

複数車両を参照する数理モデルによる交差点を左折する 車両隊列シミュレーション

VEHICLE PLATOON SIMULATION TURNING RIGHT AT INTERSECTION ACCORDING TO MULTIPLE LEADER VEHICLE-FOLLOWING MODEL

坂本 浩規¹⁾, 玉城 龍洋²⁾, 北 栄輔³⁾

Hiroki SAKAMOTO, Tatsuhiro TAMAKI and Eisuke KITA

- 1) 名古屋大学大学院情報科学研究科 (〒464-8601 名古屋市千種区不老町 1)
2) 沖縄工業高等専門学校メディア情報工学科 (〒905-2171 沖縄県名護市辺野古 905)
3) 名古屋大学大学院情報科学研究科 (〒464-8601 名古屋市千種区不老町 1, E-mail: kita@i.nagoya-u.ac.jp)

Vehicle Platoon is the method of platooning the vehicles with a short vehicle-to-vehicle distance. It is effective to increase the traffic amount safely without constructing new roads. In the previous study, the mathematical model for controlling the vehicle velocity was defined by the car-following model. A theoretical study revealed that each vehicle in a platoon could control its velocity from the information of only the nearest frontal vehicle and the lead vehicle of the platoon. When a platoon turns left at a corner, the lead vehicle in the platoon changes in order, so the following vehicles must change the lead vehicle they should refer to. In this study, model parameters are determined by numerical simulation and then, their validity is discussed through experiments.

Key Words: Car-Following Model, Vehicle Platoon, Simulation, Experiment

1. 緒論

交通渋滞は物流コストの増大をもたらすとともに、騒音や空気汚染などの原因ともなっている。様々な分野の研究者が、交通渋滞の解決方法に取り組んでいる。それらの一つに、高度道路交通システム (Intelligent Transportation Systems, ITS) の分野における隊列走行がある⁽¹⁾。

隊列走行においては、短い車間距離において、複数の車両が一行で走行する。車間距離を詰めることで交通容量を増大できるとともに、追従車両の空気抵抗が低減され、省エネルギー効果も見込まれている。しかし、安全な隊列走行を実現するためには、各種センサーなどを用いた車両の速度と行動の自動制御システムが必要となる。

本研究では、隊列走行する車両の速度制御について述べる。制御モデルは、いわゆる車両追従モデルに基づいている。車両追従モデルは、前方車両との車間距離や速度差などから自車両の速度または加速度を制御するモデルである^(2, 3, 4)。これらのモデルでは直前方車両 1 台だけを参照する機会が多いのに対して、複数の前方車両を参照するモデルも提案されている^(5, 6)。例えば、牛田⁽⁵⁾は、前方の複数台の車両との

速度差を用いた多台参照 Chandler モデルに基づき速度制御を適用している。朝比奈⁽⁶⁾は、直前方車両との車間距離と隊列の先頭車両との速度差を用いて速度制御を行っている。いずれにおいても、自車両のすぐ前方を走行する車両と隊列の先頭を走行する車両の 2 台を参照するモデルが有効であるとしている。

ところで、一般道を走行する場合、隊列の合流や分離、さらに、交差点の右左折において隊列の組み替えが起こり、そのたびに隊列の先頭車両が入れ替わることになる。本研究では、そのもっとも簡単な場合として、車両隊列が交差点で左折する場合を考える。交差点を左折するとき、隊列は左折した車両とその直後の車両で 2 つの隊列に分かれることになる。それゆえ、隊列の車両が左折するたびに、後続車両にとっての先頭車両が入れ替わり、速度制御則が更新されるようにする。数値シミュレーションとロボット車両⁽⁷⁾を用いた実験を通して、モデルの影響について検討する。

本論文は、以下のように構成されている。本節は緒論として、研究目的について述べた。第 2 節では、本研究で用いる速度の制御モデルについて述べる。第 3 節では、計算機シミュレーション結果を示し、第 4 節では、LEGO MINDSTORMS NXT による実験結果隊列走行実験について述べる。第 5 節

