

令和4年度

修士論文

バスプローブデータを用いた
道路ネットワーク交通流状態の分析

Analysis of road network traffic flow conditions

using bus probe data

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 1255066
吉田 朋恵

指導教員 西内 裕晶
副指導教員 赤塚 慎
論文副審査 高木 方隆

2023年3月10日

要旨

車両感知器に代わり近年では ETC2.0 プローブ車両の増加とともに、地方部においても道路ネットワークの交通流状態を高密度に観測できるようになった。しかしながら、ETC2.0 プローブ車両から取得できるデータはサンプルデータであるため、時間の偏りや空間の偏りが生じる。

そこで本研究では、定時かつ定路線で走行するバスに着目し、バス停において記録されたバスプローブデータを道路ネットワークの交通流状態の観測に用いる。そして、バスプローブデータから道路ネットワークの交通流状態が把握できることを明らかにする。車両感知器・ETC2.0 プローブ車両・バスから取得したデータを用いて交通流状態別の速度を分析した。分析の結果、主要渋滞区間とバス路線の重なる範囲が多いという特徴を持つ道路ネットワークではバスプローブデータを用いることで渋滞流の速度を把握できることを示唆した。

Abstract

With the recent increase in the number of ETC2.0 probe vehicles replacing vehicle detectors, it has become possible to observe traffic flow conditions in road networks at high densities even in rural areas. However, the data that can be obtained from ETC2.0 probe vehicles is sample data, and thus subject to time bias and spatial bias.

In this study, we focus on buses that run on a regular schedule and on a fixed route, and use bus probe data recorded at bus stops to observe traffic flow conditions in the road network. It is then revealed that bus probe data can be used to understand the state of traffic flow in the road network. Data obtained from vehicle detectors, ETC2.0 probe vehicles, and buses were used to analyze speeds by traffic flow condition. The analysis results suggest that bus probe data can be used to determine the speed of congested flow in a road network characterized by a large overlap area between major congested sections and bus routes.

目次

1	序章	1
1-1	研究背景	1
1-2	研究目的	2
1-3	研究構成	2
2	既往研究の整理と本研究の位置づけ	3
2-1	既往研究の整理	3
2-1-1	道路ネットワークの交通流状態の把握手法に関する研究	3
2-1-2	車両感知器データを用いた MFD に関する研究	3
2-1-3	ETC2.0 プローブデータを用いた MFD に関する研究	4
2-1-4	ETC2.0 プローブデータの課題に関する研究	4
2-2	本研究の位置づけ	5
3	本研究における道路ネットワークの交通流状態の観測	6
3-1	高知県における交通流観測	6
3-1-1	車両感知器の設置状況	6
3-1-2	ETC2.0 の普及状況	7
3-2	交通流状態の表現方法	8
3-3	分析エリアの設定	9
3-4	分析使用データの概要	10
3-4-1	車両感知器データ	10
3-4-2	ETC2.0 プローブデータ	13
3-4-3	バスプローブデータ	16
4	道路ネットワークの交通流状態の算定方法	18
4-1	車両感知器データによる MFD の描画	18
4-1-1	MFD の算出式	18
4-1-2	リンクの長さの算出	18
4-1-3	流率と密度の算出	21
4-1-4	MFD の描画	25
4-2	ETC2.0 プローブデータによる MFD の描画	28
4-2-1	MFD の算出式	28
4-2-2	分析エリア内を走行する車両の判別	28
4-2-3	流率と密度の算出	29
4-2-4	MFD の描画	32
4-3	バスプローブデータによる MFD の描画	35
4-3-1	MFD の算出式	35
4-3-2	分析エリア内を走行する車両の判別	35

4-3-3	流率と密度の算出.....	36
4-3-4	MFD の描画.....	39
5	道路ネットワークの交通流状態の比較.....	42
5-1	分析使用データ別の MFD.....	42
5-2	交通流状態の比較指標.....	44
6	道路ネットワーク速度の基礎分析.....	45
6-1	分析エリア別の速度の時系列変化.....	45
6-2	分析使用データ別の速度分布.....	48
6-2-1	車両感知器の速度分布.....	48
6-2-2	ETC2.0 プローブの速度分布.....	51
6-2-3	バスプローブの速度分布.....	54
7	折れ線回帰による交通流状態別の速度分析.....	57
7-1	折れ線回帰モデルを用いた MFD の推定.....	57
7-2	折れ線回帰モデル式.....	58
7-2-1	車両感知器の折れ線回帰モデル式.....	58
7-2-2	ETC2.0 プローブの折れ線回帰モデル式.....	65
7-2-3	バスプローブの折れ線回帰モデル式.....	72
7-3	交通流状態別の速度比較.....	79
7-3-1	自由流と混雑流の速度.....	80
7-3-2	自由流と渋滞流の速度.....	81
7-4	交通流状態別速度と分析使用データ・分析エリアの関係性に関する考察.....	82
7-4-1	交通流状態別速度と分析使用データ.....	82
7-4-2	交通流状態別速度と分析エリア.....	82
8	結論.....	84
8-1	研究成果.....	84
8-2	今後の課題.....	84

図目次

図 1	高知県の ETC2.0 セットアップ件数の推移	7
図 2	交通量と密度の関係	8
図 3	分析エリア	9
図 4	感知器配置図	10
図 5	リンクの長さの定義	19
図 6	車両感知器設置地点	19
図 7	分析エリア 0 の車両感知器 MFD	26
図 8	分析エリア 1 の車両感知器 MFD	26
図 9	分析エリア 2 の車両感知器 MFD	27
図 10	分析エリア 3 の車両感知器 MFD	27
図 11	分析エリア 0 の ETC2.0 プローブ MFD	32
図 12	分析エリア 1 の ETC2.0 プローブ MFD	33
図 13	分析エリア 2 の ETC2.0 プローブ MFD	33
図 14	分析エリア 3 の ETC2.0 プローブ MFD	34
図 15	分析エリア 0 のバスプローブ MFD	39
図 16	分析エリア 1 のバスプローブ MFD	40
図 17	分析エリア 2 のバスプローブ MFD	40
図 18	分析エリア 3 のバスプローブ MFD	41
図 19	車両感知器の MFD	42
図 20	ETC2.0 プローブの MFD	43
図 21	バスプローブの MFD	43
図 22	分析エリア 0 の速度の時系列変化	46
図 23	分析エリア 1 の速度の時系列変化	46
図 24	分析エリア 2 の速度の時系列変化	47
図 25	分析エリア 3 の速度の時系列変化	47
図 26	分析エリア 0 の車両感知器速度分布	48
図 27	分析エリア 1 の車両感知器速度分布	49
図 28	分析エリア 2 の車両感知器速度分布	49
図 29	分析エリア 3 の車両感知器速度分布	50
図 30	分析エリア 0 の ETC2.0 プローブ速度分布	51
図 31	分析エリア 1 の ETC2.0 プローブ速度分布	52
図 32	分析エリア 2 の ETC2.0 プローブ速度分布	52
図 33	分析エリア 3 の ETC2.0 プローブ速度分布	53
図 34	分析エリア 0 のバスプローブ速度分布	54
図 35	分析エリア 1 のバスプローブ速度分布	55

図 36	分析エリア 2 のバスプローブ速度分布.....	55
図 37	分析エリア 3 のバスプローブ速度分布.....	56
図 38	分析エリア 0 の車両感知器折れ線回帰モデル(自由流の R^2).....	58
図 39	分析エリア 0 の車両感知器折れ線回帰モデル(混雑流の R^2).....	59
図 40	分析エリア 0 の車両感知器折れ線回帰モデル(渋滞流の R^2).....	59
図 41	分析エリア 1 の車両感知器折れ線回帰モデル(自由流の R^2).....	60
図 42	分析エリア 1 の車両感知器折れ線回帰モデル(混雑流の R^2).....	60
図 43	分析エリア 1 の車両感知器折れ線回帰モデル(渋滞流の R^2).....	61
図 44	分析エリア 2 の車両感知器折れ線回帰モデル(自由流の R^2).....	61
図 45	分析エリア 2 の車両感知器折れ線回帰モデル(混雑流の R^2).....	62
図 46	分析エリア 2 の車両感知器折れ線回帰モデル(渋滞流の R^2).....	62
図 47	分析エリア 3 の車両感知器折れ線回帰モデル(自由流の R^2).....	63
図 48	分析エリア 3 の車両感知器折れ線回帰モデル(混雑流の R^2).....	63
図 49	分析エリア 3 の車両感知器折れ線回帰モデル(渋滞流の R^2).....	64
図 50	分析エリア 0 の ETC2.0 プローブ折れ線回帰モデル(自由流の R^2).....	65
図 51	分析エリア 0 の ETC2.0 プローブ折れ線回帰モデル(混雑流の R^2).....	66
図 52	分析エリア 0 の ETC2.0 プローブ折れ線回帰モデル(渋滞流の R^2).....	66
図 53	分析エリア 1 の ETC2.0 プローブ折れ線回帰モデル(自由流の R^2).....	67
図 54	分析エリア 1 の ETC2.0 プローブ折れ線回帰モデル(混雑流の R^2).....	67
図 55	分析エリア 1 の ETC2.0 プローブ折れ線回帰モデル(渋滞流の R^2).....	68
図 56	分析エリア 2 の ETC2.0 プローブ折れ線回帰モデル(自由流の R^2).....	68
図 57	分析エリア 2 の ETC2.0 プローブ折れ線回帰モデル(混雑流の R^2).....	69
図 58	分析エリア 2 の ETC2.0 プローブ折れ線回帰モデル(渋滞流の R^2).....	69
図 59	分析エリア 3 の ETC2.0 プローブ折れ線回帰モデル(自由流の R^2).....	70
図 60	分析エリア 3 の ETC2.0 プローブ折れ線回帰モデル(混雑流の R^2).....	70
図 61	分析エリア 3 の ETC2.0 プローブ折れ線回帰モデル(渋滞流の R^2).....	71
図 62	分析エリア 0 のバスプローブ折れ線回帰モデル(自由流の R^2).....	72
図 63	分析エリア 0 のバスプローブ折れ線回帰モデル(混雑流の R^2).....	73
図 64	分析エリア 0 のバスプローブ折れ線回帰モデル(渋滞流の R^2).....	73
図 65	分析エリア 1 のバスプローブ折れ線回帰モデル(自由流の R^2).....	74
図 66	分析エリア 1 のバスプローブ折れ線回帰モデル(混雑流の R^2).....	74
図 67	分析エリア 1 のバスプローブ折れ線回帰モデル(渋滞流の R^2).....	75
図 68	分析エリア 2 のバスプローブ折れ線回帰モデル(自由流の R^2).....	75
図 69	分析エリア 2 のバスプローブ折れ線回帰モデル(混雑流の R^2).....	76
図 70	分析エリア 2 のバスプローブ折れ線回帰モデル(渋滞流の R^2).....	76
図 71	分析エリア 3 のバスプローブ折れ線回帰モデル(自由流の R^2).....	77

図 72	分析エリア 3 のバスプローブ折れ線回帰モデル(混雑流の R^2)	77
図 73	分析エリア 3 のバスプローブ折れ線回帰モデル(渋滞流の R^2)	78

