Vol.2021-IFAT-142 No.11
Vol.2021-DC-120 No.11
2021/3/26

複数車種の運行設計領域を管理する管制センタアーキテクチャの検討

近藤 明宏[†1]　松本 貴士[†1]

概要：自動運転車両を利用するサービスの普及が見込まれている．自動運転車両を用いたサービスでは，運行管理・走行経路指示などを行う管制システムとの連携が必要である．自動運転車両は，予め設計された運行設計領域（Operational Design Domain: ODD）外では走行ができないという制約がある．管制システムでは，管理対象の個別車両の ODD に基づき走行経路指示を行う必要があるが，管理車両数や車両種類の増加に伴い，走行可能な道路を算出する処理負荷が増え，応答遅延への影響が大きくなるという課題がある．本研究では，このような課題に対応するため，車両ごとの走行可能道路算出を並列処理する管制センタアーキテクチャを提案する．

キーワード：自動運転車，運行設計領域，ODD，管制センタ

Examination of Vehicle Control Center Architecture to Manage Multiple Vehicle's Operational Design Domain

AKIHIRO KONDO[†1]　TAKASHI MATSUMOTO[†1]

Abstract: Services which use autonomous vehicle are expected to be widespread. To realize the services, autonomous vehicles need to connect to vehicle control center which manages vehicles and instructs routes. Autonomous vehicles are limited to operate within Operational Design Domain (ODD). The vehicle control center needs to instruct routes to vehicles based on individual vehicle's ODD, but there is a problem that processing time become longer because of increasing control vehicles the center need to handle. In this research, to solve this problem, we propose the architecture which parallelly processes each vehicle's passable area.

Keywords: autonomous vehicle, Operational Design Domain, ODD, vehicle Control Center

1. はじめに

近年，自動運転車両や自動運転車両を用いたサービスの研究開発が盛んになされている [1]．自動運転車両活用サービスは，公道への適用に加え，工場内敷地や港湾内敷地，鉱山などの限定領域と呼ばれる場所における自動運転車活用サービスに期待が高まっている．限定領域での自動運転車活用サービスとしては，工場における部品の倉庫から組み立て工場までの部品の運搬や，港湾におけるコンテナの移動，鉱山における鉱石や土砂の搬送・放土作業 [2]などの自動化サービスが考えられている．限定領域では，例えば人の進入を制限するなど，サービス提供者により環境を制御できるため，公道での自動運転車サービスと比較して早期の実用化が見込まれる [3].

自動運転サービスを実現するにあたり，安全性の確保と業務効率の向上が重要な課題となっている [4]. 例えば，安全性を重視して保守的なルールでサービスの運用を行う場合，低速で運行されることや頻繁に車両が停止する，幅員の広い幹線道路のみを走行する，など，車両運用の効率が低下する懸念がある．このような，自動運転車単独では困難な安全性・効率性に関する課題を解決するために，交差点に進入する車両同士が車車間通信を活用して通信をして優先度を決定する技術 [5]や，車両死角を補うために路側にセンサを設置して，当該センサから取得したデータを用いて管制センタが自動運転車両に情報を提供するインフラ協調型自動運転システムが検討されている [4].

本研究では，限定領域において自動運転を利用したサービスを実現するにあたり，効率良い車両の運用を可能にするインフラ協調型システムのアーキテクチャを提案し，当該システムの管制センタを試作してその評価を行う．

2. 背景

2.1 自動運転と運行設計領域

自動運転車両は，車両に搭載したセンサにより周辺の環境を認識し，認知した情報に基づいて行動を計画し，計画に従って車両を制御するという，「認知」「判断」「操作」の 3 つの動作の繰り返しにより自律的に移動する．車両を制御するには，これら 3 つの処理が複雑に影響するため，環境が変化した際に，動作がどう変化するのかを把握しきることは難しく安全性が保証できない．そこで，自動運転車両は，運行設計領域（Operational Design Domain: ODD）と

†1　株式会社 日立製作所
Hitachi Ltd.