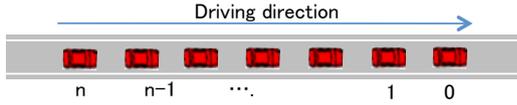


図1 n 台の車両からなる車両隊列

では、得られた結論についてまとめる。

2. 車両追従モデル

2.1. Chandler モデル

一般に、車両追従モデルでは、車両の速度や加速度を他車両との車間距離や速度差などの情報を用いて制御する。Chandler らの提案したモデルは次式で与えられる⁽²⁾。

$$\ddot{x}_n(t + \Delta t) = a(\dot{x}_{n-1}(t) - \dot{x}_n(t)) \quad (1)$$

ここで、 $x_n(t)$ は時刻 t における車両 n の位置、 Δt は応答の遅れ時間、 a は車両 n の直前方車両 $n-1$ に対する感度を表す(図1)。また、 \dot{x} 、 \ddot{x} は、それぞれ車両の速度と加速を示す。

Chandler モデルは直前方を走行する車両との速度差だけを用いるが、これを複数の前方車両を参照するモデルに拡張すると次式となる。

$$\ddot{x}_n(t + \Delta t) = \sum_{j=1}^m a_j(\dot{x}_{n-j}(t) - \dot{x}_n(t)) \quad (2)$$

ここで、 x_{n-j} は車両 $n-j$ の位置、 a_j は車両 $n-j$ に対する車両 n の感度、 m は車両 n の参照する先行車両の数、 Δt は遅れ時間を表す。このモデルでは、加速度 \ddot{x}_n は自車両の速度と各先行車両の速度の差 $\dot{x}_{n-j}(t) - \dot{x}_n(t)$ によって制御される。パラメータ m の値は、Chandler モデルでは $m = 1$ 、Bexelius モデル⁽⁸⁾ では $m = 2$ 、Wakita モデル⁽⁹⁾ では $m > 1$ となる。

2.2. 多台参照 Chandler モデルの安定解析

本研究では車両4台で構成される隊列を考える。先頭より、1番目の車両を先頭車両とし、つづいて、第1追従車両、第2追従車両、第3追従車両とする。先頭車両に生じた速度変化の影響は、後方車両に伝わるに従って小さくなる。このことに基づいて、Chandler は安定条件を導いている。牛田は、複数車両を参照する Chandler モデルについて、安定性解析を行い以下の結果を得ている⁽⁵⁾。

2.2.1. 第1追従車両の安定性解析結果

第1追従車両は先頭車両の直後を走行するので、先頭車両だけを参照する。遅れ時間を $\Delta t = 1$ とすると、安定条件は以下ようになる。

$$0 \leq a_1 \leq \frac{1}{2} \quad (3)$$

この結果より、1台追従車両の感度の最大値は $a_1 = 1/2$ となる。

2.2.2. 第2追従車両の安定性解析結果

第2追従車両は先頭車両と第1追従車両を参照することができる。遅れ時間を $\Delta t = 1$ とすると、安定条件は以下ようになる。

$$0 \leq a_2 \leq \frac{(1 - 2a_1) + \sqrt{1 - 2a_1}}{4} \quad (4)$$

表1 前方車両に対する運転者の感度

追従車両	感度
1st	$a_1 = \frac{1}{2}$
2nd	$a_1 = \frac{3}{8}, a_2 = \frac{3}{16}$
3rd	$a_1 = \frac{1}{2}, a_2 = 0, a_3 = \frac{1}{6}$

2台の前方車両に対する運転者の感度の和は次式で定義される。

$$b_2 \equiv a_1 + a_2 \leq \frac{(1 - 2a_1) + \sqrt{1 - 2a_1}}{4} \quad (5)$$

感度の和 b_2 は、 $a_1 = 3/8$ と $a_2 = 3/16$ のとき、最大値 $(b_2)_{\max} = 9/16$ となる。

2.2.3. 第3追従車両の安定性解析結果

第3追従車両は先頭車両と2台の追従車両を参照することができる。遅れ時間を $\Delta t = 1$ とすると、安定条件は以下ようになる。

$$0 \leq a_1 + a_2 \equiv b_2 \leq \frac{9}{16} \quad (6)$$

$$0 \leq a_3 \leq \frac{(3 - 4a_1 - 8a_2) + \sqrt{9 - 16(a_1 + a_2)}}{12} \quad (7)$$

