

# 隊列走行時に周辺車両の不要な加減速を低減させる 追越動作制御方式

鈴木 理基<sup>1,a)</sup> 徳田 大誠<sup>1,b)</sup> 重野 寛<sup>1,c)</sup>

受付日 2012年4月3日, 採録日 2012年10月10日

**概要:** 隊列走行は協調走行アプリケーションの1つで, 特に省エネルギー効果が期待されている. 隊列走行では縦列の車群を形成して走行するため, 複数の車両が同時に隊列と位置を入れ替える際, 不要な加減速が生じ, 二酸化炭素排出量が増加する場合がある. 本論文ではこの現象を追越動作の干渉と呼び, 分散排他制御アルゴリズムに基づいた追越動作の制御方式を提案して干渉を回避する. 提案手法では, Token-Based アルゴリズムを採用し, 隊列の隣の車線をクリティカルセクションとして, 1台ずつ隊列と位置を入れ替えるように追越動作を制御する. 本論文では, 提案手法を交通流シミュレータとネットワークシミュレータを統合した環境でシミュレーションにより評価する. そして, 二酸化炭素排出量と加減速の量, および追越動作を要求してから着手するまでの時間の観点から提案手法の有効性を示す.

キーワード: 分散排他制御, 追越, 隊列走行, 二酸化炭素排出量

## Platoon Overtaking Action Control Method For Surrounding Vehicle's Acceleration

MASAKI SUZUKI<sup>1,a)</sup> TAISEI TOKUDA<sup>1,b)</sup> HIROSHI SHIGENO<sup>1,c)</sup>

Received: April 3, 2012, Accepted: October 10, 2012

**Abstract:** Platooning is one of the cooperative driving method and is highly expected as a solution for energy saving problems. In case that two or more vehicles are overtaking a platoon, the vehicles are sometimes forced to unwillingly accelerate or decelerate. It causes extra CO<sub>2</sub> emissions and we call it overtaking conflict. We propose an overtaking action control method applied from the token-based distributed mutual exclusion algorithm in order to avoid the conflict. We regard a part of lane where is next to the platoon as the critical section and only one vehicle is allowed to enter the critical section. The proposal is evaluated by computer simulations employing combined simulation environment which consists of a network simulator and a traffic simulator. The results show that the proposal is efficient in terms of the amount of CO<sub>2</sub> emissions, the amount of the acceleration and the time from a vehicle wants to overtake till a vehicle starts to overtake. We conclude the proposal is superior to the other methods.

**Keywords:** distributed mutual exclusion, overtaking action, platoon, CO<sub>2</sub> emission

### 1. はじめに

協調走行のアプリケーションの1つに隊列自動走行 [1], [2], [3] があり, その省エネルギー効果に期待が

高まっている. 隊列自動走行は複数の車両が短い車間距離を保ち, 高密度な車群を形成しながら自動で走行する. そして, ドライバが手動で運転する一般車両と同じ道路を混走する. 主としている目的は省エネルギー効果である. 車間距離を短くすることで走行時の空気抵抗を削減し, 道路容量を増やすことで渋滞を削減する. 省エネルギー効果以外にも, 渋滞を削減することは交通流を改善する. さらに, 前後車両との相対速度を小さくすることで衝突時の安

<sup>1</sup> 慶應義塾大学大学院理工学研究科  
Graduate School of Science and Technology, Keio University,  
Yokohama, Kanagawa 223–8522, Japan

a) masaki@mos.ics.keio.ac.jp

b) tokuda@mos.ics.keio.ac.jp

c) shigeno@mos.ics.keio.ac.jp

全性を高める。

しかし、隊列自動走行の実現に際して、あらかじめ考慮すべき課題がある。隊列は隊列に参加しない車両とは走行する車線を明確に分離することが理想とされているが、日本を含めた大多数の地域では道路インフラを隊列走行用に新たに設置することは難しく、隊列と隊列に参加しない車両とが同じ車線を混走することが要件とされている [2]。混走するという要件と、道路容量という限られたリソースや道路構造に起因して、実現するうえで解決しなければならない課題がいくつかある。たとえば、追越動作の干渉 [4], [5] や、高速道路のジャンクションや車線のマージによる合流、分流時に隊列と隊列に参加しない車両が互いの走行を阻害してしまう問題等があげられる。本研究では、道路構造によらず高速道路上においてはどこでも発生しうる一般性の高い問題である追越動作の干渉に着目し、干渉を回避することで隊列に参加しない車両の走行燃費を改善する手法について検討する。隊列走行の車両が指定レーンを走行しなければならず、一般車両がすべての車線を走行するという状況では、低速車両を隊列が追い越すために、低速車両が追越車線を走行しなければならない。すると、隊列走行車両の前方に低速車両、後方に高速車両が走行していた場合、隊列が前方車両を追い越す動作と後方車両が隊列を追い越す動作が干渉する。この干渉は隊列自動走行の本来の目的であるスムーズな交通流を妨げ、当事者同士の燃費効率を阻害する。さらに、ドライバーに本来意図していない操作を強いるため、安全性の観点からも対応が必要である。現在検討されている隊列は、1つの隊列が3台から8台で構成され、およそ50メートルもの長さに及ぶことが想定されている [2]。また、国内の高速道路では片側2車線の道路の区間が多いことを考慮すると、追越動作の干渉は頻繁に発生することが予想される。

本論文では、この追越動作の干渉を回避することで、隊列の周辺を走行する車両の不要な加減速を削減する。そのために、Token-Basedの分散排他制御アルゴリズムを導入した追越動作の制御手法を提案する。また、本論文では隊列の隣の車線をクリティカルセクションと見なし、1台ずつの車両が隊列と位置を入れ替えるよう制御する。そして、提案手法をシミュレーションで評価する。交通流シミュレータとネットワークシミュレータの統合環境で車両を走行させ、追越動作を制御する手法を用いた場合と用いない場合を比較する。評価指標には二酸化炭素排出量と加速度の量および追越動作を要求してから追越動作に着手するまでの時間を用いる。

以下、2章で追越動作の干渉について説明する。3章で追越動作の制御に利用する分散排他制御方式の既存手法を紹介する。4章で追越動作の制御方式を提案し、5章でシミュレーションにより提案手法を評価する。最後に6章で結論を述べる。

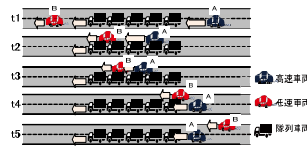


図 1 追越動作の干渉 (高速車両が減速)

Fig. 1 Overtaking conflict (High speed vehicle decelerate).

