

平成 31 年度
修士論文

追い越し挙動に着目した
暫定 2 車線道路における交通状況解析

Traffic conditions analysis
focusing on overtaking behavior in two-lane high way

高知工科大学大学院
工学研究科基盤工学専攻
社会システム工学コース

都市・交通計画研究室

学籍番号 1225078

秦 啓

指導教員 西内 裕晶

副指導教員 赤塚 慎

副指導教員 高木 方隆

2020 年 3 月 10 日

要旨

暫定 2 車線高速道路は付加車線区間と単路区間が交互に設置された道路構造を持つ道路である。暫定 2 車線道路を対象とした多くの研究が付加車線区間の適切な設置間隔を明らかにすることを目指している。しかし、整備の実態は地形的要因から理想的な付加車線の設置が行われていない。本研究は付加車線区間で追い越しを実施した車両に追い越された車両が単路区間で追い越しを実施した車両に追いつくという一連の事象を旅行時間の短縮が見込めない追い越しと考えた。また、その様な追い越しは事故リスクを増やす行為とし、減少を目指す必要があると考える。暫定 2 車線高速道路を対象に追い越し実施と追い越しによる旅行時間の短縮効果をモデル化することで、旅行時間の短縮が見込めない追い越しと旅行時間を短縮する追い越しを判別することを可能とした。

Abstract

The two-lane highway has a road structure that is additional lane sections and single road sections. Many studies on the two-lane highway aim to determine the appropriate spacing of the section of the additional lane. But the ideal additional lane has not been installed due to the topographical factors. This study aims a reduction in the number of overtaking which Does not shorten travel time. It is modeling the effect of overtaking on the two-lane highway. This effect is the value of reducing the travel time by overtaking.it is possible to distinguish between overtaking that not shortens travel time and overtaking that shortens travel time.

目次

目次.....	2
図目次.....	4
表目次.....	5
1. 序章.....	6
1-1 研究の背景.....	6
1-2 研究の目的.....	6
1-3 研究の構成.....	6
2. 既往研究の整理と本研究の位置づけ.....	7
2-1 既往研究.....	7
2-1-1 暫定2車線高速道路に関する既往研究.....	7
2-1-2 旅行時間推定に関する既往研究.....	7
2-2 本研究の位置づけ.....	8
3. 研究対象と各種用語の定義.....	9
3-1 暫定2車線高速道路.....	9
3-2 追い越し効果の定義.....	9
3-2-1 ひとつの追い越しの定義.....	9
3-2-2 追い越し効果の定義.....	10
4. シミュレーションの校正.....	11
4-1 シミュレーションソフトの紹介.....	11
4-2 高知道ビデオ観測調査.....	14
4-2-1 調査対象.....	14
4-2-2 調査日・時間.....	14
4-2-3 調査項目.....	14
4-2-4 調査場所.....	15
4-2-5 調査機材.....	15
4-2-6 調査詳細.....	15
4-2-7 人員機材の配置.....	28
4-2-8 当日の動き.....	28
4-3 ビデオ調査結果.....	29
4-4 集計方法.....	29
4-5 トラフィックカウンターデータ.....	30

4-6	OD表の作成.....	32
4-6-1	取得データの限界.....	32
4-6-2	OD作成時のルール.....	32
4-6-3	OD表.....	32
4-7	シミュレーションの再現性確認.....	33
4-7-1	シミュレーションのチューニングに使用したパラメータ.....	33
4-7-2	シミュレーションのチューニング結果.....	33
4-7-3	課題・限界・留意点.....	34
5.	追い越し効果の計算.....	35
5-1	追い越し効果の計算方法.....	35
5-1-1	計算フロー.....	36
5-1-2	追い越し効果計算の3つのパターン.....	37
5-1-3	追い越し効果の見方.....	37
5-2	追い越し効果の計算結果考察.....	38
5-2-1	概要.....	38
5-2-2	追い越し効果の分布.....	38
5-2-3	追い越し発生位置.....	39
6.	追い越し実施基準.....	41
6-1	判別分析(数量化Ⅱ類)による追い越しの効果と周辺環境, 自転車走行状況の関係性把握.....	41
6-1-1	説明変数.....	41
6-1-2	目的変数.....	41
6-2	判別分析の結果.....	41
6-2-1	高知道上り方向の追い越し効果判別(走行距離を含む).....	41
6-2-2	追い越し効果の判別(走行距離を含まない).....	44
6-3	追い越し実施基準の整理.....	47
7.	結論.....	51
7-1	本研究の成果.....	51
7-2	課題.....	51
8-3	今後の展望.....	52

目次

図 1	暫定 2 車線道路区間のイメージ図	9
図 2	旅行時間の差の考え方	10
図 3	ひとつの追い越しとその追い越し効果	10
図 4	シミュレーション入力区間	12
図 5	シミュレーションリンクデータの設定	12
図 6	シミュレーション設定の画面	13
図 7	カメラ設置位置とカメラアングル (一宮西跨道橋周辺)	16
図 8	一宮西跨道橋(高知端部流入)	17
図 9	一宮西跨道橋(高知端部流出)	17
図 10	カメラ設置位置とカメラアングル(八代橋周辺)	18
図 11	八代橋(伊野上り方向車群観測)	19
図 12	八代橋②(伊野上り方向断面)	19
図 13	八代橋(伊野付加車線車両挙動)	20
図 14	八代橋(伊野下り方向断面)	20
図 15	カメラ設置位置とカメラアングル(西山橋)	21
図 16	西山橋(土佐付加車線車両挙動)	22
図 17	西山橋(土佐上り方向断面)	22
図 18	西山橋(土佐付加車線車両挙動)	23
図 19	カメラ設置位置とカメラアングル(須崎東)	24
図 20	須崎東(断面)	25
図 21	調査員位置(土佐 SIC)	26
図 22	カメラ設置位置と撮影範囲(高知 IC)	27
図 23	高知 IC 駐車場(断面)	28
図 24	集計マニュアル 1	29
図 25	集計マニュアル 2	30
図 26	集計マニュアル 3	30
図 27	トラフィックカウンターと観測位置関係図	31
図 28	トラフィックカウンター4 地点の速度分布	31
図 29	追い越し効果計算フロー	36
図 30	追い越し効果の 3 つのパターン	37
図 31	無駄な追い越しの発生状況	38
図 32	追い越し効果の分布	39
図 33	追い越し発生位置	40

表目次

表 1	車両設定のパラメーター一覧.....	13
表 2	カメラ設置状況.....	28
表 3	OD 表の一部.....	32
表 4	変更した車両パラメーター一覧.....	33
表 5	キャリブレーション後の車両の走行パラメータ.....	34
表 6	速度の再現性.....	34
表 7	判別分析の結果(説明変数).....	42
表 8	判別分析の結果(判別的中率).....	43
表 9	判別分析の結果(説明変数の判別係数).....	43
表 10	判別分析結果(各群重心).....	43
表 11	判別分析(基本統計量).....	44
表 12	判別分析の結果(説明変数).....	45
表 13	判別分析の結果(判別の中率).....	46
表 14	判別分析の結果(説明変数の判別係数).....	46
表 15	判別分析結果(各群重心).....	46
表 16	判別分析(基本統計量).....	47
表 17	追い越し実施基準追い越し車両速度と1台後方の車両速度.....	48
表 18	追い越し実施基準追い越し車両速度と1台後方の車両速度.....	48
表 19	追い越し実施基準追い越し車両速度と密度.....	49
表 20	追い越し実施基準追い越し車両速度と1台前方の車両との距離差.....	49
表 21	追い越し実施基準追い越し車両速度と1台後方の車両との距離差.....	49
表 22	追い越し実施基準追い越し車両速度と付加車線区間の残り距離.....	49

1. 序章

1-1 研究の背景

暫定 2 車線高速道路では、単路区間と付加車線区間が交互に設置されている道路構造であるために、限られた区間内で追い越しが実施されている。また、常に 2 車線以上ある高速道路とは異なり、単路区間が存在することから、低速の車両の存在により、低速車両を先頭とした車群が形成され、すべての車両が走行したい速度で走行し続けることが出来ない道路構造である。以上道路構造が原因となり、暫定 2 車線道路区間では、付加車線区間で追い越しを実施した車両に追い越された車両が単路区間で追いつくという事象が発生している。

本研究では追い越しの主目的は旅行時間の短縮であると考え、旅行時間の短縮が見込めない追い越しの実施は事故リスクを高めるのみであり、高速道路上の事故リスク低下のために不必要な追い越し発生を減少を目指す必要があると考えた。

1-2 研究の目的

旅行時間の短縮が見込めない追い越しを減少させるために、暫定 2 車線道路における追い越し実施基準の整備を本研究の目的とする。追い越し実施基準とは追い越しの実施が旅行時間を短縮するのか短縮しないのかを判別する基準を指す。基準整備を達成するために、追い越しによる旅行時間の短縮を定量化する。また、追い越し実施時の周辺状況より、追い越しによる旅行時間の短縮結果を導くモデルを作成する。

1-3 研究の構成

本論文は全 7 章で構成されている。2 章では暫定 2 車線道路に関する研究、高速道路における追い越しに関する研究の 2 種類の既往研究を整理し、本研究の位置づけを明確にする。3 章では本研究で対象とする暫定 2 車線道路や追い越し他使用する用語の定義を示す。4 章では研究に使用したシミュレーションソフトの紹介と再現性確保のために実施した現地調査の内容を示す。5 章ではシミュレーションを用いて算出されたデータを元に本研究で定義した追い越しの効果を計算した結果を示す。6 章では追い越し効果の計算結果を元に追い越し判断基準作成のために追い越しが旅行時間を短縮したか短縮していないかを整理する。また、判別分析を用いて追い越し効果がある場合とない場合を判別分析よりモデル化した結果を示す。モデルの結果を使用して追い越し実施基準を作成する。7 章では以上の内容を元に成果と今後の課題展望を記す。

2. 既往研究の整理と本研究の位置づけ

2-1 既往研究

2-1-1 暫定2車線高速道路に関する既往研究

塩見ら(2011)¹⁾は暫定2車線道路のボトルネックにおいて、渋滞が発生する確率を通過する車群の大きさのモデルで説明した。車群の定義を明確にし、ボトルネック部において車群が大きいほど同一車群内の車両の同士が減速の影響でさらに減速し渋滞が発生することを明らかにした。4車線以上の高速道路での付加車線設置に関する研究は多く存在するが、高速道路の暫定2車線区間の付加車線設置の研究は少なく、またボトルネック現象や交通容量などについては十分に把握されていないとした。吉川ら(2005)²⁾は実データを用いて暫定2車線区間におけるボトルネック交通容量の分析を行い、渋滞による捌け交通量の低下を確認した。4車線と暫定2車線を比較しボトルネック上流に付加車線を設置することが4車線化以外の渋滞軽減に対する唯一の対策だと述べた。

中村ら(2011)³⁾は付加追い越し区間長や設置間隔といった設置水準を決定する方法論の確立を目指し、追従車密度を用いて往復2車線道路のサービスの質を定量的に表し、付加追い越し車線の設置による追従車密度の推移をマイクロ交通流シミュレーションを用いて分析することで設置水準を求める手法を開発することを目的とした。付加車線区間長が大きくなるほど追従車密度は低減するがその効果は次第に小さくなること、1車線区間では追従車密度が急激に増加しその後増加率は小さくなることを明らかにした。また、付加追い越し区間長と設置間隔に応じた追従車密度の推定を可能にした。

成嶋ら(2017)⁴⁾は高速道路の暫定2車線区間の交通機能低下箇所への効果的な付加車線設置に関する研究の一環として暫定2車線道路の速度変動の実態を分析しまとめた。成果としては、付加車線区間で回復した速度は長く維持せず、付加車線端部では速度低下がおり、交通量レベルが高いほど顕著に速度低下が現れることを明らかにした。

暫定2車線道路に関する研究の多くは付加車線の設置に注目した研究である。その他には、中央分離がされていないため重大事故発生確率が多い暫定2車線道路の安全対策としてセンターポールの設置と効果についての研究が行われている。

根川ら(2015)⁵⁾は暫定2車線高速道路の付加車線における追い越し挙動のモデルを開発した。既存のモデルを基に追い越し判断時に2台前の車両も判断材料にすることで再現性を向上させた。

2-1-2 旅行時間推定に関する既往研究

堀口ら(2003)⁶⁾は高速道路本線で収集したアップリンク情報から区間旅行時間を計測し、短期将来における区間旅行時間を推定する手法を提案した。アップリンク情報から得られる旅行時間と下流断面での累積交通量より旅行時間の予測を行っている。

鈴木ら(2002)⁷⁾は高速道路上の起終点旅行時間を推定する手法を提案した。既存のニュー

ロカルマンフィルタを用いた手法の課題を整理し、マクロ交通流モデルとカルマンフィルタを用いて間接的に起終点旅行時間を推定する方法を提案している。

山崎ら(2008)⁸⁾は ETC(Electronic Toll Collection)搭載車の流入・流出時刻から得られる IC 間の旅行時間データを使用した旅行時間信頼性評価を可能にした。

旅行時間の推定は感知器や ETC の情報からマクロ的算出されている。暫定 2 車線道路のような検知器の少ない道路における微小区間での旅行時間推定方法は存在しない。既存の ETC を使用しても暫定 2 車線道路は地方部に多いことから追い越し動作という微視的な分析に用いるにはデータの密度等推定には限界がある。

2-2 本研究の位置づけ

暫定 2 車線道路の研究から、付加車線の適切な設置について様々な事柄が明らかになったが、実際の整備時には地形的要員や予算に大きな制約があり、理想とする付加車線の設置は行えていない。また、既存の付加車線を延長や移設することは暫定 2 車線道路の採算が取れていない背景からも現実的でない。したがって、ソフト面の対策で既存の付加車線を効率よく運用することを目指す必要がある。付加車線は追い越しを実施するための区間であり、追い越しが効果的に行われているかを評価するために追い越しの効果を定量化する必要がある。ソフト面の対策として、本研究では追い越しの目的を旅行時間の短縮と考え旅行時間に注目した。高速道路を対象とした旅行時間の推定は OD 間での推定を目指すものが主であり、交通の流れとしてマクロ的に推定している。追い越しに対する旅行時間の短縮効果を定量化するためにミクロでの旅行時間推定方法を確立することが必要である。

本研究は追い越し実施と旅行時間の短縮の関係性を追い越し毎に定量化し評価することを目指す研究である。

3. 研究対象と各種用語の定義

3-1 暫定 2 車線高速道路

本研究で研究対象とした暫定 2 車線高速道路を紹介する。暫定 2 車線道路とは、4 車線以上の道路計画がある道路区間に対して、採算性、交通量の観点から 2 車線で供用し、交通量が増加した等の要因によって 2 車線の道路を平行して追加整備し、合計 4 車線として運用する道路のことである。日本全国の高規格幹線道路のうち平成 28 年供用済みの道路の約 38%が暫定 2 車線区間である。世界的にも珍しい道路構造で、先進国と比較すると、アメリカで 2.3%、ドイツで 1.1%と日本は世界的にも 2 車線高速道路が特別に多い国であり、国土の狭さや地形等が影響しているといわれている。

暫定 2 車線道路区間道路の構造の特徴として、付加車線区間と単路区間が交互に繰り返して設置されている特徴がある。単路区間とは片側 1 車線の対面通行区間のことで、1 車線しか車線がない為追い越しが行えず、前方に低速の車両が存在する場合にその車に追従することで車群が発生する。また、付加車線区間は走行車線に加えて追い越しを可能にする付加車線が設置された区間のことで、単路区間で発生した車群から高速の車両を開放する為に 6 キロから 10 キロに一度約 2 キロ前後設置されていることが多い。道路構造のイメージ図を図 1 に示す。

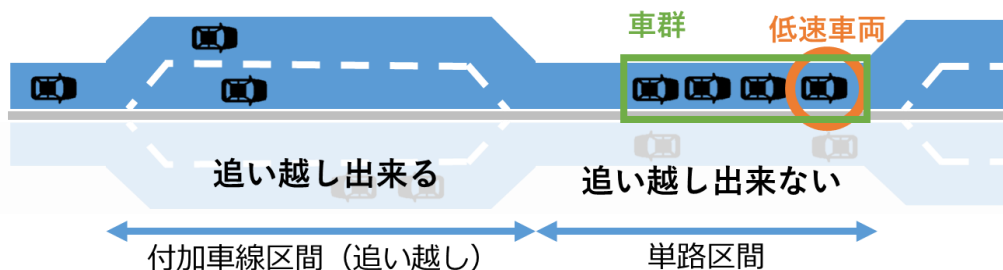


図 1 暫定 2 車線道路区間のイメージ図

3-2 追い越し効果の定義

本研究で新しく定義した「ひとつの追い越し」、「追い越し効果」の定義を紹介する。

3-2-1 ひとつの追い越しの定義

ひとつの追い越しとは「ある車両が追い越し動作を開始してから次の追い越し動作開始するまで又はある車両が対象とする区間を退出するまで」と定義した。ひとつの追い越しは追い越し効果を計算するにあたり、追い越し効果がどの段階で確定するのかを決定する必要があり、追い越しの効果を計算するために追い越しの単位として定義した。

3-2-2 追い越し効果の定義

追い越し効果の定義は「ひとつの追い越しによって短縮された旅行時間」である。追い越しの主要目的は旅行時間の短縮であると考え、追い越しを実施したことにより得られる旅行時間の短縮量を追い越しにより得られた効果と定義した。追い越しをしなかった場合の走行位置は実地調査、シミュレーションともに観測ができず、計算に使用できないため、旅行時間の差は追い越しをした車両と追い越しをした直後に後方にいた車両の旅行時間を同一地点で比較し算出することとした。図 2 は追い越しの際の旅行時間の短縮の考え方を示した図である。図 3 はひとつの追い越しとその追い越し効果を Time-Space 図で表したものである。図 3 のひとつの追い越しは追い越し動作開始から対象区間を退出した地点までの場合を示している。

