

インテリジェントなパスプランニング構築に関する調査研究

共同研究成果報告資料

神奈川工科大学 研究推進機構 先進自動車研究所（VRI）

上原 健一, 井上 秀雄

株式会社NTTデータ オートモビリティ研究研究所

目次

1. 目的
2. 2007 DARPA Urban Challenge (DUC)
3. 自律運転参照アーキテクチャ
4. ルートプランナー
5. 行動プランナー
6. 軌道生成
7. 学習の手法
8. 学習を用いた事例
9. パスプランニング周辺技術
10. まとめ

1. 目的

自動運転におけるパスプランニング手法の現状や課題を明確化する.

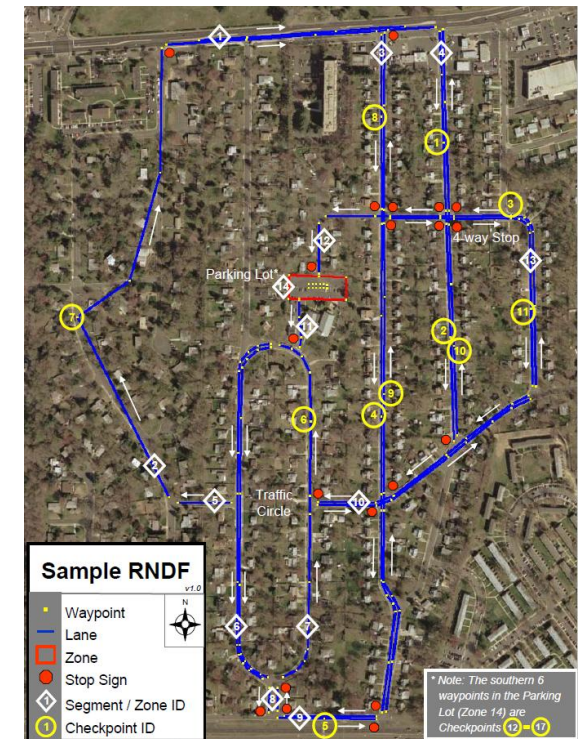
2. 2007 DARPA Urban Challenge (DUC)

- ✓ 2007年11月3日にファイナリスト11チームによる決勝がジョージ空軍基地で開催された
- ✓ コースは市街地を想定した総延長96 km (60 mi)で6時間以内に完走することが求められた
- ✓ ルールは、他の車列や障害物に対応したり車列に合流しながら、すべての交通規則に従う事も含まれていた
- ✓ 通常の車線走行の他に、駐車場での駐車や道路未整備ゾーンを走行・駐車する課題も含まれていた
- ✓ 完走は6チーム

完走 6 チーム

チーム名	車両	主体	タイム	順位
Tartan Racing	Boss	カーネギーメロン大学	4:10:20	1位
Stanford Racing	Junior	スタンフォード大学	4:29:28	2位
VictorTango	Odin	バージニア工科大学	4:36:38	3位
MIT	Talos	マサチューセッツ工科大学	約6時間	4位
The Ben Franklin Racing Team	Little Ben	ペンシルベニア大学	6時間超過	完走
Cornell	Skynet	コーネル大学	6時間超過	完走

コース例（サンプル. 実際のコースとは異なる）



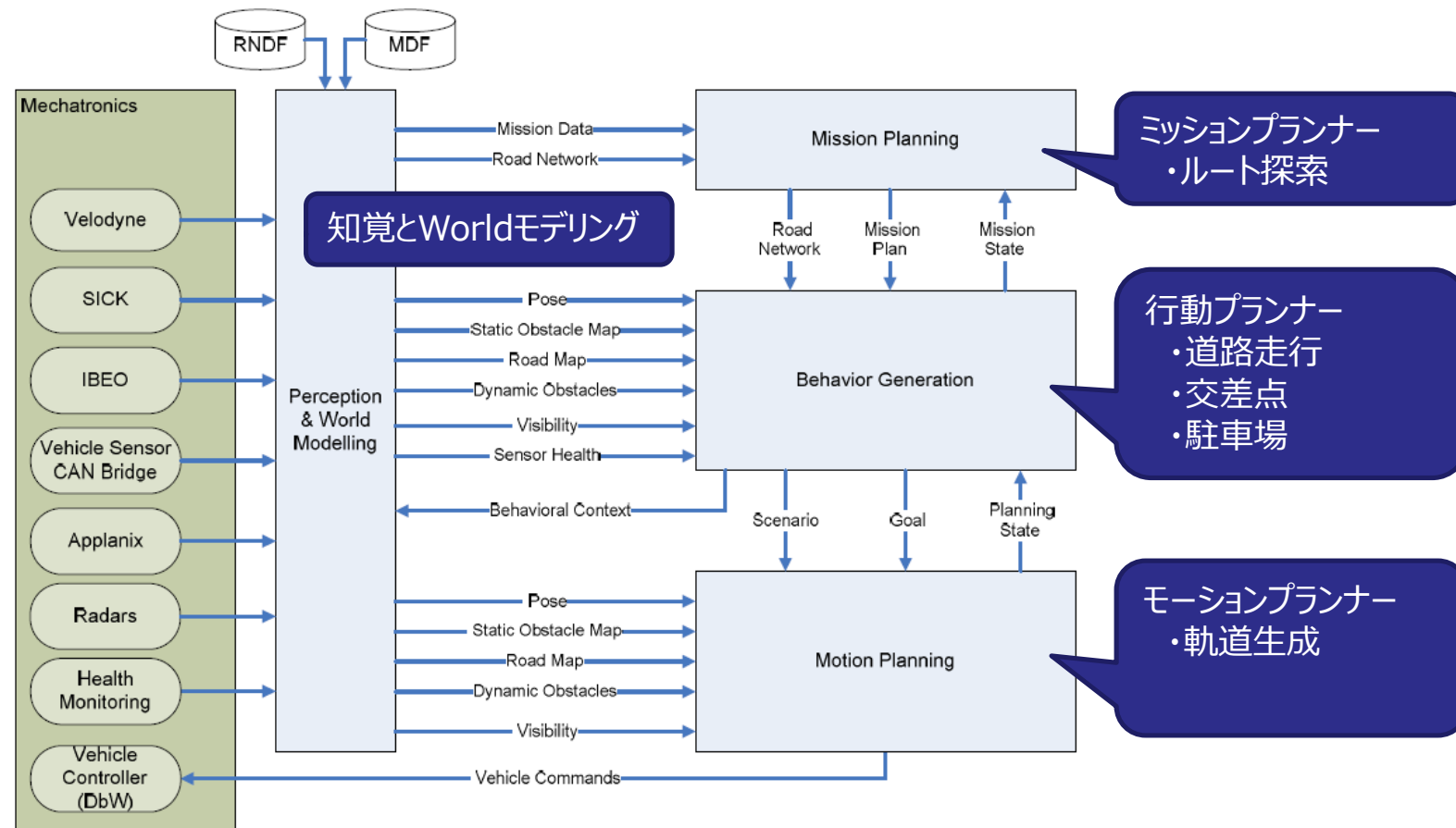
- ✓ <https://www.grandchallenge.org/grandchallenge/>

2. DUC; (1)カーネギーメロン大学 Boss アーキテクチャ



車両「Boss」

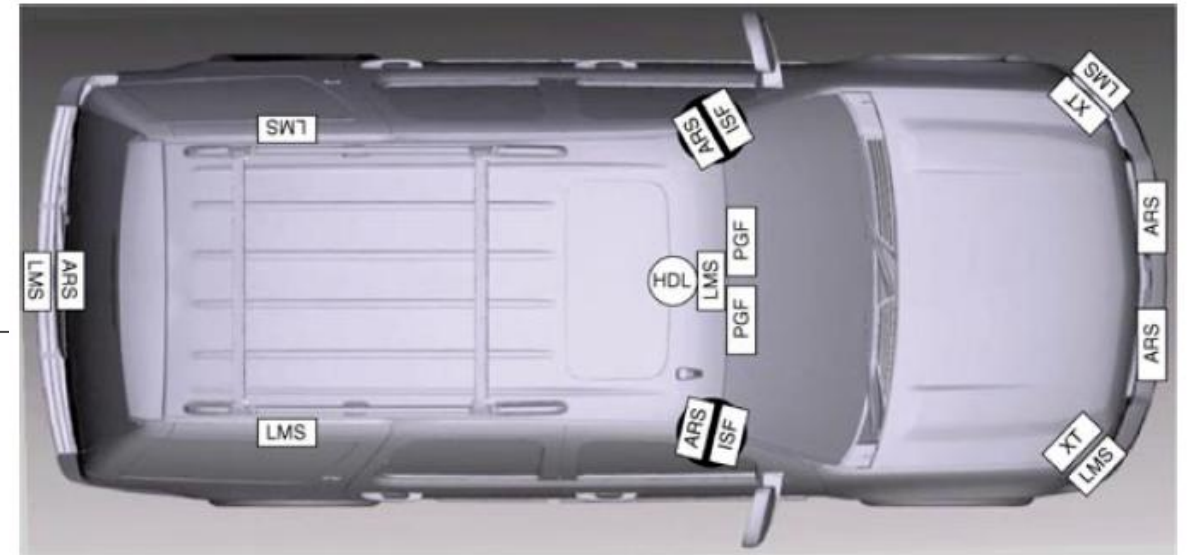
Bossのアーキテクチャは現在でも代表的なフレームワークとして頻繁に参照されている



- ✓ http://ri.cmu.edu/pub_files/2007/4/Tartan_Racing.pdf
- ✓ https://www.cmu.edu/traffic21/pdfs/urmson_christopher_2008_1.pdf

2. DUC; (1)カーネギーメロン大学 Boss システム構成

Sensor	Characteristics
Applanix POS-LV 220/420 GPS/IMU (APLX)	• Submeter accuracy with Omnistar VBS corrections • Tightly coupled inertial/GPS bridges GPS outages
SICK LMS 291-S05/S14 LIDAR (LMS)	• 180/90 deg × 0.9 deg FOV with 1/0.5-deg angular resolution • 80-m maximum range
Velodyne HDL-64 LIDAR (HDL)	• 360 × 26-deg FOV with 0.1-deg angular resolution • 70-m maximum range
Continental ISF 172 LIDAR (ISF)	• 12 × 3.2 deg FOV • 150-m maximum range
IBEO Alasca XT LIDAR (XT)	• 240 × 3.2 deg FOV • 300-m maximum range
Continental ARS 300 Radar (ARS)	• 60/17 deg × 3.2 deg FOV • 60-m/200-m maximum range
Point Grey Firefly (PGF)	• High-dynamic-range camera • 45-deg FOV

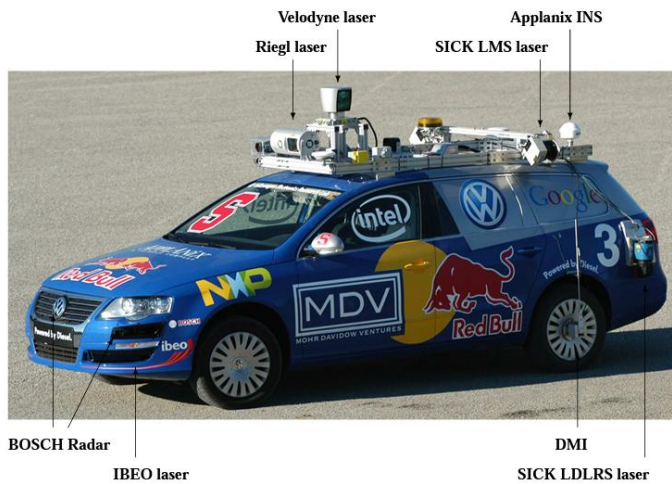


計算には10個の 2.16 GHz Core2Duoプロセッサを搭載した CompactPCI シェーシを使用。それぞれに 2 GB のメモリと 1 組のギガビットイーサネットポートが付属。各コンピュータは 4 GB のフラッシュドライブから起動するため、ディスク障害の可能性が低くなる。各コンピュータは、カスタムパルス/秒アダプタボードを介して時間同期される。

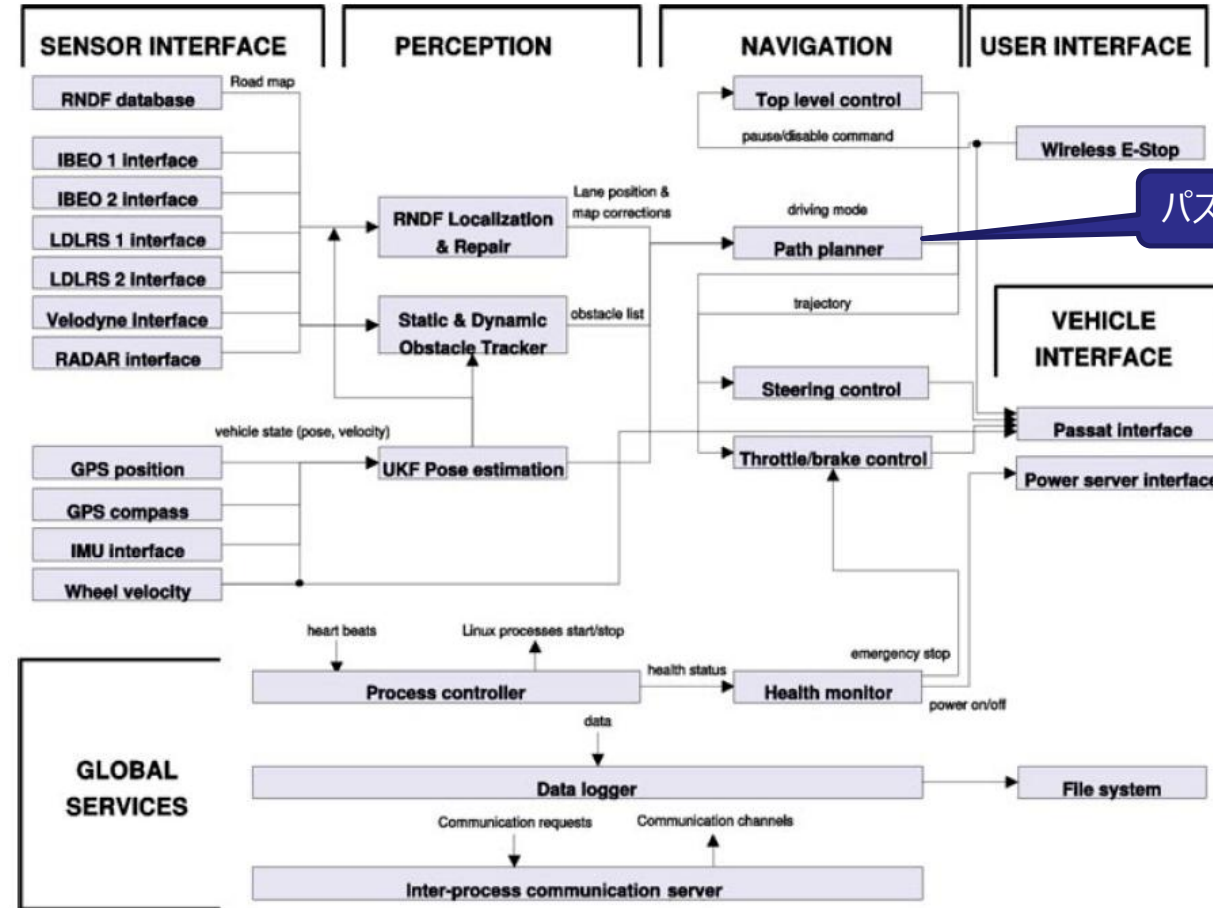
- ✓ https://www.cmu.edu/traffic21/pdfs/urmson_christopher_2008_1.pdf
- ✓ <http://writingaboutcars.com/360-sensors-give-autonomous-cars-vision/>

2. DUC; (2)スタンフォード大学 Junior アーキテクチャ

スタンフォード大学 (2007 DARPA Urban Challenge 2位)

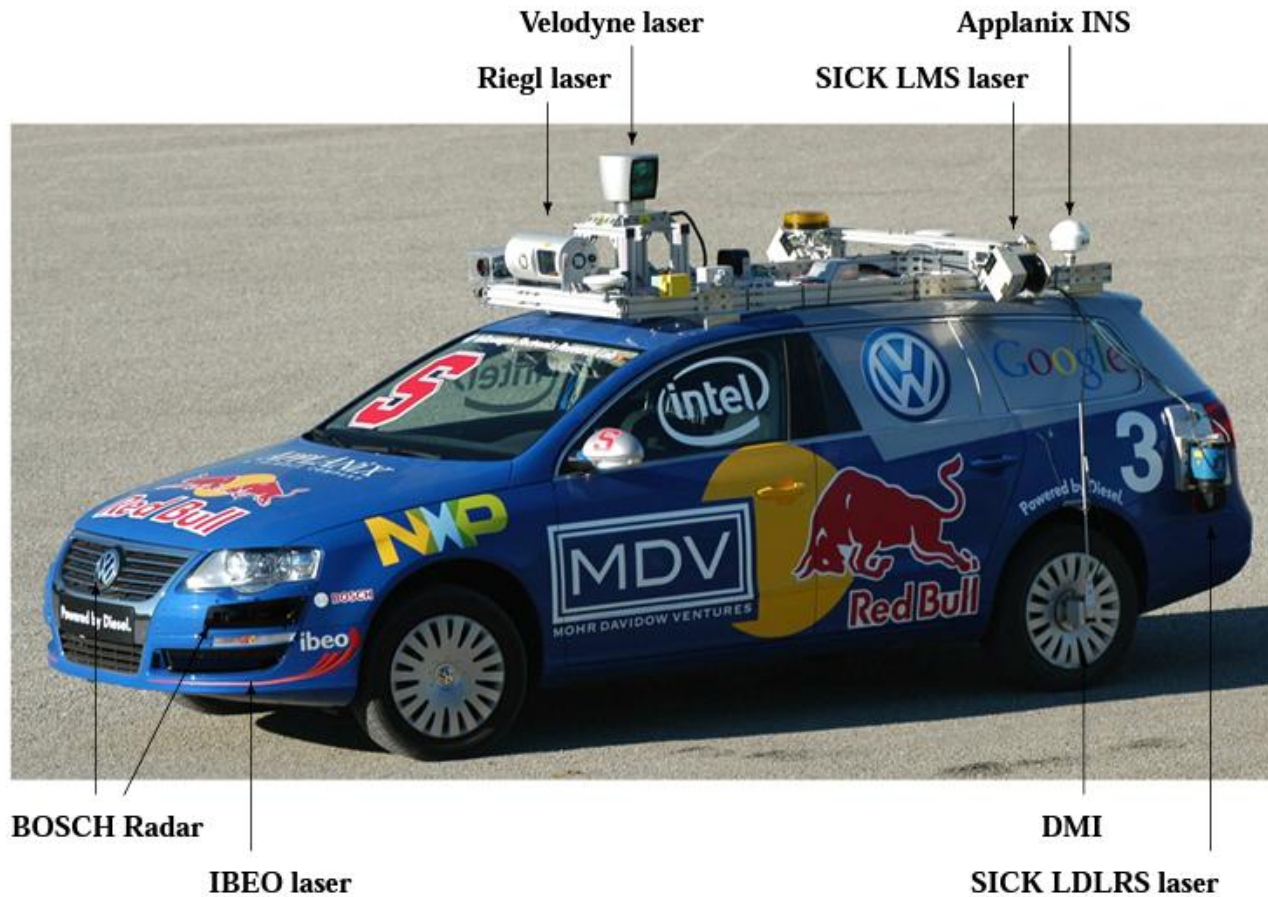


車両「Junior」



<http://robots.stanford.edu/papers/junior08.pdf>

2. DUC; (2)スタンフォード大学 Junior システム構成



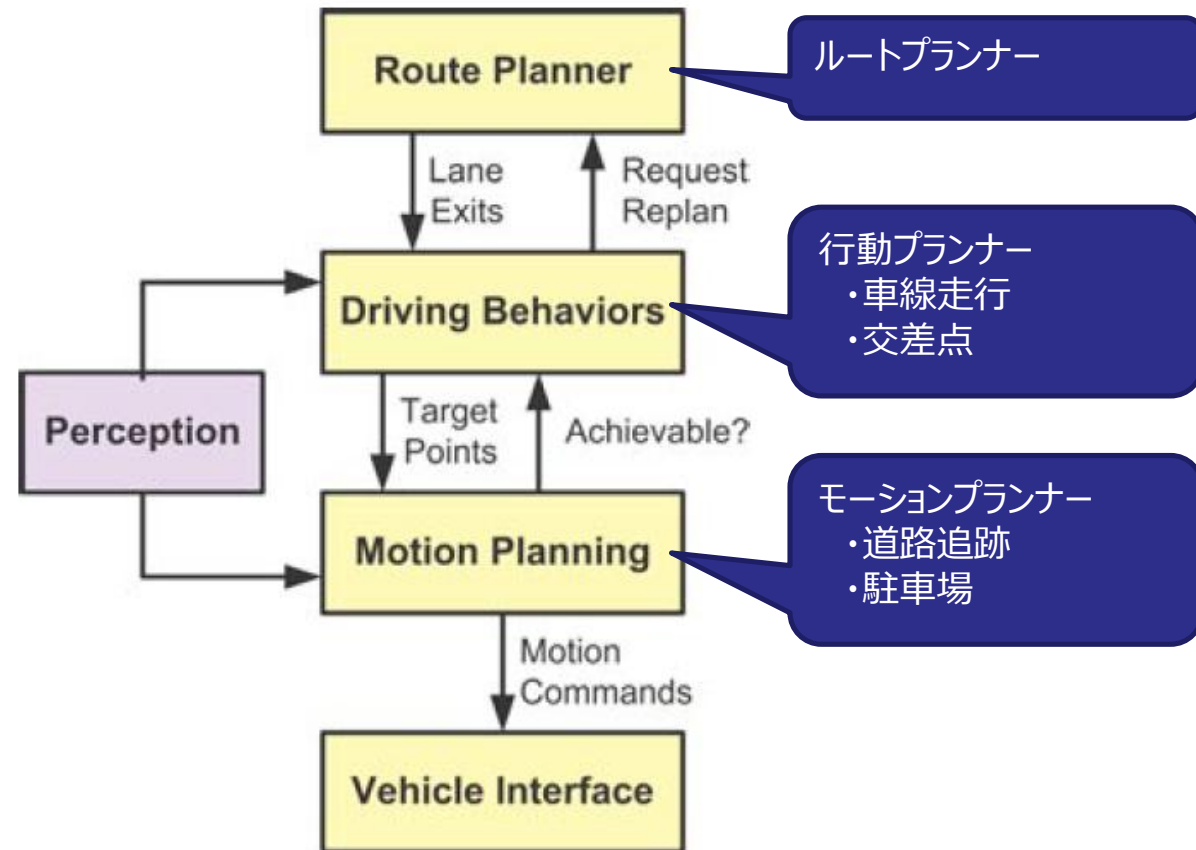
<http://robots.stanford.edu/papers/junior08.pdf>

- Applanix POS LV 420
 - ・GPS
 - ・高性能慣性測定
 - ・ホイールオドメトリ
 - ・Omnistar衛星ベースの仮想基地局サービス
 - ※位置誤差：100cm, 方向誤差：0.1 度未満
- SICK LMS 291-S14×2, RIEGL LMS-Q120 レーザーセンサー
3D道路構造と路面の赤外線反射率を計測し、レーン検出と正確な車両位置特定を行う
- Velodyne HDL-64E
障害物と移動車両の検出に使用.
水平視野：360° 垂直視野：30° 回転周波数：15Hz
- SICK LDLRS センサー×2, IBEO ALASCA XT ライダー×2
ペロダインを補完
- BOSCH 長距離レーダー (LRR2)×5
移動中の車両に関する追加情報を提供
- コンピューター システム
インテル クアッド コア サーバー×2 (Linux)
ギガビット イーサネット

2. DUC; (3)バージニア工科大学 Odin アーキテクチャ



車両「Odin」



<http://www.romela.org/wp-content/uploads/2015/05/Odin-Team-VictorTango's-Entry-in-the-DARPA-Urban-Challenge.pdf>

2. DUC; (3)バージニア工科大学 Odin システム構成



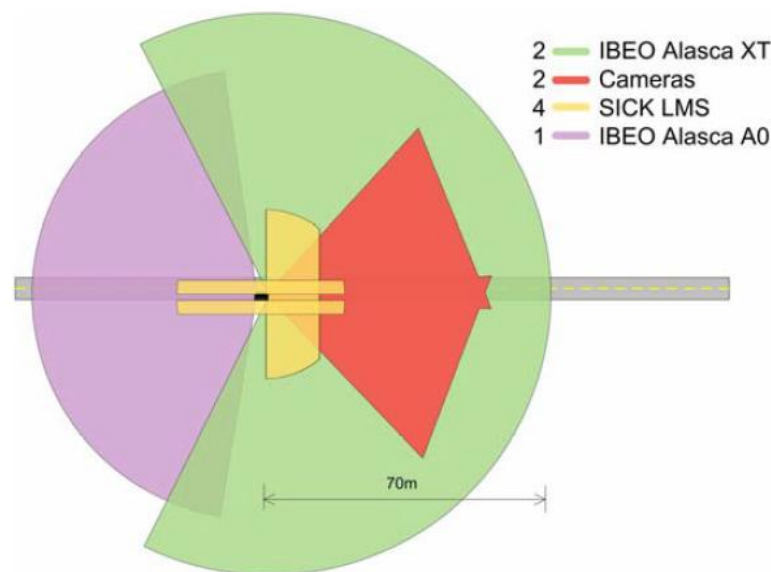
■ コンピュータシステム

Hewlett-Packard サーバー (クアッドコア×2) ×2

・Windows XP

(センサー処理ソフトウェア National Instruments LabVIEW Vision 使用の為)

・Linux



■ IEEE 1394 カラーカメラ

IBEO障害物分類ソフトの補完目的.

競技時には不使用

水平視野: 90° 解像度: 1024×768

フレームレート: 15fps

■ Novatel Propak LB+ GPS/INS

ホイール速度とステアリング角度を車両から入力して, 拡張カルマンフィルタにより10cmの精度を実現

■ IBEO Alasca XT Fusion レーザー

視野: 260°

検出のメインとして使用

公称検出距離200mだが実用70m

■ IBEO Alasca A0

視野: 150°

背後の接近車両の検出と後退時に使用

検出距離は実用50m

■ SICK LMS 291レーザー 前方×2

IBEO XTの下方を補完

■ SICK LMS 291レーザー 側方×2

車両側面の死角をカバー

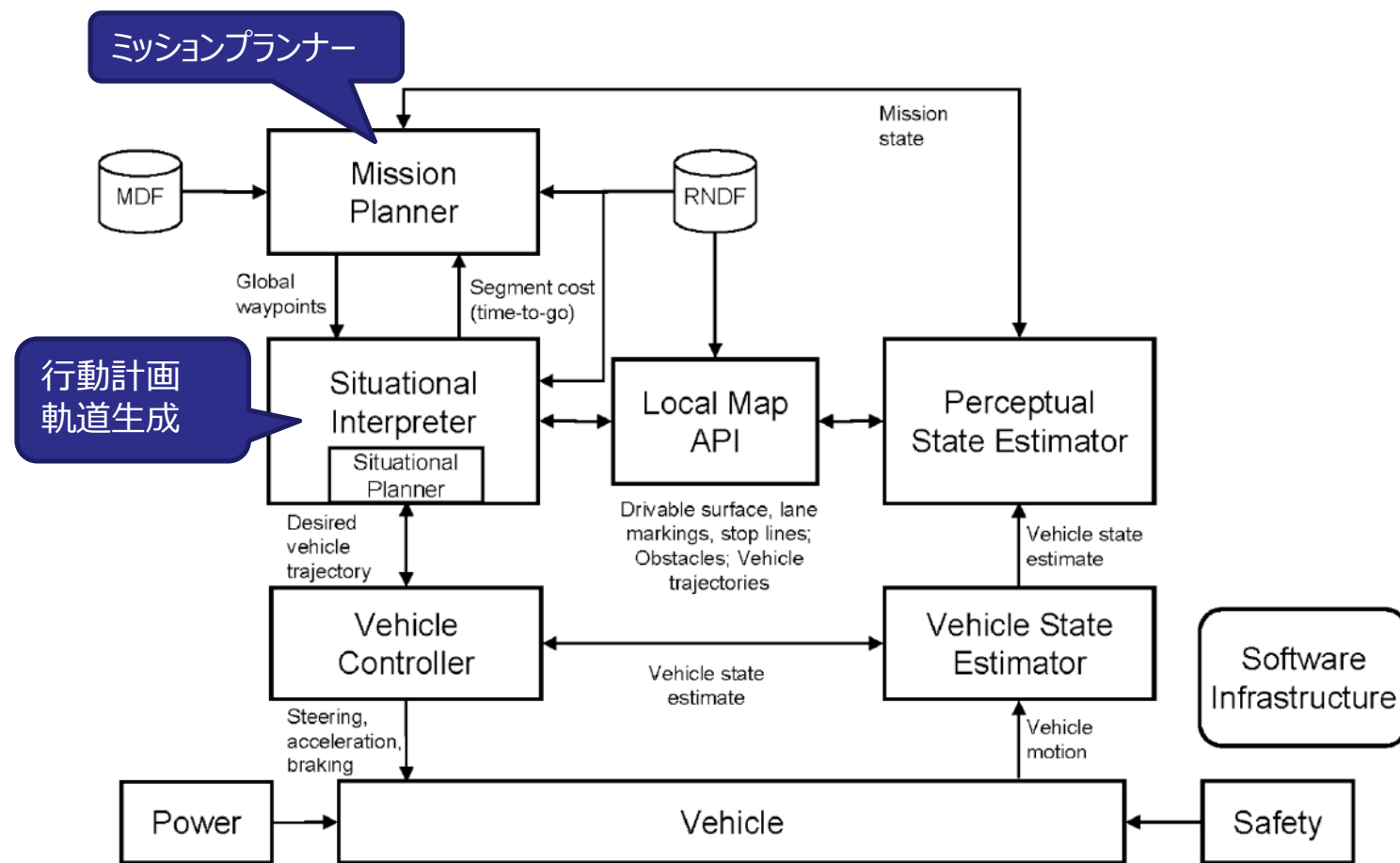
主として追越し時に使用

<http://www.romela.org/wp-content/uploads/2015/05/Odin-Team-VictorTango's-Entry-in-the-DARPA-Urban-Challenge.pdf>

2. DUC; (4)マサチューセッツ工科大学 Talos アーキテクチャ

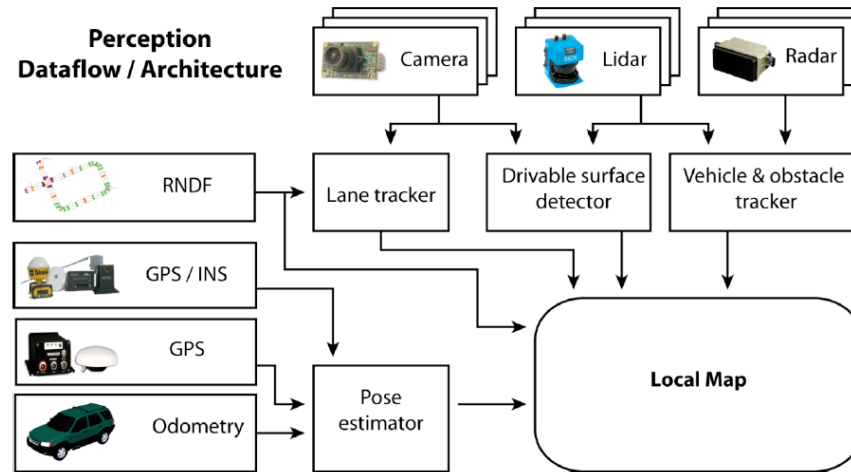


車両「Talos II」



<https://dspace.mit.edu/handle/1721.1/39822>

2. DUC; (4)マサチューセッツ工科大学 Talos システム構成



- IMU/GPS
XSens /Navcom Applanix POS LV220
- LiDAR
SICK LMS-291 S05
 - ・水平設置
近距離の障害物検出
 - ・下向きに傾斜して設置
運転可能な表面の検出
 - ・垂直に設置
道路プロファイルと車両ビッチの推定※多数の安価なセンサーが、少数の高価なセンサーよりも好ましいという原則にのっとっている
- Radar
Delphi ACC3
120m先の車両検出
- Camera
Pt. Grey Firefly MV
前方カメラで車線検出。後方カメラでも車線を検出し冗長性を確保
- コンピュータシステム
Fujitsu-Siemens BX600 Blade Cluster

<https://dspace.mit.edu/handle/1721.1/39822>

2. DUC; (5)カールスルーエ工科大学 AnnieWAY アーキテクチャ



車両「AnniWAY」

DUCファイナリスト11チームの中の1つ
(完走せず)

planningモジュールは以下の3レイヤで構成

1. ミッション計画 (Mission planning)

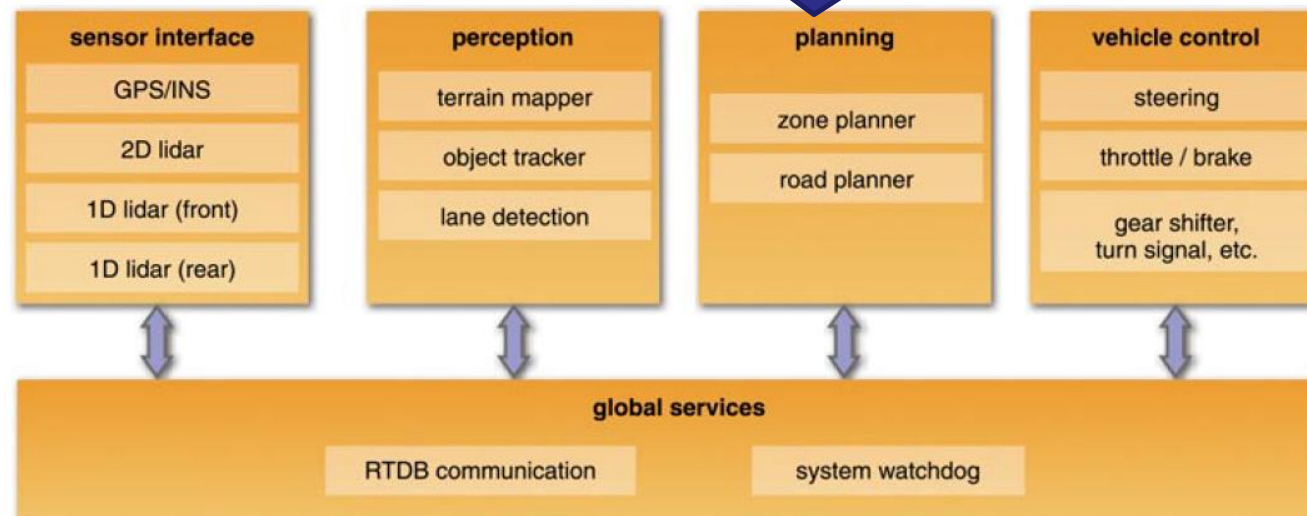
ミッションを達成するための計画を決定

2. 運転計画 (Maneuver planning)

交通規則を適用し、実際の運転操作 (例: 方向転換, 交差点, 追い越し) を計画し、対応する経路を生成

3. 衝突回避 (Collision avoidance)

認識モジュールから取得した障害マップを考慮して、計画された経路に衝突がないかどうかをテストする。衝突の可能性が高い場合は、別のパスを選択する。



<https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20252>

2. DUC; (5)カールスルーエ工科大学 AnnieWAY システム構成



■ ローカライゼーション (DGPS/INS)

高精度6軸INSとリアルタイムキネマティック (RTK)/GPS レシーバー, および正確なヘディング測定用の 2つのGPSレシーバーで構成されるデッドレコニング システム. オドメトリはホイール エンコーダーから直接取得. 差分補正を使用して動的条件下で 0.02 m を超える測位精度と, GPS アンテナ間の 2mの間隔を使用して0.1 度の方向精度を実現

■ メインLiDAR

64線 垂直視野: 26.5° 回転速度: 600rpm
水平方向角度分解能: 0.09 度
距離分解能: 最大 100 m で 5cm
秒間100 万ポイント以上を生成
車両周囲の認識に使用. 反射率マップは, レーン マーカーの検出に使用

■ サブLiDAR (前後に一つずつ)

駐車時のサポート

■ コンピューター システム

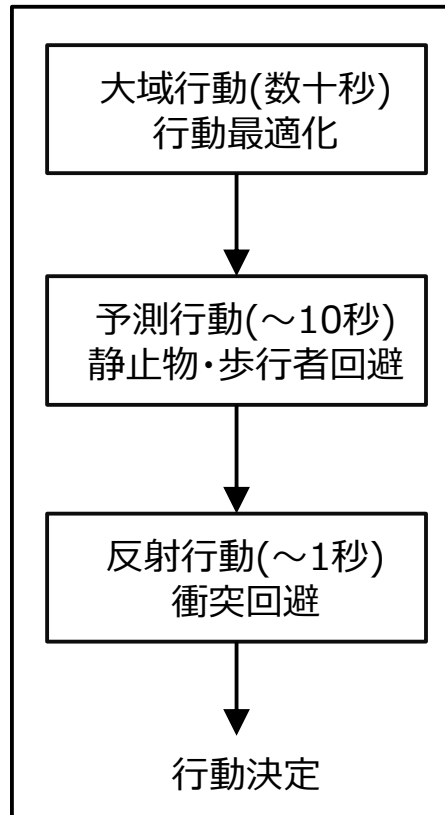
メインコンピュータをECUで増強

3. 自律運転参照アーキテクチャ

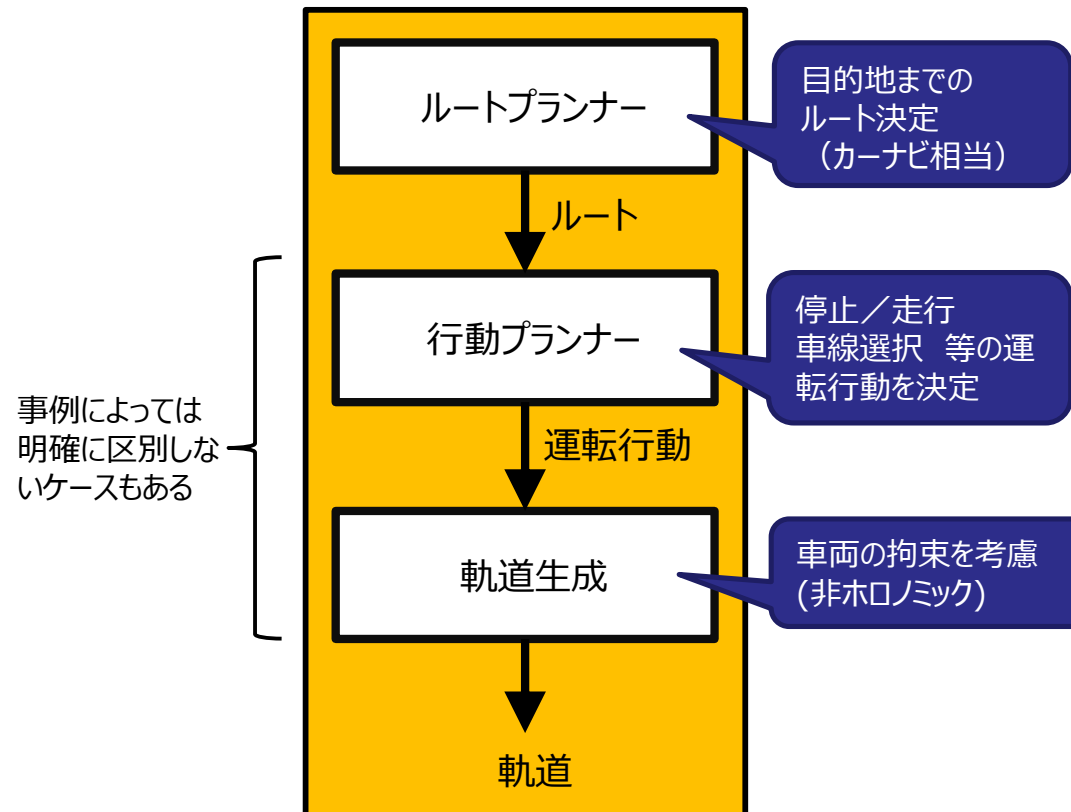
DUC事例に基づき 3 階層の参照アーキテクチャを設定する

経路：位置や姿勢を並べたデータ
軌道：時刻をパラメータとした位置と姿勢

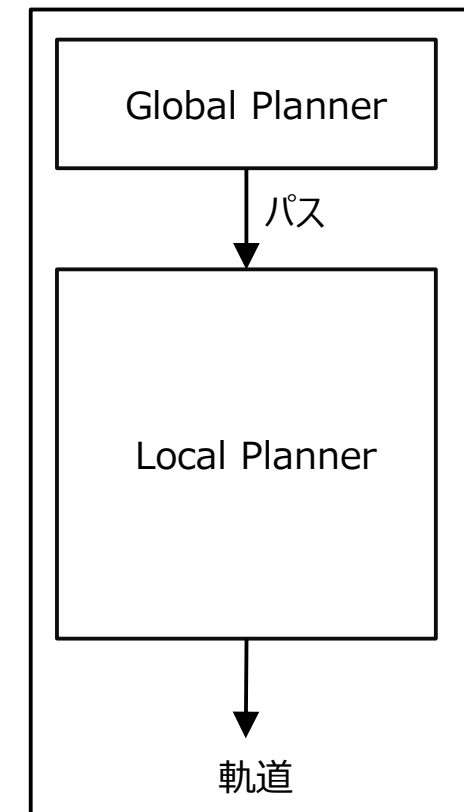
【参考】階層型自律移動アーキテクチャ



自律運転参照アーキテクチャ



【参考】ROS Navigation Stack

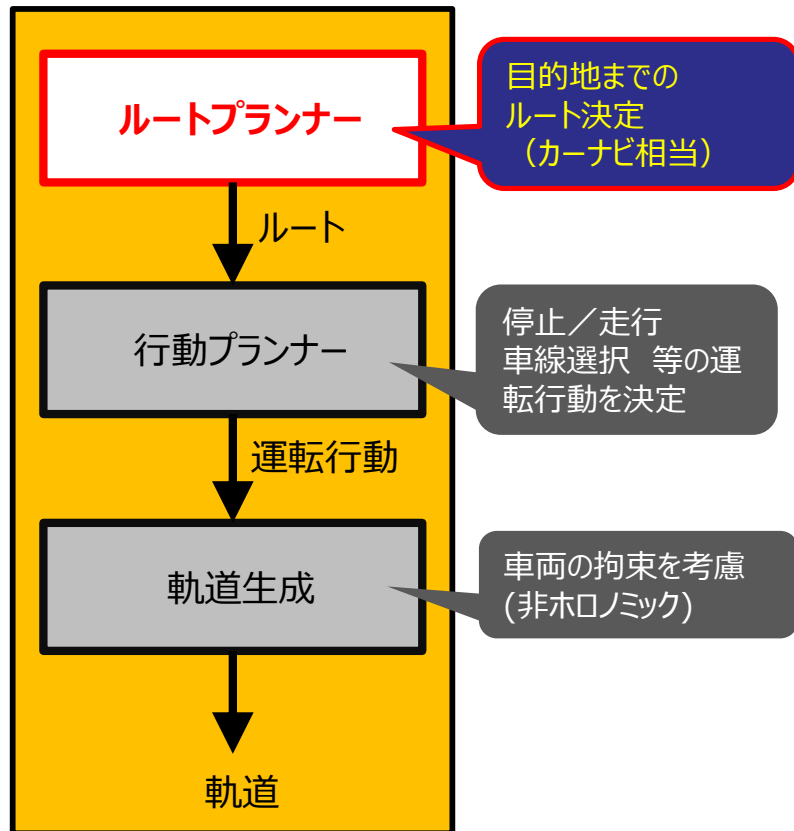


香月理絵, 荒井幸代, 大前学, 大日方五郎, 川崎敦史, 橋川雄樹, 小林祐一, 菅沼直樹, 田崎豪, 谷沢昭行, 新田修平, 野呂瀬琴, 馬場厚志, 藤吉弘亘, 目黒淳一, 森出茂樹, 谷口敦司, 山下倫央, 日本ロボット学会, 自動運転技術入門, オーム社, pp.110-111 (2021).