表目次

表 1	市町村別の車両感知器設置地点数.....	6
表 2	車両感知器設置場所一覧表の例.....	11
表 3	車両感知器データの例.....	12
表 4	運行情報・車種情報の詳細項目.....	13
表 5	トリップに関する情報・更新情報の詳細項目.....	14
表 6	ETC2.0 プローブデータの例.....	15
表 7	バスプローブデータの記載項目.....	16
表 8	バスプローブデータの例.....	17
表 9	リンク情報の例.....	20
表 10	車両感知器ごとの観測データの例.....	21
表 11	車両感知器ごとの情報の例.....	22
表 12	車両感知器ごとの走行台キロと車両存在台数の例.....	23
表 13	分析エリアの流率と密度の例.....	24
表 14	分析エリア内の ETC2.0 プローブデータの例.....	29
表 15	ETC2.0 プローブ車両ごとの情報の例.....	30
表 16	分析エリア内の ETC2.0 プローブ流率と密度の例.....	31
表 17	分析エリア内のバスプローブデータの例.....	36
表 18	バス車両ごとの情報の例.....	37
表 19	分析エリア内のバスプローブ流率と密度の例.....	38
表 20	車両感知器の交通流状態別パラメータ.....	79
表 21	ETC2.0 プローブの交通流状態別パラメータ.....	79
表 22	バスプローブの交通流状態別パラメータ.....	79
表 23	車両感知器の自由流と混雑流の速度比較.....	80
表 24	ETC2.0 プローブの自由流と混雑流の速度比較.....	80
表 25	バスプローブの自由流と混雑流の速度比較.....	80
表 26	車両感知器の自由流と渋滞流の速度比較.....	81
表 27	ETC2.0 プローブの自由流と渋滞流の速度比較.....	81
表 28	バスプローブの自由流と渋滞流の速度比較.....	81

1 序章

1-1 研究背景

ひとたび交通渋滞が発生すると、時間経過とともにその渋滞はネットワーク状に広がる。そのため、交差点や単路部におけるピンポイントな交通渋滞対策ではなく、道路ネットワーク全体における対策が必要となる。ネットワークとして面的に交通流を観測し、道路ネットワーク全体の交通容量を把握することで道路ネットワーク全体の交通制御が可能となる。従って、道路ネットワークの交通流状態を把握することが交通渋滞の対策に繋がる。

道路ネットワークの交通流状態の把握には車両感知器が使用されてきた。車両感知器から取得できるデータは時間的、空間的な分解能が均一であるという特徴を持つ。高速道路においては車両感知器が密に設置されているため、連続した時間で空間的に高密度な交通流の観測が可能である。しかしながら、一般道路においては車両感知器の設置は主要な交差点が基本であり、高密度な交通流の観測が困難であった。

平成 22 年の交通量調査では、高速道路は車両感知器を含む機械観測が 93%を占めるのに対し、直轄国道は車両感知器を含む機械観測が 37%であった。また、地方では車両感知器を含む機械観測は僅か 3.4%であり、人手観測が最も多く 64%であった¹⁾。

そこで平成 27 年の交通量調査では、車両感知器を含む機械観測が積極的に導入された。高速道路は車両感知器を含む機械観測が 97%、直轄国道は車両感知器を含む機械観測が 52%を占めた。しかしながら、地方では車両感知器を含む機械観測は僅か 3.4%と変わらず、人手観測が最も多く 61%であった¹⁾。このことから特に地方部の一般道路の車両感知器の設置状況はなかなか進まず、依然として限定的であると言える。

近年では ETC2.0 プローブ車両の増加により、一般道路においても高密度な交通流の観測が可能となった。平成 27 年 4 月末時点の ETC2.0 車載器普及台数は約 1.9 万台であり、その後 ETC2.0 車載器普及台数は年々増加し、令和 4 年 11 月末時点の ETC2.0 車載器普及台数は約 862 万台に達した²⁾。しかしながら、ETC2.0 プローブ車両から取得できるプローブデータはあくまでもサンプルデータであるため、時間の偏りや空間の偏りが問題視されている。

プローブデータにはタクシープローブデータやバスプローブデータ、商用車プローブデータなどがある。その中でもバスプローブデータは定時かつ定路線で走行するバスから取得できるデータであるため、安定した観測が可能である。時刻表の時間に合わせバス停を出発し、決まった経路でバス停を通過するバスプローブデータは ETC2.0 プローブデータの課題である時間の偏りや空間の偏りを解決できる可能性がある。

1-2 研究目的

本研究では、定時かつ定路線で走行するバスに着目し、バス停において記録されたバスプローブデータを道路ネットワークの交通流状態の観測に用いる。そして、バスプローブデータから道路ネットワークの交通流状態が把握できることを明らかにする。車両感知器データや ETC2.0 プローブデータと同様に道路ネットワークの交通流を算定し、道路ネットワークの交通流状態が表現できているかデータ間で比較する。

1-3 研究構成

本論文は全 8 章の構成である。

第 1 章「序章」では背景と目的を明らかにし、論文の構成を述べる。

第 2 章「既往研究の整理と本研究の位置づけ」では道路ネットワークの交通流状態の把握手法として提案された **Macroscopic Fundamental Diagram** に関する研究を整理し、本研究の位置づけを明確にする。

第 3 章「本研究における道路ネットワークの交通流状態の観測」では高知県における交通流観測の現状を把握し、分析エリアと分析使用データについて説明する。

第 4 章「道路ネットワークの交通流状態の算定方法」では車両感知器・ETC2.0 プローブ車両・バスから取得したデータを用いた道路ネットワークの交通流状態の算定手順について説明する。

第 5 章「道路ネットワークの交通流状態の比較」では第 4 章で算定した道路ネットワークの交通流状態をデータ間で比較した結果について述べる。

第 6 章「道路ネットワーク速度の基礎分析」では分析エリア別の速度の時系列変化と分析使用データ別の速度分布について分析した結果を述べる。

第 7 章「折れ線回帰による交通流状態別の速度分析」では折れ線回帰モデルを用いた推測方法について説明し、速度分析の結果についてまとめる。

第 8 章「結論」では本研究の成果と今後の課題について述べる。

2 既往研究の整理と本研究の位置づけ

2-1 既往研究の整理

本節では本研究に関連する既往研究について取りまとめる。道路ネットワークの交通流状態の把握手法として提案された Macroscopic Fundamental Diagram(以下、MFD)について確認したのち、MFD を用いて道路ネットワークの交通流の特性を分析した研究を観測データごとに確認する。また、現状の ETC2.0 プローブデータの課題についても確認する。

2-1-1 道路ネットワークの交通流状態の把握手法に関する研究

Daganzo³⁾は単一の近隣地域と相互に接続された近隣地域のグリッドロック現象の物理モデルについて分析した。その結果、相互に接続された近隣地域は近隣に適切な感知器があれば、リアルタイムで交通流状態を把握できることが明らかになった。そのため、モデルが都市のモビリティを向上させ、交通渋滞を緩和するための適応制御アプローチに利用できることを示した。リンクレベルのアプローチに代わる方法として道路ネットワークエリア内の交通流動量と交通密度の関係を示す MFD を提案した。

2-1-2 車両感知器データを用いた MFD に関する研究

Geroliminis ら⁴⁾は横浜市において、車両感知器を用いて速度と密度を集計した散布図が滑らかに減少する曲線に沿うことを確認した。さらに、ネットワーク全体に MFD が存在することを証明するため、ネットワーク全体をカバーするタクシー車両と車両感知器を用いて、異なる時間帯の空間平均速度と密度をエリア全体について推定した。その結果、個々のリンクの偏差よりも小さく、滑らかに減少する曲線に近い位置にあり、広大な都市部には空間平均流と密度、速度を結ぶ巨視的な基本図である MFD が存在することを明らかにした。

土生ら⁵⁾は車両感知器データから観測される MFD はエリア平均占有時間に対するエリア平均交通量にばらつきが見られ、複数の交通状態が確認されるため、時間と交通の空間分布がもたらす MFD への影響を分析した。その結果、時間帯によりエリア平均占有時間とエリア平均交通量の関係は異なり、時間帯による差異はその時間帯における渋滞頻出リンクの差異で表現され、入手困難な時間帯別 OD のパターンの傾向が渋滞の空間分布という形で見出すことができることが明らかになった。

王ら⁶⁾は那覇市 CBD の一般道路ネットワークを対象として、MFD の渋滞領域発生メカニズムを実証的に分析した。1年間の車両感知器の観測データを用いて、どの程度の頻度で渋滞領域が発生するかを調べた。その結果、平日の約 20.3%の日で MFD の渋滞領域が発生することが明らかになった。また、渋滞領域発生メカニズムを考察するため、MFD の渋滞領域が現れる日と現れない日の渋滞パターンを比較した。その結果、渋滞領域が現れる日の渋滞パターンは時間的な渋滞パターンの進展の類似度が高いという特徴とピーク時間帯の渋滞リンク数が多いだけでなく、待ち行列の延伸が発生している箇所も多いという特徴を持つことを示した。

王ら⁷⁾は仙台市、京都市の一般道路ネットワークを対象として MFD の特性を実証的に明らかにすることを目的とし、1 年間の車両感知器データから描画した MFD を分析した。その結果、仙台市の悪天候でない平日午前では MFD 上にヒステリシス・ループが常に発現することが明らかになった。また、京都市の休日の MFD では 1 つの大きなループが発現する場合があります、ループの形成過程において、走行台キロがほぼ一定レベルを維持したまま車両存在台数のみが増加し続けることが明らかになった。

酒井ら⁸⁾は渋滞パターンと MFD の関係を明らかにすることを目的とし、首都圏高速道路網において車両感知器の長期間観測データを用いて分析した。その結果、渋滞空間分布には年間を通して普遍的な 8 タイプが存在し、その日内状態推移パターンは年間を通して 4 パターンであることを示した。また、日内状態推移パターンによって MFD の形状が互いに異なることを確認し、MFD 曲線は各点において出現している渋滞空間分布タイプによって、いくつかの部分曲線に分割可能であることを示した。これより、渋滞空間分布の推移過程と MFD の形成過程が対応付けられることが明らかとなった。

2-1-3 ETC2.0 プローブデータを用いた MFD に関する研究

尾高ら⁹⁾は交通事故により CBD 街路網の交通流状態に影響を与えることを明らかにすることを目的に、松山市を対象に ETC2.0 プローブデータから集計 QK を観測し、MFD を推定した。また、交通事故の発生からの経過時間別の MFD の推定結果から交通事故が MFD の形状に影響を及ぼすことを検証した。その結果、交通事故の影響により MFD の形状に有意な差が生じていることが明らかになった。

奥原ら¹⁰⁾は 46 都道府県の各県庁所在地における道路ネットワークを対象に、ETC2.0 プローブデータを用いて事故発生の有無別、時間帯別に折れ線回帰モデルを適用し MFD を推定した。形状変化を分析した結果、事故発生の有無、時間帯の違いによって MFD の形状が有意に変化することが明らかになった。

2-1-4 ETC2.0 プローブデータの課題に関する研究

後藤ら¹¹⁾は交通の量的代表性を評価する指標として、ETC2.0 プローブデータの取得件数と交通量の相関関係を定義し、指標に対して車載器普及率や路側機の分布状況が及ぼす影響について 50 都市を対象に分析した。その結果、車載器普及率や路側機から一定範囲内にある道路延長割合が低い場合に、量的代表性が低くなることが示唆された。

2-2 本研究の位置づけ

我が国でも車両感知器から取得できるデータを用いて、道路ネットワークの交通流状態の把握手法として提案された MFD による分析がされている。さらに近年では、ETC2.0 プローブ車両の増加により、ETC2.0 プローブデータを用いた MFD による分析も進められている。しかしながら、ETC2.0 プローブデータを交通量の代表値として使用する際、データの偏りという課題も存在する。

そこで本研究では、ETC2.0 プローブデータが持つデータの偏りを解決するため、定時かつ定路線で走行するバスに着目した。バスプローブデータを用いて MFD を描画し、道路ネットワークの交通流状態を把握することが本研究の新規性である。

3 本研究における道路ネットワークの交通流状態の観測

3-1 高知県における交通流観測

バスプローブデータを用いて道路ネットワークの交通流状態を算定する MFD を描画し分析するにあたり，車両感知器データと ETC2.0 プローブデータを分析の比較対象とする．そこで本節では，分析エリアを含む高知県における交通流観測方法の現状を把握するため，車両感知器の設置状況と ETC2.0 の普及状況について述べる．

