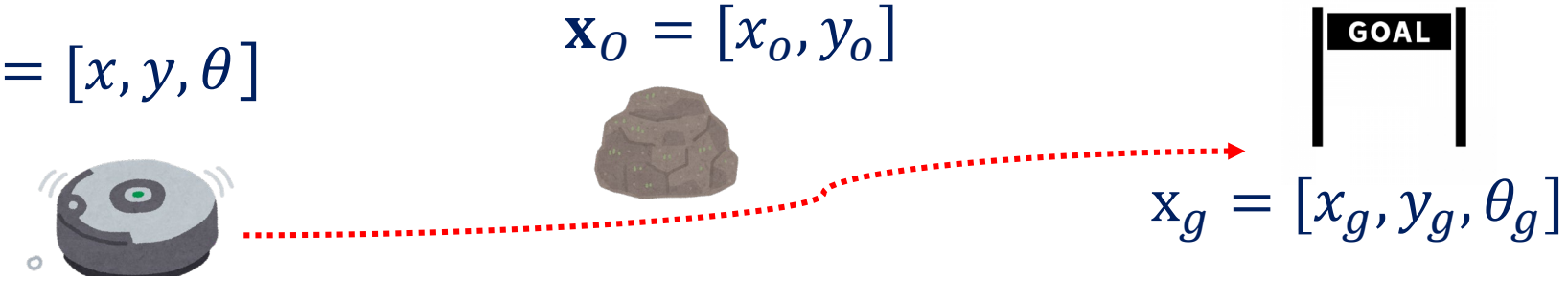
$\mathbf{x} = [x, y, \theta]$

$\mathbf{x}_O = [x_O, y_O]$

$\mathbf{x}_g = [x_g, y_g, \theta_g]$

障害物回避最適化問題の雰囲気

(目的) ロボットが障害物にぶつからずにゴールする



時刻tにおける最適化問題

時刻: $t \sim \mathrm{t} + \mathrm{N}\Delta t$ までの未来を

探索変数: $\mathbf{u}^{\mathrm{k}} = \left[v^k, \dot{\theta}^k\right]$, $\mathbf{x}^k = [x^k, y^k, \theta^k]$, $k = (1 \ldots N)$

ロボットがゴールへ　　省エネで

評価関数: $J_t = \sum_0^N \left\{ \left(\mathbf{x} - \mathbf{x}_g\right)^T Q \left(\mathbf{x}^k - \mathbf{x}_g\right) + {\mathbf{u}^k}^T R\, \mathbf{u}^k \right\} \Delta \mathrm{t}$

障害物に接触せずに　　ロボットの予測モデル

制約条件: $\left(x^k - x_o\right)^2 + \left(y^k - y_o\right)^2 \geq r_{\mathrm{safe}}^2$, $\mathbf{x}^{k+1} = f\left(\mathbf{x}^k, \mathbf{u}^k\right)$

どのように最適化問題を解くか？

1. 直接法/間接法: 数理最適化で解く

✓ 状態/入力の次元が高次元でも比較的解ける・解が滑らか

△ 問題のクラスが限定的: 勾配情報が必要・非線形性に苦戦する

Tips: 設計したMPCの最適化問題のクラスごとにソルバを変える

（例）・CasADi (IPOPT): 主双対内点法．計算は遅いが非線形問題もOK．最初に使うと良い

・Acados, FROCE PRO, etc.: 逐次２次計画法．凸最適化問題ならば高速に解ける

・C/GMRES法: 非線形問題OK & 圧倒的高速だが，扱いが難しい

2. サンプルベース法: モンテカルロ法によって解を探索

✓ 全てのクラスの最適化問題に適用可能 (非微分可能・非線形, etc.)

△ 滑らかな解が得られづらい．性能とサンプル効率にトレードオフ

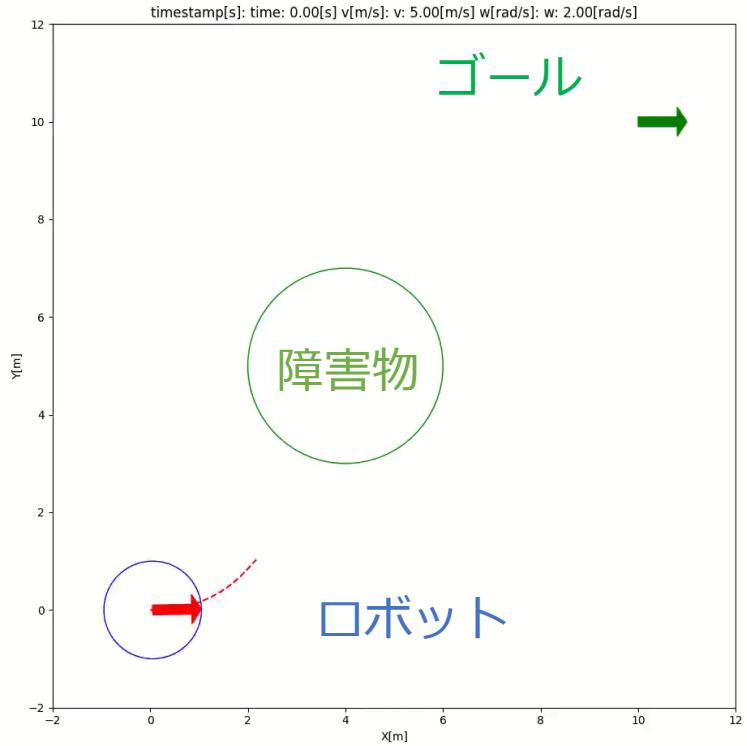
△ 入力次元に伴って探索空間が爆発的に増加 (次元の呪い)