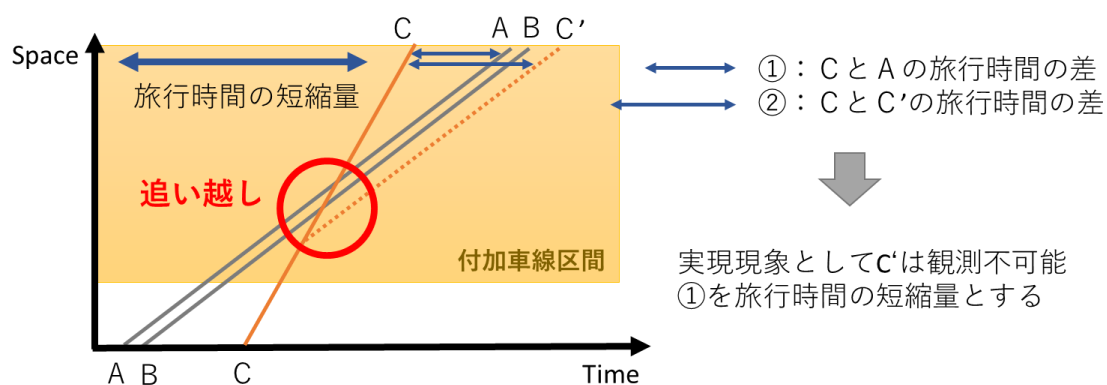


図 2 旅行時間の差の考え方

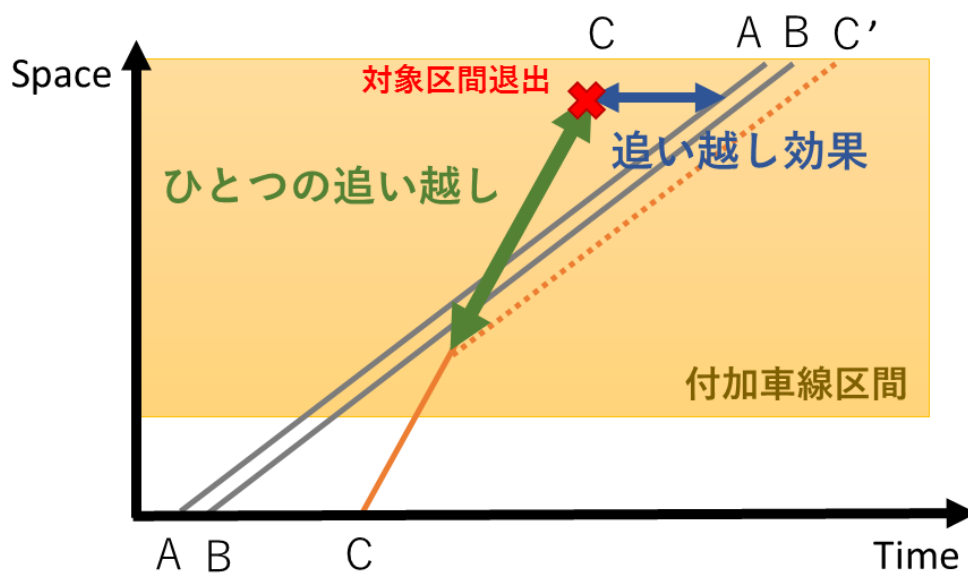


図 3 ひとつの追い越しとその追い越し効果

4. シミュレーションの校正

実際の道路上の追い越し効果を正確に推定するには、高速道路上すべてを観測しすべて車両位置、追い越しと区間退出を記録する必要がある。しかし、対象とする高速道路をすべて観測することは、現実的ではなく非常に困難である。したがって、シミュレーションソフトを使用しシミュレーションソフトの計算結果を用いて追い越し効果を計算することとする。シミュレーションソフトが実際の道路上の車両挙動を再現している必要があり、実地調査を元に作成したデータを使用して、シミュレーションのパラメータを校正する。

4-1 シミュレーションソフトの紹介

本研究では株式会社アイ・トランスポート・ラボのマイクロ交通流シミュレーション microAVENUE を使用する。microAVENUE はマイクロ交通流シミュレーションと呼ばれるシミュレーションソフトウェアであり、車両車種毎にパラメータを設定可能で車両の 1 台 1 台の挙動が結果に大きな影響を与えると考えられる分析に用いられる。本研究は車両 1 台 1 台の追い越しに着目する為、車両を詳細に制御可能であるマイクロ交通流シミュレーションを選択した。シミュレーションを実行するには道路情報、車両情報、OD 交通量、シミュレーション設定を入力する必要がある。出力結果は車両の走行位置と速度を 1 秒刻みで記録した csv データである。

道路情報には、研究対象の高知道の道路情報を入力した。入力した情報としては、車線の長さ、法定速度、飽和交通流率等である。図 4 に対象とした区間全体の図を示す。図 5 に付加車線区間、単路区間の設定画面を示す。実際の法定速度に合わせて単路部の法定速度は 70km/h、付加車線区間の法定速度は 80km/h とした。

車両情報には、希望走行速度や車両の大きさといったパラメータが設定可能である。本研究では、早い車両、遅い車両の 2 種類に分けて分析を実施した。早い車両とは一般的な車両を想定している、遅い車両とは、大型トラックなどの速度に制限のある車両を想定している。表 1 に車両に設定可能なパラメータの一覧を示す。

OD 交通量は、車種毎に始点、終点、走行開始時間(15 分刻み)をまとめたデータである。

シミュレーション設定は車両を一様なタイミングで出現させるか、ランダムなタイミングで出現させるか、シミュレーションの開始時間、終了時刻、車両発生の間隔を設定できる。図 6 に実際に使用したシミュレーションの設定画面を示す。



図 4 シミュレーション入力区間

リンク設定

リンクID E56_007_1

上流ノードID I_22_0418131903018

下流ノードID I_22_0418131859272

リンク長[m] 266.32 本線レーン数 2

右折レーン数 0 右折レーン長[m] 0.00

左折レーン数 0 左折レーン長[m] 0.00

リンク容量[pcu/h/レーン] 1800 飽和交通流率[pcu/h/レーン] 左折 1800

ジャム密度[台/km] 140 直進 1800

自由流速度[km/h] 80 右折 1800

Uターン 1800

禁止進行方向 u_turn

料金属性 固定制料金[円] 0

従量制料金[円] 0

信号停止回数の判定閾値[秒] 120

渋滞長判定の速度閾値[km/h] 10

efficient path 固定フラグ

AVNファイル拡張レコード UPDESIRESPDFACTOR=0.900:1.10,DOWNDESIRESPDFACTOR=

OK キャンセル

リンク設定

リンクID E56_008_1

上流ノードID I_22_0418132823801

下流ノードID I_22_0418131903018

リンク長[m] 7970.17 本線レーン数 1

右折レーン数 0 右折レーン長[m] 0.00

左折レーン数 0 左折レーン長[m] 0.00

リンク容量[pcu/h/レーン] 1800 飽和交通流率[pcu/h/レーン] 左折 1800

ジャム密度[台/km] 140 直進 1800

自由流速度[km/h] 70 右折 1800

Uターン 1800

禁止進行方向 u_turn

料金属性 固定制料金[円] 0

従量制料金[円] 0

信号停止回数の判定閾値[秒] 120

渋滞長判定の速度閾値[km/h] 10

efficient path 固定フラグ

AVNファイル拡張レコード UPDESIRESPDFACTOR=1.050,DOWNDESIRESPDFACTOR=1.050

OK キャンセル

図 5 シミュレーションリンクデータの設定

表 1 車両設定のパラメータ一覧

項目	説明	単位
DESIRESPDFACTOR	希望速度係数(リンクの自由流速度に対して)	
DESIRESPDFACTORSTDEV	希望速度係数標準偏差	
MAXACCELERATION	最大加速度	[m/s ²]
MAXACCELTDEV	最大加速度標準偏差	[m/s ²]
MAXACCELERATION_LOWERLIMIT	最大加速度下限値	[m/s ²]
MAXACCELERATION_UPPERLIMIT	最大加速度上限値	[m/s ²]
DESIREDACCELERATION	希望加速度	[m/s ²]
DESIREDACCELTDEV	希望加速度標準偏差	[m/s ²]
MAXDECELERATION	最大減速度	[m/s ²]
MAXDECELERATION_LOWERLIMIT	最大減速度標準偏差	[m/s ²]
MAXDECELERATION_UPPERLIMIT	最大減速度下限値	[m/s ²]
DECELERATIONGAIN	最大減速度上限値	[m/s ²]
DESIREDDECELERATION	希望減速度	[m/s ²]
DESIREDDECELTDEV	希望減速度標準偏差	[m/s ²]
NATURALDECELERATION	自然減速度	[m/s ²]
IDMDELTA	IDMのデルタパラメータ	
GRADIENTCOEF	勾配に対する係数	
FORESIGHTDURATION	前方探査距離	[m]
INSENSITIVESPDRANGE	無意識の速度低下不感帯	[km/h]
SPDDIFFTOLERANCE	速度低下許容幅	[km/h]
SPDDIFFCRUISE	巡航時速度幅	[km/h]
MEANDISTTOTURN	右左折の際に何m手前で車線変更するか(期待値)	[m]
STDVDISTTOTURN	右左折の際に何m手前で車線変更するか(標準偏差)	[m]
MINDISTTOTURN	右左折の際に何m手前で車線変更するか(最小値)	[m]
MOVEPASSINGLANEPROB	追越希望受諾確率	
RETURNCRUISINGLANEPROB	走行車線復帰確率	
MINGAPTIME	最小ギャップ	[s]
MINTTC	最小TCC(Time-To-Collision、前方車両との距離を速度の差(相対速度)で割ることで算出される衝突までの時間)	[s]
THRESHOLDSPPEEDDOWN	車線変更時に後続車の減速幅がこれ以上になる場合は車線変更しない	[km/h/s]

シミュレーション設定 ✖

シミュレーション設定名称	シミュレーション設定
シミュレーション開始時刻	2015:10:08:07:00:00
シミュレーション終了時刻	2015:10:08:09:00:00
車両発生時間間隔[秒]	900
経路コスト情報更新間隔[秒]	600
データ集計時間間隔[秒]	900
乱数シード値(車両発生スケジュール)	1000
乱数シード値(経路選択行動)	1000
乱数シード値(その他)	1000
走行レーンサイド	<input checked="" type="radio"/> 左側通行 <input type="radio"/> 右側通行
データスワップ	<input type="checkbox"/> On
最大スワップサイズ[MB]	0
サブスキャンインターバル	100
外部AVNファイル	<input type="text" value=""/> ...
AVNファイル拡張レコード	<input type="text" value=""/>
<input type="button" value="OK"/> <input type="button" value="キャンセル"/>	

図 6 シミュレーション設定の画面

4-2 高知道ビデオ観測調査

高知道をシミュレーション上で再現するために高知道で実地調査を実施した。取得したデータはシミュレーション内を走行する車両の発生量や速度等のパラメータ入力に使用する。調査は2019年9月20日に高知道の須崎東ICから高知ICの約35kmの区間で実施した。調査はビデオカメラをIC横の事務所駐車場や本線上部のオーバブリッジに設置しナンバープレートや車両の走行挙動を観測する方法で調査した。

4-2-1 調査対象

暫定2車線区間が存在する、高知自動車道の高知ICから須崎東ICの間を走行する車両を調査の対象とした。

4-2-2 調査日・時間

ビデオカメラの観測は2019年9月20日の午前7時から午前10時までの3時間とした。

4-2-3 調査項目

調査項目は大きく分けて2種類のデータの取得を目的としている。ひとつは発生集中交通量である。シミュレーション入力に必要なOD交通量に使用する。もうひとつは車両の走行に関するデータである。取得する情報は車種やナンバープレートといった車両を識別する情報や追い越しを実施の有無である。以下に詳細を示す。

■発生集中交通量

ビデオデータより車両識別情報と断面通過時間より、同一車両が調査対象区間のどの地点からどの地点へ移動したかを整理し作成する。

■車両の走行に関するデータ

・車両識別情報

ビデオデータより車種、車両のナンバー、ナンバープレートの色、断面通過時刻を集計し、観測した車両を個別識別する。

・走行状況

ビデオデータより走行速度、走行位置、走行レーンを記録し、車両識別情報と紐付け、各車両の希望走行速度や速度分布、追い越し車両の判別、ICからの退出を記録する。

・その他

付加車線を広範囲で記録するビデオデータからは、追い越し開始位置、車群内順位など位置や複数車両に掛かるデータを記録する。

4-2-4 調査場所

発生集中交通量作成に用いた調査場所とシミュレーション内ゾーンの関係を示す。

- ・高知道端部流入：高知 IC(一宮西跨道橋)
- ・高知道端部流出：高知 IC(一宮西跨道橋)
- ・高知 IC 流入：高知 IC 横駐車場, 高知 IC(一宮西跨道橋)
- ・高知 IC 流出：高知 IC 横駐車場, 高知 IC(一宮西跨道橋)
- ・伊野 IC 流入：伊野 IC(八代橋), 高知 IC 横駐車場, 高知 IC(一宮西跨道橋)
- ・伊野 IC 流出：伊野 IC(八代橋)
- ・土佐 IC 流入：土佐 IC(西山橋), 伊野 IC(八代橋)
- ・土佐 IC 流出：土佐 IC(西山橋)
- ・土佐 SIC 流入：土佐 SIC 上り
- ・土佐 SIC 流入：土佐 SIC 下り
- ・須崎 TB 流入：須崎 TB 付近(料金所駐車場)
- ・須崎 TB 流出：須崎 TB 付近(料金所駐車場)

ビデオカメラが設置できず土佐 IC からの流入や伊野 IC からの流入は直接観測できない。前後の道路本線上のカメラの情報より、データを整理し流入地点を特定する。

4-2-5 調査機材

調査にはビデオカメラ 11 台, 三脚 11 個, 車両 2 台を使用した。

4-2-6 調査詳細

高速道路周辺の OB や駐車場, 路上にビデオカメラを設置し観測を実施した。調査地点ごとにカメラの設置位置, カメラの撮影アングル, 観測項目を紹介する。

・高知 IC 付近(一宮西跨道橋)

一宮西跨道橋では，高知道端部を通る車両を記録する．流入，流出に各 1 台カメラを設置し，観測を行う．主に通過する車両の識別情報と通過時刻を記録する．カメラ設置位置とカメラアングルを図 7 に示す．また，個々のカメラの撮影したビデオデータの一部を図 8，9 に示す．ビデオカメラ 2 台と三脚 2 個，調査員 1 名を設置した．

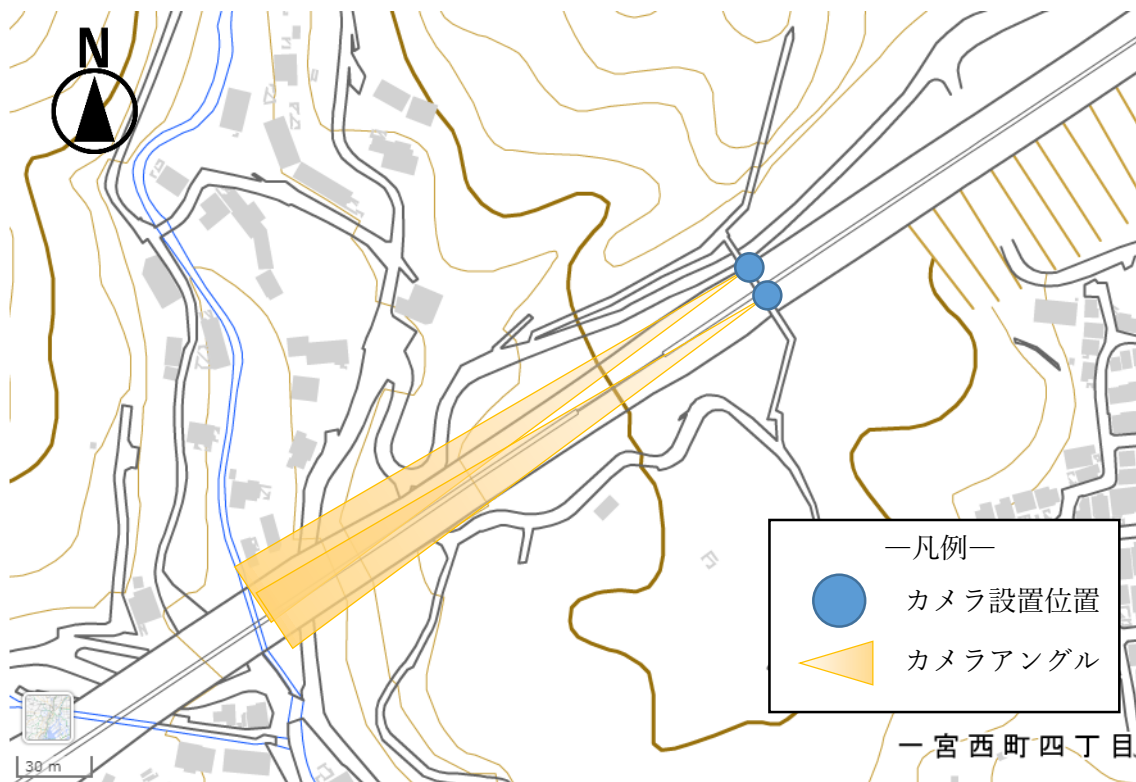


図 7 カメラ設置位置とカメラアングル (一宮西跨道橋周辺)



图 8 一宮西跨道橋①(高知道端部流入)