3-1-1 車両感知器の設置状況

高知県警察本部にある高知県交通管制センターが管轄する車両感知器の総台数は 1556 台である．また，車両感知器の総設置地点数は 1009 地点であり，そのうち高知市が 746 地点で全体の約 75% を占める(表 1)．

車両感知器は車両通過の多い主要な交差点に設置されるため，設置地点数が少ない道路ネットワークの交通流状態を観測する際，実現象との相違が懸念される．

表 1 市町村別の車両感知器設置地点数

市町村	設置地点数 [地点]
高知市	746
南国市	56
土佐市	31
須崎市	52
四万十市	53
香南市	23
香美市	7
芸西村	5
いの町	36
	1009

3-1-2 ETC2.0 の普及状況

高知県における 2017 年度から 2021 年度までの 5 年間の ETC2.0 セットアップ件数(再セットアップ件数を含む)の推移を図 1 に示す。高知県の ETC2.0 セットアップ件数は年々増加しているが、全国平均と比較するとその増加は緩やかである。また、ETC2.0 セットアップ件数の累計は 2021 年度末の時点で、沖縄県、鳥取県に次いで少ない 29,807 件であった¹⁾。

そのため、高知県における ETC2.0 プローブ車両の観測台数は少なく、ETC2.0 プローブデータが交通の代表値としての精度は低い可能性がある。

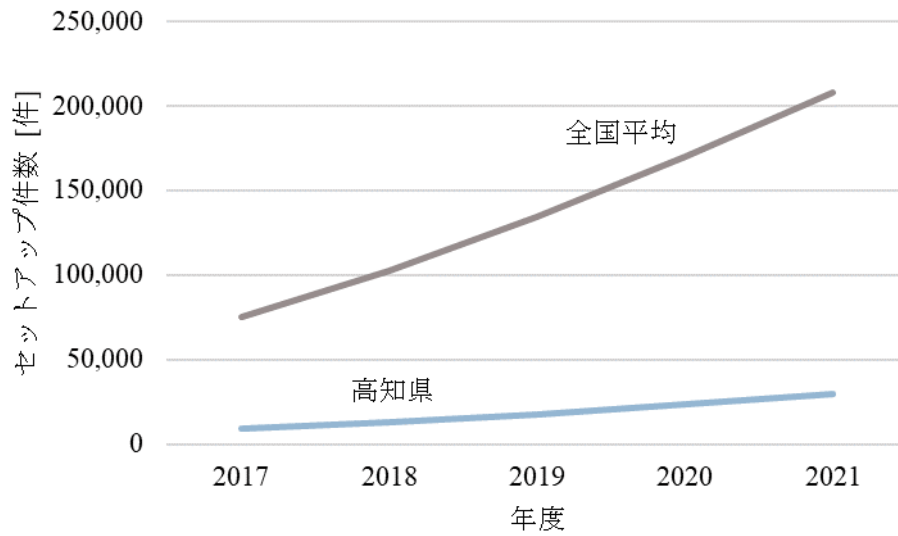


図 1 高知県の ETC2.0 セットアップ件数の推移

3-2 交通流状態の表現方法

交通工学において交通量(Q) – 密度(K)の関係が設定されており、その関係性は図 2 に示すグラフに近似する¹²⁾。グラフの左側に自由流領域、右側に渋滞領域を取り、グラフの頂点の高さが観測される交通流の交通容量を示す。また、傾きは速度を示している。実際の交通流では渋滞領域の始まりあたりまでしか観測されず、交通容量付近の交通流状態がいわゆる交通渋滞にあたる。

本研究では、交通量(Q) – 密度(K)の関係性を示すグラフを道路ネットワーク全体に採用する。交通の流れの多さを把握することが重要であるため、単位時間当たりの交通量である流率をグラフの縦軸に用いる。そして、エリアとして交通流状態を把握するため、MFD を描画し分析する。

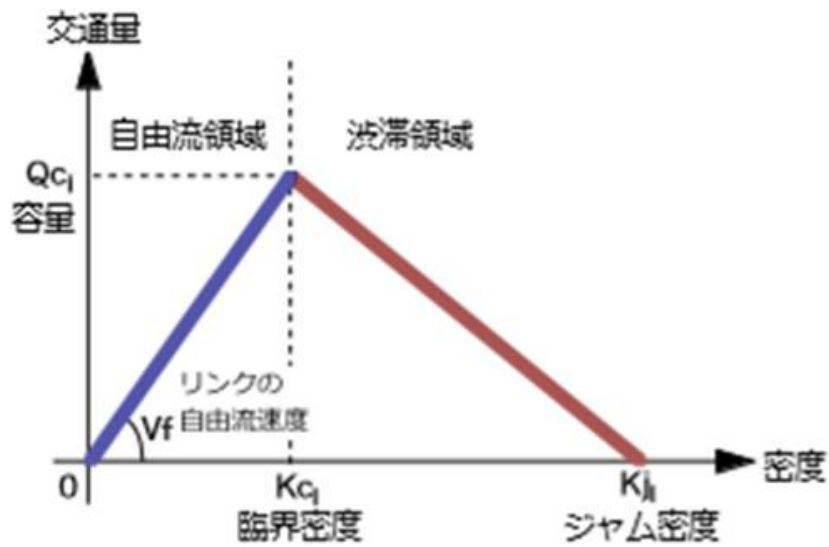


図 2 交通量と密度の関係

3-3 分析エリアの設定

本研究の分析エリアを図3に示す¹³⁾。緑色の点が車両感知器の設置地点であり、白枠で囲まれた水色の多角形が分析エリアである。高知市中心部の道路ネットワークを対象とし、主要な流入出が観測できる車両感知器の設置地点をエリアの境界とした。

さらに、交通渋滞の発生や解消の時間帯が同じであると考えられる範囲を1つのエリアとし、エリア0からエリア3までの4つのエリアを南北方向に分けた。エリアの全長はエリア0が約5km、エリア1が約7km、エリア2が約5.5km、エリア3が約6.5kmである。

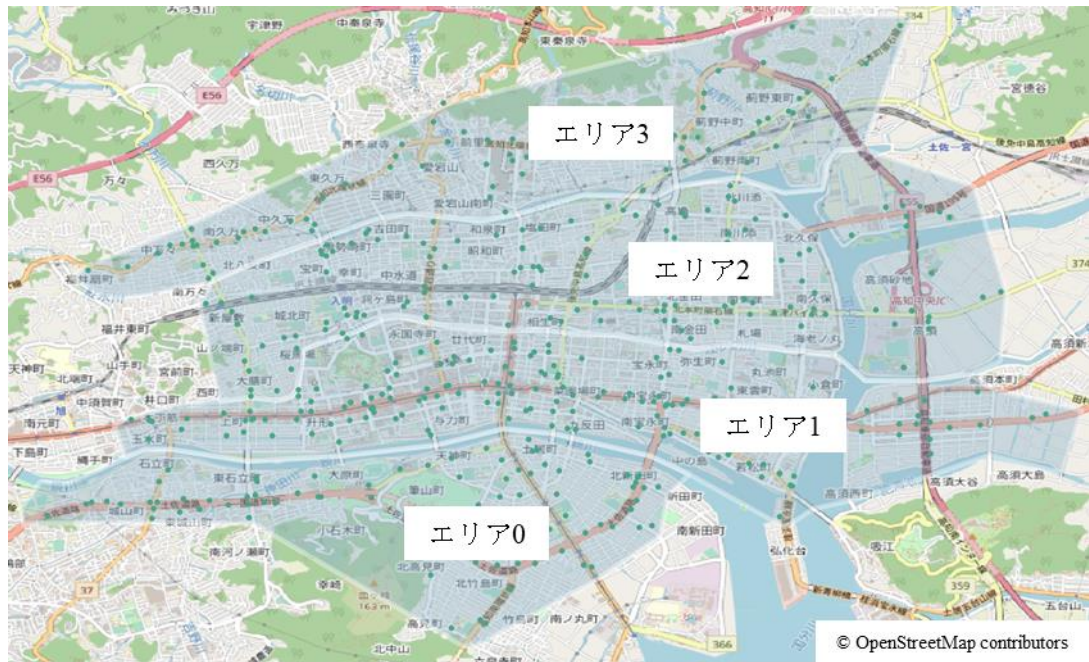


図3 分析エリア

3-4 分析使用データの概要

本節では道路ネットワークの交通流状態を算定するMFDの描画に使用する車両感知器データ・ETC2.0プローブデータ・バスプローブデータの概要について述べる。

3-4-1 車両感知器データ

高知県警察本部の高知県交通管制センターが管轄する高知市の746地点の車両感知器設置地点のうち487地点で観測された車両感知器データを使用する。分析エリア内の車両感知器を抽出するため、感知器配置図(図4)と車両感知器設置場所一覧表(表2)を参考に使用する車両感知器設置地点と車両感知器番号を決定した。

車両感知器データの例を表3に示す。データの期間は2019年4月1日から2020年3月31日までの366日間であり、データは5分単位で記録されている。データ項目は地点番号、地点名称、時間帯、交通量(台)、占有率(%), 速度(km/h)である。データの形式は各車両感知器において、5分単位で観測された交通量, 占有率, 速度が時系列順に1日分並んでいる。