Tips: 並列処理によって高速化が可能

数理最適化による自律移動制御

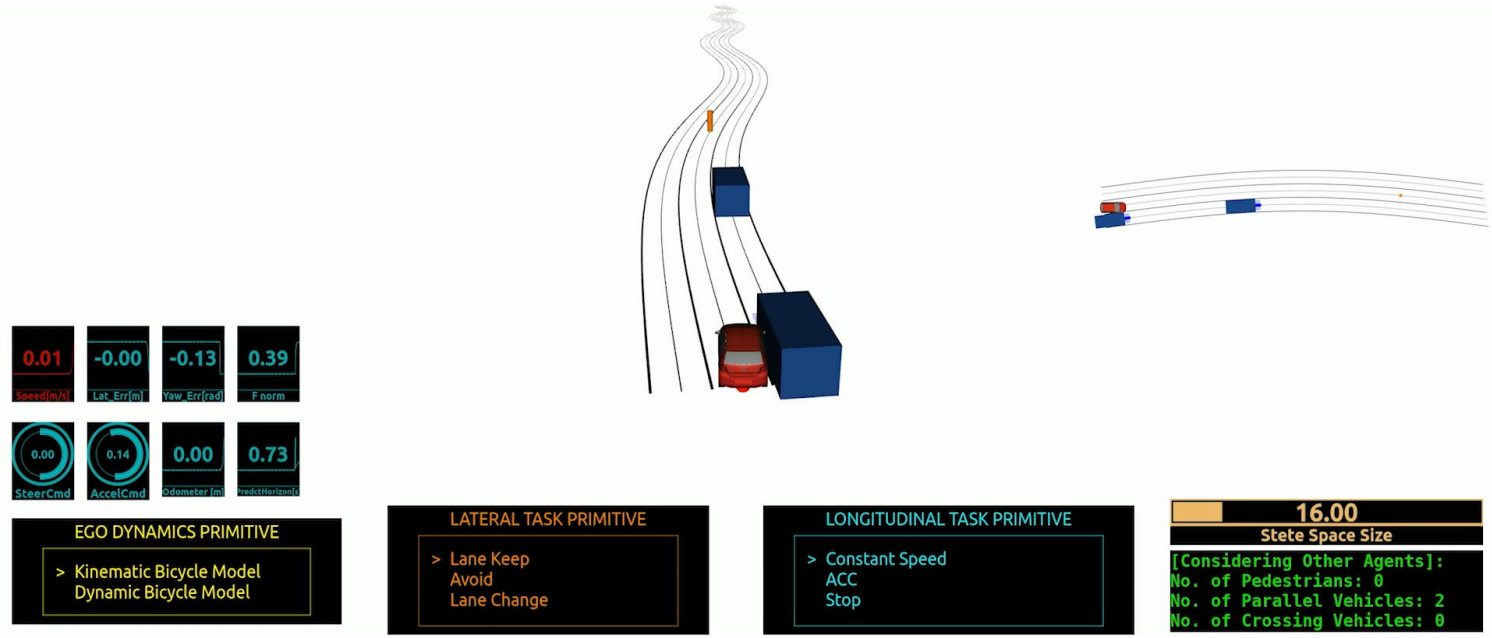
直接法: 主双対内点法 [CasADi (IPOPT)]

最適化問題の記述が簡単



間接法: C/GMRES法 [MPC Builder, Honda+, 2023]

高速に解ける



微分可能に障害物回避制約を書く必要がある = 障害物形状の認識が必要

↓

計算資源の限られた小型ロボットではコストマップとして障害物を扱いたい (=非微分可能)

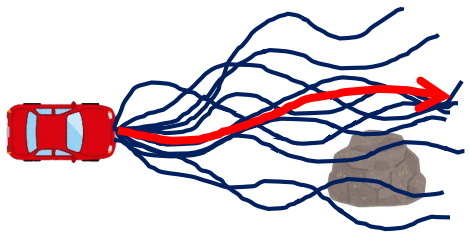
- K. Honda+, MPC Builder for Autonomous Drive: Automatic Generation of MPCs for Motion Planning and Control, IEEE IV, 2023.