Vol.2021-IFAT-142 No.11
Vol.2021-DC-120 No.11
2021/3/26

呼ばれる領域内において正常な動作が保証される条件が定義され，当該領域外では正常な動作が保証されていない．ODD の定義の例としては，道路条件，地理条件，環境条件，その他条件などがある [6] [7]．道路条件としては，高速道路，一般道等の道路種別などの条件がある．地理条件としては，都市部，山間部などの道路以外の周辺を含めた条件がある．環境条件では，天候，日照状況などセンサ等に影響を及ぼす項目がある．その他条件としては，速度の制限やインフラ協調の有無，連続稼働時間などがある．

例えば，ある自動運転車両は高速道路でのみ動作可能や，別の自動運転車両では 20km 以下での特定領域のみ動作可能などがありえる．

この ODD は，車種により異なるものとなる．これは，搭載されるセンサの種類や性能に相違があるためである．定義次第によっては，同一車種でも車両ごとに異なる場合もある．例えば，一部のセンサが失陥するなどした場合，当該車両の ODD はその内容によって限定される可能性がある．

2.2 インフラ協調型自動運転システム

インフラ協調型自動運転システムの構成を Fig. 1 に示す．

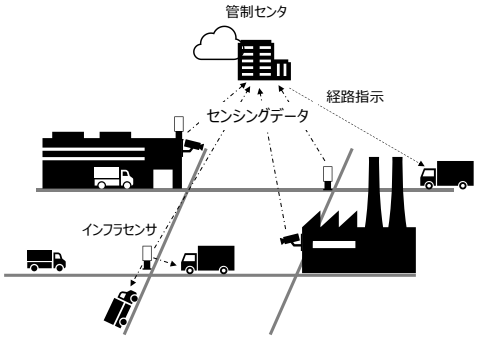

Fig. 1 Infrastructure cooperated autonomous driving system

インフラ協調型自動運転システムでは自動運転車両，インフラセンサ，管制センタが相互に連携する．各コンポーネントが連携することにより，安全で効率的な自動運転車両の運用を実現する．インフラセンサは交差点などに設置され，カメラや LiDAR などが用いられる．インフラセンサを活用することで自動運転車両のセンシング領域を拡張することができ，同一のセンサでも車両との距離に応じて異なる効果を得ることができる．インフラセンサと車両が近距離にある場合は，インフラセンサから車両に，路車間通信により死角の情報を送信することにより，死角が原因による衝突を防止するなど，安全性の向上に効果がある（Fig. 2 の上段）．インフラセンサと車両が遠距離にある場合は，インフラセンサにて検出した情報を管制センタに送信し，管制センタにおいて当該情報をもとに車両への経路指示を行うことで，あらかじめ障害物を迂回する経路を選択するなど，効率向上に効果がある（Fig. 2 の下段）．

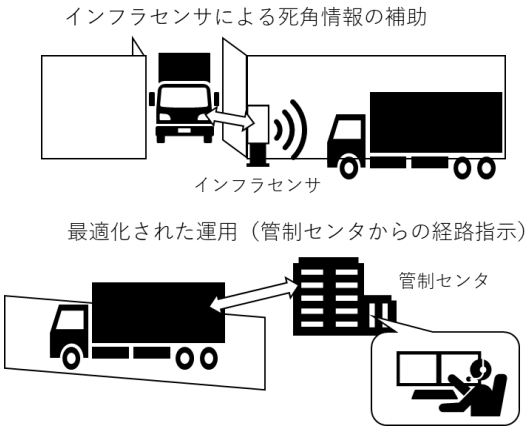

Fig. 2 overview of the autonomous vehicle system with the control center and road-side sensors

ODD 定義により算出される経路が変化する例を Fig. 3 に示す．

Fig. 3 上図は，ODD に車幅が定義されており，通行可能道路を判定する場合の例である．初期状態として，図中の青経路が車両に配布されていたとする．青色経路上の駐車車両を検出し，駐車車両横を通行不可と判断した際には，通行可能道路から駐車車両がある道路を外す．再度，経路探索を行い，当該道路を含まない赤色経路が算出され，車両に経路指示として配布する．

Fig. 3 下図では，車種により通行可能道路を判定する場合の例である．細い道路では大型車通行不可となっている道路があり，乗用車は通行可能であるが，大型車は通行不可である．この場合に，乗用車における通行可能道路には細い道路が含まれるが，大型車の通行可能道路には細い道路は含まれない．そのため，乗用車には赤色経路が配布 s れ，大型車には細い道路は青色経路が配布される．