前方車両に対する運転者の感度の和は次式で定義される。

$$b_3 \equiv a_1 + a_2 + a_3 \leq \frac{(3 - 4a_1 - 8a_2) + \sqrt{9 - 16(a_1 + a_2)}}{12} \quad (8)$$

感度の和 b_3 は、 $a_1 = 1/2$ 、 $a_2 = 0$ 、 $a_3 = 1/6$ のとき、最大値 $(b_3)_{\max} = 2/3$ とする。

各追従車両において前方車両に対する運転者の感度の和が最大となるときの運転者の感度 a_1 、 a_2 、 a_3 をまとめると表1となる。第3追従車両の場合、前方車両が3台いるにもかかわらず参照すべき車両は2台だけでよいことになる。

3. 計算機シミュレーション

3.1. 計算機シミュレーション条件

車両隊列は4台の車両からなる。先頭より先頭車両、第1追従車両、第2追従車両、第3追従車両であり、それぞれ Vehicle 0, Vehicle 1, Vehicle 2, Vehicle 3 とする。1タイムステップは実時間では0.1(s)とする。初期状態では、各車両とその直前方車両の車間距離を全て等しく20(m)とし、車両の初期速度を18(m/s)と設定する。時刻 $t = 5$ (s)において、先頭車両は急減速して停止し、10秒間停止した後、時刻 $t = 15$ (s)において再び速度18(m/s)に加速する。

第2追従車両と第3追従車両の速度制御には、次に述べるモデル1とモデル2を用いて、それらの結果を比較する。直前方車両を参照する Chandler モデルをモデル1、隊列の先頭車両と自車両の直前方車両の2台を参照する Chandler モデルをモデル2とする。ただし、第1追従車両は前方車両が1台しか存在しないので、モデル2においても、第1追従車両はモデル1に従うこととする。感度は表1に示すように与える。

3.2. 計算機シミュレーション結果

モデル1による結果を図2と図3に、モデル2による結果を図4と図5に示す。図2と図4は各車両の速度変化図、

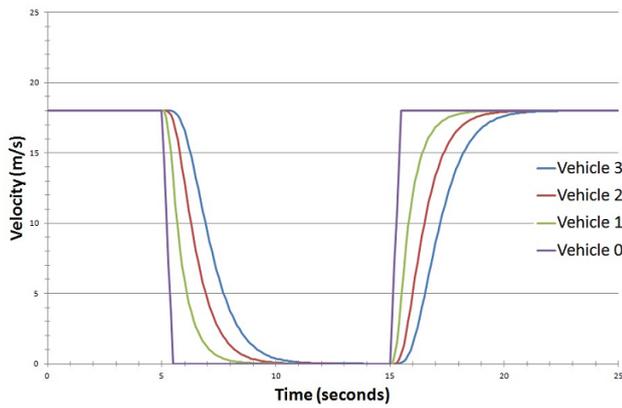


図 2 車両速度の数値シミュレーション (モデル 1)

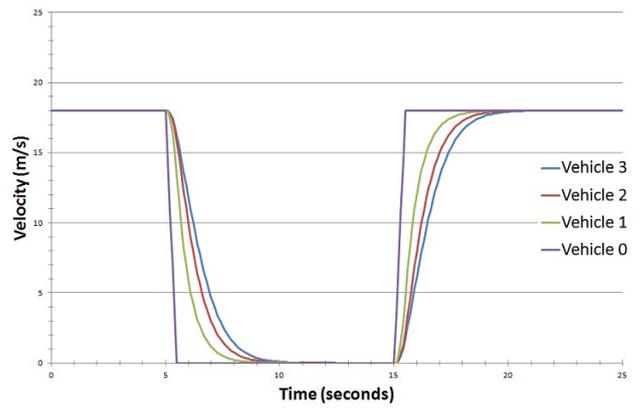


図 4 車両速度の数値シミュレーション (モデル 2)

