

インフラ型車両制御システムにおける クラウド-エッジサーバ間のハンドオーバー手法

鈴木 有也^{1,a)} 佐々木 健吾¹ 牧戸 知史¹

概要: 本研究では、インフラ型車両制御システム（都市レベルの大域的な車両制御を行うクラウドと、交差点等の局所的な車両制御を行うエッジサーバから構成される車両制御システム）において、クラウドとエッジサーバ間の車両制御処理のハンドオーバーをシームレスに実現する。ハンドオーバー時に、車両制御を続けながら制御状態（制御プログラムの内部状態）を共有する手法を、模擬システムで評価したところ、本手法を用いない場合ではコースアウト率が 42.4[%] になるのに対し、本手法を用いる場合では 5.0[%] となり、ハンドオーバー時の制御の乱れを低減できることを確認した。

A Handover Method between Cloud and Edge Servers on Infrastructure-based Vehicle Control System

NAOYA SUZUKI^{1,a)} KENGO SASAKI¹ SATOSHI MAKIDO¹

1. はじめに

近年、自動運転に関する研究開発が国内外で活発に行われている。内閣府では、SIP（戦略的イノベーション創造プログラム）において自動走行システムを研究課題のひとつに挙げている [1]。また、Google は、LIDAR、レーダ、カメラ等のセンサを車両に搭載し、自律的に道路を走行する車両システムを構築している [2]。本研究では、自動運転を実現する一形態として、インフラ型自動運転に関する技術開発を行っている [3]。インフラ型自動運転とは、Google 等が開発している自律型自動運転とは異なり、インフラに配置されているサーバ群（例えばクラウド、エッジサーバ [4] 等）のデータ管理機構に複数の車両の位置・速度等のデータを集約し、そこに配置された ITS アプリケーション（以降、単にアプリケーションと略す）が、そのデータ管理機構の情報を用いて車両群を制御する自動運転の形態であり、Connected vehicle cloud [5]、Passenger vehicles [6] 等の発展形である。データ管理機構には、図 1 に示すようなダイナミックマップ [7][8]（Dynamic Map、以降、DM

と略す）を用いる。DM は、交通の安全にかかわる静的および動的な要素（車両情報・天候状態・交通状態・詳細地図等）を格納するデータの集合体である。DM を用いることで、広範囲に分布する複数車両の位置・速度・経路等を用いた最適な車両制御を行うことができる。たとえば、車両の流れを信号で制御する従来の交差点ではなく、車両を停止させず互いに他車両をすり抜けて走行するような交差点 [9] を実現できる。また、クラウドとエッジサーバそれぞれに DM を配置し、互いに連携することで、広範囲に分布する車両情報のセンシングと低遅延な車両制御が両立可能となり、かつ、クラウドに処理負荷が集中したときにも、エッジサーバに処理負荷を分散することが可能となる。

一方で、インフラ型自動運転の実現に向けて解決すべき課題のひとつに、クラウド-エッジサーバ間における車両制御のハンドオーバー手法を確立することが挙げられる。すなわち、クラウドで制御している車両が移動してエッジサーバの車両制御領域に入った場合、クラウドの車両制御処理をエッジサーバが引き継いで、シームレスに車両制御を続行することが必要となる。その逆も同様である。ここでいうシームレスとは、車両制御処理のハンドオーバー時に車両挙動の乱れを最小限に抑え、ハンドオーバーしない場合

¹ 株式会社 豊田中央研究所
Toyota Central R&D Labs., Inc.

^{a)} naoya@mosk.tytlabs.co.jp

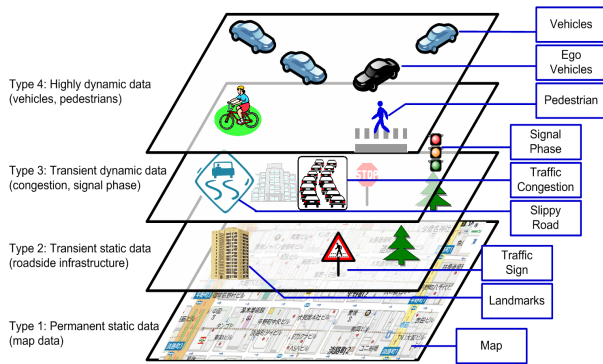


図 1 ダイナミックマップ
Fig. 1 Dynamic Map (DM)