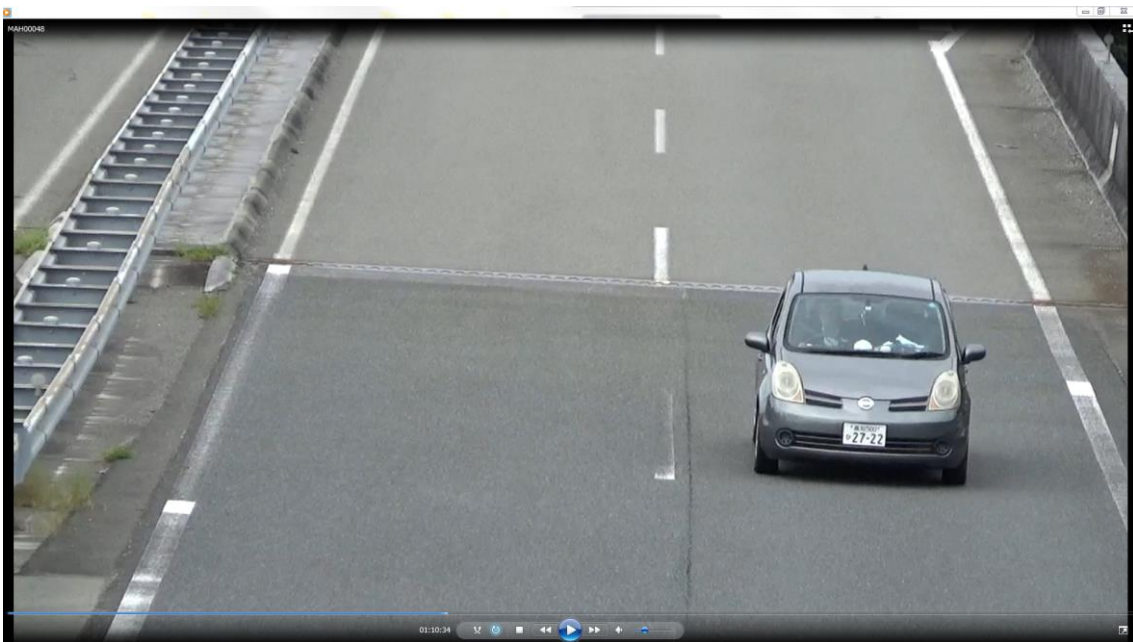


图 9 一宮西跨道橋②(高知道端部流出)

・伊野 IC 付近(八代橋)

八代橋では，伊野 IC を通る車両を記録する．上り方向単路部流出の車群観測に 1 台，上り方向断面観測に 1 台，伊野 IC 周辺付加車線走行挙動観測に 1 台，下り方向断面観測に 1 台の計 4 台を設置した．断面観測のカメラでは通過する車両の識別情報と通過時刻を記録する．単路部流出の車群観測カメラは車群の大きさ，速度を観測，付加車線走行挙動は走行レーン，追い越し開始位置を観測する．カメラ設置位置とカメラアングルを図 10 に示す．また，個々のカメラの撮影したビデオデータの一部を図 11~14 に示す．ビデオカメラ 4 台と三脚 4 個，調査員 2 名を設置した．

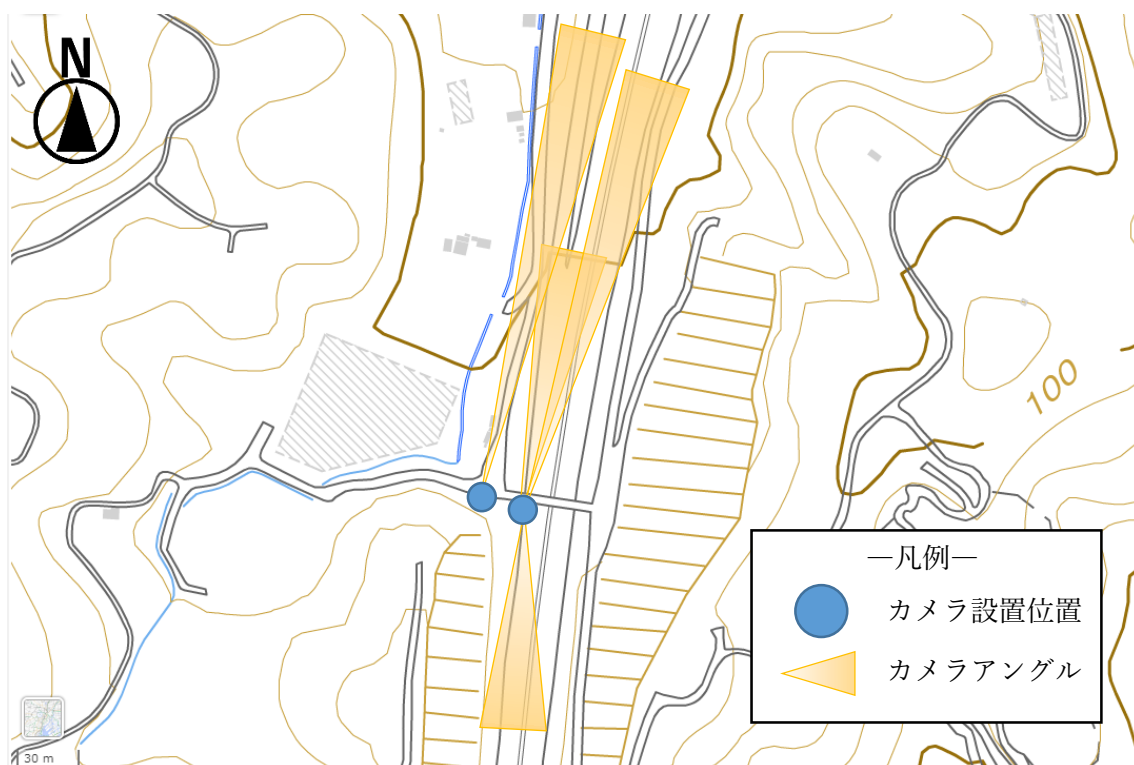


図 10 カメラ設置位置とカメラアングル(八代橋周辺)



図 11 八代橋①(伊野上り方向車群観測)



図 12 八代橋② (伊野上り方向断面)



図 13 八代橋③(伊野付加車線車両挙動)



図 14 八代橋④(伊野下り方向断面)

・土佐 IC 付近(西山橋)

西山橋では、土佐 IC を通る車両を記録する。土佐 SIC 方向の付加車線観測に 1 台、上り方向断面観測に 1 台、伊野 IC 方向付加車線走行観測兼土佐 IC 流入観測に 1 台の計 3 台を設置した。断面観測のカメラでは通過する車両の識別情報と通過時刻を記録する。付加車線観測は走行レーン、追い越し開始位置を観測する。カメラ設置位置とカメラアングルを図 15 に示す。また、個々のカメラの撮影したビデオデータの一部分を図 16~18 に示す。ビデオカメラ 3 台と三脚 3 個、調査員 2 名を設置した。付加車線区間の両端が観測できず、車群の順位は推測のみ可能である。

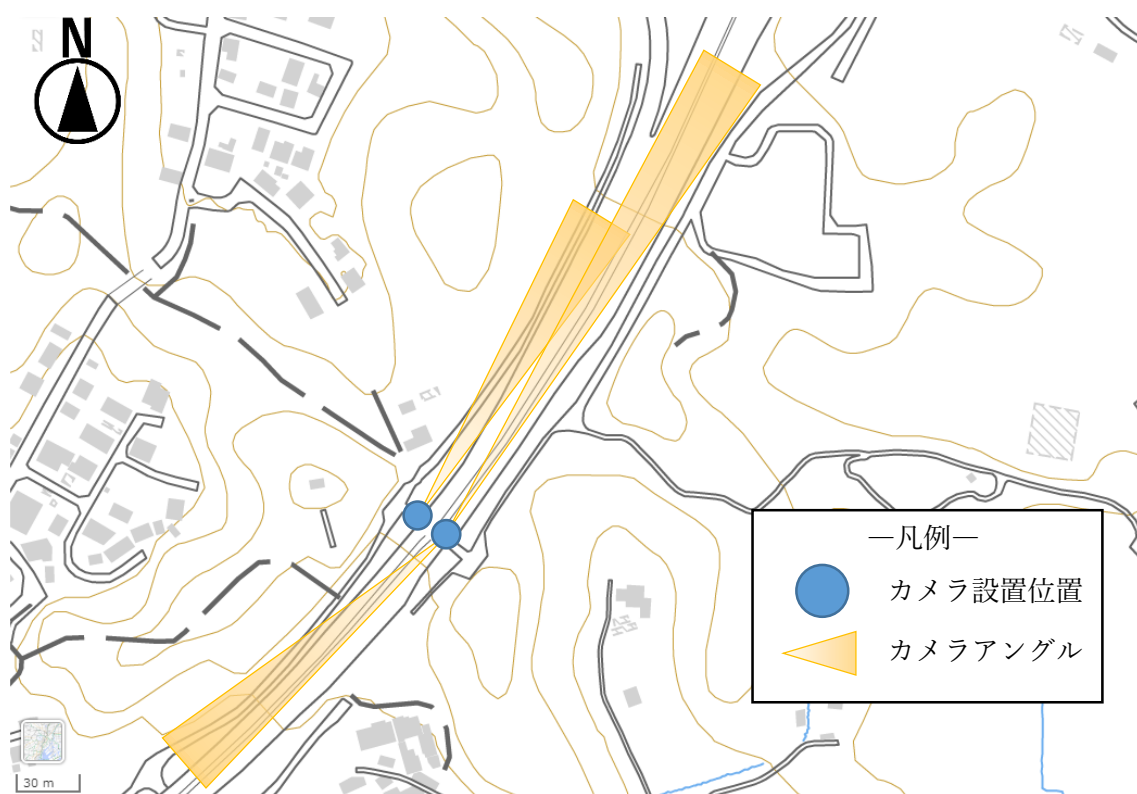


図 15 カメラ設置位置とカメラアングル(西山橋)



図 16 西山橋①(土佐付加車線車両挙動)



図 17 西山橋②(土佐上り方向断面)



図 18 西山橋③(土佐付加車線車両挙動)

・須崎東

須崎東では、須崎東料金所を通る車両を記録する。料金所を通過する車両(上り・下り)を観測する為にカメラを1台設置した。断面観測のカメラでは通過する車両の識別情報と通過時刻を記録する。カメラ設置位置とカメラアングルを図19に示す。また、カメラの撮影したビデオデータの一部を図20に示す。ビデオカメラ1つと三脚1つ調査員1名を設置した。カメラの設置位置が本線上でないこと、2車線を1台のカメラで観測する為、車両のナンバーが確認できない場合がある。

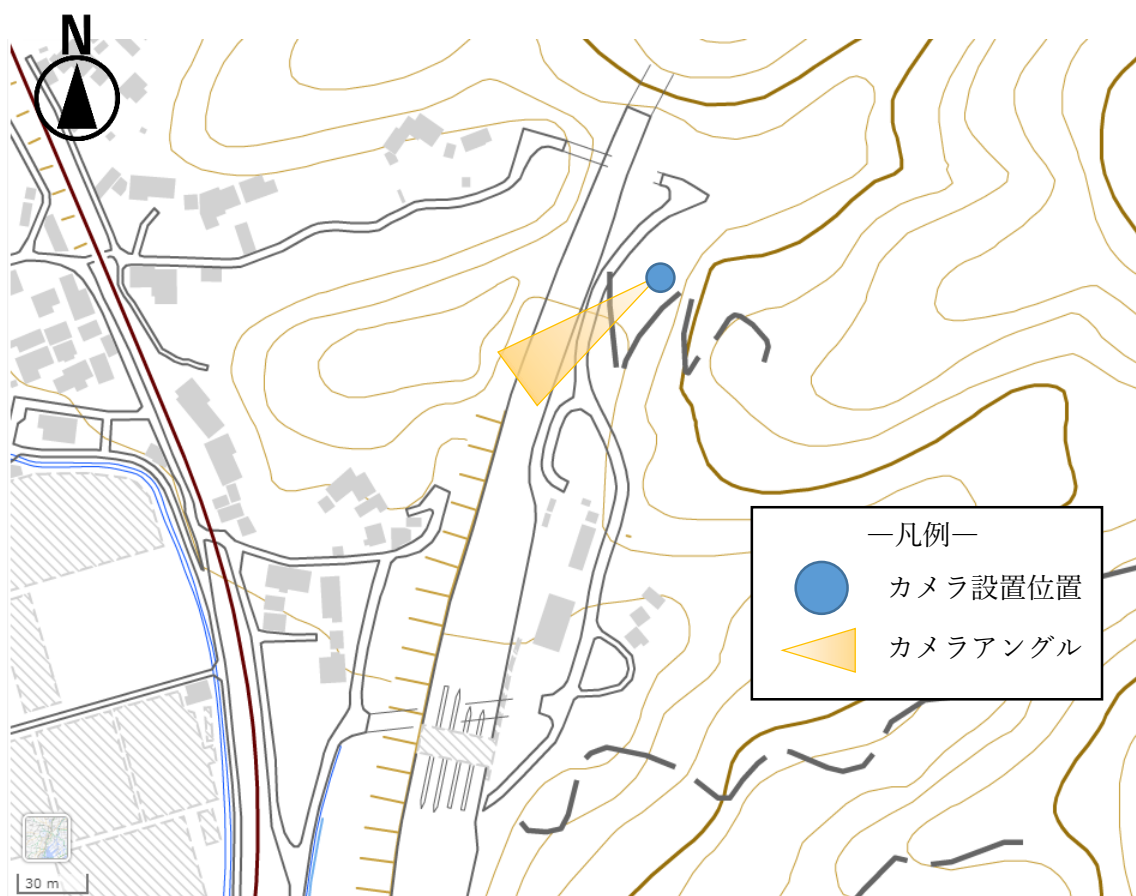


図 19 カメラ設置位置とカメラアングル(須崎東)



図 20 須崎東(断面)

・土佐 SIC

土佐 SIC では、SIC から流入流出する車両を記録する。上り、下りに調査員を 1 名ずつ計 2 名配置し、記録用紙に車両の通過時刻、車両の色、ナンバーを記録した。調査地点を図 21 に示す。



図 21 調査員位置(土佐 SIC)

・高知 IC 駐車場

高知 IC 駐車場では，高知 IC を利用する車両を記録する．料金所を通過する車両(上り・下り)を観測する為にカメラを 1 台設置した．断面観測のカメラでは通過する車両の識別情報と通過時刻を記録する．カメラ設置位置とカメラアングルを図 2 2 に示す．また，カメラの撮影したビデオデータの一部を図 2 3 に示す．ビデオカメラ 1 台と三脚 1 個，調査員 1 名を設置した．カメラの設置位置が本線上でないこと，上下方向を 1 台のカメラで観測する為，車両のナンバーが確認できない場合がある．

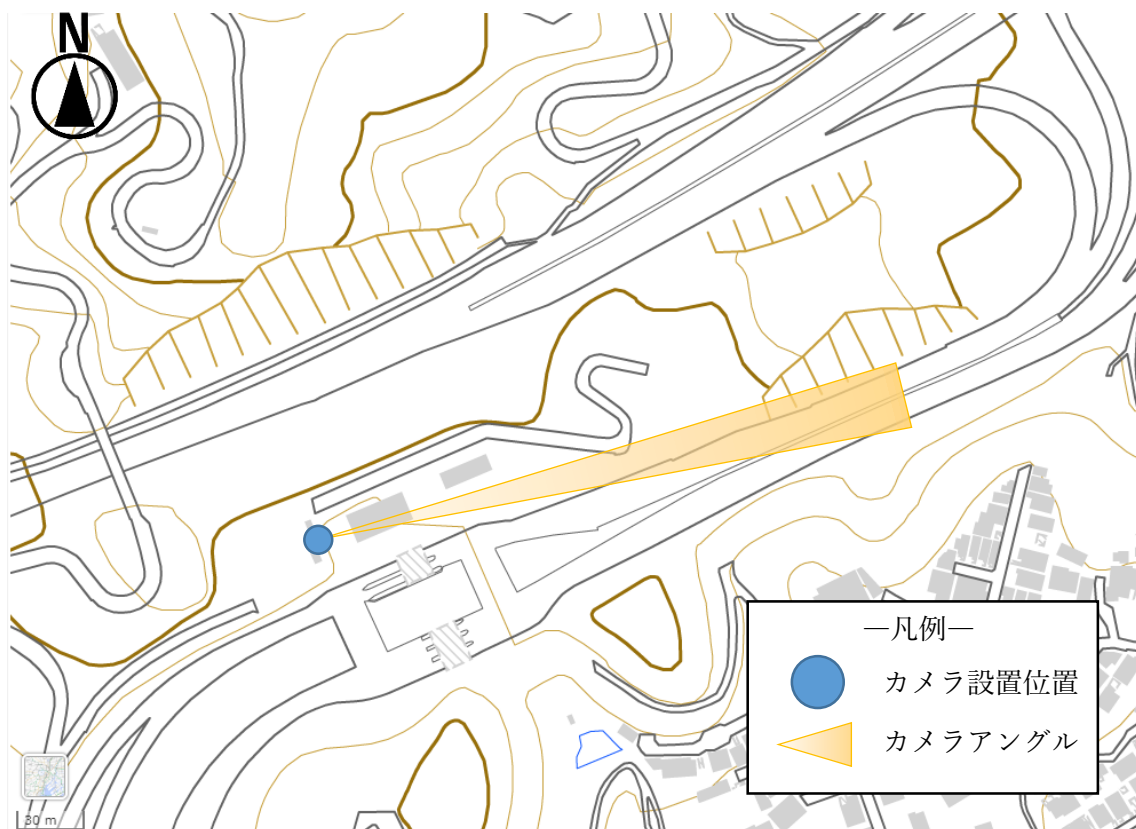


図 22 カメラ設置位置と撮影範囲(高知 IC)



図 23 高知 IC 駐車場(断面)

4-2-7 人員機材の配置

人員と機材の配置を表 2 に示す。

表 2 カメラ設置状況

地点	カメラ(台)	人数(名)
一宮西跨道橋	2	1
八代橋	4	2
西山橋	3	2
須崎東	1	1
高知 IC 駐車場	1	1
土佐 SIC 下り	0	1
土佐 SIC 上り	0	1