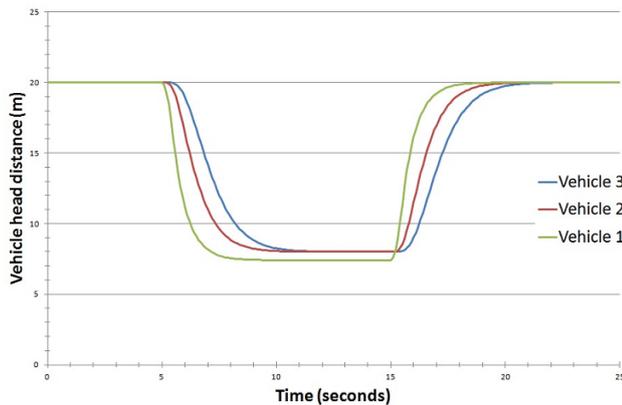


図 3 車間距離の数値シミュレーション (モデル 1)

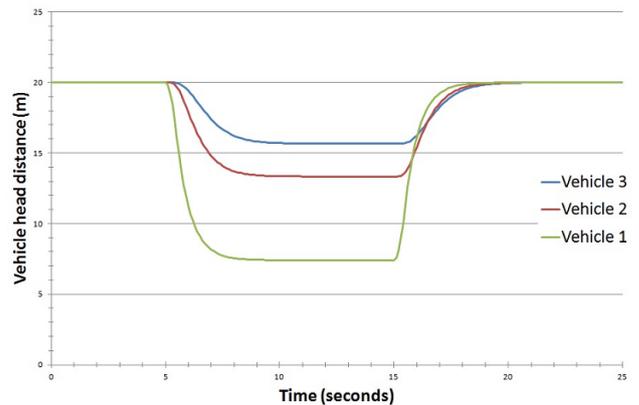


図 5 車間距離の数値シミュレーション (モデル 2)

図 3 と図 5 は各車両の車間距離図である。横軸はシミュレーション時間、縦軸は車両速度または車間距離を示す。車両にはラベル Vehicle 0, Vehicle 1, Vehicle 2, Vehicle 3 を付与し、それらは先頭車両、第 1 追従車両、第 2 追従車両、第 3 追従車両を示す。

最後尾の第 3 追従車両が停止するのは、モデル 1 において 11.8(s) であるのに対して、モデル 2 では 10.8(s) である。また、第 3 追従車両の速度が元の速度に戻るの、モデル 1 では 21.6(s) であるのに対して、モデル 2 では 20.4(s) である。モデル 1 よりもモデル 2 の方が第 3 追従車両の減速が早く始まっており、また早く元の速度に回復できていることがわかる。したがって、隊列走行時において、モデル 2 による速度制御アルゴリズムを適用した方が、車間距離の変動をより小さくした状態で走行できると考えられる。

4. 隊列走行実験

4.1. 実験条件

第 3 節の結果より、直前方車両だけを参照するモデルよりも直前方車両と先頭車両の 2 台参照モデルの方が車間距離の変動を小さくできる。しかし、先頭車両として参照すべき車両は交通状況によって異なっており、隊列として協調行動している車両群の先頭の車両と判断すべきである。

本研究では、隊列走行する車両群が交差点を左折する場合



図 6 隊列走行実験

を考える。本研究では、LEGO MINDSTORMS NXT を用いて実験車両を作成する。実験の様子を図 6 に示す。車両は、写真右後方より走行を開始し、図 6 にある黒い線に沿って走行し、黄色のところで左折を開始する。左折を開始する車両は、後続車両とは異なる行動を取ることになるので、左折を開始する車両と後続する車両の間で隊列が分かれることになる。そこで、後続する車両は、常に左折する車両の直後の車両を隊列の先頭と判断して速度制御する。プログラミング言語として Not eXactly C (NXC)^(10, 12) を用いる。また統合開発環境として、NXC プログラミングの開発環境として提