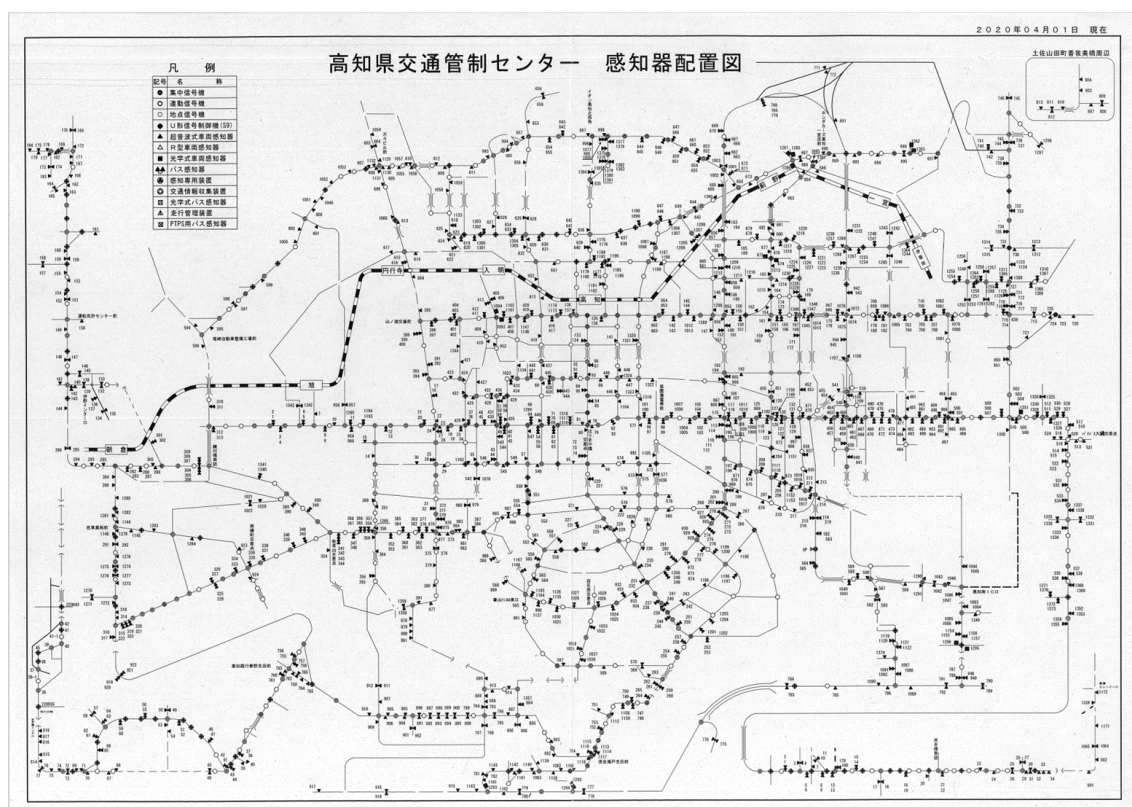


図4 感知器配置図

表 2 車両感知器設置場所一覧表の例

通番	地点名	設置場所	署別	システム通番				備考
				D1	D2	D3	D4	
1	イオン高知旭町店前	高知市旭町3丁目194番地先	高知	1	2			
2	松岡寝具店前	高知市旭町3丁目150番地先	高知	3	4			
3	旭ツール東方	高知市旭町3丁目119番地5先	高知	5	6			
4	かおり化粧品店前	高知市旭駅前町12番地先	高知	7				
5	テンキのマイマイト旭店內側	高知市旭町2丁目28番地先	高知	8	9			
6	ウソのアイ前	高知市旭町2丁目161番地先	高知	10				
7	依岡歯科前	高知市旭町1丁目115番地先	高知	11	1265			
8	シミタネ店前	高知市上町5丁目5番40号先	高知	12	13			
9	谷脇自転車商会前	高知市上町5丁目16番21号先	高知	14				
10	旅館若竹前	高知市下町49番地先	高知	15				
11	食堂若竹前	高知市大瀬町1番19号先	高知	16	17			
12	城西館前	高知市上町1丁目14番13号先	高知	18	19			
13	上町交番前	高知市上町3丁目14番11号先	高知	20	21			
14	日和崎石油前	高知市上町2丁目3番4号先	高知	22				
15	日本共産党前	高知市上町2丁目4番19号先	高知	23				
16	日本石油上町給油所前	高知市上町3丁目6番1号先	高知	24				
17	シオリ美容前	高知市上町3丁目9番8号先	高知	25	26			
18	アスマ補聴器センター前	高知市上町2丁目4番16号先	高知	27	28			
19	藤田方前	高知市上町2丁目9番39号先	高知	29				
20	垣内建設前	高知市上町3丁目11番29号先	高知	30				
21	拓業局前	高知市東石立町49番地先	南	31	32			
22	第一生命高知上町ビル前	高知市上町1丁目6番18号先	高知	33	34			
23	浅井病院前	高知市升形8番2号先	高知	35				
24	升形電停東	高知市升形3番2号先	高知	36	37			
25	高知市消防局前	高知市本町4丁目1番28号先	高知	38	39			
26	サウザンハルステル前	高知市本町4丁目2番48号先	高知	40	41	42		
27	高知クレーン会館前	高知市本町5丁目6番11号先	高知	43	44			
28	津野ビル前	高知市本町5丁目1番15号先	高知	45	46			
29	山本サウザン店前	高知市本町4丁目2番39号先	高知	48				
30	電気ビル前	高知市本町4丁目1番16号先	高知	49	50	51		
31	中の橋通り交差点東方	高知市本町2丁目2番34号先	高知	54	55	56		
32	住友商店前	高知市本町3丁目4番10号先	高知	57				
33	小野川京染店前	高知市追手筋1丁目9番18号先	高知	60				
34	ハチコマイ本町店前	高知市本町1丁目3番22号先	高知	61	62	63		
35	天狗堂本店前	高知市本町2丁目1番12号先	高知	64	1299	66		
36	中の橋通り交差点西方	高知市本町1丁目2番10号先	高知	69	70	71		
37	朝倉町通り北	高知市南はりまや町2丁目1番1号先	高知	72	73	74		
38	はりまや橋派出所前	高知市南はりまや町1丁目5番1号先	高知	78				
39	とでん西武前	高知市南はりまや町1丁目8番1号先	高知	79				
40	はりまや橋交差点南	高知市はりまや町1丁目5番1号先交差点	高知	80				
41	国際ビル高知前	高知市はりまや町1丁目1番2号先	高知	81				
42	NTT高知東営業所向側	高知市はりまや町1丁目14番1号先	高知	82	83			
43	愛媛銀行高知支店前	高知市はりまや町1丁目4番5号先	高知	84	85			
44	ドライクリーニング東京屋前	高知市はりまや町3丁目4番9号先	高知	86	87			
45	あんどろビル前	高知市はりまや町2丁目8番8号先	高知	88				
46	マキ梅花堂前	高知市はりまや町2丁目1番1号先	高知	89	90			
47	前田ビル前	高知市追手筋1丁目3番8号先	高知	91	92			
48	蓮池通電停北	高知市はりまや町2丁目3番3号先交差点	高知	93				
49	仙頭商店前	高知市南はりまや町1丁目14番9号先	高知	94				
50	アティックえる前	高知市堺町7番13号先	高知	95				
51	光洋機工商会ビル前	高知市菜園場町6番7号先	高知	96	97			
52	街蔵のフジモ前	高知市城見町7番7号先	高知	98	99			
53	四国銀行木屋橋支店前	高知市菜園場町1番21号先	高知	100	101			
54	そば処史料前	高知市中宝永町9番5号先	高知	102	103			
55	森本外科前交差点東	高知市中宝永町7番22号先交差点東	高知	104	105			
56	二条通り北	高知市日の出町2丁目8番地先	高知	106	107			
57	光洋機工下地倉庫前	高知市知寄町1丁目6番14号先	高知	108	109			
58	マジックホビー前	高知市南宝永町5番14号先	高知	110	111	112		
59	ダイハレス中宝永町店前	高知市中宝永町11番1号先	高知	113	114			
60	柳和住前	高知市中宝永町5番18号先	高知	115				
61	高知キャノンピア販売前	高知市知寄町1丁目6番21号先	高知	116	117			
62	日本宅業前	高知市知寄町2丁目3番11号先	高知	118	119			
63	日本アルミ前	高知市東雲町1番7号先	高知	120				
64	健康ビル前	高知市知寄町2丁目4番13号先	高知	121	122			
65	土佐醸造酒前	高知市稲荷町2番16号先	高知	123				
66	ザニビル前	高知市知寄町2丁目1番37号先	高知	125	869			
67	四国銀行下知支店前	高知市知寄町2丁目1番27号先	高知	126				
68	スナックキッサもんが前	高知市知寄町2丁目4番2号先	高知	127				
69	ハチコ第一ホームラン前	高知市知寄町3丁目41番地1先	高知	128	129			

表 3 車両感知器データの例

地点番号 地点名称	110011				110012				110013				110014			
	感知器	1	1	0	1	1	0	0	1	2	3	4	5	6	7	8
時間帯	交通量 (台)	占有率 (%)	速度 (km/h)	異常	交通量 (台)	占有率 (%)	速度 (km/h)	異常	交通量 (台)	占有率 (%)	速度 (km/h)	異常	交通量 (台)	占有率 (%)	速度 (km/h)	異常
00:00 ~ 00:05	7	1	37		14	2	44		6	1	52		4	1	42	
00:05 ~ 00:10	7	1	44		5	1	52		5	1	52		3	1	39	
00:10 ~ 00:15	7	1	31		9	1	50		6	1	43		6	1	44	
00:15 ~ 00:20	8	1	33		13	1	47		8	1	51		3	1	38	
00:20 ~ 00:25	8	1	44		10	1	47		11	1	46		3	1	54	
00:25 ~ 00:30	4	1	46		14	1	48		4	1	46		3	1	38	
00:30 ~ 00:35	3	1	45		8	1	53		6	1	49		5	1	38	
00:35 ~ 00:40	4	1	31		9	1	50		5	1	49		4	1	28	
00:40 ~ 00:45	5	1	45		7	1	47		14	1	46		0	0	200	
00:45 ~ 00:50	4	1	35		5	1	34		0	0	200		2	1	46	
00:50 ~ 00:55	6	1	42		8	1	48		2	1	52		5	1	37	
00:55 ~ 01:00	4	1	45		7	1	40		6	1	50		6	1	34	
01:00 ~ 01:05	2	1	52		5	1	46		5	1	50		2	1	39	
01:05 ~ 01:10	2	1	44		10	1	48		4	1	45		2	1	33	
01:10 ~ 01:15	4	1	33		9	1	45		12	1	42		1	1	56	
01:15 ~ 01:20	1	1	49		9	1	46		5	1	49		5	1	38	
01:20 ~ 01:25	1	1	39		5	1	49		2	1	49		4	1	46	
01:25 ~ 01:30	0	0	200		8	1	47		7	1	37		3	1	29	
01:30 ~ 01:35	7	1	39		7	1	40		5	1	52		2	1	44	
01:35 ~ 01:40	3	1	44		3	1	36		6	1	48		2	1	39	
01:40 ~ 01:45	5	1	42		6	1	45		3	1	39		1	1	39	
01:45 ~ 01:50	0	0	200		5	1	34		7	1	40		1	1	56	
01:50 ~ 01:55	1	1	28		5	1	43		2	1	36		2	1	39	
01:55 ~ 02:00	1	1	39		5	1	29		3	1	32		0	0	200	
02:00 ~ 02:05	3	1	37		2	1	49		3	1	36		0	0	200	
02:05 ~ 02:10	3	1	38		5	1	41		3	1	40		0	0	200	
02:10 ~ 02:15	0	0	200		3	1	44		3	1	47		0	0	200	
02:15 ~ 02:20	2	1	39		2	1	34		2	1	49		3	1	29	
02:20 ~ 02:25	3	1	45		4	1	56		2	1	49		4	1	39	
02:25 ~ 02:30	2	1	34		2	1	44		3	1	40		0	0	200	
02:30 ~ 02:35	0	0	200		2	1	56		1	1	39		2	1	33	
02:35 ~ 02:40	1	1	18		6	1	32		1	1	56		1	1	44	
02:40 ~ 02:45	3	1	49		2	1	46		4	1	45		2	1	39	
02:45 ~ 02:50	4	1	42		3	1	54		7	1	44		4	1	34	
02:50 ~ 02:55	4	1	36		1	1	49		1	1	79		1	1	39	
02:55 ~ 03:00	2	1	52		2	1	49		2	1	36		1	1	39	
03:00 ~ 03:05	2	1	46		4	1	46		3	1	56		0	0	200	
03:05 ~ 03:10	1	1	49		3	1	47		5	1	33		0	0	200	
03:10 ~ 03:15	3	1	36		3	1	28		0	0	200		0	0	200	
03:15 ~ 03:20	4	1	38		2	1	41		0	0	200		0	1	1	
03:20 ~ 03:25	4	1	45		2	1	46		5	1	46		2	1	49	
03:25 ~ 03:30	1	1	28		1	1	30		1	1	66		2	1	34	
03:30 ~ 03:35	2	1	33		2	1	46		2	1	56		1	1	44	
03:35 ~ 03:40	4	1	46		0	0	200		2	1	28		0	0	200	
03:40 ~ 03:45	4	1	46		1	1	23		1	1	44		2	1	46	
03:45 ~ 03:50	2	1	41		0	0	200		0	0	200		2	1	25	
03:50 ~ 03:55	4	1	44		6	1	47		1	1	39		0	0	200	
03:55 ~ 04:00	2	1	33		4	1	42		1	1	66		2	1	56	
04:00 ~ 04:05	2	1	44		8	1	40		2	1	52		2	1	41	
04:05 ~ 04:10	2	1	49		6	1	43		1	1	56		1	1	44	
04:10 ~ 04:15	3	1	36		4	1	54		1	1	56		1	1	22	
04:15 ~ 04:20	5	1	35		4	1	48		0	0	200		2	1	37	
04:20 ~ 04:25	6	1	41		6	1	49		5	1	55		0	0	200	
04:25 ~ 04:30	5	1	45		4	1	35		3	1	29		2	1	56	
04:30 ~ 04:35	7	1	32		8	1	36		1	1	33		0	0	200	
04:35 ~ 04:40	4	1	36		5	1	53		1	1	44		1	1	39	
04:40 ~ 04:45	5	1	36		2	1	52		1	1	39		1	1	66	
04:45 ~ 04:50	10	1	42		1	1	49		2	1	46		6	1	42	
04:50 ~ 04:55	3	1	40		7	1	50		2	1	41		3	1	44	
04:55 ~ 05:00	6	1	55		5	1	53		4	1	44		2	1	30	