4. ルートプランナー

ルートプランナーは目的地までの最適なルートを探索する。

自律運転参照アーキテクチャ

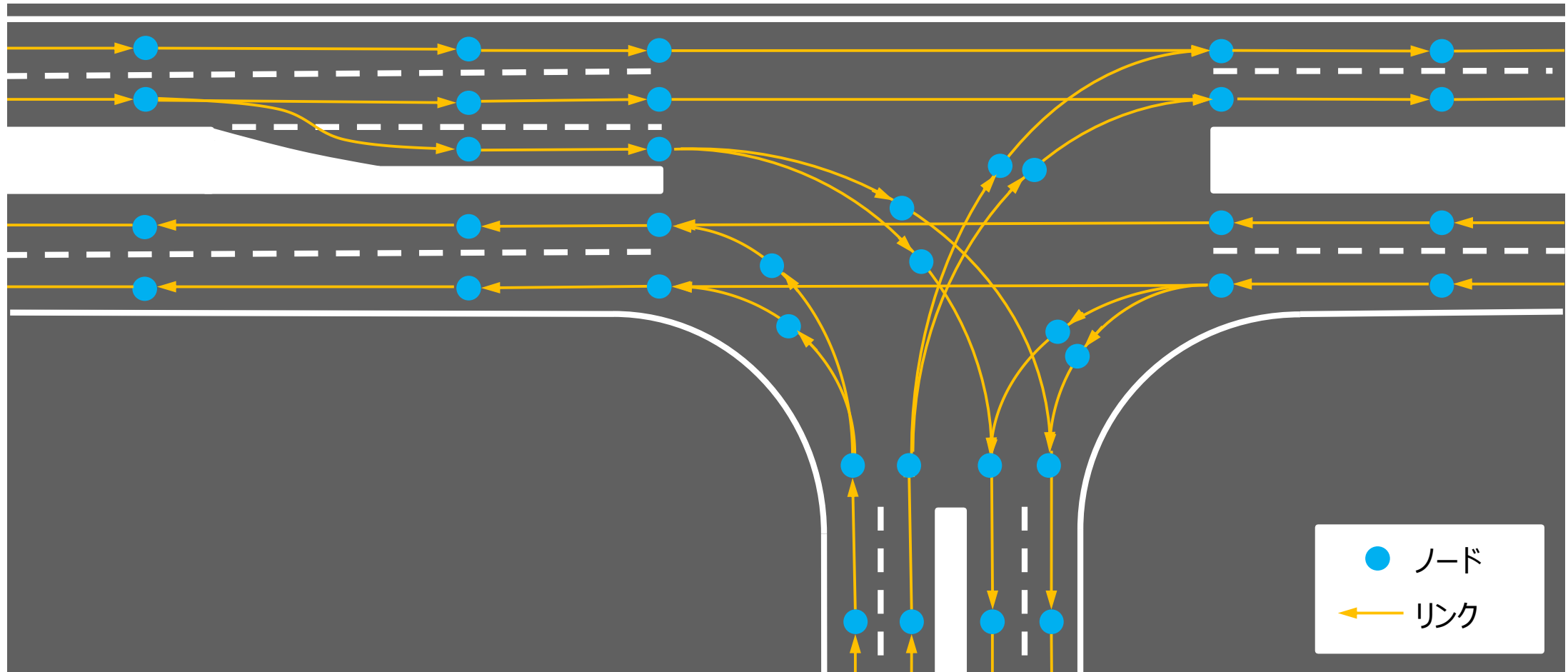


DUC主要チームのルートプランニングアルゴリズム

チーム名	車両	主体	ルートプランニングアルゴリズム
Tartan Racing	Boss	カーネギーメロン大学	Anytime D*
Stanford Racing	Junior	スタンフォード大学	Hybrid A*
VictorTango	Odin	バージニア工科大学	A*
MIT	Talos	マサチューセッツ工科大学	A*

4. ルートプランナー; 道路のグラフ化

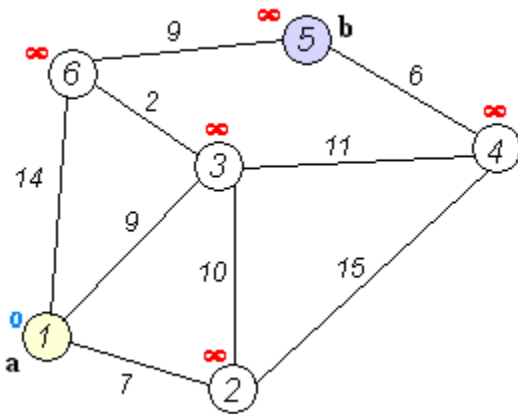
ルートプランニングに際して, 車線も考慮した道路網をノードと方向付リンクでグラフ化する (ベクターマップの情報を利用)



4. ルートプランナー; グラフ探索アルゴリズム ダイクストラ法

ダイクストラ(Dijkstra)法は, グラフ探索の基本となるアルゴリズム.

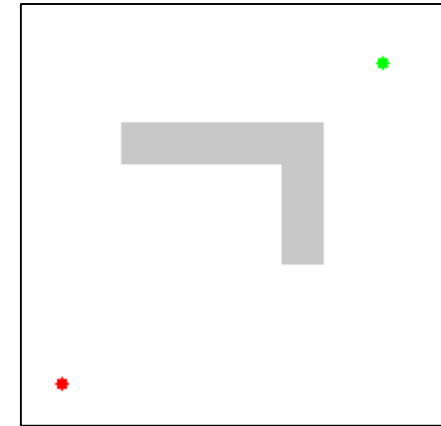
リンクに重み(所要時間等)を付加し, グラフ探索アルゴリズムで最適経路探索を行う.



赤ノード: 最短時間が確定したノード

青字: ノードまでの最短時間 (順次更新)

- ✓ 各ノードまでの最短時間を確定するまで更新してゆくアルゴリズム
- ✓ ゴールまでの全ルートを検証し, ゴールまでの最短時間が確定したら終了
- ✓ 重みは非負数のみ (非負数を含む場合はベルマン-フォード法等が利用できる)
- ✓ ROSのGlobal Plannerがサポートしている
http://wiki.ros.org/global_planner



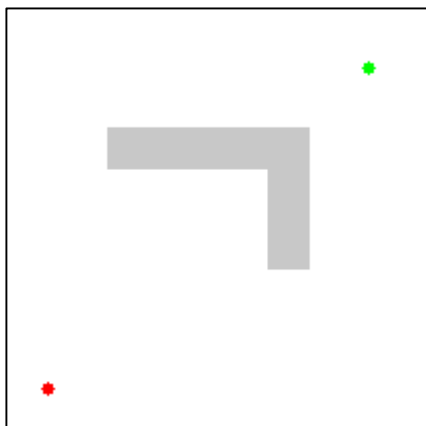
https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

4. ルートプランナー; グラフ探索アルゴリズム A*法

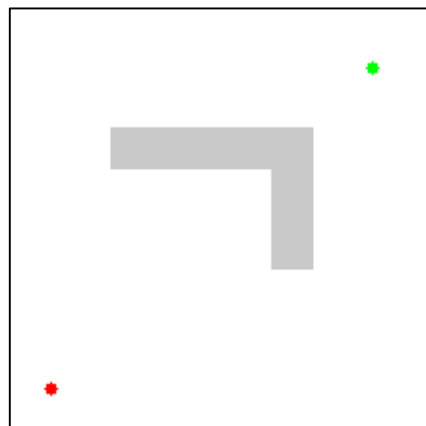
A*(エースター)法は、現在の自動運転におけるルートプランニングの基本となるアルゴリズム。

ダイクストラ法の発展形で、探索に方向付けを与えて高速化を図ったアルゴリズム。

ダイクストラ法



A*法



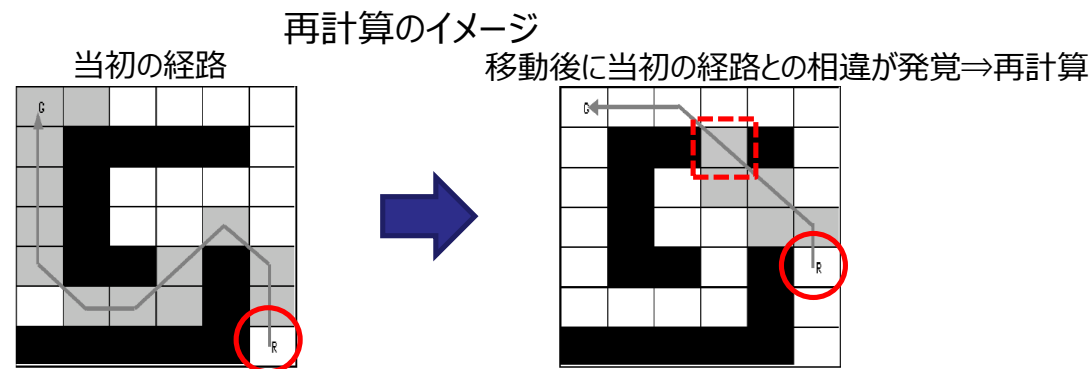
- ✓ カーナビなどでは、方向付けにはゴールまでの直線距離(ユークリッド距離)が用いられる (ゴールまでの直線距離が近い方向を優先して探索)
- ✓ ROSのGlobal Plannerがサポートしている
http://wiki.ros.org/global_planner

https://ja.wikipedia.org/wiki/A*

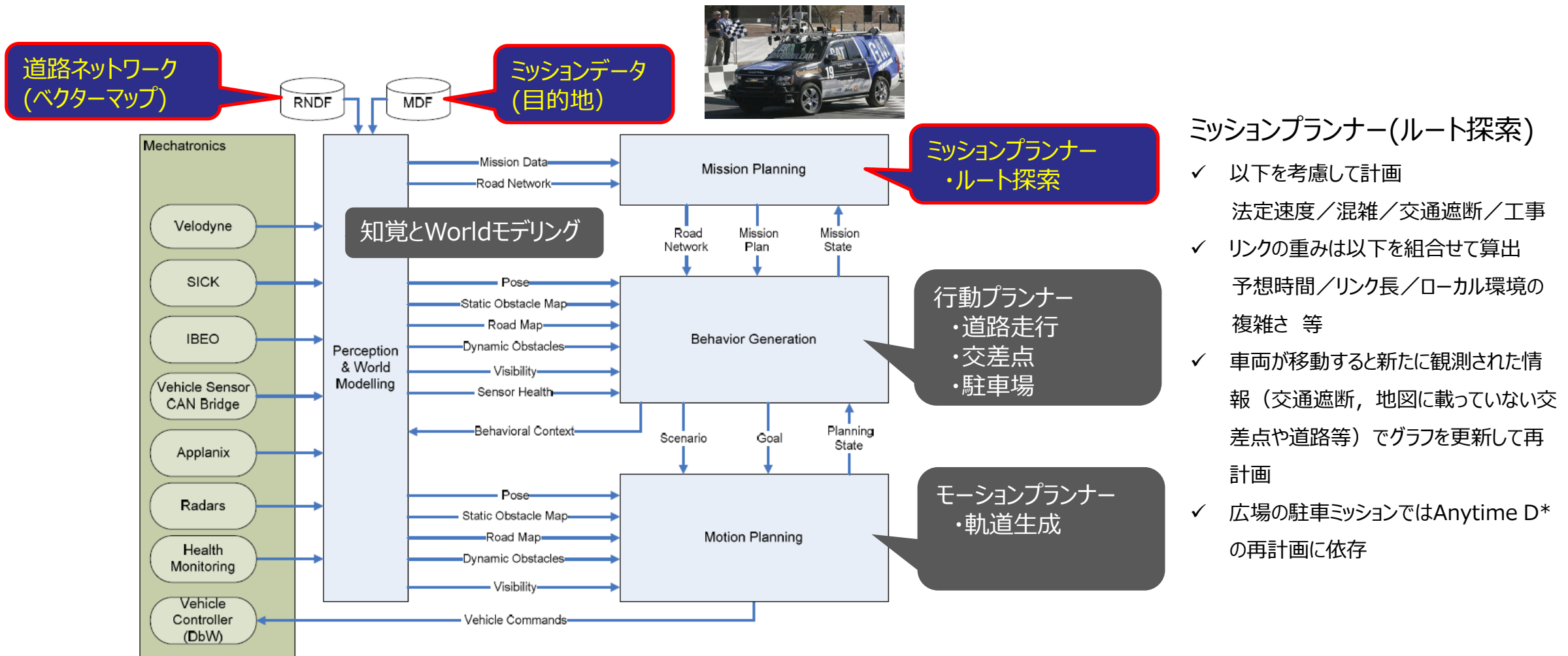
4. ルートプランナー; グラフ探索アルゴリズム A*の改良型

A*をベースに、動的な変化に対応したアルゴリズムや滑らかな経路を生成する、といった様々な改良型アルゴリズムがカーネギーメロン大学を中心に提案されている。

アルゴリズム名	概要	発表年	参照
D*	変化した箇所限定して再探索することで高速化を図る 新しい情報が発見されるとパスを段階的に修復	1995	http://robotics.caltech.edu/~jwb/courses/ME132/handouts/Dstar_ijcai95.pdf https://www.cs.cmu.edu/~motionplanning/lecture/AppH-astar-dstar_howie.pdf
D* Lite	D*を単純化したもの。わずかに効率的	2002	http://idm-lab.org/bib/abstracts/papers/aaai02b.pdf
Incremental A*	再検索時に以前の一部の検索ツリーを再利用して高速化	2002	https://proceedings.neurips.cc/paper/2001/file/a591024321c5e2bdbd23ed35f0574dde-Paper.pdf
Field D*	D*とD* Lite を拡張して線形補間を使用し、滑らかなパスを効率的に生成	2005	http://robots.stanford.edu/isrr-papers/draft/stentz.pdf
Anytime D*	D* Liteをベースとして複雑な状況に対応できるようにアルゴリズムを改善 (Bossで使用)	2005	http://www.cs.cmu.edu/~ggordon/likhachev-et.al.anytime-dstar.pdf
Hybrid A*	滑らかなパスを生成 (Juniorで使用)	2008	https://ai.stanford.edu/~ddolgov/papers/dolgov_gpp_stair08.pdf - MATLABに関数が存在「plannerHybridAStar」 https://jp.mathworks.com/help/nav/ref/plannerhybridastar.html?lang=en



4. ルートプランナー; DUCカーネギーメロン大学 Bossの事例

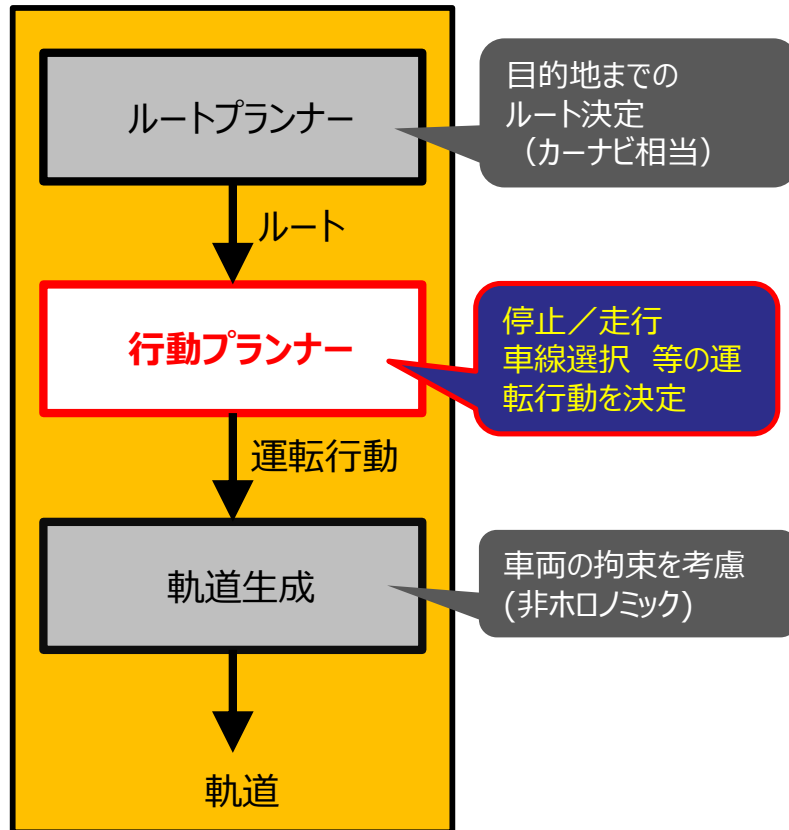


http://ri.cmu.edu/pub_files/2007/4/Tartan_Racing.pdf

5. 行動プランナー

行動プランナーは現在の状況に応じた適切な運転行動を決定する。

自律運転参照アーキテクチャ



■ 行動決定の処理フロー

- ① 自車の推奨車線, 推奨速度, 走行可能領域などの前提条件を取得
- ② 自車の位置姿勢などの自車情報を取得
- ③ 静止障害物の位置姿勢と移動障害物の予測軌道を取得
- ④ 自車の位置姿勢や障害物との位置関係などから, 自車が置かれている状態を把握
- ⑤ 自車の置かれている状態できりうる運転行動群を定義
- ⑥ 運転行動群から1つの行動を選択

■ 行動プランナーによる運転行動決定の手法

- ✓ 状態遷移図による行動決定 (DUCにおける入賞チームで採用)
 - カーネギーメロン大学 Boss
 - スタンフォード大学 Junior
 - バージニア工科大学 Odin
 - マサチューセッツ工科大学 Talos
 - カールスルーエ工科大学 AnnieWAY
- ✓ 学習による行動決定 (DUC後)
 - Iseleらの方法
 - Mirchevskaらの方法

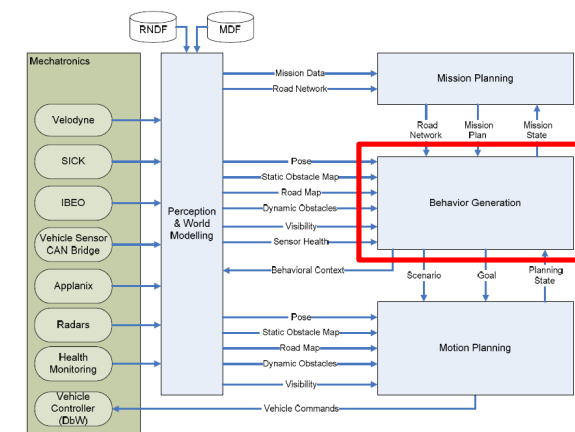
香月理絵, 荒井幸代, 大前学, 大日方五郎, 川崎敦史, 橘川雄樹, 小林祐一, 菅沼直樹, 田崎豪, 谷沢昭行, 新田修平, 野呂瀬琴, 馬場厚志, 藤吉弘巨, 目黒淳一, 森出茂樹, 谷口敦司, 山下倫央, 日本ロボット学会, 自動運転技術入門, オーム社, pp.110-111 (2021).

5. 行動プランナー；状態遷移による事例 (1)カーネギーメロン大学

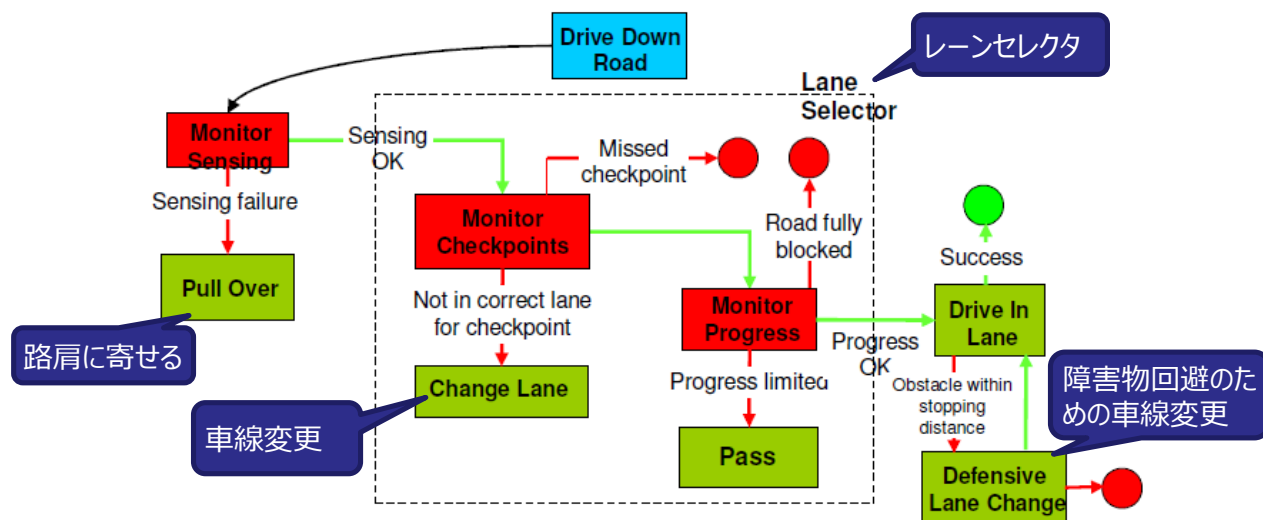
カーネギーメロン大学「Boss」の行動プランナーはState Machineで実現されている

トップレベルのState Machineと、以下の運転コンテキストに応じたSub State Machineの階層構造となっている

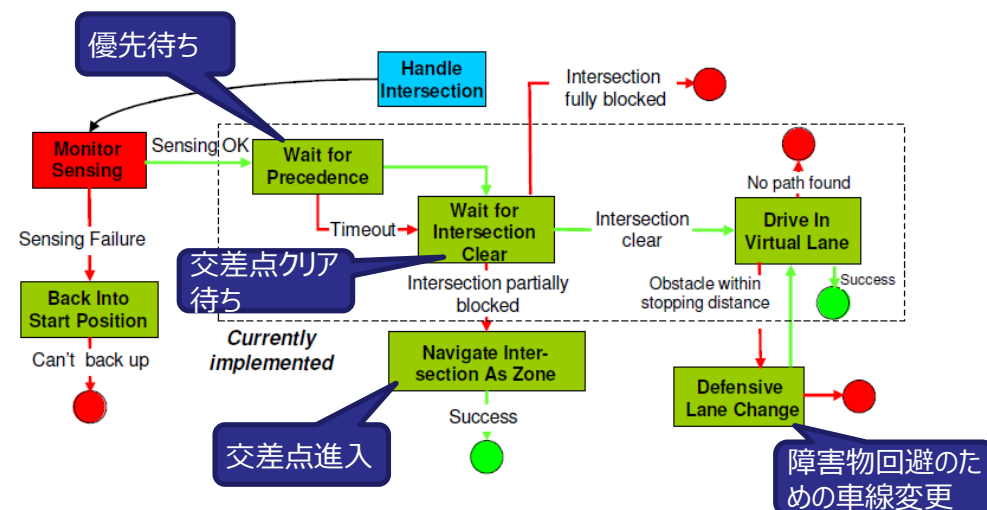
- 道路走行 (Drive Down Road)
- 交差点 (Handle Intersection)
- 駐車場等 (Achieve Zone Pose)



道路走行 (Drive Down Road)



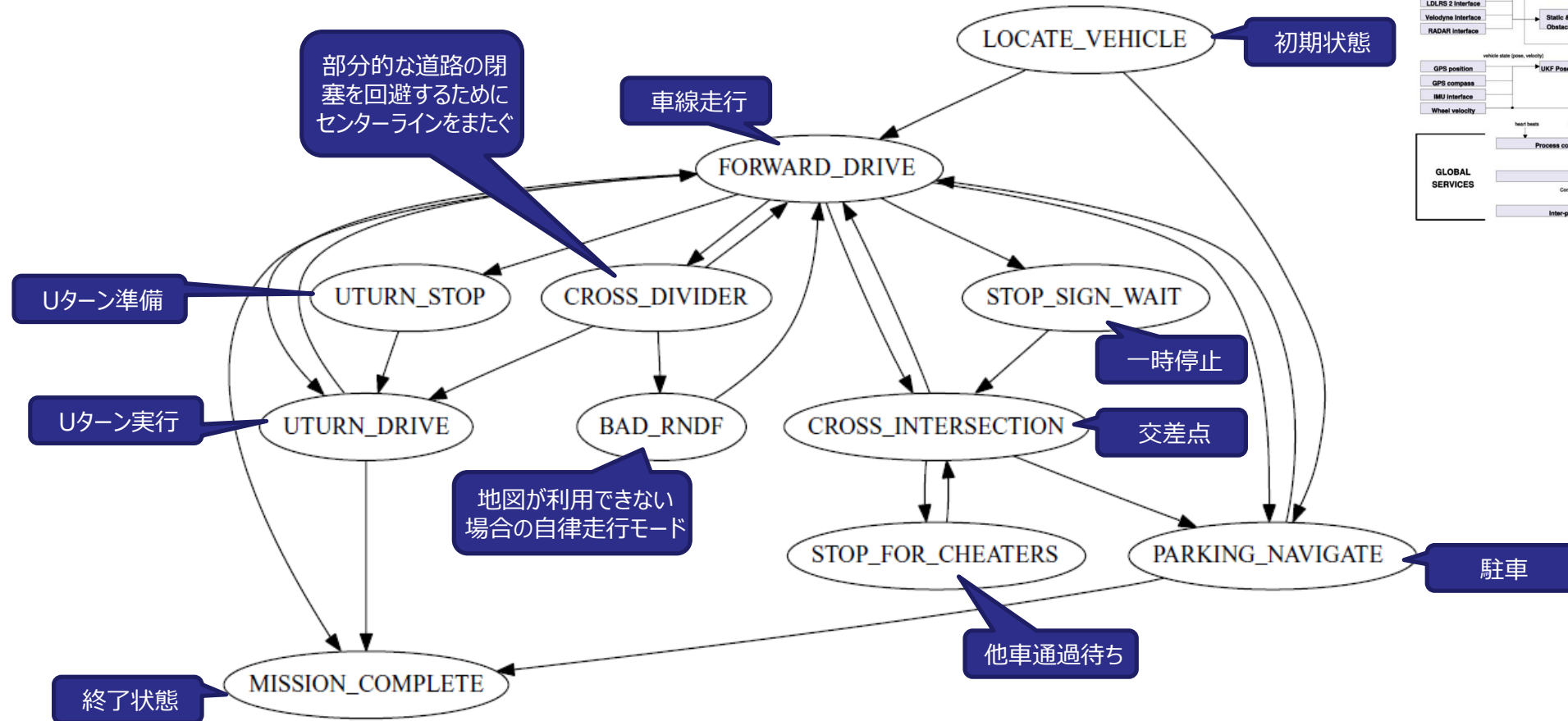
交差点 (Handle Intersection)



http://ri.cmu.edu/pub_files/2007/4/Tartan_Racing.pdf

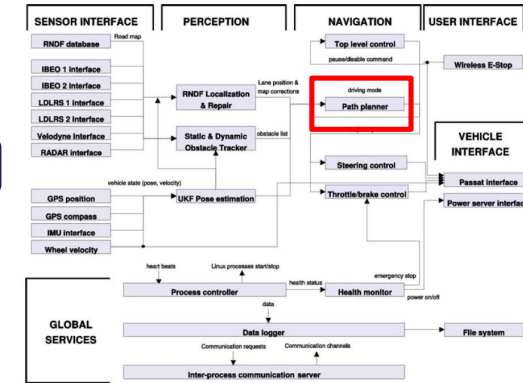
5. 行動プランナー；状態遷移による事例 (2)スタンフォード大学

スタンフォード大学「Junior」の行動プランナーは、FSM(Finite State Machine;有限状態機械)で実現されている



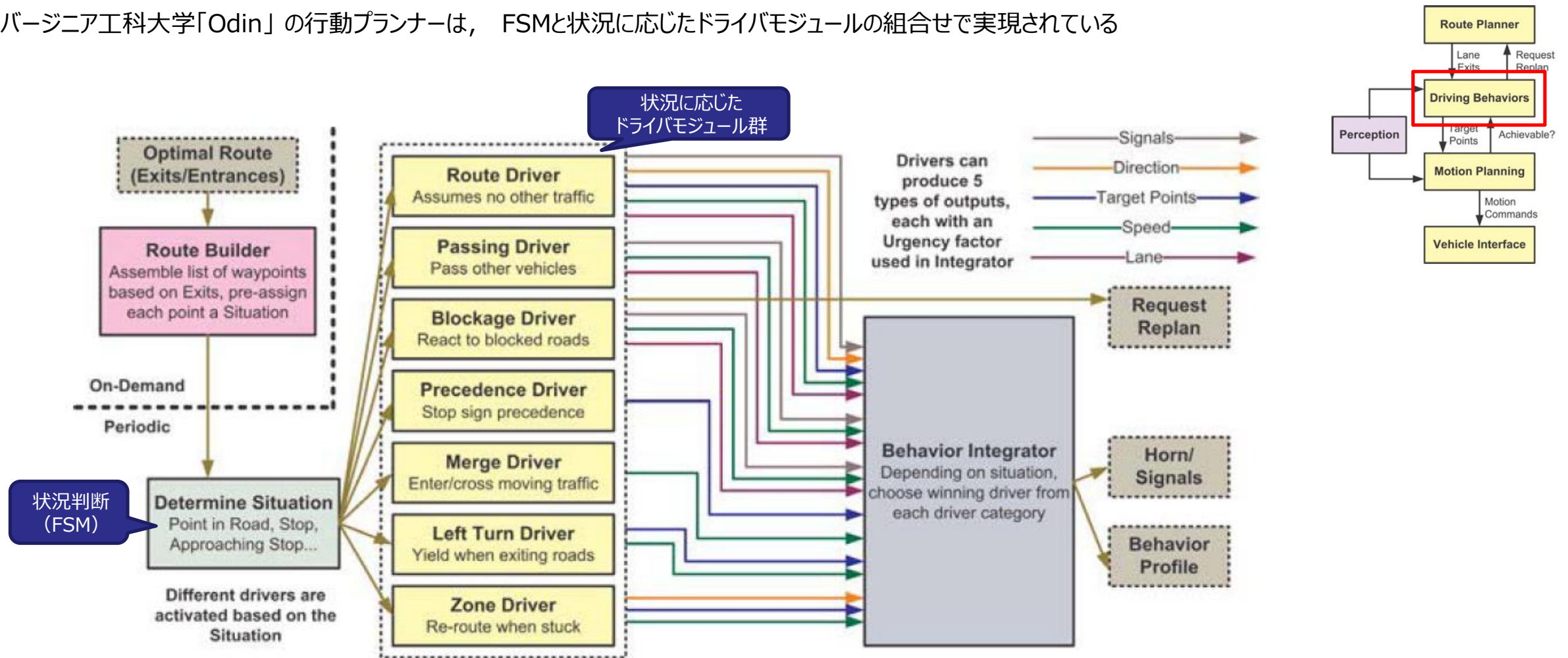
FSM

<https://onlinelibrary.wiley.com/doi/10.1002/rob.20258>



5. 行動プランナー；状態遷移による事例 (3)バージニア工科大学

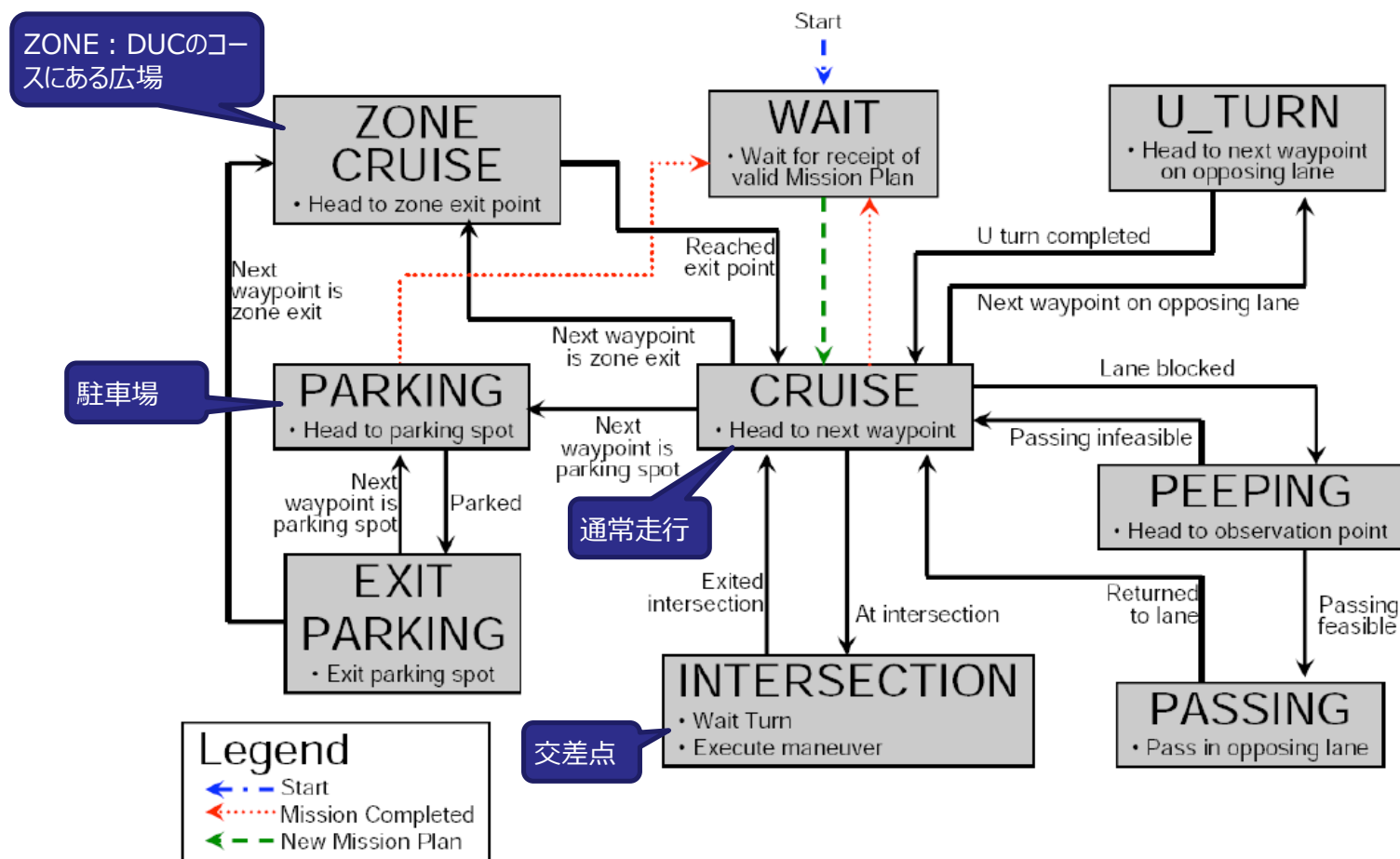
バージニア工科大学「Odin」の行動プランナーは、FSMと状況に応じたドライバモジュールの組合せで実現されている



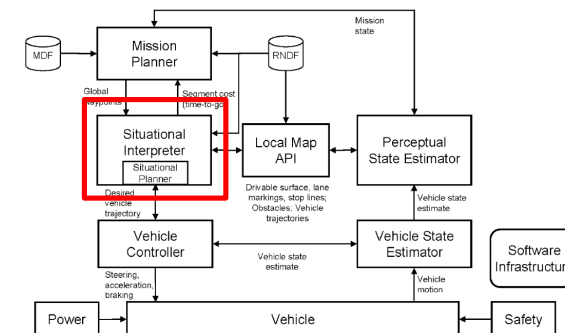
<http://www.romela.org/wp-content/uploads/2015/05/Odin-Team-VictorTango's-Entry-in-the-DARPA-Urban-Challenge.pdf>

5. 行動プランナー；状態遷移による事例 (4)マサチューセッツ工科大学

マサチューセッツ工科大学「Talos II」の行動プランナーはFSMで実現されている



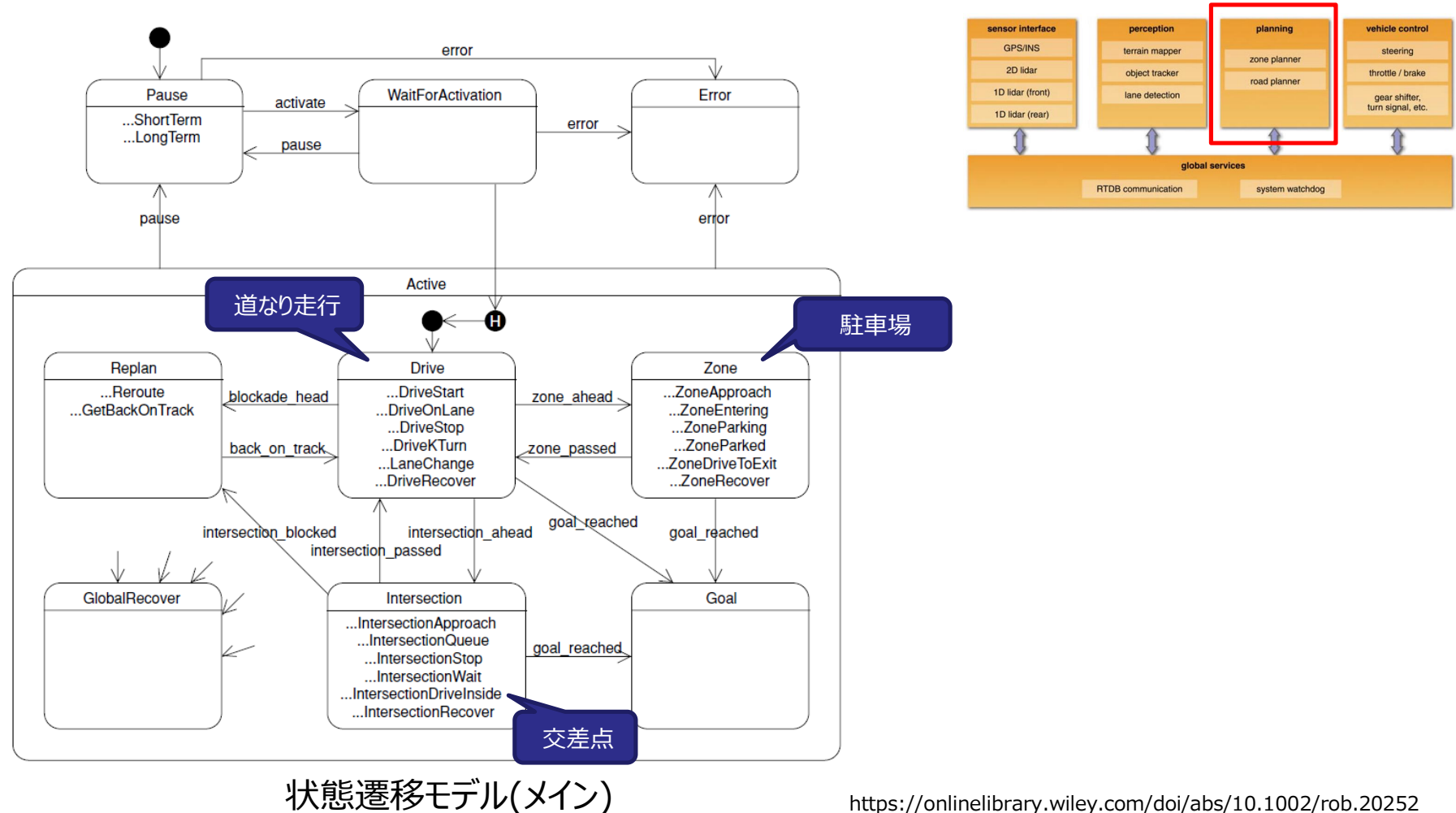
状態遷移図 (最上位)



<https://dspace.mit.edu/handle/1721.1/39822>

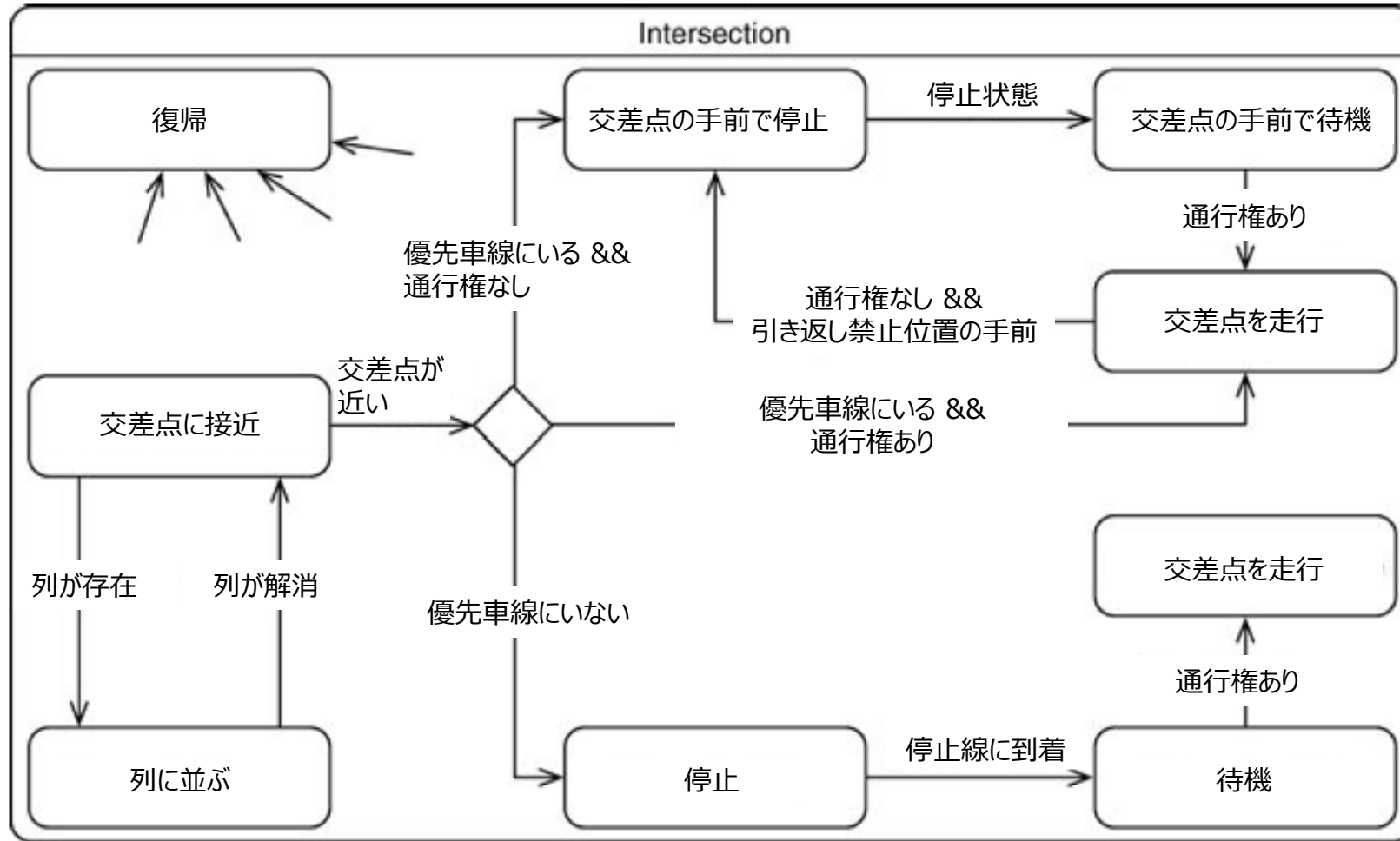
5. 行動プランナー；状態遷移による事例 (5)カールスルーエ工科大学

カールスルーエ工科大学「AnnieWAY」の行動計画(Mission Planningモジュール)は、階層型状態遷移モデル(HSM)で実現されている



<https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20252>

5. 行動プランナー；状態遷移による事例 (5)カールスルーエ工科大学



状態遷移モデル(サブステート：交差点)

<https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20252>

5. 行動プランナー; 学習による手法 Q学習について

■ 行動モデル化

- ✓ プレイヤー(エージェント)の状態(s)と行動(a)を定義する
- ✓ エージェントが行動を行うと別の状態に遷移するとともに, 別途定義した報酬(r)が得られる.

