

協調隊列自動走行が周辺に与える影響を考慮した 追い越し優先権制御方式

鈴木 理基^{†1} 原 田 亮^{†1}
神 田 翔 平^{†1} 重 野 寛^{†1}

交通状態を改善するシステムのひとつとして、協調自動隊列走行が研究されている。隊列走行では、複数の車両が高密度な車群を形成しながら走行するので、道路上の長い距離を専有しながら走行する。従って、片側二車線の道路において、隊列の前方から低速車両、後方から高速車両が接近すると低速車両が隊列に追い越される動作と高速車両が隊列を追い越す動作が同時に発生し、いずれかの車両が本来意図していたものとは異なる動作を強いられる。本稿ではこの干渉を回避するため、追い越し優先権の制御方式を提案する。token-based アルゴリズムに基づいて高々1台のみが隊列の横を走行することで、不要な加減速による二酸化炭素消費を低減する。シミュレーション評価により、提案アルゴリズムの有効性を示す。

A Passing Priority Control Method Considering Influence of Platoon on Surrounding Vehicles

MASAKI SUZUKI,^{†1} RYO HARADA,^{†1} SHOHEI KANDA^{†1}
and HIROSHI SHIGENO^{†1}

There are a lot of studies about automatic driving cooperative platooning in order to realize energy-efficient ITS system. The platoon consists of multiple vehicles and drive close to each other. It means the platoon occupies a long lane, more than 50 meters. When a vehicle in low speed is coming close to the platoon from forward and another vehicle in high speed is coming from behind, the conflict between those vehicles would occur. In this paper, priority control method is proposed and it is based on the distributed mutual exclusion, especially token-based algorithm. We evaluate the proposed method by computer simulation in terms of the amount of carbon dioxide emission and we show the effectiveness of our proposal.

1. はじめに

ITSの中でも省エネルギー効果の高いシステムの実現を目指す研究のひとつに、協調隊列自動走行がある¹⁾²⁾。隊列走行とは、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走行するものである。隊列走行は様々な利点の獲得を目標としており、そのひとつは車間距離を短く保つことで走行空気抵抗を低減し、エネルギー消費を低減することである。車間距離 4m で4台が隊列を組んで走行した場合に、15%の燃費削減が実現されている³⁾。また、道路の流量を増やすことで渋滞を削減し、交通量を増加させることも狙っている。さらに、前後の車両との相対速度を小さくすることで、衝突時の安全性を保つことを目指している。

しかし、協調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。そのひとつに、追い越し動作が干渉する問題がある。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、隊列走行の本来の目的である交通流を妨げ、当事者同士の燃費効率を阻害するだけでなく、事故を誘発する危険性をふくんでいる。そして、現在検討されている隊列は、ひとつの隊列がおおよそ50メートルもの長さに及ぶことが想定されており、国内の高速道路は片側二車線の区間が多いことを考慮すると、上記の干渉はおよそ頻繁に起こるものと考えられる。

本稿では、この干渉を回避することで燃費効率を改善することを目的とした、追い越し動作の優先権制御方式を提案する。提案方式では隊列のとなりの車線をひとつのクリティカルセクションとする。そして、前述の現象を排他制御問題と捉えることで干渉を回避する。つまり、隊列の前方を走行する低速車両と隊列の後方を走行する高速車両のどちらか一方がクリティカルセクションである車線に進入している時に、他方の車両はその車線上を走行しないように優先権を制御する。ただし本稿では、隊列はひとつの車線を走行することを想定し、燃費効率は、隊列、低速車両、高速車両の総排出燃料を対象とする。

以下、2章では、高速道路上での追い越し動作の干渉問題について説明し、3章では、排他制御方式の既存手法を紹介し、追い越し動作の干渉問題に適用することを説明する。4章で追い越し優先権制御方式を提案し、5章で提案方式をコンピュータシミュレーションによ

^{†1} 慶應義塾大学大学院 理工学研究科 開放環境化学専攻
Graduate School of Science and Technology, Keio University