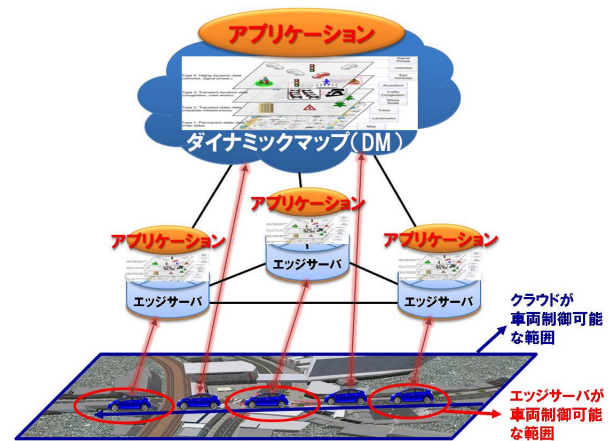


図 2 インフラ型車両制御システム

Fig. 2 Infrastructure-based vehicle control system

と同等、もしくはそれに近い車両制御が実現できることを指す。本稿は、インフラ型自動運転を実現する車両制御システム（以下、インフラ型車両制御システムと呼ぶ）におけるクラウドーエッジサーバ間のシームレスな車両制御ハンドオーバー手法の実現に焦点を当てる。

本稿の構成を以下に示す。第 2 章でインフラ型車両制御システムについて述べ、第 3 章で提案手法を説明する。第 4 章で評価を行い、第 5 章でまとめと今後の展開を述べる。

2. インフラ型車両制御システム

2.1 概要

インフラ型車両制御システムの概要を図 2 に示す。本システムは、クラウド、複数のエッジサーバ、および複数の車両から構成される。近年では、クラウドとの通信遅延を低減させるために、欧州の標準化機構である ETSI が、モバイルエッジコンピューティングの規格化を進めており、次世代移動通信システム 5G[10][11]におけるキーテクノロジーとして位置づけている [4]。本研究では、複数存在する 5G 基地局の一部にエッジサーバが配置されることを前提とする。よって、図 2 に示すように、クラウドが車両制御可能な範囲の中に、エッジサーバが車両制御可能な範囲が部分的に含まれることになる。クラウドとエッジサーバは、各車両から周期的に車両情報（位置、速度等）をセンシングし、制御量（アクセル量・ブレーキ量・ステアリング量等）を算出したのちに、各車両に対してそれらを送信する。各車両は、受信した制御量にしたがって走行を変化させる。

クラウドとエッジサーバには、それぞれ DM とアプリケーションが配置される。DM は、クラウドとエッジサーバによって、各車両から周期的に送信されてくる車両情報をもとに構築される。よって、クラウドが構築する DM とエッジサーバのそれでは DM の構築範囲が異なる。すなわち、エッジサーバが構築する DM は、エッジサーバが基地局に配置されると仮定すると、エッジサーバのセル範囲である半径数 [km] 程度の車両情報を入力として DM を構築

するのに対し、クラウドが構築する DM は、各エッジサーバが有する車両情報に加えて、エッジサーバを有しない基地局からの車両情報をも入力として DM を構築するため、半径数十 [km] を範囲とする DM となることを本研究では想定する。

一方、アプリケーションは、DM の情報を用いて各車両の制御量を算出する。アプリケーションには、例えば、車両速度を制御して全体の交通流を最適化するアプリケーション、死角からの車両の飛び出しなどを検知しブレーキ制御を行って衝突を回避するアプリケーション等が挙げられる。これらのアプリケーションから車両に対して制御量が送信される。

2.2 課題

インフラ型車両制御システムにおいて解決すべき課題のひとつに、クラウドーエッジサーバ間における車両制御のハンドオーバー手法を確立することが挙げられる。以降、説明を簡潔にするために、車両制御処理を実行する主体を制御主体と呼ぶ。たとえば、クラウドが車両を制御しているのであれば、制御主体はクラウドである。同様に、エッジサーバが車両を制御しているのであれば、制御主体はエッジサーバである。本システムにおいて常に車両を制御しようとした場合、クラウドとエッジサーバそれぞれの車両制御可能な範囲を、車両が自由に通過できることを考慮しなければならない。よって、エッジサーバが車両制御可能な範囲において車両が走行している場合は、そのエッジサーバが制御主体となって車両を制御し、車両が移動して、そのエッジサーバの車両制御可能な範囲を逸脱した場合は、クラウドが制御主体となって、車両制御をそのエッジサーバから引き継ぐ必要がある。本研究では、この引き継ぎを車両制御ハンドオーバー、あるいは単にハンドオーバーと呼ぶ。しかも、そのハンドオーバー時には、制御主体の切り替えによる車両制御の空白時間を極力なくし、車両制御を行うア