■ 行動最適化

- ✓ 各状態における行動の価値(Q値)を表にしたものをQ-tableと呼ぶ. プレイヤーはQ値が最も高くなる行動をとる
- ✓ Q学習とは, 設定されたゴール状態までに, 最大の累積報酬が得られるようにQ-tableを学習することである
- ✓ 学習は試行を繰り返し収束するまで行うが, 逐次行動価値を更新しながら試行するため, モンテカルロ法より収束が早い

Q-table

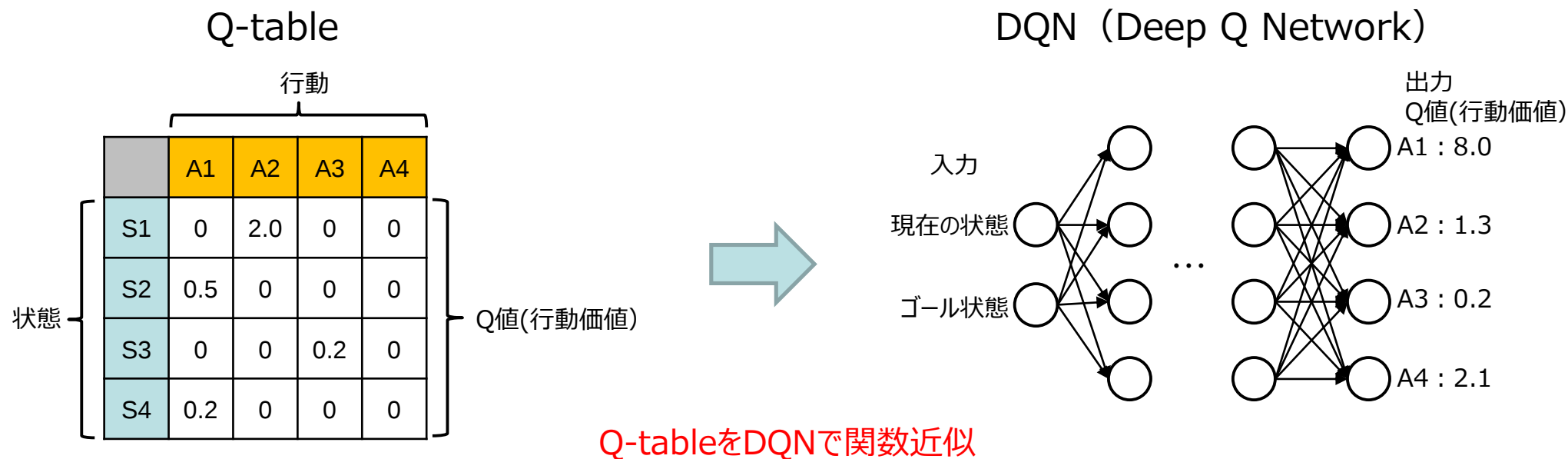
		行動			
		A1	A2	A3	A4
状態	S1	0	2.0	0	0
	S2	0.5	0	0	0
	S3	0	0	0.2	0
	S4	0.2	0	0	0

Q値(行動価値)

- ✓ <https://qiita.com/ishizakiiii/items/5eff79b59bce74fdca0d#q-learning>
- ✓ https://www.tcom242242.net/entry/ai-2/%E5%BC%B7%E5%8C%96%E5%AD%A6%E7%BF%92/%E6%B7%B1%E5%B1%A4%E5%BC%B7%E5%8C%96%E5%AD%A6%E7%BF%92/%E3%80%90%E6%B7%B1%E5%B1%A4%E5%BC%B7%E5%8C%96%E5%AD%A6%E7%BF%92%E3%80%91deep_q_network_%E3%82%92tensorflow%E3%81%A7%E5%AE%9F%E8%A3%85/

5. 行動プランナー; 学習による手法 DQN(Deep Q Network)について

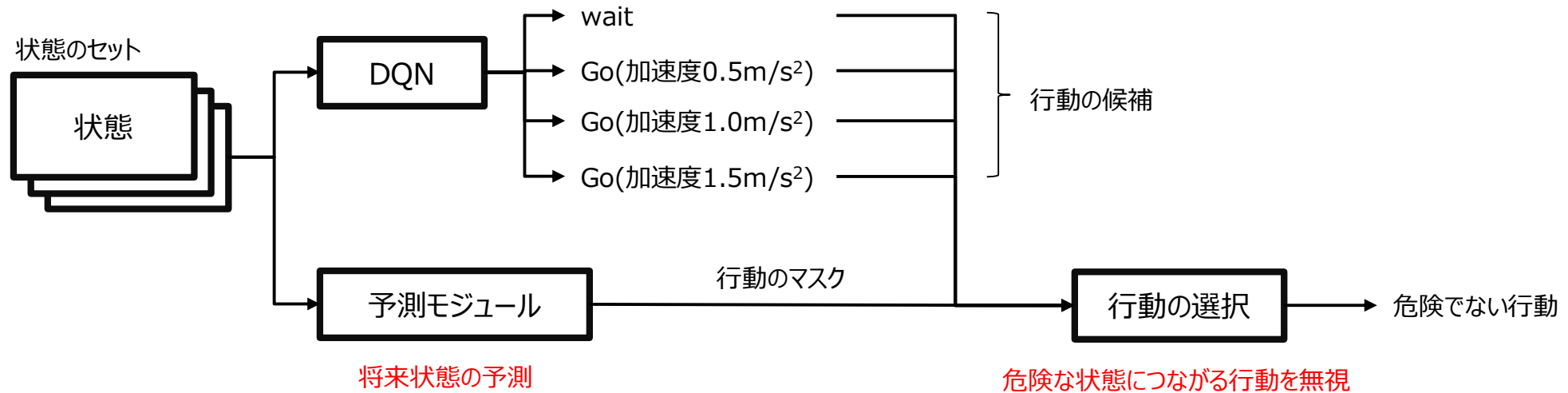
- ✓ Q-tableでは, 行動と状態数が膨大な時に必要なメモリ空間が大きくなりすぎてしまう
- ✓ Q-tableをニューラルネットワークDQN(Deep Q Network)で関数近似することにより, メモリ空間の爆発問題を防ぐ
- ✓ 学習は教師データなしで試行を繰り返し, 報酬値の変化を観測することで最適化される



✓ https://twitter.com/ImAI_Eruel/status/1303677795806056451/photo/1

5. 行動プランナー；学習による事例 (1)Iseleらの方法

- ✓ DQNを使用した強化学習による行動計画を無信号交差点の左折をモデルケースとして提案
- ✓ 他車と衝突しない安全な運転行動を出力できるように，探索空間に拘束をかけるための予測モジュールを追加

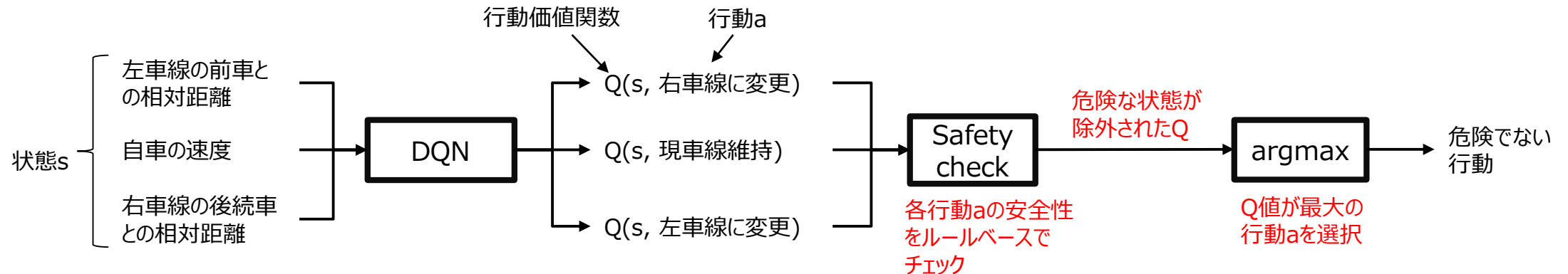


✓ <https://arxiv.org/abs/1910.00399>

✓ 香月理絵, 荒井幸代, 大前学, 大日方五郎, 川崎敦史, 橋川雄樹, 小林祐一, 菅沼直樹, 田崎豪, 谷沢昭行, 新田修平, 野呂瀬琴, 馬場厚志, 藤吉弘亘, 目黒淳一, 森出茂樹, 谷口敦司, 山下倫央, 日本ロボット学会, 自動運転技術入門, オーム社, pp.110-111 (2021).

5. 行動プランナー；学習による事例 (2)Mirchevskaらの方法

- ✓ DQNを使用した強化学習による行動計画を高速道路での車線変更をモデルケースとして提案
- ✓ 強化学習とルールの併用

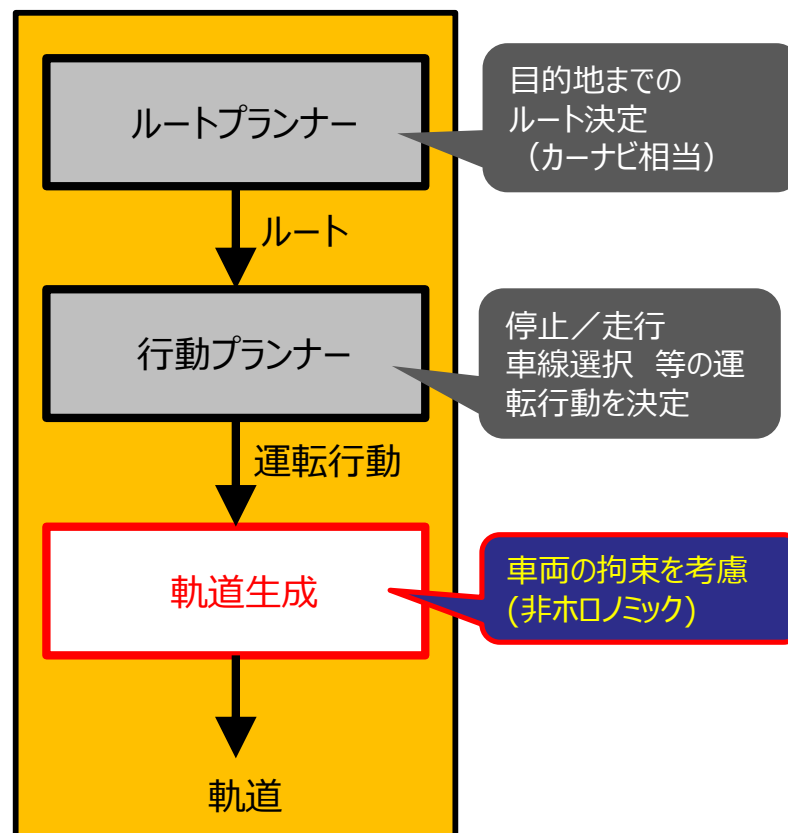


- ✓ https://www.researchgate.net/publication/327690711_High-level_Decision_Making_for_Safe_and_Reasonable_Autonomous_Lane_Changing_using_Reinforcement_Learning
- ✓ 香月理絵, 荒井幸代, 大前学, 大日方五郎, 川崎敦史, 橋川雄樹, 小林祐一, 菅沼直樹, 田崎豪, 谷沢昭行, 新田修平, 野呂瀬琴, 馬場厚志, 藤吉弘亘, 目黒淳一, 森出茂樹, 谷口敦司, 山下倫央, 日本ロボット学会, 自動運転技術入門, オーム社, pp.110-111 (2021).

6. 軌道生成

軌道生成モジュールは車両の拘束（非ホロノミック拘束、車は急に曲がれない）を考慮した軌道を生成する。

自律運転参照アーキテクチャ



軌道：時刻をパラメータとした位置と姿勢

香月理絵, 荒井幸代, 大前学, 大日方五郎, 川崎敦史, 橘川雄樹, 小林祐一, 菅沼直樹, 田崎豪, 谷沢昭行, 新田修平, 野呂瀬琴, 馬場厚志, 藤吉弘巨, 目黒淳一, 森出茂樹, 谷口敦司, 山下倫央, 日本ロボット学会, 自動運転技術入門, オーム社, pp.110-111 (2021).

6. 軌道生成；制御方式一覧①

手法	説明	メリット	デメリット
PID制御	古典制御（1入力 1 出力）の最も基本的なもので現在でも様々なところで使用されている Proportional：比例 ⇒ 現在の目標との偏差により制御 Integral：積分 ⇒ 過去の偏差の累積により制御 Differential: 微分 ⇒ 偏差の変化率より未来を予測して制御	システムのモデルが全く分からなくてもパラメータチューニング（試行錯誤的な調整）によってそこそこの性能を出しやすい 高速な計算が可能	モデルを使用できないため車両の拘束を考慮した精密な制御が行えない 1入力 1 出力であるため様々な制約を考慮できない
Pure pursuit	古典制御の一種で、操舵制御のための手法。一定距離先の前方注視点を目標に、その点に到達するような旋回制御を実現する。PD制御のD(微分)が局所的な傾きをもとに未来を線形に予測して補正をかける代わりに、前方注視点という未来予測に基づいて補正をかける。DUCで多くのチームが採用し、Autowareも採用している。類似手法として「ヨーレイト追従制御」がある。	微分が不要 高速な計算が可能	同上
H_{∞} 制御	ロバスト制御の一種で、数式モデルに含まれる「不確かさ」の範囲を考慮し、想定される最悪の場合でも安定した結果が得られるように制御する手法。	誤差や外乱などの不確かさに強い	不確かさを考慮する必要があるため、設計難度が高い 保守的な制御になるため、最高性能が出せない
Dynamic Window Approach	現在の速度と運動モデルから次の時刻までに取りうる制御入力の範囲(Dynamic Window)を計算し、その範囲内の制御入力をサンプリングして、事前に決めた評価関数が最大になる制御入力を計算する。ROSのlocal_plannerで採用されている。	高速な計算が可能 車両モデルを取り入れることができる パラメータをチューニングしやすい	一定制御入力という仮定があるため、複雑な走行ができない

- ✓ <https://controlabo.com/control-types/>
- ✓ https://qiita.com/fumiya_sato/items/55a8163bca31f7061066
- ✓ <https://myenigma.hatenablog.com/entry/20140624/1403618922>
- ✓ https://qiita.com/taka_horibe/items/47f86e02e2db83b0c570
- ✓ <https://tech.tier4.jp/entry/2019/03/29/173543>
- ✓ 香月理絵, 荒井幸代, 大前学, 大日方五郎, 川崎敦史, 橘川雄樹, 小林祐一, 菅沼直樹, 田崎豪, 谷沢昭行, 新田修平, 野呂瀬琴, 馬場厚志, 藤吉弘亘, 目黒淳一, 森出茂樹, 谷口敦司, 山下倫央, 日本ロボット学会, 自動運転技術入門, オーム社, pp.139-153 (2021).
- ✓ 二宮芳樹, 武田一哉, 自動運転, コロナ社, pp.143-153 (2021).

6. 軌道生成；制御方式一覧②

手法	説明	メリット	デメリット
線形二次レギュレータ (Linear-Quadratic Regulator:LQR)	最適制御の代表的な手法で、制御目的を達成するための最適な制御入力を計算する手法。 制御目的を評価関数で表現し、それを最小化する入力を導出する。	評価関数に対し最適な制御が可能 ゲイン決定をオフラインで実施可能	道路線形などの先読み情報を考慮した予測機能がないため、違和感のある遅れた車両挙動になる可能性がある
モデル予測制御 (Model Predictive Control:MPC)	現代制御に分類される手法。 制御対象のモデルに基づいて「制御周期ごとに最適化問題を解き」、その結果を使って制御する。 車両モデルを使用し、現在を開始点として、少し将来までの最適制御問題を制御中に解き続けることで最適フィードバック制御を実現する手法	制約条件を陽に扱うことができるため 制御性能を最大化できる 制約を考慮した最適な制御が可能 予測機能を持つため、フィードフォワード制御とフィードバック制御の利点を兼ね備えた制御を行える 制御中に目標軌道が変わっても対応できる	古典制御より計算量大 (線形MPCならばマイコンでも可能)
非線形MPC (HMPC)	非線形モデルのMPC。非線形システムに対して使用する。	操舵と車速同時制御などより高度な制御が可能	線形MPCよりさらに計算量が大 高性能CPUが必要
モンテカルロMPC (MCMPC)	MPCの一手法であり、モンテカルロ法で操作入力列の候補を生成し、それぞれの評価値の大小から、制御入力を決定する。	評価関数の勾配が不要 物体間の衝突のような不連続な事象にも対応できる 並列化が容易なため、リアルタイム制御を実現しやすい	モンテカルロ法に基づくため、結果に再現性がない 外乱に対して収束に時間がかかる傾向がある

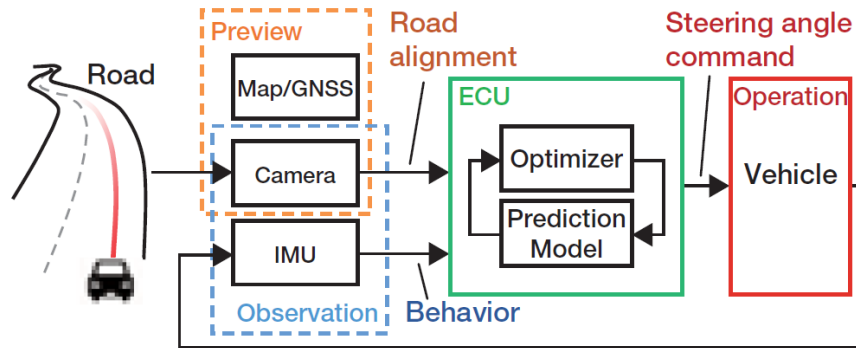
- ✓ <https://controlabo.com/control-types/>
- ✓ https://qiita.com/fumiya_sato/items/55a8163bca31f7061066
- ✓ <https://myenigma.hatenablog.com/entry/20140624/1403618922>
- ✓ https://qiita.com/taka_horibe/items/47f86e02e2db83b0c570
- ✓ <https://tech.tier4.jp/entry/2019/03/29/173543>
- ✓ 香月理絵, 荒井幸代, 大前学, 大日方五郎, 川崎敦史, 橘川雄樹, 小林祐一, 菅沼直樹, 田崎豪, 谷沢昭行, 新田修平, 野呂瀬琴, 馬場厚志, 藤吉弘亘, 目黒淳一, 森出茂樹, 谷口敦司, 山下倫央, 日本ロボット学会, 自動運転技術入門, オーム社, pp.139-153 (2021).
- ✓ 二宮芳樹, 武田一哉, 自動運転, コロナ社, pp.143-153 (2021).

6. 軌道生成; MPCの事例

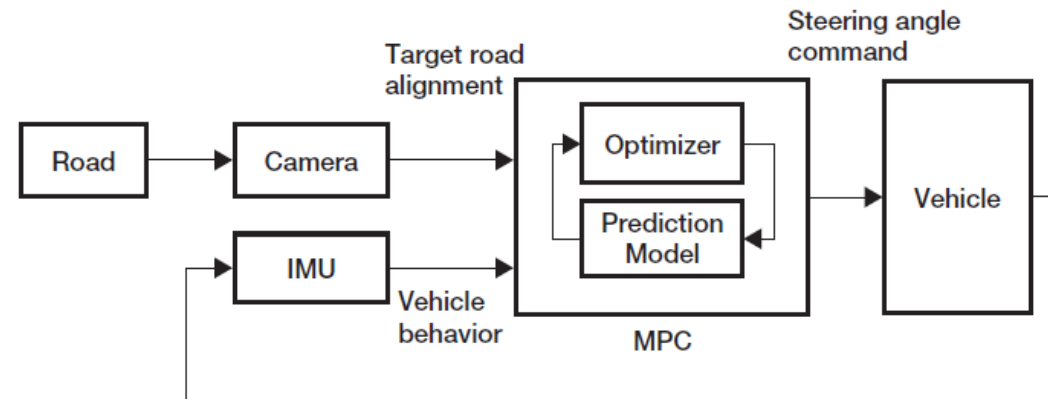
主体	発表年	説明	使用CPU	参考文献等
日立アステモ	2017	ACCの性能向上のために速度制御をPID制御からMPCに変更した。MATLABのModel Predictive Control Toolboxを使用して設計及びコード生成し、実車テスト中	Renesas SH-4A 32-bit processor	✓ https://jp.mathworks.com/company/user_stories/hitachi-astemo-develops-a-model-predictive-controller-for-adaptive-cruise-control-with-model-based-design.html ✓ https://in.mathworks.com/content/dam/mathworks/tag-team/Objects/m/FASTzero17_MPC4ACC_20170719.pdf
ホンダ (2017)	2017	3つのカメラのみを使用した一般道での自動運転制御を実験中。操舵制御にMPCを使用することで、従来手法の操舵遅れによるコーナーでの膨らみ／蛇行を改善している。		✓ https://web.motormagazine.co.jp/_ct/17086799
トヨタ自動車 入江ら	2022	熟練ドライバのような自動運転を目指し、MPCを利用した制御をMATLABのModel Predictive Control Toolboxを使用してシミュレーション。一般的な製品としてのハードウェア構成（カメラ, GNSS, IMU）を前提。	一般的なコンシューマ向きCPUを想定	✓ https://www.jstage.jst.go.jp/article/jsaeijae/12/1/12_20214210/_article/-char/ja/ ✓ 自動車技術 Vol.76, pp.16-24
三菱電機	2022	NMPCを用いて、車両の縦方向と横方向の運動を統合して制御するコントローラーを開発。	dSPACE MicroAutoBoxII (IBM PPC 750 GL, 900MHz)	✓ https://www.mitsubishielectric.co.jp/corporate/randd/list/mechatronics/b236/index.html ✓ 自動車技術 Vol.76, pp.10-15

6. 軌道生成; MPCの事例 トヨタ自動車 入江ら

- ✓ MPCを利用して「うまい運転」を自動運転で実現する
- ✓ 製品への実装を意識した簡素なハードウェア構成（カメラ, 地図, GNSS, IMU）を想定
- ✓ 簡素化のために縦方向制御(車速)と横方向制御(操舵)を分業し, 横方向制御のみに焦点をあてる（予測区間で車速一定と仮定）
→非線形な車両ダイナミクスモデルを線形モデル化（計算負荷の抑制）



システム構成

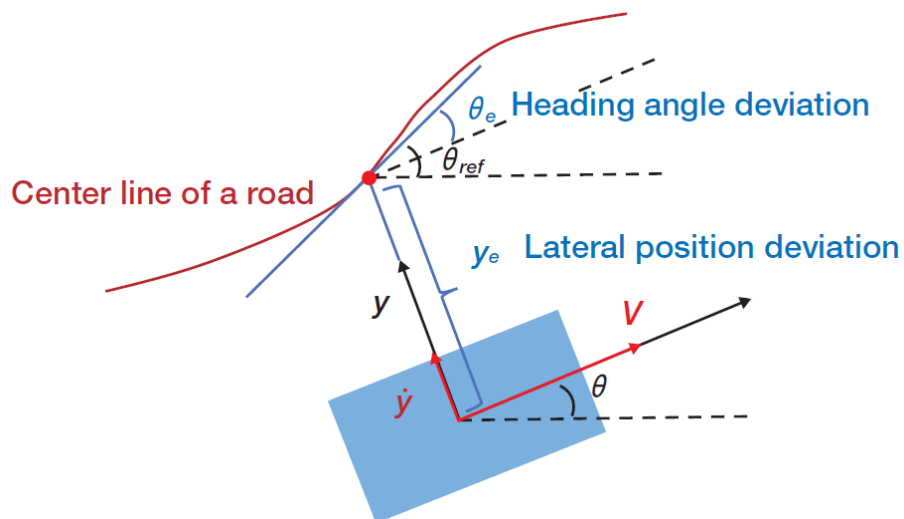


操舵角制御系の構成

6. 軌道生成; MPCの事例 トヨタ自動車 入江ら

予測モデル（車線中心線と自車の関係を定式化）

予測モデルの座標



予測モデルの状態量

$$x_e := \begin{bmatrix} \dot{\theta}_e \\ \theta_e \\ \dot{y}_e \\ y_e \\ \delta \\ z \end{bmatrix} = \begin{bmatrix} \dot{\theta} - \dot{\theta}_{ref} \\ \theta - \theta_{ref} \\ \dot{y} - V\theta_e \\ y - y_{ref} \\ \delta \\ \int y_e \end{bmatrix}$$

車線の中心線を基準とした偏差(角度, 距離)

- 車両の向きと道路の向きの差の微分値
- 車両の向きと道路の向きの差
- 車線中央と車両の横方向偏差の微分値
- 車線中央と車両の横方向偏差
- 舵角
- y_e の積分値（外乱に対するロバスト性のため）

状態方程式

$$\begin{cases} \dot{x}_e = A_e x_e + B_e u + G_e \dot{\theta}_{ref} \\ y_m = C_e x_e + H_e \dot{\theta}_{ref} \end{cases} \quad A_e, B_e, G_e, C_e, H_e \text{は係数行列}$$

$$y_m = \begin{bmatrix} x_e^T & \dot{\theta} \end{bmatrix}^T = \begin{bmatrix} \dot{\theta}_e & \theta_e & \dot{y}_e & y_e & \delta & z & \dot{\theta} \end{bmatrix}^T$$

観測出力 ヨーレイト

6. 軌道生成; MPCの事例 トヨタ自動車 入江ら

最適化問題

$$\begin{aligned}
 & \min_u \int_t^{t+T_p} \left(\sum_{i=1}^7 \underset{\substack{\uparrow \\ \text{重み}}}{q_i} y_{mi}^2 + \underset{\substack{\uparrow \\ \text{重み}}}{r_{\dot{u}}} \dot{u}^2 + \underset{\substack{\uparrow \\ \text{重み}}}{r_u} u^2 \right) d\tau \\
 & \text{s.t.} \\
 & y_{emin} \leq y_e \leq y_{emax} \quad \text{車線幅による制約} \\
 & u_{min} \leq u \leq u_{max} \quad \text{操舵角指令の制約} \\
 & \dot{u}_{min} \leq \dot{u} \leq \dot{u}_{max} \quad \text{操舵角の指令の変化率の制約}
 \end{aligned}$$

QPソルバで最適化

最適解

$$u^* = [u_t^* \ u_{t+1}^* \ \cdots \ u_{t+p-1}^*]^T$$

↓
最初の要素を操舵角制御入力

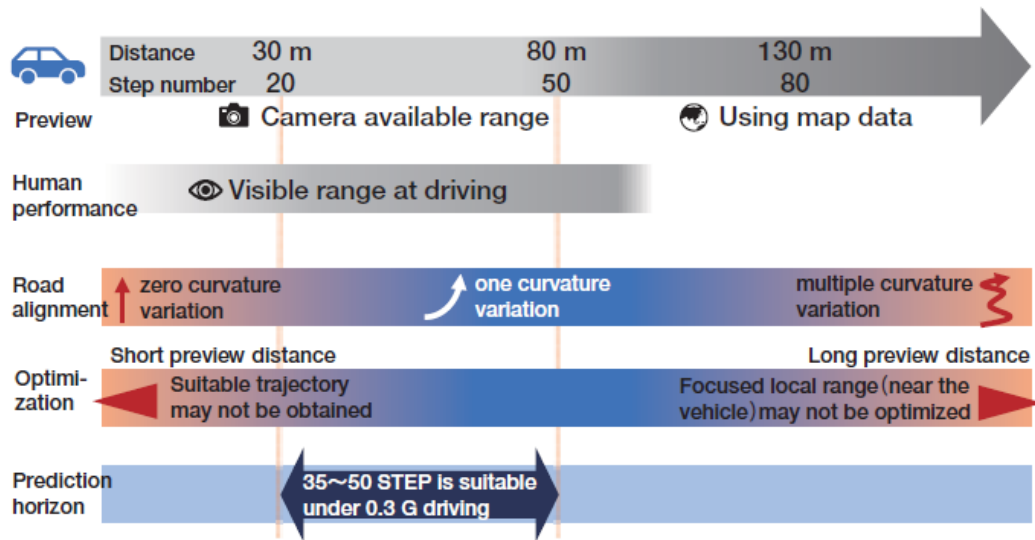
評価関数

$$\int_t^{t+T_p} (q_1 \dot{\theta}_e^2 + q_2 \theta_e^2 + q_3 \dot{y}_e^2 + q_4 y_e^2 + q_5 \delta^2 + q_6 z^2 + q_7 \dot{\theta}^2 + r_{\dot{u}} \dot{u}^2 + r_u u^2) d\tau$$

目的	変数名	効果
ヨーレイト制御	$\dot{\theta}_e$	旋回中に車線中心線に近づける
	θ_e	車線中心線の接線に車両方位角を向ける
	$\dot{\theta}$	横加速度の低減のためにヨーレイトを抑制する
横位置制御	$\dot{y}_e$	横変位の急激な変化を抑制する
	y_e	旋回および直進しながら車両を中心線に近づける
スムーズ性	δ	横方向の車両挙動を滑らかにする
外乱抑制	z	外乱の影響を抑える
予測区間	T_p	先読み距離による目標軌道と計算コストの適正化をする

6. 軌道生成; MPCの事例 トヨタ自動車 入江ら

予測区間



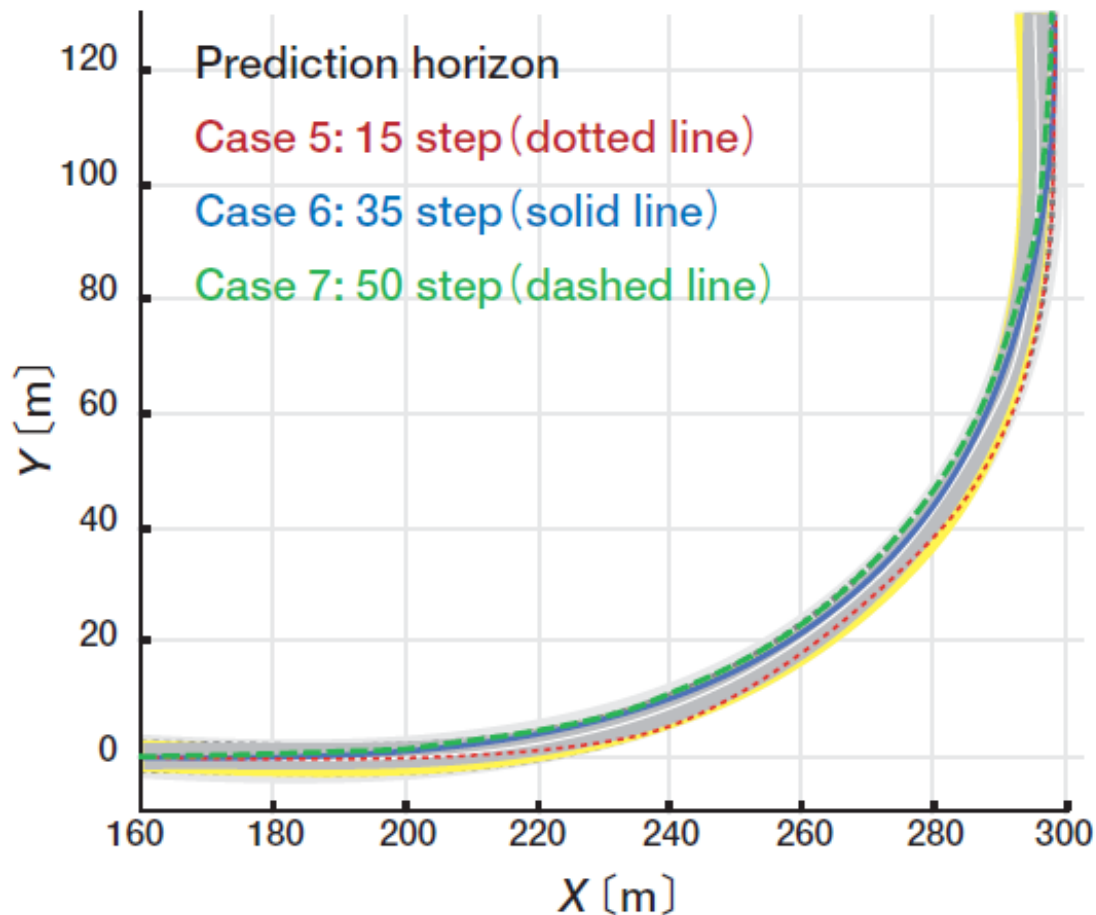
車速60km/h, 制御周期0.1sとする

Case 5 : 15step (1.5s) $\div$ 25 m

Case 6 : 35step (3.5s) $\div$ 58.3 m

Case 7 : 50step (5.0s) $\div$ 83.3 m

予測区間による生成軌道の相違



6. 軌道生成; MPC関連ツール

- MATLAB Model Predictive Control Toolbox

<https://jp.mathworks.com/products/model-predictive-control.html>

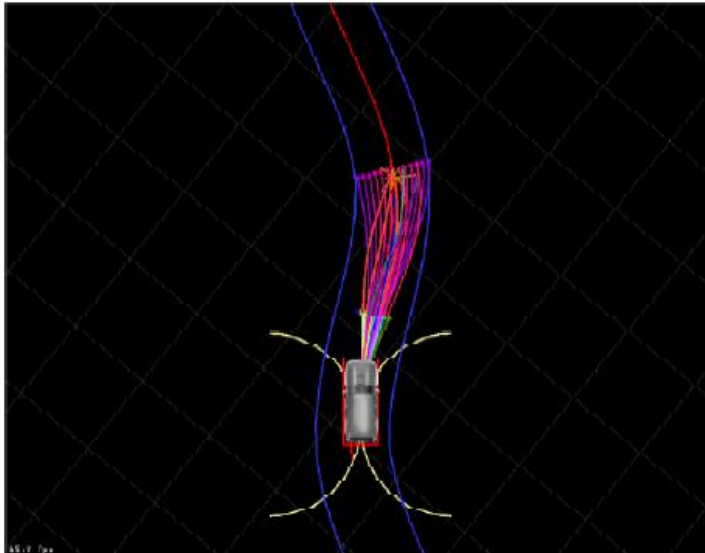
- Autoware mpc_follower

https://github.com/autowarefoundation/autoware_ai_planning/tree/master/mpc_follower

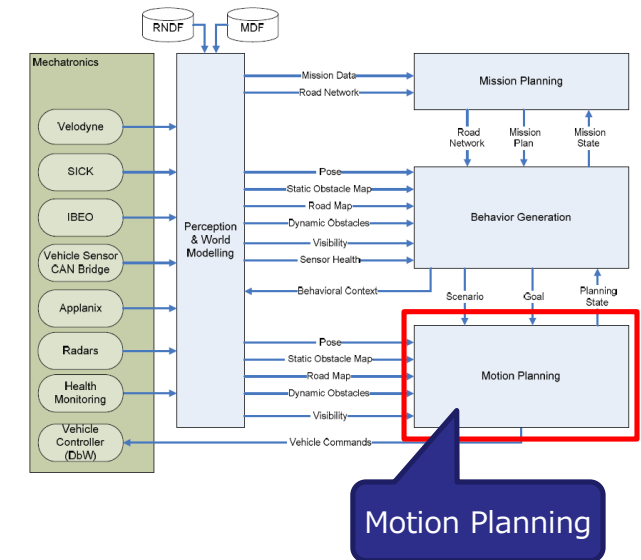
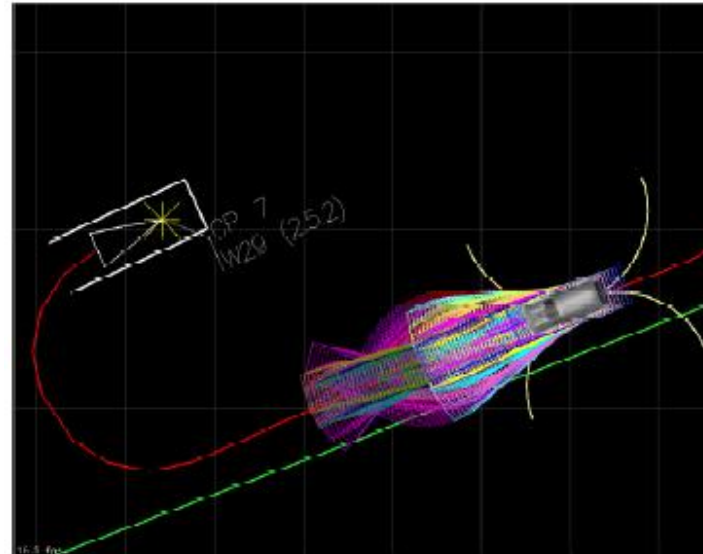
6. 軌道生成; 事例 (1)カーネギーメロン大学

カーネギーメロン大学「Boss」の軌道生成部は「Motion Planning」モジュールと呼ばれている
Model Predictive Trajectory Generation(MPTG)によって軌道を生成する
軌道を生成する際のポイントのサンプリングにLattice Plannerを使用

車線走行時



道路未整備ゾーン走行時



https://www.ri.cmu.edu/pub_files/2007/4/Tartan_Racing.pdf

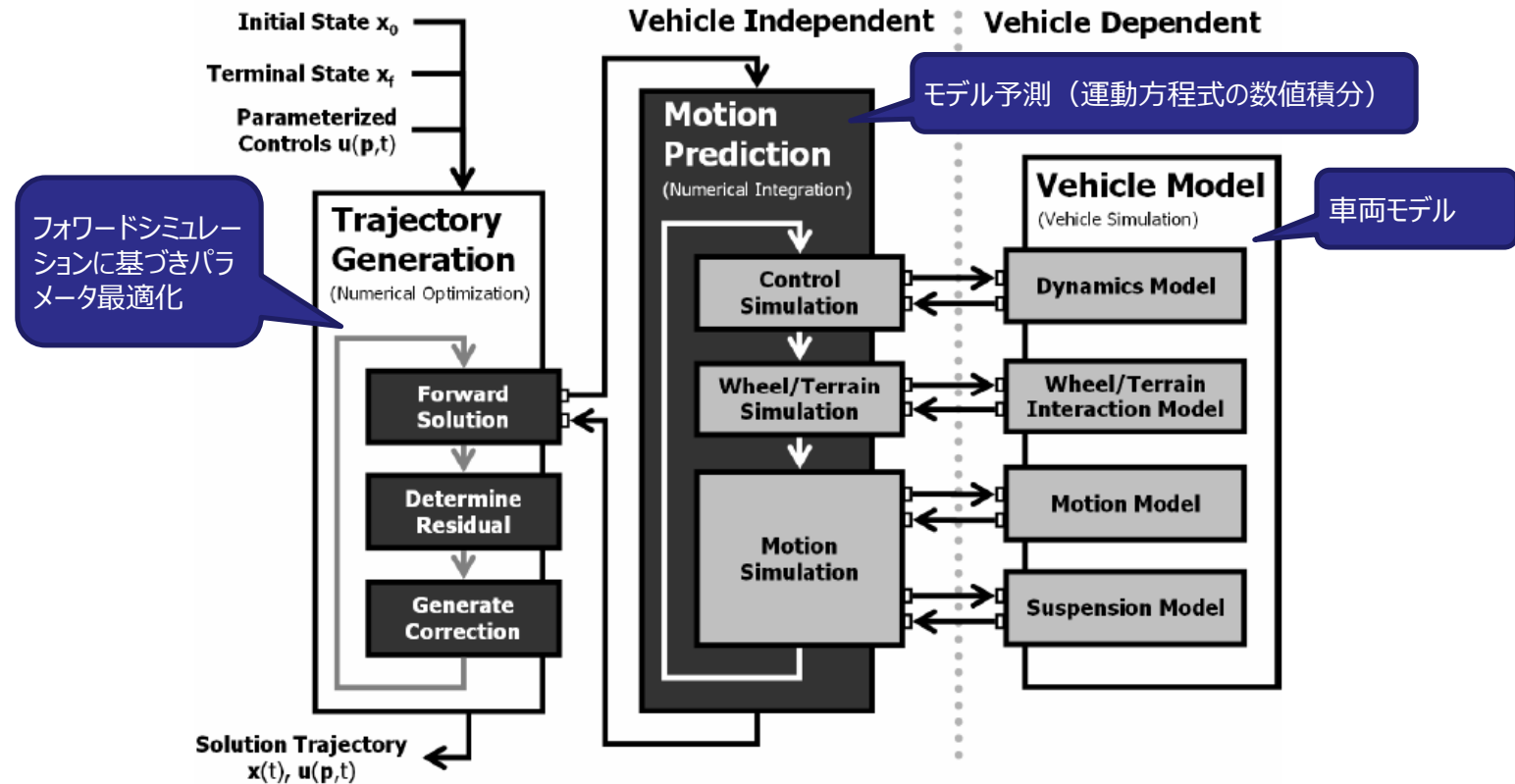
<https://www.ri.cmu.edu/publications/adaptive-model-predictive-motion-planning-for-navigation-in-complex-environments/>

6. 軌道生成; 事例 (1)カーネギーメロン大学

Model Predictive Trajectory Generation(MPTG)

アーキテクチャ

- ✓ カーネギーメロン大学 (CMU) のHowardとKellyによって開発された軌道生成アルゴリズム
- ✓ ローカル目標への実行可能な軌道を計算するために、**フォワードシミュレーション**と忠実度の高い**車両モデル**を使用して、予定軌道をたどったあとの自車位置を予測し、誤差が最小になるように最適化された軌道を生成する (MPCに近い考え方)
- ✓ もともとはNASAの火星探査機用として開発

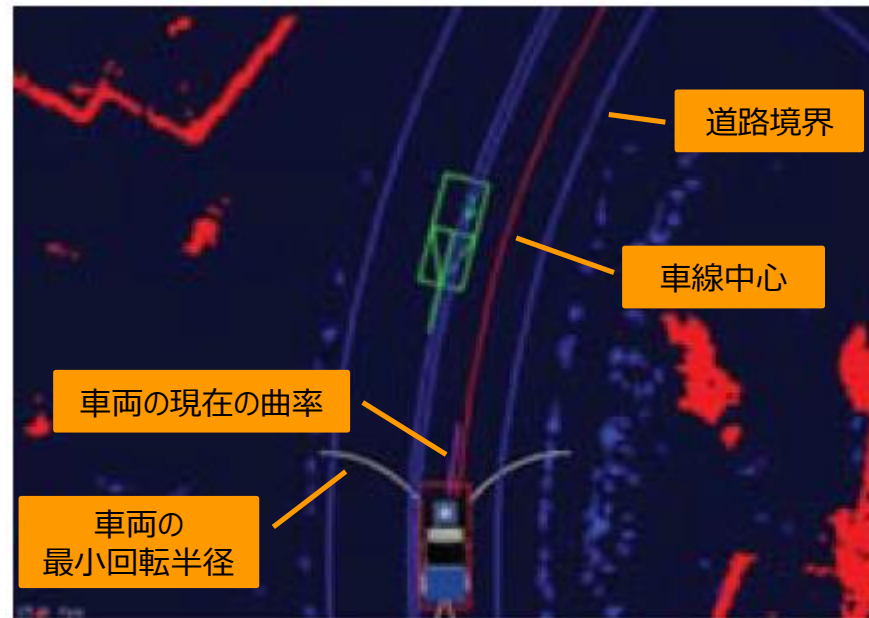


<https://www.ri.cmu.edu/publications/optimal-rough-terrain-trajectory-generation-for-wheeled-mobile-robots/>

6. 軌道生成; 事例 (1)カーネギーメロン大学

車線走行時の軌道生成方法

車両及び環境の現在の状態



軌道生成後



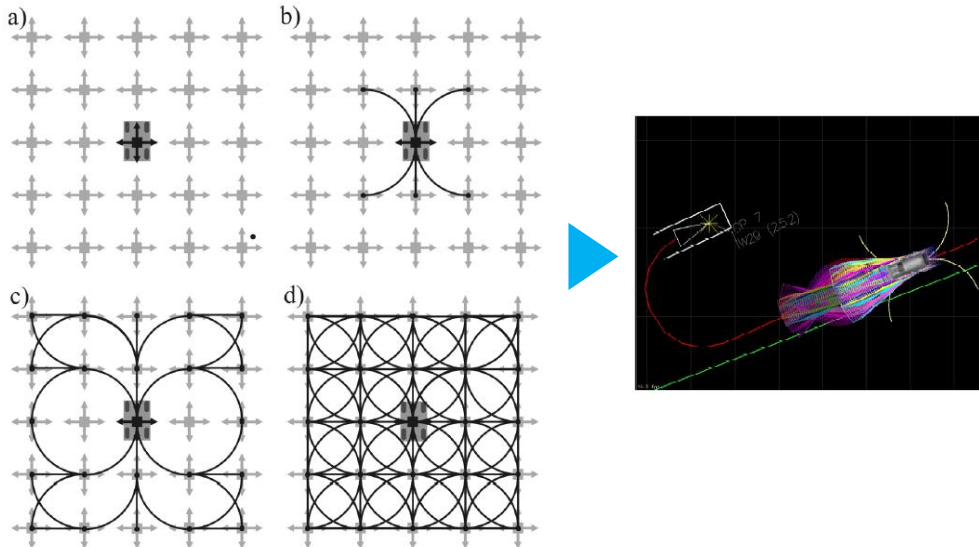
https://www.ri.cmu.edu/pub_files/2007/4/Tartan_Racing.pdf

<https://www.ri.cmu.edu/publications/adaptive-model-predictive-motion-planning-for-navigation-in-complex-environments/>

6. 軌道生成; 事例 (1)カーネギーメロン大学

車線未整備ゾーン走行時の軌道生成方法「State Lattice Planner」

- State Lattice(状態格子)と呼ばれる検索空間を導入
- サンプリングしたノードで格子を構成
- 各ノードは4次元(x, y, 向き, 曲率)の状態を持つ
- パスにはコストを割当て可能
- パス決定後の軌道はMPTGで生成

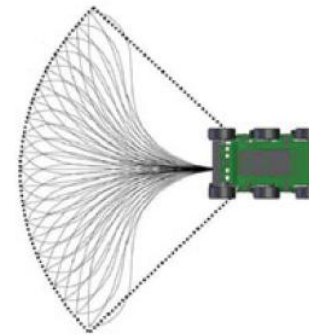


一部の計算をオフライン化することにより高速化が可能
(計算済みデータをあらかじめ組み込んでおく)

Latticeノードのサンプリング手法 (状況に応じてサンプリング手法を選択する)

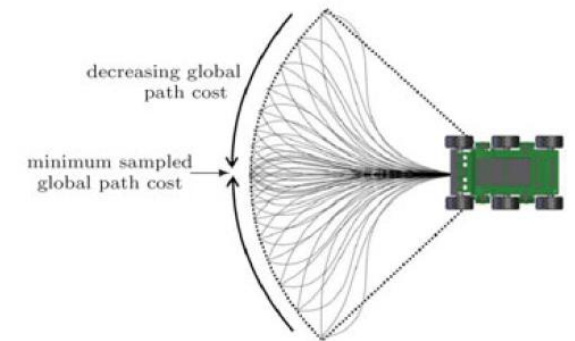
Uniform terminal state sampling

扇形の弧状に等間隔にサンプリング



Focused terminal state sampling

目標の方向を重点的にサンプリング



その他, 車線に沿ったサンプリング方法もある

- ✓ https://www.ri.cmu.edu/pub_files/2005/9/ISAIRAS05.pdf
- ✓ http://vigir.missouri.edu/~gdesouza/Research/Conference_CDs/IEEE_IROS_2009/papers/0877.pdf
- ✓ https://www.ri.cmu.edu/pub_files/pub4/howard_thomas_2008_1/howard_thomas_2008_1.pdf
- ✓ <https://myenigma.hatenablog.com/entry/2017/07/21/115833>