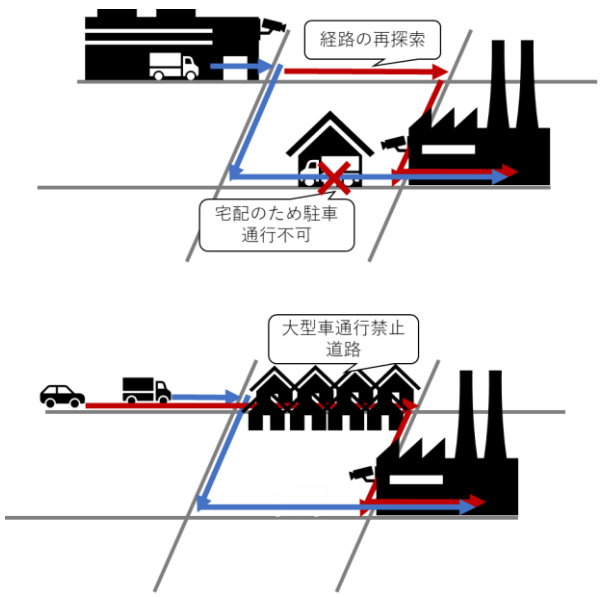

Fig. 3 examples of route change according to ODD definition

本研究では，このような，車両の運行設計領域を管理し，

Vol.2021-IFAT-142 No.11
Vol.2021-DC-120 No.11
2021/3/26

道路状況に応じて各車両に運行経路を指示することで自動運転車の運用効率を向上する管制センタのアーキテクチャを検討した．また，クラウド上に当該センタ機能を実装して評価することで，提案方式の有効性を確認した．

3. インフラ協調型自動運転システムの管制センタ

本研究の対象とする管制センタは道路の状況と，車両のODD をもとに, 各車両の通行可能な道路を管理する. また, 車両の現在位置と目的地から経路を算出し，当該車両に経路を送信する．経路算出に当たって，インフラセンサから障害物などの情報を受信した場合には，管理対象車両のODD に基づいて当該地点の迂回要否を判断し, 必要に応じて迂回経路の算出(経路の再計算)と車両への指示を行う．

3.1 システム要件

Table 1 に本研究の対象とする管制センタのシステム要件を示す.

Table 1 the requirements of the system

No.	要件
要件 1	道路状況を管理できること
要件 2	車両ごとの ODD を扱えること
要件 3	車両の ODD と道路状況から，対象車両が走行可能な道路を算出できること
要件 4	各車両が走行可能な道路から経路を算出できること

まず，管制センタにて道路状況と ODD を管理する必要がある．そのため，要件 1 として,「道路状況を管理できること」，要件 2 として,「車両ごとの ODD を扱えること」をシステム要件とする．

次に，各車両の経路を算出するためには，各車両がどの道路を通行可能であるのかという情報が必要になる．このとき，ある車両が当該道路を通行可能であるかどうかは，当該道路の状況と，当該車両で定義された ODD により異なる．そこで，要件 3 として,「車両の ODD と道路状況から，対象車両が走行可能な道路を算出できること」をシステム要件とする．

その後，走行可能道路の中から各車両が走行する経路を決定する必要があるため, 要件 4 として,「各走行可能な道路から経路を算出できること」をシステム要件とする．

3.2 管制システム構築における課題

管制システム構築に当たっては，システムの処理負荷変化への対応が課題となる．管制システムを運用するにあたり，業務変化や業務量変化により，管理する車両台数，管理する車種が変化し, 処理負荷が変わることが多い. また, 自動運転サービスの提供先である工場用途や鉱山用途などの用途が異なれば，管制センタの処理負荷は変わる．さらに，同じ用途向けでも事業規模により，管理台数等は変わるため，管制センタの処理負荷の変化対応可否は重要である．

処理負荷が増加する要因には，管理車両数，管理領域の広さ, 道路状況の更新頻度, ODD の定義項目内容, ODD の定義項目数がある．

管理車両数は，管理する車両が増えるほど，管理するODD の種類が増えるため, 通行可能道路の算出が必要な数が増え, 処理負荷が増える. また, 管理車両が増えるほど, 経路を算出する車両数も増えるため, 処理負荷に影響する．

管理領域の広さは，地図に含まれるリンク数やレーン数に影響するため，経路算出時の処理時間や道路状況の更新頻度に影響がある．管理領域が広くなるほど，基本的にはリンク数は増加し，探索空間が広くなる．そのため，経路算出の処理時間は長くなる．また，リンク数が多くなるほど，道路の母数が増えるため，道路が変化する確率が増える．

道路状況の更新頻度は，道路状況に変化がある度に，管制センタ側で処理を開始するため処理負荷に影響がある．

ODD の定義項目内容は, 通行可能な道路を算出する際の処理内容に影響がある. 例えば,“バス専用レーンはバスのみ通行可能とする”のような車種と道路種別により道路の通行可否を判断する際と，障害物の大きさと車幅を考慮し障害物横の通行可否を判断する際には, 処理負荷が異なる．

ODD の定義項目数は，各 ODD の定義処理を実行する数に影響するため，処理負荷増加の要因になる．

これらの各要因がどの要件に影響するかをまとめた表を Table 2 に示す.