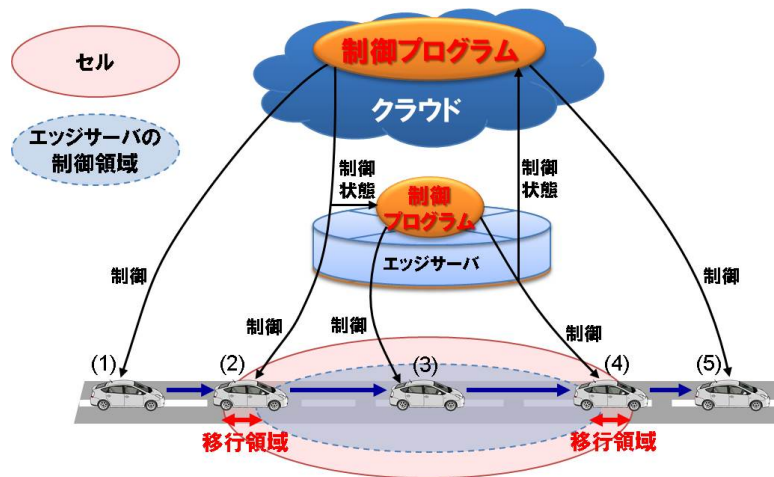


図 3 車両制御ハンドオーバー手法
Fig. 3 Vehicle control handover method

プリケーションの引き継ぎを適切に行うことにより、車両挙動の乱れを最小限に抑える必要がある。理想は、ハンドオーバーしない場合と同等、もしくはそれに近い車両制御の実現である。

移動局のハンドオーバー手法に関しては、従来技術が数多く存在し [12]、近年では、移動局の経路情報を用いて基地局が効率よくハンドオーバーを行う手法 [13] も存在する。しかしながら、これらの手法は、移動局である車両を基地局が制御することを前提としておらず、ハンドオーバー時に車両制御処理をシームレスに引き継ぐことまで考慮されていないという問題点がある。本研究は、この問題点の解決に焦点を当てる。

3. 提案手法

3.1 特徴

本研究が提案する車両制御ハンドオーバー手法（以下、本手法と略す）の特徴を以下に述べる。図 3 に示すように、エッジサーバが設置されている基地局のセルの内側に移行領域を設置して、車両が移行領域に進入したときには（図 3-(2)、および (4)）、それまでの制御主体であるクラウドまたはエッジサーバが車両制御を続けると同時に、他方に制御状態を送信する。そして、制御状態を受信した制御主体が、その制御状態を用いて制御処理を引き継ぐことによりハンドオーバーを実現する。ここでいう制御状態とは、制御プログラムの内部状態（変数）を指す。その内部状態を、送信側の制御主体と受信側の制御主体で共有することにより、車両の挙動の乱れを最小限にして車両制御処理を引き継ぐことをねらう。クラウドの制御プログラムとエッジサーバのそれは同一であっても異なってもよい。異なっている場合は、互いの制御状態を送受信する際に、制御状態の変換を適切に行うことで対応する。以降の説明では、説明を簡潔にするために、同一の制御状態を持つ制御

プログラムがクラウドとエッジサーバに配置されているものとする。

3.2 方法

本手法による車両制御ハンドオーバーの詳細を、図 3 中の車両位置に付与している番号に沿って説明する。図 3-(1) のように、エッジサーバが配置されている基地局のセル外で車両が走行している場合、クラウドは、エッジサーバが存在しない基地局を介して車両を制御する。車両は、クラウドに対して車両データ（位置、速度等）を周期的に送信し、クラウドは、その車両データを入力として制御プログラムを動作させ、制御データ（アクセル量、ブレーキ量、ステアリング量等）を車両に送信する。

図 3-(2) は、制御主体がクラウドの時に車両が移行領域に進入した場合を表している。この場合、クラウドの制御プログラムが、車両から周期的に受信する車両データに基づき、車両が移行領域に進入したことを検知する。その後、クラウドからエッジサーバに制御データとともに制御状態データ（制御プログラムの内部状態）を送信する。エッジサーバは、クラウドから送信されてきた制御データと制御状態データのうち、制御データのみを車両に送信する。エッジサーバでは、制御状態データが送信されてきたということは車両が移行領域に進入したと判断し、エッジサーバの制御プログラムを実行する準備を始める。エッジサーバの制御プログラムの準備が整うまで、エッジサーバを介してクラウドが車両を制御し続ける。たとえ、エッジサーバに多数の制御プログラムが存在し、実行準備が整うまでに時間がかかっても、車両がクラウドの制御下にあるため安定した車両制御が行われる。

図 3-(3) は、エッジサーバによる車両制御を表している。エッジサーバの制御プログラムの実行準備完了後、クラウドから制御データと制御状態データが送信されてきたと