6. 軌道生成; Lattice Planner実装例

- Apollo

<https://github.com/ApolloAuto/apollo/tree/master/modules/planning/planner>

- Autoware

https://github.com/autowarefoundation/autoware_ai_planning/tree/master/lattice_planner

- ROS (個人)

https://github.com/larics/lattice_planner

- Python (個人のサンプルコード, MPTGのサンプルもあり)

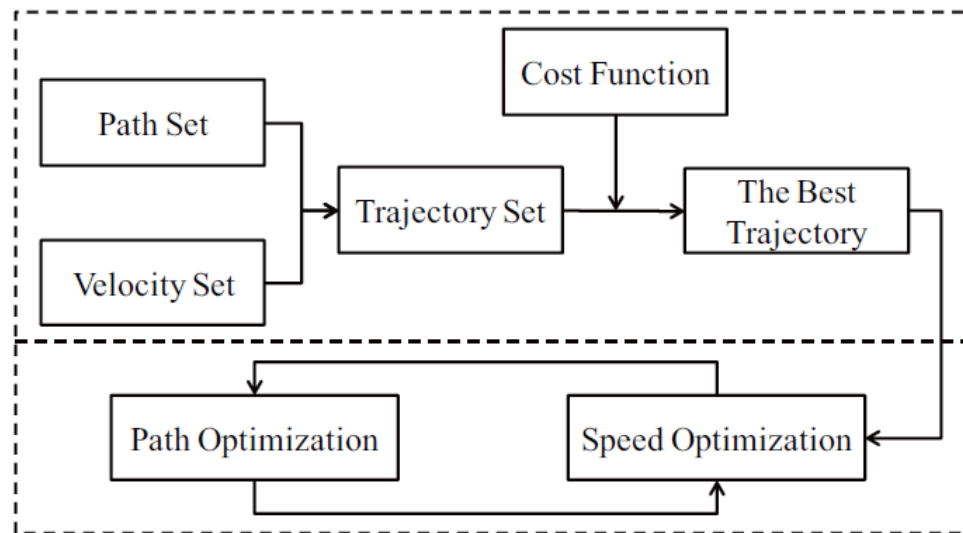
<https://github.com/AtsushiSakai/PythonRobotics/tree/master/PathPlanning/StateLatticePlanner>

6. 軌道生成; 事例 (1)カーネギーメロン大学

DUC後のWenda Xuらの事例（北京大学とカーネギーメロン大学の共同研究）

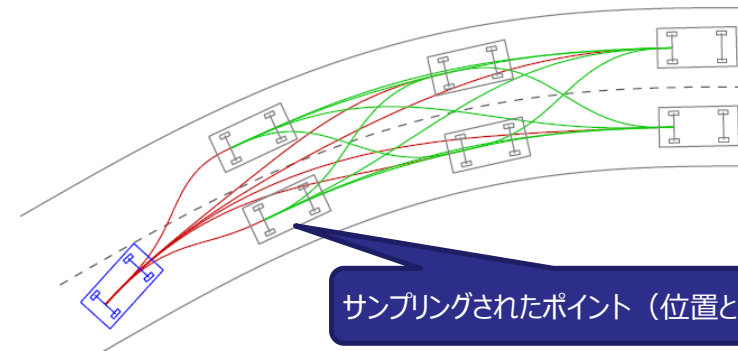
3次多項式または4次多項式でポイント間を結合してなめらかな軌道を生成

2パート軌道生成フレームワーク



パスと速度を別々に軌道候補を生成し、
コスト関数でベストな軌道を選定した後両者を最適化

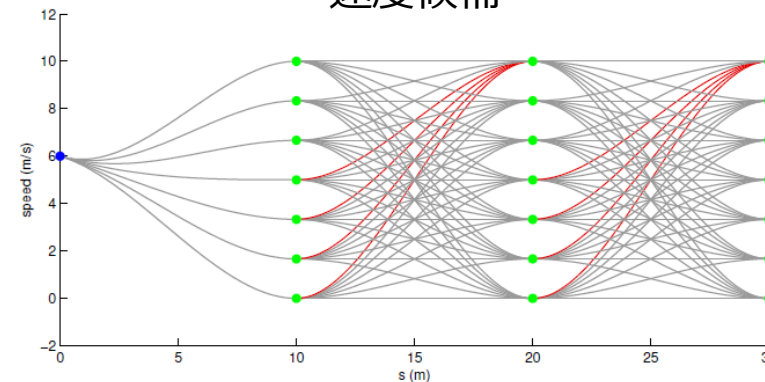
パス候補



緑：3次多項式
赤：4次多項式
※4次は計算負荷大

サンプリングされたポイント（位置と姿勢）

速度候補



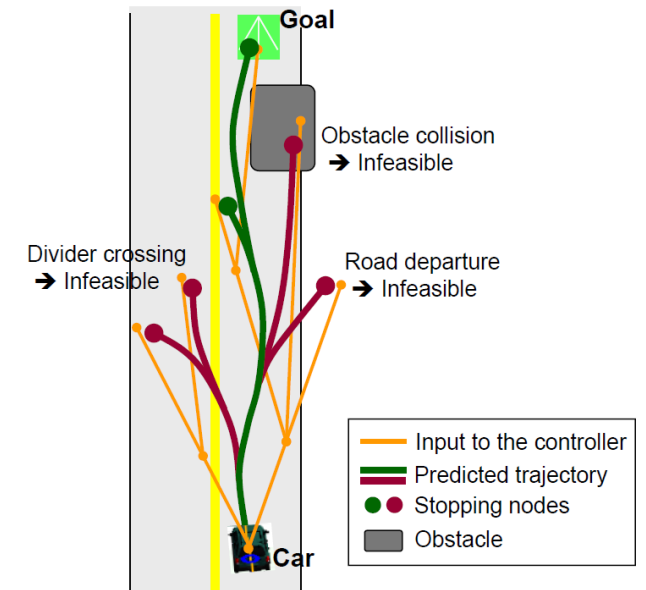
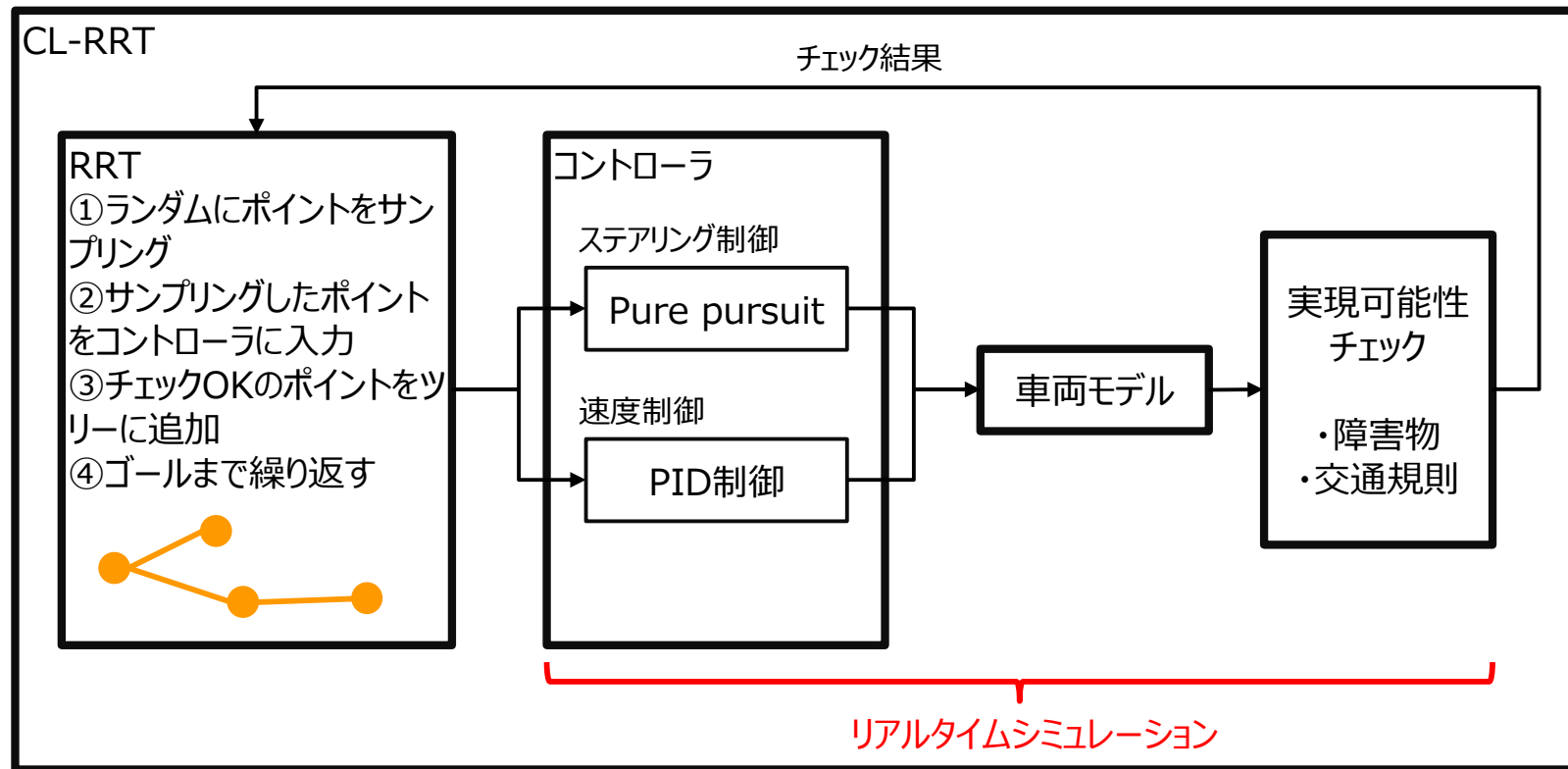
赤：加速度限界超過

https://www.ri.cmu.edu/pub_files/2012/5/ICRA12_xuwd_Final.pdf

6. 軌道生成; 事例 (2)マサチューセッツ工科大学

Rapidly-exploring Random Tree (RRT)と、車両モデルによるリアルタイムシミュレーションを組み合わせたClosed-Loop RRT(CL-RRT)による軌道生成。RRTはランダムにサンプリングしたポイントをツリー上に伸ばしていくことでゴールまでたどり着くパスを探索する方法（最適性は保証されない）。

（派生版のRRT*, Informed-RRT*は最適性が保証される）



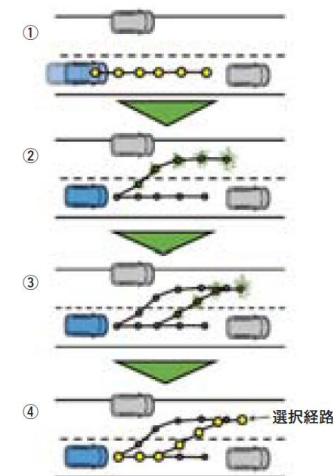
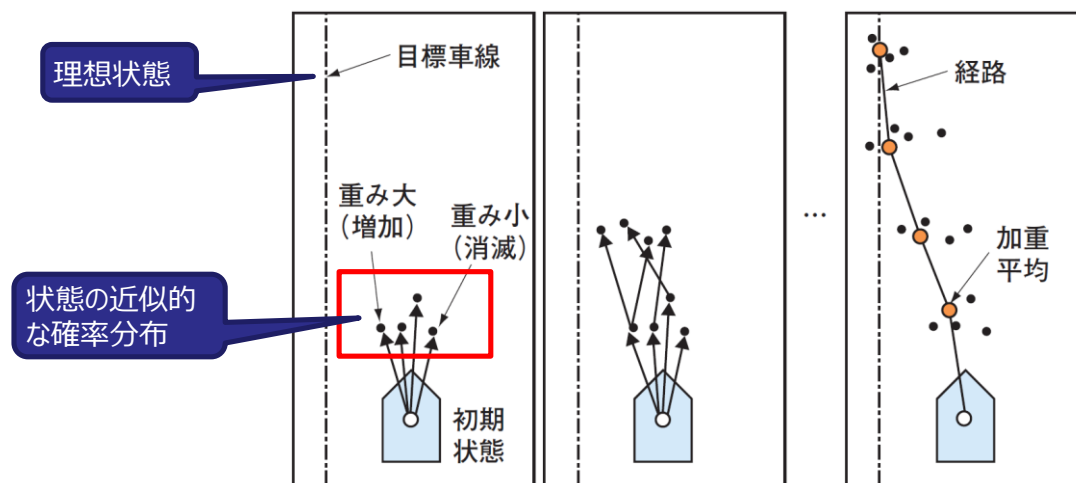
<https://dspace.mit.edu/handle/1721.1/39822>
<http://acl.mit.edu/papers/KuwataGNC08.pdf>
<http://acl.mit.edu/papers/KuwataTCST09.pdf>
<https://arxiv.org/pdf/1601.06326.pdf>

6. 軌道生成; PF-RRT

RRT系の派生として、パーティクルフィルタ(Particle Filter)を使用したPF-RRTが存在する。

パーティクルフィルタとは条件付き確率分布をパーティクルと呼ばれるデータ群によって近似的に表現する状態推定器（再帰的ベイズフィルタの一種）。

- ①理想的な車両状態（車線中央，目標車速，障害物から安全な距離を確保）を推定対象として仮定
- ②車両運動モデルに乱数を与えて複数のパーティクルを作成
- ③理想との差によってパーティクルを重み付け
- ④パーティクルの状態量の加重平均をとることで理想状態への経路を得る
- ⑤①～④をRRTのようにランダムにサンプリングしたポイントで繰り返し，到達コストが最小となる軌道を選択

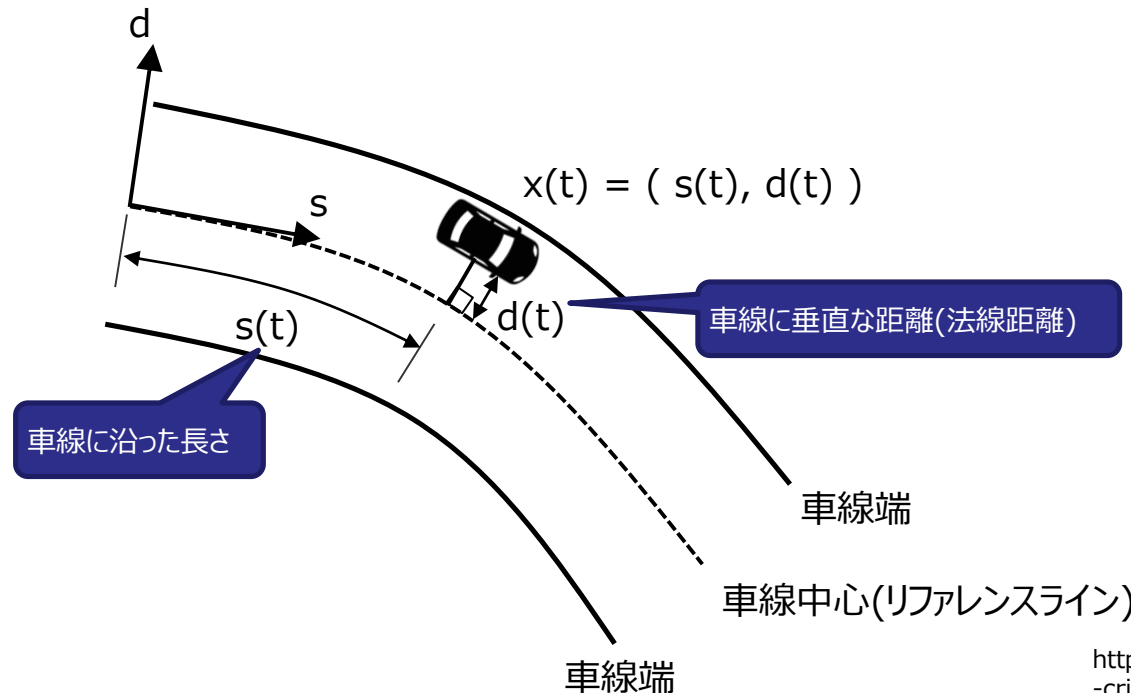


ランダムにサンプリングしたポイントで繰り返し，到達コストが最小となる軌道を選択

- ✓ 自動車技術 Vol.76, pp.12-13
- ✓ <https://www.mitsubishielectric.co.jp/corporate/randd/list/mechatronics/b236/index.html>

6. 軌道生成; Werlingらの手法

- BMW, BOSHE, カールスルーエ工科大学 共同研究
- DWAよりなめらかな軌道を生成できる
- 車線に沿った座標系を定義し, その座標系で軌道を生成する
- 軌道候補の生成はs軸方向, d軸方向別々に行う. 横軸を時間tとしたt-s平面, t-d平面上の目標点までの5次曲線を複数描き, 両者を組合わせた複数の軌道候補の各コストを算出し, コスト最小となる軌道を選択する
- 緊急回避軌道には対応していない



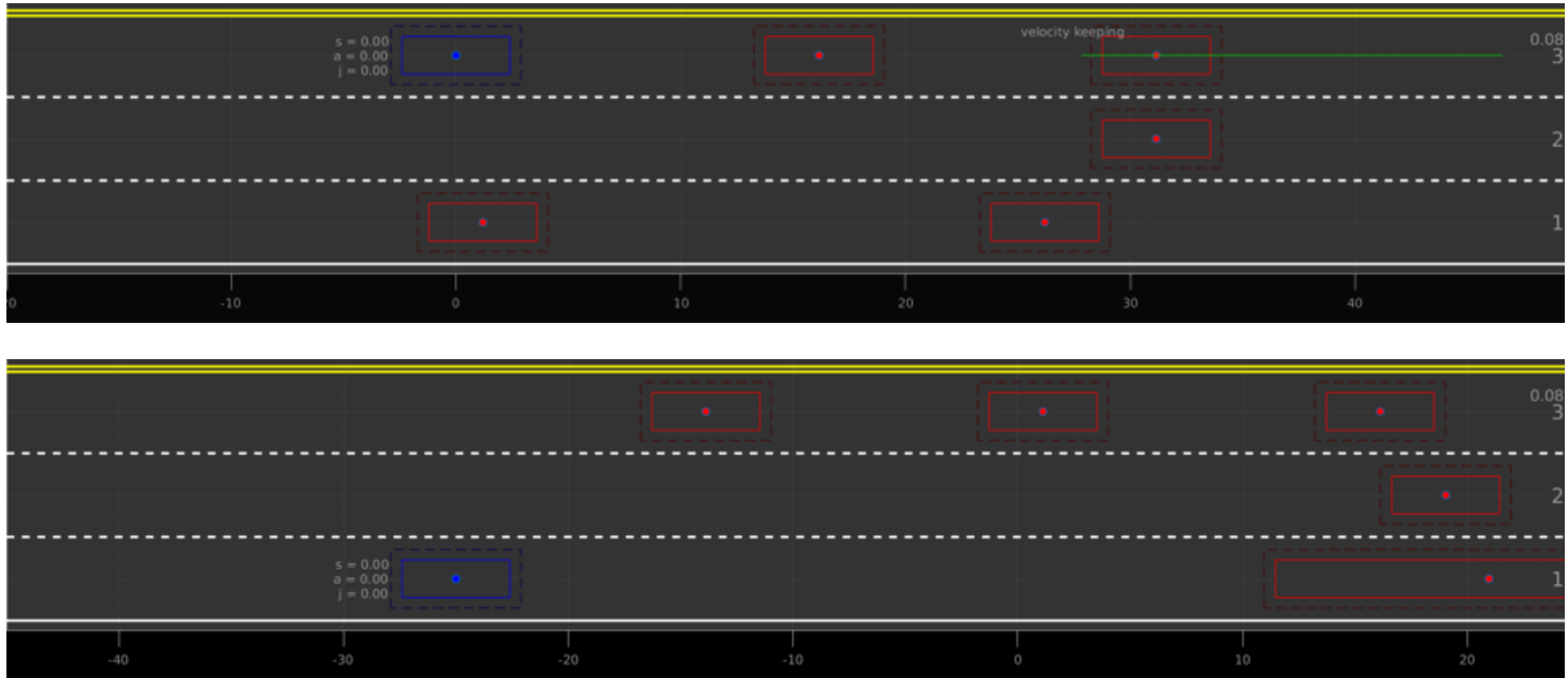
このような座標系を以下のように呼称する

- ・フレネ - セレ標構 (Frenet-Serret frame)
- ・動標構 (moving frame)
- ・SD(動)座標系, SL座標系

利点: 直線, コーナーを同様に扱う
欠点: カーブ走行時の横Gを算出できない
直交座標系への変換が必要

https://www.researchgate.net/publication/254098780_Optimal_trajectories_for_time-critical_street_scenarios_using_discretized_terminal_manifolds

6. 軌道生成; Werlingらの手法 実装例①



https://github.com/jkpld/SDC_ND_T3P1_Path_Planning

6. 軌道生成; Werlingらの手法 実装例②

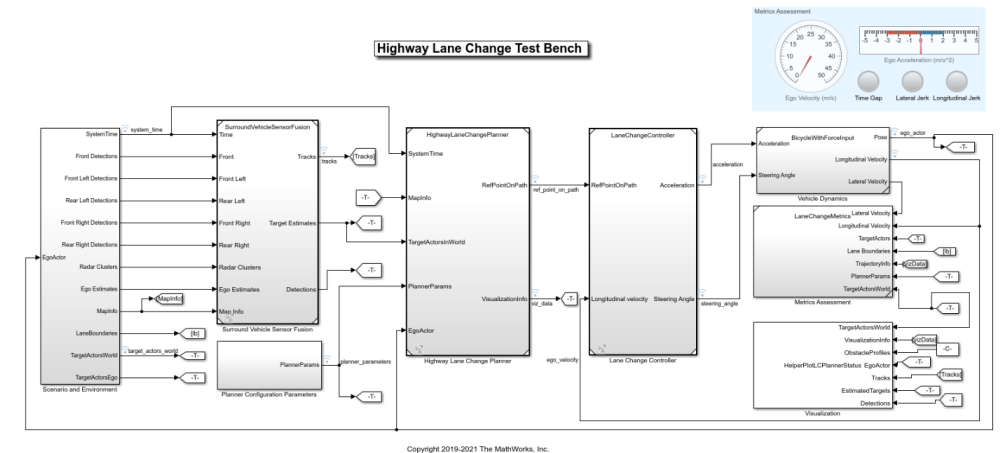
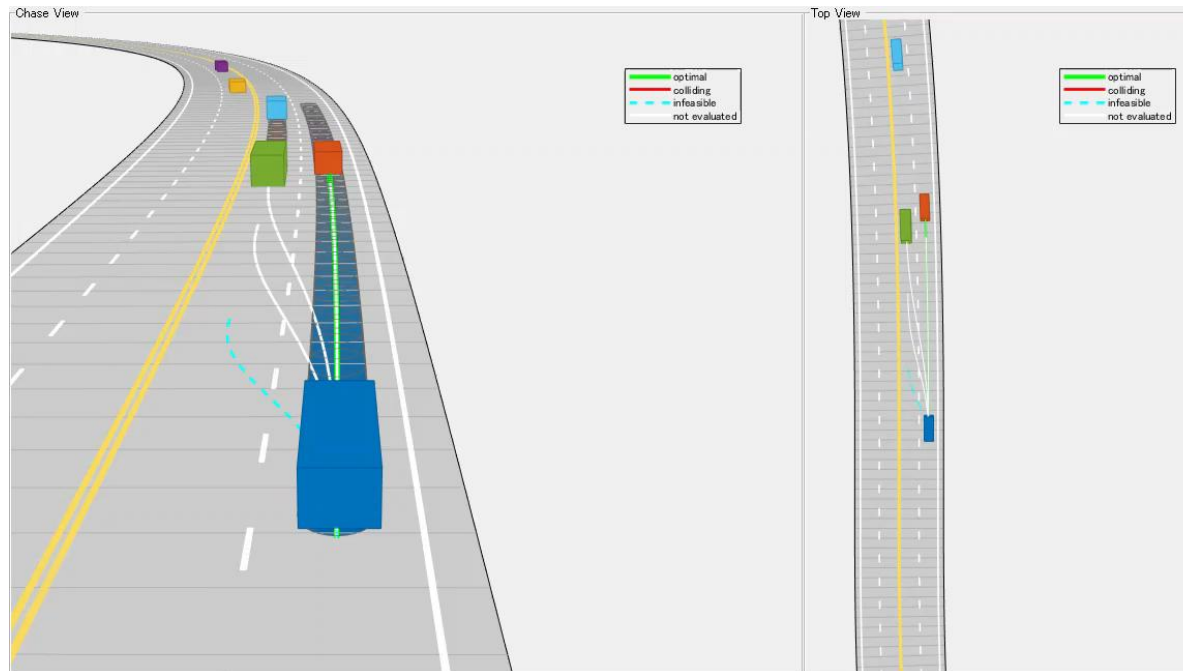
■ MathWorks社 MATLAB Navigation Toolbox 内

関数 : trajectoryOptimalFrenet

<https://jp.mathworks.com/help/nav/ref/trajectoryoptimalfrenet.html>

サンプル : Highway Lane Change

<https://jp.mathworks.com/help/driving/ug/highway-lane-change.html>



6. 軌道生成; Werlingらの手法 課題

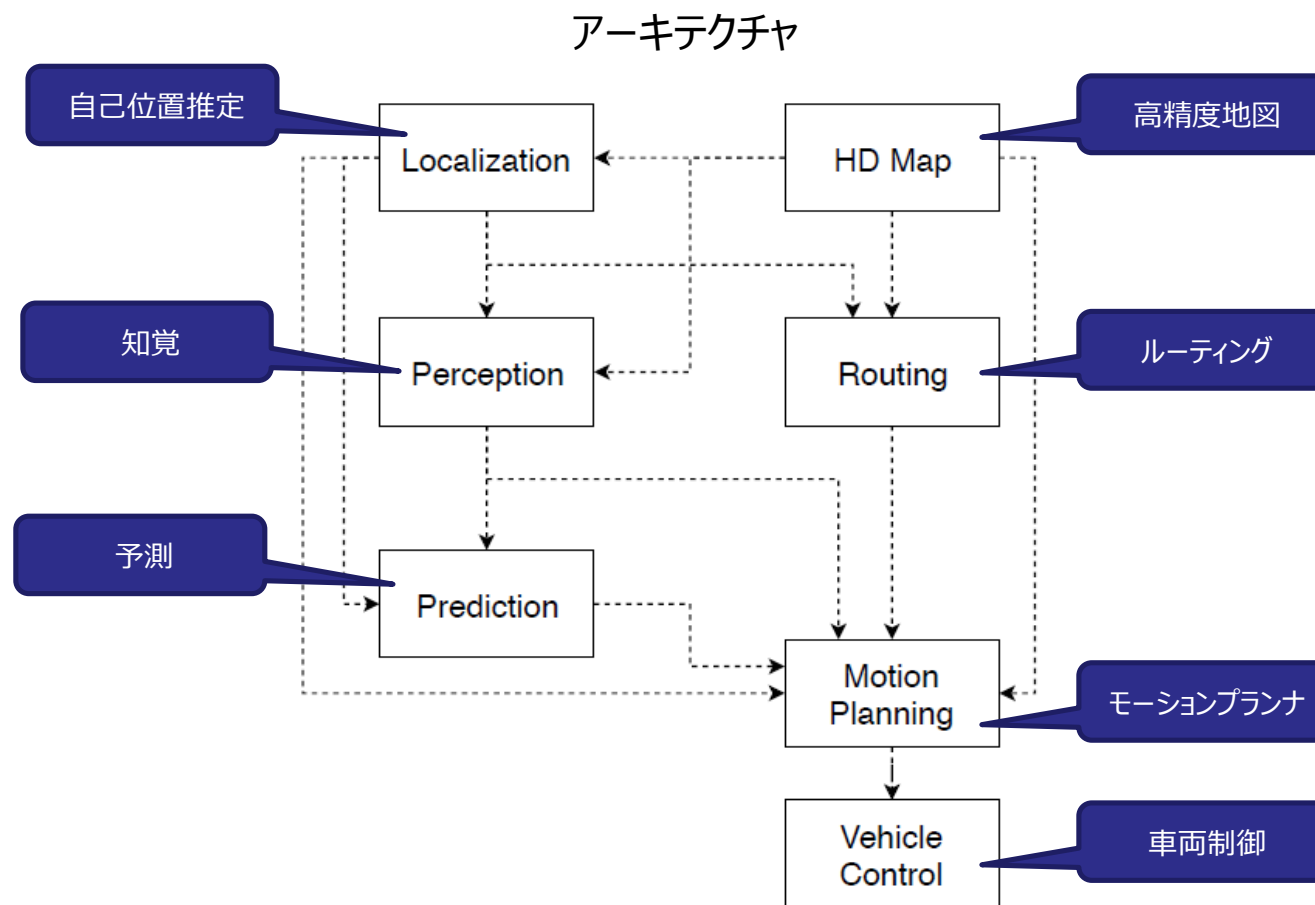
- CL-RRTやLattice plannerに比べて複数の曲線が連携された軌道を生成できない
- 計算コストが高い
- ステアリングホイールの動きがぎくしゃくする場合がある

6. 軌道生成; Apolloの事例



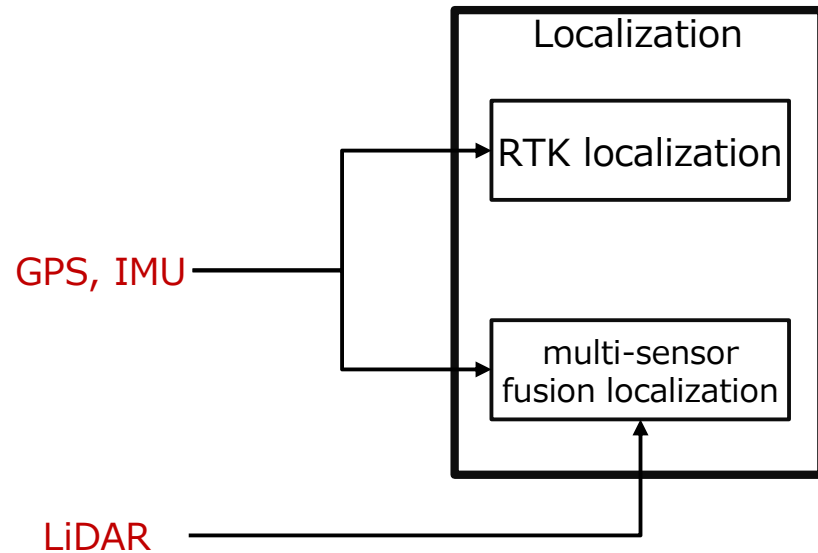
車両「萝卜快跑(Apollo Go)」

<https://www.apollo.auto/robotaxi/>

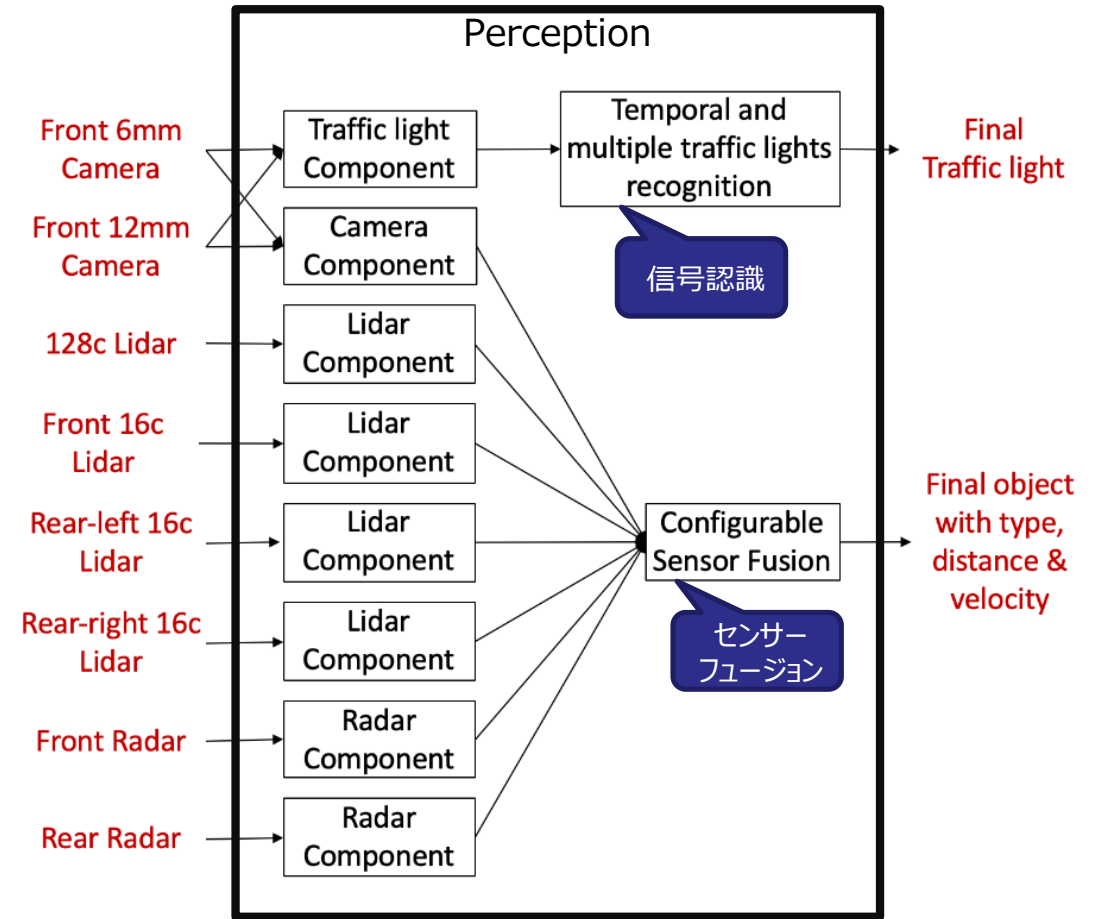


Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, Qi Kong: Baidu Apollo EM Motion Planner, arXiv:1807.08048, 2018. (<https://arxiv.org/abs/1807.08048>)

6. 軌道生成; Apolloの事例 センサ構成



Localizationモジュール

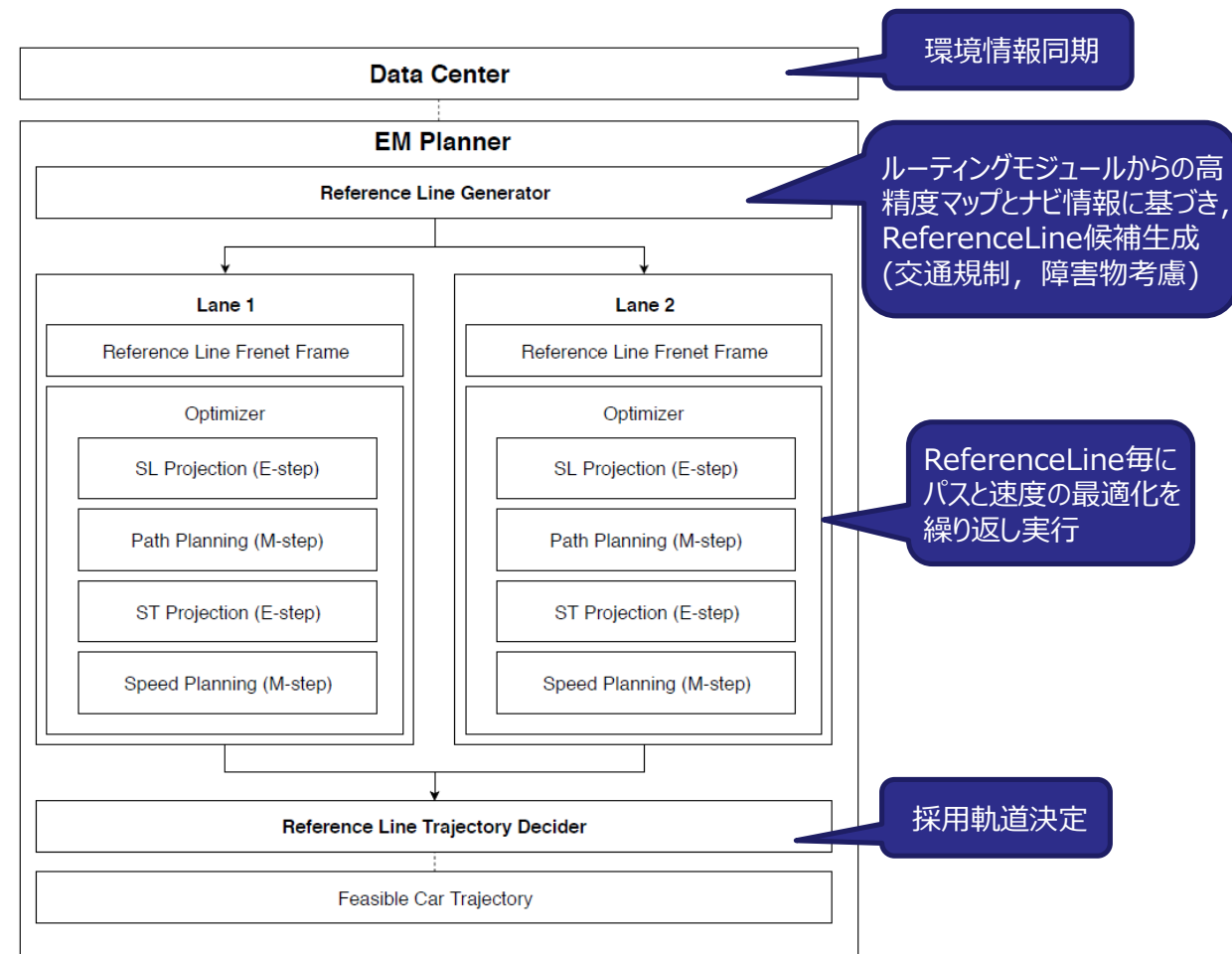


Perceptionモジュール

<https://www.apollo.auto>

6. 軌道生成; Apolloの事例 EMプランナー

- ✓ マルチレーンおよびシングルレーンの自動運転をカバー
- ✓ 並行して計算された車線レベルの軌道を比較することによって軌道を決定する
- ✓ フレネ - セレ標構に基づいてパスと速度の最適化を繰り返す
- ✓ 経路と速度の最適化のために、動的計画法とスプラインベースの二次計画法を組み合わせ使用
- ✓ 交通ルール、障害物の決定、および滑らかさを同時に処理するためのスケラブルで調整が容易なフレームワーク
- ✓ 高速道路と低速の市街地走行シナリオの両方に拡張可能
- ✓ 8秒間または200m先までの軌道を生成
- ✓ 緊急事態対応は対象外

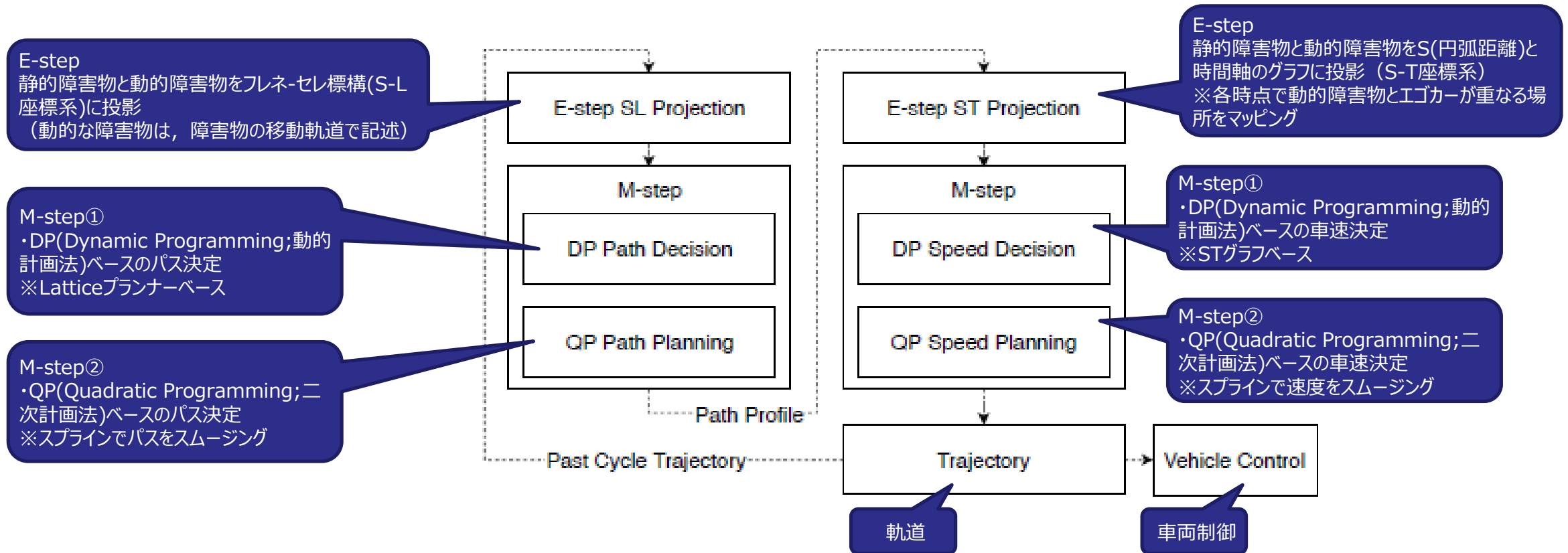


Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, Qi Kong: Baidu Apollo EM Motion Planner, arXiv:1807.08048, 2018. (<https://arxiv.org/abs/1807.08048>)

6. 軌道生成; Apolloの事例【参考】EMアルゴリズム

- ✓ EMアルゴリズムは観測データの確率モデル（潜在変数を含む確率分布形状とパラメータ）を推定（最尤推定）するための手法
- ✓ EステップとMステップを収束するまで繰り返し実行し推定を行う
 - E (Expectation; 期待値)ステップ：現在推定されている潜在変数の分布に基づいて、モデルの尤度の期待値を計算
 - M(Maximization; 最大化) ステップ：Eステップで求めた尤度の期待値を最大化するようなパラメータを求める
- ✓ 要はデータをクラスタリングするための一手法
- ✓ 機械学習, 音声認識, 因子分析などに用いられる

6. 軌道生成; Apolloの事例 パスと速度の最適化(EMステップ)

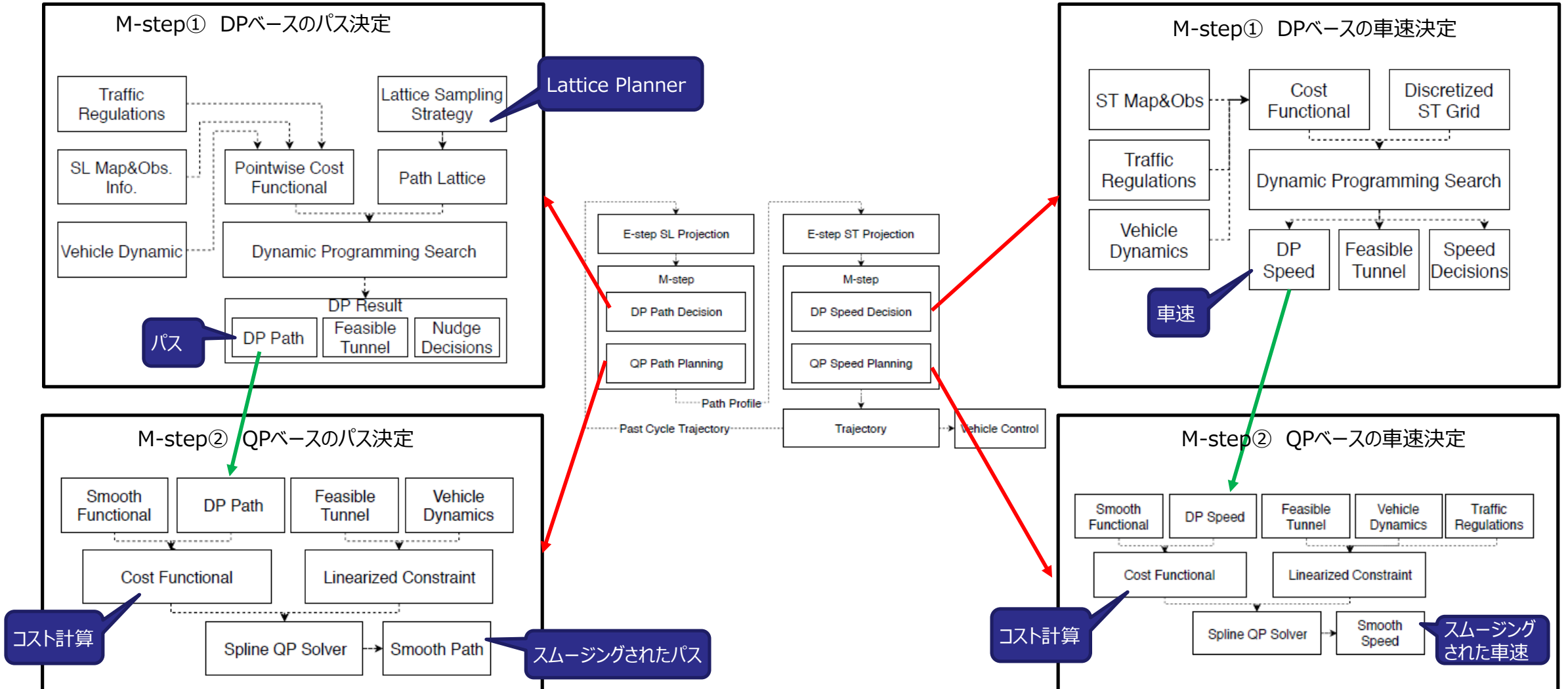


EMアルゴリズムの反復計算方法(E-step, M-step)を参考にしていて、最尤推定は行っていない可能性がある

動的計画法：対象となる問題を複数の部分問題に分割し、部分問題の計算結果を記録しながら解いていく手法を総称してこう呼ぶ。

Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, Qi Kong: Baidu Apollo EM Motion Planner, arXiv:1807.08048, 2018. (<https://arxiv.org/abs/1807.08048>)