※ 運転手 1 名 トラブル対応等 臨機応変に対応

4-2-8 当日の動き

調査当日の車両の動きを示す。2 台の車に計 10 名で調査地に向った。

・車両 1〔6 名〕

行き：土佐山田→高知 OV→高知 IC 駐車場→いの OV→土佐 SIC くだり→調査

帰り：調査→土佐 SIC のぼり→いの OV→高知 IC 駐車場→高知 OV→土佐山田

・車両2〔4名〕

行き：土佐山田→須崎→土佐 SIC のぼり→土佐 OV→調査

帰り：調査→土佐 OV→土佐 SIC くだり→須崎→土佐山田

4-3 ビデオ調査結果

西山橋に設置するカメラ1台を除き調査時刻までに設置が完了した。土佐OVのカメラに関して開始5分のデータが欠損した。また、高知OVについては調査中にカメラアングルが変わりデータが50分欠損した。また、土佐OVのカメラについても録画容量不足が調査中に発生し調査終了前10分が欠損した。須崎東については想定の通りカメラ画質の関係でナンバープレートが読み取れないという結果となった。その他の地点のカメラについては問題なくデータを取得した。

4-4 集計方法

集計用のシートにカメラより取得可能なデータを集計する。集計時の作業マニュアルを図24図25図26に示す。調査で取得したビデオデータより、車両の通過時刻、ナンバー、色、走行位置、その他記録可能な項目を記録する。時刻は図中の赤線に車両が重なった時の動画の時刻を記録した。集計作業はすべてのカメラでなく対象区間上り線の発生集中交通量算出に必要な5つのビデオカメラと土佐SICの流入のデータを中心に行った。

調査結果の集計作業



土佐OV 「04_土佐OV-②」

ビデオデータから
下記の集計表を作成する

必ず記録

確認できる場合は記録

ID	時刻 時 分 秒	方向	ナンバー	色	車種	プレートの色 白、黄、緑、黒	その他	origin	destination	レーン
0	0 0 55	上り	1 6 4 6	イエロー・ゴールド	軽自動車	黄		土佐SIC	土佐IC	減速車線
1	0 1 14	上り	2 9 3 3	白・シルバー	トラック	白		-	-	走行車線
2	0 1 14	上り	7 9 8 2	白・シルバー	不明	白		-	-	追越車線
3	0 1 20	上り	9 0 3 0	グレー	不明	白		-	-	走行車線
4	0 2 6	上り	7 5 7 6	オレンジ	軽自動車	黄		-	土佐IC	減速車線
5	0 2 6	上り	8 6 9 5	青系	不明	白		-	-	走行車線
6	0 2 10	上り	1 2 0 7	白・シルバー	不明	白		-	-	走行車線
7	0 2 11	上り	9 7 3 0	白・シルバー	軽自動車	黄		-	-	追越車線
8	0 2 12	上り	5 1 2 8	グレー	不明	白		-	土佐IC	減速車線
9	0 2 15	上り	6 6 7 1	白・シルバー	軽自動車	黄		-	土佐IC	減速車線
10	0 2 18	上り	2 9 5 6	白・シルバー	不明	白		-	-	走行車線

図 24 集計マニュアル 1

調査結果の集計作業

ID	時刻 時 分 秒	方向	ナンバー	色	車種	プレートの色 白、黄、緑、黒	その他	origin	destination	レーン
0	0 0 55	上り	1 6 4 6	イエロー・ゴールド	軽自動車	黄		土佐SIC	土佐IC	減速車線
1	0 1 14	上り	2 9 3 3	白・シルバー	トラック	白		-	-	走行車線
2	0 1 14	上り	7 9 8 2	白・シルバー	不明	白		-	-	減速車線
3	0 1 20	上り	9 0 3 0	グレー	不明	白		-	-	走行車線

車両の色を記録
リスト内から近いものを選択

色	色名	色コード
白	白	FFFFFF
黒	黒	000000
赤	赤	FF0000
青	青	0000FF
黄	黄	FFFF00
緑	緑	00FF00
紫	紫	800080
茶	茶	8B4513
金	金	FFD700
銀	銀	C0C0C0
銅	銅	CD853F
鉄	鉄	4682B4
鉛	鉛	483D8B
水	水	4682B4
紺	紺	000080
黒	黒	000000
白	白	FFFFFF
赤	赤	FF0000
青	青	0000FF
黄	黄	FFFF00
緑	緑	00FF00
紫	紫	800080
茶	茶	8B4513
金	金	FFD700
銀	銀	C0C0C0
銅	銅	CD853F
鉄	鉄	4682B4
鉛	鉛	483D8B
水	水	4682B4
紺	紺	000080
黒	黒	000000
白	白	FFFFFF

時刻を記録
車両が赤い線に重なった時のビデオ上時刻を記録

ナンバーとプレートの色を記録
プレートから読み取り

走行しているレーンを記録
左から「減速車線、走行車線、追越車線」
2つのレーンに跨って走行している場合
⇒走行車線を優先的に記録
減速車線と記録した車両のみ、調査場所を記録

図 25 集計マニュアル 2

調査結果の集計

高知OV 「01_高知OV-②」

いのOV 「02_いのOV-①」

高知IC 「02_高知IC駐車場」

須崎東 「05_須崎東IC」

図 26 集計マニュアル 3

4-5 トラフィックカウンターデータ

調査当日のトラフィックカウンターデータを nexco 西日本より頂いた。トラフィックカウンター(以降、トラカンと表記)とは交通量の観測地点において通過する車両数を自動的に

計測するための観測機器で、今回の対象区間内には上り方向 4 つ、下り方向 4 つの合計 8 つのトラカンが設置されている。トラカン設置位置と調査地点関係を図 27 にしめす。

トラフィックカウンターで取得した速度のデータをシミュレーションの再現性確認の際の比較対象として使用した。のぼり方向に設置されているトラカン 4 地点分の速度分布の結果を図 28 に示す。各地点の平均速度を見てみると 122.6kp と 147.6kp は概ね 80km/h、132.4kp、135kp は概ね 85km/h と平均値に開きが見られた。すべてのトラカン設置位置は単路部にあり似た条件であるといえる。しかし、122.6kp 地点に関しては交通量が他地点に比べ多いことが最高速度の低下、平均速度の低下を招いたと考えられる。147.6kp 地点については、トラカン設置位置がトンネル内であるため平均速度が低下したのではないかと考えられる。

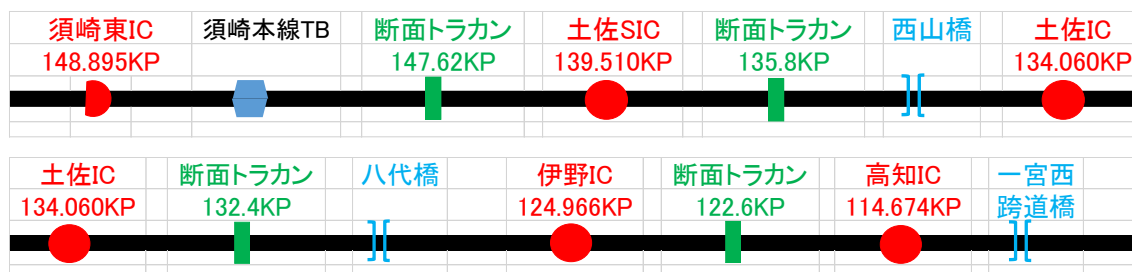


図 27 トラフィックカウンターと観測位置関係図

調査時間帯のデータ

高知ー伊野 122.6kp
 伊野ー土佐 132.4kp
 土佐ー土佐SIC 135.8kp
 土佐SICー須崎東 147.62kp

	122.6KP	132.4KP	135.8KP	147.62KP
平均	79.1629213	85.9217877	88.49122807	79.77841
標準偏差	6.22890836	8.11265135	7.301660807	6.757302
MAX	97	113	113	103
MIN	66	62	72	53

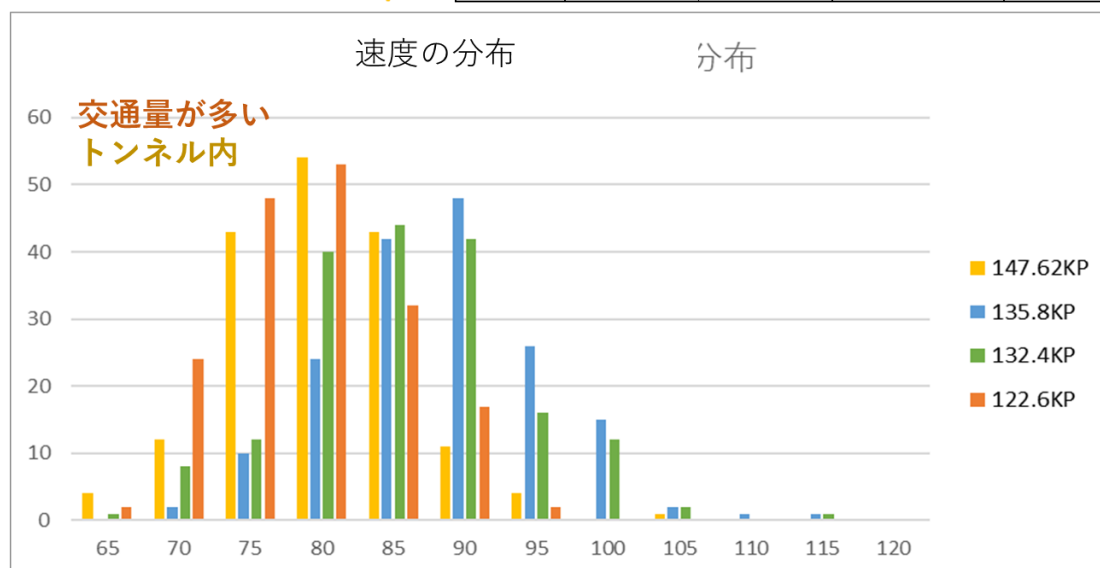


図 28 トラフィックカウンター4 地点の速度分布

4-6 OD表の作成

ビデオデータから集計した結果を用いて、OD表を作成する。

4-6-1 取得データの限界

今回の調査では、土佐 IC と伊野 IC からの流入は直接観測が出来なかったため、前後の区間の車両ナンバーを確認し、土佐 IC と伊野 IC からの流入と判定し、IC間の平均旅行時間で逆算しデータの欠損している部分を補うこととした。

4-6-2 OD作成時のルール

例えば、土佐 OV で観測なし、伊野 OV で観測ありの場合、流入可能な地点は土佐 IC からのみと判断できるので伊野 OV のみの観測データであっても、流入位置を特定しデータを補完している。流出方向に関しては次の地点で観測がない場合は観測地点の直後に流出したと判断した。須崎東から流入し、土佐 OV で観測され、伊野 OV でも観測されたが高知 IC 駐車場、高知 OV どちらにも観測がない場合は伊野 IC で流出したと判断する。ナンバーが同じで近い時間に同一地点を通過した場合は車両の色、ナンバープレートの色、車種をビデオより確認しマッチングを行う。

4-6-3 OD表

OD表でマッチングが行えた車両は7時30分から9時30分までの2時間のデータで1849台であった。7時30分から9時30分までのデータとした理由は研究対象区間通過にはSAにて休むことがなければ端から端まで約30分で通行が可能であり、流入流出のデータがすべての時間帯で概ねさう事から、開始から30分後を基点に2時間とした。作成したOD表の一部を表3に示す。プレートのナンバーが同じであっても色や観測地点と通過時刻に矛盾がない(平均的な通過に使用する時間±2分以内)場合に同一の車両と判断しIDを振り分ける。作成したOD表はシミュレーションソフトの入力データとして使用した。

表3 OD表の一部

時刻 時 分 秒	方向	ナンバー				色	車種	プレートの色	その他	origin	destination	レーン	観測地点	ID
		1桁目	2桁目	3桁目	4桁目									
1 2 14	上り	9	9	8	2	青系	軽自動車	黄	ラバン	須崎東IC	高知IC	走行車線	01須崎東	4
1 8 51	上り	9	9	8	2	青系	軽自動車	黄	ラバン	須崎東IC	高知IC	走行車線	03土佐OV	4
1 20 7	上り	9	9	8	2	青系	軽自動車	黄	ラバン	須崎東IC	高知IC	走行車線	04伊野OV	4
1 27 45	いの方向 (IC)	9	9	8	2	青系	軽自動車	黄	ラバン	須崎東IC	高知IC	流出(IC)	05高知IC流出	4
1 21 14	上り	8	8	5	5	白・シルバー	軽自動車	黄		土佐IC	高知道端部	走行車線	04伊野OV	6
1 31 43	上り	8	8	5	5	白・シルバー	軽自動車	黄		土佐IC	高知道端部	走行車線	07高知OV	6
0 38 5	高松方向 (IC)	7	0	0	7	青系	軽自動車	黄		土佐IC	高知IC	流出(IC)	05高知IC流出	10
2 4 46	いの方向 (IC)	7	0	0	7	白・シルバー	トラック	緑		伊野IC	高知IC	流出(IC)	05高知IC流出	13
2 49 17	上り	7	0	0	7	白・シルバー	-	白	BMW	須崎東IC	土佐IC	減速車線	03土佐OV	14
1 45 1	上り	5	9	7	4	白・シルバー	ワゴン	白		土佐IC	高知IC	走行車線	04伊野OV	19
1 52 35	高松方向 (IC)	5	9	7	4	白・シルバー	ワゴン	白		土佐IC	高知IC	流出(IC)	05高知IC流出	19

4-7 シミュレーションの再現性確認

道路交通センサスより、全国の山間地の自動車専用道路の大型車混入率は約 20%である。為遅い車両と早い車両の混在比率を 20 : 80 で固定した。その上でパラメータの変更項目は車両の希望走行速度や巡航時速度の速度幅の許容値といったパラメータとした。調査日とシミュレーション結果で再現性を確認する値は付加車線区間と単路区間を有する 3 区間の速度とした。詳細を以下に示す。

4-7-1 シミュレーションのチューニングに使用したパラメータ

■重点パラメータ

本研究は総旅行時間に着目した研究であり、旅行時間の再現性が高いことが望ましいが、比較可能なデータがないため、区間平均速度の再現性が高ければ高知道を再現できたものとする。区間は付加車線区間と単路区間どちらも有する土佐 SIC-土佐 IC 区間、土佐 IC-伊野 IC 区間、伊野 IC-高知 IC 区間とした。シミュレーション結果の平均速度の値と、区間に対応するトラカン 3 地点（須崎東は単路部のみなので今回は対象としない）の平均速度と比較する。

■変更パラメータ

変更するのは、早い車両、遅い車両の走行挙動に関わる車両情報である。変更したパラメータは速度に関するパラメータのみで表 4 に示す。早い車両と遅い車両はシミュレーションソフトに用意されていた車両の情報を使用し早い車両、遅い車両の初期値とした。

表 4 変更した車両パラメータ一覧

項目	説明	単位
DESIREDSPDFACTOR	希望速度係数（リンクの自由流速度に対して）	
DESIREDSPDFACTORSTDEV	希望速度係数標準偏差	
INSENSITIVESPDRANGE	無意識の速度低下不感帯	[km/h]
SPDDIFFTOLERANCE	速度低下許容幅	[km/h]

4-7-2 シミュレーションのチューニング結果

再現度の高かったケースの車両の走行パラメータを表 5 に示す。また、3 区間の再現度の結果を表 6 に示す。

表 5 キャリブレーション後の車両の走行パラメータ

項目	説明	遅い車両	早い車両	単位
DESIREDSPDFACTOR	希望速度係数（リンクの自由流速度に対して）	1.2	1.4	
DESIREDSPDFACTORSTDEV	希望速度係数標準偏差	0.02	0.03	
INSENSITIVESPDRANGE	無意識の速度低下不感帯	5	5	[km/h]
SPDDIFFTOLERANCE	速度低下許容幅	5	5	[km/h]
MAXACCELERATION	最大加速度	2.4	3.2	[m/s ²]
MAXDECELERATION	最大減速度	3.6	3.6	[m/s ²]
MAXACCELSTDEV	最大加速度標準偏差	0.2	0.2	[m/s ²]
MAXACCELERATION_LOWERLIMIT	最大加速度下限値	0.6	0.6	[m/s ²]
MAXACCELERATION_UPPERLIMIT	最大加速度上限値	4	4	[m/s ²]
DESIREDACCELERATION	希望加速度	1.304	1.304	[m/s ²]
DESIREDACCELSTDEV	希望加速度標準偏差	0.125	0.125	[m/s ²]
MAXDECELERATION_LOWERLIMIT	最大減速度標準偏差	0.2	0.2	[m/s ²]
MAXDECELERATION_UPPERLIMIT	最大減速度下限値	4	4	[m/s ²]
DECELERATIONGAIN	最大減速度上限値	1	1	[m/s ²]
DESIREDDECELERATION	希望減速度	1.623	1.623	[m/s ²]
DESIREDDECELSTDEV	希望減速度標準偏差	0.24	0.24	[m/s ²]
IDMDELTA	IDMのデルタパラメータ	4	4	
GRADIENTCOEF	勾配に対する係数	1	1	
FORESIGHTDURATION	前方探査距離	10	10	[m]
SPDDIFFCRUISE	巡航時速度幅	0	0	[km/h]
MEANDISTTOTURN	右左折の際に何m手前で車線変更するか（期待値）	300	300	[m]
STDVDISTTOTURN	右左折の際に何m手前で車線変更するか（標準偏差）	50	50	[m]
MINDISTTOTURN	右左折の際に何m手前で車線変更するか（最小値）	100	100	[m]
MOVEPASSINGLANEPROB	追越希望受諾確率	0.2	0.75	
RETURNCRUISINGLANEPROB	走行車線復帰確率	0.1	0.1	
MINGAPTIME	最小ギャップ	2	1.5	[s]
MINTTC	最小TCC(Time-To-Collision、前方車両との距離を速度の差（相対速度）	2	2	[s]
UNSAFEDECELERATION	車線変更時に後続車の減速度がこれ以上になる場合は車線変更しない	-30	-30	[km/h/s]