表 2 停止と再加速の時間

Model	1st stop	1st reacceleration	2nd stop	2nd reacceleration	3rd stop
Model 1	2.776	2.923	3.364	3.234	2.613
Model 2	1.706	3.449	2.535	3.832	2.053

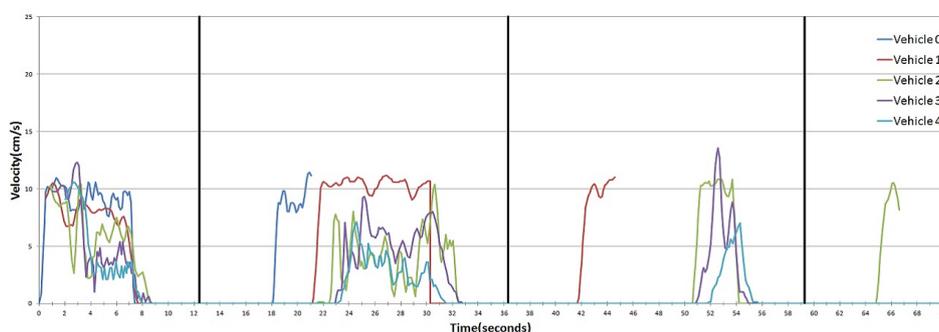


図 7 車両の速度変動 (モデル 1)

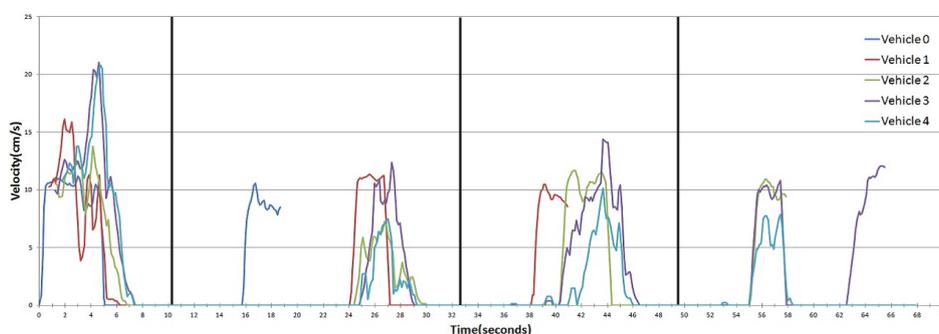


図 8 車両の速度変動 (モデル 2)

供されている Brick Command Center(BricxCC)⁽¹¹⁾ を使用する。

4.2. 速度制御アルゴリズム

4.2.1. モデル 1 の場合

モデル 1 は、1 台の前方車両を参照する Chandler モデルであり、式 (1) で与えられる。式 (1) における速度差の項 $\dot{x}_{n-1}(t) - \dot{x}_n(t)$ は車間距離の差分として、次式で近似的に求めることとする。

$$\Delta \dot{x}_{n-1}(t) \simeq \frac{\Delta x_{n-1}(t) - \Delta x_{n-1}(t - \Delta t)}{\Delta t} \quad (9)$$

ここで、 $\Delta x_{n-1}(t)$ は時刻 t における車両 $n-1$ と車両 n の車間距離を示す。

$$\Delta x_{n-1}(t) = x_{n-1}(t) - x_n(t) \quad (10)$$

式 (9) を用いるためには、直前方車両との車間距離を測定する必要がある。本実験では、LEGO MINDSTORMS NXT の有する超音波センサを用いて直前方車両との車間距離を測定する。

4.2.2. モデル 2 の場合

モデル 2 では、隊列の先頭車両と、自車両の直前方車両の 2 台を参照し、これら 2 台の情報で速度を制御する 2 台参照型 Chandler モデルであり、式 (2) で与えられる。式 (2) に

含まれる変数 $\dot{x}_0(t)$ の値は隊列の先頭車両であり、Bluetooth 通信を利用して得る。LEGO MINDSTORMS NXT では、Bluetooth 通信において、1 台の発信車両と 3 台までの受信車両を設定できる。そこで、先頭車両を発信車両、それ以外の追従車両を受信車両とする。