6. 軌道生成; Apolloの事例 Mステップ詳細



6. 軌道生成; KAIT研究事例

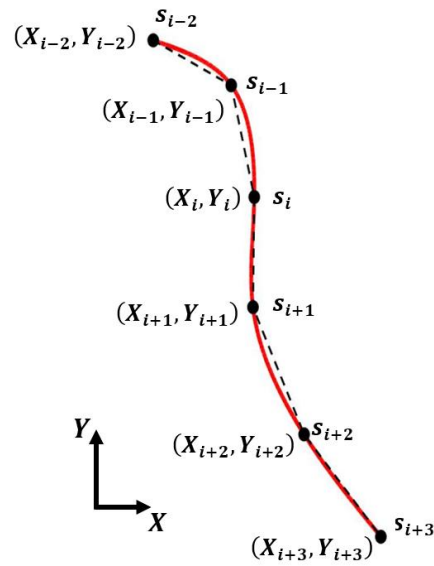
「曲線座標系を用いた自動運転制御アルゴリズムに関する研究」

スキッティパッタナクン・フリット (神奈川工科大学 自動運転・運転支援研究室) 2021年3月修士論文

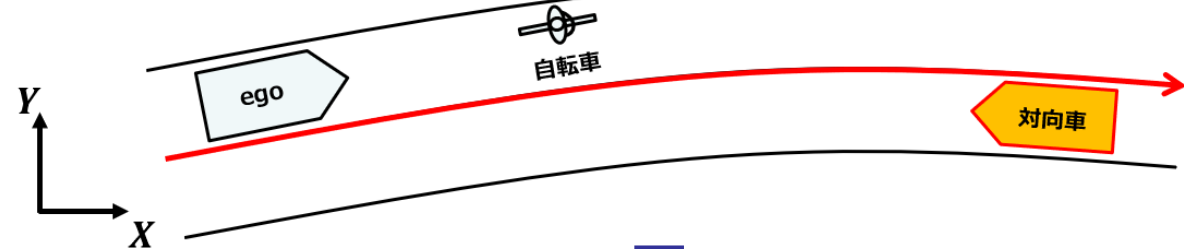
(https://cp.kanagawa-it.ac.jp/wp-content/uploads/laboratory_c11_2020_06.pdf)

- ✓ コーナーと対向車を考慮した並走自転車追越しシナリオ
- ✓ Waypoint間をスプライン補間
- ✓ 直線とコーナーを同様に扱うため, **SD座標系(フレネ-セレ標構)**を使用

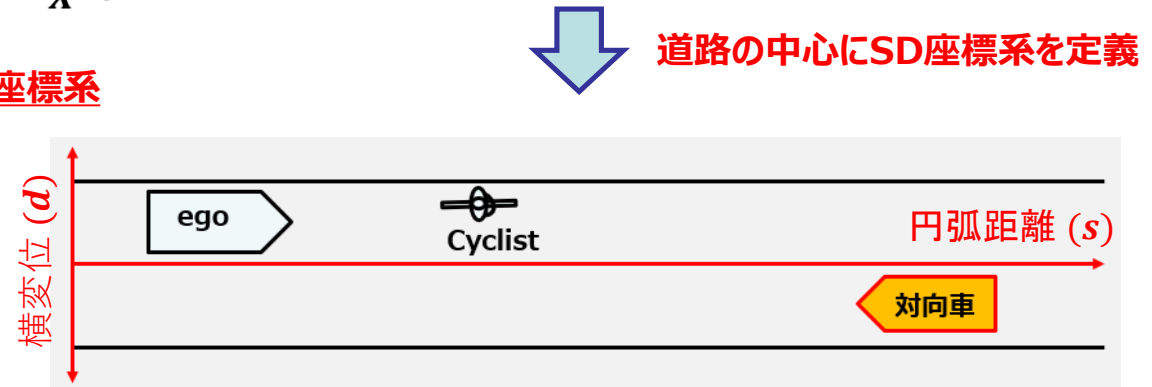
道路の中心のWaypointをスプライン補間



直交座標系



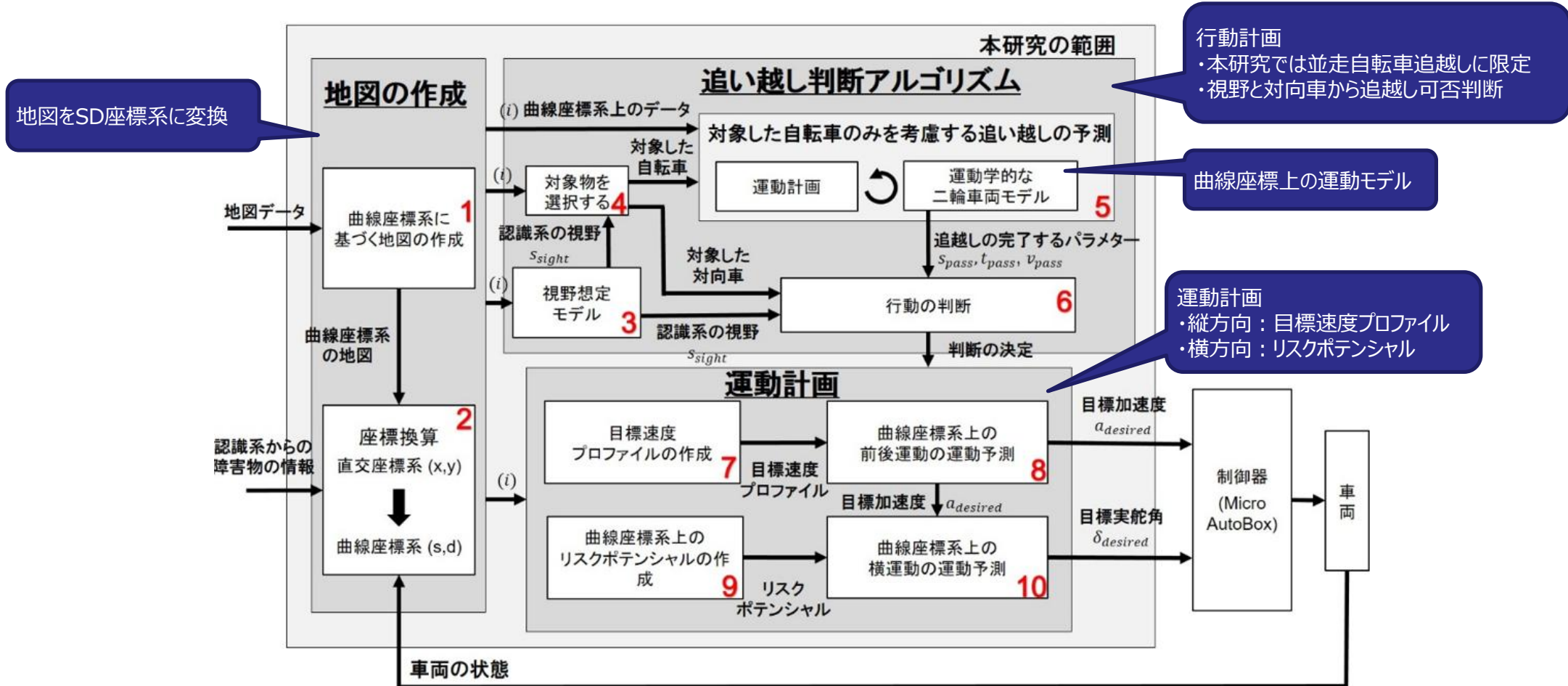
SD座標系



https://cp.kanagawa-it.ac.jp/wp-content/uploads/laboratory_c11_2020_06.pdf
<https://revistas.udistrital.edu.co/index.php/reviving/article/view/9574/11565>

6. 軌道生成; KAIT研究事例

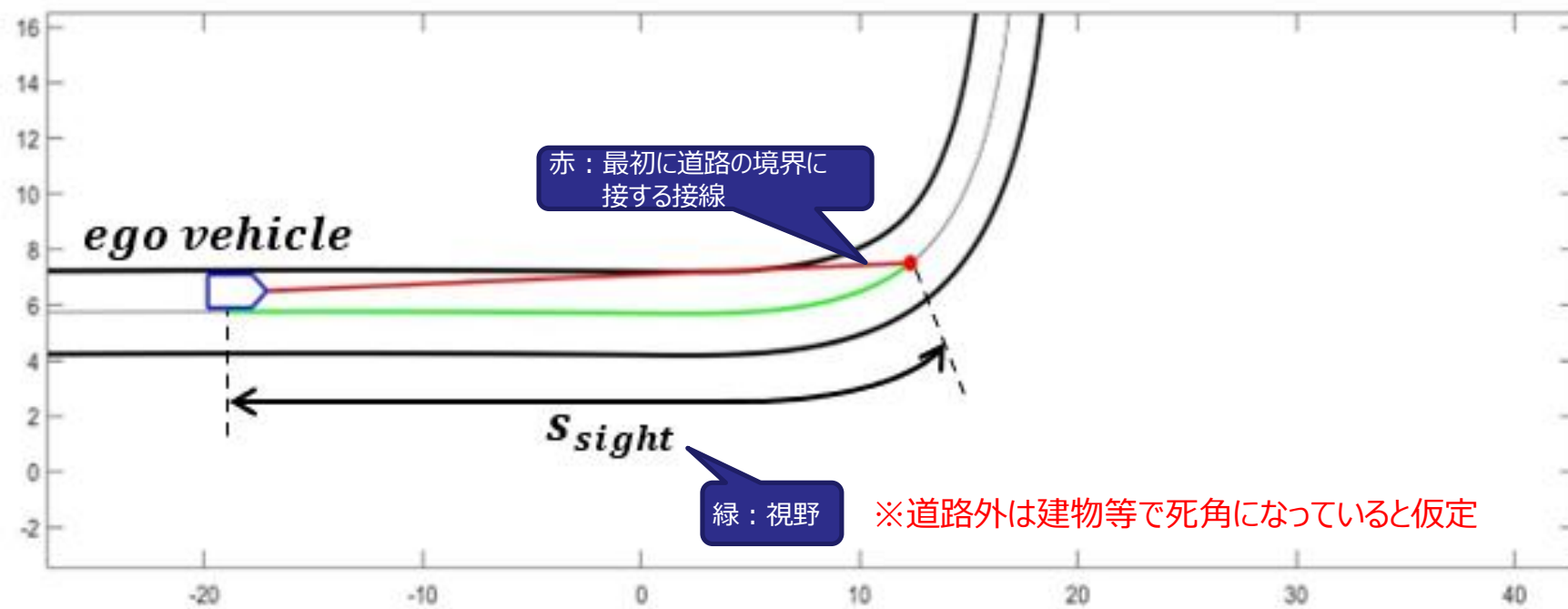
ソフトウェアアーキテクチャ



https://cp.kanagawa-it.ac.jp/wp-content/uploads/laboratory_c11_2020_06.pdf

6. 軌道生成; KAIT研究事例

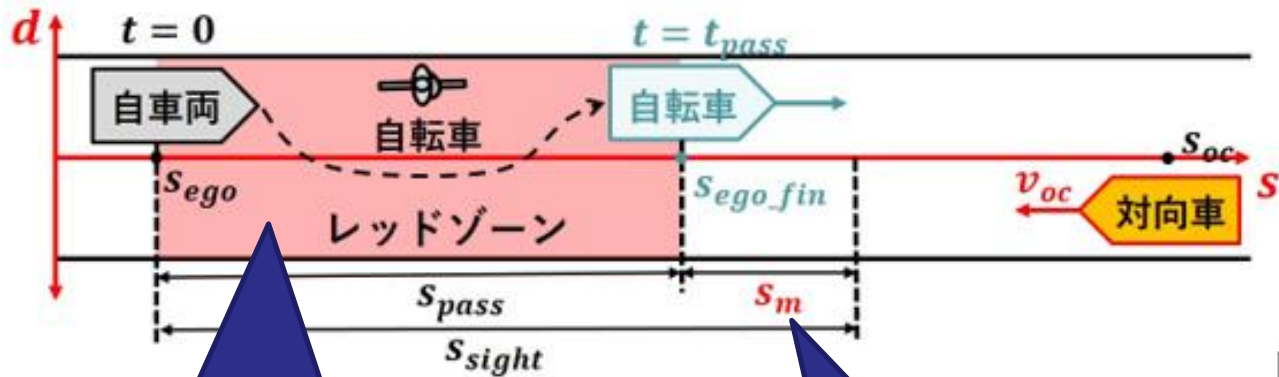
行動計画：①視野推定モデル



https://cp.kanagawa-it.ac.jp/wp-content/uploads/laboratory_c11_2020_06.pdf

6. 軌道生成; KAIT研究事例

行動計画：②視野の範囲の対象物に対する行動を決定

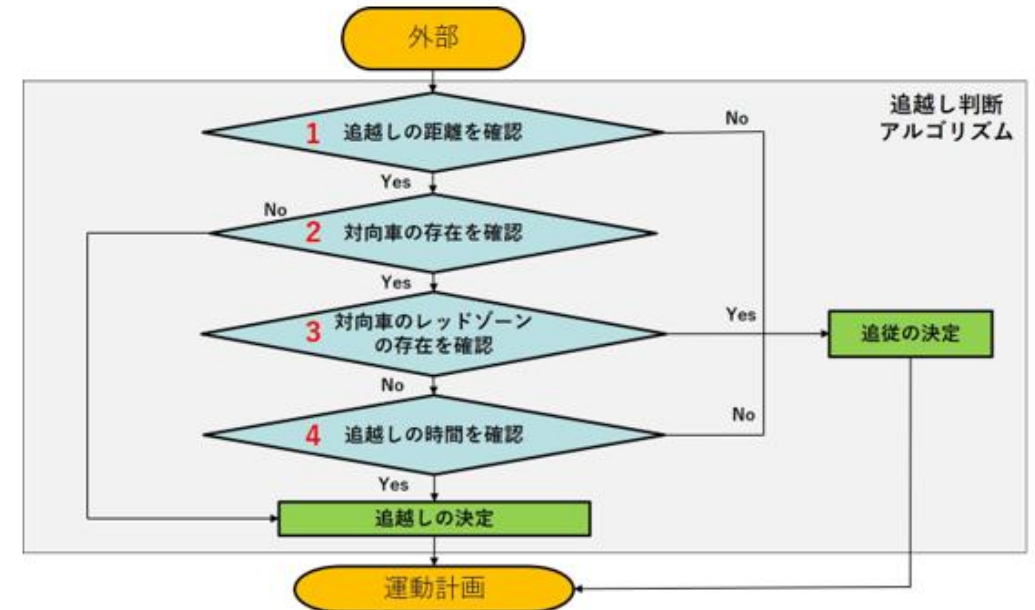


追越しの期間に
対向車線の占有が必要な領域
ここに対向車が入らなければ追越し可能

視野

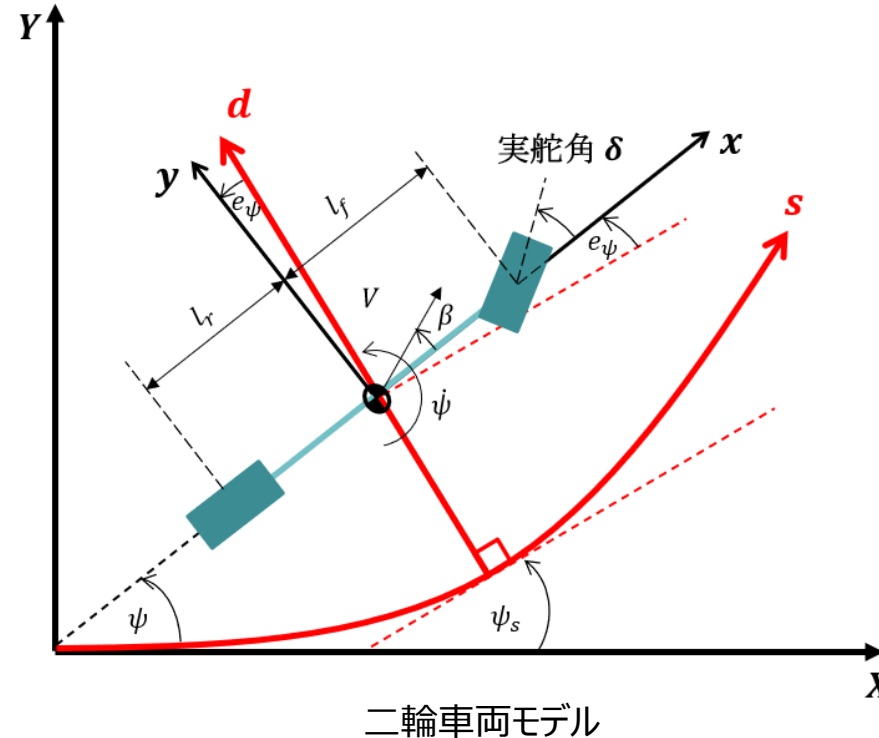
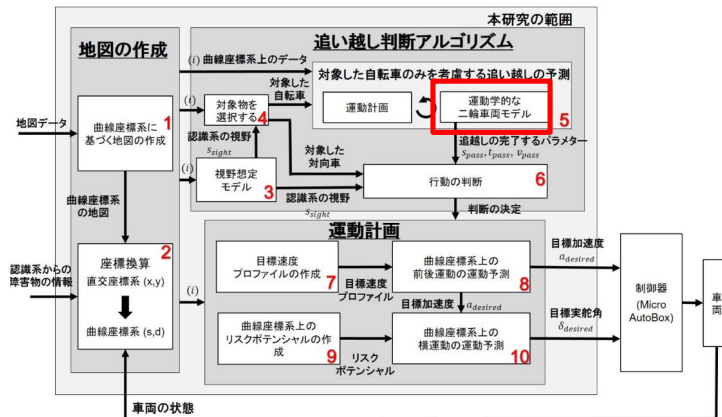
安全余裕

行動決定アルゴリズム



6. 軌道生成; KAIT研究事例

曲線座標系上の運動モデル：二輪車両モデル



速度

$$s\text{-軸} \quad \dot{s} = \frac{V}{1-\kappa d} \cdot \cos(e_\psi + \beta) \quad (1)$$

$$d\text{-軸} \quad \dot{d} = V \cdot \sin(e_\psi + \beta) \quad (2)$$

ヨーレート

$$\dot{\psi} = \frac{V}{l_r} \sin(\beta) \quad (3)$$

進行方向の差(s-軸から)

$$e_\psi = \psi - \psi_s \quad (4)$$

加速度

$$\dot{V} = a \quad (5)$$

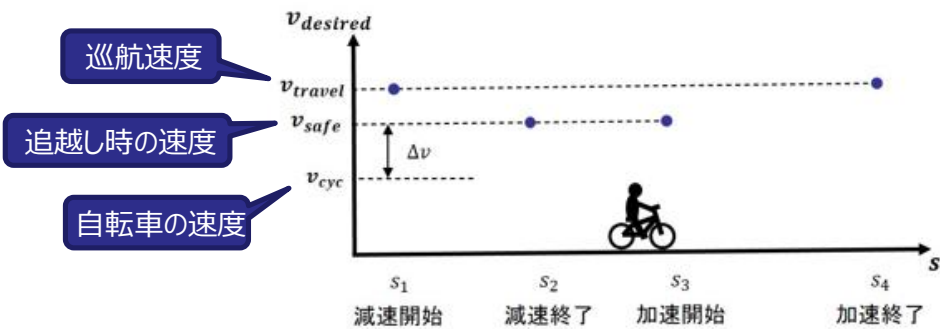
横滑り角

$$\beta = \arctan \left[\frac{l_r}{l_f + l_r} \tan(\delta) \right] \quad (6)$$

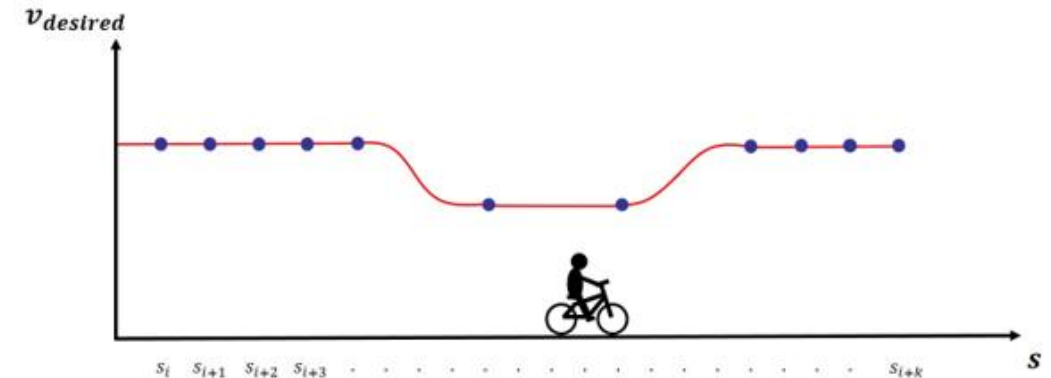
6. 軌道生成; KAIT研究事例

運動計画：①並走自転車を追い越しまたは追従するための目標速度プロファイルを作成

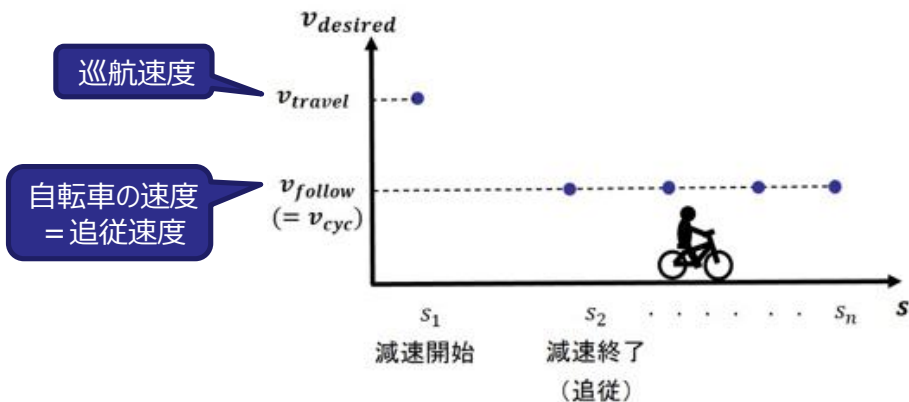
自転車を追い越すケース



自転車を追い越すケースの速度プロファイルの例
(station間は多項式で補間する)



自転車に追従するケース



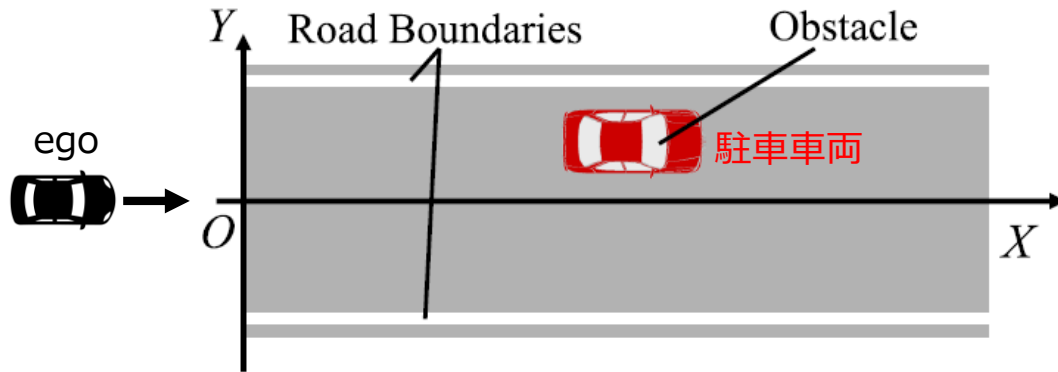
※station: 目標速度を算出するポイント(着目する対象の周囲を中心に設定する)

https://cp.kanagawa-it.ac.jp/wp-content/uploads/laboratory_c11_2020_06.pdf

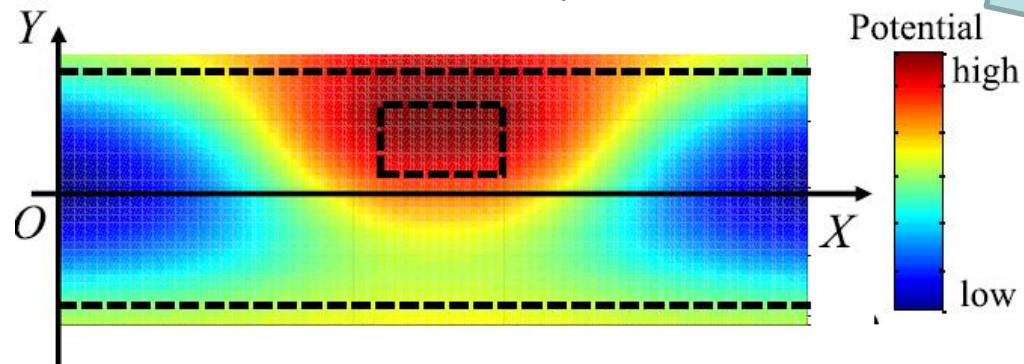
6. 軌道生成; KAIT研究事例

リスクポテンシャル

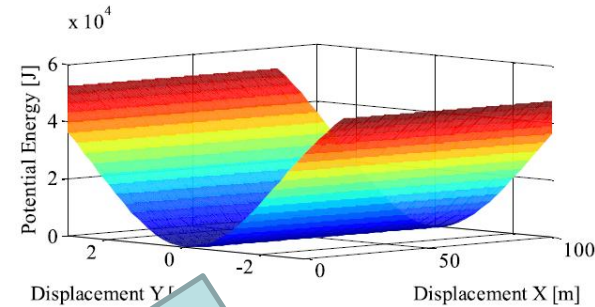
駐車車両回避シナリオの例



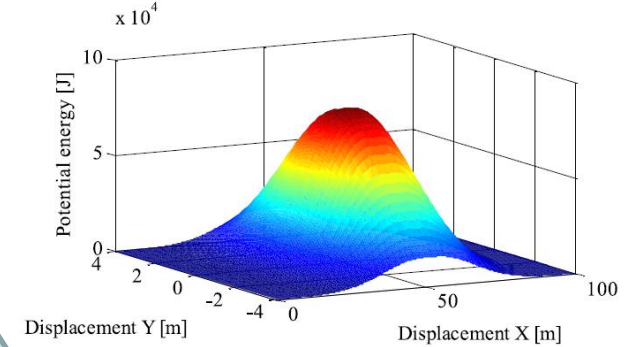
リスクポテンシャル



道路のリスクポテンシャル

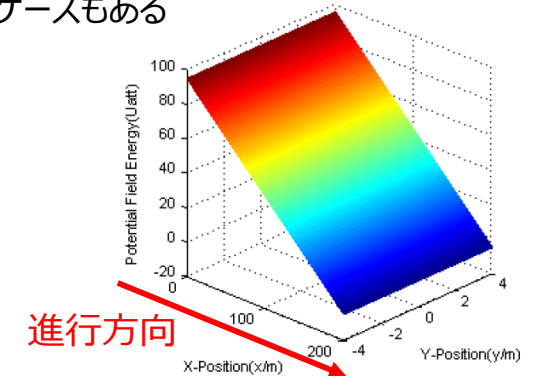


駐車車両のリスクポテンシャル



合成

進行方向を動機付けするために、Waypoint方向にポテンシャルを設定するケースもある

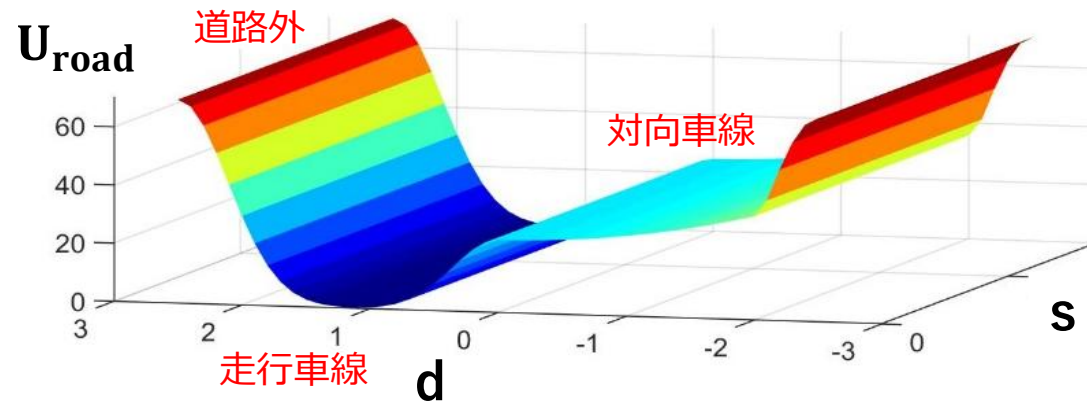
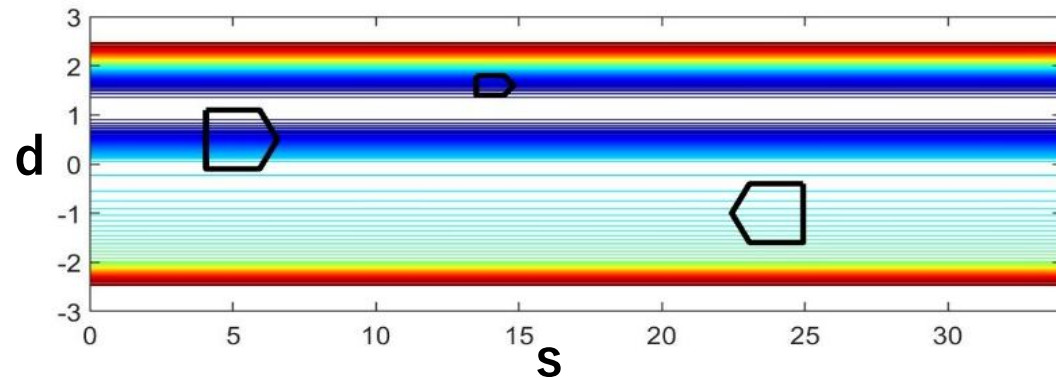


- ✓ https://www.jstage.jst.go.jp/article/ipsjjip/25/0/25_32/_article
- ✓ https://www.jstage.jst.go.jp/article/sicejl/54/11/54_820/_article/-char/ja/
- ✓ https://cp.kanagawa-it.ac.jp/wp-content/uploads/laboratory_c11_2020_06.pdf

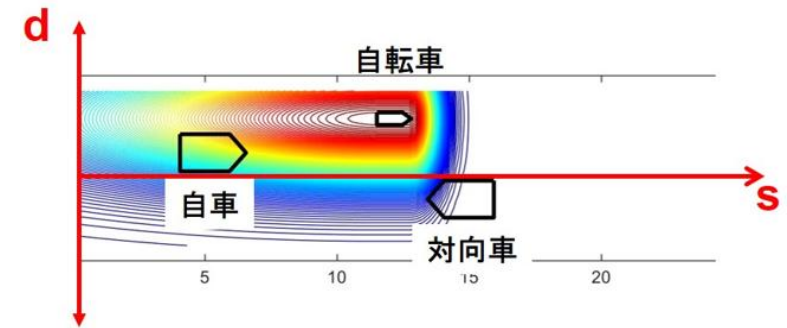
6. 軌道生成; KAIT研究事例

運動計画：②SD座標系上にリスクポテンシャルフィールドを作成

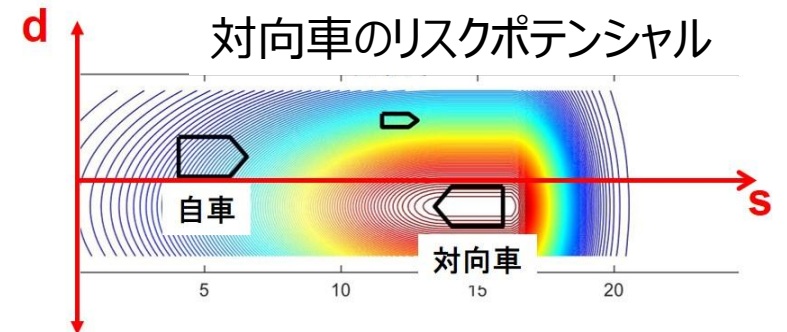
道路のリスクポテンシャル



自転車のリスクポテンシャル



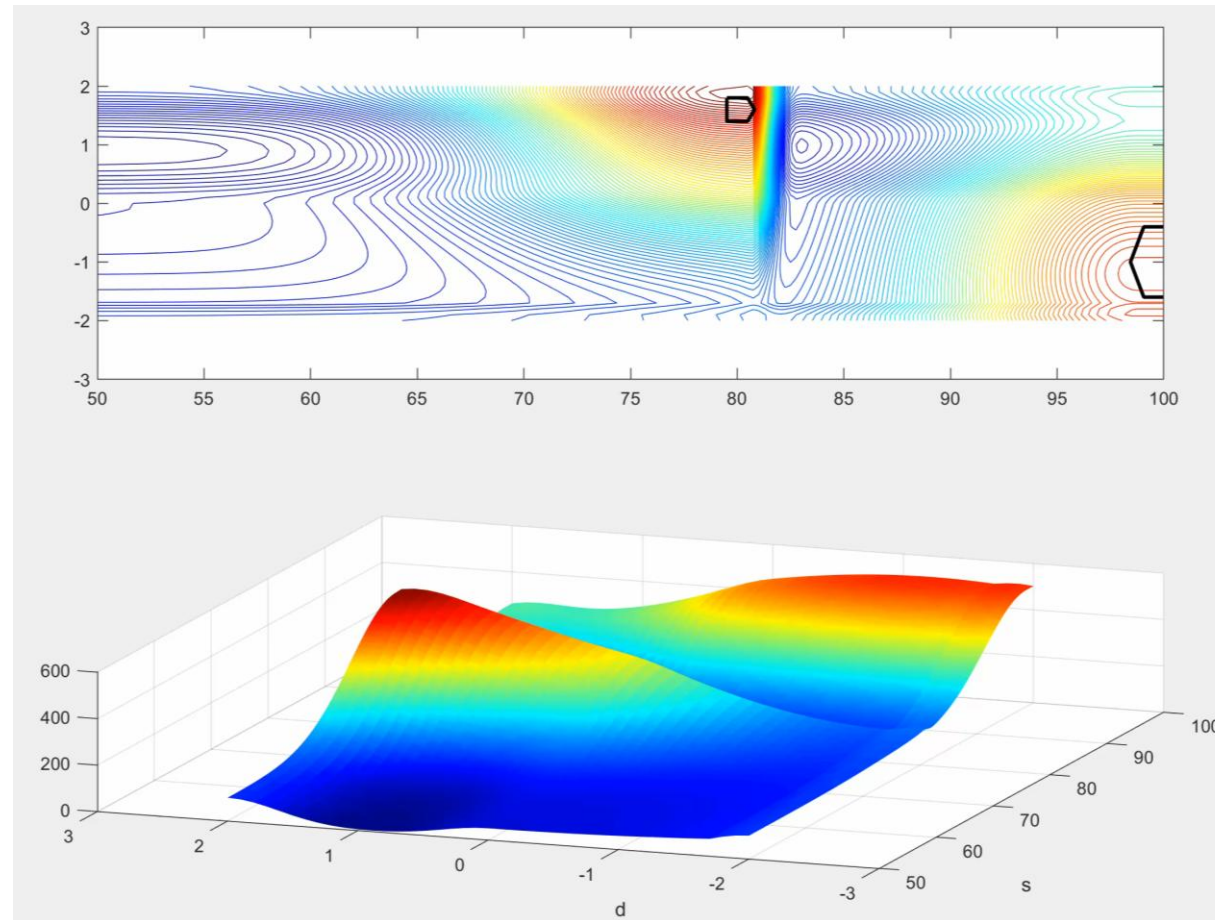
対向車のリスクポテンシャル



自車とリスク対象の相対速度に応じてフィールドを変形

6. 軌道生成; KAIT研究事例

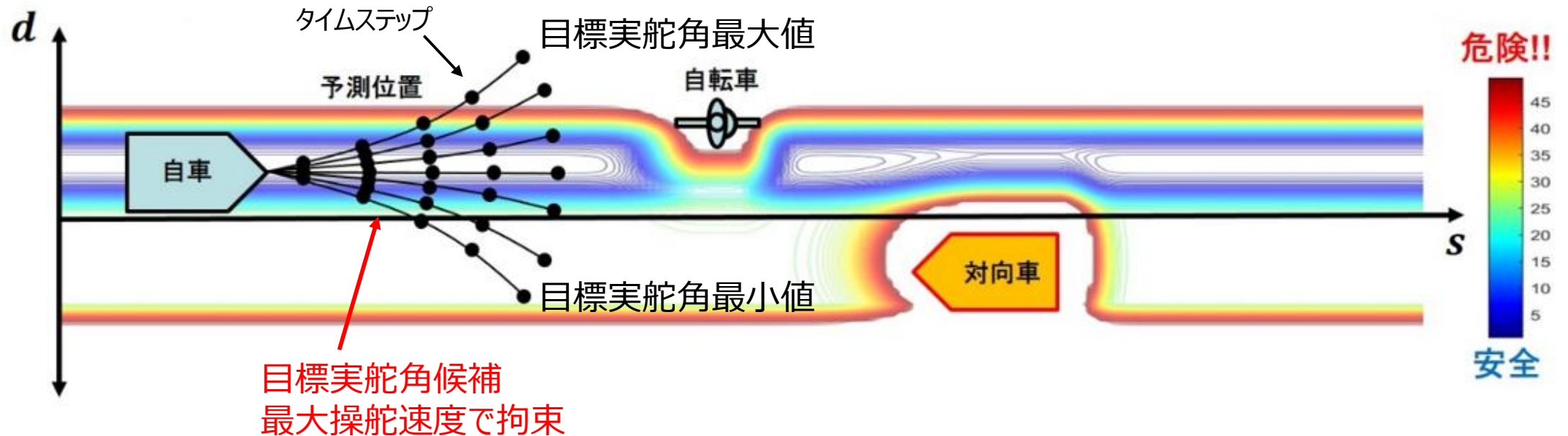
運動計画：②SD座標系上にリスクポテンシャルフィールドを作成；合成結果



https://cp.kanagawa-it.ac.jp/wp-content/uploads/laboratory_c11_2020_06.pdf

6. 軌道生成; KAIT研究事例

運動計画：③目標実舵角最小値～最大値の間で目標実舵角候補を作成（最大操舵速度で拘束）



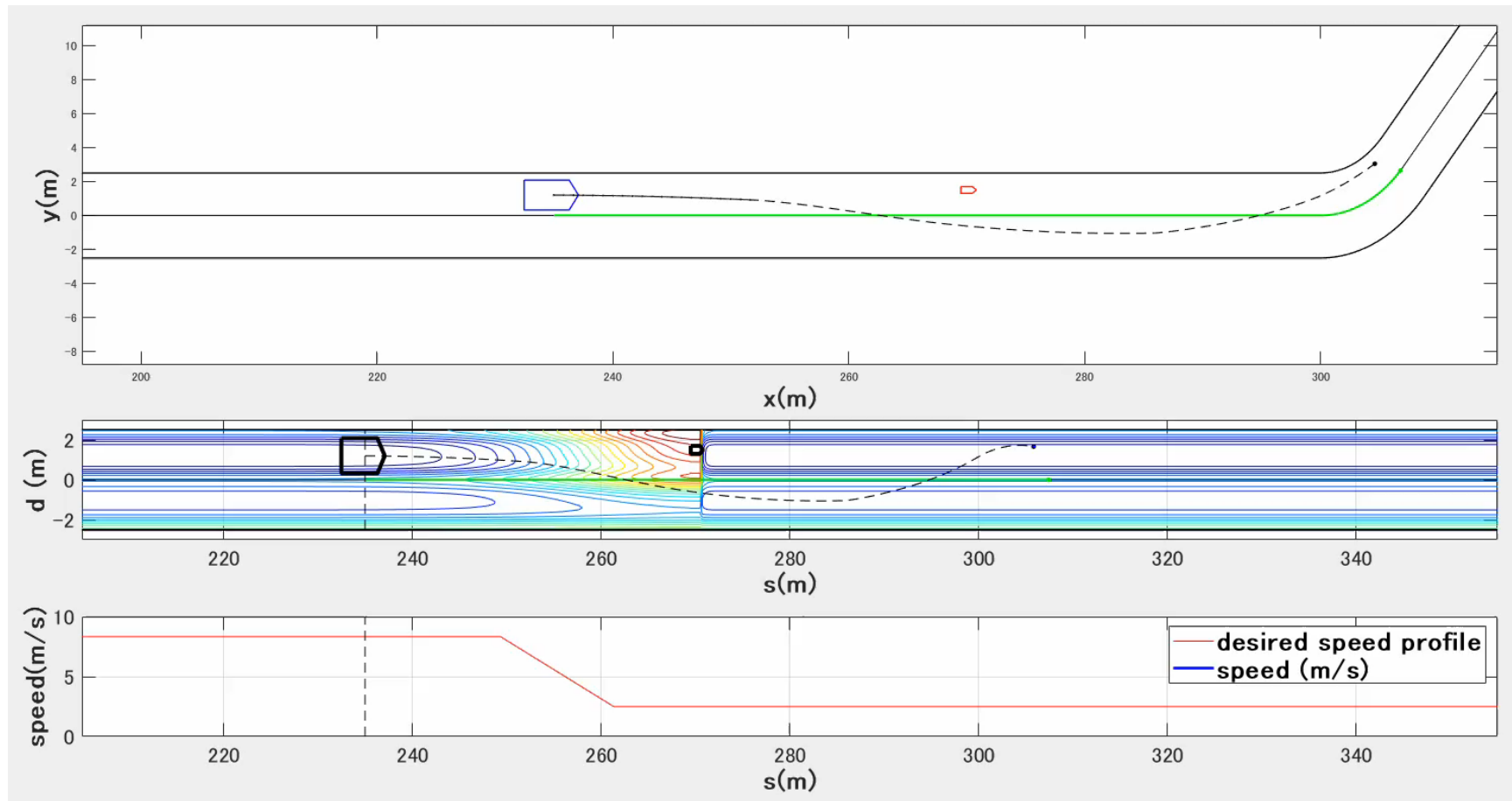
※さらに詳細な運動モデルにも置換え可能

運動計画：④実舵角候補の各タイムステップにおけるリスクポテンシャル値の合計が最小となる実舵角候補を採用

https://cp.kanagawa-it.ac.jp/wp-content/uploads/laboratory_c11_2020_06.pdf

6. 軌道生成; KAIT研究事例

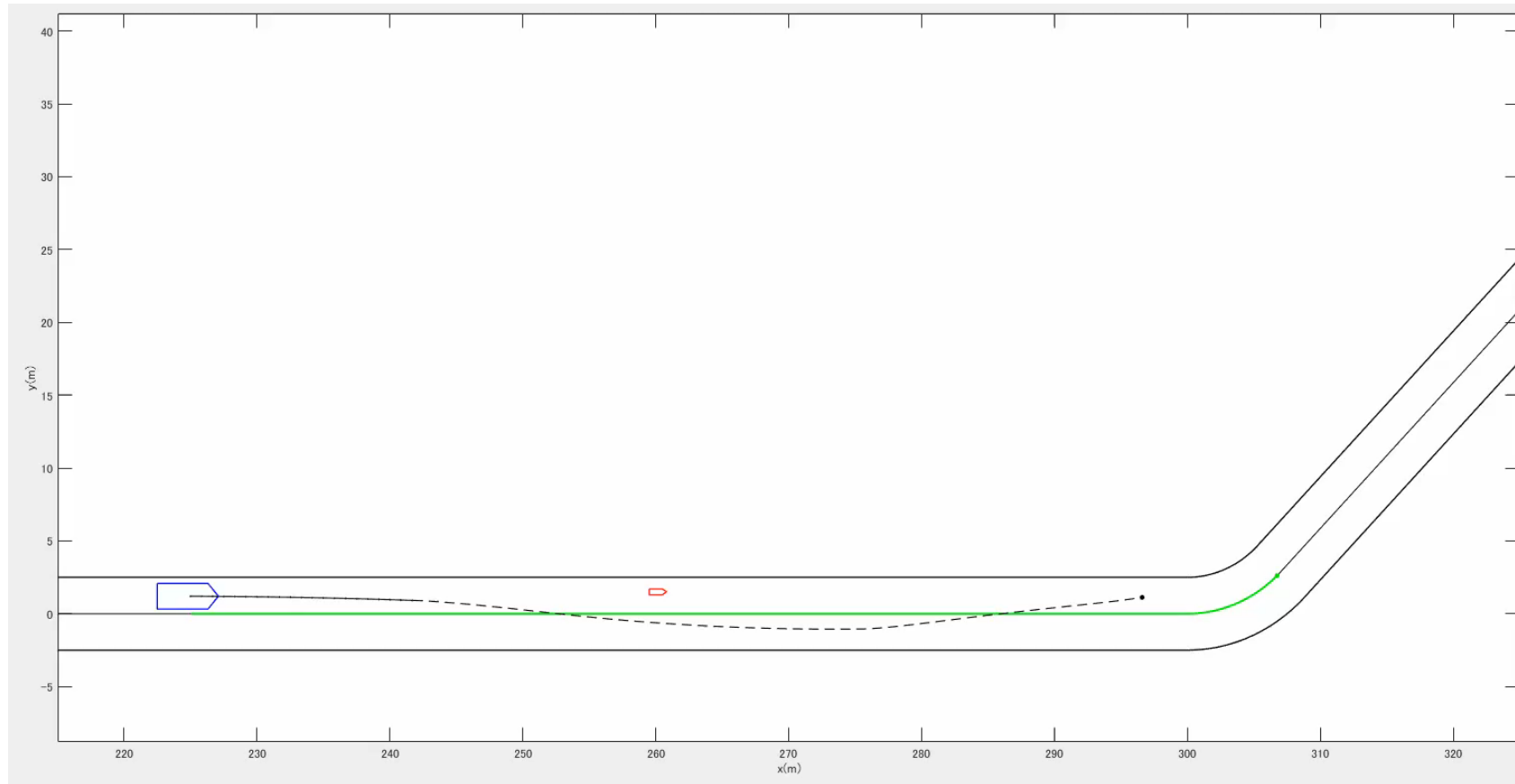
シミュレーション結果：追従してから追越しするケース



https://cp.kanagawa-it.ac.jp/wp-content/uploads/laboratory_c11_2020_06.pdf

6. 軌道生成; KAIT研究事例

シミュレーション結果：対向車による追越し中止ケース



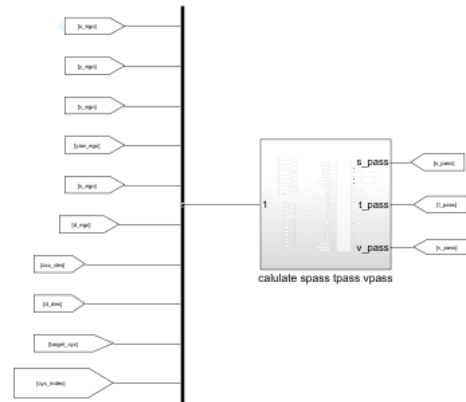
https://cp.kanagawa-it.ac.jp/wp-content/uploads/laboratory_c11_2020_06.pdf