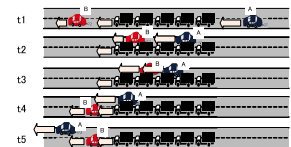


図 2 追越動作の干渉 (低速車両が加速)

Fig. 2 Overtaking conflict (Low speed vehicle accelerate).

## 2. 追越動作の干渉

隊列自動走行は高速道路で実施される場合と市街地道路で実施される場合が考えられるが、エネルギー ITS では、高速道路上で実現することが検討されている。高速道路では一般車両はすべてのレーンを走行可能とし、隊列自動走行に参加する車両は、隊列走行用の一部の車線である指定レーンを走行するものとする。各車両はGPSを用いて定期的に自身の位置を知ることができ、車両間で十分に高い精度で時間同期をとることができるものとする。また、車両間通信装置を用いて車両は相互に通信できるものとし [6], [7], 特に隊列自動走行に参加する車両はアドホックネットワークを構築できるものとする。各車両の通信範囲は802.11pを想定したおよそ400メートル程度を想定し、隊列自動走行を実現し維持するために十分な距離を通信できるものとする。

図 1 および図 2 に追越動作の制御を行わず、追越動作の干渉が発生する事例を示す。いずれの例でも、隊列の隣を前方から低速車両 B、後方から高速車両 A が接近する。片側 2 車線の道路で、一方を隊列が専有しているので、車両 B と車両 A は位置を入れ替えることができないので、2つの追越動作が干渉する。解消するためには、図 1 のように高速車両 A が減速するか、図 2 のように低速車両 B が加速することで、もう一方の車両の速度に合わせなければならない。速度を合わせた2台の車両が隊列を避けられるところまで移動したところで、2つの車線を使って高速車両が前方に、低速車両が後方に入れ替わる。位置を入れ替えるために速度を調整した車両は、改めて速度を元に戻し、隊列と位置を入れ替える。

いずれの場合も、先に譲った車両は不要な加減速をするので、燃費効率が悪くなってしまいます。さらに、ドライバーに意図していない動作を強いるので、交通流も妨げ、安全性にも悪影響が考えられる。つまり、協調隊列自動走行の本来の目的とそぐわない影響を及ぼす。

## 3. 排他制御手法

追越動作の干渉が発生する原因は、異なる方向から接近する2台以上の車両が同時に位置を入れ替えようとする際

に、相手の存在にあらかじめ気づくことができないことにある。なぜなら、複数の車両で構成する隊列は長いため、ドライバの視界から追越動作の干渉が発生するかどうか判断することが難しいからである。本論文では、この問題を通信によるドライバ支援によって解決する。追越動作の干渉を回避する方法には、ドライバの遠方映像をドライバに視覚的に伝えることでドライバの判断を支援する方法も考えられるが、本論文では、排他制御を導入して追越動作のタイミングをドライバに指示することで運転動作を支援する方法に着目する。隊列の隣の車線をクリティカルセクションと見なし、車両間で分散的に排他制御を行う。一般的に分散排他制御方式は大きく2つに分類される [8]。1つは Permission-Based アルゴリズムであり、もう1つは Token-Based アルゴリズムである。以下、それぞれのアルゴリズムを比較して、Token-Based アルゴリズムが追越動作の干渉の解決手段として適していることを示す。

### 3.1 Permission-Based アルゴリズム

Ricart らによって提唱された Permission-Based アルゴリズム [9] は、あるプロセス A がクリティカルセクションを要求する場合、プロセス A はクリティカルセクションを要求する可能性のある他のプロセスに対して、承認を求め、すべてのプロセスから承認が得られた場合にのみ、プロセス A はクリティカルセクションの処理に進入する。プロセス A からの要求を受信したプロセス B がクリティカルセクションを必要としていない際、プロセス B はプロセス A に承認を通知する。プロセス B がプロセス A と同様にクリティカルセクションを要求していた際、タイムスタンプを比較して、早い時刻に要求したプロセスを優先する。あるいは、クォーラムベースアルゴリズムと呼ばれる、選出されたメンバのうち過半数に支持されることで承認と見なし、投票によるアルゴリズムによってクリティカルセクションへの進入を許可するプロセスを選出する。

### 3.2 Token-Based アルゴリズム

Token-Based アルゴリズム [10] は、ある時間に1つのプロセスがクリティカルセクションに進入するように制御する。プロセス間でトークンをやりとりし、トークンを保持しているプロセスのみがクリティカルセクションに進入する権利を有する。簡単な制御によって、デッドロックと飢餓状態を回避することを保証する。さらに、Permission-Based アルゴリズムに比べ、少ないメッセージで合意形成を実現する。Token-Based アルゴリズムにおいてプロセスは、論理的にリング構造、あるいはツリー構造を構成することが想定される [11]。ツリー構造の場合、あるプロセス A がトークンを保持している際にプロセス B がクリティカルセクションを要求すると、プロセスの配置によっては要求してからトークンを取得するまでの時間が長くなること

がある。トークンを要求してから取得する過程は、あるプロセスがトークンの要求メッセージを送信し、トークンを保持しているプロセスが受信するというものである。そして、トークンを保持しているプロセスがトークンを送信し、トークンを要求しているプロセスがトークンを受信する。したがって、プロセスの通信トポロジの論理構造が深さ  $m$  の完全二分木構造で、2つのプロセスがそれぞれ葉ノードだった場合、トークンを要求してから取得するまでワーストケースで  $4m$  ホップの通信を必要とする。Token-Based アルゴリズムにおいてツリー構造を採用する場合は、あまり深くない構造であることが推奨されている。

しかし、Token-Based アルゴリズムの課題は2つある [12]。1つは、メッセージのオーバーヘッドを抑えたまま競合したプロセス間で公平なスケジューリングを行うことである。もう1つは、トークンを損失した場合に、トークンを再発行するために複雑な手続きが必要な場合があることである。トークンを再発行するためには、再発行する主体を選択しなければならず、リーダーを選出するアルゴリズムを用いて、様々な手法が提案されている [13]。