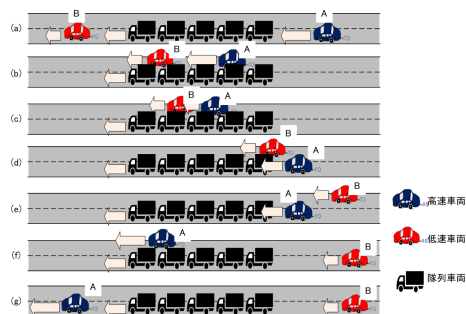


図 1 制御の無い場合の追い越し (ケース 1)
Fig. 1 Taking Over Without Special Control:
Case1

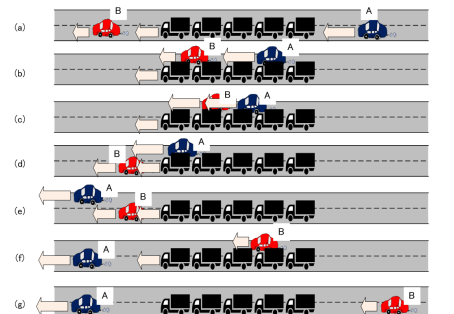


図 2 制御の無い場合の追い越し (ケース 2)
Fig. 2 Taking Over Without Special Control:
Case2

り定量評価する。6 章で結論を述べる。

2. 追い越し動作の干渉問題

ある検討例²⁾では、協調自動隊列走行を高速道路上で実現することを想定している。高速道路では一般車両用レーンの他に、隊列走行する車両は、隊列走行用の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、車車間通信装置を用いて相互に通信できるものとする。隊列内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。

図 1 および 2 に追い越し優先権の制御を行わず、追い越し動作の干渉が発生する場合を示す。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。片側二車線の道路で、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。解消するためには、低速車両が加速するか、高速車両が減速することで、もう一方の車両の速度に合わせなければならない。速度を合わせた二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、二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、改めて速度をもとに戻し、隊列と位置を入れ替える。

この場合、不要な加減速を強いられる車両が存在するので、その車両の燃費効率が悪くなってしまう。さらに、ドライバーに意図していない動作を強いることになるので、交通流も妨げることとなり、安全性にも悪影響が考えられる。つまり、協調隊列自動走行が狙う本

来の目的と全くそぐわない影響を及ぼしてしまう。現在検討されている一例では、隊列は大型車両が 3 台から 8 台程度列をなして走行することが想定されている。つまり、一隊列の長さは 50m 以上なので、上記の干渉が頻繁に発生することが見込まれる。

3. 排他制御

追い越し動作の干渉問題を解決する方法のひとつに排他制御を導入することが考えられる。隊列の横の車線をクリティカルセクションとみなし、車両間で分散処理し、排他制御を行う。一般的に、分散排他制御方式は大きくふたつに分類される⁴⁾。ひとつは permission-based アルゴリズムである。もうひとつは token-based アルゴリズムである。

Ricart らによって提唱された permission-based アルゴリズム⁵⁾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、プロセスは他のプロセスに対して、承認を求める。全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、クォーラムやクォーラムグループを利用して投票によって承認するプロセスを選択する。

token-based アルゴリズムは、ある時間にひとつのプロセスがクリティカルセクションに侵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、排他制御の要件である、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⁶⁾。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。メッセージの送信にマルチキャストを利用することでメッセージ量を低減する手法が提案されている^{7),8)}。

しかし、token-based アルゴリズムの課題はふたつある⁹⁾。ひとつは、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。もうひとつは、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ことである。特にトークンを再発行するアルゴリズムは、再発行する主体を選択するア

ルゴリズムと言い換えることもでき、様々な手法が提案されている¹⁰⁾。そして、分散処理アルゴリズム¹¹⁾が提案されている。

追い越し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全ての車両が承認するべきである。しかし、一台の車両の通信範囲を考慮すると、隊列が車両間の通信を中継する必要がある。また、クリティカルセクションは常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構造としてノードを論理的に構成することが望ましい。以上より、本稿の環境では、隊列をコーディネーターとした、token-based アルゴリズムが適している。

4. 追い越し優先権制御方式

本稿では、前述の干渉を解消するため、追い越し優先権制御方式を提案する。提案方式は token-based アルゴリズムに従って追い越し処理の干渉を解消する。隊列が走行している隣の車線をクリティカルセクションとし、トークンを保持している車両のみが隊列の隣の車線を走行するように制御する。トークンは隊列が主体となって発行および管理する。車両は隊列に接近すると自動的に検知し、隊列にトークンを要求する。トークンの取得後、追い越し動作を実施し、完了するとトークンを隊列に返還する。