3-4-2 ETC2.0 プローブデータ

2019年4月1日から2020年3月31日までの1年間に観測されたETC2.0プローブデータを使用する。2019年は1次メッシュごと、2020年は2次メッシュごとにデータが分かれており、本研究では分析エリアである高知市を含むメッシュ・コードを使用する。そのため、2019年は5033のメッシュ・コード、2020年は503324のメッシュ・コードのデータを用いて分析する。

ETC2.0プローブデータには運行情報・車種情報・トリップに関する情報・更新情報が記録されており、色付きの項目を分析に使用した(表4, 表5)。また、ETC2.0プローブ車両の1トリップが走行距離200mごと、進行方向変化45度以上、挙動履歴が閾値を超えた場合に記録され(表6)、トリップ起点時刻が同日の場合は1トリップ内の全てのデータが同じファイルに含まれる。

表4 運行情報・車種情報の詳細項目

運行情報		RSU-IDコード
		受信時刻
		運行日
		運行ID1
車種情報	0: 軽二輪 1: 大型 2: 普通 3: 小型 4: 軽自動車 5~15: 未定義	自動車の種別
	0: 未使用 1: 乗用 2: 貨物 3: 特殊 4: 乗合 5~15: 未定義	自動車の用途

表 5 トリップに関する情報・更新情報の詳細項目

トリップに関する情報	マップマッチング前の情報	GPS時刻
		通し番号
		トリップ番号
		トリップの起点時刻
		トリップの終点時刻
	0：完全トリップ 1：上流側不完全 2：下流側不完全 3：両側不完全	トリップの完全性
	0：起点 1：終点 2：通過点 3：その他	トリップ起終点フラグ
		タグ番号
		経度
		緯度
	0：走行距離による蓄積 1：進行方向変化による蓄積 2：挙動履歴による蓄積	蓄積条件
	未使用 0：高速 1：市内高速 2：一般道路 3：その他 電波ビーコン5、8GHz帯データ形式仕様書 【走行履歴情報1】 4-2-5 + 4-3-7	道路種別コード
		速度
		高度
	マップマッチング後の情報	マッチングフラグ
		マッチング後経度
		マッチング後緯度
		DRMバージョン
		2次メッシュコード
		流入ノード
		流出ノード
		流入ノードからの距離
		確定フラグ
	交通調査基本区間番号	
1：上り 2：下り	上り・下りコード	
	管理者コード	
更新情報	更新日時	

表 6 ETC2.0 プローブデータの例

RSU-IDコード	受信時刻	運行日	運行ID	自動車の種別	自動車の用途	GPS時刻	通し番号	トリップ番号	トリップの起点時刻
4007203F	20190403174344	20190401	452584	2	1	20190401000932	1	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401000939	2	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401000946	3	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401000952	4	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401000958	5	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001004	6	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001011	7	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001017	8	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001024	9	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001031	10	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001038	11	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001044	12	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001051	13	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001058	14	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001105	15	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001112	16	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001121	17	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001128	18	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001135	19	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001143	20	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001152	21	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001200	22	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001210	23	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001223	24	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001232	25	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001251	26	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001307	27	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001323	28	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001334	29	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001347	30	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001431	31	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001444	32	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001456	33	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001506	34	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001518	35	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001532	36	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001554	37	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001605	38	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001617	39	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001632	40	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001647	41	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001659	42	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001713	43	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001729	44	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001742	45	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001754	46	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001806	47	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001822	48	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001838	49	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001852	50	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001914	51	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001920	52	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401001926	53	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401002003	54	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401002230	55	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401002232	56	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401002319	57	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401002326	58	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401002351	59	1	20190401000932
4007203F	20190403174344	20190401	452584	2	1	20190401002405	60	1	20190401000932

3-4-3 バスプローブデータ

とさでん交通株式会社が運行する路線バスのバスロケーションデータを本研究ではバスプローブデータとして使用し、分析に使用する項目は色付きで示す(表7)。GPSを利用してバス停の位置情報を収集し、利用者にバスの運行情報を提供するバスロケーションシステムで取得したデータである。

データの期間は車両感知器データやETC2.0プローブデータと同様に2019年4月1日から2020年3月31日までである。データはバス停ごとに記録され、ETC2.0プローブデータと同様に1トリップごとに並んでいる(表8)。

表7 バスプローブデータの記載項目

	記載項目		記載項目
1	運行日付	19	通過時刻
2	仕業番号	20	遅れ時間
3	仕業内連番	21	距離
4	系統コード	22	GPS位置情報・緯度
5	曜日区分	23	GPS位置情報・経度
6	便番号	24	補正位置情報・緯度
7	臨時便フラグ	25	補正位置情報・経度
8	臨時便連番	26	会社コード
9	系統内連番	27	営業所コード
10	停留所コード	28	登録者
11	出発時刻(予定)	29	登録業務ID
12	車両コード(予定)	30	登録日
13	車載端末コード(予定)	31	登録時分秒
14	系統番号	32	更新者
15	系統名(正式名称)	33	更新業務ID
16	停留所名(正式名称)	34	更新日
17	停留所名(MC用)	35	更新時分秒
18	車番		

表 8 バスプローブデータの例

系統内連番	停留所コード	出発時刻(予定)	車両コード(予定)	車載端末コード(予定)	系統番号	系統名(正式名称)	停留所名(正式名称)	停留所名(MC用)	車番	通過時刻
0	4000	6:50	4018	356793060138716	G2W1	一宮BT→比島→上	一宮バスターミナル	一宮BT	き378	6:49
1	4150	6:50	4018	356793060138716	G2W1	一宮BT→比島→上	布師田分岐	布師田分岐	き378	6:52
2	4151	6:51	4018	356793060138716	G2W1	一宮BT→比島→上	薊野東	薊野東	き378	6:52
3	4152	6:51	4018	356793060138716	G2W1	一宮BT→比島→上	薊野西	薊野西	き378	6:53
4	4153	6:51	4018	356793060138716	G2W1	一宮BT→比島→上	薊野西	薊野西	き378	6:54
5	4154	6:52	4018	356793060138716	G2W1	一宮BT→比島→上	比島橋	比島橋	き378	6:54
6	4155	6:53	4018	356793060138716	G2W1	一宮BT→比島→上	比島	比島	き378	6:55
7	16	6:54	4018	356793060138716	G2W1	一宮BT→比島→上	中比島	中比島	き378	6:55
8	15	6:54	4018	356793060138716	G2W1	一宮BT→比島→上	比島アパート前	比島アパート	き378	6:56
9	14	6:55	4018	356793060138716	G2W1	一宮BT→比島→上	日赤前	日赤前	き378	6:57
10	13	7:00	4018	356793060138716	W1	一宮BT→比島→上	高知駅バスターミナル	高知駅B T	き378	6:58
11	12	7:03	4018	356793060138716	W1	一宮BT→比島→上	高知駅前	高知駅前	き378	7:02
12	11	7:03	4018	356793060138716	W1	一宮BT→比島→上	高知橋	高知橋	き378	7:03
13	10	7:04	4018	356793060138716	W1	一宮BT→比島→上	蓮池町通	蓮池町通	き378	7:04
14	9	7:05	4018	356793060138716	W1	一宮BT→比島→上	北はりまや橋	北はりまや橋	き378	7:06
15	61	7:08	4018	356793060138716	W1	一宮BT→比島→上	堺町	堺町	き378	7:07
16	62	7:08	4018	356793060138716	W1	一宮BT→比島→上	堀詰	堀詰	き378	7:08
17	63	7:09	4018	356793060138716	W1	一宮BT→比島→上	大橋通	大橋通	き378	7:10
18	64	7:10	4018	356793060138716	W1	一宮BT→比島→上	高知城前	高知城前	き378	7:10
19	65	7:12	4018	356793060138716	W1	一宮BT→比島→上	県庁前	県庁前	き378	7:10
20	66	7:13	4018	356793060138716	W1	一宮BT→比島→上	グランド通	グランド通	き378	7:12
21	67	7:13	4018	356793060138716	W1	一宮BT→比島→上	升形	升形	き378	7:13
22	68	7:14	4018	356793060138716	W1	一宮BT→比島→上	上町一丁目	上町一丁目	き378	7:13
23	69	7:15	4018	356793060138716	W1	一宮BT→比島→上	上町二丁目	上町二丁目	き378	7:17
24	4300	7:16	4018	356793060138716	W1	一宮BT→比島→上	月の瀬橋	月の瀬橋	き378	7:19
25	4301	7:18	4018	356793060138716	W1	一宮BT→比島→上	河の瀬	河の瀬	き378	7:19
26	4302	7:18	4018	356793060138716	W1	一宮BT→比島→上	市民会館前	市民会館前	き378	7:20
27	4303	7:19	4018	356793060138716	W1	一宮BT→比島→上	南河ノ瀬	南河ノ瀬	き378	7:21
28	4305	7:19	4018	356793060138716	W1	一宮BT→比島→上	幸崎	幸崎	き378	7:21
29	4306	7:21	4018	356793060138716	W1	一宮BT→比島→上	吉野緑ヶ丘	吉野緑ヶ丘	き378	7:21
30	4307	7:21	4018	356793060138716	W1	一宮BT→比島→上	東赤坂	東赤坂	き378	7:22
31	4308	7:23	4018	356793060138716	W1	一宮BT→比島→上	学校入口	学校入口	き378	7:23
32	4309	7:29	4018	356793060138716	W1	一宮BT→比島→上	土佐塾校	土佐塾校	き378	7:27
0	4309	17:30	4018	356793060138716	W1G2	土佐塾→駐輪→上	土佐塾校	土佐塾校	き378	17:29
1	4308	17:35	4018	356793060138716	W1G2	土佐塾→駐輪→上	学校入口	学校入口	き378	17:35
2	4307	17:35	4018	356793060138716	W1G2	土佐塾→駐輪→上	東赤坂	東赤坂	き378	17:36
3	4306	17:37	4018	356793060138716	W1G2	土佐塾→駐輪→上	吉野緑ヶ丘	吉野緑ヶ丘	き378	17:38
4	4305	17:37	4018	356793060138716	W1G2	土佐塾→駐輪→上	幸崎	幸崎	き378	17:38
5	4304	17:40	4018	356793060138716	W1G2	土佐塾→駐輪→上	幸崎駐輪場	幸崎駐輪場	き378	17:39
6	4303	17:41	4018	356793060138716	W1G2	土佐塾→駐輪→上	南河ノ瀬	南河ノ瀬	き378	17:41
7	4302	17:42	4018	356793060138716	W1G2	土佐塾→駐輪→上	市民会館前	市民会館前	き378	17:42
8	4301	17:43	4018	356793060138716	W1G2	土佐塾→駐輪→上	河の瀬	河の瀬	き378	17:44
9	4300	17:43	4018	356793060138716	W1G2	土佐塾→駐輪→上	月の瀬橋	月の瀬橋	き378	17:44
10	69	17:45	4018	356793060138716	G2	土佐塾→駐輪→上	上町二丁目	上町二丁目	き378	17:47
11	68	17:46	4018	356793060138716	G2	土佐塾→駐輪→上	上町一丁目	上町一丁目	き378	17:47
12	67	17:46	4018	356793060138716	G2	土佐塾→駐輪→上	升形	升形	き378	17:48
13	66	17:47	4018	356793060138716	G2	土佐塾→駐輪→上	グランド通	グランド通	き378	17:48
14	65	17:48	4018	356793060138716	G2	土佐塾→駐輪→上	県庁前	県庁前	き378	17:50
15	64	17:48	4018	356793060138716	G2	土佐塾→駐輪→上	高知城前	高知城前	き378	17:50
16	63	17:49	4018	356793060138716	G2	土佐塾→駐輪→上	大橋通	大橋通	き378	17:51
17	62	17:50	4018	356793060138716	G2	土佐塾→駐輪→上	堀詰	堀詰	き378	17:53
18	61	17:52	4018	356793060138716	G2	土佐塾→駐輪→上	堺町	堺町	き378	17:54
19	9	17:53	4018	356793060138716	G2	土佐塾→駐輪→上	北はりまや橋	北はりまや橋	き378	17:56
20	10	17:53	4018	356793060138716	G2	土佐塾→駐輪→上	蓮池町通	蓮池町通	き378	17:56
21	11	17:54	4018	356793060138716	G2	土佐塾→駐輪→上	高知橋	高知橋	き378	17:57
22	12	17:57	4018	356793060138716	G2	土佐塾→駐輪→上	高知駅前	高知駅前	き378	17:57
23	13	18:00	4018	356793060138716	G2	土佐塾→駐輪→上	高知駅バスターミナル	高知駅B T	き378	17:59
24	14	18:01	4018	356793060138716	G2	土佐塾→駐輪→上	日赤前	日赤前	き378	18:02
25	15	18:02	4018	356793060138716	G2	土佐塾→駐輪→上	比島アパート前	比島アパート	き378	18:03
26	16	18:02	4018	356793060138716	G2	土佐塾→駐輪→上	中比島	中比島	き378	18:04
27	4155	18:03	4018	356793060138716	G2	土佐塾→駐輪→上	比島	比島	き378	18:05
28	4154	18:04	4018	356793060138716	G2	土佐塾→駐輪→上	比島橋	比島橋	き378	18:06
29	4153	18:04	4018	356793060138716	G2	土佐塾→駐輪→上	薊野西	薊野西	き378	18:07
30	4152	18:04	4018	356793060138716	G2	土佐塾→駐輪→上	薊野	薊野	き378	18:08