### 3.3 比較

追越動作の干渉問題を解決するにあたって Permission-Based アルゴリズムを用いた場合、ある車両からのリクエストに対して、隊列の通信範囲に存在するすべての車両が承認する。また、クリティカルセクションはつねに隊列の横に存在する。したがって、隊列をルートとしたツリー状のアドホックネットワークを構築することが想定される。1台の車両の通信範囲を考慮すると、追越動作を要求した車両と、承認する車両の通信を隊列が中継する必要がある。したがって、メッセージ量と、合意を形成するまでの時間という観点から Permission-Based アルゴリズムよりも Token-Based アルゴリズムの方が追越動作制御に適している。

また、アプリケーションを正常に動作させるために、クリティカルセクションの安全が保たれているとすると、Permission-Based アルゴリズムでは通信範囲内のすべての車両との通信が維持されなければならないのに対して、Token-Based アルゴリズムではトークンを保持する車両と、トークンを要求する車両との通信が維持されていれば、アプリケーションは正常に動作する。さらに、メッセージが届かなかったとき、Permission-Based アルゴリズムでは、その車両との通信が再度構築されるまでアプリケーションを再開できないのに対して、Token-Based アルゴリズムではその車両がクリティカルセクションに進入しないことが確認できれば、隊列がトークンを再発行することでアプリケーションを再開できる。その際、隊列のトークンは隊列に対して一意に定まり、複数の車両が隊列を中心としたトポロジを形成するので、隊列の代表車両であるマスタ車

両がトークンを再発行することが妥当な想定である。したがって、隊列走行というアプリケーションに Token-Based アルゴリズムを適用した場合には、リーダ選出のプロセスを必要とせず、アプリケーションを再開できる。頑健性の観点から Token-Based アルゴリズムが追越動作制御に適している。

以上より、中継の回数、合意形成までの時間、頑健性の観点から、隊列をコーディネータとした、Token-Based アルゴリズムが適していると予想される。本論文では、両手法を実装し、シミュレーションにより性能を評価してそれぞれの性能を確認する。

#### 4. 追越動作制御

本論文では、分散排他制御アルゴリズムに基づいた追越動作制御方式を提案し追越動作の干渉を回避する。提案手法は Token-Based アルゴリズムを採用する。マスタ車両と呼ばれる隊列の先頭車両が隊列の運行やメンバを管理しており、提案手法でも隊列のトークンを管理する。走行している隊列の隣の車線をクリティカルセクションとし、マスタ車両からトークンを受け取った車両のみが隊列を追い越す。

##### 4.1 追越動作制御方式の 4 フェーズ

提案手法は 4 つのフェーズを繰り返す。図 3 に 4 つのフェーズにおける隊列と車両の動作を示す。まず、将来隊列を追い越す車両と隊列がそれぞれ独立に走行している独立走行フェーズ。次に、隊列を追い越す車両が隊列に近づき、その存在を検知する接近フェーズ。そして、トークンを取得して、追越動作を実施する追越フェーズ。最後に、追越動作を完了して、トークンを隊列に返却する離脱フェーズ。以下、それぞれのフェーズを説明する。

###### (a) 独立走行フェーズ

初期状態は隊列と、隊列と位置を入れ替える車両がそれぞれ独立に走行している。隊列のマスタ車両は自分が管理する隊列のトークンを保持している。各車両はビーコンに

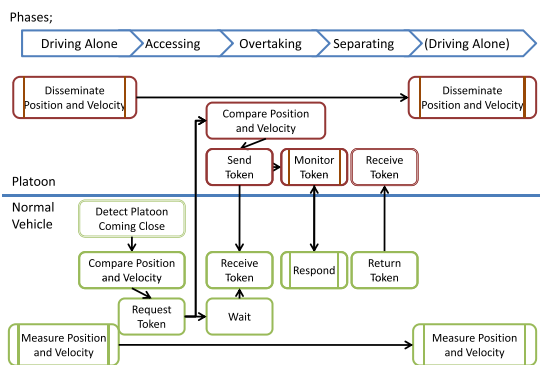


図 3 追越動作制御の処理の流れ

Fig. 3 Flow of overtaking action control.

より定期的に自身の位置や速度を周囲に通知し、さらに、自身が隊列のマスタ車両か、隊列のメンバ車両か、隊列に参加していない車両か識別できる。したがって、隊列の周囲を走行する車両は隊列が近くを走行していることを検知できるので、一定の距離よりも隊列に接近して、追越動作が発生すると判断すると、隊列のマスタ車両にトークン要求メッセージを送信する。

###### (b) 接近フェーズ

接近フェーズは隊列を追い越す車両が地理的に隊列に近づいてから、クリティカルセクションに進入するまでである。車両はビーコンで受信したメッセージから、隊列に接近したことを検知し、速度を比較する。図 4 に接近フェーズでの車両の動作例を示す。後方から高速車両 A が接近するとき、車両 A は隊列に対してトークン要求メッセージを送信する。隊列はトークン要求メッセージを受信すると、車両 A と位置、速度を比較する。車両 A の速度の方が大きい場合には車両 A へトークンを送信する。すでにトークンを他の車両に渡していた場合は、車両 A を待たせる。ほぼ同じタイミングで複数の車両からトークン要求メッセージを受信した場合はスケジューリングアルゴリズムに従ってトークンを送信する。トークンの取得を待っている車両は隊列の側で待機する。本研究では、車両の優先度については研究の対象外であるものとして検討せず、スケジューリングアルゴリズムには先着順に処理する FCFS ポリシに従ってトークンを授受する。

###### (c) 追越フェーズ

追越フェーズは隊列と位置を入れ替える車両がクリティカルセクションに進入してからクリティカルセクションを出るまでである。隊列のトークンは隊列と 1 対 1 の関係にあるので、1 度に 1 台の車両だけがクリティカルセクションを走行する。隊列のマスタ車両は他車両にトークンを渡す際に、どの車両が隊列のトークンを保持しているかを記録している。そして、ビーコンにより定期的にその車両の位置を確認し、トークンが存在していることと位置を監視する。

###### (d) 離脱フェーズ

離脱フェーズは車両が隊列のクリティカルセクションを出てから、マスタ車両にトークンを返却するまでである。

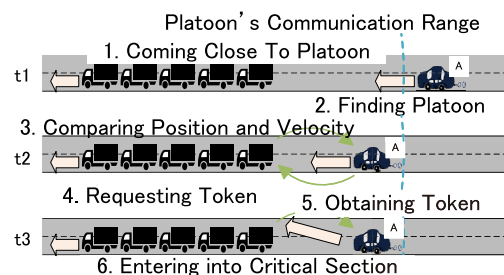


図 4 接近フェーズの動作

Fig. 4 Actions in access phase.