4.1 車両接近の検知

定常状態において、隊列走行している車両は自分が所属している隊列のトークンを保持した状態で、隊列の識別子をアナウンスしながら走行している。隊列の識別子には隊列フィルタ^{12),13)}を使用する。隊列フィルタは隊列内の隊列の先頭を走行するマスタ車両によって分散管理されている。したがって、隊列の周囲を走行している車両は隊列の接近を知ることができる。車両は、一定の距離よりも隊列に近づくと、隊列に対してトークンを要求する。

隊列フィルタを用いて隊列を識別しているので、隊列のメンバは、マスタ車両によって管理されている。従って、その場の状況に応じて動的にメンバーの追加および離脱が可能であるので、クリティカルセクションの進入を待機している車両は、隊列に参加した状態で待機する。不要な加減速を避けるだけでなく、待機中の燃料消費を低減する。

4.2 トークン要求時の動作

図3に車両がトークンを要求する動作を示す。トークンを取得するためのリクエストを送信するタイミングはドライバーが行う場合と、システムが自動的に行う場合が考えられるが、本稿ではシステムが自動的にリクエストを送信することを想定する。高速度で走行している車両は、前方車両と一定距離以内に接近すると、前方車両に対して、追い越すためのリ

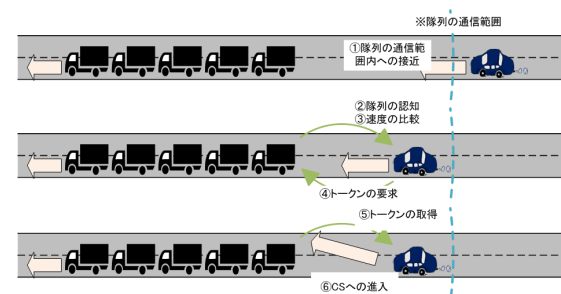


図3 トークン要求時の動作

Fig.3 Outline When Vehicle Request Token

クエストを送信する。後方車両からのリクエストを受信した車両は後方車両と自車両の速度を比較する。後方車両の速度の方が大きい場合には後方車両へトークンを送信する。そして、自車両の速度が後方車両の速度よりも大きい場合には、トークンを送信しない。トークンを受け取った車両は前方を走行する、相対的に速度の遅い車両を追い越す。トークンを持たない車両は走行車線を走行するものとし、後方から速度の速い車両に追い越されるのを待つ。高速車両は車両を追い越した後、所有していたトークンを返還する。

隊列に2台がほぼ同時に接近し、ほぼ同時に隊列に対してトークンを要求する場合、先にリクエストを送信した方が優先されるように制御する。リクエスト送信時にはパケットヘッダにタイムスタンプを挿入し、受信した隊列はタイムスタンプを比較する。タイムスタンプの時刻が早い車両にトークンを送信し、車両はクリティカルセクションに進入する。遅かった方の車両はトークンを待機し、隊列に速度を合わせて走行する。先にトークンを取得した車両は隊列を追い越した後、トークンを隊列走行している車両に送信する。トークンを待機している車両は、定期的に関リクエストを送信している。トークンが隊列に戻り次第、隊列から待機車両へトークンが送信される。待機車両はトークンを受信した後、クリティカルセクションへ進入し、隊列との追い越し処理を実施する。以上のように、隊列と低速車両、高速車両間での、ふたつの追い越し処理が干渉することを回避する。

4.3 排他性の保証（トークン損失時の処理）

本提案は token-based アルゴリズムを採用しているので、トークンが存在する限りは、簡単な手続きで排他性を保証する。しかし、トークンが損失した際には特別な手続きを経てトークンを再度発行する必要がある。本稿では、トークンを発行する車両である隊列のマス

タ車両がトークンを再発行する。

既存の token-based アルゴリズムでは、トークンの損失を検出した時点で、リーダー選出のアルゴリズムに従って、トークンを再発行するノードを選択する。選択されたノードは、全てのノード、あるいはクリティカルセクションに進入する可能性のある全てのノードから合意を得ることで、損失したトークンを無効とし、新しいトークンを発行する。

本提案では、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。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と、トークンを保持しているべき車両の所在を検索する。通信範囲はクリティカルセクションよりも十分に広いので、隊列車両の通信範囲に車両が見つからない場合は、協調範囲内にトークンは存在しないと判断する。その後、隊列車両はトークンを再発行する。従って、トークンは一度にひとつしか存在しないので、排他性が保証される。