4 道路ネットワークの交通流状態の算定方法

4-1 車両感知器データによる MFD の描画

本節では車両感知器データから流率と密度を算出し、MFD を描画するまでの手順について述べる。

4-1-1 MFD の算出式

式(1)の算出式を用いて MFD を描画し、走行台キロが流率、車両存在台数が密度を示す指標とする。走行台キロ P_t は式(1)で示すように、時間帯 t に観測されたリンク i の交通量 q_t^i とリンク i の長さ l^i の積の分析エリア全体での総和である。車両存在台数 N_t は式(2)で示すように、時間帯 t に分析エリア内のリンク i の各感知地点で観測されたリンク i の交通量 q_t^i 、速度 v_t^i から計算された交通密度 k_t^i とリンク i の長さ l^i の積の分析エリア全体での総和である。

$$P_t = \sum_i^I q_t^i l^i \quad (1)$$

$$N_t = \sum_i^I k_t^i l^i = \sum_i^I \left(\frac{q_t^i}{v_t^i} \right) l^i \quad (2)$$

P_t : 走行台キロ [台・km/時間]

N_t : 車両存在台数 [台・時/時間]

q_t^i : 車両感知器の交通量 [台]

k_t^i : 車両感知器の密度 [台・h/km]

v_t^i : 車両感知器の速度 [km/h]

l^i : リンクの長さ [km]

4-1-2 リンクの長さの算出

リンクの長さ l^i の定義は車両感知器設置地点を含む対象道路の交差点間の距離とする(図 5)。交差点間に分析エリアの境界線がある場合は交差点から境界線までをリンクの長さとする。GEOFABRIK downloads¹⁴⁾から道路の情報を取得し、QGIS 上に表示した。道路の中心線を計測し、距離の計測には QGIS の線の長さを測る機能を用いた。

車両感知器設置地点を QGIS 上に表示するため、車両感知器設置場所一覧表(表 2)に記載された設置地点の住所からジオコーディング¹⁵⁾を用いて緯度と経度を取得した。さらに、車両感知器設置地点を QGIS 上で地図と重ねて合わせ、感知器配置図(図 4)の信号機との位置関係を比較した。修正が必要な車両感知器設置地点については緯度と経度を再度取得した。修正を加えた車両感知器設置地点を図 6 に示す¹³⁾。そして、車両感知器の番号とリンクの番号、リンクの長さを分析エリアごとにまとめた(表 9)。

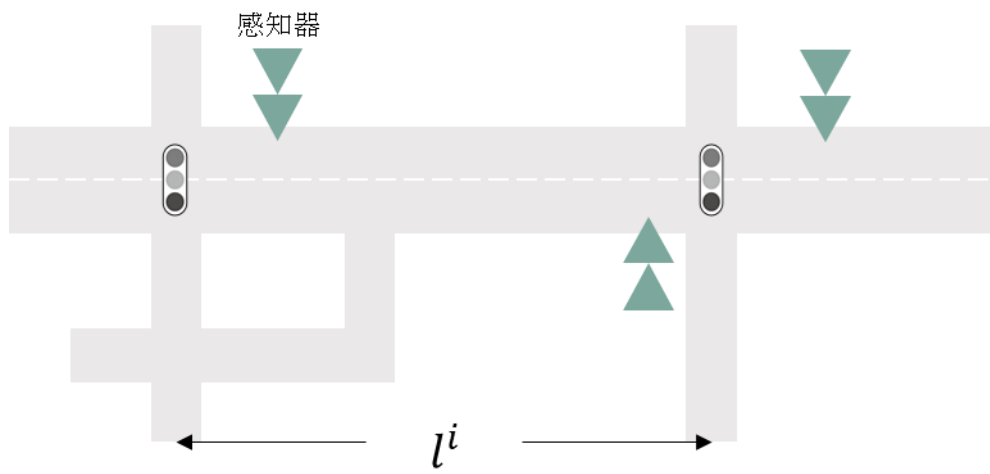


図 5 リンクの長さの定義

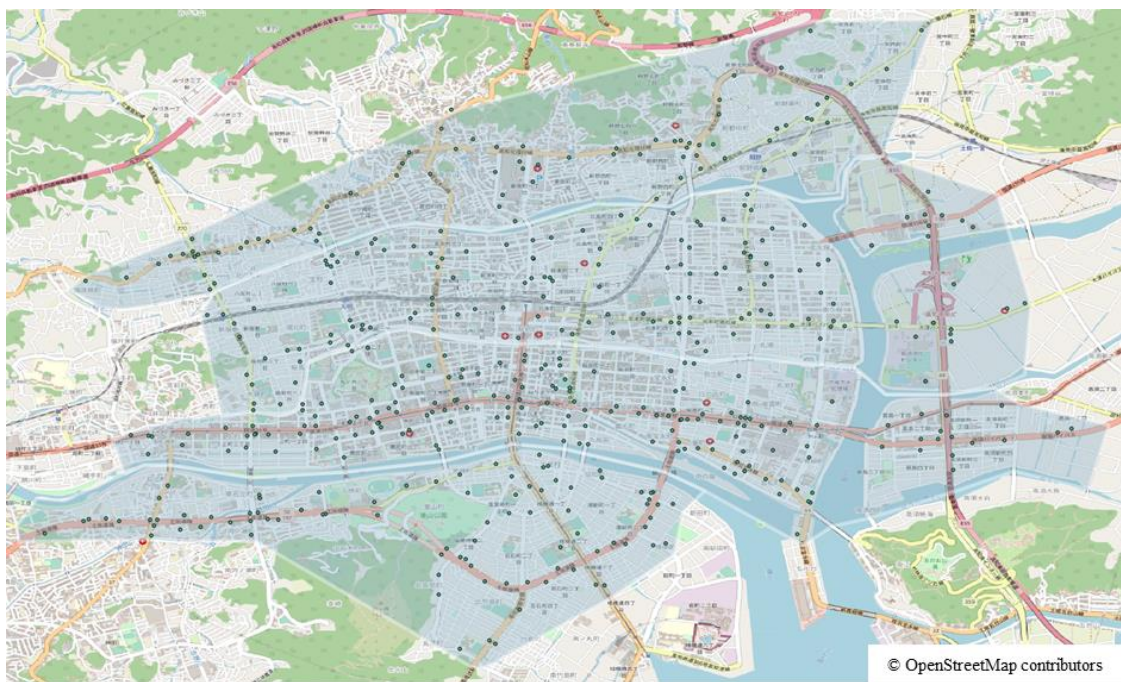


図 6 車両感知器設置地点

表 9 リンク情報の例

車両感知器番号	リンク番号	リンクの長さ[km]
110341	1	0.241
110342	1	0.241
110343	1	0.241
110344	1	0.241
110347	2	0.160
110348	2	0.160
110365	3	0.472
110366	3	0.472
110367	3	0.472
110368	3	0.472
110356	3	0.472
110357	3	0.472
111266	3	0.472
110359	4	0.147
110364	4	0.147
110358	5	0.089
110351	6	0.087
110369	7	0.161
110352	8	0.226
110353	8	0.226
110384	9	0.159
110385	9	0.159
110360	9	0.159
110361	9	0.159
110382	10	0.319
110383	10	0.319
110362	10	0.319
110363	10	0.319
110376	10	0.319
110377	10	0.319
110978	10	0.319
110375	11	0.082
110378	11	0.082
110975	12	0.152
110976	12	0.152
110370	12	0.152
110371	12	0.152
110372	12	0.152
110031	13	0.095
110032	13	0.095

4-1-3 流率と密度の算出

5分間に車両感知器で観測された交通量、占有率、速度のデータを車両感知器の番号ごとに分けた(表 10)。さらに、分析エリア内の車両感知器の観測データを1時間ごとに集計し、リンクの情報を加え、車両感知器の番号、交通量、速度、車両感知器が属するリンクの長さのデータにまとめた(表 11)。交通量とリンクの長さから流率にあたる走行台キロと交通量と速度、リンクの長さから密度にあたる車両存在台数を車両感知器ごとに算出した(表 12)。最後に分析エリア内の車両感知器の流率と密度をそれぞれ足し合わせた(表 13)。

表 10 車両感知器ごとの観測データの例

交通量[台]	占有率[%]	速度[km/h]
7	1	37
7	1	44
7	1	31
8	1	33
8	1	44
4	1	46
3	1	45
4	1	31
5	1	45
4	1	35
6	1	42
4	1	45
2	1	52
2	1	44
4	1	33
1	1	49
1	1	39
0	0	200
7	1	39
3	1	44
5	1	42
0	0	200
1	1	28
1	1	39
3	1	37
3	1	38
0	0	200
2	1	39
3	1	45
2	1	34

表 11 車両感知器ごとの情報の例

車両感知器番号	交通量[台]	速度[km/h]	リンクの長さ[km]
110031	38	33.3	0.095
110032	31	69.2	0.095
110200	60	43.4	0.232
110201	110	46.7	0.232
110202	0	200.0	0.232
110222	19	60.8	0.21
110223	39	29.2	0.499
110224	21	76.2	0.143
110225	45	44.8	0.143
110226	3	166.4	0.192
110227	2	175.0	0.192
110228	56	46.8	0.189
110229	39	77.0	0.189
110230	7	123.7	0.126
110231	24	63.6	0.345
110232	40	44.6	0.345
110234	59	50.7	0.345
110235	43	46.7	0.345
110237	0	0.0	0.127
110238	0	0.0	0.127
110240	46	48.8	0.126
110241	37	46.1	0.126
110248	58	43.3	0.17
110249	87	38.8	0.17
110266	16	78.3	0.216
110267	12	97.6	0.216
110268	21	51.1	0.165
110269	19	94.3	0.232
110270	2	168.3	0.139
110271	72	47.8	0.139

表 12 車両感知器ごとの走行台キロと車両存在台数の例

走行台キロ[台・km/時間]	車両存在台数[台・時/時間]
3.610	0.108
2.945	0.043
13.920	0.321
25.520	0.547
0.000	0.000
3.990	0.066
19.461	0.667
3.003	0.039
6.435	0.144
0.576	0.003
0.384	0.002
10.584	0.226
7.371	0.096
0.882	0.007
8.280	0.130
13.800	0.310
20.355	0.402
14.835	0.318
5.796	0.119
4.662	0.101
9.860	0.228
14.790	0.382
3.456	0.044
2.592	0.027
3.465	0.068
4.408	0.047
0.278	0.002
10.008	0.209
16.402	0.362
0.000	0.000

表 13 分析エリアの流率と密度の例

流率[台・km/時間]	密度[台・時/時間]
1827.788	51.308
1163.166	36.784
921.426	28.557
906.543	29.425
1479.281	39.719
2729.971	69.930
8055.977	250.067
15726.033	677.997
14968.433	612.525
13451.159	465.349
13944.388	497.881
12869.810	448.636
12850.068	490.598
13602.456	483.438
13886.694	513.558
13318.611	482.034
13925.238	546.800
15841.680	640.068
15098.297	770.927
12075.972	467.767
8238.591	255.923
6120.988	196.156
4212.471	127.928
2511.529	66.708
1845.332	48.170
1177.751	33.895
930.930	28.262
934.568	32.307
1357.225	41.480
2661.133	72.905

4-1-4 MFD の描画

車両感知器データより算出した流率と密度を用いてエリア 0 からエリア 3 までそれぞれの MFD を描画した(図 7~10). 縦軸に流率[台・km/時間], 横軸に密度[台・時/時間]をとり, 1 点は 1 時間単位の流率と密度の値を示す.