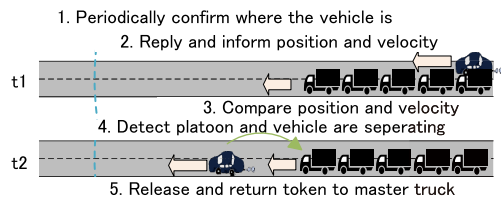


図 5 離脱フェーズの動作

Fig. 5 Process in separation phase.

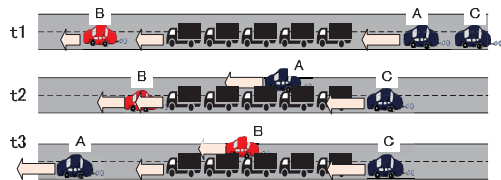


図 6 複数台が同時に同一方向から接近する例

Fig. 6 Accessing two vehicles from same direction.



図 7 縦列に車両が隊列を追い越す例

Fig. 7 In case long queue of vehicles are overtaking.

図 5 に離脱フェーズの動作を示す。マスタ車両と追越動作中の車両はビーコンを使って、定期的に互いの位置を確認する。同時にマスタ車両はトークンのタイムスタンプと、どの車両がトークンを保持しているかアナウンスする。そして、マスタ車両と追越動作中の車両はそれぞれお互いの位置と速度を比較する。追越動作中の車両がクリティカルセクションを出たことをどちらかの車両が検知すると、離脱の手続きを開始する。マスタ車両が先に検知するとトークンを保持している車両にトークン返却要求メッセージを送信する。追越動作中の車両が先に検知すると、自発的にマスタ車両にトークンを返却する。

#### 4.2 同一方向から接近する車両の制御

提案手法では、厳密に FCFS のポリシーに従ってトークンを車両に渡す。図 6 に 2 台の車両が続けて隊列に接近する場合の動作を示す。隊列に対して前方から 1 台の低速車両 B が接近し、後方から 2 台の高速車両 A, C が接近する。各車両が隊列に対してトークンを要求し、タイムスタンプの順番は早い方から A, B, C であったとする。提案アルゴリズムでは、A, C が続けてクリティカルセクションに進入するのではなく、A, B, C の順番でクリティカルセクションに進入する。要求を出した順番どおりにクリティカルセクションに進入させることで、公平性を保つためである。さらに、何台も続けて隊列を追い越そうとする場合を考える。図 7 に例を示す。車両 B は隊列からのトークンを待ち、隊列の近くを走行する。立て続けに車両がクリティカルセクションに進入することを許してしまうと、そ

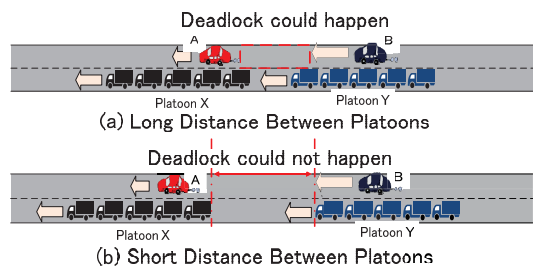


図 8 隊列間の距離

Fig. 8 Distance margin between platoons.

これらのすべての車両がクリティカルセクションから退出するまで、車両 B は待たなければならない。つまり、車両 B はクリティカルセクションに進入することを保証されない。この問題は飢餓問題と呼ばれ、それぞれの車両がいつかはクリティカルセクションに進入することを保証するために、提案手法は厳密に FCFS のポリシーに従ってトークンを渡す。

#### 4.3 隊列どうしの距離の制御

追越フェーズでは、複数の隊列の地理的な位置関係と、周辺車両の追越動作のタイミングによってデッドロックが発生する場合があるので、デッドロックを回避するために隊列どうしの距離を一定以上に保つように制御する。図 8 にデッドロックが発生する場合の例と、デッドロックを回避する制御を示す。前方を隊列 X が走行し、低速車両 A を追い越しているのと同時に、後方を隊列 Y が走行し、高速車両 B が隊列を追い越している。(a) に示す、隊列間でクリティカルセクションにマージンをとらなかった場合、車両 A は隊列 X に追い越されたのと同時に、隊列 Y に追い越される必要がある。しかし、隊列 Y は車両 B に追い越されているので、車両 A は隊列 X のクリティカルセクション上で隊列 Y のトークンを待機しなければならない。一方で、車両 B は隊列 Y を追い越したのと同時に隊列 X を追い越さなければならない。しかし、隊列 X は車両 A を追い越しているので、車両 B は隊列 Y のクリティカルセクション上で隊列 X のトークンを待機しなければならない。つまり、車両 A と車両 B はお互いがお互いの追越動作が終了するのを待たなければならない、さらに相手の追越動作が終了しなければ自分の追越動作を終了できないため、動作が行き詰ってしまう。この現象を追越動作のデッドロックと呼ぶ。提案手法では、隊列 X と隊列 Y の間で十分に距離をあけることで、デッドロックを回避する。例 (b) に示す、隊列間で十分に距離を開けた場合、車両 A は隊列 X に追い越された後に、隊列 Y の前方の走行車線で隊列 Y のトークンを待機する。また、車両 B は隊列 Y を追い越した後、隊列 X の後方の走行車線で隊列 X のトークンを待機する。以上より、デッドロックを回避する。