ただし、本提案アルゴリズムは、トークンを持たない車両がクリティカルセクションに入らないことが前提とされなければならない。つまり、通信機器の故障等、通信頻度に比べて十分に長い時間、車両間で通信できない場合には保証されない。例えば、追い越し処理中の車両の通信機が故障した場合、隊列車両や周辺車両からはトークンが失われたように判断される。しかし、現実には車両は追い越し処理中なので、トークンを再発行してしまうと、クリティカルセクションに複数台の車両が進入することが可能となる。従ってこの前提が保たれない場合には、トークン損失時に、センシングやドライバーの視覚等、通信以外の方法で車両が周囲を走行していないことを確認する必要がある。

4.4 公平性と飢餓問題

提案手法では、一度にクリティカルセクションに進入する車両は、たとえ同じ方向に走行していたとしても、1台のみとする。図4に、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が接近する例を示す。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、タイムスタンプの早い方から順番にクリティカルセクションに進入するので、クリティカルセクションに進入する順番もA,B,Cである。

ここで、提案手法とは異なるアルゴリズムも考えられる。最初にクリティカルセクションに進入する車両はAである。2番目にクリティカルセクションに進入する車両はCであるとする。道路上において、隊列との相対的な運動方向は2次元であるので、A,Cの運動方

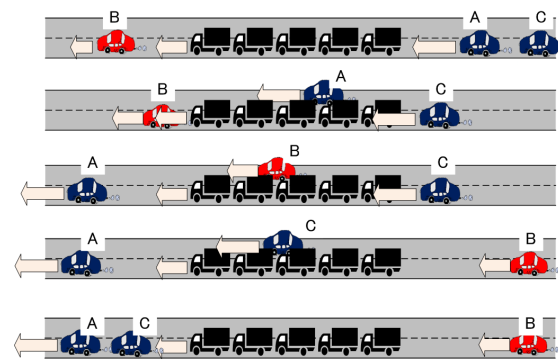


図4 3台が同時に接近する場合

Fig.4 3 Vehicles Coming Close To Platoon At The Same Time

向が同じなので、AとCは互いに干渉することなく、クリティカルセクションでの走行を進めることができる。従って、リソースを効率よく使用するので、直感的に効率の良いアルゴリズムといえる。

しかし、要求を出した順番通りにクリティカルセクションに進入していないので、公平性が保たれているとはいえない。さらに、車両Cに続いて無限に、高速車両が後方から隊列に接近した場合、車両Bは永久にクリティカルセクションに進入できず、隊列の前方を走行しなければならない。以上のことから、本提案では一度にクリティカルセクションに進入する車両は、たとえ同じ方向に走行していたとしても、1台のみとする。そして、要求を出した順番通りにクリティカルセクションに進入することで、公平性を保ち、同時に飢餓問題を回避する。

4.5 デッドロックの保証（クリティカルセクションのマージン）

本提案アルゴリズムでは、デッドロックを回避するために、例外として、隊列間の距離を十分に広くとらなければならない。提案手法では、token-based アルゴリズムを用いているので、クリティカルセクションがひとつの場合にはデッドロックは発生しない。従って、複数のクリティカルセクションを同時に要求しないように制御することで、デッドロックを回避することができる。高速道路上を走行する車両は2次元方向に運動するので、3つ以上のクリティカルセクションを同時に要求することは起こり得ない。つまり、上記のように2つのクリティカルセクションを同時に要求しないように制御することでデッドロックを回避

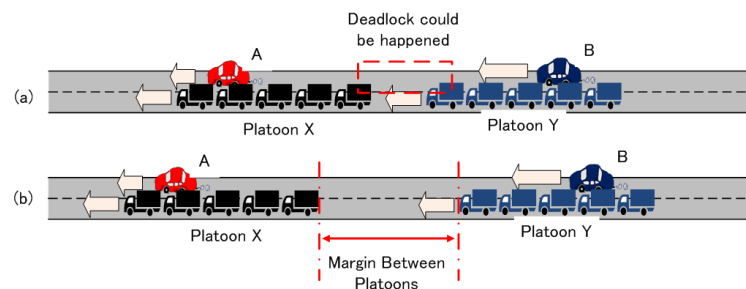


図 5 隊列間の距離
Fig. 5 Distance Margin Between Platoons

する。

図 5 に隊列間の距離を示す。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 Y 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 X 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 を追い越しているため、車両 B は隊列 Y 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。つまり、デッドロックが発生してしまう。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ことができる。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ことができる。以上より、デッドロックを回避する。