サンプルベースMPC

- コストマップを扱える (微分不可能・非線形)
- 超複雑な予測モデルを扱える (モデルベース強化学習) → 近年ML界隈でも人気

◎古典的なサンプルベースMPCの例

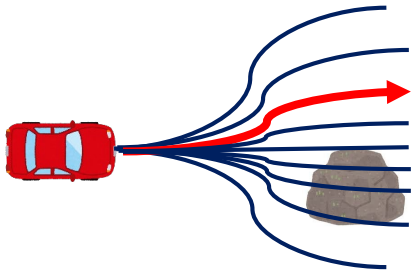
Single shooting method



1. ランダムサンプリング
2. ベストスコア軌道を選択

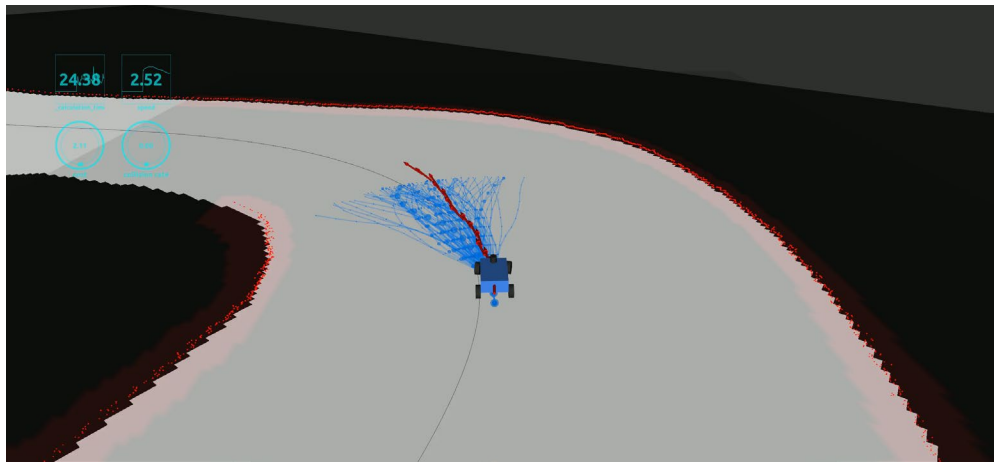
△ サンプル効率

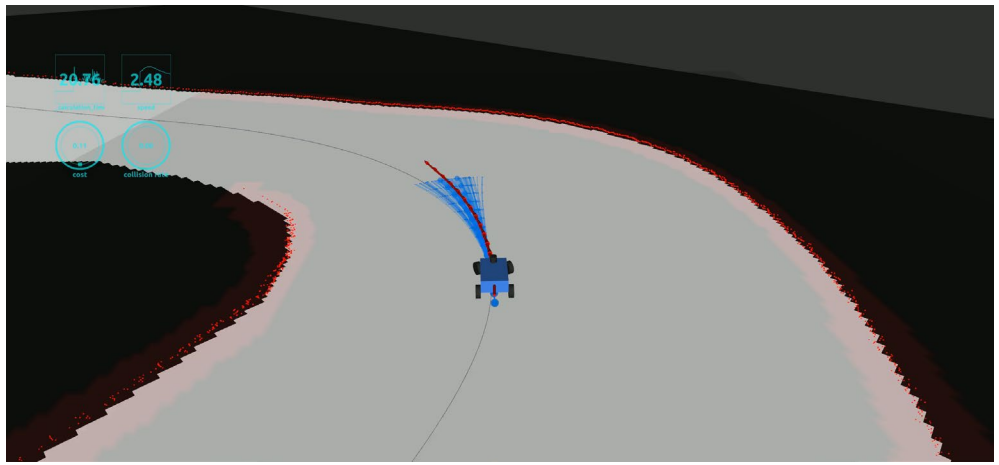
Dynamic Window Approach / State Lattice Planner



1. 平滑化済み軌道候補を散布
2. ベストスコア軌道を選択

△ 最適性





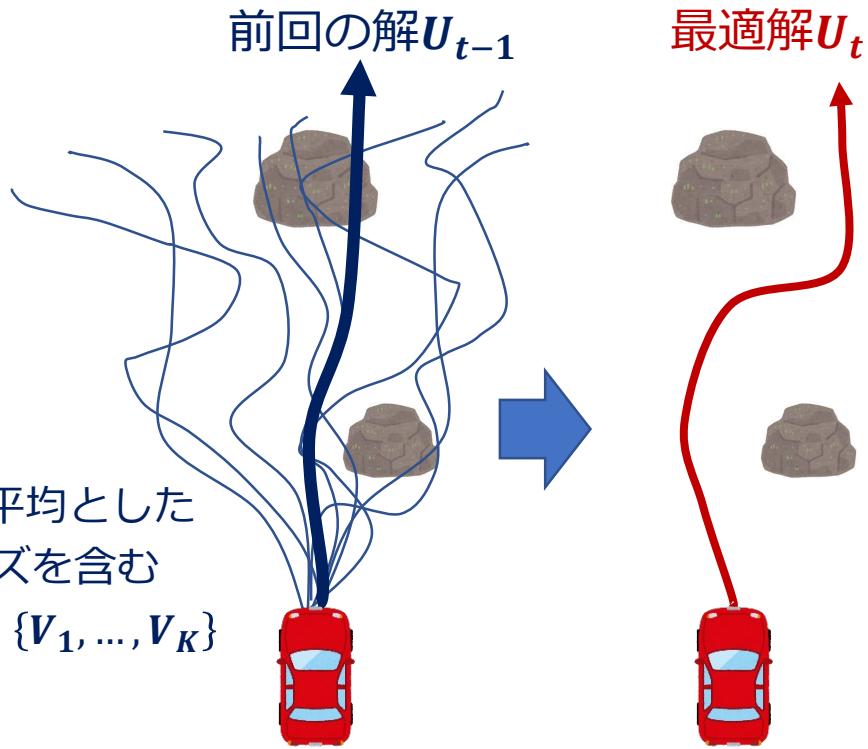
サンプル効率の良い & 滑らかな解を得ることができる手法が必要!

確率的MPC: モデル予測積分制御 (MPPI)

確率最適制御・情報論の観点から理論的にサンプルベース手法を改良できないか？

→ **Model Predictive Path Integral control (MPPI) [Williams+, 2018] から急速に発展中**

実装は簡単！:



前回の解U_{t-1}を平均とした
$\mathcal{N}(0,\Sigma)$のノイズを含む
サンプル系列 $V = \{V_1, \ldots, V_K\}$

1. 前回の解を平均としてランダムサンプリング

ランダムサンプル

$$V_k = U_{t-1} + \epsilon_k \qquad \epsilon_k \sim \mathcal{N}(0, \Sigma)$$

2. コスト関数を用いて重みを計算

重み　　状態コスト

$$w(V_k) = \frac{1}{\eta} \exp(-\frac{1}{\lambda} S(V_k) + \lambda \sum_{t=0}^{T-1} u_t^{\mathrm{T}} \Sigma v_t)$$