6. 軌道生成; KAIT研究事例

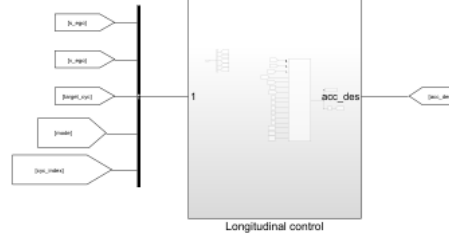
シミュレーション用Simulinkモデル

課題：計算速度

SD座標系変換



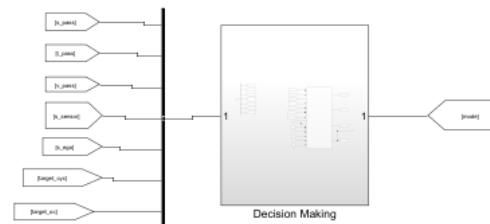
縦方向制御



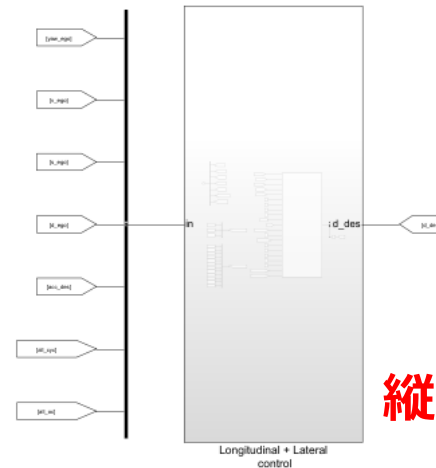
車両モデル



行動決定



縦 + 横方向制御



https://cp.kanagawa-it.ac.jp/wp-content/uploads/laboratory_c11_2020_06.pdf

7. 学習の手法

2007 DARPA Urban Challengeより後では、ディープラーニングが注目されたこともあり、学習を用いた自動運転の研究が進んでいる。

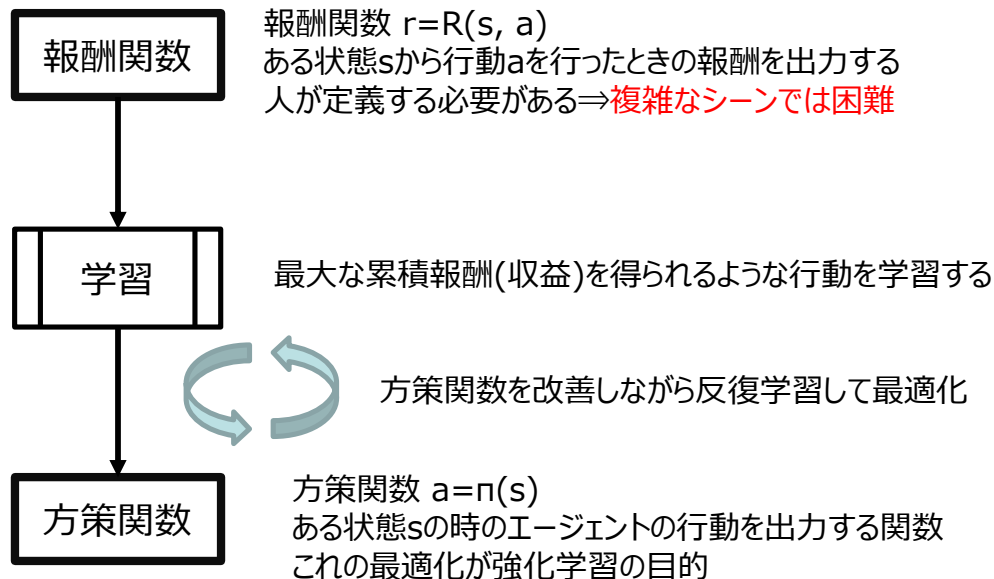
本章では、以下の学習手法について整理する。

- 強化学習
- 逆強化学習
- 模倣学習
- 能動学習

7. 学習の手法; (1)強化学習(RL)

- ✓ 強化学習(RL; Reinforcement Learning)は, 現在の状態でエージェント(意思決定主体)が取るべき行動を決定する問題を扱う機械学習の一種
- ✓ エージェントは行動を選択することで環境から報酬を得る
- ✓ 強化学習は一連の行動を通じて報酬が最も多く得られるような方策(policy)を学習する
- ✓ つまり, 強化学習とは方策の最適化問題である

強化学習(RL)



【学習の方法】

■ 価値ベース

ある状態 s における行動 a の価値を出力する行動価値関数 $Q(s, a)$ を学習し, それが最大となる行動をとるように方策関数を調整してゆく

■ 方策ベース

方策関数を直接最適化

※行動価値関数 $Q(s, a)$

現在の方策関数で, 状態 s から a の行動をとったあとに, ゴールまでたどり着いた時に累積報酬和の期待値を出力する
時間がかかるほど報酬が減る時間割引率 (γ) という概念も含まれている

- ✓ <https://ja.wikipedia.org/wiki/%E5%BC%B7%E5%8C%96%E5%AD%A6%E7%BF%92>
- ✓ https://twitter.com/ogawa_yutaro_22/status/1399831931311644672
- ✓ <https://qiita.com/triwave33/items/5e13e03d4d76b71bc802>
- ✓ <https://qiita.com/shionhonda/items/ec05aade07b5bea78081>
- ✓ <https://qiita.com/fridericusgauss/items/aa5215c29646963bda29>

7. 学習の手法; (1)強化学習 手法一覧

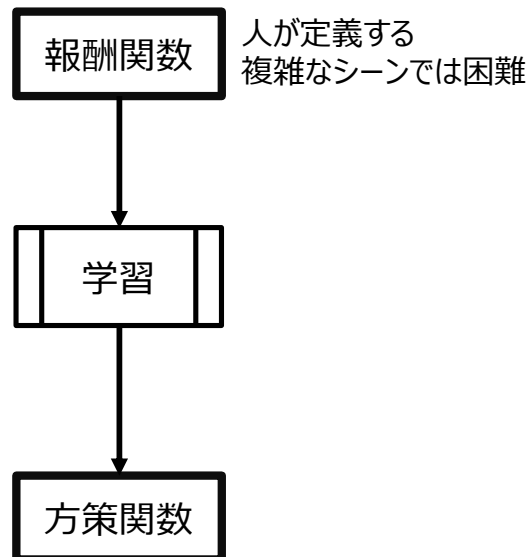
手法			発表年	説明	論文等
動的計画法 (DP; Dynamic Programming) ※複雑なケースには適用困難				対象となる問題を複数の部分問題に分割し、部分問題の計算結果を記録しながら解いていく手法 環境を完全にモデル化できる場合に使える手法	
モンテカルロ法 ※複雑なケースには適用困難				報酬のみを頼りに、行動価値関数を推定する方法 収束までに時間がかかる	
TD法 (Temporal Difference Learning; 時間的差分学習) ※実用的	価値ベース ※決定論的, 離散的	Q学習	1992	学習時に方策を考慮しない, 最大報酬が得られることのみに着目して行動 (「方策OFF」: リスクがあっても最短距離を選択する傾向)	
		SARSA	1994	学習時に方策を考慮する (「方策ON」: リスクを回避して迂回するような傾向あり)	
		DQN	2013	Q学習にディープラーニング適用	https://web.stanford.edu/class/psych209/Readings/MnihEtAlHassabis15NatureControlDeepRL.pdf
		Gorilla	2015	DQNを並列分散	https://arxiv.org/pdf/1507.04296.pdf
		Agent57	2020	DQN系最新	https://arxiv.org/pdf/2003.13350.pdf
	方策ベース ※確率論的, 連続的 (ロボット制御のように, 行動空間が連続な場合や多変数の場合に使われる)	Actor-Critic	1977	行動を決めるActor(行動器)を直接改善しながら, 方策を評価するCritic(評価器)も同時に学習させる	
		方策勾配法	1992	方策関数を確率的勾配法などで最適化	
		PPO	2017	方策勾配法系最新	https://arxiv.org/pdf/1707.06347.pdf
		Soft Actor-Critic	2019	Actor-Critic系最新	https://arxiv.org/pdf/1801.01290.pdf https://arxiv.org/pdf/1812.05905.pdf

- ✓ <https://qiita.com/shionhonda/items/ec05aade07b5bea78081>
- ✓ <https://qiita.com/fridericusgauss/items/aa5215c29646963bda29>
- ✓ https://twitter.com/ImAI_Eruel/status/1303677795806056451/photo/1
- ✓ <https://www.slideshare.net/yutarooogawa1116/20200925-238655136>

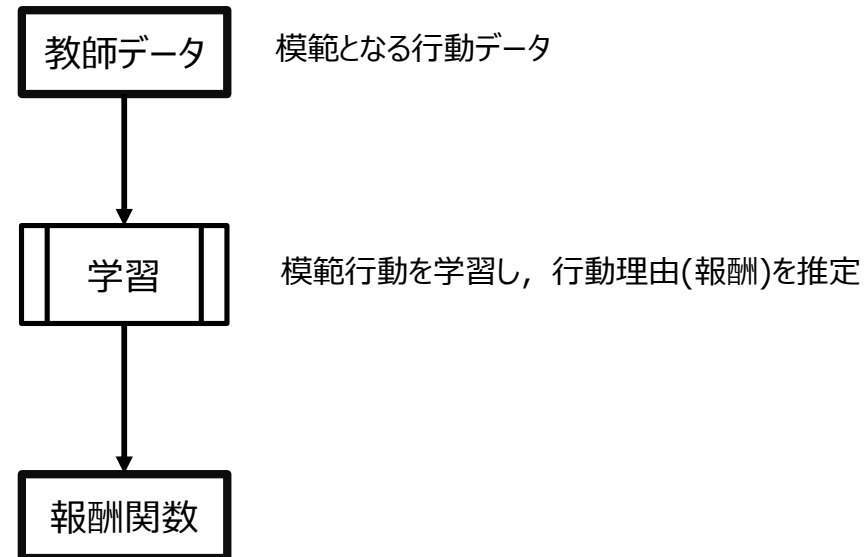
7. 学習の手法; (2)逆強化学習(IRL)

- ✓ 逆強化学習(IRL; Inverse Reinforcement Learning)は, 強化学習とは逆に模範的な行動を示す教師データを学習することで報酬関数を推定する
- ✓ 強化学習における報酬関数は人手で定義する必要があるが, 複雑なシーンでは困難であり, この点をカバーする

強化学習(RL)



逆強化学習(IRL)



https://twitter.com/ogawa_yutaro_22/status/1399831931311644672

7. 学習の手法; (2)逆強化学習 手法一覧

分類	特徴	手法	発表年	説明	論文等
マージン最大化	✓ 完全な教師データが必要	線形計画逆強化学習	2000	線形計画問題として報酬関数を解く	https://ai.stanford.edu/~ang/papers/icml00-irl.pdf
		Apprenticeship learning	2004	射影法を応用した解法を示す	https://ai.stanford.edu/~ang/papers/icml04-apprentice.pdf
		Max Margin Planning	2006	サポートベクタマシン(SVM)を使用	https://martin.zinkevich.org/publications/maximummarginplanning.pdf
ベイズ推定	✓ ある報酬の下で, エキスパートの軌跡が得られる確率をベイズ推定	ベイジアン逆強化学習	2007	モンテカルロ法の計算量が課題	https://www.aaai.org/Papers/IJCAI/2007/IJCAI07-416.pdf
		Map IRL	2011	モンテカルロ法の計算量が課題	https://www.semanticscholar.org/paper/MAP-Inference-for-Bayesian-Inverse-Reinforcement-Choi-Kim/8c802ca0f26177d2dda7664778ab7bb3337edfb7
	✓ 完全な教師データでなくとも頑健な推定が行える	Scalable Bayesian IRL	2021	モンテカルロ法を使用しない為計算量が改善	https://arxiv.org/pdf/2102.06483.pdf
最大エントロピー原理	✓ 報酬関数を関数近似し勾配法で解く	Maximum Entropy IRL	2008	報酬関数を線形関数で近似. 本方式で勾配法が導入されたことで, ディープラーニングに繋がってゆく	https://www.aaai.org/Papers/AAAI/2008/AAAI08-227.pdf
	✓ 完全な教師データでなくとも推定できる	Deep Maximum Entropy IRL	2015	ディープラーニングによるMaximum Entropy IRL	https://arxiv.org/abs/1507.04888
	✓ 勾配法が適用できるため, ディープラーニングに発展可能	Guided Cost Learning	2016	Adversarial IRLの前身	https://arxiv.org/abs/1603.00448
		Adversarial IRL	2018	GAN(Generative Adversarial Networks: 敵対的生成ネットワーク)を使用したIRL	https://arxiv.org/pdf/1710.11248.pdf

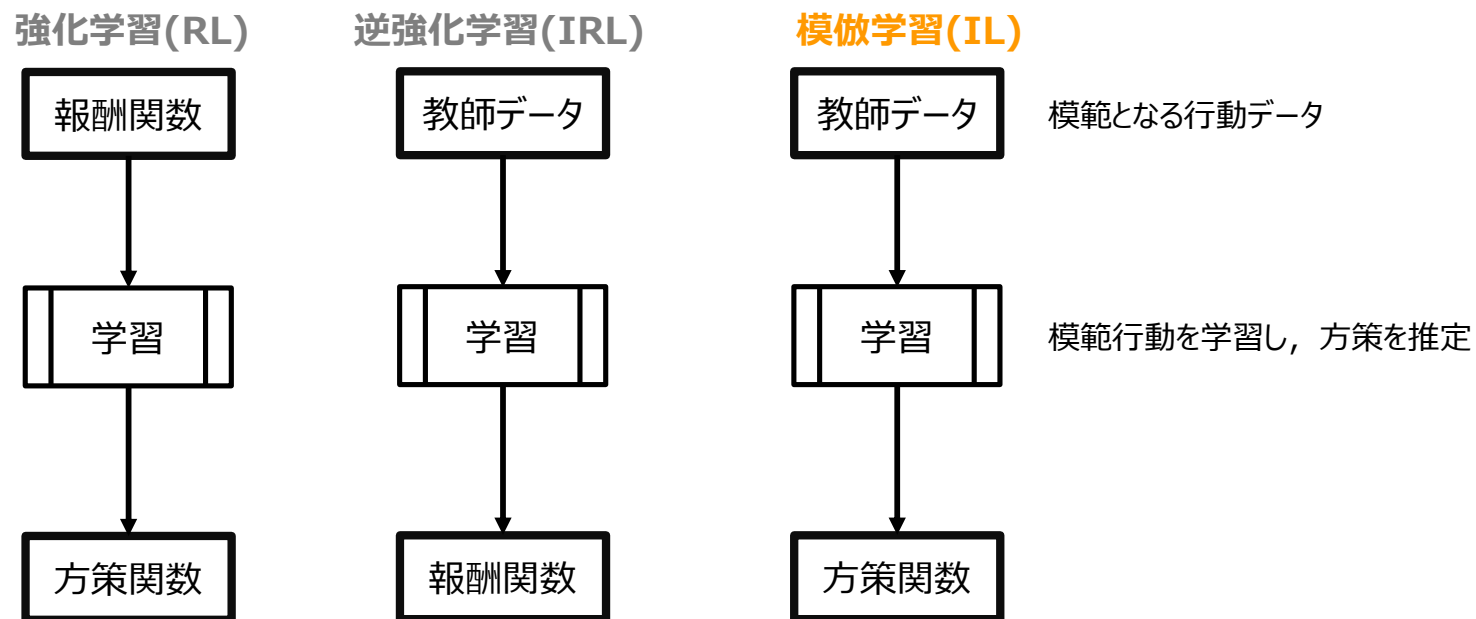
✓ https://opac.ll.chiba-u.jp/da/curator/900119578/WLA_0036.pdf

✓ <https://qiita.com/shiro-kuma/items/aaab6184aea7d285b103>

✓ <https://ai-scholar.tech/articles/%E9%80%86%E5%BC%B7%E5%8C%96%E5%AD%A6%E7%BF%92/scalable-birl-avirl>

7. 学習の手法; (3)模倣学習(IL)

- ✓ 模倣学習(IL; Imitation Learning)は, 模範的な行動を示す教師データを学習することで方策関数を推定する



https://twitter.com/ogawa_yutaro_22/status/1399831931311644672

7. 学習の手法; (3)模倣学習 手法一覧

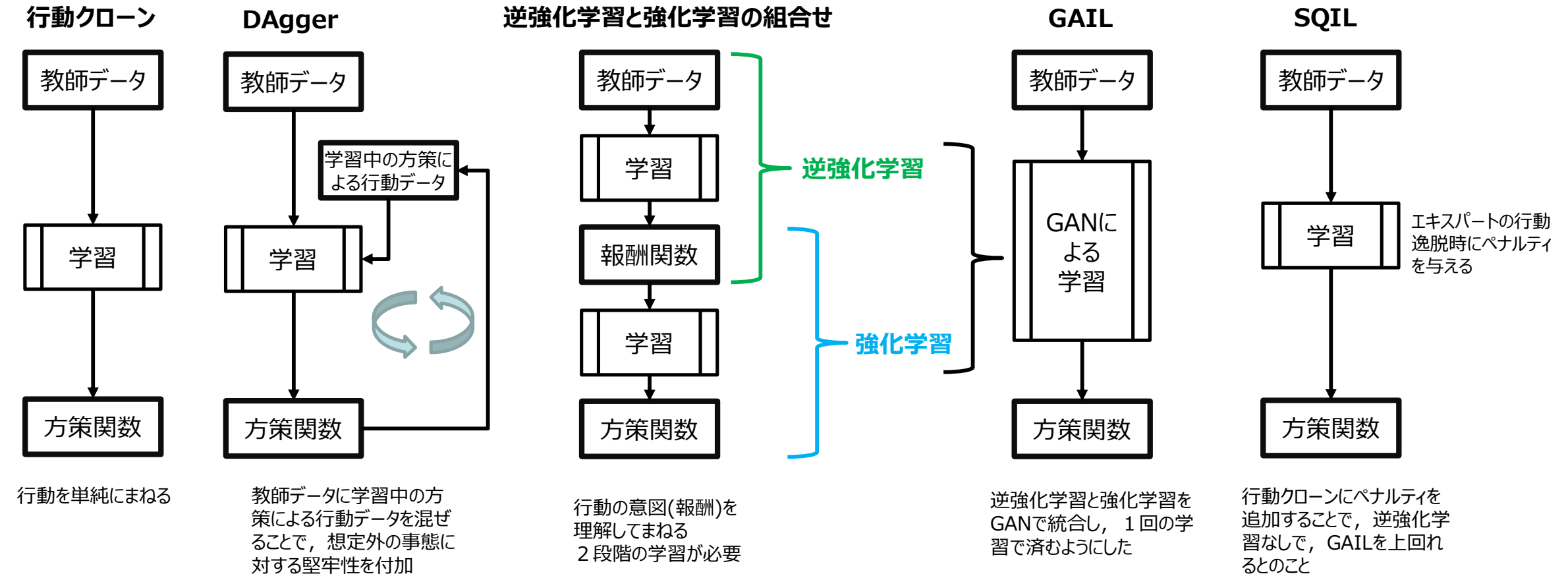
手法	発表年	説明	論文等
行動クローン(Behavoir Cloning)	1999	エキスパートの単純な模倣. 想定外の状況に対して脆弱 (一度間違うと回復できない)	
DAgger(Dataset Aggregation)	2010	行動クローンの欠点を克服するため, 学習時にエキスパートの模範データに自身の学習中の方策による行動データも混ぜ込んで繰り返し再学習し, 方策を更新してゆくことで, 想定外の状況に陥ったとしてもエキスパートの行動に回復できるようにすることを狙う	https://arxiv.org/pdf/1011.0686.pdf
逆強化学習と強化学習の組合せ		次の2段階によって行う ①エキスパートの模範行動を教師データとしてIRLを実行して報酬関数を得る ②IRLで得た報酬関数を入力としてRLを実行して方策関数を得る 単純な行動クローンより, エクスパートの意図(報酬)を学習することで, 方策が向上するとされる	
GIRL(Generative Adversarial Imitation Learning)	2016	逆強化学習と強化学習の組合せによる模倣学習の構造を変形すると, GANと同じ構造になることから, GANによって逆強化学習と強化学習を統合して一回の学習で完了できるようにしたもの ※GAN(Generative Adversarial Networks; 敵対的生成ネットワーク) 以下の2つのネットワークを競争させることで学習(敵対的学習)するネットワーク. 入力データに類似したデータを生成することができる ・Generator(教師データを模倣してデータを生成) ・Discriminator(Generatorが生成したデータの真贋を鑑定する)	https://arxiv.org/pdf/1606.03476.pdf
SQIL(Soft Q Imitation Learning)	2019	行動クローンを改良したシンプルな手法でGAILを上回ったという. 行動クローンでエキスパートの行動から外れた場合にペナルティを与える(報酬を0にする)ことで, GAILの考え方を取り入れつつも逆強化学習なしで実現できるとしている.	https://arxiv.org/pdf/1905.11108.pdf

✓ <https://recruit.gmo.jp/engineer/jisedai/blog/imitation-learning/>
 ✓ <https://ailog.site/2019/11/11/word6/>
 ✓ <https://qiita.com/3110atsu1217/items/c44130d59a0406b148c3>

✓ <https://qiita.com/ikeyasu/items/512e0f42e3845068f433>
 ✓ <https://tech.morikatron.ai/entry/2020/10/12/100000>
 ✓ https://kiyosocyberclub.web.fc2.com/OpenAI_Gym/Gym02-03.html

7. 学習の手法; (3)模倣学習 手法図解

ILの各手法を図にまとめる



7. 学習の手法; (4)能動学習(AL)

- ✓ 能動学習(AL; Active Learning)は, 教師あり学習において, 優先的にアノテーションすべきデータを自動選定することで, 効率的に学習するしくみ. 以下のような手法が存在する.

- ✓ Membership Query Synthesis

モデルの学習に有効なデータを疑似的に生成する

- ✓ Stream-Based Selective Sampling

各データ毎にラベルを付けるかどうか判断する

- ✓ Pool-Based Sampling

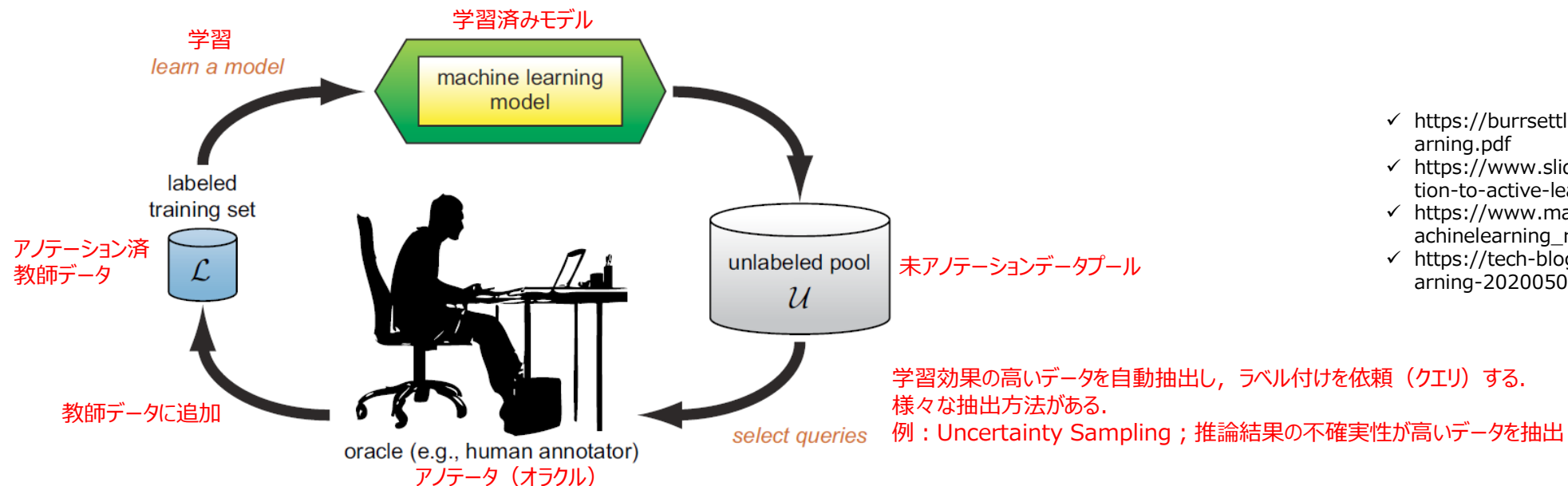
未アノテーションのデータの集合 (プール) から, 学習効果の高いデータを抽出する (下図. 最も一般的)

【参考】

ALの効果を疑問視する論文も存在する

簡単なData Augmentationを導入した方が効果的とのこと

<https://arxiv.org/pdf/1912.05361.pdf>



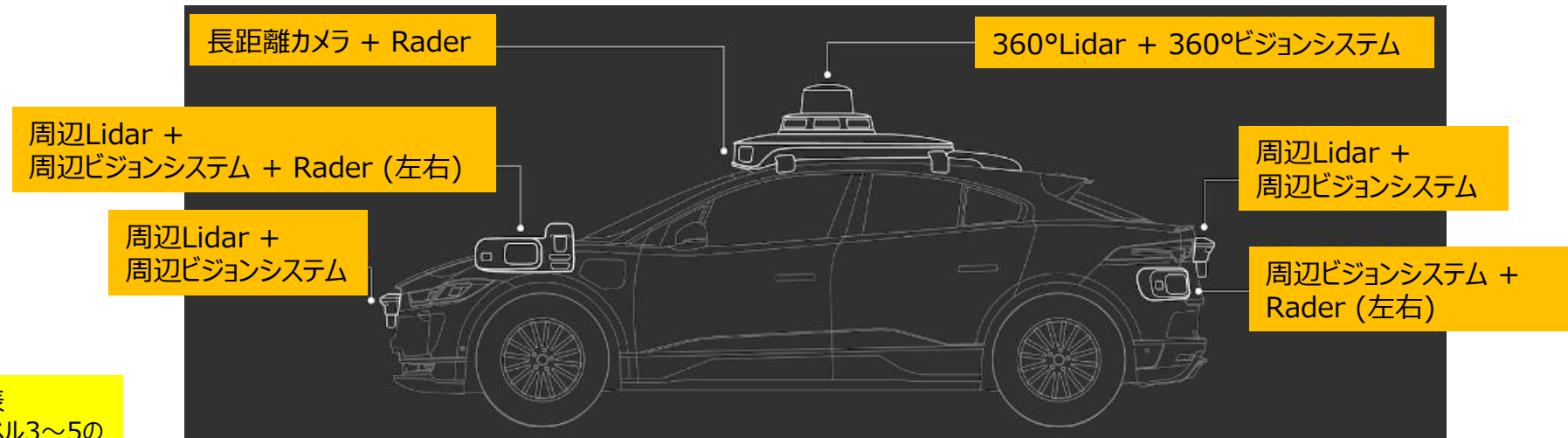
- ✓ <https://burrsettles.com/pub/settles.activelearning.pdf>
- ✓ <https://www.slideshare.net/shuyo/introduction-to-active-learning-25787487>
- ✓ https://www.mamezou.com/techinfo/ai_machinelearning_rpa/ai_tech_team/5
- ✓ <https://tech-blog.abeja.asia/entry/active-learning-20200503>

8. 学習を用いた事例; (1)Waymo 概要



第5世代 Waymo Driver

- 2020年10月アリゾナ州フェニックスで完全無人自動運転タクシー「Waymo One」運用開始
 - SafetyDriver非搭載
 - 従業員, TrustedTesterのみから徐々に一般に拡大する
- 2022年3月サンフランシスコで完全無人自動運転タクシー「Waymo One」運用開始
 - SafetyDriver非搭載
 - 現在, 従業員のみ



センサ構成

【参考】NHTSA2022年6月発表
2021年7月以降の自動運転レベル3～5の
ADS車事故件数
Waymo 62件で首位
https://jidounten-lab.com/u_35918

<https://blog.waymo.com/2020/03/introducing-5th-generation-waymo-driver.html>

8. 学習を用いた事例; (1)Waymo 機械学習プランナー

- 現在のWaymoのプランナーは機械学習と明示的な推論を組合わせたもので、1000万マイル以上の公道と数十億マイルのシミュレーションでテストされており、様々なシナリオで最良の運転決定を下す
- 完全な機械学習によるプランナーはまだテスト段階であり、2018年、米国カリフォルニアにあるWaymo社の自動運転テストコース（Castle）で実車テストを行った段階



機械学習プランナー搭載車と同型の車両
(第4世代Waymo Driver Chrysler Pacifica)

- ✓ <https://sites.google.com/view/waymo-learn-to-drive/>
- ✓ <https://arxiv.org/abs/1812.03079>
- ✓ https://blog.waymo.com/2019/08/learning-to-drive-beyond-pure-imitation_26.html
- ✓ <https://medium.com/waymo/learning-to-drive-beyond-pure-imitation-465499f8bcb2>
- ✓ <https://xtech.nikkei.com/atcl/nxt/mag/rob/18/012600001/00019/>

How a Driverless Car Works

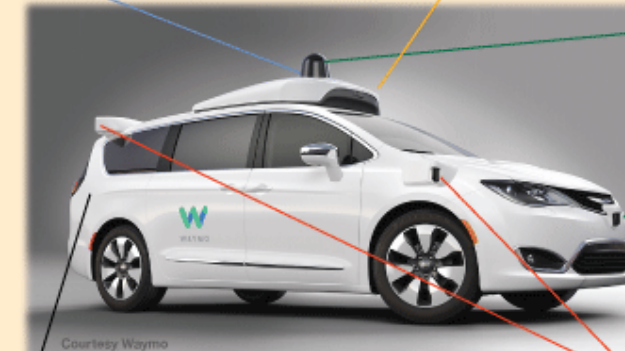
Driverless vehicles use lasers, cameras, computer processors and software, radar, GPS and lidar (light detection and ranging) devices to create and constantly update a 3D map of their surroundings. Self-driving car companies also use artificial intelligence and machine learning to program the vehicles to learn from their experiences and act more like human drivers.

Vision System

High-resolution cameras with a 360-degree view work at long range and in low light.

Supplemental Sensors

GPS helps the vehicle pinpoint its location; an audio detection system "hears" emergency sirens hundreds of feet away.



Lidar

Emits millions of laser pulses each second and measures how long each takes to reflect off a surface and return.

Onboard Computer

Software uses machine learning to predict how objects on the road will behave; a secondary computer can stop the vehicle if the main computer fails.

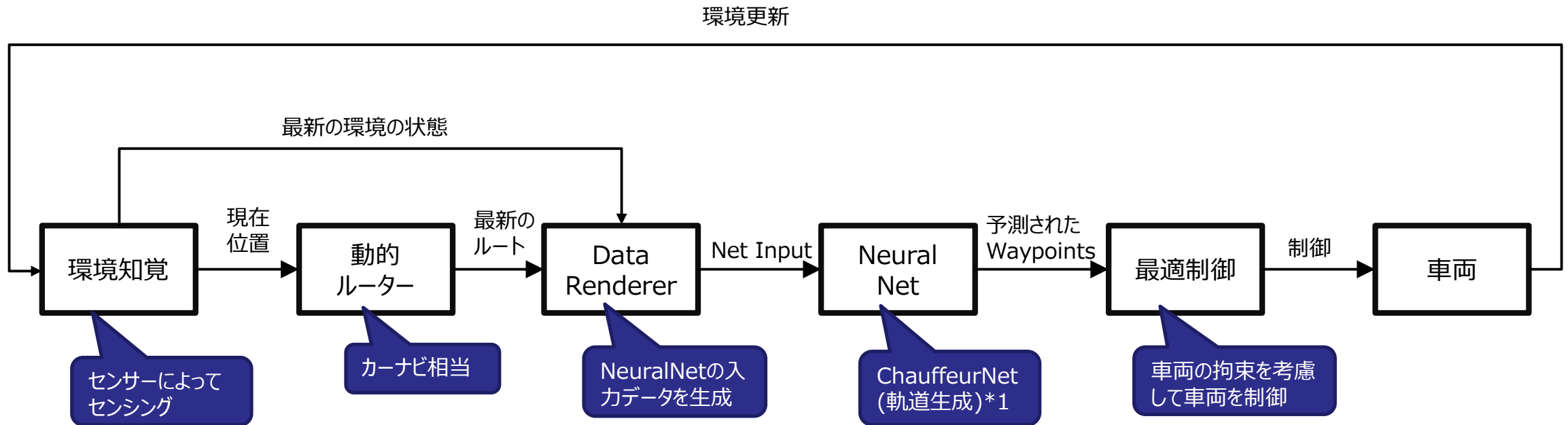
Radar

Radio waves track the speed of other objects around the vehicle.

Source: Waymo

<https://library.cqpress.com/cqresearcher/document.php?id=cqresrre2019020102>

8. 学習を用いた事例; (1)Waymo 機械学習プランナーのソフトウェアアーキテクチャ



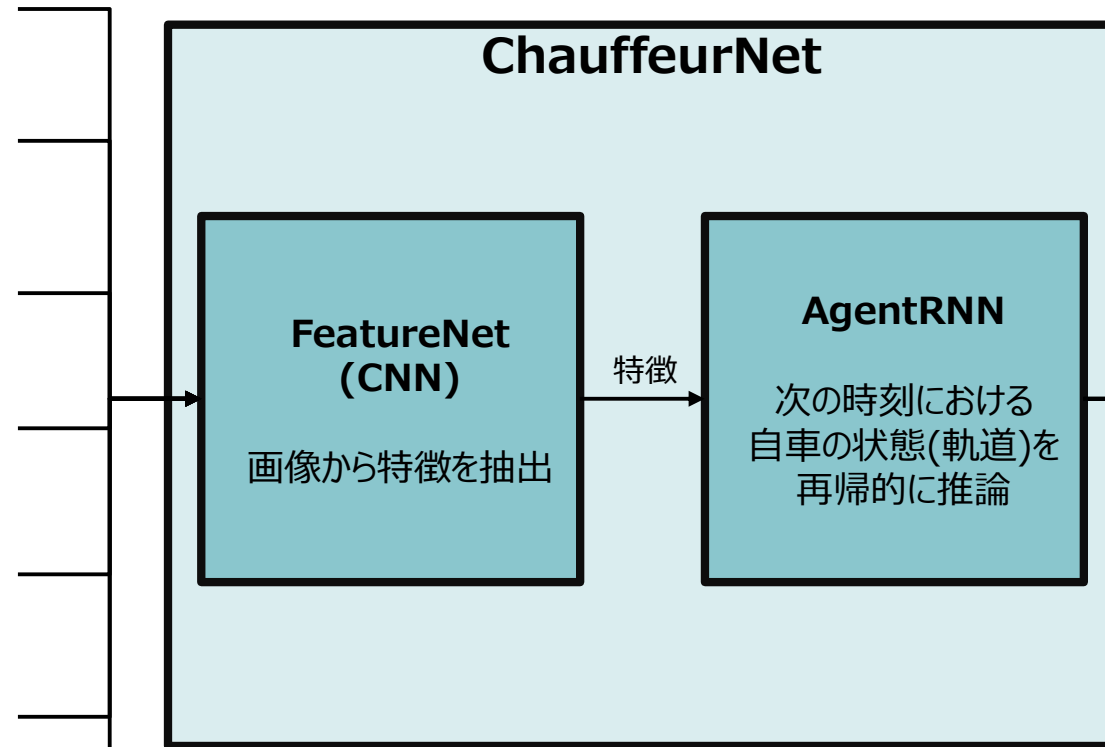
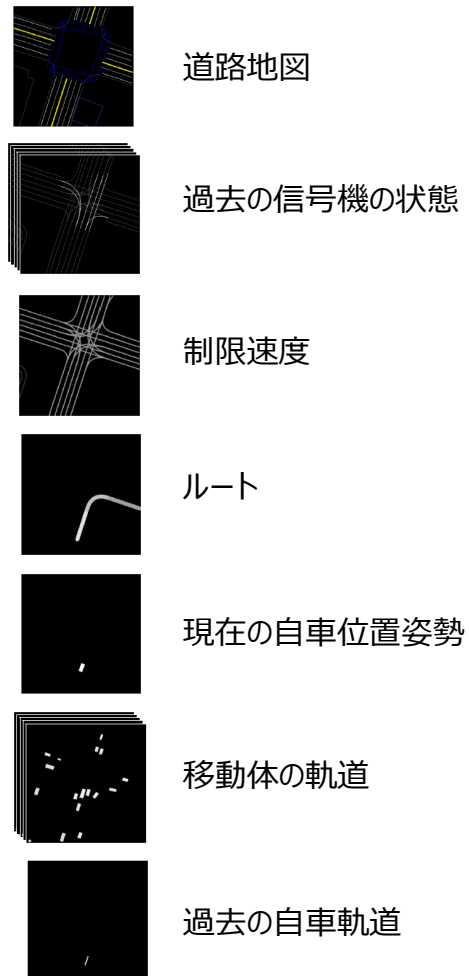
*1 ディープラーニング(教師あり模倣学習)による軌道生成方式を採用

<https://arxiv.org/abs/1812.03079>

8. 学習を用いた事例; (1)Waymo NeuralNet「ChauffeurNet」

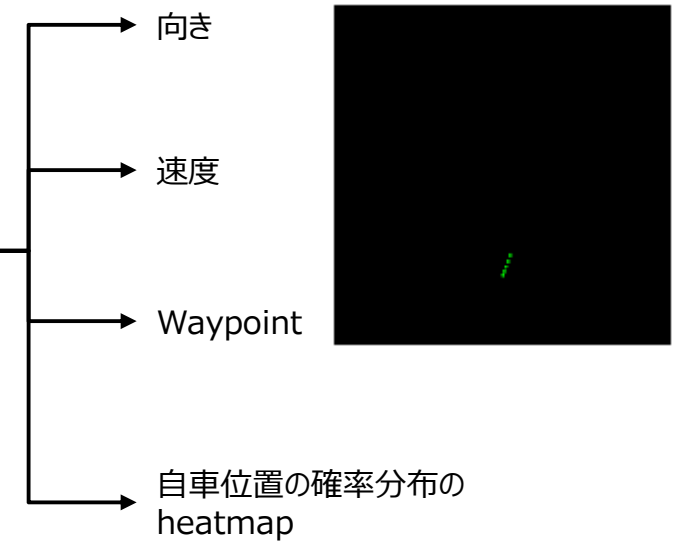
【入力データ】

周囲の状況をTop-down view画像にレンダリング



【出力データ】

予測された向き, 速度, Waypointおよび自車位置予測Heatmap



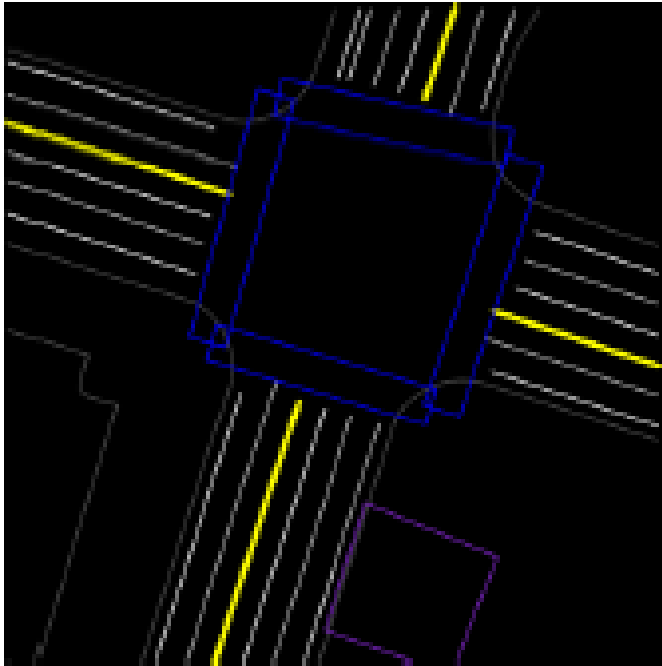
学習方法: エキスパートによる模範的な運転例を教師データとした模倣学習

<https://arxiv.org/abs/1812.03079>

8. 学習を用いた事例；(1)Waymo 入力データ①

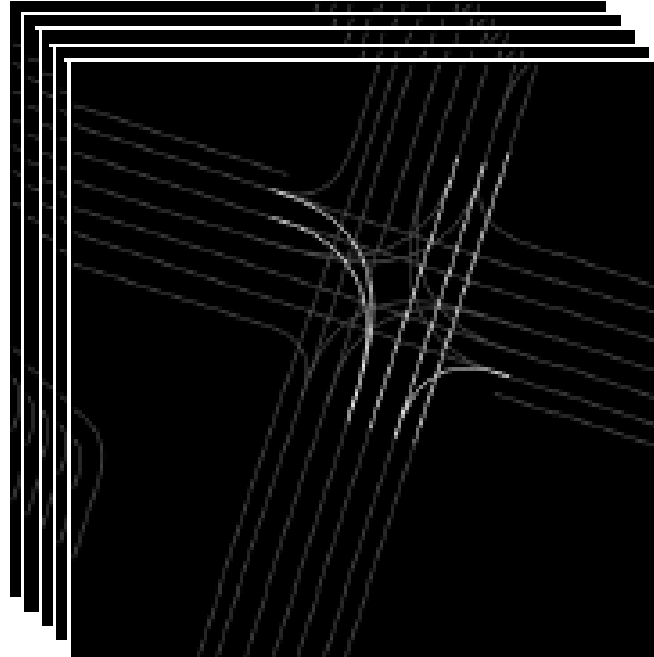
(a)道路地図

車線，停止標識，横断歩道，縁石等を色分けしてレンダリングしたカラー画像



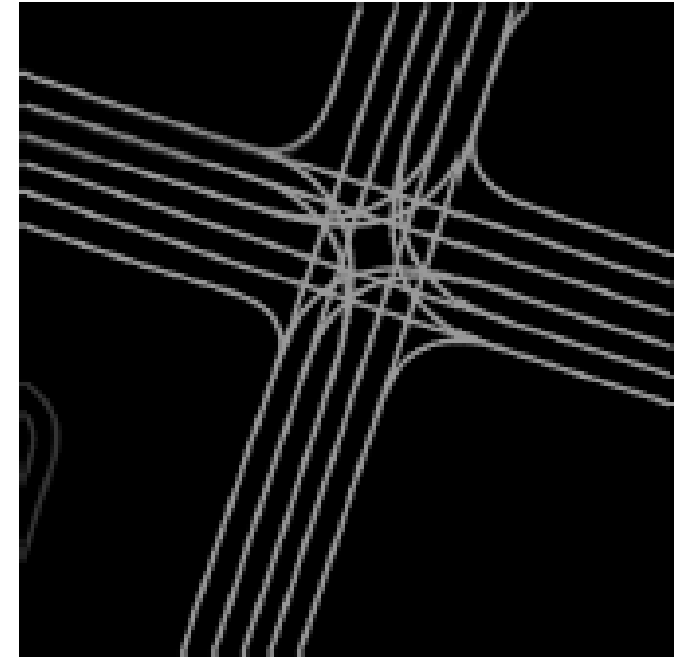
(b)過去の信号機の状態

過去のタイムステップにおける信号機の状態を示すグレースケール画像
明るさは信号の状態を示す
(赤：もっとも明るい，黄色：中間，青または不明：暗い)