Table 2 the affecting items to processing time

要因	要件 1	要件 2	要件 3	要件 4
(1)管理車両数		✓	✓	✓
(2)管理領域の広さ	✓		✓	✓
(3)道路状況の更新頻度	✓		✓	✓
(4)ODD の定義項目内容		✓	✓	
(5)ODD の定義項目数		✓	✓	

Table 2 によると，各要件において影響要因の数が一番多いのは，要件 3 であることが分かる．また，管制センタを運用するに伴い, (1)管理車両数, (4)ODD の定義項目内容, (5)ODD の定義項目数は変化しやすい要因である．そのため，要件 3 は負荷が増大しやすいと言える．よって，要件

3 を満たすにあたり，並列化可能な状態で実現することが，運用に伴う処理負荷増加に対応しやすい構造であると言える．

3.3 提案アーキテクチャ

3.3.1 モジュール構成

前述の要件および課題に基づき，本研究における管制システムのアーキテクチャを以下の通り提案する．この構成図を Fig. 4 に示す．

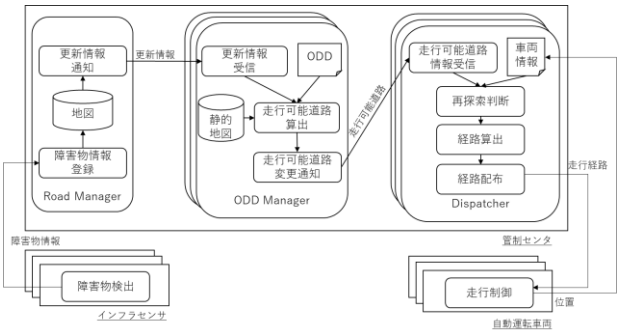

Fig. 4 overview of the architecture

Road Manager

要件 1 に対応するため，各インフラセンサから障害物情報を受信し，管理領域内の障害物情報を一元管理するコンポーネントを配置する．このコンポーネントを Road Manager とする．ここでの道路状況の管理には，Local Dynamic Map（LDM）を適用する．LDM は，地理的情報，周辺車両，道路状態，交通状況などを扱い，更新頻度に応じて階層構造で各情報を管理する [8] [9] [10]．LDM の第 1 階層では，道路形状や交差点等の地図の静的データを扱う．第 2 階層では，準静的なデータとして，信号，標識などを扱う．第 3 階層では，準動的なデータとして交通事故，渋滞，工事，路面状況などを扱う．第 4 階層では動的データとして，車両，歩行者，障害物位置などを扱う．

Road Manager は，インフラセンサから障害物情報を受信後，障害物登録処理により，地図に動的な情報として，障害物を記録する．その後，各 ODD Manager に変更があるリンクやレーンの情報を更新情報として，更新情報通知から送信する．更新情報には，LDM での第 3 階層や第 4 階層に該当する動的な情報をもつ．

ODD Manager

要件 2 および要件 3 に対応するため，複数の ODD を管理し，要件 1 で管理する道路状況を入力として通行可能な道路を算出するコンポーネントを配置する．このコンポーネントを ODD Manager とする．

ODD Manager は，Road Manager から更新情報を受信し，定義された ODD から走行可能な道路を算出する．

Road Manager から更新情報を更新情報受信にて受信し，予め定義された ODD，受信した更新情報，静的地図から，各 ODD に応じた通行可能な道路の算出を走行可能道路算出が行う．通行可能道路の算出には，管理領域内全てにおいて，通行可能かの判断を行う．その後，走行可能道路変更通知が各 Dispatcher に対し，通行可能道路を送信する．通行可能道路には，どのリンクやレーンが走行可能であるかという情報をもつ．