表 6 速度の再現性

速度[km/h]	土佐SIC-土佐IC		土佐IC-伊野IC		伊野IC-高知IC	
	トラカン	シミュレーション	トラカン	シミュレーション	トラカン	シミュレーション
平均	88.57542	84.57469542	85.92179	82.91351703	79.23464	68.36531523
中央	88	85.77175804	85	71.00323269	79	67.81218552
最高	113	99.97108195	113	137.6787261	97	117.9111682
最低	72	59.998225	62	51.36084782	66	21.79611252
標準偏差	7.31465	13.44814547	8.135408	25.25517381	6.302374	16.68632045
平均の再現度	95.48%		82.64%		86.28%	

4-7-3 課題・限界・留意点

平均速度については概ね調査日のトラフィックカウンターデータと合う結果が得られた。しかし、最低速度や最高速度では平均に比べ大きな開きがある、伊野 IC-高知 IC 間では伊野 IC からの流入台数が多く合流部での減速の影響により最低速度が 20km/h と低い値となったと考えられる。

5. 追い越し効果の計算

再現性の確保できたシミュレーションを用い追い越し効果を計算する。

5-1 追い越し効果の計算方法

シミュレーションの出力結果(1秒刻みの車両走行位置のデータ)を元に追い越しの効果
を計算する。計算はpythonで作成したコードを用いて計算する。1秒刻みのままでは計算
に要する時間が膨大となる為、5秒刻みにデータに間引いた結果を用いて追い越し効果の
計算を実施した。

5-1-1 計算フロー

追い越し効果の計算をする際のフローチャートを図 29 に示す。

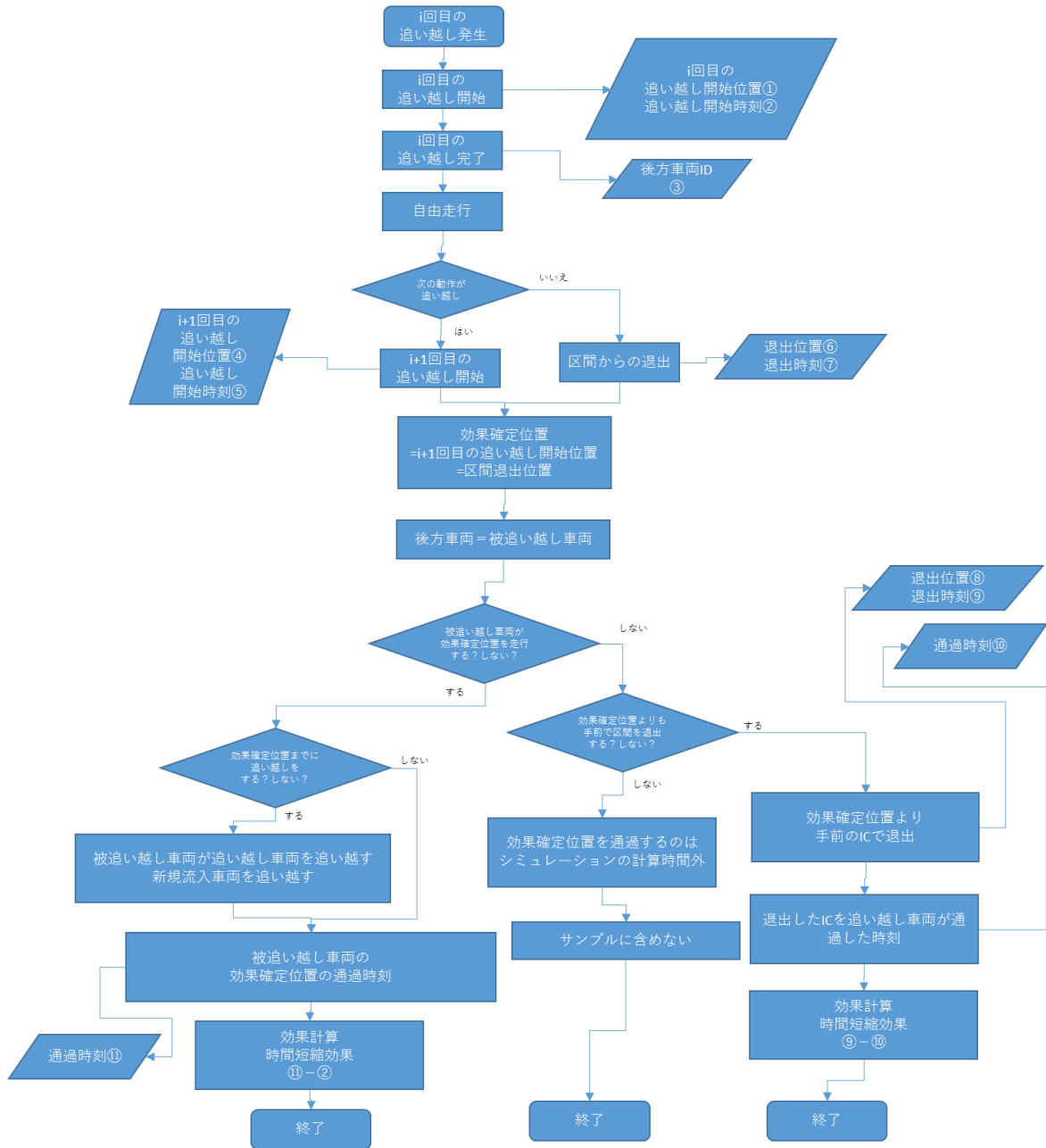


図 29 追い越し効果計算フロー

5-1-2 追い越し効果計算の3つのパターン

計算の過程が異なる3つのパターンについて図39を用いて紹介する。図のパターン1は車両Cが追い越しを行いその後区間を退出した場合で区間退出位置を通過する際の後方車両Aとの通過時刻の差をCの追い越しの旅行時間短縮と考え、追い越し効果とする。パターン2は車両Cが追い越しを行いその後追い越しをもう一度行う場合で最初の追い越しは次の追い越し実施によりひとつの追い越しの効果は確定され次の追い越し実施位置を后方車両Aが通過した時刻の差をCの追い越し効果とする。パターン3は車両Cが次の追い越しも区間退出もしていないが、后方車両Aが区間退出した場合で、追い越しをしなかった場合のC'を見てわかるとおり、前方にいたはずの車両が区間退出により存在しなくなるので、車両Cは希望走行速度(追い越し後の速度)で走行する。Aの区間退出以降旅行時間の増減は起きない為、パターン3については、后方車両Aの区間退出位置で追い越し効果を計算する。

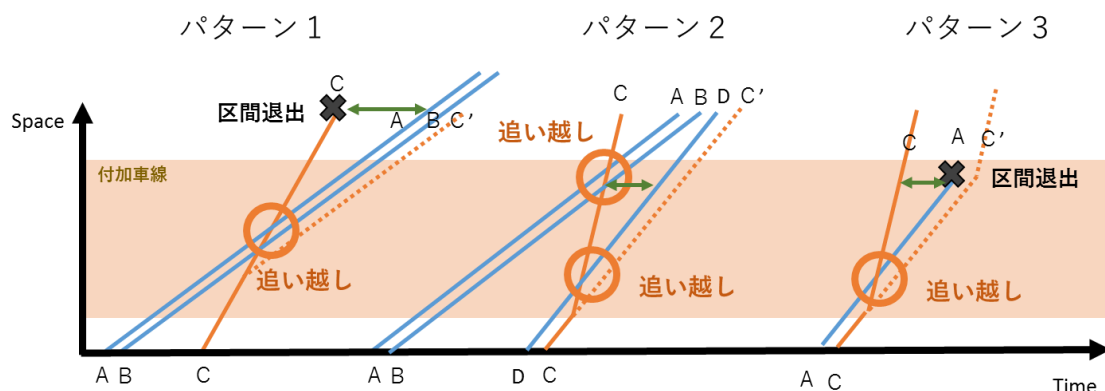


図 30 追い越し効果の3つのパターン

5-1-3 追い越し効果の見方

本研究で問題としている追い越しは追い越しを実施したにもかかわらず旅行時間の短縮が起きない追い越しである。前述の追い越しが発生する2つのパターンを紹介する。ひとつは追い越しした後、付加車線区間追い越しをされた車両に追いつかれる様な場合。もうひとつは追い越しを実施した車両が追い越された車両に追い越される場合である。1つ目に関しては追い越しした台数分旅行時間が短縮されているとも見えるが、本研究ではその影響を消す為追い越し効果の結果から-5秒した結果を基準に0秒より大きい追い越しと0秒以下の追い越しで分けて効果的か無意味な追い越しが判別した。追い越し効果が0秒以下の追い越し発生イメージを図31に示す。このような場合、追い越しの効果は限りなく0秒に近づき0秒以上になることはない。

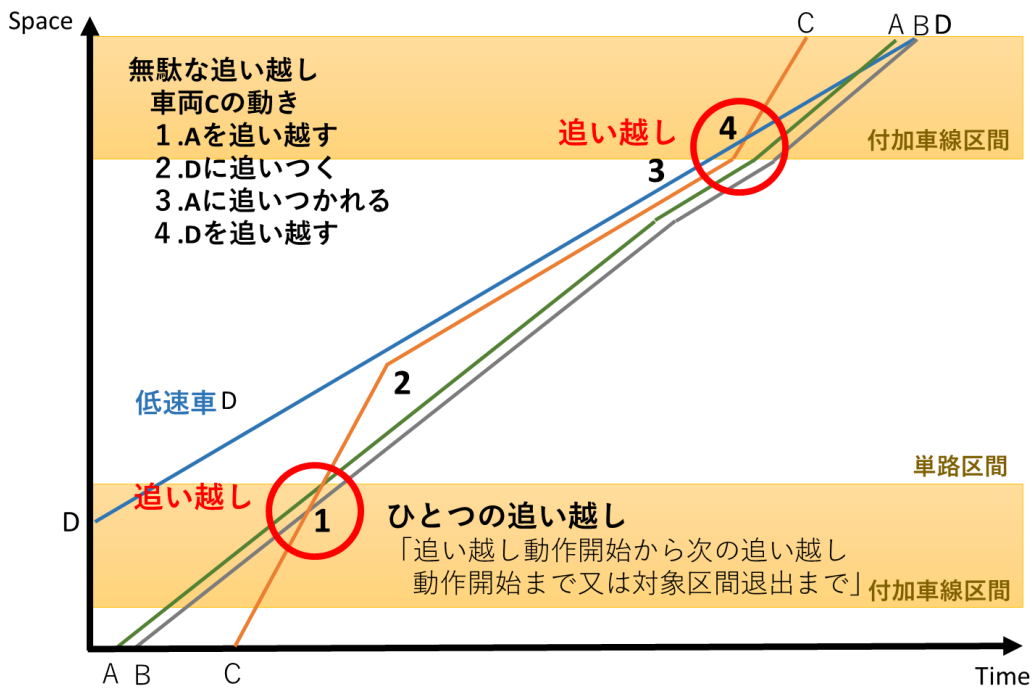


図 31 無駄な追い越しの発生状況

5-2 追い越し効果の計算結果考察

追い越し効果の計算結果を紹介する。

5-2-1 概要

3 回分の計算結果を紹介する。全車両は全体 1 8 4 9 台、総追い越し発生回数は 9 7 0 回、追い越し効果が 0 秒以下の追い越しが 5 2 5 回、追い越し効果が 0 秒より大きい追い越しが 4 4 5 回、追い越し車両台数が 5 5 9 台であった。すべての追い越しのうち 56%が本研究でなくしたい追い越しであり、この結果より実際の高速道路上でも 5 秒に満たない追い越し効果の追い越しが半数程度発生している可能性が示唆できる。

5-2-2 追い越し効果の分布

追い越し効果の分布を図 32 に示す。追い越し効果が 0 秒の追い越しが全体の追い越しの約半数を占めている。追い越し効果が負の大きい値である追い越しは少なく、ほとんどの追い越しが追い越し効果 0 秒付近に集中していることがわかる。

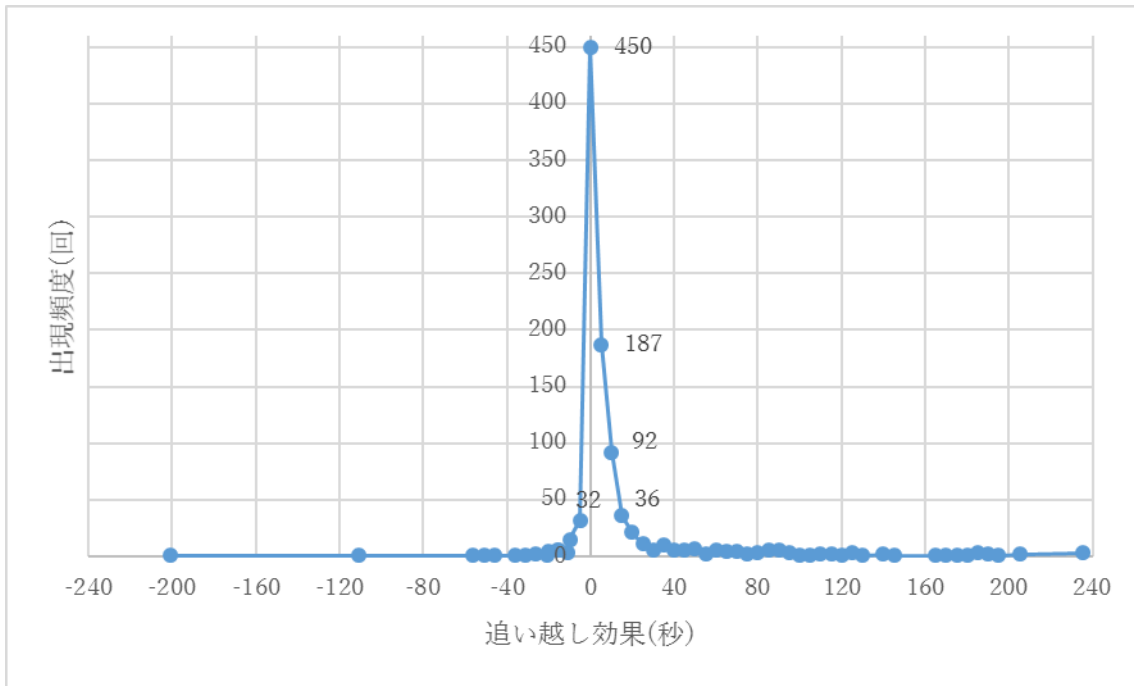


図 32 追越し効果の分布

5-2-3 追越し発生位置

追越し開始位置と追越し効果の結果を図 33 に示している。高知 IC での追越しは残りの区間がすべて片側 2 車線確保されているため追越しこうが 0 秒以下のものが存在しない。また、のこり区間が短いことから追越し効果が小さい結果となっている。土佐 SIC や土佐 IC での追越しは伊野 IC での追越しに比べ追越し効果の結果に幅がある、これは伊野 IC 以降の区間は交通量が多く前方車両に追いついてしまう可能性が高く、追越しによって得られる旅行時間の短縮量にばらつきが少ないことが原因だと考えられる。

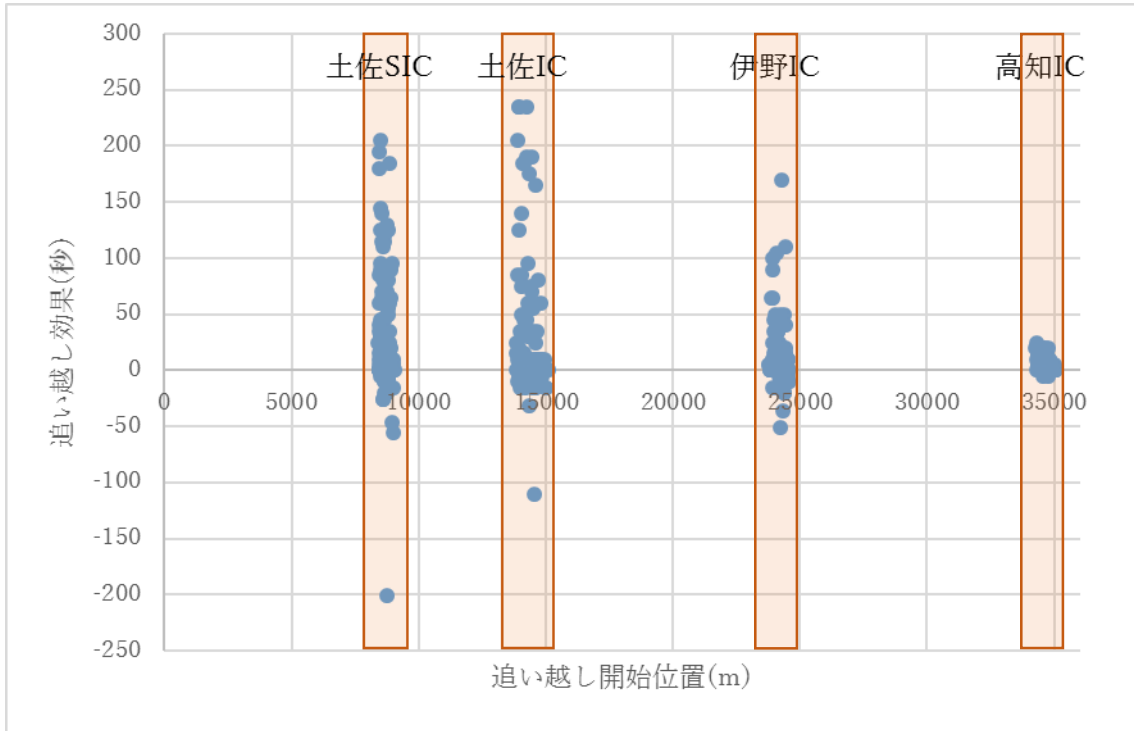


図 33 追い越し発生位置

6. 追い越し実施基準

追い越し効果が 0 秒以下と追い越し効果が 0 秒より大きい 2 つの追い越しの発生条件を追い越し実施時の周辺環境や自車の走行状態によって判別する。

6-1 判別分析（数量化Ⅱ類）による追い越しの効果と周辺環境，自車走行状況の関係性把握判別分析に使用する変数を紹介する。

6-1-1 説明変数

5 秒刻みのシミュレーション結果より以下の情報を抽出する。説明変数は大きく 2 つのグループに分けることが出来る。ひとつは追い越し実施時の車両に関する情報，もうひとつは追い越し実施時の周辺状況である。