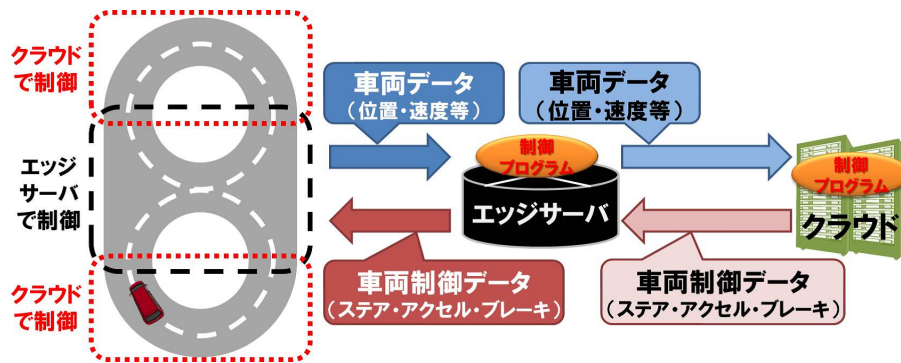


図 4 マイクロカープロトタイプの構成

Fig. 4 The structure of the micro car prototype

き、エッジサーバはそれを検出し、そのときの制御状態を入力として制御処理を開始する。エッジサーバが制御主体の間、クラウドは、制御処理を中断するため、他の処理にその計算リソースを割り当てることができる。

図 3-(4) は、制御主体がエッジサーバの時に車両が移行領域に進入した場合を表している。エッジサーバは、車両から周期的に受信する車両データに基づき、車両が移行領域に進入したことを検知する。その後、クラウドに対して制御データと制御状態データの送信を開始する。エッジサーバは、クラウドに対してそれらを送信しながら車両を制御し続ける。

図 3-(5) は、クラウドの制御再開を表している。クラウドの制御プログラムの実行準備完了後、エッジサーバから制御状態が送信されてきたときに、クラウドがそれを入力として制御処理を再開する。クラウドはエッジサーバを介して制御データを車両に送信する。エッジサーバは、クラウドから送信されてきた制御データを検出し、車両の制御処理を中断する。移行領域を通過した後は、クラウドが、エッジサーバが存在しない基地局を介して車両を制御する。

4. 評価

4.1 評価方法

4.1.1 マイクロカープロトタイプ

本手法の評価には、インフラ型車両制御システムの模擬システムとしてマイクロカープロトタイプを用いる。図 4 にマイクロカープロトタイプの構成を示し、図 5 に実物の外観を示す。マイクロカープロトタイプは、市販の 1/10 スケールのラジコンを改造したマイクロカーを複数台走行させることができるプラットフォームであり、超音波を利用した測位システムにより車両データ（位置、速度、および車両角度）の周期的なセンシングが可能で、かつ外部から車両制御データ（アクセル量、ブレーキ量、ステアリング量）を送信することによってマイクロカーを制御できる。エッジサーバとクラウドを模擬した PC を 2 台直列に接続し、それぞれで車両制御プログラムを動作させ、かつマイ

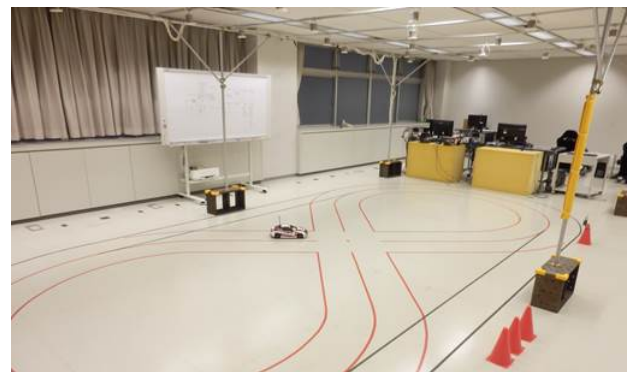


図 5 マイクロカープロトタイプの外観

Fig. 5 The exterior of the micro car prototype

クロカーの位置により制御主体を切り替えることで、インフラ型車両制御システムを模擬する。エッジサーバを模擬した PC では、Click modular router[14] が動作し、到着した車両データをエッジサーバで処理するか、もしくはクラウドに転送するかを判断を行うことができる。また、クラウドから送信されるメッセージは、すべてエッジサーバを介する。このような構成にすることにより、インフラ型車両制御システムと車両制御ハンドオーバーを模擬する。

4.1.2 コース

走行コースは、図 6 に示すように、縦 4[m] × 横 8[m] の平面上に設置された 8 の字コースを使用する。また、道路幅と車体のスケールは、図 7 に示すように、都市部の道路と乗用車（ここでは都市部の自動車専用道路とそこを走行する乗用車モデル）に対して約 1/10 になるように設計している。

4.1.3 制御主体になる領域の設置

本評価では、制御主体の切り替えパターンを 2 種類用意し、前述のコース上において、クラウドが制御主体になる領域とエッジサーバが制御主体になる領域を定義する。一方は、本手法を用いない場合、すなわち移行領域を設置せずに切り替えを行う場合（図 8）である。この場合は、 $X = -2.0[m]$ と $+2.0[m]$ においてクラウドとエッジサーバ間の制御主体の切り替えを行うように設定する。移行領域