また、ふたつの隊列が道路上で合流する場合が考えられる。その際にはどちらの隊列のクリティカルセクションにも車両が進入していないことが条件となる。

4.6 提案方式を用いた際の動作

提案アルゴリズムを用いることで、処理 1：隊列が前方を走行する低速車両を追い越す処理と、処理 2：後方を走行する高速車両が隊列 Y を追い越す処理を分離することができる。

図 6 に処理 1 を先に実施する例を示す。低速車両と高速車両が追い越しに必要なトークン

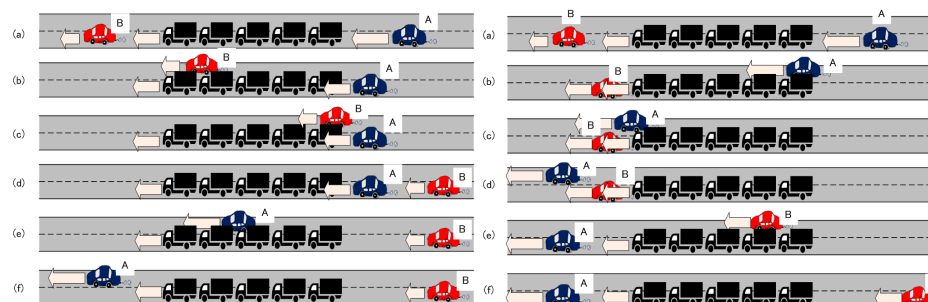


図 6 優先権を制御する場合の追い越し
(低速車両が先に優先権を取得)

Fig. 6 Taking Over With Special Control: Case1

図 7 優先権を制御する場合の追い越し
(高速車両が先に優先権を取得)

Fig. 7 Taking Over With Special Control: Case2

を要求する。低速車両の方が先にトークンを要求している場合、低速車両がトークンを取得し、クリティカルセクションに進入する。高速車両はトークンを取得するまで、クリティカルセクションに進入せず、待機する。つまり、処理 1 を実施している間、処理 2 を待機している高速車両は隊列に速度を合わせ、隊列の後方を走行する。処理 1 を終えた低速車両がトークンを返却した後、高速車両がトークンを取得し、クリティカルセクションに進入する。高速車両が必要以上の加減速をしていないことを示している。

図 7 に処理 2 を先に実施する例を示す。前述の例と同様、低速車両と高速車両が追い越しに必要なトークンを要求する。高速車両の方が先にトークンを要求している場合、高速車両がトークンを取得し、クリティカルセクションに進入する。低速車両はトークンを取得するまで、待機する。処理 2 を終えた高速車両がトークンを返却した後、高速車両がトークンを取得し、クリティカルセクションに進入する。前述の例と同様に、低速車両が必要以上の加減速をしていない。

5. 定量評価

本提案の有効性を示すため、交通流シミュレータ VISSIM version 5.00-02¹⁴⁾¹⁵⁾ を用いたコンピュータシミュレーションにより評価する。二酸化炭素排出量の算出には次式 1 に表される大口らによるモデル¹⁶⁾を用いる。

表 1 シミュレーション条件
Table 1 Simulation Conditions

道路モデル	道路長	1.00km
	道路形状	直線
	車線数	2 車線 (一方通行)
	隊列走行指定レーン	1 レーン
	非隊列車両用レーン	2 レーン
車両モデル	車線幅員	5.4m
	隊列内車両台数	3 台/隊列
	隊列車両間距離	4 m
	大型車両全長	12 m
	普通自動車全長	4.7 m
	隊列車両目標速度	80 km/h
	低速車両目標速度	60 km/h
	高速車両目標速度	100 km/h

$$E = 0.3K_C T + 0.028K_C D + 0.056K_C \sum_{k=1}^K \delta_k (v_k^2 - v_{k-1}^2) \quad (1)$$

但し、 E は二酸化炭素排出量、 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、 δ_k は加速時に 1、その他の場合に 0 となるダミー変数、 v は車両速度とする。