・ 追い越し実施車両の情報

追い越し車両速度：追い越し開始時の速度

追い越し開始位置：追い越しを開始した位置

追い越し確定位置：追い越し効果の確定した位置

走行距離：追い越し確定位置から追い越し開始位置の差でひとつの追い越しの走行距離

・ 追い越し時の周辺状況

密度：追い越し実施時の同一区間に存在する車両台数と区間長より求めた密度(台/m)

前方車速度：前方車両の速度(2 台前まで記録)

後方車速度：後方車両の速度(2 台後ろまで記録)

前方車位置：前方車両の位置(2 台前まで記録)

後方車位置：後方車両の位置(2 台後ろまで記録)

付加車線区間の残り距離：追い越し実施位置から付加車線の端部までの距離

単路区間の残り距離：付加車線の先にある単路区間距離

6-1-2 目的変数

追い越し効果

追い越し効果(秒)で記録した結果を一時的に - 5 秒した後の結果を，0 秒以下のもの，0 秒より大きいもの 2 つに判別する。

6-2 判別分析の結果

6-2-1 高知道上り方向の追い越し効果判別(走行距離を含む)

説明変数に走行距離を含んだ場合の判別分析の結果を表 7～表 10 に示す。

説明変数は 10 個の変数が選ばれ，すべて 5%以内で有意な変数である。相関比 0.21 で判

別の中率が 72%の判別モデルが作成できた。

変数毎に結果を紹介する。残った変数のうち追い越し車両に関する変数は走行距離のみであった。係数がマイナスであることから、ひとつの追い越しの走行距離が大きくなるほど、追い越しによる旅行時間の短縮発生しないことを示している。速度に関する変数について見てみると、追い越し実施車両の前後 2 台の速度が追い越しに寄る旅行時間に影響する変数であることがわかる。中でも 1 台後方車両の速度の係数が 0.037 であることから他の速度の変数に比べ非常に結果への影響力が大きいことがわかる。一台後方車両の速度が速いほど、追い越しによる旅行時間の短縮は少なくなるということが言える。車両の距離差の変数について見てみると、前方車両との距離の差はマイナスが入力値になるため、すべての距離の差の変数は計算後マイナスになることがわかる。ここでも 1 台後方車両の変数の係数が他の変数より大きく影響が大きいことがわかる。車両同士の距離が離れているほど追い越し効果が 0 秒より大きくなるということを示している。付加車線の残り距離について見て見ると、係数がマイナスであり、付加車線の残り距離が長いほど、旅行時間の短縮は 0 秒より大きくなることが多いことを示している。同一の付加車線内で次の追い越しを実施する場合追い越し効果は 0 秒より大きくなる、付加車線の残り距離が長いということは、同一付加車線区間内の追い越しが発生しやすいためだと考えられる。単路区間の残り距離について見てみると、係数がプラスであることから、単路区間が長くなれば、旅行時間の短縮は 0 秒以下になることを示している。これは、単路区間が長くなれば単路区間での車群の発生による追従が発生する可能性が高くなるため、この結果は妥当だと考えられる。

表 7 判別分析の結果(説明変数)

変数の数	相関比 η^2	Wilks' lambda	F 値	自由度1	自由度2	P 値
10	0.2123	0.7877	45.3147	10	1681.00	P < 0.001
判別関数に含まれる変数						
変 数	Wilks' lambda	F 値	自由度1	自由度2	P 値	* : P<0.05 ** : P<0.01
03_1台前方の車両速度[km/h]	0.9967	5.5123	1	1681	0.0190	*
03_2台前方の車両速度[km/h]	0.9911	15.0819	1	1681	P < 0.001	**
03_1台後方の車両速度[km/h]	0.9448	98.2762	1	1681	P < 0.001	**
03_2台後方の車両速度[km/h]	0.9977	3.9142	1	1681	0.0480	*
05_1台前方の車両との距離差[m]	0.9897	17.4246	1	1681	P < 0.001	**
05_2台前方の車両との距離差[m]	0.9970	5.0979	1	1681	0.0241	*
05_1台後方の車両との距離差[m]	0.9123	161.6115	1	1681	P < 0.001	**
07_付加車線区間の残り距離[m]	0.9676	56.3369	1	1681	P < 0.001	**
10_単路区間の残り距離[m]	0.9942	9.7332	1	1681	0.0018	**
10_走行距離[m]	0.9931	11.6108	1	1681	P < 0.001	**

表 8 判別分析の結果（判別的中率）

判別結果		予測値		
		追い越し効果 ≤ 0秒	追い越し効果 > 0秒	判別的中率
観測値	追い越し効果 ≤ 0秒	708	207	77.38%
	追い越し効果 > 0秒	266	511	65.77%
			全 体	72.04%

表 9 判別分析の結果（説明変数の判別係数）

判別係数	
変 数	関数1
03_1台前方の車両速度[km/h]	-0.0068
03_2台前方の車両速度[km/h]	0.0092
03_1台後方の車両速度[km/h]	0.0377
03_2台後方の車両速度[km/h]	-0.0053
05_1台前方の車両との距離差[m]	0.0009
05_2台前方の車両との距離差[m]	0.0003
05_1台後方の車両との距離差[m]	-0.0415
07_付加車線区間の残り距離[m]	-0.0022
07_単路区間の残り距離[m]	0.0002
10_走行距離[m]	-0.0001
定数項	-0.0760

表 10 判別分析結果(各群重心)

各群の重心	
00_追い越しの効果	関数1
0秒以下	0.4782
0秒より大きい	-0.5631

表 11 判別分析（基本統計量）

基本統計量							
目的変数	変数	n	平均	不偏分散	標準偏差	最小値	最大値
全 体	03_1台前方の車両速度[km/h]	1692	66.951	426.345	20.648	0.000	139.959
	03_2台前方の車両速度[km/h]	1692	69.968	539.618	23.230	11.802	139.962
	03_1台後方の車両速度[km/h]	1692	63.378	240.918	15.522	0.801	100.000
	03_2台後方の車両速度[km/h]	1692	72.259	471.448	21.713	8.248	139.702
	05_1台前方の車両との距離差[m]	1692	-124.429	122277.313	349.682	-3876.167	-0.006
	05_2台前方の車両との距離差[m]	1692	-268.609	275245.718	524.639	-3925.951	-2.613
	05_1台後方の車両との距離差[m]	1692	22.993	300.898	17.346	0.007	107.676
	07_付加車線区間の残り距離[m]	1692	545.941	48259.014	219.679	1.259	1061.207
	07_単路区間の残り距離[m]	1692	7170.301	4302160.098	2074.165	4387.150	9337.380
	10_走行距離[m]	1692	14140.249	28147579.852	5305.429	8343.848	25305.544
損	03_1台前方の車両速度[km/h]	915	65.334	215.473	14.679	20.800	139.959
	03_2台前方の車両速度[km/h]	915	71.964	516.341	22.723	19.587	139.962
	03_1台後方の車両速度[km/h]	915	65.620	236.577	15.381	3.409	100.000
	03_2台後方の車両速度[km/h]	915	71.032	424.260	20.598	8.248	139.702
	05_1台前方の車両との距離差[m]	915	-46.547	14690.596	121.205	-2514.218	-0.006
	05_2台前方の車両との距離差[m]	915	-155.804	103014.688	320.959	-3294.701	-6.815
	05_1台後方の車両との距離差[m]	915	18.086	190.376	13.798	0.014	88.759
	07_付加車線区間の残り距離[m]	915	531.532	55373.871	235.317	1.259	1034.167
	07_単路区間の残り距離[m]	915	7264.346	4131707.903	2032.660	4387.150	9337.380
	10_走行距離[m]	915	14177.983	27753900.455	5268.197	8343.848	25048.255
得	03_1台前方の車両速度[km/h]	777	68.854	668.558	25.856	0.000	139.941
	03_2台前方の車両速度[km/h]	777	67.617	557.496	23.611	11.802	139.810
	03_1台後方の車両速度[km/h]	777	60.736	233.426	15.278	0.801	100.000
	03_2台後方の車両速度[km/h]	777	73.703	523.772	22.886	11.802	137.739
	05_1台前方の車両との距離差[m]	777	-216.143	233579.794	483.301	-3876.167	-0.074
	05_2台前方の車両との距離差[m]	777	-401.449	445786.586	667.673	-3925.951	-2.613
	05_1台後方の車両との距離差[m]	777	28.770	369.657	19.226	0.007	107.676
	07_付加車線区間の残り距離[m]	777	562.909	39407.958	198.514	47.569	1061.207
	07_単路区間の残り距離[m]	777	7059.553	4485758.961	2117.961	4387.150	9337.380
	10_走行距離[m]	777	14095.814	28643885.940	5351.998	8352.812	25305.544

6-2-2 追い越し効果の判別（走行距離を含まない）

走行距離の変数は追い越し効果の確定位置が付加車線区間、IC の退出に限られるため、高知道の影響を大きく受けた変数である可能性が高いため、説明変数に走行距離を含まない場合の判別分析を実施した。結果を表 12～表 16 表 10 に示す。

説明変数は 6 個の変数が選ばれ、すべて 5%以内で有意な変数である。相関比 0.25 で判別中率が 76%の判別モデルが作成できた。

変数毎に結果を紹介する。残った変数のうち追い越し車両に関する変数は走行速度のみであった。係数がプラスであることから、追い越し時の速度が速いほど旅行時間の短縮は大きくなることを示している。他の速度に関する変数について見てみると、1 台後方の車両速度のみが有意という結果であった。1 台後方車両の速度の係数が-0.047 であり、追い越し車

両速度の係数が 0.033 であることから、追い越し時の速度差がない場合には後方車両の速度の影響が大きくなるため、追い越し効果が 0 秒以下になる場合が多いことがいえる。密度について見てみると、判別係数は 2.65 となっている。密度が高くなるほど、追い越し効果が 0 秒より大きい結果が増えることがいえる。密度が高い場合、追い越しを実施しても車両に追いつくことが増えることも増加すると考えられる。今回、密度の判別係数がプラスになった要因として考えられることは、密度が高いということは走行している車両が多い状況である。車両が多い場合には追い越し実施車両以外の追い越し車両が多数存在し、単路部で追い越した車両に追い越された車両が追いついたとしても、他の追い越し車両の存在が 2 車両間に複数台存在することが考えられる。追い越し効果計算は追い越し車両と追い越された車両の同一地点通過時刻を比較することで求めることから、2 車両間の距離が他の車両によって十分に大きく取られる場合、同一車群に追いついていたとしても旅行時間の短縮量は一定量保存されていることにより係数がプラスになったと考えられる。車両の距離差の変数について見てみると、前方車両との距離の差はマイナスが入力値になるため、すべての距離の差の変数は計算後プラスになることがわかる。ここでも 1 台後方車両との距離差と 1 台前方車両との距離差と比べると係数の大小関係より一台後方車両の影響が大きいことがわかる。計算結果がプラスであることから、距離の差が大きいほど追い越し効果は 0 秒より大きくなることを示している。付加車線の残り距離について見て見ると、係数がプラスであり、付加車線の残り距離が長いほど、旅行時間の短縮は 0 秒より大きくなることを示している。同一の付加車線内で次の追い越しを実施する場合追い越し効果は 0 秒より大きくなる、付加車線の残り距離が長いということは、同一付加車線区間内の追い越しが発生しやすいためだと考えられる。

表 12 判別分析の結果(説明変数)

変数の数	相関比 η^2	Wilks' lambda	F 値	自由度1	自由度2	P 値
6	0.2570	0.7430	97.1639	6	1685.00	P < 0.001
判別関数に含まれる変数						
変 数	Wilks' lambda	F 値	自由度1	自由度2	P 値	* : P<0.05 ** : P<0.01
01_追い越し車両速度[km/h]	0.9189	148.7937	1	1685	P < 0.001	**
03_1台後方の車両速度[km/h]	0.8972	193.1454	1	1685	P < 0.001	**
04_密度[台/m]	0.9969	5.2083	1	1685	0.0226	*
05_1台前方の車両との距離差[m]	0.9591	71.9167	1	1685	P < 0.001	**
05_1台後方の車両との距離差[m]	0.9615	67.4293	1	1685	P < 0.001	**
07_付加車線区間の残り距離[m]	0.9974	4.4185	1	1685	0.0357	*

表 13 判別分析の結果（判別的中率）

判別結果		予測値		
		追い越し効果 ≤ 0 秒	追い越し効果 > 0 秒	判別的中率
観測値	追い越し効果 ≤ 0 秒	747	168	81.64%
	追い越し効果 > 0 秒	225	552	71.04%
			全 体	76.77%

表 14 判別分析の結果（説明変数の判別係数）

判別係数	
変 数	関数1
01_ 追い越し車両速度[km/h]	0.0333
03_1台後方の車両速度[km/h]	-0.0479
04_密度[台/m]	2.6537
05_1台前方の車両との距離差[m]	-0.0012
05_1台後方の車両との距離差[m]	0.0255
07_付加車線区間の残り距離[m]	0.0005
定数項	-1.4554

表 15 判別分析結果(各群重心)

各群の重心	
00_ 追い越しの効果	関数1
0秒以下	-0.5417
0秒より大きい	0.6379

表 16 判別分析（基本統計量）

基本統計量							
目的変数	変 数	n	平 均	不偏分散	標準偏差	最小値	最大値
全 体	01_追い越し車両速度[km/h]	1692	100.759	511.536	22.617	2.567	139.980
	03_1台後方の車両速度[km/h]	1692	63.378	240.918	15.522	0.801	100.000
	04_密度[台/m]	1692	0.043	0.002	0.042	0.002	0.125
	05_1台前方の車両との距離差[m]	1692	-124.429	122277.313	349.682	-3876.167	-0.006
	05_1台後方の車両との距離差[m]	1692	22.993	300.898	17.346	0.007	107.676
	07_付加車線区間の残り距離[m]	1692	545.941	48259.014	219.679	1.259	1061.207
損	01_追い越し車両速度[km/h]	915	94.649	547.563	23.400	2.567	139.980
	03_1台後方の車両速度[km/h]	915	65.620	236.577	15.381	3.409	100.000
	04_密度[台/m]	915	0.041	0.002	0.042	0.002	0.124
	05_1台前方の車両との距離差[m]	915	-46.547	14690.596	121.205	-2514.218	-0.006
	05_1台後方の車両との距離差[m]	915	18.086	190.376	13.798	0.014	88.759
	07_付加車線区間の残り距離[m]	915	531.532	55373.871	235.317	1.259	1034.167
得	01_追い越し車両速度[km/h]	777	107.956	373.882	19.336	5.094	139.915
	03_1台後方の車両速度[km/h]	777	60.736	233.426	15.278	0.801	100.000
	04_密度[台/m]	777	0.045	0.002	0.042	0.002	0.125
	05_1台前方の車両との距離差[m]	777	-216.143	233579.794	483.301	-3876.167	-0.074
	05_1台後方の車両との距離差[m]	777	28.770	369.657	19.226	0.007	107.676
	07_付加車線区間の残り距離[m]	777	562.909	39407.958	198.514	47.569	1061.207

6-3 追い越し実施基準の整理

追い越し実施基準を走行距離の変数を含まない判別分析の結果より作成する。判別分析より作成した結果を追い越し効果が0秒より大きいものを1、追い越し効果が0秒以下を-1として再度集計した。

追い越し車両速度と1台後方の車両速度について整理した追い越し実施基準を紹介する。基本統計量の結果より平均値に近い値を用いる。密度を0.04台/m、1台前方車両との距離差を-200m、1台後方の車両との距離差を20m、付加車線区間の残り距離を500mとした場合の結果を表17に示す。表の青は追い越しの効果が0秒より大きいため追い越しを実施したほうが旅行時間短縮がされるもの、赤は対称に旅行時間が0秒以下になるため追い越しを実施しても到着が早くなることを表している。

表 17 追い越し実施基準追い越し車両速度と 1 台後方の車両速度

		01_追い越し車両速度[km/h]									総計
		55	65	75	85	95	105	115	125	135	
03_1台後方の車両速度[km/h]	30	-1	1	1	1	1	1	1	1	1	0.777778
	35	-1	1	1	1	1	1	1	1	1	0.777778
	40	-1	-1	1	1	1	1	1	1	1	0.555556
	45	-1	-1	-1	1	1	1	1	1	1	0.333333
	50	-1	-1	-1	1	1	1	1	1	1	0.333333
	55	-1	-1	-1	-1	1	1	1	1	1	0.111111
	60	-1	-1	-1	-1	-1	1	1	1	1	-0.111111
	65	-1	-1	-1	-1	-1	-1	1	1	1	-0.333333
	70	-1	-1	-1	-1	-1	-1	1	1	1	-0.333333
総計	-1	-0.555556	-0.333333	0.111111	0.333333	0.555556	1	1	1	0.234568	

追い越し車両速度を 55km/h から 135km/h の範囲で、1 台後方の車両速度を 30km/h から 70km/h の範囲で、密度を 0.002 台/m から 0.120 台/m の範囲で、1 台前方車両との距離差を -1000m から 0m の範囲で、1 台後方の車両との距離差を 0m から 80m の範囲で、付加車線区間の残り距離を 100m から 900m の範囲で変更させた場合の判別結果を以下の表に示す。範囲についてはシミュレーションの出力結果の最大値と最小値を元に決定した。網羅的に計算したため、実際には発生しないケースも表には含まれている。表のグラデーションにメリハリがあるものはメリハリがない表に比べ計算結果に強く影響することが視覚的に把握できる。