静的地図には，LDM での第 1 階層や第 2 階層に該当する静的な情報をもつ．走行可能道路には，どのリンクやレーンが走行可能かという情報をもつ．走行可能道路が，リンクレベルの情報をもつか，レーンレベルの情報をもつかは，Road Manager にて管理する地図の詳細度に従う．

ODD Manager は，ODD が異なる車両分稼働する．ODD が異なる単位は，車種が基本ではあるが，ユースケースにより，各車両で異なる ODD を定義する必要がある場合もある．その場合には，各車両分の ODD を定義し，各車両分の ODD Manager を稼働させる．各 ODD Manager 内の処理は独立しているため，常に並列処理が可能である．

Dispatcher

要件 4 に対応するため，要件 3 にて算出する通行可能な道路を入力とし，目的地までの経路を算出するコンポーネントを配置する．このコンポーネントを Dispatcher とする．

Dispatcher は，ODD Manager から走行可能道路を走行可能道路情報受信にて受信後，車両情報を利用し，経路を算出する．

車両情報には，車両の現在地，目的地，走行経路情報をもつ．車両現在地は各車両から位置情報を受信し，更新する．

走行可能道路を受信後，車両現在地，走行可能道路から現在の走行経路から再探索が必要か判断するのが再探索判断処理である．走行可能道路が更新されたことにより走行予定であった道路が走行できなくなった場合に，再探索が必要であると判断する．

再探索が必要であると判断した場合に，経路算出が車両の現在地，目的地，走行可能道路から目的地までの経路を算出する．

経路を算出後，経路配布が当該車両に対し，算出した経路を配布する．各車両は経路を受信後，受信した経路に従い走行する．

Dispatcher は，管理する車両分稼働する．しかし，ユースケースによりある車両で算出した経路が別の車両経路に影響を及ぼす際には，並列処理ができないため，Dispatcher の並列処理はできない．例えば，経路途中の立ち寄りがあり，具体的には，宅配の荷物の受け取り依頼や相乗りタクシーでのユーザのピックアップなどがある．この場合には，ある車両が受け取りやピックアップを行えば，他の車両にて受け取り等の立ち寄りは不要である．このように，他車両の走行経路により，目的地そのものが変化するユースケー

スなどでは，並列化は難しい．

3.3.2 並列化の実現方法

まず，静的なデータと動的なデータを分けて管理することにより，地図へのアクセス集中回避と各コンポーネント間の通信量削減を実現している．ODD Manager の通行可能道路算出には，動的なデータと静的なデータの両方が必要である．しかし，Road Manager のみで静的な情報を管理すると，通行可能道路の算出処理時に Road Manager にアクセスが集中してしまい処理負荷が増大する．そこで，静的なデータは変化しないため，予め ODD Manager にコピーを渡すことにより負荷を分散させている．また，動的なデータのみ Road Manager から通行道路算出処理に渡すことで最小の通信量を実現している．

次に，ODD の管理，静的地図，通行可能道路の算出処理を ODD Manager として，ひとつのコンポーネントとすれば，通行可能道路の算出は，入力である動的な道路状況にのみに依存させることができる．つまり，各通行可能道路の算出処理を独立させることができ，並列化処理が可能になる．

これらの方法を用いて，ODD Manager を複数稼働させ，並列処理化する．また，可能であれば Dispatcher も同様に複数稼働し，並列処理させる．

4. 評価

4.1 評価内容

管制センタの管理車両台数が増加するに伴い，ODD Manager の処理負荷が増大する課題があること，また，この課題に対して，ODD Manager の並列化が有効であることを確認する．それには，管理車両数を増やして処理時間が長くなること，また，ODD Manager の処理インスタン数を増やした際に，処理時間が短縮されることを確認する．

本研究の管制センタは限定領域での自動運転サービスへの適用を想定しているため，管理領域として面積約 1.34 ㎢，限定領域内で管理する車両数 40 台とした．管制センタは，管理する限定領域ごと（例えば，工場ごと）に構築することを想定する．また，本評価においては，車両毎に ODD が異なると仮定し，管理対象の ODD は，道路種別，障害物有無という 2 種類であるとした．評価シナリオとしては，特定の道路上に障害物が発生し，経路の再探索処理が行われるケースを対象とした．すなわち，本評価における道路状況の更新は 1 回のみ行われる．ODD が車両毎に異なると仮定したのは，車両数≧車種数であるため，本仮定により車種ごとの ODD の相違も包含できるためである．