また、式 (2) の右辺第 1 項の $\dot{x}_{n-1}(t) - \dot{x}_n(t)$ は通常の Chandler モデルの場合と同様に、車間距離の差分として式 (9) で近似的に求める。モデル 2 においても、LEGO MINDSTORMS NXT の有する超音波センサを用いて直前方車両との車間距離を測定する。

4.3. 実験結果

モデル 1、モデル 2 の車両の速度変化図を、それぞれ図 7 と図 8 に示す。ラベル Vehicle 0, Vehicle 1, Vehicle 2, Vehicle 3, Vehicle 4 はそれぞれ先頭車両、第 1 追従車両、第 2 追従車両、第 3 追従車両、第 4 追従車両を表している。横軸に時間経過、縦軸に車両速度をとる。各図に記載された黒い縦線は、前方車両が左折行動に入ったことから、隊列の先頭車両が、左折行動に入った車両からその次の車両に入れ替わった時点を表している。表 2 では、先頭車両が停止するための減速を開始してからすべての追従車両が停止するまでの所要時間 (stop) と、新しい隊列が編成され先頭車両が再加速を開始してからすべての追従車両が再加速を完了するまでの所要時

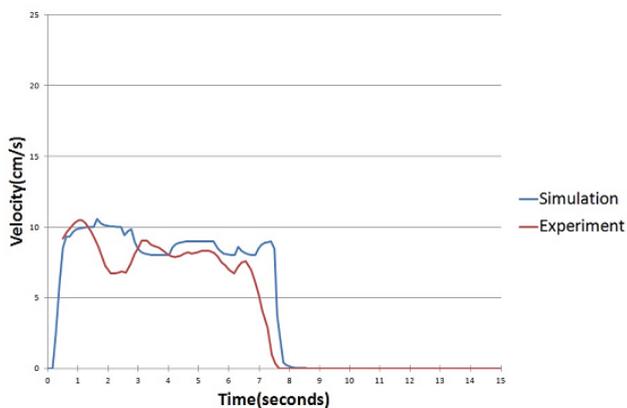


図 9 Vehicle 1 の速度変動の比較 (モデル 1)

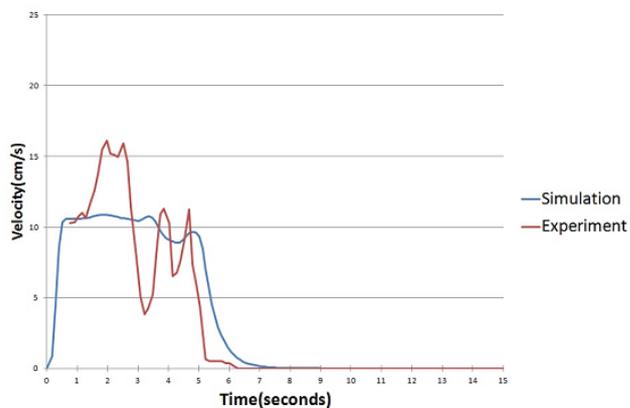


図 12 Vehicle 1 の速度変動の比較 (モデル 2)

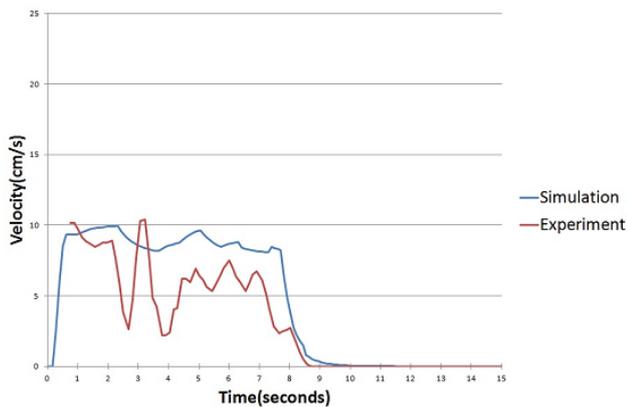


図 10 Vehicle 2 の速度変動の比較 (モデル 1)