追い越し実施基準追い越し車両速度と 1 台後方の車両速度について整理したものを表 18 に示す。追い越し実施基準追い越し車両速度と密度について整理したものを表 19 に示す。追い越し実施基準追い越し車両速度と 1 台前方の車両との距離差について整理したものを表 20 に示す。追い越し実施基準追い越し車両速度と 1 台後方の車両との距離差について整理したものを表 21 に示す。追い越し実施基準追い越し車両速度と付加車線区間の残り距離について整理したものを表 22 に示す。

表 18 追い越し実施基準追い越し車両速度と 1 台後方の車両速度

		01_追い越し車両速度[km/h]									総計
		55	65	75	85	95	105	115	125	135	
03_1台後方の車両速度[km/h]	30	0.696845	0.868313	0.965249	0.998171	1	1	1	1	1	0.94762
	35	0.529492	0.754001	0.903521	0.98171	1	1	1	1	1	0.907636
	40	0.336534	0.599451	0.802926	0.932327	0.99177	1	1	1	1	0.851445
	45	0.126658	0.414266	0.663466	0.84545	0.95519	0.995885	1	1	1	0.777879
	50	-0.08139	0.208505	0.487426	0.723823	0.884774	0.97348	0.999543	1	1	0.688462
	55	-0.29081	-0.0032	0.289895	0.561043	0.776406	0.915866	0.987654	1	1	0.581873
	60	-0.49063	-0.21125	0.079561	0.369913	0.627801	0.820302	0.942844	0.993141	1	0.459076
	65	-0.66529	-0.41655	-0.12803	0.16278	0.444444	0.691358	0.86374	0.964335	0.997714	0.323833
	70	-0.80658	-0.60128	-0.33699	-0.04801	0.243713	0.520805	0.748057	0.898491	0.980338	0.177615
総計	-0.07169	0.179139	0.414114	0.614134	0.769344	0.879744	0.949093	0.983996	0.997561	0.635049	

表 19 追い越し実施基準追い越し車両速度と密度

		01_追い越し車両速度[km/h]									
		55	65	75	85	95	105	115	125	135	総計
04_密度[台/m]	0.002	-0.16735	0.085048	0.329218	0.545953	0.717878	0.84545	0.929584	0.975309	0.99497	0.584007
	0.01	-0.15135	0.100594	0.344765	0.557842	0.727481	0.851395	0.933242	0.97668	0.995428	0.592897
	0.015	-0.14037	0.110654	0.353909	0.565615	0.733425	0.855967	0.935528	0.978052	0.995885	0.59874
	0.02	-0.13123	0.120713	0.363054	0.573388	0.738912	0.859625	0.938272	0.978967	0.996342	0.604227
	0.04	-0.09191	0.160951	0.399634	0.602652	0.76086	0.875171	0.946959	0.983082	0.998171	0.626175
	0.06	-0.05213	0.200732	0.434385	0.630544	0.782807	0.889346	0.954275	0.987197	0.998628	0.64731
	0.08	-0.01097	0.240055	0.467764	0.657979	0.802926	0.902149	0.961591	0.990398	0.999086	0.667886
	0.1	0.030178	0.278006	0.500686	0.684042	0.821216	0.91358	0.96845	0.992227	0.999543	0.687548
	0.12	0.069959	0.315501	0.533608	0.709191	0.838592	0.925011	0.973937	0.994056	1	0.70665
		総計	-0.07169	0.179139	0.414114	0.614134	0.769344	0.879744	0.949093	0.983996	0.997561

表 20 追い越し実施基準追い越し車両速度と1台前方の車両との距離差

		01_追い越し車両速度[km/h]									
		55	65	75	85	95	105	115	125	135	総計
05_1台前方の車両との距離差[m]	-1000	0.366865	0.58299	0.752477	0.877763	0.959153	0.994818	1	1	1	0.837118
	-800	0.199207	0.443682	0.643042	0.797897	0.90977	0.975004	0.998171	1	1	0.774086
	-600	0.01387	0.284865	0.514708	0.700655	0.840573	0.934766	0.987197	1	1	0.697404
	-400	-0.17086	0.106843	0.365341	0.58299	0.751562	0.877763	0.959153	0.994818	1	0.607512
	-200	-0.34278	-0.08246	0.197074	0.442463	0.641823	0.797592	0.90977	0.974699	0.998171	0.504039
	0	-0.49642	-0.26109	0.012041	0.283036	0.513184	0.698522	0.840268	0.934461	0.987197	0.390134
		総計	-0.07169	0.179139	0.414114	0.614134	0.769344	0.879744	0.949093	0.983996	0.997561

表 21 追い越し実施基準追い越し車両速度と1台後方の車両との距離差

		01_追い越し車両速度[km/h]									
		55	65	75	85	95	105	115	125	135	総計
05_1台後方の車両との距離差[m]	0	-0.85277	-0.67032	-0.40924	-0.107	0.203018	0.497028	0.736168	0.896205	0.979881	0.141442
	10	-0.71925	-0.47599	-0.18107	0.130316	0.428898	0.684499	0.863283	0.964792	0.998171	0.299294
	20	-0.54047	-0.2524	0.056699	0.362597	0.633745	0.828075	0.950617	0.99497	1	0.448204
	30	-0.32373	-0.01646	0.291724	0.572931	0.790581	0.92684	0.99177	1	1	0.581517
	40	-0.08779	0.221765	0.512117	0.747142	0.901235	0.982167	1	1	1	0.697404
	50	0.148148	0.449474	0.699588	0.874257	0.971193	0.999086	1	1	1	0.793527
	60	0.381344	0.646091	0.838134	0.954733	0.995428	1	1	1	1	0.868414
	70	0.592135	0.801097	0.935071	0.992227	1	1	1	1	1	0.924503
	80	0.757202	0.909008	0.983996	1	1	1	1	1	1	0.961134
		総計	-0.07169	0.179139	0.414114	0.614134	0.769344	0.879744	0.949093	0.983996	0.997561

表 22 追い越し実施基準追い越し車両速度と付加車線区間の残り距離

		01_追い越し車両速度[km/h]									
		55	65	75	85	95	105	115	125	135	総計
07_付加車線区間の残り距離[m]	100	-0.23823	0.026063	0.265661	0.490169	0.686786	0.812529	0.915409	0.967993	0.990855	0.54636
	200	-0.18793	0.064472	0.300412	0.532693	0.704161	0.831276	0.928669	0.970736	0.995428	0.571102
	300	-0.14586	0.093278	0.344765	0.565615	0.721994	0.856882	0.934614	0.975309	0.997257	0.593761
	400	-0.11294	0.134431	0.388203	0.584819	0.748514	0.874257	0.939643	0.984454	0.997714	0.615455
	500	-0.08048	0.184728	0.42021	0.610882	0.777321	0.882487	0.949703	0.987654	0.998171	0.636742
	600	-0.03567	0.228166	0.445816	0.647462	0.797897	0.892547	0.962048	0.989026	0.998628	0.658436
	700	0.014632	0.258802	0.479652	0.67947	0.809328	0.909922	0.967535	0.990398	1	0.67886
	800	0.056241	0.290352	0.523091	0.700046	0.826246	0.925011	0.970279	0.993599	1	0.698318
	900	0.085048	0.331962	0.559214	0.716049	0.851852	0.932785	0.973937	0.996799	1	0.716405
		総計	-0.07169	0.179139	0.414114	0.614134	0.769344	0.879744	0.949093	0.983996	0.997561

作成した表を見ると多くの場合で追い越し効果は 0 秒より大きい結果となるように見える。1 台後方の車両速度, 1 台後方車両との距離差の二つの変数は結果に与える影響が大きいことが表中の最大値と最小値の差よりわかる。作成したモデルより追い越し実施基準を整理した。しかし, 表を見る際には今回入力したデータの幅の中でのみ有効であり, 一様に発生するわけではなく, 無駄な追い越しが半数を超えていたことから, 表中のシチュエーションの中でも発生条件に偏りがあることを留意する必要がある。

7. 結論

7-1 本研究の成果

暫定 2 車線道路における追い越し実施基準の整備を目的として、初めに、追い越しによる効果を定義し計算方法を確立した。その後、高知道暫定 2 車線区間を対象とし、実地調査を実施し暫定 2 車線道路区間の走行環境を把握した。調査結果を元に、シミュレーションの再現性を確認し、再現性が確保されたシミュレーションを用いて、追い越し効果を計算した。追い越し効果が 0 秒以下と 0 秒より大きい 2 つに追い越しをわけ、どのような発生状況の違いによる影響なのかを判別分析によりモデル化した。作成したモデルを元に追い越し実施の基準を整理した。

本研究の成果は追い越し効果を新しく定義し追い越しの単位を定義したこと。また、追い越し効果の計算方法を確立したこと。加えて、追い越し実施時の状況より追い越しによる旅行時間の短縮が発生するか判別可能としたことである。作成した追い越し判断基準は使用する指標がどの暫定 2 車線道路上でも観測可能な指標としており、高知道以外の付加車線区間の追い越しでも使用できるものとなっている。

7-2 課題

使用した調査結果の集計方法には仮定も多く含んでおり、シミュレーションに入力する OD の信頼性が十分とは言えない。また、シミュレーションソフトの校正では、速度の再現性を高めたが、目指すべきは旅行時間での再現性を確保することであり、観測環境を整え調査の際の旅行時間を計算の為にすべての IC の流入を観測するなど OD の信頼性を上げることによって本研究の分析結果すべての精度向上が見込めると考える。

シミュレーションソフト上の車両挙動再現の限界から実状に対し偏った結果を出力している可能性は否定できない。したがって、作成した追い越し実施基準と付加車線区間の観測結果を比較検証することが研究成果の信頼性向上に必要である。

作成した追い越し実施基準は他地点でも使用可能な指標のみで作成したものの、他区間へ転用が可能かは不明である、同様の手法で基準を作成することは可能であるが、本研究の成果が高知道の車両特性に強く依存していた場合、手を加えずに転用することは望ましくない。

指標に用いた密度の係数が正となった結果については発生している現象を考えると理解可能な結果である。しかし、本研究の基準作成は追い越しをした場合の集計であり、しなかった場合の結果は得られない。複数車両の利害関係を考え、車両の追い越しが周辺車両に与える影響を評価に加える等、考慮する車両を増やし逐次的なモデル作成により密度が高くなるほど追い越しが効果的に行えるという結果がより感覚に合った結果になるのではと考える。

8-3 今後の展望

追い越し実施基準の転用，精度向上のためにも他の暫定 2 車線区間でも同様の追い越し効果分析を進め，データを蓄積する必要がある．また，追い越し効果の一般道や 4 車線以上の区間への適応方法の検討や，自動運転車両を考慮した付加車線区間の効率的な設置基準の検討といった研究を進めていく必要がある．今回示した判別分析の結果は自動運転車両の追い越し判断に適応可能であり，今後自動運転車両が交通流に混在していく際の車両制御方法検討の一助となることを期待する．今後分析を進める事，旅行時間の最小化という視点での付加車線設置間隔の検討や既存ストックの効率的運用に向けた施策提言が可能となる事に期待する．

謝辞

本研究及び論文作成にあたっては、多くの方々よりご指導とご支援をいただきました。ここに深く感謝いたします。

この研究を遂行するにあたり、終始熱心なご指導を頂いた指導教員の高知工科大学システム工学群西内裕晶准教授に深く感謝いたします。

また、2年間にわたり研究の幹となる内容への理解、言語化には副指導教員である高知工科大学システム工学群赤塚慎准教授ならびに高知工科大学システム工学群高木方隆教授との意見交換なく達成できなかつたと思います。有り難うございます。

シミュレーションソフトの提供、ならびにデータの解析に関する意見交換にご協力をいただいたばかりでなく、貴重な時間をさいて技術指導をして頂いた株式会社アイ・トランスポート・ラボ甲斐慎一郎様に感謝いたします。

西日本高速道路株式会社四国支社安松谷隆之様には高知道における調査において調査計画書の作成協力から道路事務所への連絡、調査日のトラフィックカウンターデータ提供等研究におけるデータ取得に協力頂き有り難うございました。

令和2年3月10日

秦 啓

参考文献

- 1) 塩見康博, 吉井稔雄, 北村隆一: Platoon-Based Traffic Flow Model for Estimating Breakdown Probability at Single-Lane Expressway Bottlenecks. 19th International Symposium on Transportation and Traffic Theory. 2011
- 2) 吉川良一, 長浜和実, シンジャン, 吉井稔雄, 北村隆一: 高速道路暫定 2 車線区間におけるボトルネック交通容量に関する考察. 土木計画学論文集 22 巻 p839-845. 2005
- 3) 中村英樹, 小林正人, CATBAGAN Jerome L: 追従車密度を考慮した往復 2 車線道路における付加追い越し車線設置水準に関する研究. 土木学会論文集 D3. 67(3), 270-282. 2011
- 4) 成嶋晋一, 葛西 誠, 邢健, 後藤秀典, 辻光弘: ETC2.0 データによる高速道路の暫定 2 車線区間の交通実態分析. 交通工学論文集 2017 年 3 巻 2 号 p. A_125-A_134. 2017
- 5) 根川 拓, 佐野 可寸志, 西内 裕晶: 暫定 2 車線高速道路付加車線内における追い越し挙動のモデル化. 交通工学論文集, 1(4), A_24-A_30, 2015
- 6) 堀口 良太, 赤羽 弘和, 尾高 寛信: 累積交通量とアップリンク情報を用いた高速道路旅行時間の短期予測, 第 2 回 ITS シンポジウム 2003
- 7) 鈴木宏典, 中 辻 隆: フィードバック原理に基づく交通状態推定手法を応用した高速道路上起終点旅行時間の推定, 土木学会論文集 No. 695/IV-54, 137-148, 2002
- 8) 山崎 浩気, 宇野 伸宏, 倉内 文孝, 嶋本 寛, 小笹 浩司, 成田 博: ETC データを用いた都市間高速道路の旅行時間信頼性評価 に関する研究, 土木計画学論文集 vol.25 no.4 2008

付録

・Python 計算コード

シミュレーションソフトより出力された5秒区切り CSV ファイルを元に追い越し効果に関する計算項目を算出するコード。Python3.0以降で動作確認済み。

```
# -*- coding: utf-8 -*-

#pandas のインストール
import pandas as pd
import numpy as np
import os
import glob
import csv
import pandas as pd

print("csv conect")
All_Files = glob.glob('*5sec.csv')

# フォルダ中の全 csv をマージ
list = []
for file in All_Files:
    list.append(pd.read_csv(file))
df = pd.concat(list)

# csv 出力
df.to_csv('comm_datas.csv', encoding='utf_8_sig')

#print(df.shape)
#print(df.head())
print("csv conect fin")

#test01
print("test01")
#結合した CSV の読み込み
#data = pd.read_csv('kochi_vpos1.csv_VID0000-000000.csv', sep=",")
data = pd.read_csv('comm_datas.csv', sep=",")
```



```

#print(data.head())

#空白の列の追加
data['distance from the base[m]'] = ""
data['point[m]'] = ""
#print(data.head())

#始点ごとに絶対位置を算出し付与
for index, row in data.iterrows():
    print("test01:" + str(index / len(data) * 100) + "%")
    if data['Origin'][index] == 'zone_susakihigashi':
        data['distance from the base[m]'][index] = -478.79
        data['point[m]'][index] = data['distance from the
base[m]'][index]+data['ODO[m]'][index]
    elif data['Origin'][index] == 'zone_tosaSIC':
        data['distance from the base[m]'][index] = 9223.75
        data['point[m]'][index] = data['distance from the
base[m]'][index]+data['ODO[m]'][index]
    elif data['Origin'][index] == 'zone_tosa':
        data['distance from the base[m]'][index] = 13923.68
        data['point[m]'][index] = data['distance from the
base[m]'][index]+data['ODO[m]'][index]
    elif data['Origin'][index] == 'zone_ino':
        data['distance from the base[m]'][index] = 23330.70
        data['point[m]'][index] = data['distance from the
base[m]'][index]+data['ODO[m]'][index]
    elif data['Origin'][index] == 'zone_kochi':
        data['distance from the base[m]'][index] = 33847.97
        data['point[m]'][index] = data['distance from the
base[m]'][index]+data['ODO[m]'][index]
    else:
        data['distance from the base[m]'][index] = 100000
        print(error)
        data['point[m]'][index] = data['distance from the
base[m]'][index]+data['ODO[m]'][index]

```

```

#distance from the base[m]が計算された CSV の出力
#data.to_csv('datas_01.csv', index=False)
#print(open('datas_01.csv').read())
#print("fin_01")

#point[m]が計算された CSV の出力
data.to_csv('datas_02.csv', index=False)
print("fin_02")

#時間と距離でソートする
data = data.sort_values(['Time','point[m]'],ascending=[True,True])

#インデックスをリセットする
data.reset_index(drop=True, inplace=True)

#ソート後の CSV 出力
data.to_csv('datas_03.csv', index=False)
print("fin_03")

#print(data.head())

#test02i
print("test02")
#deta_03.csv を開く
data = pd.read_csv('datas_03.csv', sep=",")

#print(data.head())

#不要な行の削除(LINKID に E56 を含む行のみ抽出)
data = data[data.LinkID.str.contains('E56')]
#インデックスをリセットする
data.reset_index(drop=True, inplace=True)

# 後方車 ID の行追加
data['back vehicle ID'] = ""
# 前方車 ID の行追加

```

```

data['front vehicle ID'] = ""

#後方車 ID の検索
for index in range(1,len(data)):
    print("test02_zennpou:" + str(index / len(data) * 100) + "%")
    index01 = index - 1
    if data['Time'][index] == data['Time'][index01]:
        data['back vehicle ID'][index] = data['VehicleID'][index01]
    else:
        data['back vehicle ID'][index] = 'Null'

#後方車 ID の計算済み CSV の出力
data.to_csv('datas_04.csv', mode='w', index=False)

print("fin_04")

#index のリセット
index = 0

#前方車 ID の検索
for index in range(0,len(data)-1):
    print("test02_kouhou:" + str(index / len(data) * 100) + "%")
    index02 = index + 1
    if data['Time'][index] == data['Time'][index02]:
        data['front vehicle ID'][index] = data['VehicleID'][index02]
    else:
        data['front vehicle ID'][index] = 'Null'

#前方車 ID の検索結果 CSV の出力
data.to_csv('datas_05.csv', mode='w', index=False)

print("fin_05")

#print(data.head())

#test03