前述のシナリオの実行に当たっては，Road Manager にて障害物情報を記録し，各車両の ODD Manager に変化がある道路の情報を通知する．その後，ODD Manager にて通行可否判定を行う．通行可否判定には，道路種別での通行判定，障害物横の通行可否判定を行う．ただし，今回は，車幅情報等を用いずに障害物がある道路は通行不可と判定するとした．つまり，障害物が発生した道路は通行不可へと必ず変化する．通行可能道路の算出後，ODD Manager から Dispatcher に通行可能道路を送信する．その後，Dispatcher にて，変化した通行可能道路を用いて，再探索を行うシナリオとした．

Table 3 に，評価時の各要因の設定値をまとめる．

Table 3 parameters in the evaluation environment

要因	評価パラメータ
(1)管理車両数	2, 5, 10, 20, 30, 40
(2)管理領域の広さ	面積：約 1.34 ㎢ リンク数：332 本 レーン数：881 本
(3)道路状況の更新頻度	1 回
(4)ODD の定義項目内容	道路種別，障害物有無
(5)ODD の定義項目数	2

4.2 評価環境

Amazon Web Services [11]上に評価環境を構築し，評価を行った．評価環境の情報を Table 4 に示す．提案したアーキテクチャの各コンポーネントを Python にて実装し，各コンポーネント間の通信には，MQTT を用いた．

Table 4 the information of the evaluation environment

項目	値
インスタンスタイプ	t2.xlarge
OS	Windows
VCPU	4
メモリ	16GB

4.3 評価方法

前述の環境において，ODD Manager の処理インスタンス数を変えた際の処理時間変化を比較する．具体的には，実験 1 として 1 インスタンス上に ODD Manager を車両数分動作させた際の処理時間と，実験 2 として 2 インスタンス上の各インスタンスに車両数/2 分を動作させた際の処理時間を比較した．実験 1 の構成図を Fig. 5 に示し，実験 2 の構成図を Fig. 6 に示す．今回は，Dispatcher に関しては 1 プロセスで逐次処理させ，平均処理時間を算出する．

車両数を変化させ，ODD Manager の他の処理時間の影響項目は一定とした．ODD Manager は車両数分稼働する．各車両数の実験回数は 6 回試行した処理時間の平均時間を処理時間とする．

Vol.2021-IFAT-142 No.11
Vol.2021-DC-120 No.11
2021/3/26

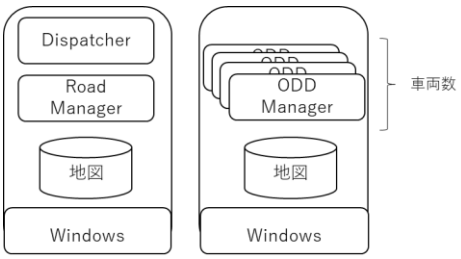

Fig. 5 environment of experiment #1

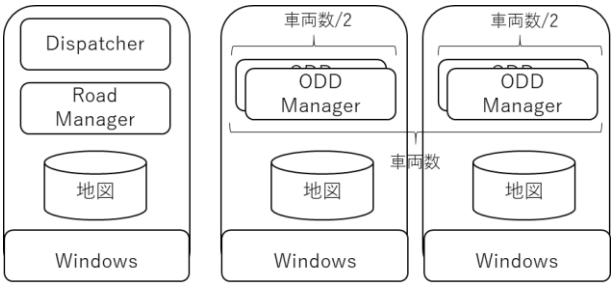

Fig. 6 the environment of the experiment #2

4.4 評価結果

実験 1 の実験結果を Table 5 に示し，実験 2 の実験結果を Table 6 に示す．また，各コンポーネントの比較結果を，ODD Manager は Fig. 7，Road Manager は Fig. 8，Dispatcher は Fig. 9 に示す．

ODD Manager の結果は，全ての車両数を処理するまでの全体時間を示している．Road Manager の結果は全ての道路状況を処理するまでの時間である．Dispatcher の値は 1 車両分の経路を算出するまでの時間である．

Fig. 7 より ODD Manager に注目すると，車両数が増加するに伴い全体の処理時間が長くなることがわかる．さらに，実験 1 と実験 2 を比較すると，実験 2 による 2 インスタンスによる処理時間は実験 1 による 1 インスタンで処理する際よりも約半分の処理時間になっていることが分かる．つまり，ODD Manager が扱う処理数が増えた際には，ODD Manager の処理インスタンスを増やすことで全体の処理時間を減らすことが可能であることが分かる．