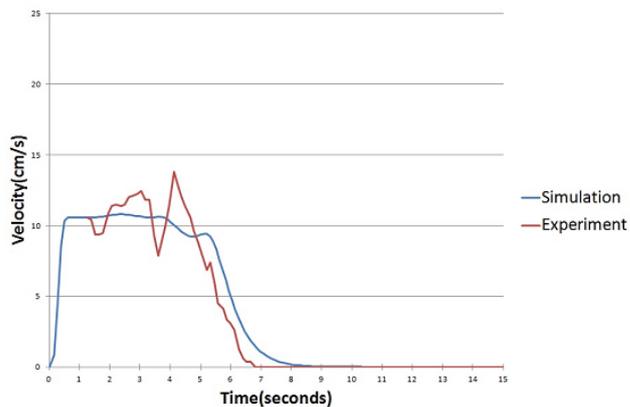


図 13 Vehicle 2 の速度変動の比較 (モデル 2)

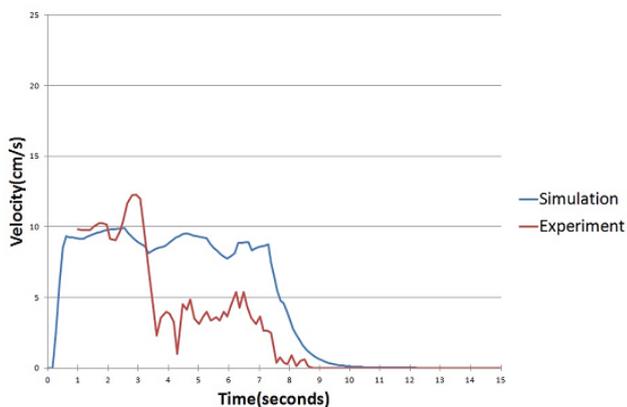


図 11 Vehicle 3 の速度変動の比較 (モデル 1)

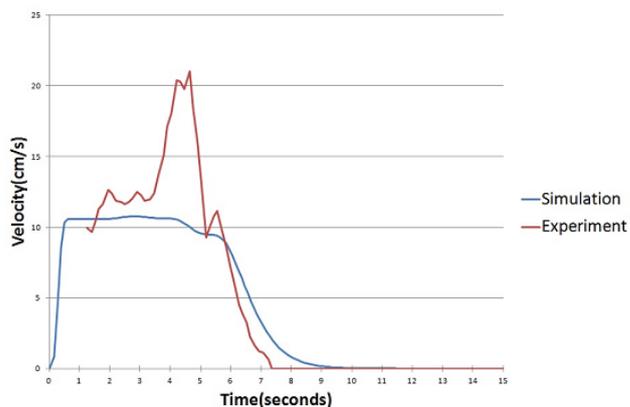


図 14 Vehicle 3 の速度変動の比較 (モデル 2)

間 (reacceleration) を示す。図 7, 図 8 から, 先頭車両が交差点手前で停止するたびに, 追従車両は先頭車両に従って停止することがわかる。また, 黒い縦線の時点において, 停止前に先頭車両であった車両の直後の車両が新しく先頭車両となり, 追従車両は新しい先頭車両の再加速に従って速度を変化させていることが示されている。各モデルにおける停止にかかる時間及び再加速にかかる時間を表 2 に示す。停止にかかる時間は, 1 回目の停止, 2 回目の停止, 3 回目の停止のいずれにおいても, モデル 2 が短いことがわかる。

つづいて, 数値シミュレーション結果とロボット車両による実験結果を比較する。モデル 1 およびモデル 2 において, シミュレーション開始後 15 秒後までの第 1 追従車両, 第 2 追従車両, 第 3 追従車両の速度変化を比較する。モデル 1 の結果を図 9, 図 10, 図 11 に, モデル 2 の結果を図 12, 図 13, 図 14 に示す。モデルによらず, それぞれの追従車両の減速のタイミングはほぼ一致しているが, 速度変動には数値シミュレーションと実験によって違いが見える。実験においては, 車両の左右の車輪の回転速度を変更することで, 車両が黒いライン上を走行するように車両の進行方向を制御している。これによる制御遅れは数値シミュレーションでは考慮されていない。このために, 実験結果と数値シミュレーションのずれが発生すると考えられる。