その他のシミュレーション条件を表 1 に示す。追い越しが発生しない場合と追い越しが発生する場合について評価を行う。特に追い越しが発生する場合では、優先権を制御せずに追い越しが干渉する場合、優先権を制御して前方車両が先に優先権を取得する場合と後方車両が先に優先権を取得する場合の 3 通りのシナリオで評価する。各シナリオで車両発生タイミング等を左右する乱数のシードを変えて 3 回ずつ試行し、その平均値をそれぞれ求める。道路を走行する車両は一般車両と大型車両の二種で、隊列は大型車両のみで構成されるものとし、指定されたレーンを走行する。そして、隊列を構成しない車両は全ての車線を走行する。車両は道路の端点から流入し、可能な限り目標速度で走行し、反対側の端点から流出する。隊列内の車間距離は 4m とするが、隊列走行中の燃費削減率は考慮せず、大型車と普通自動車とを区別しないものとする。

図 8、9 に追い越しが発生しない場合の各車両の位置と二酸化炭素排出量の推移を示す。測定を開始した時点で、隊列を起点として高速車両が前方、低速車両が後方を走行している。測定に使用した 1km 区間での全ての車両の二酸化炭素排出量の合計は 231.20g-C であった。

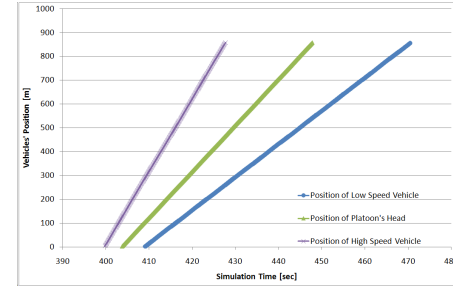


図 8 追い越しが発生しない場合の各車両の位置の推移
Fig. 8 Positions of Vehicles In Case Taking Over Does Not Happen

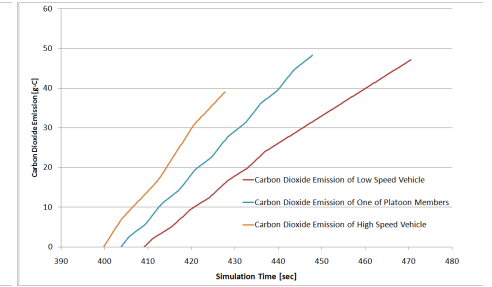


図 9 追い越しが発生しない場合の各車両の二酸化炭素排出量の推移
Fig. 9 Amount Of Carbon Dioxide Emissions In Case Taking Over Does Not Happen

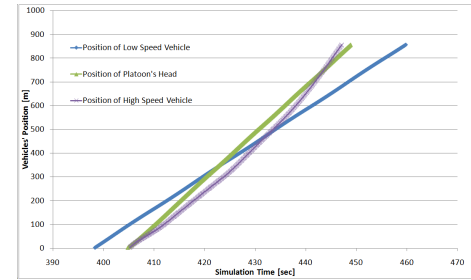


図 10 優先権を制御しない場合の各車両の位置の推移
Fig. 10 Positions of Vehicles In Case Taking Over Processes Without Proposal Method

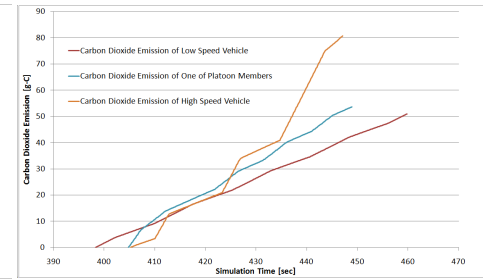


図 11 優先権を制御しない場合の各車両の二酸化炭素排出量の推移
Fig. 11 Amount Of Carbon Dioxide Emissions In Case Taking Over Processes Without Proposal Method

追い越しが発生しない場合、各車両は直線の高速度道路を加減速せずに走行するので、ほとんど無駄の無い量の二酸化炭素を排出する。

図 10、11 に優先権を制御せず、追い越しの干渉が発生する場合における、各車両の位置と二酸化炭素排出量の推移を示す。追い越しの干渉発生時には後方車両が減速し、低速車両の追い越しを先に処理するように干渉を解消した。測定に使用した 1km 区間での全ての車両の二酸化炭素排出量の合計は 292.46g-C であった。これは理想的な状況と比べて 26.50%増加している。高速車両の二酸化炭素排出量は増加し、理想的な状況と比べ、106.91%増加