3. 重み付け和によって解を更新

更新解

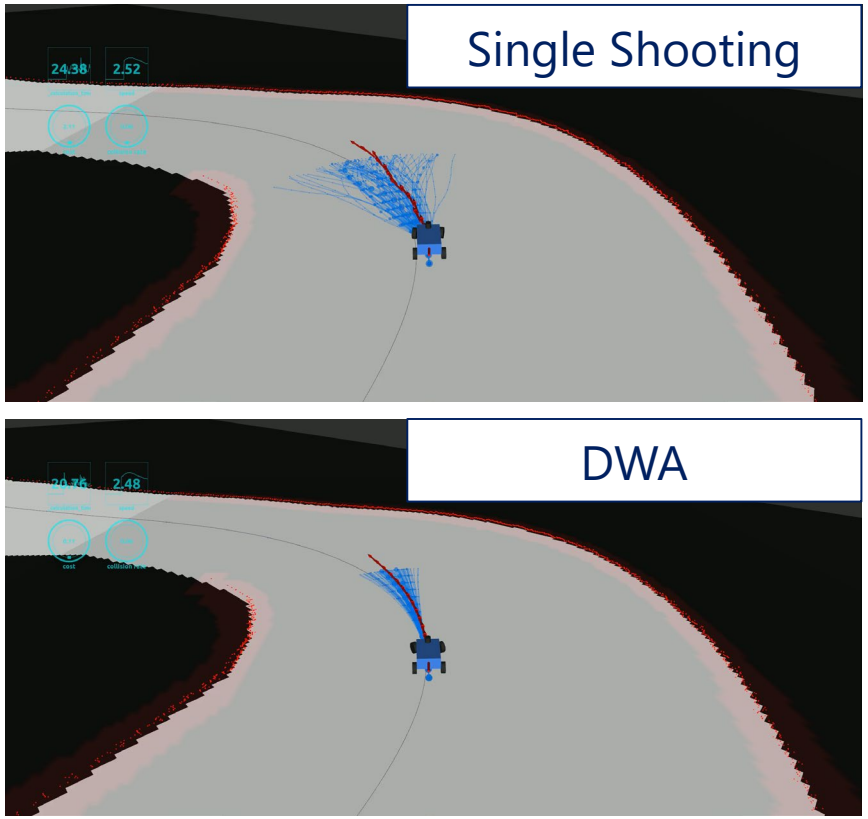
$$U_t = U_{t-1} + \sum_{k=1}^{K} w(V_k) V_k$$

- G. Williams+, Information Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving, IEEE T-RO, 2018

確率的MPC:モデル予測積分制御(MPPI)