5. 結論

本研究では, 車車間通信が実現された状況での隊列走行において, 交差点を曲がる場合や隊列を組み替える場合に向けた, 車両の速度制御アルゴリズムについて検討した。隊列の再編成の具体的な例として, 隊列が交差点を左折する場合を考えた。隊列走行の計算機シミュレーションと, LEGO MINDSTORMS NXT によって作成したロボット車両による隊列走行実験によって検討した。車両の速度制御には, 直前方車両のみを参照する Chandler モデル, 直前方車両と隊列の先頭車両の 2 台を参照する Chandler モデルを比較した。その結果, 直前方車両のみを参照する Chandler モデルよりも直前方車両と隊列の先頭車両の 2 台を参照する Chandler モデルの方が, 追従車両の停止や, 再加速による速度の回復が早く行われることがわかった。

今後の課題として, 先行車両に対する遅れ時間の乱れを考慮した車両追従モデルの構築が挙げられる。本研究では遅れ時間は一定であるとしていたが, 実際の車両においては, 遅れ時間の乱れが想定される。車両追従モデルの安定性は遅れ時間に依存するため, 遅れ時間の乱れによって安定条件を満たさなくなるケースが生じることも考えられる。また, 本研究では交差点を左折する場合を取り上げたが, 隊列の合流や分離においても隊列を組み替えることとなる。隊列走行を行

うにあたり想定される他の状況においても検証する必要がある。

謝辞

本研究は JSPS 科研費 19K04140 の助成を受けたものです。

参考文献

- (1) トラックの隊列走行について. <https://www.mlit.go.jp/common/001178890.pdf>. 2023 年 10 月 8 日参照.
- (2) R. E. Chandler, R. Herman, and E. W. Montroll. Traffic dynamics; studies in car-following. *Operations Research*, Vol. 6, No. 2, pp. 165–184, 1958.
- (3) G. F. Newell. Nonlinear effects in the dynamics of car following. *Operations Research*, Vol. 9, No. 2, pp. 209–229, 1961.
- (4) 杉山雄規. 交通流の物理, *ながれ* 22, 2003.
- (5) E. Kita, C. Ushida, K. Asahina, T. Tamaki, Y. Wakita. Platoon Simulation of Vehicle Robots According to Vehicle following Model, *International Journal of Advances in Computer Science and Its Applications*, Vol.4, Issue 4, pp.205-207, 2014.
- (6) 北 栄輔, 朝比奈鋼司, 牛田千智, 玉城龍洋, 村井 希. 多台参照追従モデルによる隊列走行シミュレーションについて. *日本計算数理工学論文集*, Vol.19, pp.79-84, No.16-191201, 2019.
- (7) レゴマインドストーム. <https://www.lego.com/ja-jp/themes/mindstorms/about>. 2023 年 10 月 8 日参照.
- (8) S. Bexelius. An extended model for car-following. *Transportation Research*, Vol. 2, No. 1, pp. 13–21, 1968.
- (9) 脇田佑希子, 清水光輝, 玉城龍洋, 北 栄輔. 合流車両による交通渋滞緩和のためのセルオートマトンシミュレーション, *日本計算数理工学論文集*, Vol.10, pp.75-80, 2010.12.
- (10) *Next byte codes, not exactly c, and superpro c*. <http://bricxcc.sourceforge.net/nbc/>. 2023 年 10 月 8 日参照.
- (11) *Bricx command center*. <http://bricxcc.sourceforge.net/>. 2023 年 10 月 8 日参照.
- (12) *LEGO Mindstorms NXT2.0 ユーザガイド*. <http://www2.ocn.ne.jp/~takamoto/LegoMindstormsNXT2.0UserGuide.pdf>. 2023 年 10 月 8 日参照.