分析エリア 0 は流率の最大が 17000[台・km/時]程度, 密度の最大が 1100[台・時/時間]程度である. 自由流付近のデータの幅は狭く, 自由流では同じような流率と密度の値を取るデータが多く重なっていることがわかる. 渋滞流に近づくにつれデータの分布の幅が広がっている. さらに, 交通容量(臨界交通流率)から低下していく様子も見られた.

分析エリア 1 は流率の最大が 22000[台・km/時]程度, 密度の最大が 1650[台・時/時間]程度である. 自由流のデータの幅は一定であり, 渋滞流では同じ密度で複数の流率が観測された. また, 分析エリアの交通容量(臨界交通流率)付近のデータがいくつも観測された.

分析エリア 2 は流率の最大が 22000[台・km/時]程度, 密度の最大が 1650[台・時/時間]程度であり, 分析エリア 1 と同じくらいの容量である. すべての交通流状態においてデータの幅がほぼ一定である. また, 分析エリアの交通容量(臨界交通流率)付近のデータがいくつも観測された.

分析エリア 3 は流率の最大が 11000[台・km/時]程度, 密度の最大が 800[台・時/時間]程度であり, 他の分析エリアよりも容量が小さい. 渋滞流だけでなく, 自由流においてもデータの幅は広く, 自由流において, 傾きである速度が低いデータも見られた. また, 交通容量(臨界交通流率)に近いデータが見られた.

すべての分析エリアの MFD から, 交通容量(臨界交通流率)に近いデータが見られた. そのため, 高知市においても渋滞流の観測が可能であることが明らかになった.

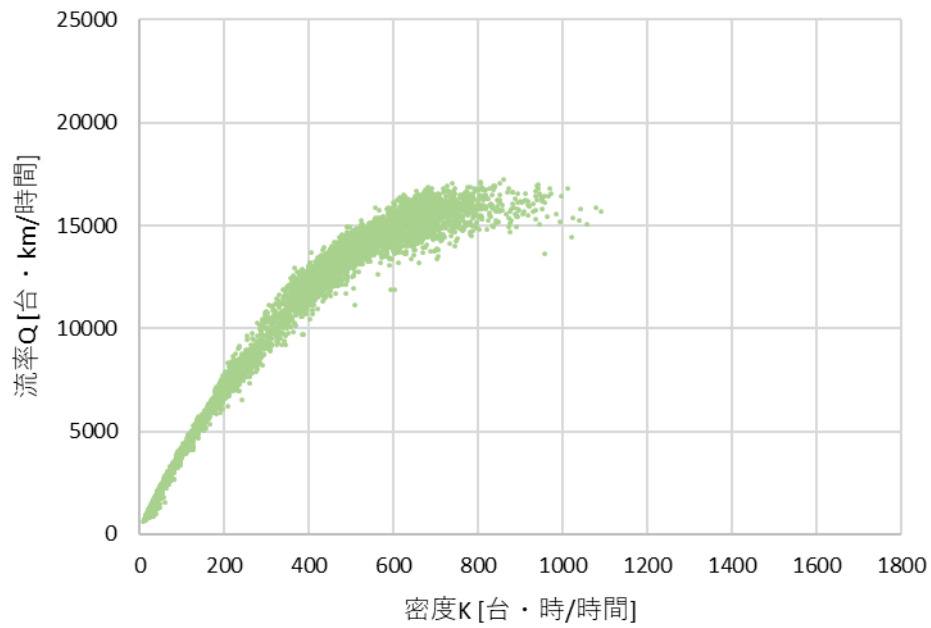


図 7 分析エリア 0 の車両感知器 MFD

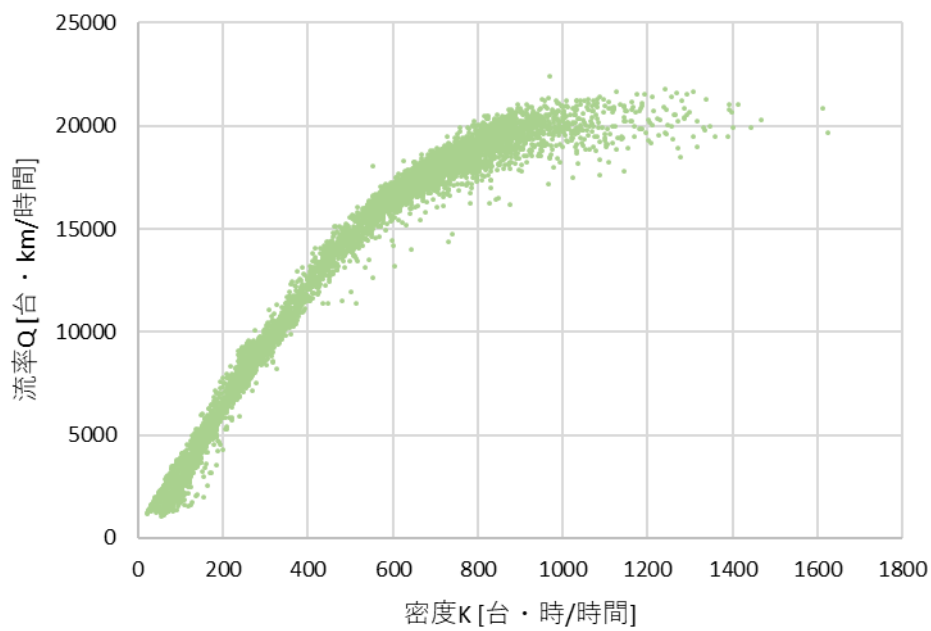


図 8 分析エリア 1 の車両感知器 MFD

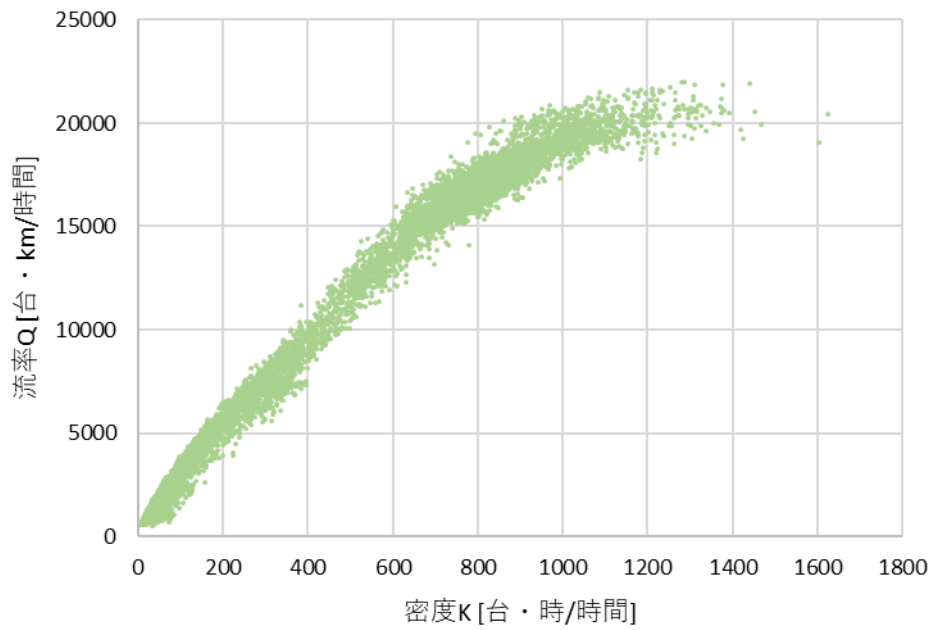


図 9 分析エリア 2 の車両感知器 MFD

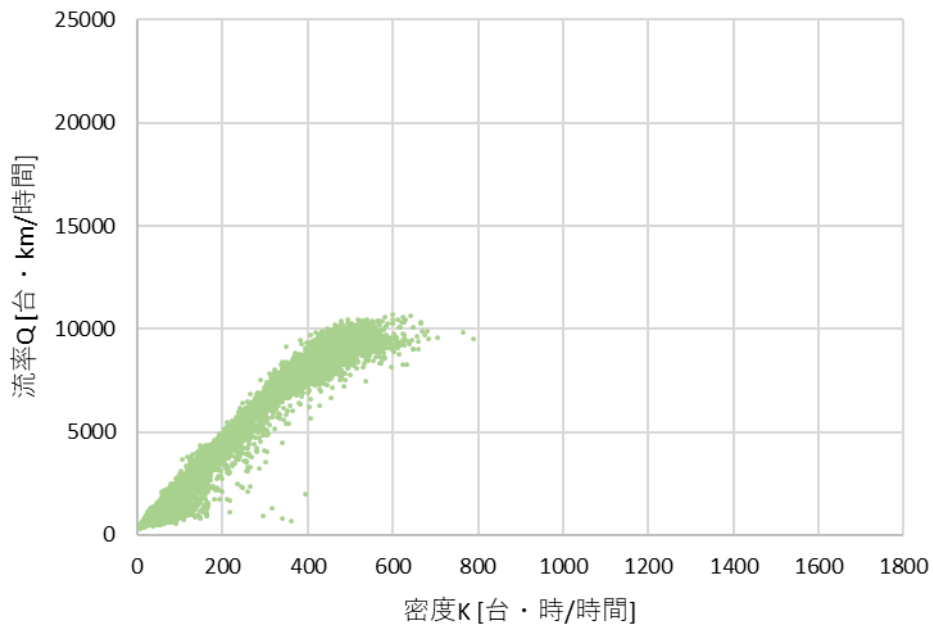


図 10 分析エリア 3 の車両感知器 MFD

4-2 ETC2.0 プローブデータによる MFD の描画

本節では ETC2.0 プローブデータから流率と密度を算出し、MFD を描画するまでの手順について述べる。

4-2-1 MFD の算出式

奥原ら¹⁰⁾の算出式を用いて 1 時間あたりに分析エリア内を走行する ETC2.0 プローブ車両の軌跡データを集計し、MFD を描画する。プローブエリア流率 q_t は式(3)で示すように、時間帯 t における ETC2.0 プローブ車両 i の走行距離 d_{it} の総和である。走行距離の単位は km とする。プローブエリア密度 k_t は式(4)で示すように、時間帯 t における ETC2.0 プローブ車両 i の走行時間 o_{it} の総和である。走行時間は時間の単位で集計する。また、 A_t は時間帯 t に分析エリア内を走行した ETC2.0 プローブ車両の集合である。

$$q_t = \sum_{i \in A_t} d_{it} \quad (3)$$

$$k_t = \sum_{i \in A_t} o_{it} \quad (4)$$

q_t : 時間帯 t における ETC2.0 プローブ流率 [台・km/時間]

k_t : 時間帯 t における ETC2.0 プローブ密度 [台・時/時間]

d_{it} : 時間帯 t における ETC2.0 プローブ車両 i の走行距離 [km]

o_{it} : 時間帯 t における ETC2.0 プローブ車両 i の走行時間 [時]

A_t : 時間帯 t に分析エリア内を走行した ETC2.0 プローブ車両の集合

4-2-2 分析エリア内を走行する車両の判別

分析エリア内を走行する ETC2.0 プローブ車両を抽出するため、分析エリアの境界線の緯度と経度を取得する。QGIS 上で分析エリアのポリゴンを作成し、フィールド計算機を用いてポリゴンの頂点の緯度と経度を算出した。

ETC2.0 プローブデータが分析エリア内に含まれるか、分析エリアのポリゴンを用いて内外判定をする。そこで、分析エリアの境界線の緯度と経度、ETC2.0 プローブデータのマッチング後緯度とマッチング後経度を比較した。

4-2-3 流率と密度の算出

分析エリア内に含まれる ETC2.0 プローブデータを抽出し, ETC2.0 プローブデータの GPS 時刻を参考に 1 時間ごとのデータに分類した(表 14). ETC2.0 プローブデータの運行 ID1 とトリップの起点時刻, トリップの終点時刻が一致するデータを ETC2.0 プローブ車両 1 台の 1 トリップとみなし計算する.