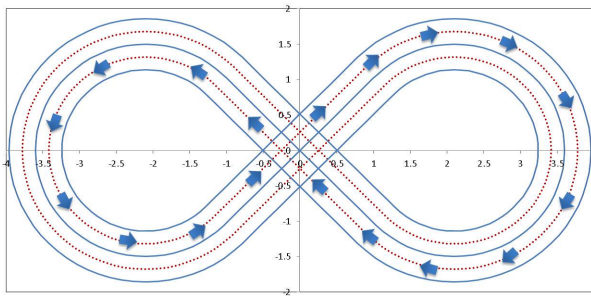


図 6 マイクロカープロトタイプのコース (単位: [m])

Fig. 6 The course of the micro car prototype

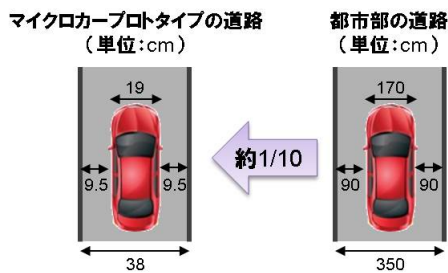


図 7 道路幅のスケール

Fig. 7 Road scale

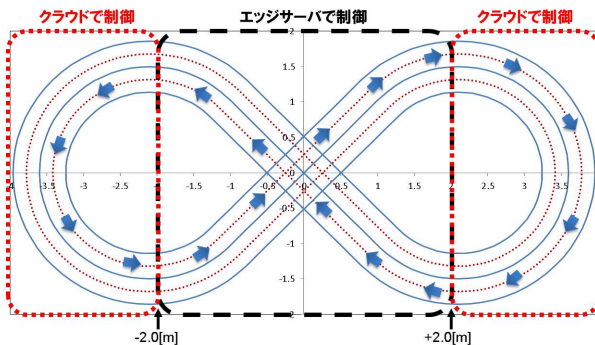


図 8 移行領域を設置しない場合

Fig. 8 The case of installing no handover regions

を設置しない場合は、制御状態の共有を行わずに切り替えるものとする。

もう一方は、本手法を用いる場合、すなわち移行領域を設置して切り替えを行う場合 (図 9) である。この場合は、 $X = -2.25[m] \sim -1.75[m]$, および $X = +1.75[m] \sim +2.25[m]$ の範囲を移行領域と定め、その範囲にマイクロカーが進入した場合に本手法を用いて制御主体の切り替えを行う。

加えて、制御主体を切り替えずにクラウドのみですべて制御した場合をも示し、第 4.2 節で示す評価項目においてこれら 2 種類と比較する。

4.1.4 車両制御アルゴリズムと制御状態データ

クラウド、およびエッジサーバでは、ラテラル制御アルゴリズム [15], およびロンジチュージナル制御アルゴリズム

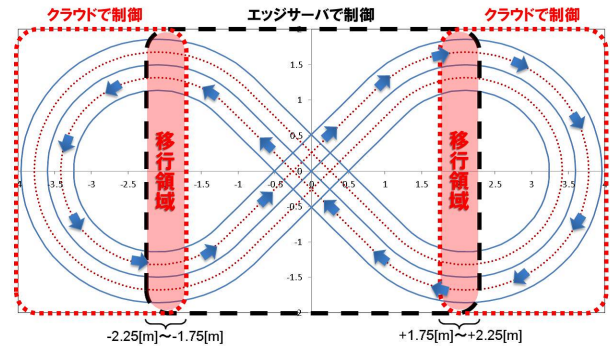


図 9 移行領域を設置する場合

Fig. 9 The case of installing handover regions

ム [16] を用いることにより車両制御を行う。これらのアルゴリズムを用いることで、周期的にセンシングした車両データ (位置、速度、車両角度) を入力として、マイクロカーを目標点まで到達させるための車両制御データ (アクセル量、ブレーキ量、ステアリング量) を車両に与えることができる。

ラテラル制御アルゴリズムは、操舵のためのアルゴリズムで、現在位置、現在位置における車両角度、目標位置、および目標位置における車両角度を与えると、現在位置と目標位置の 2 地点を結ぶ軌跡を 3 次曲線で近似したうえで、現在とるべき車両角度 α を得ることができる。 α を、現在位置ととるべき目標角度として PID 制御の計算を行うことで、車両に与えるべきステアリング量を求めることができる。

一方、ロンジチュージナル制御アルゴリズムは、加減速のためのアルゴリズムである。加速に関しては、現在車速と目標車速を入力として PID 制御の計算を行うことにより、車両に与えるべきアクセル量を求めることができる。減速に関しては、現在車速、停止位置までの残りの距離、および減速度を入力として、もし、指定された減速度で現在車速から減速した時に、停止位置までの残り距離で停止できない場合は、停止位置までの残り距離で停止可能な車速を逆算し、その速度を目標車速とする。その目標車速と現在車速を用いて PID 制御の計算を行うことで、車両に与えるべきブレーキ量を求めることができる。