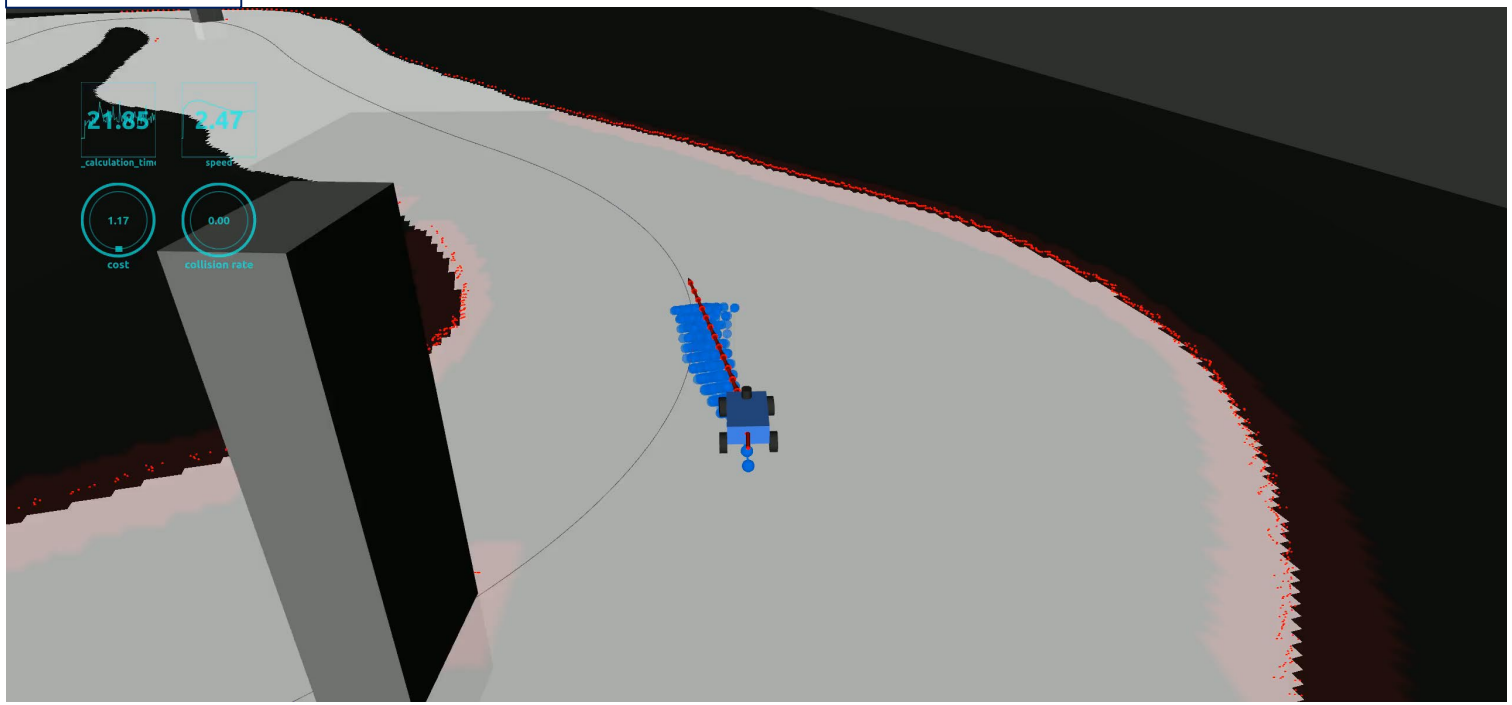
確率最適制御・情報論の観点から理論的にサンプルベース手法を改良できないか？

→ **Model Predictive Path Integral control (MPPI) [Williams+, 2018] から急速に発展中**



MPPI

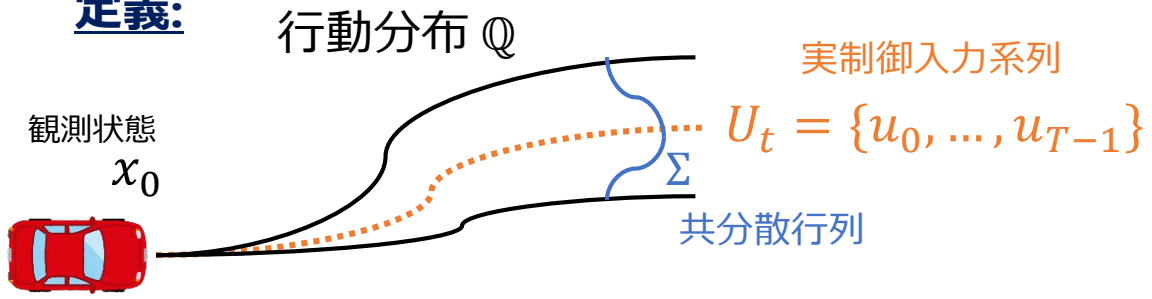
滑らかな解 & 高サンプル効率



理論的説明も充実 (次ページ以降)

(MPPI理論) 0. 準備

定義:



ガウスノイズ付与制御入力: $v_\tau \sim \mathcal{N}(u_\tau, \Sigma)$

状態方程式: $x_{\tau+1} = F(x_\tau, v_\tau)$

行動分布ℚの確率密度関数:

$$q(\{v_\tau\}_{0\dots T-1}) = Z^{-1}\exp\left(-\frac{1}{2}\sum_{\tau=0}^{T-1}(v_\tau - u_\tau)^{\mathrm{T}}\Sigma(v_\tau - u_\tau)\right)$$

確率最適制御問題:

$$U_t^* = \underset{U_t \in \mathcal{U}}{\operatorname{argmin}}\, \mathrm{E}_{\mathbb{Q}}\left[\phi(x_T) + \sum_{\tau=0}^{T-1}\left(c(x_\tau) + \frac{\lambda}{2}u_\tau \Sigma\, v_\tau + \beta u_\tau\right)\right]$$

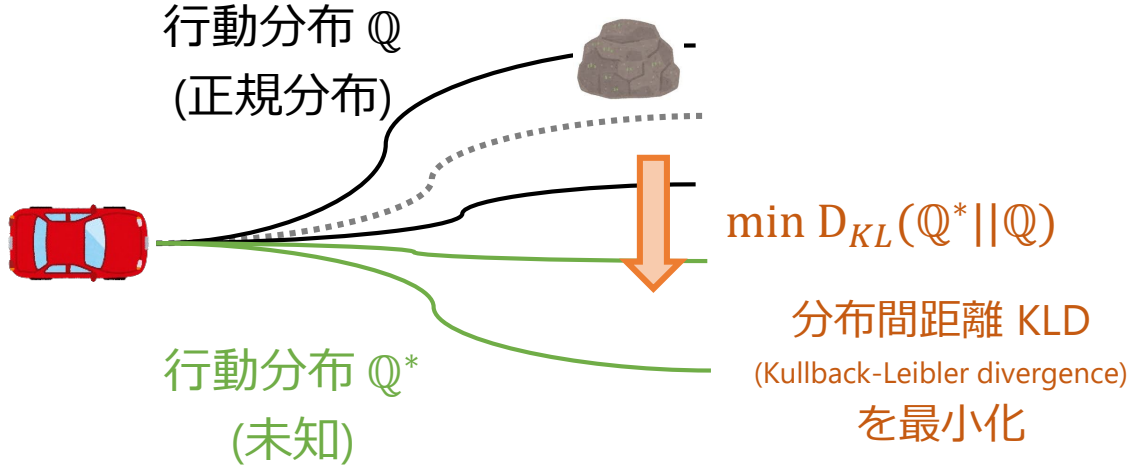
状態コスト　入力ペナルティ

入力制約条件を満たす集合

$S(x_0, \{v_0, \dots, v_{T-1}\})$

→ 期待値最小化は直接は解けない！

アプローチ：最適行動分布ℚ*を正規化行動分布ℚで近似



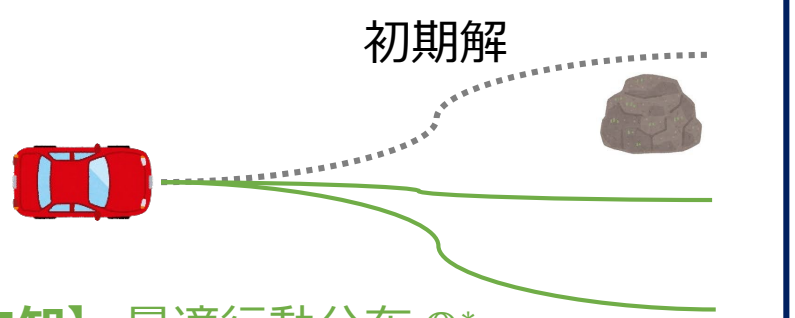
手法の概要

1. 最適行動分布ℚ*を推定 (分布形状: 未知)
2. 重点サンプリングによってKLDを最小化する

$$U_t^* \cong \underset{U_t \in \mathcal{U}}{\operatorname{argmin}}\, \mathrm{D}_{KL}(\mathbb{Q}^*||\mathbb{Q})$$