車両数の増加に伴い全体処理時間が増加する原因としては，各インスタンスにおいて車両数分のプロセスを並列実行するリソースが足りずに，逐次実行に近い状態で処理しているためと思われる．

次に，Fig. 8 の Road Manager に注目すると，車両数に関わらずほぼ同じ処理時間で処理できていることがわかる．Road Manager はインフラセンサから受けた道路状況変化を記録し，各 ODD Manager に変化を通知する役割である．そのため，車両数に依存しない結果となったと思われる．

最後に，Fig. 9 の Dispatcher に注目すると，車両数の増加に少しずつ処理時間が増加しているが，ほぼ同じ処理時間で処理できていることが分かる．

Table 5 the results of the experiment #1 [ms]

車両数	ODD Mgt.	Road Mgt.	Dispatcher
2	56	69	414
5	93	70	424
10	167	70	432
20	294	86	438
30	445	71	439
40	581	71	447

Table 6 the results of the experiment #2 [ms]

車両数	ODD Mgt.	Road Mgt.	Dispatcher
2	73	73	409
5	69	71	407
10	92	72	425
20	166	72	423
30	236	73	437
40	296	76	440

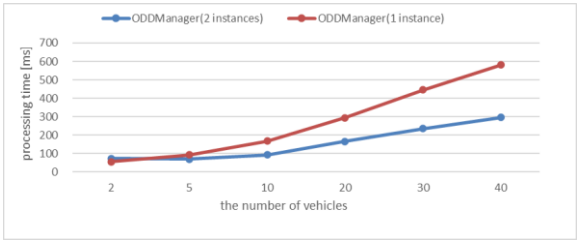

Fig. 7 the results of ODD Mgt.

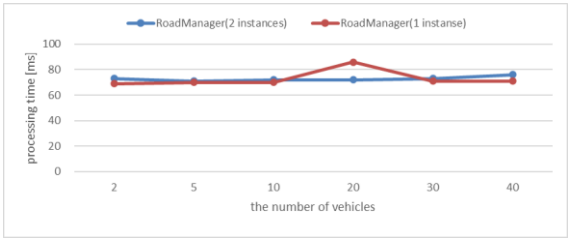

Fig. 8 the results of Road Mgt.

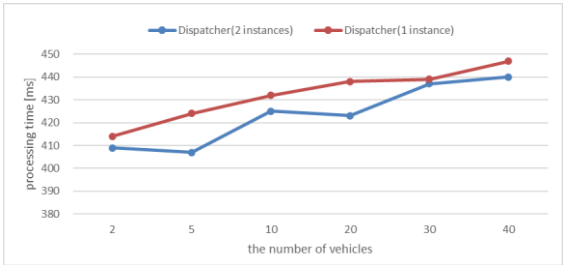

Fig. 9 the results of Dispatcher

Vol.2021-IFAT-142 No.11
Vol.2021-DC-120 No.11
2021/3/26

4.5 考察

ODD Manager の全体処理時間を車両数で割った値を車両あたりの処理時間と定義し，この値を比較する．Table 7 には実験 1 の車両あたりの処理時間を示し，Table 8 には実験 2 の車両あたりの処理時間を示す．Fig. 10 には，それらをグラフ化したものを示す．

これらから分かるのは，車両数を増加させた際に車両あたり処理時間が短くなるが，ある数まで増加させると車両あたり処理時間は短くならない．車両あたり処理時間が短くならないのは，各インスタンスにおいて車両数分のプロセスに十分なリソースが割り当たらずに逐次処理に近い状態になったと考えられる．例えば，1 車両分の処理時間が 10ms であった場合に，2 車両分を完全に並列処理できたとすると全体処理時間は 10ms になる．2 車両分を処理したので車両あたり処理時間は 5ms となる．さらに，5 車両分を処理した際に全体処理時間が 10ms であった場合には，車両あたり処理時間は 2ms となる．しかし，完全に並列処理できない場合には，全体処理時間は 10ms より長くなるため，車両あたり処理時間を算出した際に，ある車両数から車両あたり処理時間が短くならなくなる．

つまり，車両あたり処理時間は，車両数が増加する際に，いくつのインスタンスにて並列処理させれば良いかのひとつの指標となると考える．インスタンス数を増加させたが，車両数の増加により車両あたり処理時間が短くならなくなった場合には，さらにインスタンスを用意し，並列処理させた方が良いと言える．