本評価では、クラウドとエッジサーバそれぞれ、同一の車両制御アルゴリズムによってマイクロカーを制御する。よって、制御状態 (制御プログラムの内部状態) は同一であり、制御状態データは、主に PID 制御の偏差等からなる 259 [byte] である。

4.2 評価項目

評価項目は、表 1 に示すように、マイクロカーの (1) 走行軌跡、(2) 中心線からのずれ、および (3) コースアウト率とし、それぞれを (a) 本手法を用いない場合 (制御状態引き継ぎなし), (b) 本手法を用いる場合 (制御状態引き継ぎ)

表 1 評価項目

Table 1 Evaluation items

評価項目	説明
(1) 走行軌跡	5 周分走行したときの走行軌跡を示し ハンドオーバーによる制御への影響を表す
(2) 中心線からのずれ	車両が本来走行すべき中心線からの ずれを箱ひげ図で示しばらつきを表す
(3) コースアウト率	コースアウトの割合を示すことで 車両制御の安定性を表す

ぎあり), (c) 制御主体の切り替えを行わない場合 (切り替えなし) の 3 点において比較する. (c) においては, すべての領域をクラウドで制御した結果を示す.

マイクロカーの走行軌跡に関しては, 5 周分走行したときの走行軌跡をグラフ上に可視化する. また, 中心線からのずれに関しては, コースの中心線から車体がどれくらい離れて走行したかを, 車体から中心線への垂線の長さによって定量化し, その分散を箱ひげ図で示すことによって, (a) ~ (c) の車両制御の安定性を表す. コースアウト率に関しては, 図 7 に示したように, 道路の中心線から車体が 9.5[cm] ずれた場合はコースアウトとなるため, その割合を示す.

各項目を評価する際には, 本手法による制御主体の切り替えの効果に焦点を当てるため, ネットワークの不安定性によるパケット遅延, パケット消失, パケット到着順序逆転等がもたらす影響は除いたうえで評価を実施する. インフラ型車両制御システムにおいて, ネットワークの不安定性が車両制御にどのような影響を与えるかは, 文献 [3] をご参照いただきたい.

4.3 評価結果

4.3.1 マイクロカーの走行軌跡

各評価項目におけるマイクロカーの走行軌跡を, 図 10, 図 11, および図 12 に示す. 図 12 は, 制御主体の切り替えを行わずにクラウドですべての領域を制御した場合であり, この走行軌跡が基準となる. 図 10 は, 本手法を用いない場合, すなわち, 移行領域を設けず, かつクラウドとエッジサーバ間で制御状態を引き継がずに制御主体を切り替える場合である. 図 10 を概観すると, 図 12 と比較して, 全体的にコースの中心線から大きく外れて走行しているのがわかる. 特に, 制御主体の切り替えのあとで走行軌跡が大きく乱れているのがわかる. この理由としては, 制御主体の切り替え時にタイムラグが生じ, マイクロカーの制御主体が不在になる時間があることと, 制御主体の切り替え時に制御状態を引き継がないために, 適切な制御データ (アクセル量, ブレーキ量, ステアリング量) が出力されるまでに時間を要するからである. しかしながら, 切り替え後にある程度の時間が経過すると, コースの中心線を

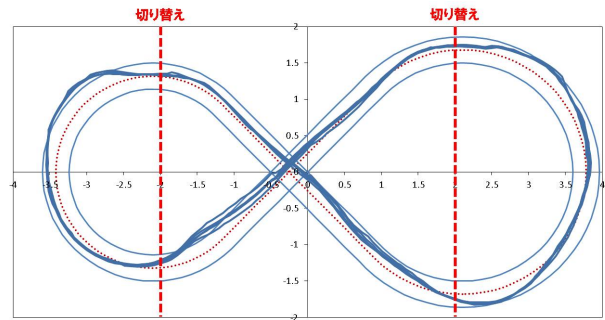


図 10 本手法を用いない場合の走行軌跡 (制御状態引き継ぎなし)

Fig. 10 Micro car orbit (not applying our method)

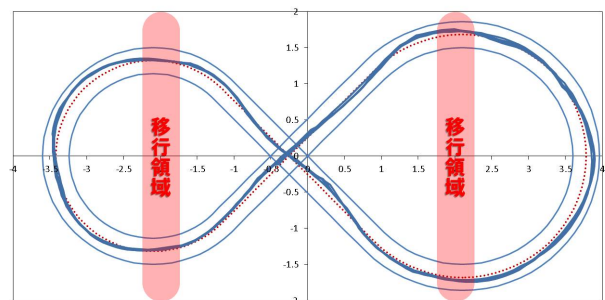


図 11 本手法を用いる場合の走行軌跡 (制御状態引き継ぎあり)

Fig. 11 Micro car orbit (applying our method)