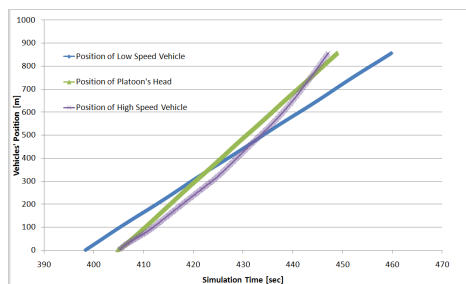


図 12 優先権を制御する場合の各車両の位置の推移 (ケース 1)

Fig. 12 Positions of Vehicles In Case Taking Over Processes With Proposal Algorithm

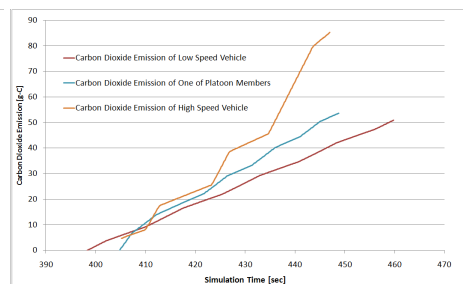


図 13 優先権を制御する場合の各車両の二酸化炭素排出量の推移 (ケース 1)

Fig. 13 Amount Of Carbon Dioxide Emissions In Case Taking Over Processes With Proposal Algorithm

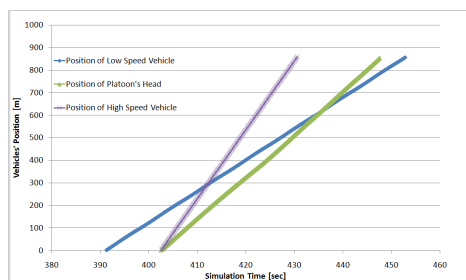


図 14 優先権を制御する場合の各車両の位置の推移 (ケース 2)

Fig. 14 Positions of Vehicles In Case Taking Over Processes With Proposal Algorithm

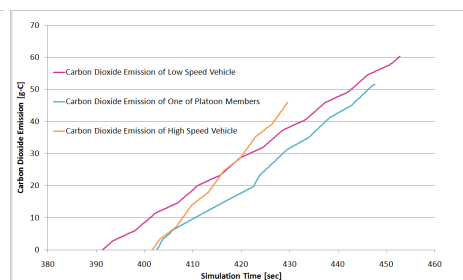


図 15 優先権を制御する場合の各車両の二酸化炭素排出量の推移 (ケース 2)

Fig. 15 Amount Of Carbon Dioxide Emissions In Case Taking Over Processes With Proposal Algorithm

した。

図 12, 13, 14, 15 に、追い越しの干渉が発生する場合における、優先権を制御する場合の各車両の位置と二酸化炭素排出量の推移を示す。図 12, 13 は低速車両が先に優先権を取得した場合 (ケース 1) を示し、図 14, 15 は高速車両が先に優先権を取得した場合 (ケース 2) を示す。

ケース 1 では、測定に使用した 1km 区間での全ての車両の二酸化炭素排出量の合計は 273.47g-C であった。これは優先権を制御しない場合と比べて 6.50%改善している。最初に

優先権を取得しなかった高速車両の二酸化炭素排出量は、優先権を制御しない場合と比べ、15.14%改善しているが、理想的な状況と比べて 75.58%増加している。トークンを待機するために、隊列に速度を合わせて走行する際に、目標速度から減速しているため、再度加速することによって燃料消費のロスが生じているものと考えられる。

ケース 2 では、測定に使用した 1km 区間での全ての車両の二酸化炭素排出量の合計は 273.68g-C であった。これは優先権を制御しない場合と比べて 6.42%改善している。そして、低速車両が先に優先権を取得した場合と比べると、0.08%二酸化炭素排出量は増加しているが、有意差は無いといえる。最初に優先権を取得しなかった低速車両の二酸化炭素排出量に着目すると、優先権を制御しない場合と比べて 16.06%増加している。しかし、高速車両の二酸化炭素排出量は 24.63%改善しており、合計で改善したといえる。

以上のシミュレーション結果から、追い越し優先権を制御することによって、二酸化炭素排出量を改善する場合があることを示した。隊列走行中の燃費削減率を考慮した場合には、さらに大きな改善率を示すことが見込まれる。

6. おわりに

省エネルギー効果の高い ITS の実用化のため、協調隊列自動走行が研究されている。そして、協調隊列自動走行ではあらかじめ考慮すべき問題があり、そのひとつが追い越しの干渉である。ひとつの隊列あたり 5 台程度の車両で構成されることが想定されており、隊列の長さは 50m 以上に及ぶ。隊列の前方から低速車両が接近し、同時に後方から高速車両が接近する場合に、低速車両と高速車両の追い越し動作が干渉してしまう。動作が干渉することで、低速車両や高速車両は不要な加減速を強いられるので、通常時に走行するよりも、二酸化炭素排出量は増加してしまう。