Kullback-Leibler divergence (KLD) の最小化

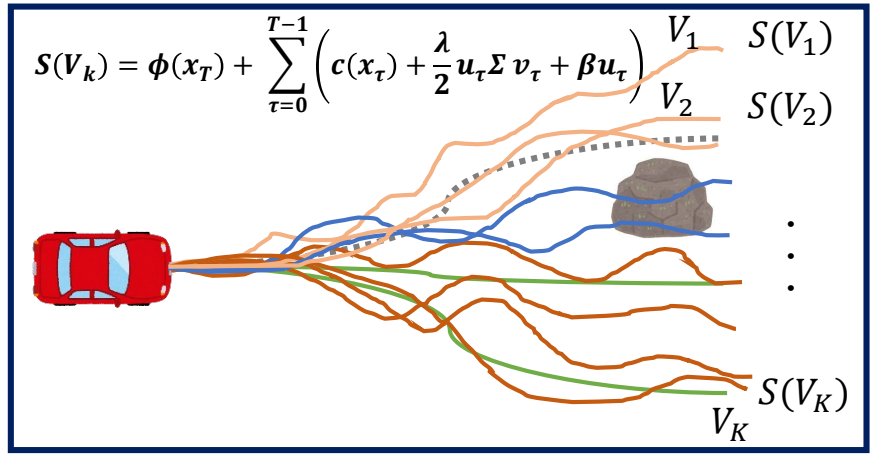
(MPPI理論) 1. 最適行動分布$\mathbb{Q}^*$の推定



【未知】 最適行動分布 $\mathbb{Q}^*$

ランダムサンプリング: $V_1, \dots, V_K$
&
尤度 (コスト) 計算: $S(V_1), \dots, S(V_K)$
↓
最適確率密度: $q^*(V_1), \dots, q^*(V_K)$を推定

※ 最適行動分布$\mathbb{Q}^*$は直接は求まらない



導出:

1. 自由エネルギーを定義: $\mathcal{F}(V_k) = \log\left(\mathrm{E}_{\mathbb{P}}\left[\exp\left(-\frac{1}{\lambda}S(V_k)\right)\right]\right)$
   - ・経路積分/ベイズ理論で良く出る式
   - ・λ: 温度パラメータ

   状態コストの期待値　$\mathbb{P}$: 制御しない時の行動分布

2. イエンセンの不等式で下界を計算 & 式展開: $-\lambda\mathcal{F}(V_k) \leq \mathrm{E}_{\mathbb{Q}}[S(V_k)] + \lambda \mathrm{D}_{KL}(\mathbb{Q}||\mathbb{P})$
   - $\mathbb{P}$, $\mathbb{Q}$は正規分布を仮定

3. 正規分布の確率密度関数を代入: $-\lambda\mathcal{F}(V_k) \leq \mathrm{E}_{\mathbb{Q}}\left[\phi(x_T) + \sum_{\tau=0}^{T-1}\left(c(x_\tau) + \frac{\lambda}{2}u_\tau \Sigma\, v_\tau + \beta u_\tau\right)\right]$

   = 解きたかった確率最適制御問題
   → 下限値が自由エネルギーで抑えられている
   → 等号条件を満たすqが最適確率密度関数

4. **最適確率密度関数: $q^*(V_k) = \eta^{-1}\exp\left(-\frac{1}{\lambda}S(V_k)\right)p(V_k)$**

(MPPI理論) 2. 逐次重点サンプリングによるKLD最小化

(A) 最適行動分布$\mathbb{Q}^*$と行動分布$\mathbb{Q}$のKLDを最小化

$$U_t^* \cong \underset{U_t \in \mathcal{U}}{\operatorname{argmin}} \, \mathrm{D}_{KL}(\mathbb{Q}^* || \mathbb{Q})$$

$$= \underset{U_t \in \mathcal{U}}{\operatorname{argmin}} \left\{ \mathrm{E}_{\mathbb{Q}^*}[\log q] - \mathrm{E}_{\mathbb{Q}^*}[\log q^*] \right\} \quad \text{(KLDの定義)}$$

$$= \underset{U_t \in \mathcal{U}}{\operatorname{argmax}} \, \mathrm{E}_{\mathbb{Q}^*}[\log q(V_k, U_t)] \quad (q^* \text{は} U_t \text{に非依存})$$

$$= \underset{U_t \in \mathcal{U}}{\operatorname{argmax}} \, \mathrm{E}_{\mathbb{Q}^*} \left[\frac{1}{2} \sum_{\tau=0}^{T-1} (v_\tau - u_\tau)^{\mathrm{T}} \Sigma (v_\tau - u_\tau) \right]$$

(qは正規分布密度関数)

$$= \mathrm{E}_{\mathbb{Q}^*}[V_k]$$

行動分布$\mathbb{Q}$が正規分布であることを仮定する

→ **KLD最小化問題は閉形式で求まる！**

→ ただし，最適行動分布$\mathbb{Q}^*$から直接サンプルはできない

※ Q^*を正規分布に近似し，サンプリングした期待値が平均となるのはもちろん自明

(B) 重点サンプリングによって$\mathbb{Q}^*$からのサンプルを近似

$$U_t^* = \mathrm{E}_{\mathbb{Q}^*}[V_k]$$

$$= \int q^*(V_k) V_k \, \mathrm{d}V$$

最適確率密度関数:

$$q^*(V_k) = \eta^{-1} \exp\left(-\frac{1}{\lambda} S(V_k)\right) p(V_k)$$

$$= \int \underbrace{\frac{q^*(V_k)}{q(V_k, U_t)}}_{} q(V_k, U_t) V_k \, \mathrm{d}V \quad \text{(重点サンプリング)}$$