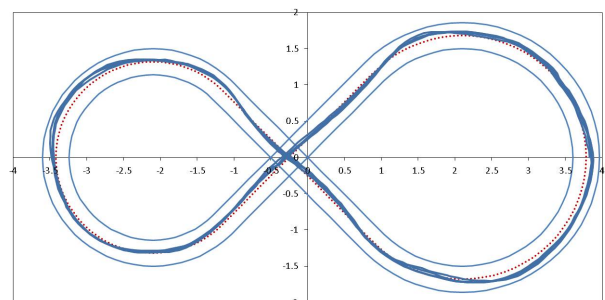


図 12 制御主体の切り替えを行わない場合 (切り替えなし)

Fig. 12 Micro car orbit (no handover)

走行するよう徐々に戻ってきているため, コースを走行できないほどの致命的な状態には至っていない.

図 11 は, 本手法を用いる場合, すなわち, 移行領域を設け, かつクラウドとエッジサーバ間で制御状態を引き継いで制御主体を切り替える場合である. 図 12 と比較しても, 図 11 に大きな走行軌跡の乱れは認められず, おおむね中心線に沿った走行制御ができています. また, 図 10 では制御主体の切り替え後に走行軌跡が大きく乱れているのに対し, 図 11 では, そのような乱れは認められない. よって, 本手法を用いる場合, 制御主体の切り替えによる車両制御の乱れを大幅に低減できることがわかる.

4.3.2 コースの中心線からのずれ

コース 5 周分の車両データ (位置, 速度等) より, コースの中心線からのずれを表したグラフが図 13 である. 車

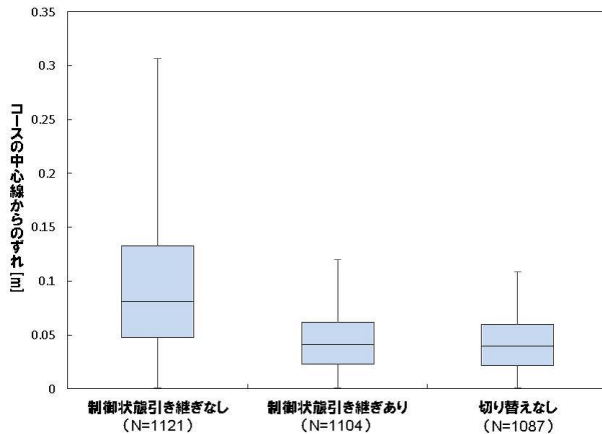


図 13 コースの中心線からのずれ

Fig. 13 Deviation from the course center line

両データのサンプル数は、それぞれグラフ中に N で示している。その N の分布を箱ひげ図で表している。箱ひげ図は、長方形とその両側に出たひげで、下から最小値、第 1 四分位点、中央値、第 3 四分位点、および最大値を表現する。本手法を用いない場合（制御状態引き継ぎなし）は、中央値が 0.1[m] 以内であるものの、大きく上ひげが出ており、最大 0.31[m] のずれが認められる。これは切り替えなしの 3.1 倍に相当する。一方で、本手法を用いる場合（制御状態引き継ぎあり）は、中央値が 0.05[m] 以内であって切り替えなしとほぼ変わらず、かつ最大値が 0.12[m] であり、切り替えなしの 0.11[m] と比較して 1.09 倍の増加で抑えられている。これらのことから、本手法を用いると、切り替えなしとほぼ同等の走行制御が行えていることが確認できる。

4.3.3 コースアウト率

図 7 に示したように、コースの幅員が 38[cm] で車体幅が 19[cm] であるため、コースの中心線から 9.5[cm] ずれた場合にコースアウトとなり、車体がコースからはみ出る。コース 5 周分の車両データのうち、コースアウトとして判定された割合をコースアウト率として図 14 に示す。この結果は、車線の幅員が 3.8[m] で車幅が 1.9[m] の場合に相当する。車両データのサンプル数は、図 21 と同様、それぞれグラフ中に N で示している。本手法を用いない場合（制御状態引き継ぎなし）は、40[%] 以上がコースアウトと判定され、制御の乱れが顕著であることがわかる。一方で、本手法を用いる場合（制御状態引き継ぎあり）は、コースアウト率が 5.0[%] であり、切り替えなしの 2.6[%] には劣るものの、本手法を用いない場合と比較して 1/8 以下にコースアウト率を抑えることができています。