本稿では、この追い越し動作の干渉を回避するため、追い越し優先権の制御方式を提案した。隊列の隣のレーンをクリティカルセクションとして、token-based アルゴリズムに基づいて優先権を制御する。一度に一台のみが追い越し動作を処理することで、低速車両や高速車両の不要な加減速を無くした。トークン損失時には、隊列車両のみがトークンを再発行することでクリティカルセクションの排他性を保証する。そして、トークン要求時にはパケットヘッダにタイムスタンプを挿入し、早く要求した車両から 1 台ずつ順にクリティカルセクションに進入することで、公平性を保ち、飢餓問題を回避する。

本稿の有効性を示すため、交通流シミュレータを用いたシミュレーション評価を行った。シミュレーションの結果、追い越し動作の干渉が発生する場合には、提案手法は優先権制

御を行わない場合と比較して、二酸化炭素排出量の観点から 6.50%改善していることを示した。

今後の課題として、無線通信によりトークンを授受することで、特に通信状態が悪い場合に、提案手法が追いつきの干渉に対して与える影響、そしてその環境下においても車の挙動を制御する方法について検証、検討する。さらには、二酸化炭素排出モデルも、前方車両との距離や空気抵抗を考慮したモデルを導入することを検討している。

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) 独立行政法人新エネルギー・産業技術総合開発機構：エネルギー ITS 推進事業, <http://www.nedo.go.jp/activities/portal/p08018.html> (2011).
- 3) (財)日本自動車研究所：自動運転・隊列走行技術の研究開発, NEDO 省エネルギー技術フォーラム (2010).
- 4) Raynal, M.: A simple taxonomy for distributed mutual exclusion algorithms, *SIGOPS Oper. Syst. Rev.*, Vol.25, pp.47–50 (1991).
- 5) Ricart, G. and Agrawala, A.K.: An optimal algorithm for mutual exclusion in computer networks, *Commun. ACM*, Vol.24, pp.9–17 (1981).
- 6) Swaroop, A. and Singh, A.: A Token-Based Group Mutual Exclusion Algorithm for Cellular Wireless Networks, *India Conference (INDICON), 2009 Annual IEEE*, pp.1–4 (2009).
- 7) Ortiz, A., Thiebolt, F., Jorda, J. and M'zoughi, A.: How to use multicast in distributed mutual exclusion algorithms for grid file systems, *High Performance Computing Simulation, 2009. HPCS '09. International Conference on*, pp.122–130 (2009).
- 8) Naimi, M., Trehel, M. and Arnold, A.: A log (N) distributed mutual exclusion algorithm based on path reversal, *J. Parallel Distrib. Comput.*, Vol.34, pp.1–13 (1996).
- 9) Singhal, M.: A taxonomy of distributed mutual exclusion, *J. Parallel Distrib. Comput.*, Vol.18, pp.94–101 (1993).
- 10) Agrawal, D. and ElAbbad, A.: An efficient and fault-tolerant solution for distributed mutual exclusion, *ACM Trans. Comput. Syst.*, Vol.9, pp.1–20 (1991).
- 11) 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*, Vol.11 (1996).
- 12) 内川亜美, 羽鳥遼平, 黒木智也, 重野 寛：隊列走行のための動的な隊列管理手法の提案, 第 37 回 ITS 研究会, Vol.67, pp.1–6 (2009).
- 13) Uchikawa, A., Hatori, R., Kuroki, T., Shigeno, H. : Filter Multicast : A Dynamic Platooning Management Method, 2nd IEEE Intelligent Vehicular Communications System Workshop (IVCS '10) (2010).
- 14) PTV Japan: ミクロシミュレーション・交通需要予測 PTV Vision VISSIM, <http://www.ptv-vision.jp/web/vissim.html> (2011).
- 15) Visual Solutions, Inc.: The smarter faster way for model based system development — VisSim, <http://www.vissim.com/> (2011).
- 16) 谷口正明大口敬：都市部道路交通における自動車の二酸化炭素排出量推定モデル, 土木学会論文集, No.695/IV-54, pp.125–136 (2002).