#### 4.4 トークンの再発行

トークンを返却する前に追越動作中の車両がマスタ車両の通信範囲から離脱した場合、隊列と紐付くトークンを再発行しなければならない。提案手法は、接近する車両からトークンの要求を受信すると、トークンを管理しているマスタ車両がトークンを送信する。その際に、隊列車両はどの車両がトークンを保持しているかを記録する。つまり、マスタ車両は周囲を走行する車両にトークンを渡す際に、相手車両のIDを記録し、どの車両が追越動作を要求し、動作中なのかを監視できるようにする。マスタ車両はトークンを保持している車両を通信により定期的に監視し、トークンが失われたことを検知する。隊列車両の通信範囲に車両が見つからない場合は、通信範囲はクリティカルセクションよりも十分に広いことを仮定しているため、協調範囲内にトークンは存在しないと判断する。また、通信機器の故障等を想定して、クリティカルセクションを車両が走行していないことを確認する。その後、隊列のマスタ車両がトークンを再発行する。したがって、トークンは1度に1つしか存在しないので、排他性が保証される。もし、通信範囲がクリティカルセクションよりも狭い場合には、隊列自動走行を維持することができないので、提案手法の動作も保証できない。通信範囲がクリティカルセクションよりは大きい十分に大きくない場合は、トークンを取得する前に隊列に接近してしまい、トークンの取得を待たなければならない、不要な加減速が発生する。したがって、追越動作制御は正常に動作するが、性能は低下する。

### 5. シミュレーション

本論文ではシミュレーションにより、提案手法を評価する。

#### 5.1 シミュレーション環境と条件

評価指標は、二酸化炭素排出量とそのモデルの加速度に依存する項、平均速度、そして追越動作を要求してから追越動作に着手するまでの時間である。シミュレーションはネットワークシミュレータ Scenargie 1.4 [14] と交通流シミュレータ MATES (ADVENTURE\_Mates Version 0.11 beta) [15] を用いる。それぞれのシミュレータを互い違いに実行し、一方の出力をもう一方の入力にする。すなわち、ネットワークシミュレータで計算した通信の成否をもとに車両の運動を交通流シミュレータが計算し、交通流シミュレータが計算した車両の運動の結果に依存してネットワークシミュレータが通信の成否を計算する。両シミュレータは一定のシミュレーション時間 (0.1 秒) ごとに実行を切り替える。

二酸化炭素排出量  $E$  は式 (1) に示す大口らによるモデル [16] で求める。

$$E = 0.3K_C T + 0.028K_C D + E_A, \quad (1)$$

ここで、 $K_C$  は燃料消費量を二酸化炭素排出量に変換する変換係数、 $T$  は旅行時間、 $D$  は旅行距離、 $E_A$  は式 (2) で示される加速度項である。

$$E_A = 0.056K_C \sum_{k=1}^K \delta_k (v_k^2 - v_{k-1}^2), \quad (2)$$

ここで、 $K$  は測定対象の車両台数、 $\delta_k$  は加速時に 1、その他の場合に 0 となるダミー変数、 $v$  は車両速度である。

評価するシナリオは、隊列と隊列に参加しない一般車両が高速道路を想定した一方通行の 2 車線の直線道路を走行するときに、各評価指標を測定する。比較する手法は、提案手法と Permission-Based で追越動作を制御する手法、追越動作を制御せずに、前方車両との距離に従って追越動作に着手する手法である。特に距離に従って追越動作に着手する手法は、前方車両との距離が 150m の手法と 70m の手法を比較する。主なパラメータを表 1 に示す。シミュレーション時間は 1,000sec とし、乱数のシードを変えて 3 回実行し、評価値の平均をとる。車両の挙動を正確に評価するため、車両の希望速度は 3 種類とし、低速車両は 60 km/h、高速車両は 100 km/h、隊列は 80 km/h を目標に走行する。すべての一般車両を運転するドライバはシステムからの指示に従うものとするが、各車両は追従モデル [15] に従って走行し、車両どうしが衝突することはないものとする。隊列の流入量は一定とし、隊列に参加しない車両の流入量を変化させた。その範囲は十分に空いている状態から渋滞が起きる直前の状態まで変化させ、それぞれの場合で評価指標を測定した。記載していないパラメータは ITS シミュレータ利用推進委員会が採用したパラメータに準拠する [17]。

#### 5.2 結果と考察

図 9 に隊列に参加しない車両の流入量に対する車両の平均速度を示す。どの手法も、車両の流入量が増えるに従い平均速度が低下する。そして、速度の手法間での有意差はない。道路上の車両密度が増えると追越動作がスムーズに行えず、追越動作の制御の有無にかかわらず、待ち時間が増えるためである。隊列に参加しない車両の流入がないとき、平均速度は 22.2 m/s、すなわち 80.0 km/h である。これは隊列の希望速度である。流入量が増加するにつれ、平均速度は低下し、流入量が 900 台/lane/hour のときには 16.7 m/s である。

隊列に参加しない車両の流入量に対する平均二酸化炭素排出量に関して、図 10 に、単位距離あたりの平均を示す。図 11 に、単位時間あたりの平均を示す。ただし、いずれも二酸化炭素排出量は隊列に参加しない車両 1 台あたりの排出量である。それぞれのグラフが追越動作を制御することで二酸化炭素排出量が低減することを示してい

表 1 シミュレーションの主なパラメータ

Table 1 Simulation conditions.

Model	Parameter Name	Value
Road Model	Road Length	10.0 km
	Number of Lanes	2(One Way)
	Dedicated Lane for Platoon	1
	Lane Width	3.5 m
Platoon Model	Number of Platoon Members	3
	Distance between Platoon Members	4.0 m
	Platoon Member Length	12 m
	Platoon Member Width	2.5 m
	Platoon Member Height	3.8 m
	Platoon's Expected Speed	80 km/h
	Volume of Platoon Inflow	60 Platoon/(lane·hour)
Non-Platoon Model	Volume of Inflow	Variable
	Low Speed Vehicle's Expected Speed	60 km/h
	High Speed Vehicle's Expected Speed	100 km/h
	Vehicle Size (L, W, H)	4.7 m, 2.5 m, 3.8 m
Network Protocol Model	Radio Communication Standard	802.11p
	Frequency	5.9 GHz
	Modulation	OFDM(QPSK1/2)
	Transmission Speed	6 Mbps
	Bandwidth	10 MHz
	Transmission Power	20 dBm
	MAC	CSMA/CA
	Propagation Model	ITU-R P.1411
Application Configuration	Request Range	100 m
	Permission Range	Communication Range

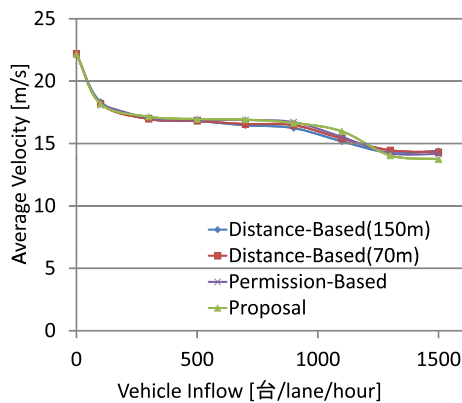


図 9 平均速度と車両流入量の関係  
Fig. 9 Average speed.