4.4 評価のまとめ

これらの評価をまとめると以下ようになる。マイクロカーの走行軌跡に関して、本手法を用いる場合は、そうで

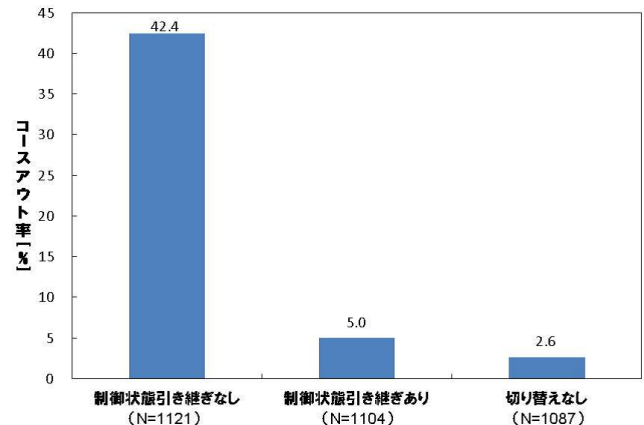


図 14 コースアウト率

Fig. 14 Course out rate

ない場合と比較して制御の乱れを大幅に低減でき、切り替えを行わない場合と同様にコースの中心線に沿った走行制御が可能であることが確認できた。コースの中心線からのずれに関しては、本手法を用いない場合では、切り替えなしと比較して最大値が約 3 倍となるのに対し、本手法を用いると 10[%] 以内の増加で抑えられることが確認できた。コースアウト率に関しては、本手法を用いない場合と比較して 1/8 以下にコースアウト率を抑えられることが確認できた。以上より、本手法により、車両制御の安定性を保ちながら、制御主体の切り替えが行えることが確認できた。

5. まとめと今後の展開

本稿では、インフラ型自動運転を実現するために、クラウド-エッジサーバ間における車両制御のハンドオーバー手法を提案し、マイクロカープロトタイプを用いて評価した。マイクロカーの走行軌跡、コースの中心線からのずれ、およびコースアウト率を測定することにより、本手法を用いる場合は、そうでない場合と比較して、車両制御の安定性を保ちながら、制御主体の切り替えが行えることを確認した。

今後の展開としては、本手法をエッジサーバ間のハンドオーバー技術に応用することが挙げられる。また、実車に適用し、制御主体の切り替えが車両制御にどの程度影響するかを評価することが挙げられる。

参考文献

- [1] 内閣府政策統括官：SIP（戦略的イノベーション創造プログラム）自動走行システム研究開発計画（2015）。
- [2] Guicco, E.: How google's self-driving car works, IEEE Spectrum Online (online), available from <http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works> (accessed 2017-01-24).
- [3] Sasaki, K., Suzuki, N., Makido, S. and Nakao, A.: Vehicle control system coordinated between cloud and mobile edge computing, In *proc. of 2016 55th Annual Confer-*

- ence of the Society of Instrument and Control Engineers of Japan(SICE), pp. 1122–1127 (2016).
- [4] ETSI: Mobile Edge Computing - A key technology towards 5G, Introductory technical white paper, European Telecommunications Standards Institute (2015).
 - [5] Ericsson: Connected vehicle cloud, Ericsson (online), available from <http://www.ericsson.com/ourportfolio/transport/connected-vehicle-cloud> (accessed 2016-02-24).
 - [6] Cisco: Passenger vehicles, Cisco (online), available from <http://www.cisco.com/web/strategy/transportation/passenger.html> (accessed 2016-02-24).
 - [7] ETSI: Intelligent transport systems(ITS); Vehicular communications; Basic set of applications; Local dynamic map(LDM), EN 302 895, European Telecommunications Standards Institute (2014).
 - [8] 渡辺陽介, 高木建太郎, 手嶋茂晴, 二宮芳樹, 佐藤健哉, 高田広章: 協調型運転支援のための交通社会ダイナミックマップの提案, DEIM Forum 2015 (2015).
 - [9] Dresner, K. and Stone, P.: A multiagent approach to autonomous intersection management, *Journal of Artificial Intelligence Research*, Vol. 31, No. 1, pp. 591–656 (2008).
 - [10] NTT ドコモ: ドコモ 5G ホワイトペーパー (2014).
 - [11] NTT ドコモ: 5G 無線アクセス技術, NTT ドコモテクニカルジャーナル, Vol. 23, No. 4, pp. 18–29 (2016).
 - [12] Dutta, A. and Schulzrinne, H.: *Mobility protocols and handover optimization*, Wiley-IEEE press (2014).
 - [13] 田村豊武, 近藤毅幸, 植田佳央: 移動体通信システム, 移動体通信方法, 基地局およびその制御方法と制御プログラム, 特許第 5842405 号 (2016).
 - [14] Kohler, E., Morris, R., Chen, B., Jannotti, J. and Kaashoek, M.: The click modular router, *ACM Transactions on Computer Systems (TOCS)*, Vol. 18, No. 3, pp. 263–297 (2000).
 - [15] 津川定之, 村田智: 自律車両の操舵アルゴリズム, システム制御情報学会論文誌, Vol. 2, No. 10, pp. 360–362 (1989).
 - [16] 加藤晋, 津川定之: DGPS を用いた自律車両のための速度制御, 日本機械学会ロボティクス・メカトロニクス講演会’00 講演論文集 (2000).