```

```

print("test03")
#csv の読み込み
data = pd.read_csv('datas_05.csv', sep=",")

#print(data.head())

#ID と時間で並び替え
data = data.sort_values(['VehicleID','Time'],ascending=[True,True])

#index のリセット
data.reset_index(drop=True, inplace=True)

#CSV の保存
data.to_csv('datas_06.csv', index=False)
print("fin_06")

#print(data.head())

#追越の列追加
data['overtaking'] = ""
# 退出の列追加
data['situation'] = ""

#追越の判定
for index in range(1,len(data)):
    print("test03_overtake:" + str(index / len(data) * 100) + "%")
    index01 = index - 1
    if data['VehicleID'][index] == data['VehicleID'][index01]:
        if data['back vehicle ID'][index] == data['front vehicle ID'][index01]:
            data['overtaking'][index] = 1
        else:
            data['overtaking'][index] = 0

#CSV へ出力
data.to_csv('datas_07.csv', mode='w', index=False)

```

```

print("fin_07")

#print(data.head())

#test04
print("test04")
#csv の読み込み
data = pd.read_csv('datas_07.csv', sep=",")

#区間退出の検索
for index in range(0,len(data)-1):
    print("test04_situation:" + str(index / len(data) * 100) + "%")
    index02 = index + 1
    if data['VehicleID'][index] == data['VehicleID'][index02]:
        data['situation'][index] = 0
    else:
        data['situation'][index] = 1

#CSV の保存
data.to_csv('datas_08.csv', mode='w', index=False)

print("fin_08")

#csv の読み込み
df = pd.read_csv('datas_08.csv', sep=",")
#空欄の穴埋め
df01=df.fillna('Null')

#CSV の保存
df01.to_csv('datas_09.csv', mode='w', index=False)

print("fin_09")

```

```

#print(df01.head())

#test05
print("test05")
#csv の読み込み
df = pd.read_csv('datas_08.csv')

#print(df.head())

#一番大きい VehicleID の抽出
df00 = df.iloc[-1,2]
#print(df00)
df01 = df00 + 1
#print(df01)

df02 = pd.DataFrame()
#df01 = 'ID'
#df01.to_csv('id_list.csv', mode='w', index=False)
df02['ID'] = ""

#print(df02)

#idlist の作成
for i in range(0,df01,1):
    tmp_se = pd.Series( [i], index=df02.columns )
    df02 = df02.append( tmp_se, ignore_index=True )

#print(df02.head())
df02.to_csv('id_list.csv', mode='w', header=False, index=False)

df03 = pd.read_csv('id_list.csv')
print("fin_id_list")

#print(df03.head())

#print(df03.head())

```

```

#test06
print("test06")
#IDlist の読み込み
id_list = pd.read_csv('id_list.csv')
#print(id_list)

#csv の読み込み
df = pd.read_csv('datas_09.csv',header=0)
#print(df.head())

for i in range(0,len(id_list),1):
    pass
    ID = id_list.iloc[i,0]
    print("test06:" + str(ID))
    df = pd.read_csv('datas_09.csv',header=0)
    df_temporary = df[df['VehicleID'] == ID]
    df=df_temporary
    df.to_csv(str(ID) +'.csv', mode='w', index=False)

print('fin_idlist.csv')

#test07 追越を抜き出すプログラム
print("test07")
# csv ファイルの読み込み
df = pd.read_csv('datas_08.csv')
#print(df.head())

#追越と最後の行抽出
df_overtaking = df.query('overtaking == 1 or situation == 1')
#print('overtaking')
#print(df_overtaking.head())
df_overtaking.to_csv('overtaking.csv', mode='w', index=False)

print('fin_overtaking.csv')

```

```

#追越車両のみ抽出
df_overtaking_list = df_overtaking.query('overtaking == 1')
df_overtaking_list_01 = df_overtaking_list['VehicleID']

#重複の削除
    #重複の抽出
is_complete_duplicate_keep_first = (df_overtaking_list_01.duplicated(keep='first'))
    #重複行の削除
df_overtaking_list_02 = df_overtaking_list_01[~is_complete_duplicate_keep_first]

#追越車両リストを表示
#print(df_overtaking_list_02)
df_overtaking_list_02.to_csv('overtaking_Vehicle_list.csv', mode='w', index=False)
df01 = pd.read_csv('overtaking_Vehicle_list.csv')

print('fin_overtaking_Vehicle_list')

#追越車両のリストを元に追い越し行の抽出
ID = df01.iloc[0,0]
#print(ID)
df03 = pd.read_csv('overtaking.csv', header=0)
df_temporary = df03[df03['VehicleID'] == ID]
df04=df_temporary
df04.to_csv('overtaking_list.csv', mode='w', index=False)
for i in range(1,len(df01),1):
    pass
    ID = df01.iloc[i,0]
    #print(ID)
    #df03 = pd.read_csv('overtaking.csv', header=0)
    df_temporary = df03[df03['VehicleID'] == ID]
    df04=df_temporary
    df04.to_csv('overtaking_list.csv', mode='a', header=False, index=False)
print('fin_overtaking_list')

#test08
print("test08")

```



```

df = pd.read_csv('overtaking_list.csv')
df.fillna('Null')

df_t = df.assign(Time_effect = 0)
df_t['Time_effect_point'] = 0
df_t['error'] = 0
#print(df_t)
t = len(df_t)-1

for i in range(0,t):
    pass
    df1 = df.iloc[i,24]
    #print("index")
    #print(i)
    print("test08:" + str(df.iloc[i,2]))
    #print(df1)
    if df1 == 1 :
        pass
        print('Overtaking')
        next_back_vehicleID = df.iloc[i,22]
        overtaking_effect_point = df.iloc[i+1,21]
        #print(int(next_back_vehicleID))
        #print(overtaking_effect_point)
        if next_back_vehicleID == 'Null':
            pass
            df_t.iat[i,26] = 'Null'

        else:
            #print( next_back_vehicleID.astype(object))
            #print(str(int(next_back_vehicleID))+'.csv')

            df2 = pd.read_csv( str(int(next_back_vehicleID)) +'.csv', header=0)
            #print(df2)
            df3 = df2[df2['point[m]'] > overtaking_effect_point]
            #print(df3)
            if df3.empty:

```

```

        df4 = df2.loc[df2['point[m]'].idxmax()]
        df_t.iloc[i,28] = 1
    else:
        df4 = df3.loc[df3['point[m]'].idxmin()]
        df_t.iloc[i,28] = 0
        #print(df4)
    df5 = df4['Time']
    df6 = df4['point[m]']
    #print(df5)
    df_t.iloc[i,26] = df5
    df_t.iloc[i,27] = df6
    #print(df_t.iloc[i,26])
    #print(df_t.iloc[i,27])

elif df1 == 0 :
    print('Exit')
    #next_back_vehicleID = df.iloc[i,22]
    #print(int(next_back_vehicleID))
    #overtaking_effect_point = df.iloc[i+1,20]
    #print(overtaking_effect_point)
    df_t.iat[i,26] = 'Null'

else:
    print('etc')

df_t.to_csv('overtaking_effect.csv', mode='w', index=False)

#test09 区間ごとの車群密度、区間速度を求める。

print("test09")

#csv の読み込み
data = pd.read_csv('datas_09.csv')

data['section'] = 0

```

```

print(data.head())

for index, row in data.iterrows():
    print("test09:" + str(index / len(data) * 100) + "%")
    if data['point[m]'][index] < 8177:
        data['section'][index] = 1
    elif data['point[m]'][index] < 13511:
        data['section'][index] = 2
    elif data['point[m]'][index] < 23549:
        data['section'][index] = 3
    elif data['point[m]'][index] < 34375:
        data['section'][index] = 4
    else:
        data['section'][index] = 5

data.to_csv('datas_10.csv', index=False)
print("fin_9")

print("test10")

df = pd.read_csv('time_section.csv')
data = pd.read_csv('datas_10.csv')

#print(data.columns)

#print(data.Time)
#df['TIME'] = ""
df["count"] = ""
df["TIME"]=""

for index, row in df.iterrows():
    TIME = str(df['Day'][index] + ' 0' + df['Time'][index])
    df["TIME"][index] = TIME

```

```

print(TIME + ':' + str(df['section'][index]))
data_00 = data[data.Time == TIME]
#print(data_00)
data_01 = (data_00 == df['section'][index])
#print(data_01)
df["count"][index] = data_01.values.sum()
print(data_01.values.sum())

df.to_csv('mitudo.csv', mode='w', index=False)

print('fin_10')

#test09 区間ごとの車群密度、区間速度を求める。

print("test11")

df = pd.read_csv('overtaking_effect.csv')
data = pd.read_csv('datas_03.csv')
#インデックスをリセットする
data.reset_index(drop=True, inplace=True)

#print(df.head())
#print(data.head())

df["front_1_speed"] = ""
df["front_1_point"] = ""
df["front_2_speed"] = ""
df["front_2_point"] = ""
df["front_3_speed"] = ""
df["front_3_point"] = ""
df["back_1_speed"] = ""
df["back_1_point"] = ""
df["back_2_speed"] = ""
df["back_2_point"] = ""

```

```

df["nokori_kyori"] = ""
df["tannro_kyori"] = ""

for index, row in df.iterrows():
    print(index)
    print(df["Time"][index])
    print(df['VehicleID'][index])
    if df['overtaking'][index] == 1:
        N_time = df['Time'][index]
        N_ID = df['VehicleID'][index]
        N_LinkID = df['LinkID'][index]

        data01 = data[data.Time == N_time]
        #print(data01.head())
        data02 = data01[data01.VehicleID == N_ID]
        #print(data02.head())
        data03 = int(data02.index.values)
        print(data03)
        front_1_vecle_index = data03 + 1
        front_2_vecle_index = data03 + 2
        front_3_vecle_index = data03 + 3
        #print(front_1_vecle_index)
        #print(front_2_vecle_index)
        #print(front_3_vecle_index)

        if df["Time"][index] == data["Time"][front_1_vecle_index]:
            df["front_1_speed"][index] = data['Speed[km/h]'][front_1_vecle_index]
            df["front_1_point"][index] = data['point[m]'][front_1_vecle_index]
        else:
            df["front_1_speed"][index] = "Null"
            df["front_1_point"][index] = "Null"

        if df["Time"][index] == data["Time"][front_2_vecle_index]:
            df["front_2_speed"][index] = data['Speed[km/h]'][front_2_vecle_index]
            df["front_2_point"][index] = data['point[m]'][front_2_vecle_index]
        else:

```

```

df["front_2_speed"][index] = "Null"
df["front_2_point"][index] = "Null"

if df["Time"][index] == data["Time"][front_3_vecle_index]:
    df["front_3_speed"][index] = data['Speed[km/h]'][front_3_vecle_index]
    df["front_3_point"][index] = data['point[m]'][front_3_vecle_index]

else:
    df["front_3_speed"][index] = "Null"
    df["front_3_point"][index] = "Null"

back_1_vecle_index = data03 - 1
back_2_vecle_index = data03 - 2
if df["Time"][index] == data["Time"][back_1_vecle_index]:
    df["back_1_speed"][index] = data['Speed[km/h]'][back_1_vecle_index]
    df["back_1_point"][index] = data['point[m]'][back_1_vecle_index]
else:
    df["back_1_speed"][index] = "Null"
    df["back_1_point"][index] = "Null"

if df["Time"][index] == data["Time"][back_2_vecle_index]:
    df["back_2_speed"][index] = data['Speed[km/h]'][back_2_vecle_index]
    df["back_2_point"][index] = data['point[m]'][back_2_vecle_index]
else:
    df["back_2_speed"][index] = "Null"
    df["back_2_point"][index] = "Null"

if df["LinkID"][index] == "E56_020_1":
    df["nokori_kyori"][index] = 436.61 + df["DistFromStopLine[m]"][index]
    df["tannro_kyori"][index] = 4387.15

elif df["LinkID"][index] == "E56_019_1":
    df["nokori_kyori"][index] = df["DistFromStopLine[m]"][index]
    df["tannro_kyori"][index] = 4387.15

elif df["LinkID"][index] == "E56_015_1":

```

```
df["nokori_kyori"][index] = 702.95 + df["DistFromStopLine[m]"][index]  
df["tannro_kyori"][index] = 8421.54
```

```
elif df["LinkID"][index] == "E56_014_1":  
    df["nokori_kyori"][index] = 336.4 + df["DistFromStopLine[m]"][index]  
    df["tannro_kyori"][index] = 8421.54
```

```
elif df["LinkID"][index] == "E56_013_1":  
    df["nokori_kyori"][index] = 248.42 + df["DistFromStopLine[m]"][index]  
    df["tannro_kyori"][index] = 8421.54
```

```
elif df["LinkID"][index] == "E56_012_1":  
    df["nokori_kyori"][index] = 181.43 + df["DistFromStopLine[m]"][index]  
    df["tannro_kyori"][index] = 8421.54
```

```
elif df["LinkID"][index] == "E56_011_1":  
    df["nokori_kyori"][index] = 70.21 + df["DistFromStopLine[m]"][index]  
    df["tannro_kyori"][index] = 8421.54
```

```
elif df["LinkID"][index] == "E56_010_1":  
    df["nokori_kyori"][index] = df["DistFromStopLine[m]"][index]  
    df["tannro_kyori"][index] = 8421.54
```

```
elif df["LinkID"][index] == "E56_007_1":  
    df["nokori_kyori"][index] = 782.39 + df["DistFromStopLine[m]"][index]  
    df["tannro_kyori"][index] = 9337.38
```

```
elif df["LinkID"][index] == "E56_006_1":  
    df["nokori_kyori"][index] = 414.03 + df["DistFromStopLine[m]"][index]  
    df["tannro_kyori"][index] = 9337.38
```

```
elif df["LinkID"][index] == "E56_005_1":  
    df["nokori_kyori"][index] = df["DistFromStopLine[m]"][index]  
    df["tannro_kyori"][index] = 9337.38
```

```
else:
```

```
df["nokori_kyori"][index] = "error"
df["tannro_kyori"][index] = "error"

else:
    df["front_1_speed"][index] = "Null"
    df["front_1_point"][index] = "Null"
    df["front_2_speed"][index] = "Null"
    df["front_2_point"][index] = "Null"
    df["front_3_speed"][index] = "Null"
    df["front_3_point"][index] = "Null"
    df["back_1_speed"][index] = "Null"
    df["back_1_point"][index] = "Null"
    df["back_2_speed"][index] = "Null"
    df["back_2_point"][index] = "Null"

df.to_csv('hanbetu.csv', mode='w', index=False)

print('fin_11')

print("test12")

hanbetu = pd.read_csv('hanbetu.csv')
df = pd.read_csv('mitudo.csv')
data = pd.read_csv('datas_03.csv')

hanbetu['section'] = 0

#print(hanbetu.head())

for index, row in hanbetu.iterrows():
    print("test12:" + str(index / len(hanbetu) * 100) + "%")
    if hanbetu['point[m]'][index] < 8177:
        hanbetu['section'][index] = 1
```



```

elif hanbetu['point[m]'][index] < 13511:
    hanbetu['section'][index] = 2
elif hanbetu['point[m]'][index] < 23549:
    hanbetu['section'][index] = 3
elif hanbetu['point[m]'][index] < 34375:
    hanbetu['section'][index] = 4
else:
    hanbetu['section'][index] = 5

print(hanbetu.head())

hanbetu.to_csv('hanbetu-2.csv', mode='w', index=False)
print('fin_12')

hanbetu = pd.read_csv('hanbetu-2.csv')
df = pd.read_csv('mitudo.csv')
data = pd.read_csv('datas_03.csv')

hanbetu['mitudo'] = 0
#print(hanbetu.head())
for index, row in hanbetu.iterrows():
    #print("test13:" + str(index / len(hanbetu) * 100) + "%")
    #print(index)
    section01 = hanbetu['section'][index]
    Time01 = hanbetu['Time'][index]
    print(str(Time01) + " : " + str(section01))
    if Time01 == "2015/10/08 07:00:00":
        hanbetu['mitudo'][index] = str("error")
        print("error")
    elif Time01 == "2015/10/08 07:00:05":
        hanbetu['mitudo'][index] = str("error")
        print("error")
    elif Time01 == "2015/10/08 07:00:15":
        hanbetu['mitudo'][index] = str("error")
        print("error")
    elif Time01 == "2015/10/08 07:00:20":

```

```
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:00:25":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:00:30":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:00:35":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:00:40":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:00:45":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:00:50":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:00:55":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:01:00":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:01:05":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:01:10":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:01:15":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:01:20":
```

```
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:01:25":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:01:30":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:01:35":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:01:40":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:01:45":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:01:50":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:01:55":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:02:00":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:02:05":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:02:10":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:02:15":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:02:20":
```

```
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:02:25":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:02:30":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:02:35":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:02:40":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:02:45":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:02:50":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:02:55":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:03:00":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:03:05":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:03:10":
        hanbetu['mitudo'][index] = str("error")
        print("error")
elif Time01 == "2015/10/08 07:03:15":
        hanbetu['mitudo'][index] = str("error")
        print("error")
else:
```

```
dfa = df[df["TIME"] == Time01]
#print(df2)
dfb = dfa[dfa["section"] == section01]
#print(df3)
#print(df3["count"])
#print(df3.iat[0, 2])
hanbetu['mitudo'][index] = str(dfb.iat[0, 3])
print(str(index) + ":" + str(hanbetu['mitudo'][index]))

hanbetu.to_csv('hanbetu-3.csv', mode='w', index=False)

print('Fin')
```