重み: $w(V_k) = \frac{1}{\eta} \exp\left(-\frac{1}{\lambda} S(V_k) + \lambda \sum_{t=0}^{T-1} u_t^{\mathrm{T}} \Sigma v_t\right)$

$$= \mathrm{E}_{\mathbb{Q}}[w(V_k) V_k]$$

$\mathcal{N}(U_t^{\mathrm{init}}, \Sigma)$ (制御入力分布$\mathbb{Q}$) からV_kをサンプリング

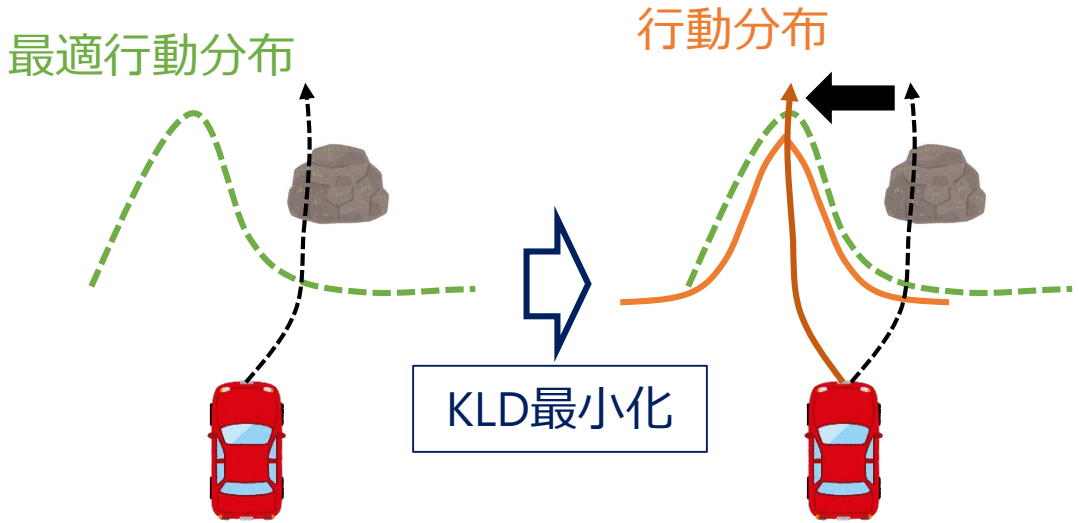
→ 重み付け平均によって**KLD最小化の大域最適解が求まる**

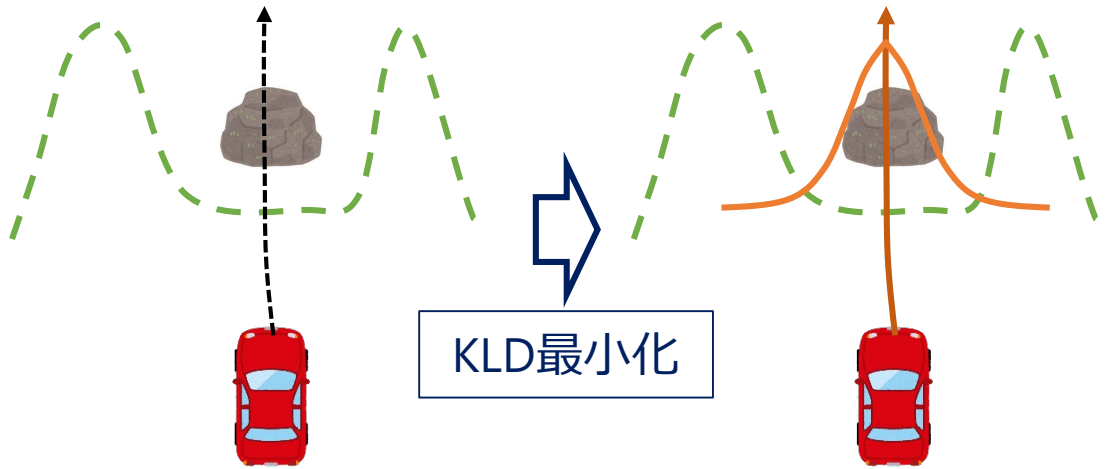
※ 大数の法則より，十分なサンプル数で最適性が保証

※ U_t^{init}: 前回の最適解U_{t-1}を用いることが多い (継承サンプリング)

MPPIまとめ

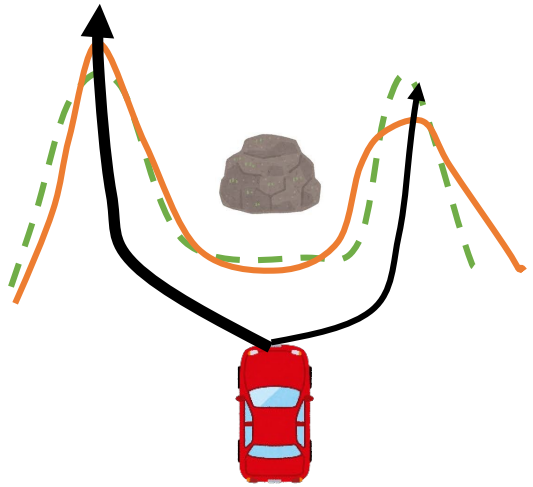
- **手法のキモ**:
  - 最適行動分布に近づくような正規近似行動分布を求める
  - 逐次重点サンプリングで解くが，**閉形式**で求まる (**イテレーション不要**)
  - サンプルを十分に取れば**KLDを最小化する意味での大域最適解**が求まる
    → サンプル効率・軌道の滑らかさが従来法 (CEM等) よりも高いことが知られている
- **欠点**:
  - 行動分布が正規分布であるため，多峰性の最適行動分布には覆いかぶさるような解が出ることがある
    → 行動分布を正規分布で近似するのは無理があるのでは？