Table 7 average processing time of the experiment #1 [ms]

車両数	ODD Mgt.	ODD Mgt./車両数
2	56	28.0
5	93	18.6
10	167	16.7
20	294	14.7
30	445	14.8
40	581	14.5

Table 8 average processing time of the experiment #2 [ms]

車両数	ODD Mgt.	ODD Mgt./車両数
2	73	36.5
5	69	13.8
10	92	9.2
20	166	8.3
30	236	7.9
40	296	7.4

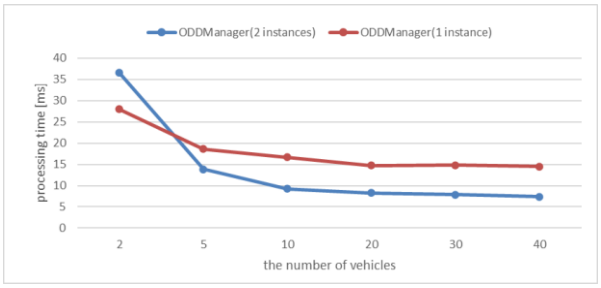

Fig. 10 average processing time of each process

5. 関連研究

現状，ODD や LDM は，ある車両に注目し，その車両における ODD や LDM について検討されることが多い．ある車両に関して，当該車両付近の LDM を作成し，当該車両の ODD と LDM から当該車両の軌跡生成を生成することが行われている．

しかし，工場などの運用最適化を行う際には，管制センタにて，各車両の ODD を管理し，限定領域全体の LDM を管理する方が，効率が良い．特に車両遠方を考慮した最適な経路を算出する場合に，効果が大きい．車両付近の LDM を利用するだけでは，遠方を考慮した最適な経路を算出することは難しく，経路上の通行不可地点は現地付近に到達して初めて認知するためである．

奥村らは，車両遠方の特に交差点した手法を提案している [4]．自動運転車と管制センタの連携による，見通しの悪い交差点などにおける交差点通過時の車両制御方法を提案している．

しかし，経路全体の最適化や各車両・車種により ODD が異なる点には着目しておらず，管制センタにて各車両の ODD や LDM の道路状況を統合管理する議論はなされていない．

6. まとめ

本研究では，管制センタにて複数車種の ODD を管理するためのアーキテクチャについて提案した．管制センタにて複数車種を管理する際には，管理車種の増加に伴い，ODD から算出する走行可能道路の算出負荷が増加し，処理時間が長くなる課題がある．

本提案では，この課題に対し，並列処理可能にすることで処理時間短縮可能なアーキテクチャを提案し，実験から並列処理による負荷分散が可能であることを示した．

参考文献

[1] “実用化に向け進化した e-Palette を公開,” トヨタ自動車, [オンライン]. Available: https://global.toyota/jp/newsroom/corporate/34527255.html. [アクセス日: 1 3 2021].

[2] 濵田 朋之 , 齋藤 真二郎,“鉱山合理化のためのダンプトラック自律運転システム,” 日立評論, 2017-10, pp. 527-531, 2017.

[3] 自動走行ビジネス検討会, “「自動走行の実現に向けた取組報告と方針」 Version 4.0,” 2020.

[4] 奥村文洋ら,“自動運転車の運行設計領域拡大のための時空間制約を用いた交通管制システム,” 電子情報通信学会論文誌, 2021.

[5] Weigang Wu, Jiebin Zhang, Aoxue Luo , Jiannong Cao, “ Distributed mutual exclusion algorithms for intersection traffic control,” IEEE Transactions on Parallel and Distributed Systems, 2014.

[6] 公益社団法人 自動車技術会, “自動車用運転自動化システムのレベル分類及び定義,” 2018.02.01.

[7] Automated Vehicle Safety Consortium, “AVSC Best Practice for Describing an Operational Design Domain: Conseptual Framework and Lexicon,” 2020.04.

[8] ETSI, “ETSI TR 102 863 v1.1.1(2011-06) Intelligent Transport Systems(ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map(LDM); Rationale for and guidance on standardization, ” 2011.06.

[9] ETSI, “ETSI EN 302 895 V1.1.1 (2014-09) Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)”.

[10] “ISO/TS 17931:2013 Intelligent transport systems − Extension of map database specifications for Local Dynamic Map for applications of Cooperative ITS,” 2013.06.

[11] Amazon Web Services, “https://aws.amazon.com/jp/”.