(c)制限速度

制限速度に応じて車線中心を色分けしたグレースケール画像

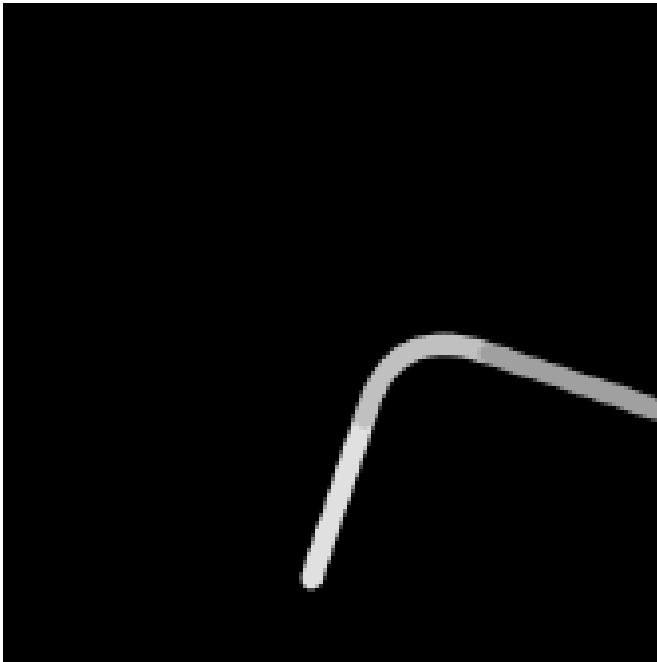


<https://arxiv.org/abs/1812.03079>

8. 学習を用いた事例; (1)Waymo 入力データ②

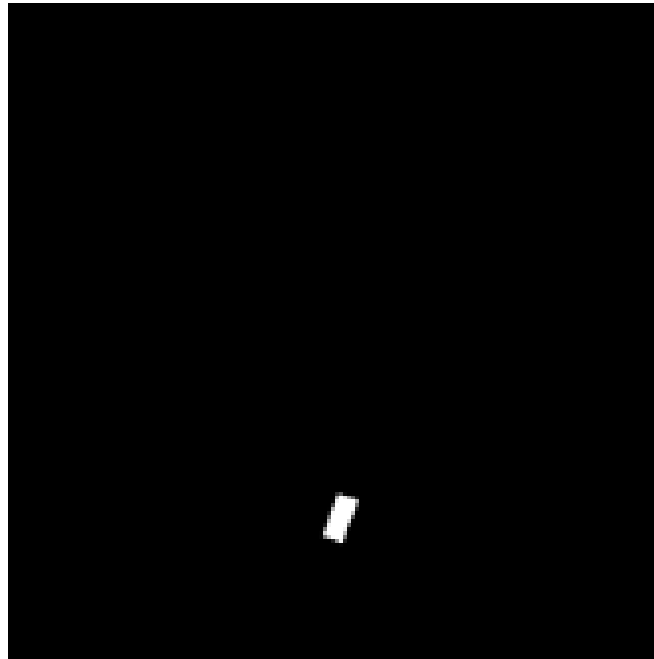
(d)ルート

ルーターによって生成された目的ルート



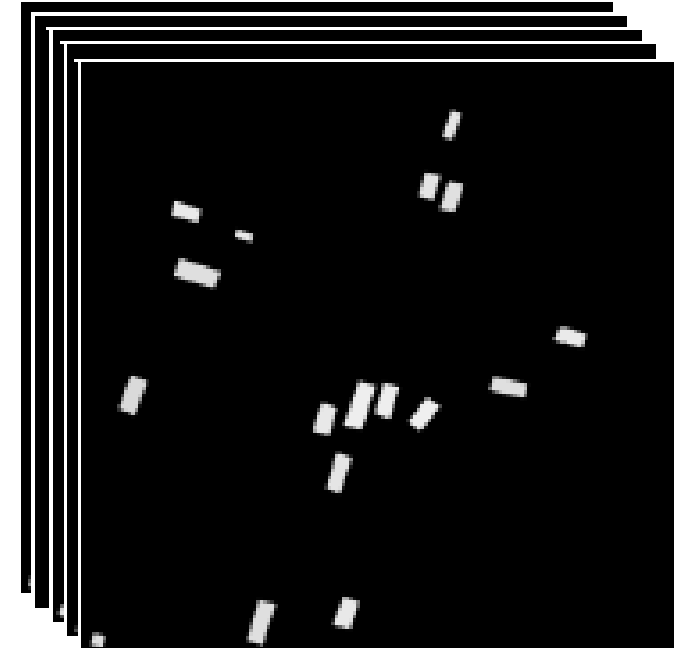
(e)現在の自転車位置姿勢

現在のタイムステップの自転車位置と姿勢
(自転車のBounding Box)



(f)移動体の軌道

環境内の動的オブジェクト(車両, サイクリスト, 歩行者)のBoundingBoxの, 過去のタイムシーケンス毎の位置と姿勢(軌道)

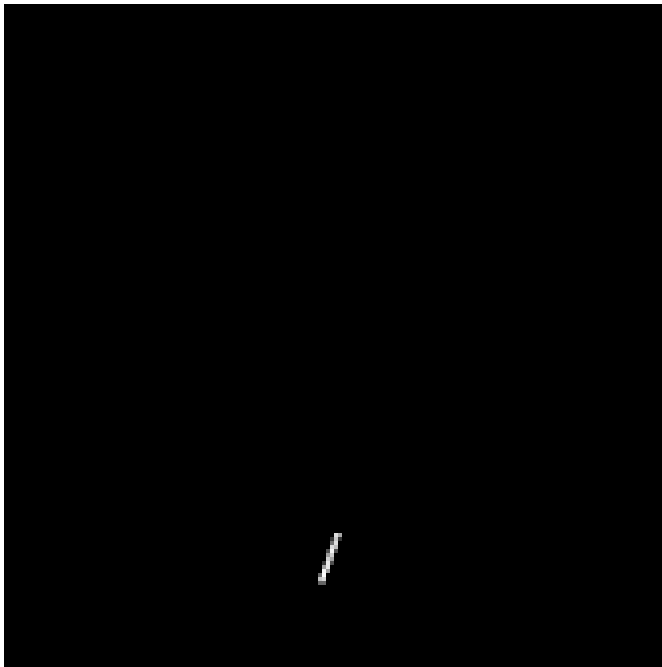


<https://arxiv.org/abs/1812.03079>

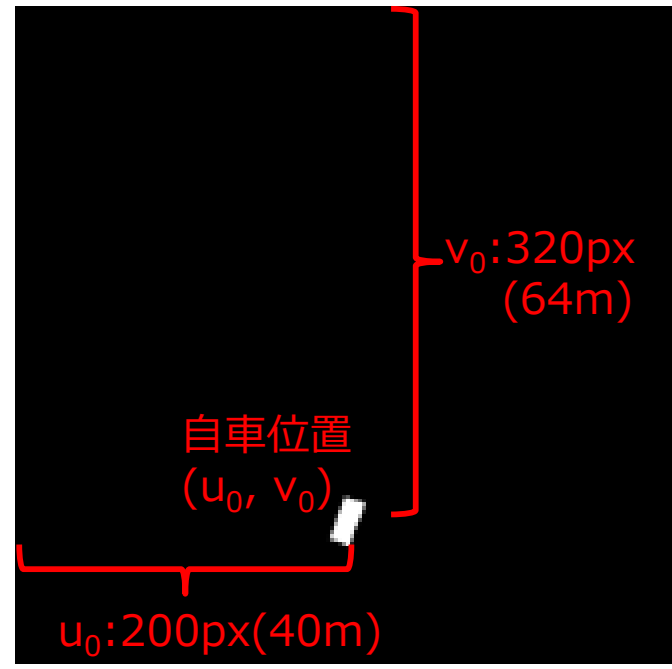
8. 学習を用いた事例; (1)Waymo 入力データ③

(g)過去の自車軌道

過去の自車位置姿勢をポイントの軌跡としてレンダリングした画像



入力画像のパラメータ（論文の実験時における値）



- ✓ ピクセル解像度 : 0.2m/px
- ✓ 過去の信号や移動体のサンプリング時間 : 10s
- ✓ 過去の自車軌道のサンプリング時間 : 1s

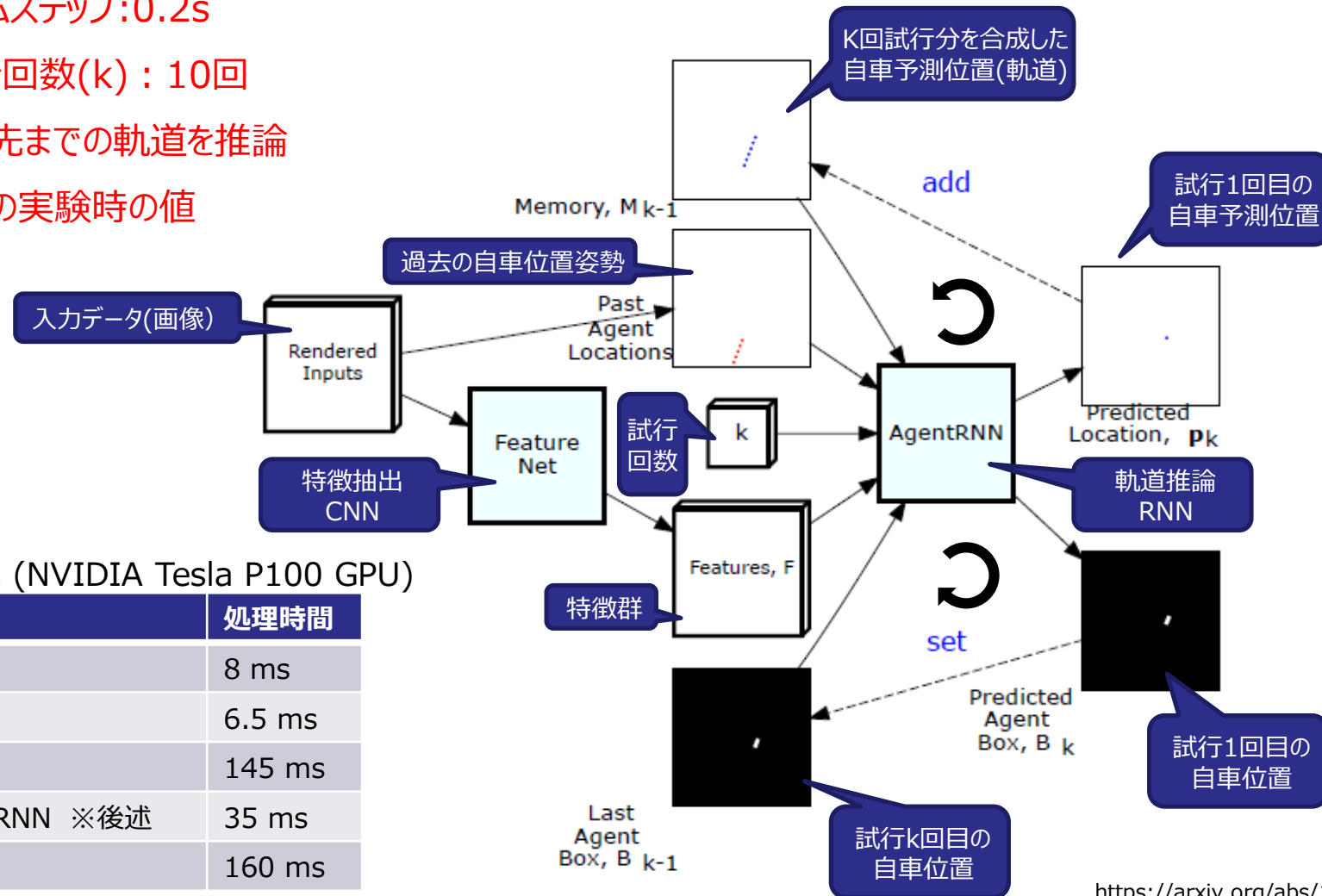
H: 400px(80m)

W: 400px(80m)

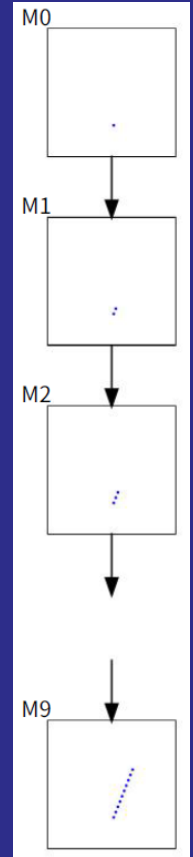
<https://arxiv.org/abs/1812.03079>

8. 学習を用いた事例; (1)Waymo ChauffeurNetの動作

- ✓ タイムステップ: 0.2s
- ✓ 試行回数(k) : 10回
- ⇒ 2秒先までの軌道を推論
- ※論文の実験時の値



軌道RNN推論の
試行毎に予測位置が
追加されていくイメージ



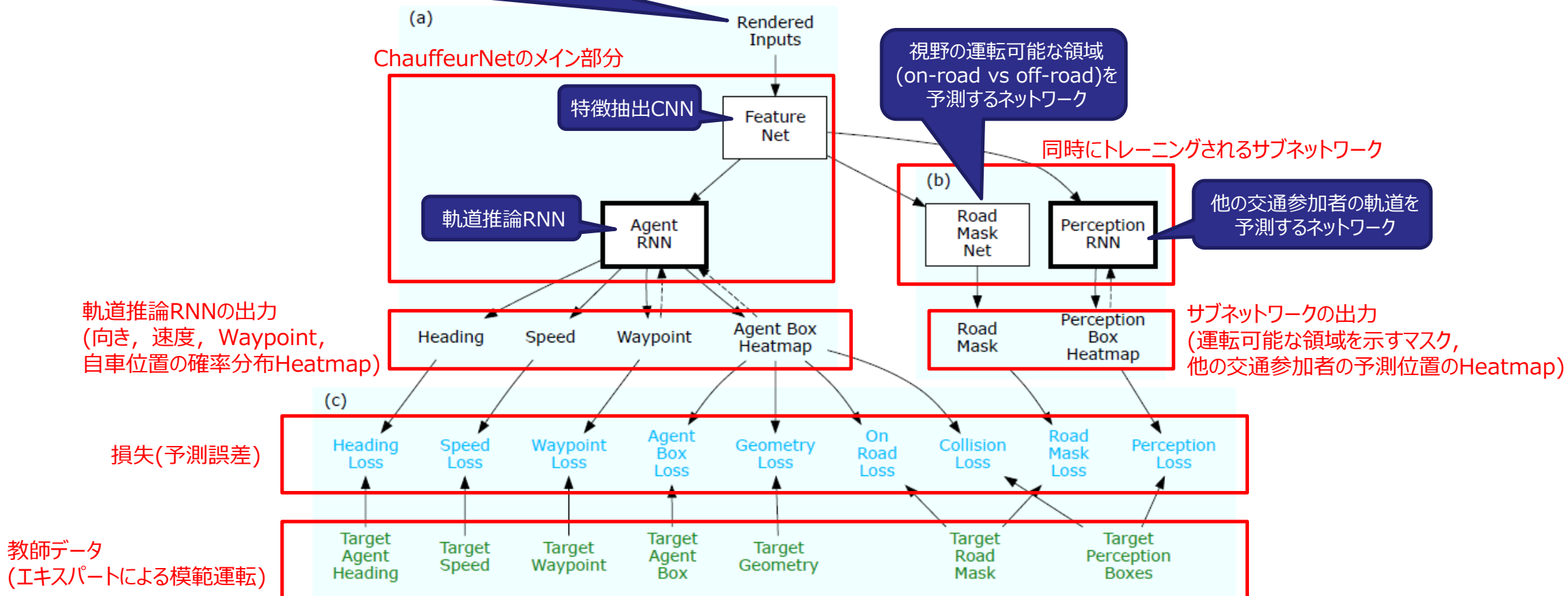
Runtime性能 (NVIDIA Tesla P100 GPU)

処理	処理時間
Rendering	8 ms
FeatureNet	6.5 ms
AgentRNN	145 ms
PerceprionRNN ※後述	35 ms
合計	160 ms

<https://arxiv.org/abs/1812.03079>

8. 学習を用いた事例; (1)Waymo ChauffeurNetの学習

入力データ(画像)
データ数: 2600万件(60日分の連続運転相当)
Data Augmentationとして, 自車位置姿勢をランダムに $\pm 25^\circ$ の範囲で回転



<https://arxiv.org/abs/1812.03079>

8. 学習を用いた事例；(1)Waymo ChauffeurNetの学習 過去の自車位置姿勢の一部除外

- ✓ 入力データの一つに「過去の自車軌道」が含まれている。過去の軌道はエキスパートドライバのものであるため、ChauffeurNetは過去軌道から外挿するだけの「チート」を学んでしまう。これは実際の走行時には破綻する。例えば、以下のように間違った学習が行われる恐れがある

一時停止標識で停止することを学習する際に、教師データの過去の軌道はエキスパートドライバのものであるため、「過去の軌道で（一時停止に備えて）減速されている場合のみ、一時停止標識で停止する」というように学習されてしまう場合がある

これに対処するため、教師データの50%から過去の自車軌道を除外した。
これにより、過去の自車軌道に頼り過ぎないで判断する学習が可能となった

<https://arxiv.org/abs/1812.03079>

8. 学習を用いた事例; (1)Waymo ChauffeurNetの学習 摂動(外乱)データの追加

- ✓ 教師データをベースに、軌道摂動(外乱)データを作成して追加した.

(ただし、摂動データの軌道が誤って学習されないように、重みを1/10とした)



オリジナル

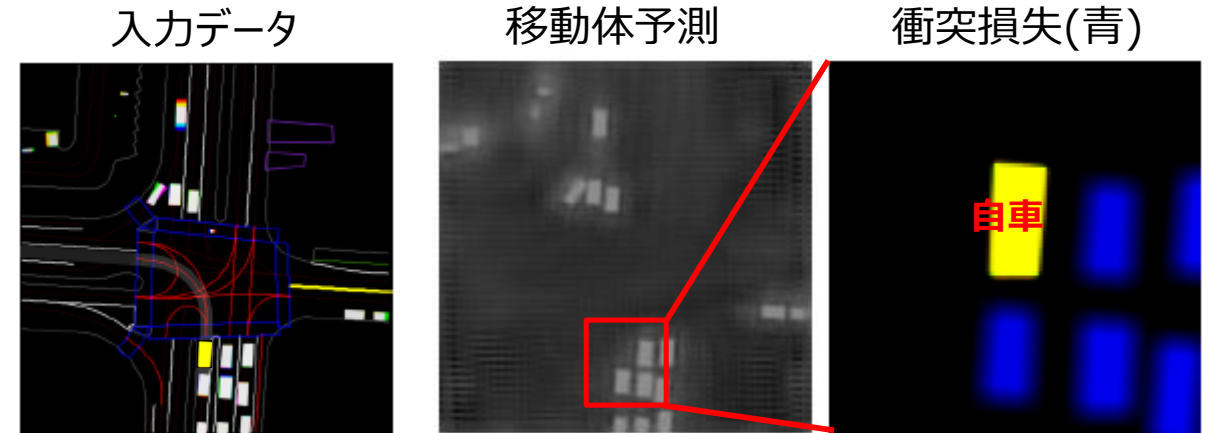


外乱を追加

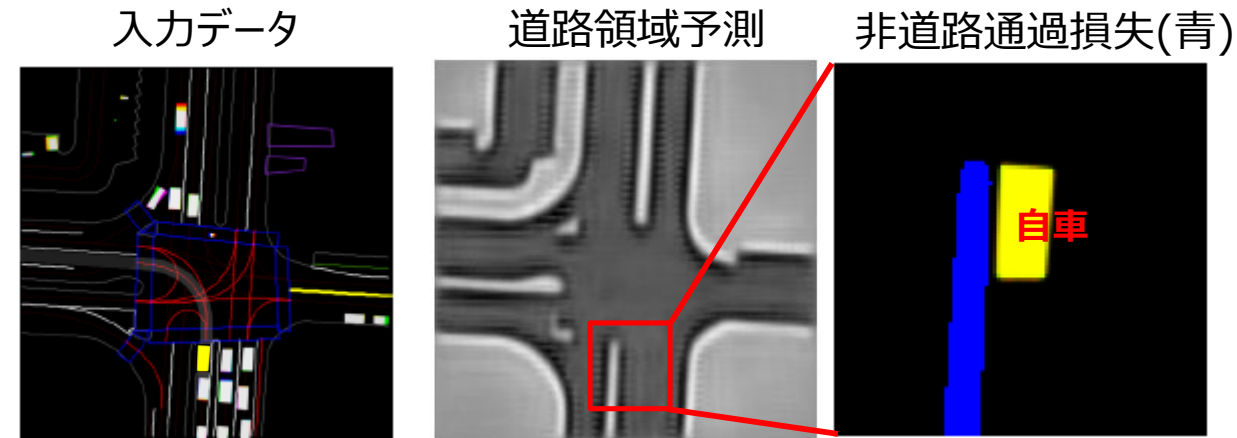
<https://arxiv.org/abs/1812.03079>

8. 学習を用いた事例; (1)Waymo ChauffeurNetの学習 環境損失の追加①

- ✓ 教師データに衝突の事例がないため, 学習中に他の交通参加者に衝突するような軌道を推論した場合に間違いを修正できるように, 衝突検出時に損失が大きくなるような特殊な損失を導入した (衝突損失)



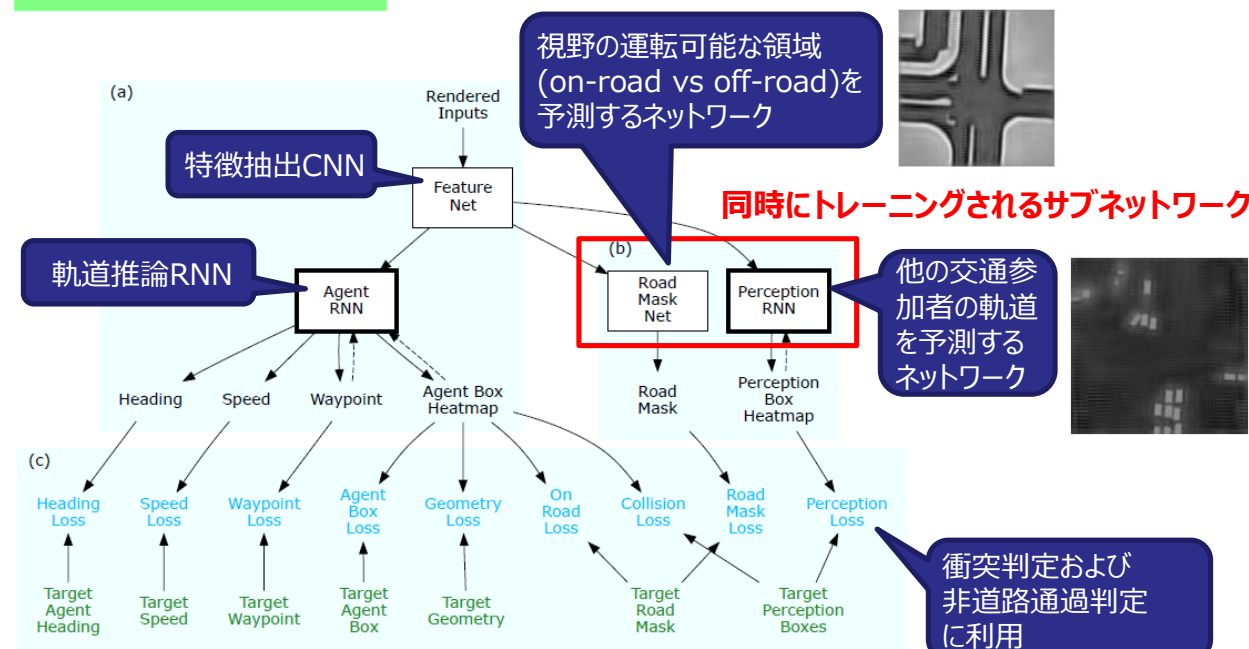
- ✓ 追加した外乱データにより, 中央分離帯や縁石に乗り上げるケースが生成される. これを誤って学習しないように, 道路と非道路領域を示すマスクを導入し, 非道路を通過する軌道を推論した場合に損失が大きくなるような, 特殊な損失を導入した (道路損失)



<https://arxiv.org/abs/1812.03079>

8. 学習を用いた事例; (1)Waymo ChauffeurNetの学習 環境損失の追加②

- ✓ ジオメトリに沿った軌道が生成されるように、推論された Waypointをなめらかな車幅と等しい幅の曲線にフィッティングし、各Waypointがこれを逸脱すると損失が大きくなるような、特殊な損失を導入した（ジオメトリ損失）
- ✓ 他の交通参加者の軌道推論と道路／非道路判定を行うサブネットワークと損失を追加した。AgentRNNと同じ FeatureNetを入力とし、同時にトレーニングすることで FeatureNetが両者に適する形でトレーニングされることが期待できる。出力損失は前述の衝突判定や非道路追加判定に利用する



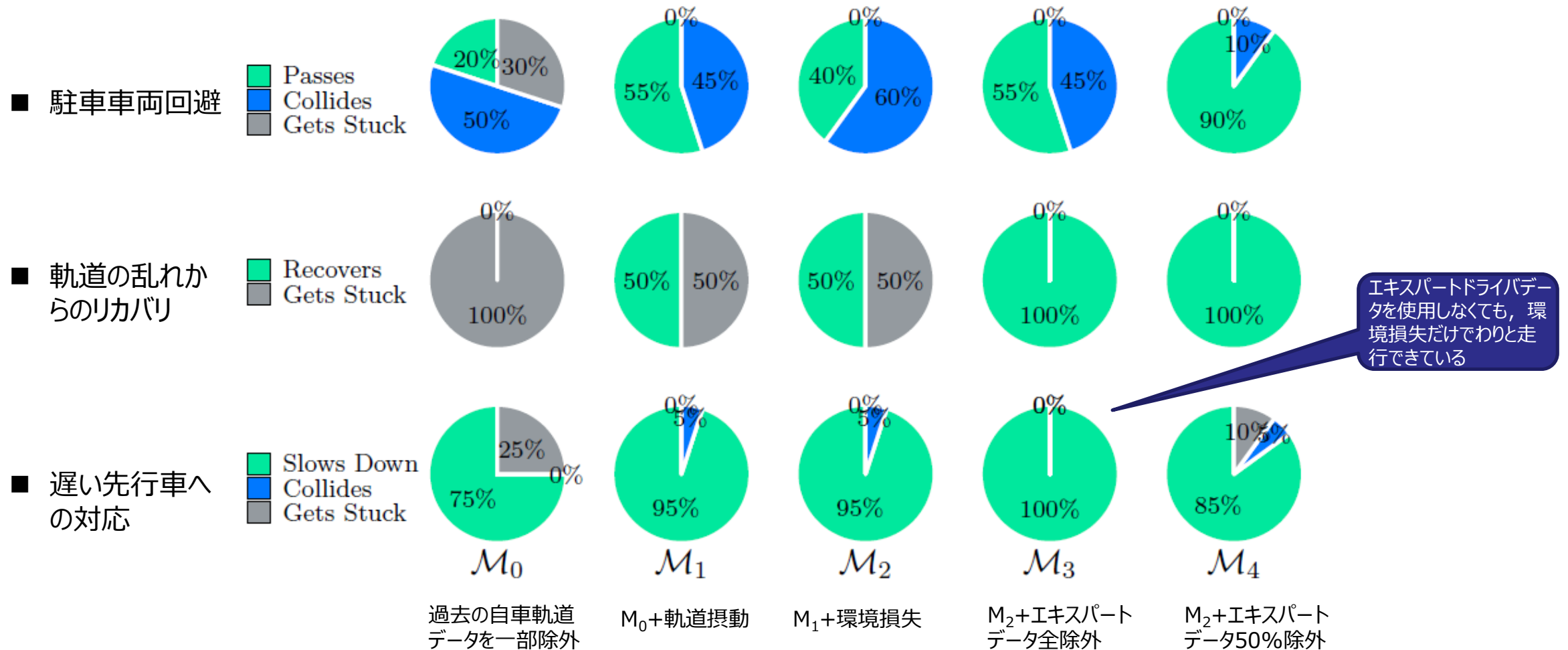
<https://arxiv.org/abs/1812.03079>

8. 学習を用いた事例; (1)Waymo ChauffeurNetの学習 エキスパートドライバデータの一部除外

- ✓ エキスパートドライバの教師データが過大評価されて, 追加した環境損失が過小評価されないように, エキスパートドライバの教師データの半数をランダムに抽出し, これらの重みを 0 とした (imitation dropout)

<https://arxiv.org/abs/1812.03079>

8. 学習を用いた事例；(1)Waymo ChauffeurNetの学習 各モデルの性能



<https://arxiv.org/abs/1812.03079>

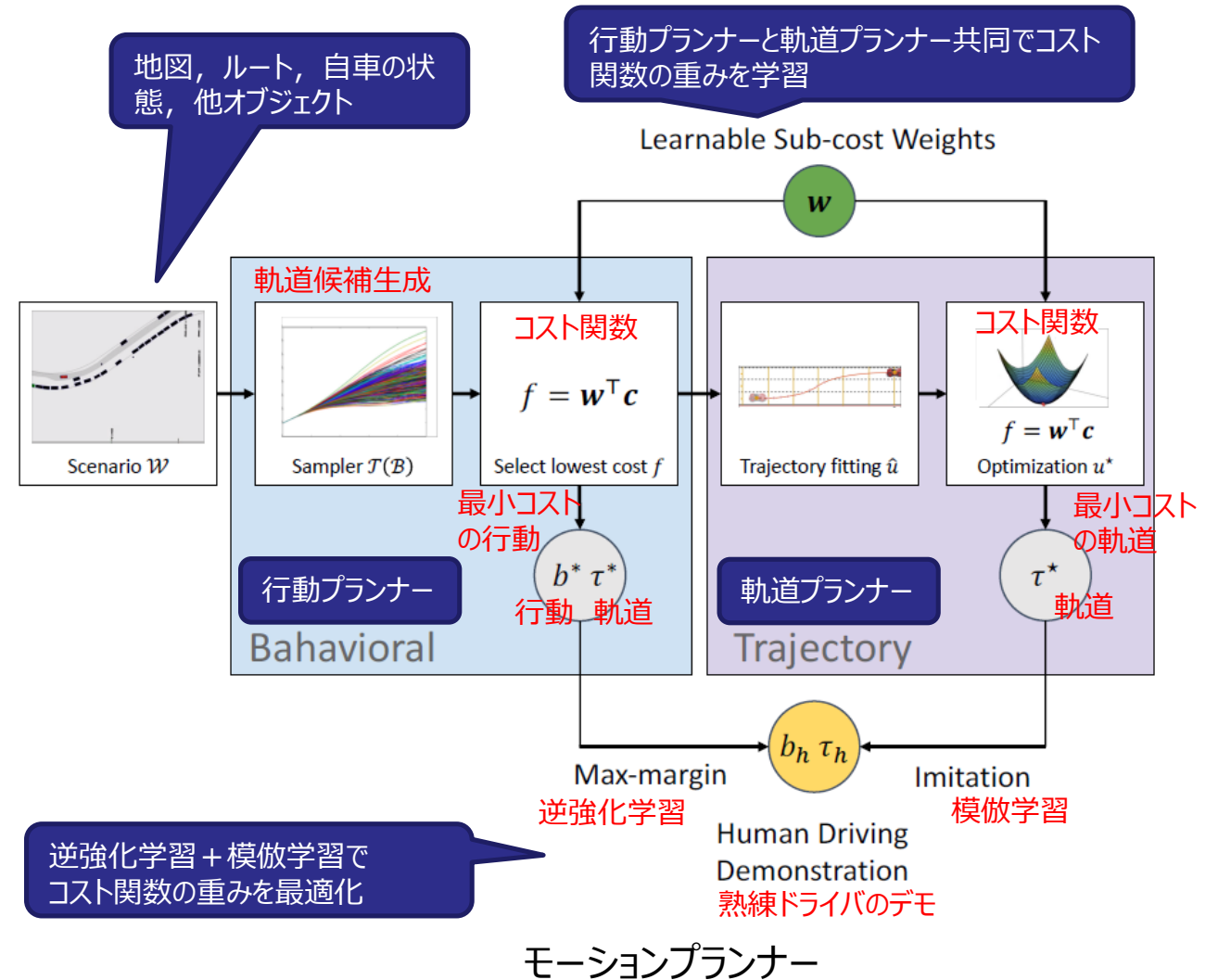
ビデオ : <https://sites.google.com/view/waymo-learn-to-drive/>

8. 学習を用いた事例; (1)Waymo まとめと課題

- ✓ 画像による中間レベルの入出力表現により, データ収集と, 知覚と制御を学習する負担を軽減できた
- ✓ 実際の運転で模倣学習をうまく実行するためには, 望ましくない行動を思いとどまらせる**適切な損失を追加することが必要**. これにより, エキスパートの教師データに望ましくない行動例が明示的に存在していなくても, 衝突や道路外走行を回避することを可能にした
⇒おそらく緊急回避のような軌道は学習できていないので, 緊急時対応は難しいと思われる
- ✓ このモデルはまだモーションプランニングアプローチを置き換えるものではないが, これは機械学習された運転モデルにとって良い一歩である. しかし, 改善の余地がある (エンドツーエンドのアプローチとの比較, エキスパートドライバの教師データ半数削除の代替案 等)
- ✓ おそらく**強化学習フレームワーク**によって, シミュレーション上で, まれで困難なシナリオを徹底的に調査・再現してこのような模倣学習モデルを強化することが, 特に高度にインタラクティブなシナリオで, これらのモデルのパフォーマンスを向上させる鍵になる

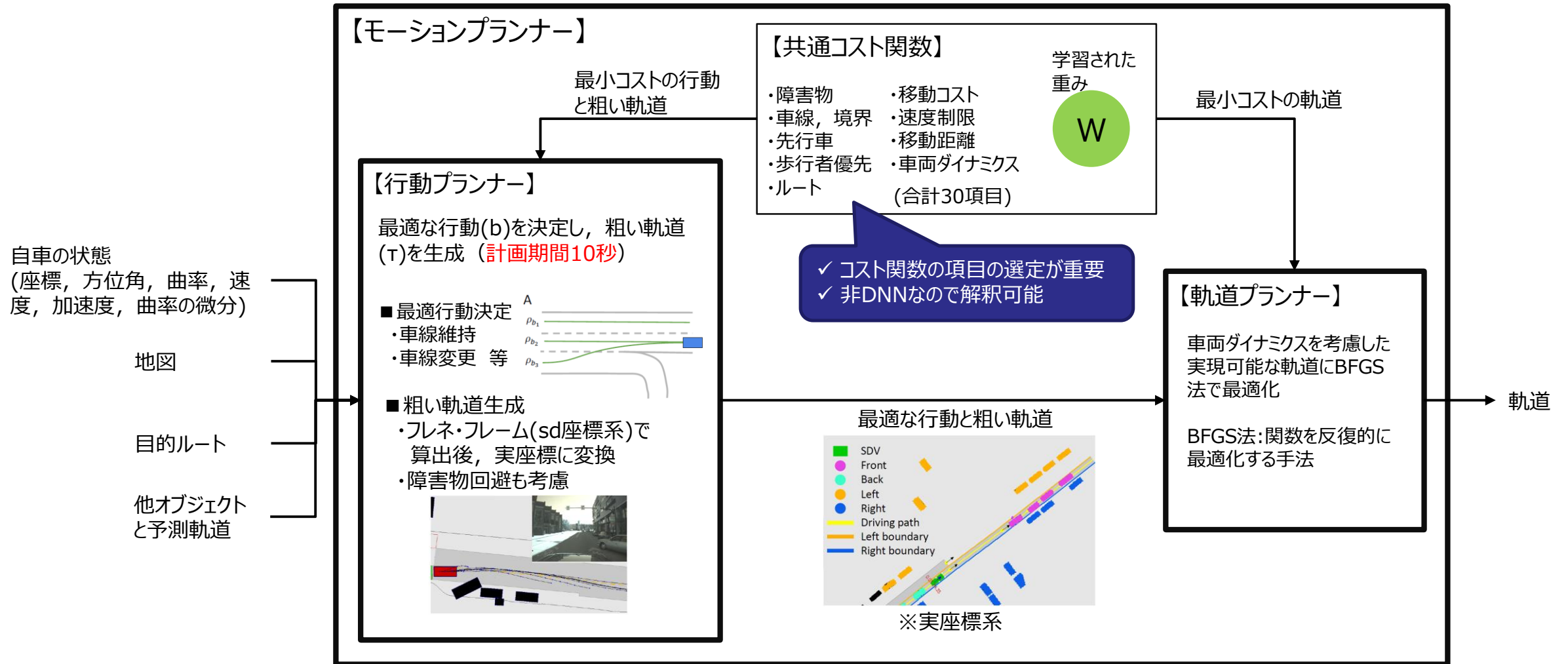
8. 学習を用いた事例; (2)Sadat (Uber)らの方法 概要

- ✓ SadatらはUberのエンジニア
- ✓ WaymoやApolloの事例を視野に改善提案（2019年の比較的新しい論文）
- ✓ ルールベースや状態マシンに基づくプランナーでは、複雑な都市シナリオを処理できないと主張
- ✓ 行動プランナーと軌道プランナー共同でコスト関数の重みを学習
 - ※両者は別々に開発されるケースが多いが、両者不整合時の調整に手間がかかるのでこの点を改善した
- ✓ コスト関数の重みの学習には逆強化学習(Maximum Margin Planning法)と模倣学習を組合わせて使用
- ✓ 学習にディープラーニングを用いない為、コスト関数の解釈が可能



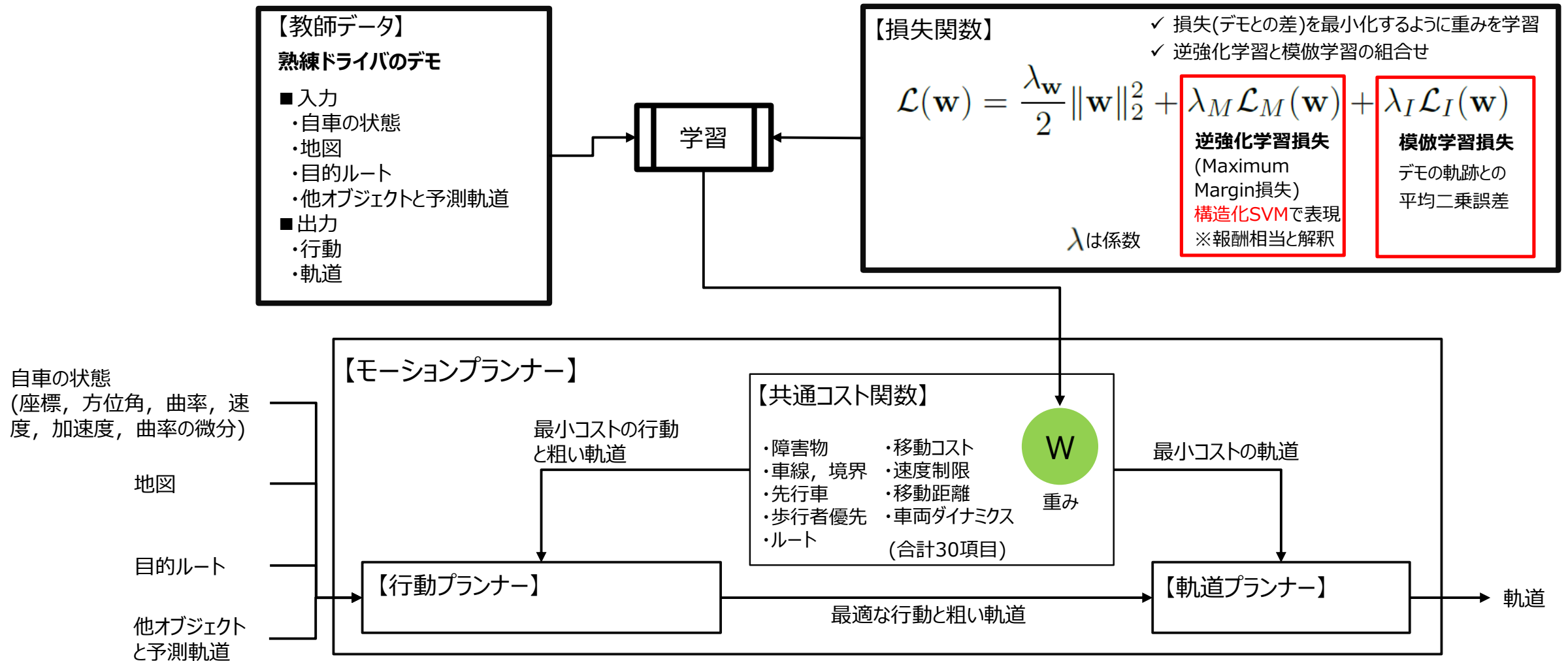
- ✓ <https://arxiv.org/abs/1910.04586>
- ✓ <https://martin.zinkevich.org/publications/maximummarginalplanning.pdf>
- ✓ 香月理絵, 荒井幸代, 大前学, 大日方五郎, 川崎敦史, 橋川雄樹, 小林祐一, 菅沼直樹, 田崎豪, 谷沢昭行, 新田修平, 野呂瀬琴, 馬場厚志, 藤吉弘巨, 目黒淳一, 森出茂樹, 谷口敦司, 山下倫央, 日本ロボット学会, 自動運転技術入門, オーム社, pp.10-111 (2021).

8. 学習を用いた事例; (2)Sadatらの方法 モーションプランナーの動作



<https://arxiv.org/abs/1910.04586>

8. 学習を用いた事例; (2)Sadatらの方法 モーションプランナーの学習



- ✓ <https://arxiv.org/abs/1910.04586>
- ✓ <https://martin.zinkevich.org/publications/maximummarginplanning.pdf>

8. 学習を用いた事例; (2)Sadatらの方法 実験結果 データセット

実験に利用したデータセットは以下の2つ.

■ ManualDrive

ドライバーがすべての交通規則を順守してスムーズかつ慎重に運転するように指示された, 実世界における人間の運転記録のセット.
トレーニング, 検証およびテスト セットには, それぞれ 12000, 1000, 1600のシナリオが含まれている.
このデータセットにはラベル付きオブジェクトは含まれていない.

■ TOR-4D

非常に挑戦的なシナリオで構成された実世界の運転データセット.