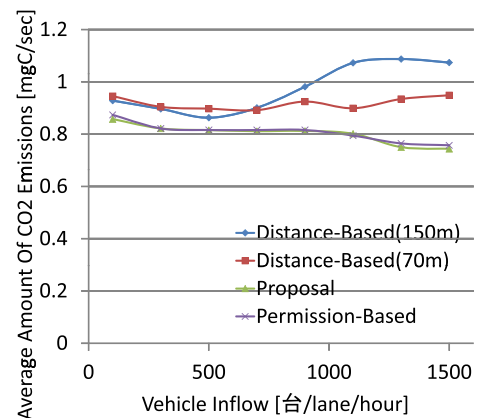


図 10 時間平均二酸化炭素排出量と車両流入量の関係  
Fig. 10 Average CO<sub>2</sub> emissions per trip time.

る。車両の流入量が小さい場合には、追越動作を制御する場合としない場合の差は小さい。車両の流入量が大きくなるにつれて、排出量の差が大きくなる。単位距離あたりの排出量の観点では、典型的な流入量が900台/lane/hourのときに、150mで追越動作に着手する手法に比べ、提案手法は17%改善した。提案手法とPermission-Basedの手法間の差はほとんどない。そして、70mで追越動作に着手する手法に比べ、提案手法は12%改善した。提案手法とPermission-Basedの手法間の差はほとんどない。単位時

間あたりの排出量の観点では、典型的な流入量である900台/lane/hourのときに、150mで追越動作に着手する手法に比べ、提案手法は19%改善した。そして、70mで追越動作に着手する手法に比べ、提案手法は12%改善した。提案手法とPermission-Basedの手法間に差はほとんどない。以上より、提案手法を含む、追越動作を排他制御アルゴリズムをもとに制御する手法が二酸化炭素排出量の観点で有効であることが示された。

隊列に参加しない車両の流入量に対する式(2)で示され

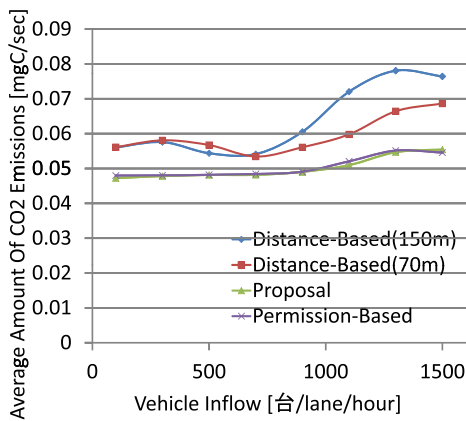
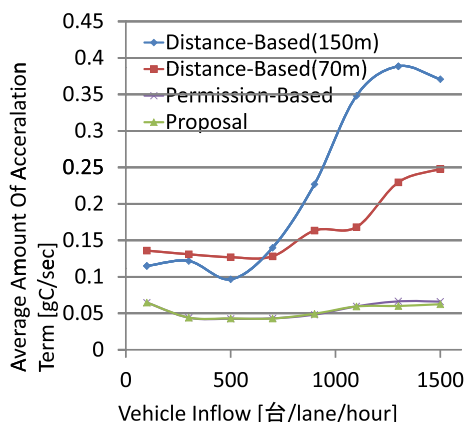
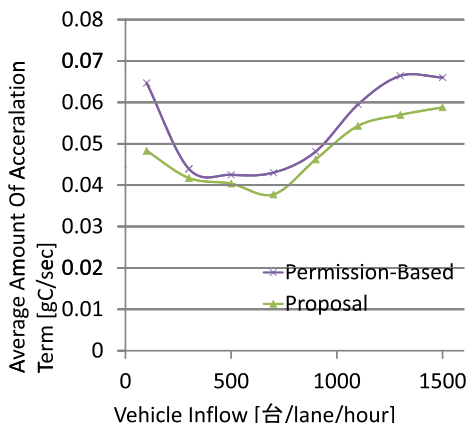


図 11 距離平均二酸化炭素排出量と車両流入量の関係  
 Fig. 11 Average CO<sub>2</sub> emissions per trip distance.



(a) 4 methods

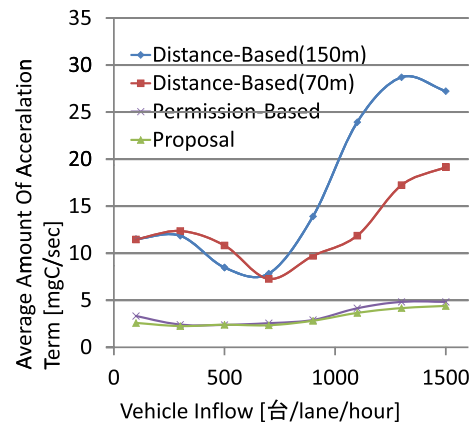


(b) Proposal And Permission-Based

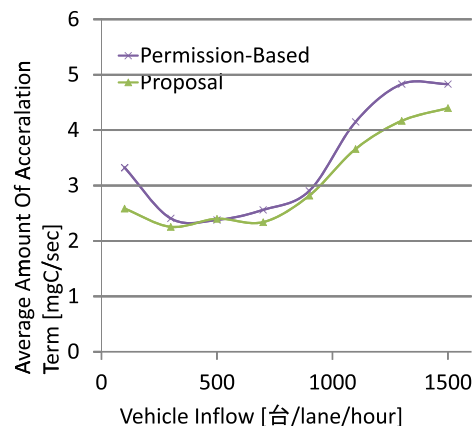
図 12 時間平均加速度量と車両流入量の関係

Fig. 12 Average acceleration term per trip time.