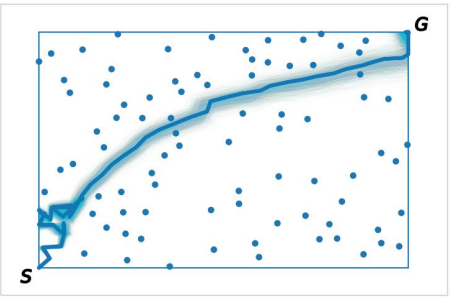
確率MPC最新動向: Bayesian MPC

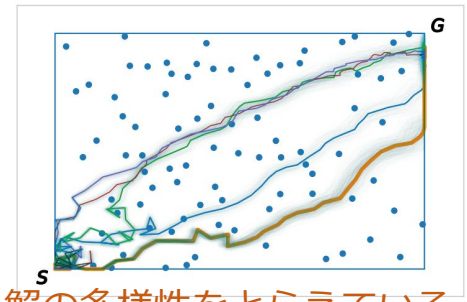
- **問題点**:
  MPPIでは行動分布を正規分布として近似するため多峰性分布を表現できない



- **手法**: MPCをベイズ推論として捉え，変分推論する (**Bayesian MPC**)
  - VI-MPC [1]
    - KLD最小化を変分推論問題として解く
    - 鏡像降下法によってKLDを最小化
  - SV-MPC [2]
    - サンプル (パーティクル) を最適輸送問題として"輸送"する
    - SVGD [3] とよばれる方法でKLDの最小化

[1]

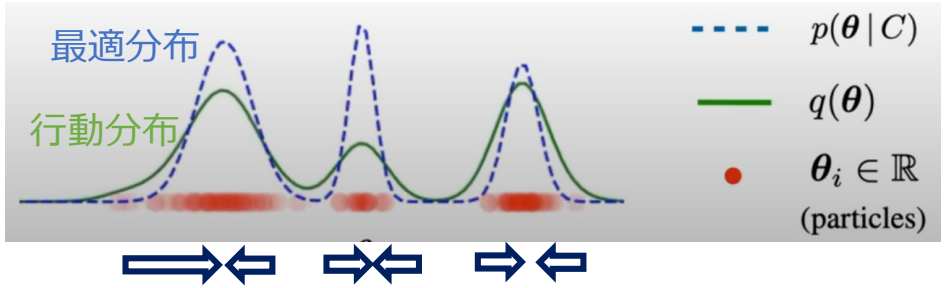




解の多様性をとらえている

- **特徴:**
  - パラメトリックな分布であればどんな行動分布もOK
    - 実装的には行動分布は混合ガウス分布をすることが多い
  - MPPIのように閉形式で解くことは未だ不可能

[2]

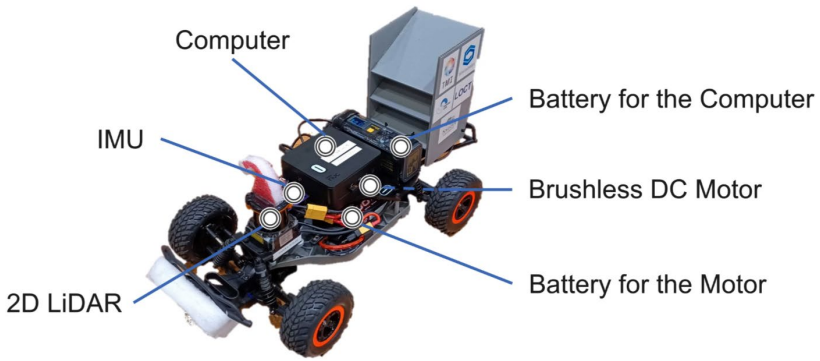


パーティクルを輸送して分布を表現

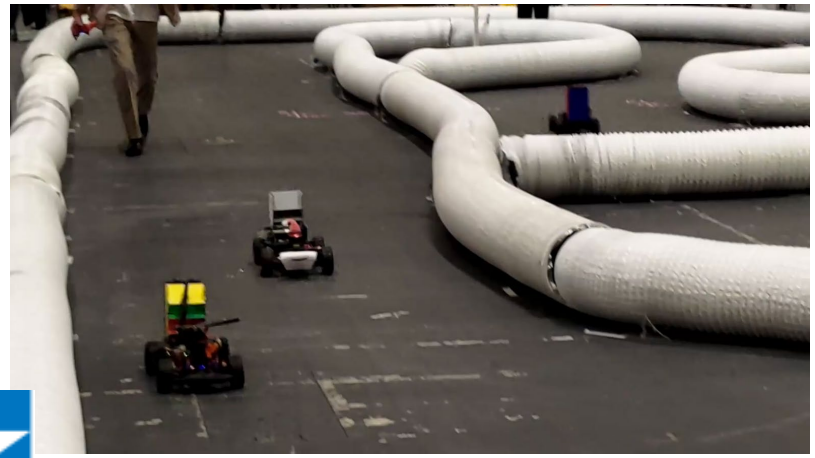
[1] M. Okada+, Variational Inference MPC for Bayesian Model-based Reinforcement Learning, PMLR, 2020.
[2] A. Lambert+, Stein Variational Model Predictive Control, CoRL, 2020.
[3] Q. Liu+, Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm, NIPS, 2016.

まとめ

- F1tenthは楽しみながら自動運転の技術チャレンジが出来る & 論文にも繋がる!
  - つくばチャレンジに近い技術・学術・教育のエコシステム
  - 日本でも手軽に参加可能な同様の枠組みがあると．．．
- 確率的MPCは実装簡単・性能良好なのでおすすめです！



Sponsored by TMI SHINMEI LOCT SUZUKI LAB.



TAMIYA