※上記はUber内部のデータセットで未公開と思われる

<https://arxiv.org/abs/1910.04586>

8. 学習を用いた事例; (2)Sadatらの方法 実験結果

未知の環境におけるシミュレーションの結果, 本手法の人間との高い類似性を確認

		人間ドライバの軌道との類似性 (1秒間 / 2秒間 / 3秒間 値が小さいほど類似性が高い)			快適性 (平均Jerkと横方向加速度)		速度違反	ルートの進捗	人と異なるシナリオを選択する割合
		L_2 (m)			Driving Metrics				
		1.0s	2.0s	3.0s	Jerk	Lat. accel.	Speed lim.	Progress	Diff. behavior (%)
人間ドライバ	HUMAN	-	-	-	0.237	0.168	0.14	3.58	-
比較用 ベースライン	Oracle ACC	0.700	1.560	3.670	-	-	0.34	3.64	-
	PT	0.254	0.941	2.190	0.147	0.232	0.26	3.81	8.5
本手法	Ours w	0.095	0.712	1.992	0.140	0.264	0.39	3.81	0.0
	Ours w_b	0.094	0.699	1.945	0.135	0.234	0.27	3.72	0.0
	Ours w_b^t	0.066	0.514	1.495	0.143	0.279	0.28	3.78	0.0

$\mathbf{W}$: 全てのシナリオに対して単一の重みベクトルを学習 (重みの数: 30 = コスト関数の項目数)

$\mathbf{W}_b$: 動作(車線維持, 左車線変更, 右車線変更)ごとに個別の重みベクトルを学習 (重みの数: 90 = コスト関数の項目数 $\times$ 3)

$\mathbf{w}_b^t$: 動作ごとおよびタイムステップごとに個別の重みベクトルを学習 (重みの数: 1890 = コスト関数の項目数 $\times$ 3 $\times$ 21ステップ)

Oracle ACC : 正解値の運転経路を使用

PT : 行動プランナーの単純なバージョン (縦方向の時空間のみで推論)

<https://arxiv.org/abs/1910.04586>

8. 学習を用いた事例；(3)VOLVO 能動学習

- ✓ VOLVOによる運転軌跡の時系列データにアノテーションを付加するためにALフレームワークを開発する研究
- ✓ 未知の運転シナリオの軌跡の発見も目的
- ✓ VOLVO提供データセット使用（VOLVOが公開しているCirrus datasetと同一かは不明）
- ✓ エゴカーとターゲットカーの以下の3つの軌道を例として前項のデータから取得して実施
 - ✓ 自車がターゲットカーの右側を走行
 - ✓ 自車がターゲットカーの左側を走行
 - ✓ カットイン
- ✓ 結論としては、アノテーション作業や未知のシナリオの軌跡の検出にALは有効

<https://doi.org/10.1016/j.engappai.2022.104972>

8. 学習を用いた事例：自動運転用データセット

名称	説明	URL等
KITTI	ドイツの中規模都市であるカールスルーエ周辺～高速道路での運転から得られた画像が提供されている。画像は、最大15台の車と30人の歩行者が映っている。	https://www.cvlibs.net/datasets/kitti/
Cityscapes	都市部の自動車前方映像を用いて、セマンティックセグメンテーションやインスタンスセグメンテーションモデルを学習するために作られた、交通向けシーン画像のデータセット。	https://www.cityscapes-dataset.com/dataset-overview/
Apollo Scape	Baiduが公開した自動運転車用のデータセット。RGBの映像とそれに対するセグメンテーション、また3Dのレベルでも映像・セグメンテーションが提供されている。画像数は20万でセグメンテーションクラス数は車や自転車、人など25種類。	https://apolloscape.auto/scene.html
Mapillary Vistas Dataset	自動運転のための、路上画像のデータセット。25,000のアノテーション済みの高解像度データを提供。現在も増え続けており、しかも世界各国の画像が含まれている(日本の道路の画像もある)。2018/7/30に、道路標識などについてより細かく定義した30クラスが追加された(ただ「標識」ではなくスピード制限なのか方向指示なのかなど)。また、アノテーションの精緻化も行われている。	https://www.mapillary.com/dataset/vistas
DeepDrive	Berkleyの公開した自動運転車のためのデータセット。様々な天候や都市、時間帯を収録した10万のHD動画(計1100時間)から構成される。画像フレーム10万についてはバウンディングボックス(人、バスなど計10クラス)、運転可能領域、白線(走行可能なレーンや横断歩道)がアノテーションされている。1万についてはピクセルレベルのセグメンテーションがアノテーションされている。	https://bdd-data.berkeley.edu/
Lyft Level 5 AV Dataset 2019	Ford FusionにLiDAR(3台)/カメラ(7台)を搭載し記録したデータを基にしており、レーンや横断歩道、信号、駐車位置など細かなセグメントがアノテーションされている。	https://level-5.global/data/
Waymo Open Dataset	Waymo が公開した自動運転のためのデータセット。LiDAR(5台)/カメラ(5台)を使い様々な天候や都市、時間帯を収録した20秒の動画(10Hz)に、合計120万個の2Dラベルと1200万個の3Dラベルがアノテーションされている。Google Colabatory で実行できるチュートリアル が用意されている。	https://waymo.com/open/
Pandaset	LiDARセンサーメーカーのHesaiとアノテーションプラットフォームのScaleAIが共同で構築した自動運転車のためのデータセット。都市部での運転を意識しており、急勾配の坂や歩道、様々な時間帯で収録を行っている(場所はシリコンバレー)。	https://scale.com/open-av-datasets/pandaset#overview
DriveSeg	トヨタの米国法人Toyota Motor Sales USAとマサチューセッツ工科大学(MIT) が2020年6月リリース	https://agelab.mit.edu/methods/mit-driveseg-dataset-for-dynamic-driving-scene-segmentation
Audi Autonomous Driving Dataset(A2D2)	Audiが2020年4月に公開した。時間と労力を必要とするデータセット作製に関わる参入障壁を取り除き、研究開発を促進する狙い。	https://www.a2d2.audi/a2d2/en.html
Cirrus dataset	VOLVOのデータセット。長距離に重点を置いた LiDAR スキャン パターンの不均一な分布を提供。対応するカメラ画像、均一なスキャン パターン、およびアノテーションも含まれる。	https://developer.volvocars.com/resources/cirrus/

※ Sadatらの事例のManualDriveとTOR-4DはUber内部の非公開データセットと思われる

※上記の中にActive Learningでアノテーション対象が選別されているような説明は見受けられない。VOLVOのCirrus datasetは、VOLVOでActive Learningによる軌道生成の論文が存在するので、Active Learningが使用されているかもしれない

- ✓ <https://github.com/arXivTimes/arXivTimes/tree/master/datasets>
- ✓ <https://cvml-expertguide.net/terms/dataset/image-dataset/kitti/>
- ✓ <https://cvml-expertguide.net/terms/dataset/image-dataset/cityscapes/>
- ✓ https://jldouten-lab.com/u_dataset-27

9. パスプランニング周辺技術; (1)自己位置推定

分類	手法	説明	課題	備考
高精度地図 利用なし	白線認識	カメラ, LiDAR等による白線認識で車線内の位置を推定する。 GNSS, IMU, 簡易地図などと組み合わせて適用することが多い。	縦方向の位置が不明であるため, 合流等への対応が困難	
高精度地図 利用	点群地図	LiDARで計測した点群と, MMS(Mobile Mapping System)等であらかじめ作成した点群 地図とを照合して自己位置を推定する。代表的なマッチング方式として以下のものがある ・ICP (Interactive Closed Point) ・NDT(Normal Distribution Transform) 高速 点群地図と, カメラ画像による特徴点との照合する手法もある。	・高精度な初期位置が必要 ・累積誤差の蓄積 ・特徴が乏しい場所では照合困難	
	反射強度マップ	LiDARの反射強度マップによる照合。 LiDARの反射強度を上空から見た画像に変換した反射強度マップをあらかじめ作成し, 走行 時に照合する。反射点の信頼性の高いところを利用する方法		
	ベクターマップ	白線や物標の位置が示されたベクターマップを使用し, 走行時にカメラ画像やLiDARから得ら れた物標と照合する。	・ベクターマップの整備	
GNSS	GNSS	精度数m	・トンネル内 ・ビルによる反射 (マルチパス) ・視認衛星数不足 ・受信周期	
	GNSS + みちびき	精度1m以下 (RTK補正情報利用時)		
	RTK-GNSS	精度数cm 地上に基地局が必要 (ジェノバ等の配信サービスあり) ・GNSS+RTK一体型装置 ・オープンソース RTKLIB		✓ https://www.jenoba.jp/ ✓ https://www.rtklib.com/
	センサ統合による補正 (IMU, 車速)	ベイズフィルタ, カルマンフィルタ, パーティクルフィルタ, 拡張カルマンフィルタ(EKF)などを使用し てGNSSをIMUや車速で補正 ・Autoware (拡張カルマンフィルタ) ・GNSS+IMU+ホイールオドメトリ一体型装置 ・Eagleeye (GNSS+IMU+ホイールオドメトリ) (安価なGNSSとIMUを組合わせて一体型装置同様の機能をソフトウェアで実現する。ROS ベースのオープンソース)		✓ https://gitlab.com/autowarefoundation/autoware.ai/core_perception/tree/master/ekf_localizer ✓ https://github.com/MapIV/eagleeye ✓ https://www.jstage.jst.go.jp/article/sicetr/55/11/55_754/_pdf

- ✓ 香月理絵, 荒井幸代, 大前学, 大日方五郎, 川崎敦史, 橘川雄樹, 小林祐一, 菅沼直樹, 田崎豪, 谷沢昭行, 新田修平, 野呂瀬琴, 馬場厚志, 藤吉弘亘, 目黒淳一, 森出茂樹, 谷口敦司, 山下倫央, 日本ロボット学会, 自動運転技術入門, オーム社, pp.139-153 (2021).
✓ 二宮芳樹, 武田一哉, 自動運転, コロナ社, pp.143-153 (2021).

9. パスプランニング周辺技術; (1)自己位置推定 DUCの事例

■ カーネギーメロン大学 Boss

以下を組合せて実現

- 高精度地図（提供されるベクターマップ相当の高精度地図）
- GNSS + 高性能IMU + ホイールオドメトリ一体型装置（Applanix POS LV 420）
Omnistar仮想基地局サービスにより精度向上
位置誤差10cm程度になるが、信号が途切れやすく信頼性不足
- SICK LMS 291-S05/S14 LiDAR

POS LVのみでは信頼性不足のため、SICK LiDARにより、反射強度から車線を検出し、レーン検出と正確な車両位置特定を行っている

※高精度地図が提供されるDUCでは使用していないが、高精度地図なしで未知の道路を走行できる「道路計上推定システム」も持っている

LiDARとカメラの情報を融合して、テストではGNSS信号なしで5.7km車線を維持できた



■ スタンフォード大学 Junior

Bossとほぼ同様の構成

- ✓ https://www.cmu.edu/traffic21/pdfs/urmson_christopher_2008_1.pdf
- ✓ <http://writingaboutcars.com/360-sensors-give-autonomous-cars-vision/>
- ✓ <http://robots.stanford.edu/papers/junior08.pdf>

9. パスプランニング周辺技術；(2)行動予測

分類	手法	説明	備考
歩行者の経路予測	ベイズモデル	Context-based pedestrian path prediction 車載カメラで撮影された歩行者のコンテキストを考慮して経路を予測するもの。 動的ベイジアンネットワーク(Dynamic Bayesian Network; DBN)に、SLDS(Switching Linear Dynamic System)と呼ばれる歩行者の動きを判断するモードを加えたものを用いる。SLDSでは、歩行者の座標、頭の向き、車両までの距離、縁石までの距離を観測データとして用いる。これら複数の観測データを用いて内部の状態を逐次更新することで高精度な予測を実現。	
	深層学習モデル	深層学習を用いて歩行者の行動をモデル化する。 過去の観測データから未来の各時刻における予測結果を逐次計算する必要があるため、時系列データを取扱えるRNN(Recurrent Neural Network)やLSTM(Long Short-Term Memory)が用いられる。 その他歩行者間のインタラクションを考慮した手法として、Social-LSTMやSocial GANがある	
他車両の経路予測	カルマンフィルタ	車両の動きを等加速度運動でモデル化し、入力ノイズに頑健な予測を行うことを実現	
	ガウス回帰課程	交差点等のある固定シーンにおける車両の動きを学習し、車両の加減速を考慮した予測を実現	
	深層学習モデル	Social-LSTMやDESIRE等 DESIREは複数の予測候補を生成できる	

✓ 香月理絵, 荒井幸代, 大前学, 大日方五郎, 川崎敦史, 橘川雄樹, 小林祐一, 菅沼直樹, 田崎豪, 谷沢昭行, 新田修平, 野呂瀬琴, 馬場厚志, 藤吉弘亘, 目黒淳一, 森出茂樹, 谷口敦司, 山下倫央, 日本ロボット学会, 自動運転技術入門, オーム社, pp.139-153 (2021).

9. パスプランニング周辺技術; (2)行動予測 FiveAIの事例

- ✓ 2015年に英・ブリストルで設立された自動運転スタートアップ企業
 - AIを活用した自動運転システムプラットフォームの開発
 - 自動運転システムの安全性を測定・保証する仕組みについても研究
- ✓ 2019年ロンドンで公道で自動運転車試験を実施
- ✓ 2022年4月 BOSCHに買収

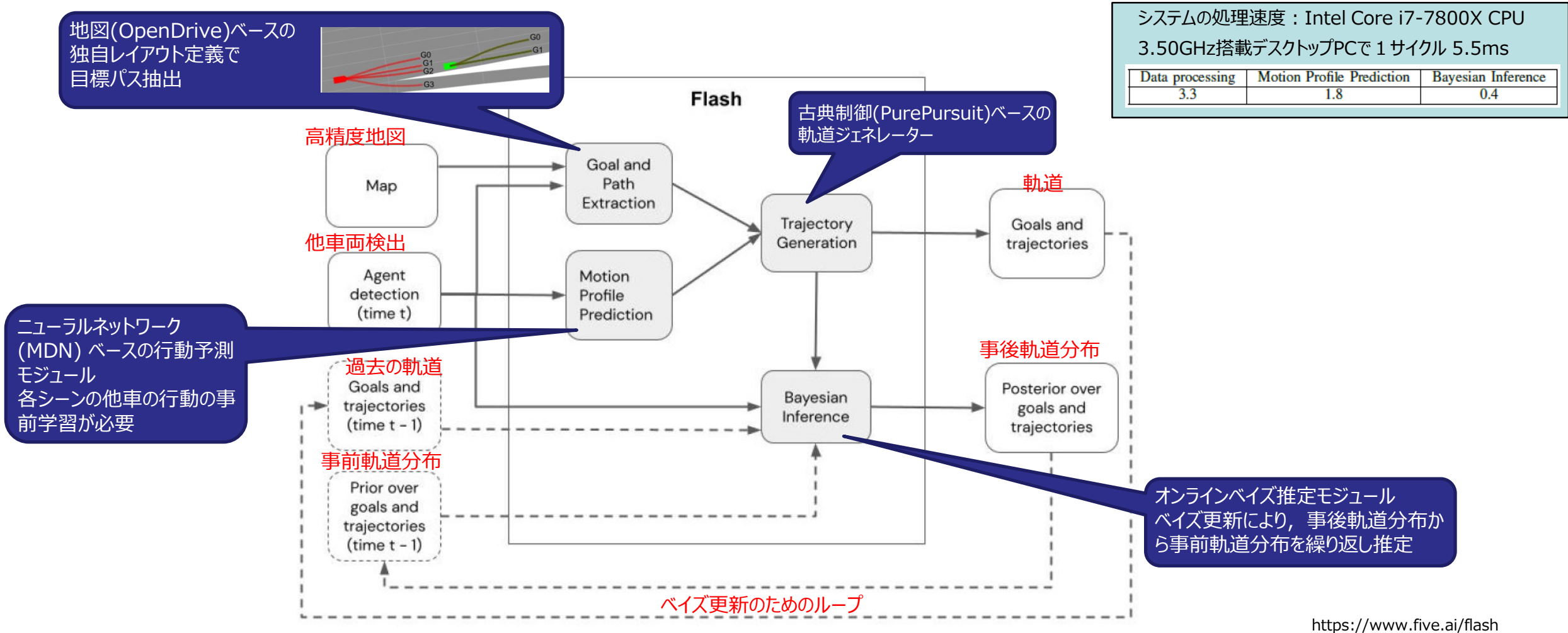


FiveAIの車両（搭載センサ：カメラ, LiDAR, Radar, GNSS等）

- ✓ <https://www.five.ai/>
- ✓ <https://www.autonomousvehicleinternational.com/opinion/what-the-uks-most-advanced-av-trials-teach-us-about-building-public-trust.html>
- ✓ <https://fresearchlondon.com/trend/fiveai/>
- ✓ https://jidounten-lab.com/u_startup-45-matome#FiveAI

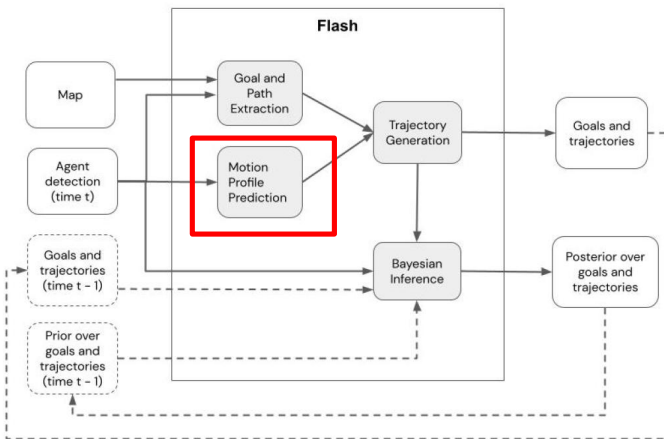
9. パスプランニング周辺技術；(2)行動予測 FiveAIの事例「Flash」

ベイズ推定と、学習したモーションプロファイルを使用した、自動運転のための高速で軽量な行動予測フレームワーク「Flash」



9. パスプランニング周辺技術；(2)行動予測 FiveAIの事例「Flash」 行動予測モジュール

行動予測には、ニューラルネットワークMDNによる事前学習を採用



- ✓ ニューラルネットワークによるアプローチとして、MDN (Mixture density networks)を採用
- ✓ MDNは不確実性をモデル化し、単一の出力ではなく分散を含む混合パラメーター(確率分布)を予測出力できる
- ✓ MDNは多変量ガウス関数を使用した多変量ガウス混合モデルの形式で学習する
- ✓ 交通環境や実行中の運転行動(車線追従と車線変更)および他車両の数毎にデータを分類してトレーニングしたMDNをシーンに応じて使い分けている。(4つのレーンフォローモデルと、16のレーンチェンジモデルで構成)
- ✓ MDNの特徴量はシーンに応じて変更している(他車両の相対速度/加速度, 相対距離, 車種(乗用車, トラック, バイク)等)
- ✓ 予測範囲は半径60m. 予測期間は5秒

MDN (Mixture density networks)

論文 : C. M. Bishop, "Mixture density networks,"

1994.https://publications.aston.ac.uk/id/eprint/373/1/NCRG_94_004.pdf

MDNを歩行者の行動予測に利用した他の事例

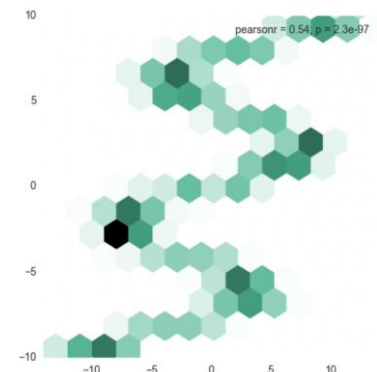
<http://mprg.jp/data/FLABResearchArchive/Bachelor/B17/Abstract/iesaki.pdf>

確率的DeepLearningライブラリ「Edward」に実装がある

<http://edwardlib.org/tutorials/mixture-density-network>

※Edwardはベイズ統計と機械学習、深層学習、確率的プログラミングを融合させたPython&TensorFlowライブラリ

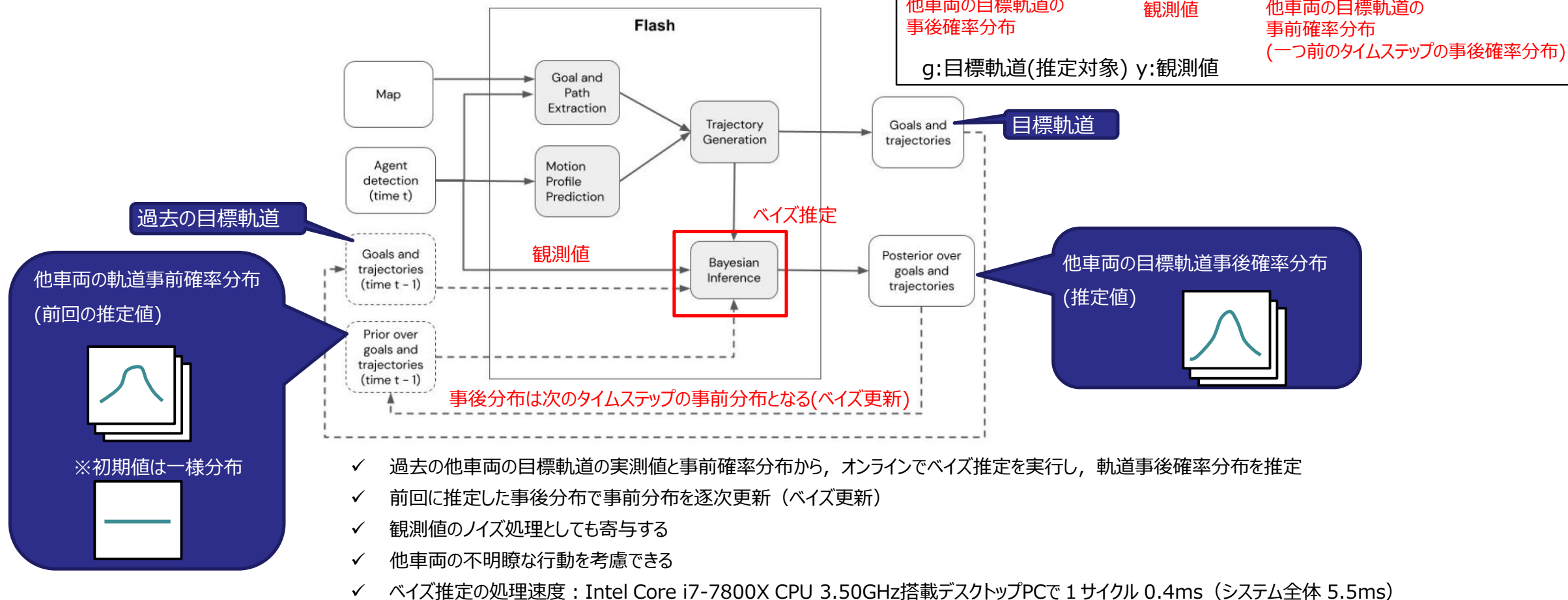
MDNの出力イメージ
(確率分布が出力される)



<https://www.five.ai/flash>

9. パスプランニング周辺技術; (2)行動予測 FiveAIの事例「Flash」 ベイズ推定モジュール

オンラインベイズ推定により, 他車両の軌道の不確実性を考慮



<https://www.five.ai/flash>

9. パスプランニング周辺技術; (2)行動予測 ベイズ推定

Xが起こったときにその原因がAであった確率を推定したい. 観測結果から原因の確率(分布)を推定する

ベイズの定理

A : 原因

X : Aが原因で起こったとされる事象 (観測結果)

Aが起きたときにXが起こった
条件付き確率(分布)
(尤度; 観測結果)

Aが起きる確率(分布)
(事前確率)

$$P(A|X) = \frac{P(X|A) P(A)}{P(X)}$$

Xが起きたときにAが起こる
条件付き確率(分布)
(事後確率; 推定対象)

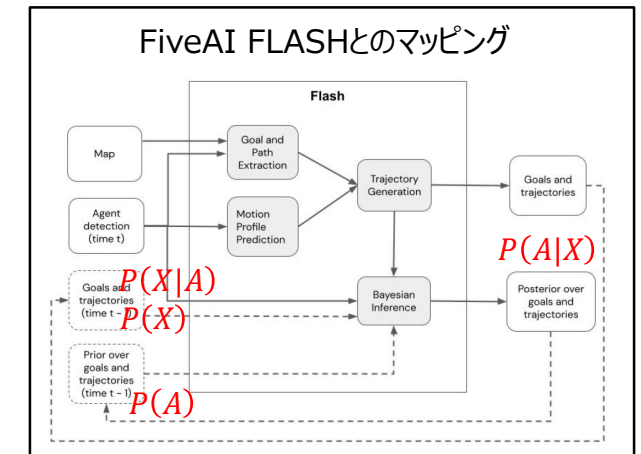
Xが起こる確率分布
(規格化定数; 観測結果)

- ✓ 新しいデータが観測されるたびに事後確率 $P(A|X)$ で事前確率 $P(A)$ を更新することをベイズ更新とよぶ
- ✓ 最初は事前確率 $P(A)$ は不明なため, 無情報分布として, 一様分布が使用されることが多い
- ✓ ベイズ更新の試行が進んでいくと初期分布の影響は小さくなる

ベイズ更新ループ

注: 各試行は独立でなければならない (過去の結果に影響されない)

<https://okigaru-data-science.com/bayesian-inference/>



9. パスプランニング周辺技術；(2)行動予測 May Mobilityの事例

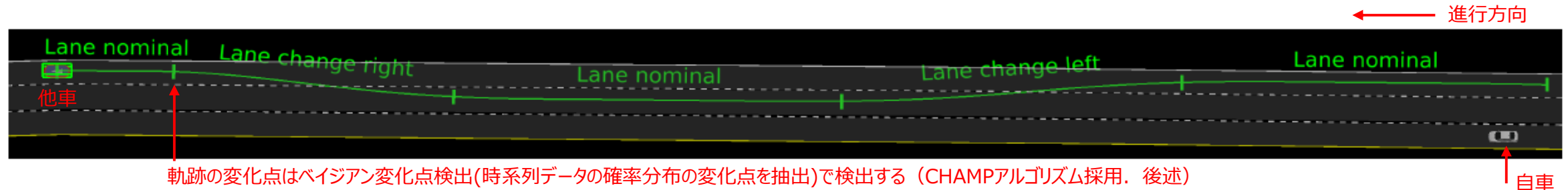
- ✓ ミシガン大学の自動運転開発チームのメンバーらによって、2017年に設立されたスタートアップ企業
- ✓ 自動運転技術の開発や自動運転移動サービスの提供に取り組んでおり、アメリカ国内で運行実績があるほか、日本の広島県でも実証実験を行っている
- ✓ 戦略的パートナー：トヨタ、MONET、東京海上日動、豊田通商



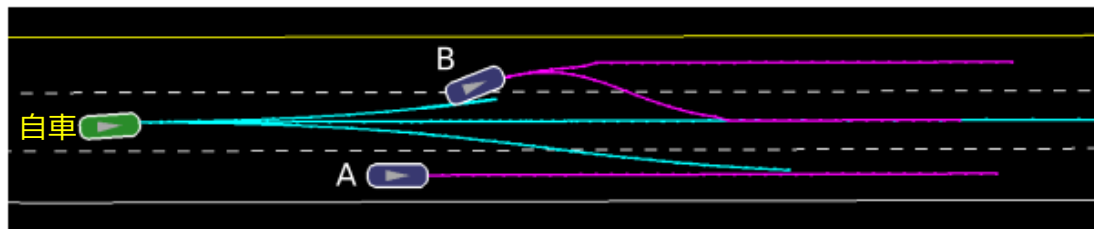
- ✓ <https://maymobility.com/>
- ✓ <https://maymobility.co.jp/>
- ✓ https://jidouten-lab.com/u_33691

9. パスプランニング周辺技術; (2)行動予測 May Mobilityの事例「Multi-Policy Decision Making」

Multi-Policy Decision Making(MPDM); あらかじめ運転行動(ポリシー)を想定し, 他車が実行している可能性のあるポリシーを確率分布で推定する



- ✓ 軌跡を変化点で区切り, 変換点に挟まれた区間毎の確率分布と各ポリシーの確率分布(事前に準備)を比較することで, 他車の実行中のポリシーを推定する. どのポリシーにも当てはまらない場合は, 異常な運転行動として扱う
- ✓ 論文執筆時のポリシー (今後追加)
 - lane-nominal : 車線維持
 - lane-change-right/lane-change-left : 右車線変更/左車線変更
 - turnright/turn-left/go-straight/yield : 交差点 右折/左折/直進/譲歩
- ✓ 自車のポリシーと他車の各ポリシーとのフォワードシミュレーションを行い, 報酬関数で評価して最終的に自車のポリシーを決定する
報酬関数のパラメータ: 目標までの距離, 障害物までの最小距離, 車線の優先順位, 乗り心地(最大ヨーレートと前後ジャーク)
- ✓ 処理時間: 平均処理時間は 84 ms, 最悪の処理時間は 106 ms



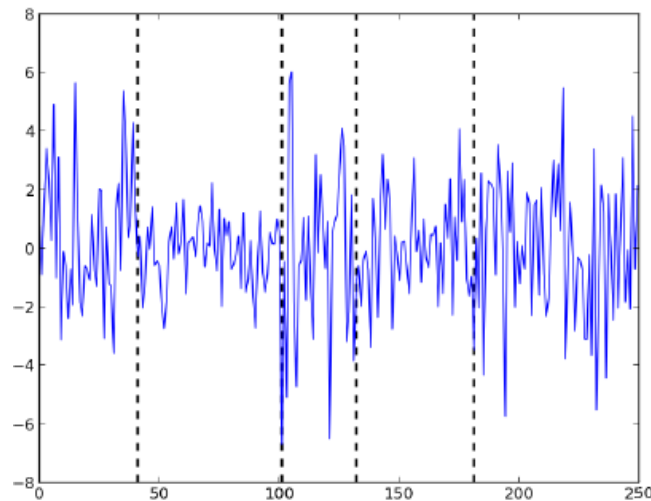
✓ <http://www.roboticsproceedings.org/rss11/p43.pdf>

9. パスプランニング周辺技術; (2)行動予測 May Mobilityの事例「CHAMPアルゴリズム」

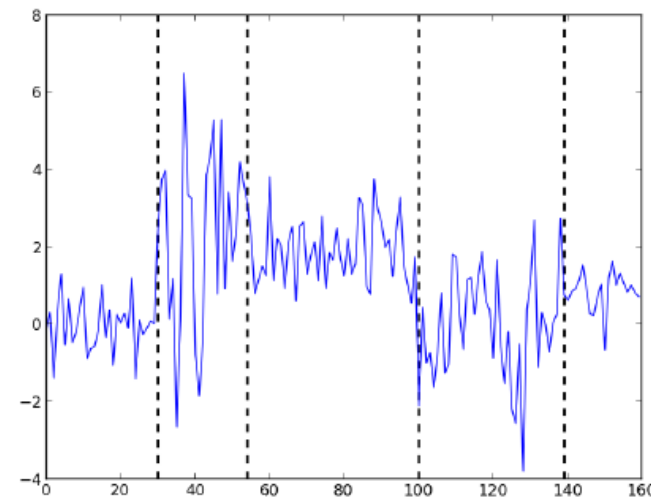
- ✓ CHAMPはオンラインベイジアン変化点検出のためのアルゴリズム
- ✓ ベイジアン変化点検出が大幅に簡素化され, より多くのタイプのモデルで使用できるようになり, パフォーマンスが向上する
- ✓ ROSの実装あり: <http://wiki.ros.org/changeoint>

CHAMP セグメント化例

分散が変化する平均ゼロのガウスデータ5セグメント



平均と分散の両方が変化するガウスデータ5セグメント



<https://people.cs.umass.edu/~sniekum/pubs/CHAMP.pdf>

9. パスプランニング周辺技術；(2)行動予測 FiveAIとMay Mobilityの比較

観点	FiveAI Flash		May Mobilitu MPDN	
	説明	評価	説明	評価
行動予測方法	ニューラルネットワーク(MDN)による事前学習で推定	○	あらかじめ想定した想定行動(ポリシー)の確率分布と、他車の軌跡の確率分布を比較して他車の実行中のポリシーを推定（統計処理）	○
事前準備	シーンおよび他車両の数毎にMDNをトレーニングする必要がある	△	シーン毎に各ポリシーの確率分布を事前に準備する必要がある	△
不確実性への対応 (ロバスト性)	オンラインベイズ推定によって他車の行動の確率分布をオンラインで予測することで対応する	○	他車の軌跡の確率分布の類似性でポリシーを推定しているため、ある程度のロバスト性は確保されていると思われる	○
他車の異常行動への対応	同上	○	どのポリシーにも該当しない車両を異常車両として分類し、異常車両の現在の状態のみに基づき対処	△
処理速度	Intel Core i7-7800X CPU 3.50GHz搭載デスクトップPCで 1 サイクル 5.5ms	○	平均処理時間は 84 ms 最悪の処理時間は 106 ms	△

※他車の異常行動への対応は、オンラインでベイズ推定を行っているFiveAIに利がありそうだが、MPDNにFiveAIのオンラインベイズ推定を組み合わせることも可能と思われる

10. まとめ (1) パスプランニング関連技術一覧 (サーベイ論文より)

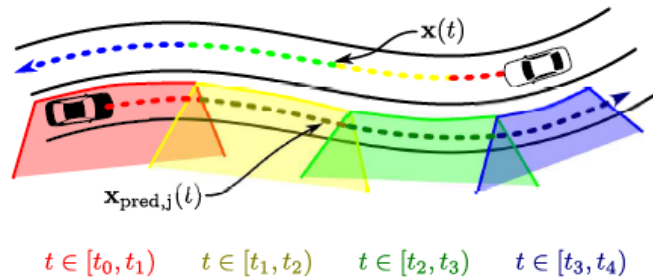
分類	手法	長所	短所	備考
グラフ探索	ダイクストラ	グローバルプランニングに適す	膨大な数のノードがあるとアルゴリズムが遅くなる 結果のパスは連続的ではない リアルタイム制御には適さない	4章参照
	A*	ダイクストラに基づくがより高速	結果のパスは連続的ではない 探索を高速化するルールの定義が容易でないことが多い	4章参照
	State Lattice	位置, 速度, 加速度, 時間といった複数の次元を処理できる ローカルプランニングや動的環境に適す	状態格子(State Lattice)の全ての可能性を評価する為計算コスト大	DUCのCMUではMPTGと組み合わせて使用(6章)
サンプリングベース	RRT	多次元システムで迅速な解が得られる グローバルプランニングおよびローカルプランニングに適す	結果として得られる軌道は連続的でなくぎざぎざしている パスの最適性は計算時間に強く依存する	6章参照
曲線補間	線と円	低計算コスト 実装が容易 自動車のような車両では最短経路が保証される	パスは連続的でなくぎざぎざしているため, パスセグメント間の遷移が快適ではない プランナーはグローバルウェイポイントに依存する	
	クロソイド曲線	道路の設計と同じ曲線を使用できる ローカル計画に適す	曲線を定義する積分に時間がかかる 曲率は連続的だが, 滑らかではない プランナーはグローバルウェイポイントに依存する	
	多項式曲線	低計算コスト 曲線の連続連結が可能であり, 快適性に優れる	通常4次以上となり, 運動状態を達成するための係数の計算が困難 プランナーはグローバルウェイポイントに依存する	6章参照(DUC CMUの事例)
	ベジエ曲線	低計算コスト 曲線を定義する制御点のため, 曲線を直感的に操作できる 曲線の連続連結が可能であり, 快適性に優れる	曲線の次数を上げると順応性が失われ計算時間が長くなる (より多くの制御点を評価して正しく配置する必要がある) プランナーはグローバルウェイポイントに依存する	
	スプライン曲線	低計算コスト 一般的で連続的な曲率パスが得られる	パート間の連続性に重きを置いているため, 道路の適合性と曲率最小化の観点から最適解ではない可能性がある プランナーはグローバルウェイポイントに依存する	6章参照(KAITの事例)
数値最適化	関数最適化	道路と自車両の制約, および他の道路利用者を考慮することができる	計算時間大 プランナーはグローバルウェイポイントに依存する	MPTGやMPCも関数最適化の一種(6章)

✓ https://www.researchgate.net/publication/284766879_A_Review_of_Motion_Planning_Techniques_for_Automated_Vehicles

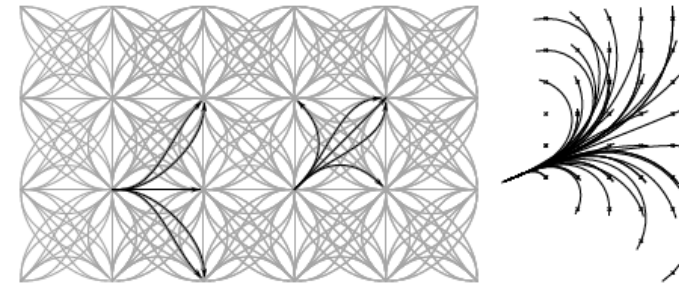
10. まとめ (1) パスプランニング関連技術図解 (サーベイ論文より)



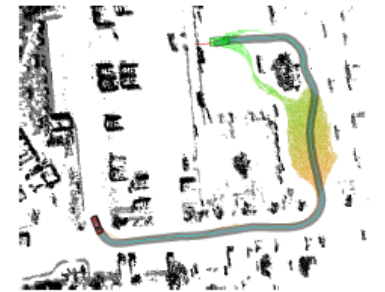
ダイクストラ



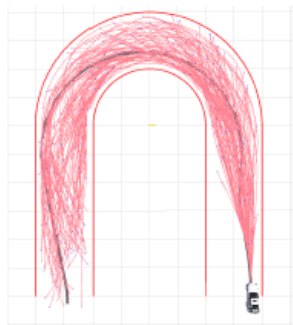
関数最適化



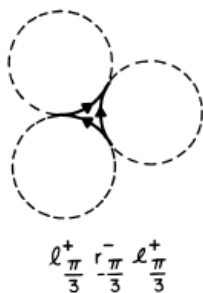
State Lattice



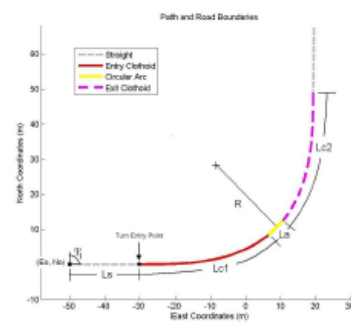
A*



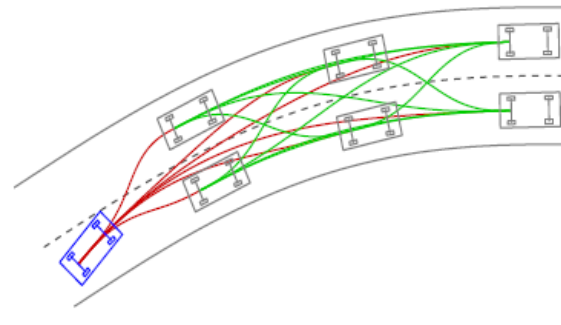
RRT



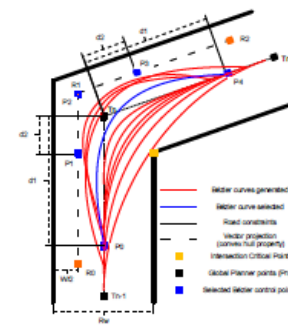
線と円



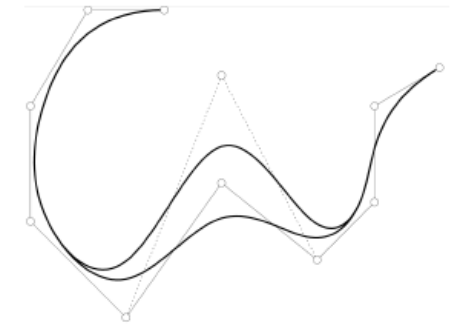
クロソイド



多項式



ベジェ曲線

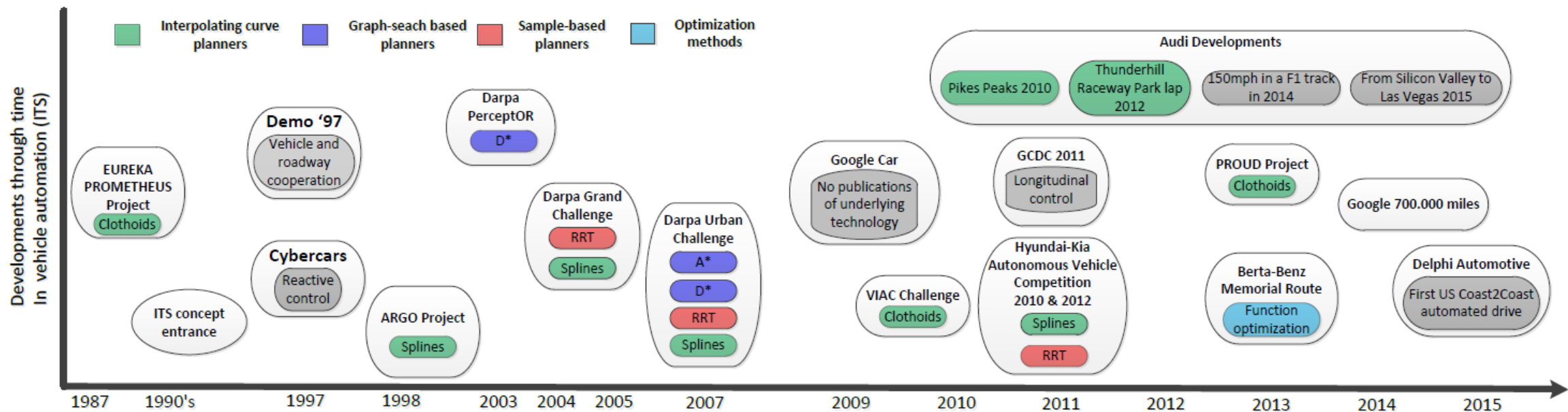


スプライン

✓ https://www.researchgate.net/publication/284766879_A_Review_of_Motion_Planning_Techniques_for_Automated_Vehicles

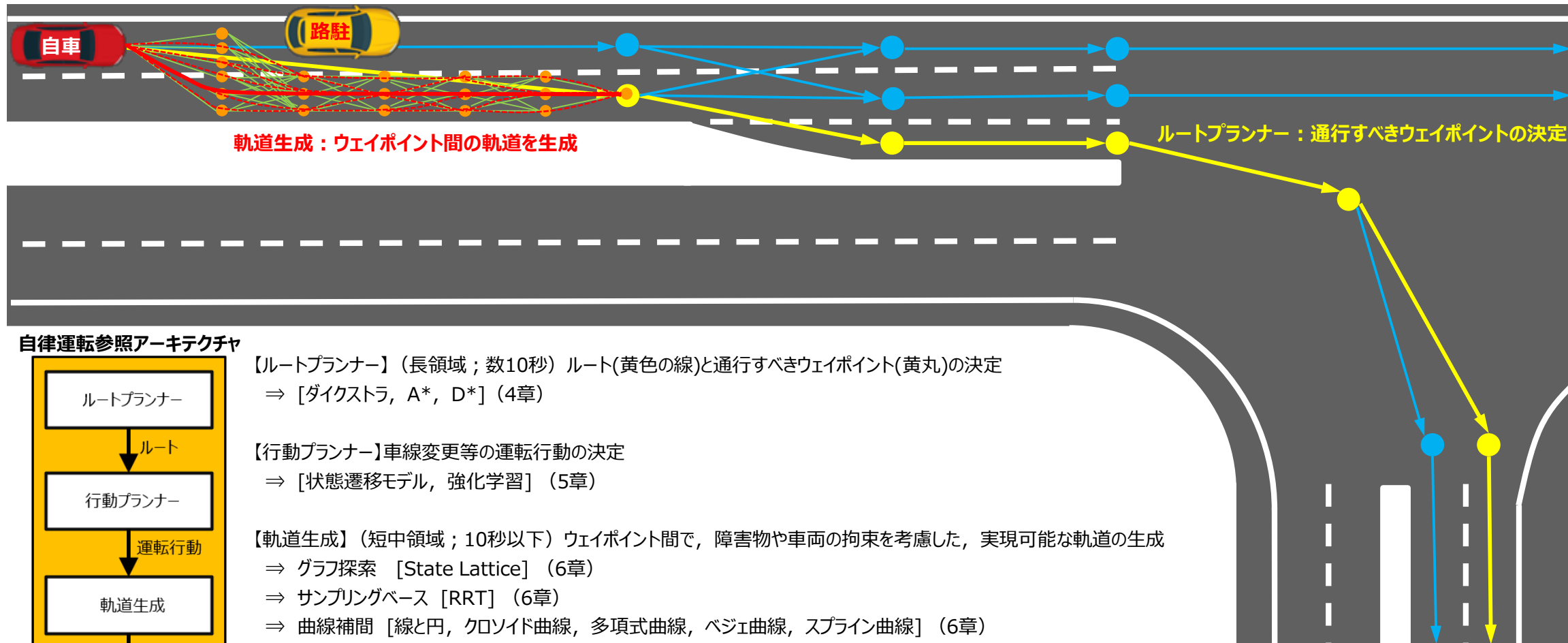
10. まとめ (1) パスプランニング関連技術図解 (サーベイ論文より)

時系列からみた各技術の採用事例



✓ https://www.researchgate.net/publication/284766879_A_Review_of_Motion_Planning_Techniques_for_Automated_Vehicles

10. まとめ (2) 短中長領域のパスと関連技術の関係



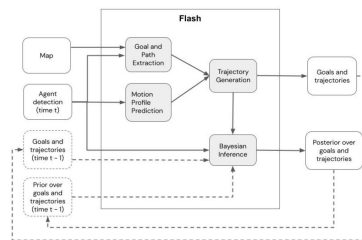
10. まとめ (3) 課題

- ✓ 実用的なセンサ構成
- ✓ 状態遷移による行動計画の限界 ⇒ 学習による方法は発展途上
- ✓ End to Endな軌道生成（知覚から直接軌道生成）⇒ 学習による方法はまだ研究段階．単純な模倣では難しく強化学習の仕組みが必要（行動の意図を理解して模倣）
- ✓ 滑らかな軌道生成 ⇒ 多項式, MPC（処理速度が課題）
- ✓ 緊急時の対応（急ブレーキ／急回避）
- ✓ リスク予測（他車両や歩行者の行動の不確実性や死角等の潜在リスクの統合）⇒ ベイズモデルの導入 等
- ✓ ロバスト性 ⇒ ベイズ推定等確率分布で取り扱うことで対応 等

10. まとめ (4) 調査結果を踏まえた参考構成例

これまでの調査結果を踏まえた参考構成例を次頁に示す。参考構成例は以下の考えに基づき設定した。

- 冒頭で定義した自律運転参照アーキテクチャ（ルートプランナー／行動プランナー／軌道生成）をベースとした
- 自転車回避という先進的なシナリオに対応したアルゴリズムを実現しているKAITの研究事例を取り入れるものとした。本年度における調査結果のうち、以下の要素も含まれており、参考構成として適切と判断した。
 - SD座標系（フレネ-セレ標構）
 - スプライン補間
 - 車両運動モデル（MPC）
 - リスクポテンシャル
- Five-AI「Flash」のベイズ推定の考え方を取り入れた



<https://www.five.ai/flash>

10. まとめ (4) 調査結果を踏まえた参考構成例