る加速度依存項  $E_A$  の、単位距離あたりの平均を図 12 (a) に、単位時間あたりの平均を図 13 (a) に示す。ただし、いずれも二酸化炭素排出量は隊列に参加しない車両 1 台あたりの排出量である。両グラフは二酸化炭素排出量の差は加速度項の差から生じていることを示す。加速度依存項の差は加速度の差を示しているの、追越動作を制御する手法は提案手法も Permission-Based 手法も追越動作の干渉を回避することで不要な加減速を減らしていることが



(a) 4 methods



(b) Proposal And Permission-Based

図 13 距離平均加速度量と車両流入量の関係

Fig. 13 Average acceleration term per trip distance.

分かる。車両の流入量が高い場合には、追越動作を制御する手法間でもわずかに差が見られる。図 12 (b) および図 13 (b) に提案手法と Permission-Based 手法の加速度依存項の単位時間あたりの平均と単位距離あたりの平均を示す。Permission-Based の手法よりも提案手法の方が低い値を示す傾向があり、たとえば渋滞の直前の状況である 1,300 台/lane/hour のときに、Permission-Based の手法に対して提案手法は、単位距離あたりの平均を 8%改善した。また、単位時間あたりの平均を 10%改善した。

図 14 に隊列に参加しない車両の流入量を変化させた場合の、追越動作を要求してから追越動作に着手するまでの待機時間の変化を示す。流入量がい小さいときには手法間の差は小さい。流入量が大きくなるにつれて、Permission-Based の手法は提案手法に比べて高い値を示す。これは、車両密度が高くなることで、すべての周辺車両から合意を得るプロセスに時間がかかることが要因である。車両がスムーズに通行できる流入量が 500 台/lane/hour のときに、Permission-Based の手法に対して提案手法は 47.6%追越動作に着手するまでの時間を短縮した。渋滞が起きる直前の状態である流入量が 1,300 台/lane/hour のときに、Permission-Based の手法に対して提案手法は 43.0%追越動



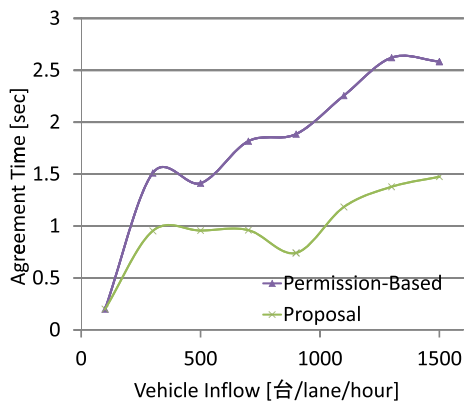


図 14 合意形成所要時間と車両流入量の関係

Fig. 14 Average time from send request till start overtaking action.

作に着手するまでの時間を短縮した。待機時間が短くなることで、2つの効果が期待できる。1つは車両が追越動作を開始する隊列と車両間の距離である要求開始距離を小さくすること、もう1つはドライバーへの精神的負担を軽減することである。図 14 より車両流入量が 1,300 台/lane/hour のときの Permission-Based の手法と提案手法の待機時間の差はおよそ 1.2 秒であった。両手法間での 1.2 秒という待機時間の違いから、高速で走行する車両であれば、提案手法は Permission-Based の手法に比べクリティカルセクションに進入するまでおよそ 30m 程度早く権利を取得することができる。Permission-Based の手法を用いると車両は隊列に合わせて加減速をしなければならない確率が高い一方で提案手法では加減速の必要とせずクリティカルセクションに進入する権利を得る確率が高くなる。つまり、待機時間が短いことで加速度量を低減させることができる。また直感的に、クリティカルセクションに進入する権利を得たときにクリティカルセクションまで十分な距離を確保できる方が急な運転操作をする必要がなく精神的負担が小さいことが想像できる。特に車両の流入量が大きく周囲が混雑している場合には、ドライバーの運転操作の選択肢も限られているので、時間的な制約を緩める効果が大きいことが予想される。つまり、車両の流入量が増加する条件下でも待機時間を短く保つことができる提案手法は、ドライバーの精神的負担を軽減するという点で意義がある。

以上の結果より、二酸化炭素排出量の観点から分散排他制御アルゴリズムに基づいて追越動作を制御することが有効であることが分かり、特に車両の流入量が高い場合にその効果が高いことが分かった。そして、追越動作を制御する方式の中でも Token-Based アルゴリズムに基づいた提案手法が Permission-Based の手法よりも加速度の量と追越動作に着手するまでの時間の観点から有効であることが分かった。

## 6. おわりに

本論文では、隊列自動走行の本来の目的を阻害する現象である追越動作の干渉を回避するため、分散排他制御アルゴリズムを隊列走行アプリケーションに導入した追越動作の制御方式を提案した。走行中の隊列の隣の車線をクリティカルセクションとして、Token-Based のアルゴリズムの FCFS ポリシに基づいて 1 台ずつ追越動作を実施するように制御する。追越動作の干渉を回避して、隊列を追い越す高速車両や隊列に追い越される低速車両の不要な加減速を低減させた。提案手法の有効性を示すため、シミュレーションにより評価した。評価環境に交通流シミュレータとネットワークシミュレータを互い違いに実行し、お互いの演算結果をもう一方の演算の入力とする統合環境を利用した。評価結果より、二酸化炭素排出量の観点から分散排他制御アルゴリズムを追越動作の制御に導入することの有効性を示した。車両の流入量が多いほど効果は高く、提案手法は距離を基準に追越動作を着手する方式に比べ、平均二酸化炭素排出量の観点で、典型的な流入量に対し、単位距離あたりの平均を 11.8%削減し、単位時間あたりの平均を 12.3%削減した。さらに、提案手法は Permission-Based の手法に比べ、加速度の量と追越動作に着手するまでの時間の観点から有効性を示した。車両の流入量が多いほど効果は高く、提案手法は Permission-Based の手法に比べ、二酸化炭素排出量の加速度項を単位距離あたりの観点で 8%削減し、単位距離あたりの観点で 10%削減した。以上より、提案手法の有効性を示した。

今後の課題について以下に述べる。まず、FCFS ポリシを緩和することが考えられる。本研究では厳密な FCFS を用いたが、数台を連続して同時にクリティカルセクションに進入するように制御することで、系全体の処理の最適化と車両間の公平性の観点からスケジューリングアルゴリズムをより効果的にできる可能性がある。また、車両の運動に関わる異常動作についても今後検討する必要がある。本研究では 4.4 節で述べたとおりトークンの紛失等の追越動作の制御アプリケーションの異常動作については検討したが、車両の運動に影響する異常動作については検討の対象外とした。解散や再構築等の車両の運動に関する手続きやモデルが明らかになった後、障害を回避し、正常な動作に復旧する手続きを検討する。そして、本研究では隊列の周辺を走行する車両はマスタ車両の指示に従うものとしたが、隊列の前方を低速のまま走行する車両について今後検討する必要がある。指示に従わない車両の追越動作を優先的に処理するか、あるいは対象の車両を障害物として扱うことで異常動作によって対象の車両を避けることが考えられる。また、隊列の解散や再構成等の隊列の柔軟な運転動作についても本研究では検討の対象外とした。隊列は希望速度に従って走行し、周囲の車両が隊列の運動に合わせて

走行することを前提としたが、たとえば、隊列が低速車両を追い越す場合にはいったん隊列が解散してメンバ車両が個別に低速車両を追い越した後に再度隊列を構成する方法も考えられる。隊列の解散や再構成のモデルが明らかになれば、トークンを待機する車両が一時的に隊列に参加することも考えられ、提案手法のさらなる付加価値もあわせて定量的な評価を実施する。

参考文献

[1] Shladover, S.E.: Automated Vehicles for Highway Operations (Automated Highway Systems), *The Institution of Mechanical Engineers Part I of Systems and Control Engineering*, pp.53-75 (2005).

[2] New Energy and Industrial Technology Development Organization (NEDO): Development of Energy-saving ITS Technologies available from (<http://www.nedo.go.jp/english/index.html>) (accessed 2012-10-15).

[3] Tsugawa, S. and Kato, S.: Energy ITS: Another application of vehicular communications, *Communications Magazine*, Vol.48, No.11, pp.120-126, IEEE (2010).

[4] Suzuki, M., Harada, R., Kanda, S. and Shigeno, H.: Overtaking priority management method between platoons and surrounding vehicles considering carbon dioxide emissions (poster), *IEEE Vehicular Networking Conference (VNC)*, pp.238-245 (2011).

[5] Suzuki, M., Tokuda, T. and Shigeno, H.: Overtaking Action Control Method Based On FCFS Policy Considering CO2 Emissions, *The 6th International Conference on Mobile Computing and Ubiquitous Networking*, pp.106-113 (2012).

[6] Stibor, L., Yungpeng, Z. and Reumerman, H.-J.: Evaluation of Communication Distance of Broadcast Messages in a Vehicular Ad-Hoc Network Using IEEE 802.11p, *IEEE Wireless Communications and Networking Conference (WCNC)*, pp.254-257 (2007).

[7] Kanda, S., Uchikawa, A., Harada, R. and Shigeno, H.: Implementation and experiment of multi-modal transmission system for stable communication, *IEEE Vehicular Networking Conference (VNC)*, pp.315-322 (2010).

[8] Raynal, M.: A simple taxonomy for distributed mutual exclusion algorithms, *ACM SIGOPS Operating Systems Review*, Vol.25, pp.47-50 (1991).

[9] Ricart, G. and Agrawala, A.K.: An optimal algorithm for mutual exclusion in computer networks, *Comm. ACM*, Vol.24, pp.9-17 (1981).

[10] Manivannan, D. and Singhal, M.: A Decentralized Token Generation Scheme for Token-Based Mutual Exclusion Algorithms, *INT'L JOURNAL OF COMPUTER SYSTEMS: SCIENCE AND ENGINEERING* (1996).

[11] Swaroop, A. and Singh, A.K.: A Token-Based Group Mutual Exclusion Algorithm for Cellular Wireless Networks, *IEEE India Conference (INDICON)*, pp.1-4 (2009).

[12] Singhal, M.: A Taxonomy of Distributed Mutual Exclusion, *Journal of Parallel and Distributed Computing*, Vol.18, pp.94-101 (1993).

[13] Agrawal, D. and Abbadi, A.E.: An efficient and fault-tolerant solution for distributed mutual exclusion, *ACM Transactions on Computer Systems*, Vol.9, pp.1-20 (1991).

[14] Ohwada, Y., Maeno, T., Kaneda, S., Hisanaga, R. and Takai, M.: Realistic ITS Simulation with Scenargie Sim-

ulator, *IPSJ Symposium Series*, pp.233-234 (2008).

[15] Yoshimura, S., Nishikawa, H. and Moriyasu, S.: Development of Multi-Agent Traffic and Environmental Simulator MATES, *Japan Society for Simulation Technology*, Vol.23, No.3, pp.228-237 (2004).

[16] Oguchi, T., Katakura, M. and Taniguchi, M.: Carbon-dioxide emission model in actual urban road vehicular traffic conditions, *Journal of Infrastructure Planning and Management (JSCE)*, No.695/IV-54, pp.125-136 (2002).

[17] 財団法人日本自動車研究所：平成 22 年度 ITS 通信シミュレーション評価シナリオ (Ver.1.0) (2011).



鈴木 理基 (学生会員)

1982 年生。2006 年慶應義塾大学工学部情報工学科卒業。2008 年同大学大学院理工学研究科博士前期課程修了。2008~2010 年株式会社野村総合研究所在籍。2010 年慶應義塾大学大学院理工学研究科博士後期課程入学。現在、同課程在学中。ITS の研究に従事。



徳田 大誠 (学生会員)

1988 年生。2011 年慶應義塾大学工学部情報工学科卒業。現在、同大学大学院博士前期課程在学中。ITS の研究に従事。



重野 寛 (正会員)

1990 年慶應義塾大学工学部計測工学科卒業。1997 年同大学大学院理工学研究科博士課程修了。現在、同大学理工学部教授。博士 (工学)。情報処理学会学会誌編集委員、同論文誌編集委員、同マルチメディアと分散処理研究会幹事等を歴任。現在、情報処理学会高度交通システム研究会幹事、同モバイルコンピューティングとワイアレス通信研究会運営委員、電子情報通信学会英文論文誌 B 編集委員、同ネットワークシステム研究専門委員会委員。Vice Chair of IEEE ComSoc Asia Pacific Board Technical Affair Committee。ネットワーク・プロトコル、モバイルコンピューティング、ITS 等の研究に従事。著書『コンピュータネットワーク』(オーム社)、『ユビキタスコンピューティング』(オーム社)等。電子情報通信学会、IEEE、ACM 各会員。