走行時間は ETC2.0 プローブデータの一番遅い GPS 時刻から一番早い GPS 時刻を引くことによって求める. 走行距離は ETC2.0 プローブデータの速度の平均値と走行時間を掛け合わせるによって求める. ETC2.0 プローブ車両ごとに求めた速度と走行時間, 走行距離を表 15 に示す. 最後に分析エリア内の ETC2.0 プローブ車両の走行時間と走行距離をそれぞれ足し合わせ流率と密度を算出した(表 16).

表 14 分析エリア内の ETC2.0 プローブデータの例

RSU-IDコード	受信時刻	運行日	運行ID1	自動車の種別	自動車の用途	GPS時刻	通し番号	トリップ番号	トリップの起点時刻
40072043	20190405162105	20190401	69760	2	1	20190401002332	102	20	20190401000027
40072043	20190405162105	20190401	69760	2	1	20190401002344	103	20	20190401000027
40072043	20190405162105	20190401	69760	2	1	20190401002357	104	20	20190401000027
40072043	20190405162105	20190401	69760	2	1	20190401002409	105	20	20190401000027
40072043	20190405162105	20190401	69760	2	1	20190401002516	106	20	20190401000027
40072043	20190405162105	20190401	69760	2	1	20190401002543	107	20	20190401000027
40072043	20190405162105	20190401	69760	2	1	20190401002556	108	20	20190401000027
40072043	20190405162105	20190401	69760	2	1	20190401003350	109	20	20190401000027
40072043	20190405162105	20190401	69760	2	1	20190401003402	110	20	20190401000027
40072043	20190405162105	20190401	69760	2	1	20190401003405	111	20	20190401000027
40072043	20190405162105	20190401	69760	2	1	20190401003433	112	20	20190401000027
40072043	20190405162105	20190401	69760	2	1	20190401003435	113	20	20190401000027
40072042	20190403171359	20190401	326160	2	1	20190401004116	9	6	20190401003759
40072042	20190403171359	20190401	326160	2	1	20190401004204	10	6	20190401003759
40072042	20190403171359	20190401	326160	2	1	20190401004046	7	6	20190401003759
40072042	20190403171359	20190401	326160	2	1	20190401004056	8	6	20190401003759
40072042	20190401005626	20190401	13	2	1	20190401003502	12	1	20190401002531
40072042	20190401005626	20190401	13	2	1	20190401003524	13	1	20190401002531
40072042	20190401005626	20190401	13	2	1	20190401003638	15	1	20190401002531
40072042	20190401005626	20190401	13	2	1	20190401003655	16	1	20190401002531
600A2046	20190401115725	20190401	47906	2	1	20190401000259	3	2	20190401000230
600A2046	20190401115725	20190401	47906	2	1	20190401000314	4	2	20190401000230
600A2046	20190401115725	20190401	47906	2	1	20190401000338	5	2	20190401000230
4007203F	20190401182102	20190401	211834	2	1	20190401004742	2	11	20190401004730
4007203F	20190401182102	20190401	211834	2	1	20190401004753	3	11	20190401004730
4007203F	20190401182102	20190401	211834	2	1	20190401002850	12	10	20190401002535
4007203F	20190401182102	20190401	211834	2	1	20190401003027	13	10	20190401002535
4007203F	20190401182102	20190401	211834	2	1	20190401003043	14	10	20190401002535
4007203F	20190401182102	20190401	211834	2	1	20190401003058	15	10	20190401002535
4007203F	20190401182102	20190401	211834	2	1	20190401003110	16	10	20190401002535

表 15 ETC2.0 プローブ車両ごとの情報の例

速度[km/h]	走行時間[h]	走行距離[km]
44.750	0.027	1.193
46.083	0.090	4.148
43.045	0.113	4.843
63.400	0.012	0.757
44.000	0.013	0.574
45.375	0.036	1.651
52.667	0.019	0.995
35.600	0.043	1.543
49.632	0.090	4.481
39.000	0.009	0.336
52.429	0.076	3.976
61.786	0.100	6.161
39.000	0.018	0.683
54.000	0.056	3.030
36.750	0.115	4.236
50.714	0.049	2.493
38.500	0.022	0.845
58.731	0.131	7.700
35.167	0.040	1.397
62.500	0.003	0.208
51.875	0.029	1.527
53.240	0.154	8.208
43.462	0.061	2.668
54.979	0.186	10.232
60.000	0.000	0.000
20.294	0.206	4.188
38.200	0.014	0.520
47.900	0.043	2.036
45.833	0.015	0.688
46.300	0.054	2.482

表 16 分析エリア内の ETC2.0 プローブ流率と密度の例

流率[台・km/時間]	密度[台・時/時間]
43.542	0.970
21.132	0.479
6.445	0.150
7.240	0.155
13.843	0.273
17.055	0.380
120.828	2.951
414.032	12.192
391.279	12.779
257.976	7.743
157.042	4.585
229.566	6.436
259.910	7.019
178.022	4.562
187.893	5.112
248.915	6.802
232.511	6.030
260.377	7.159
405.752	8.989
262.978	6.857
126.048	2.814
110.685	2.648
48.044	1.111
30.425	0.661
28.317	0.746
6.900	0.138
5.920	0.104
6.730	0.146
7.858	0.152
22.268	0.482

4-2-4 MFD の描画

ETC2.0 プローブデータより算出した流率と密度を用いてエリア 0 からエリア 3 までそれぞれの MFD を描画した(図 11~14). 縦軸に流率[台・km/時間], 横軸に密度[台・時/時間]をとり, 1 点は 1 時間単位の流率と密度の値を示す.

流率の最大値は分析エリア 0 が 1000[台・km/時]程度, 分析エリア 1 が 1600[台・km/時]程度, 分析エリア 2 が 1700[台・km/時]程度, 分析エリア 3 が 500[台・km/時]程度である. また, 密度の最大値は分析エリア 0 が 30[台・時/時間]程度, 分析エリア 1 が 52[台・時/時間]程度, 分析エリア 2 が 52[台・時/時間]程度, 分析エリア 3 が 16[台・時/時間]程度である.

直線的な分布であるが, データの幅は変化している. そのため, 自由流領域と定義しても, その中に複数の交通流状態が見られる可能性がある. しかしながら, 傾きである速度の変化はあまり見られない. これは実際に渋滞流であったとしても高知市において ETC2.0 プローブ車両の割合が少ないため, データから算出される速度の値に変化が見られないと推測できる.

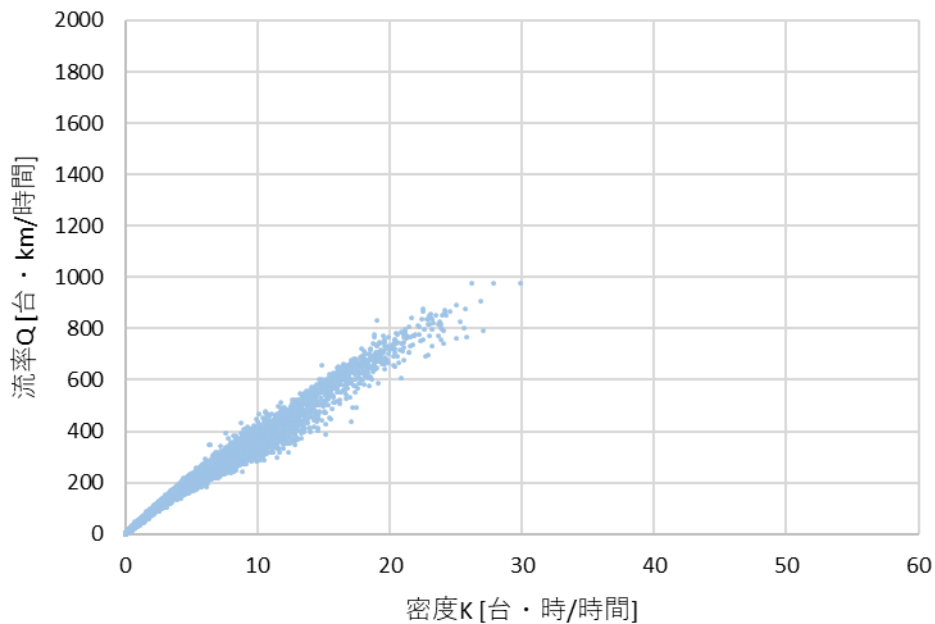


図 11 分析エリア 0 の ETC2.0 プローブ MFD

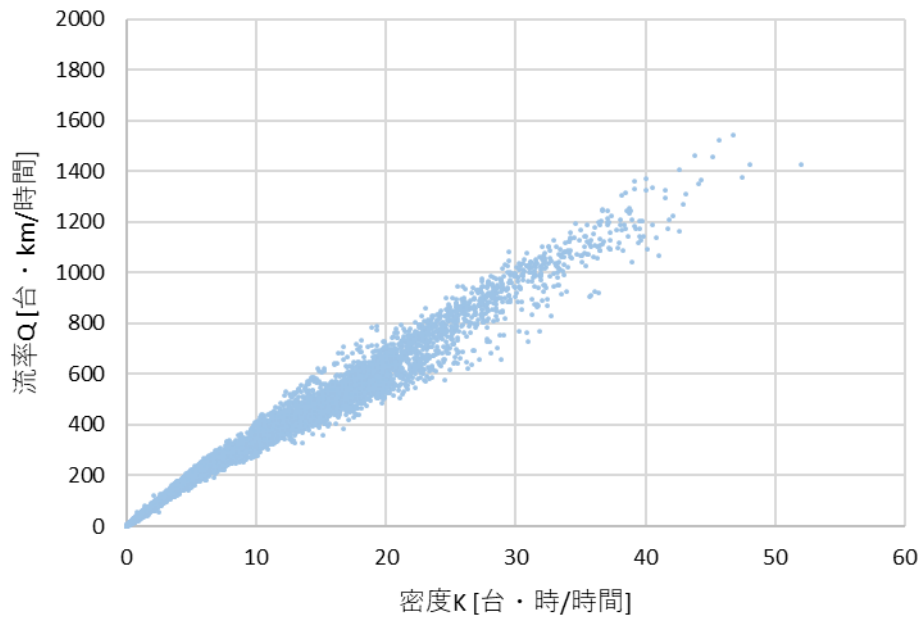


図 12 分析エリア 1 の ETC2.0 プローブ MFD

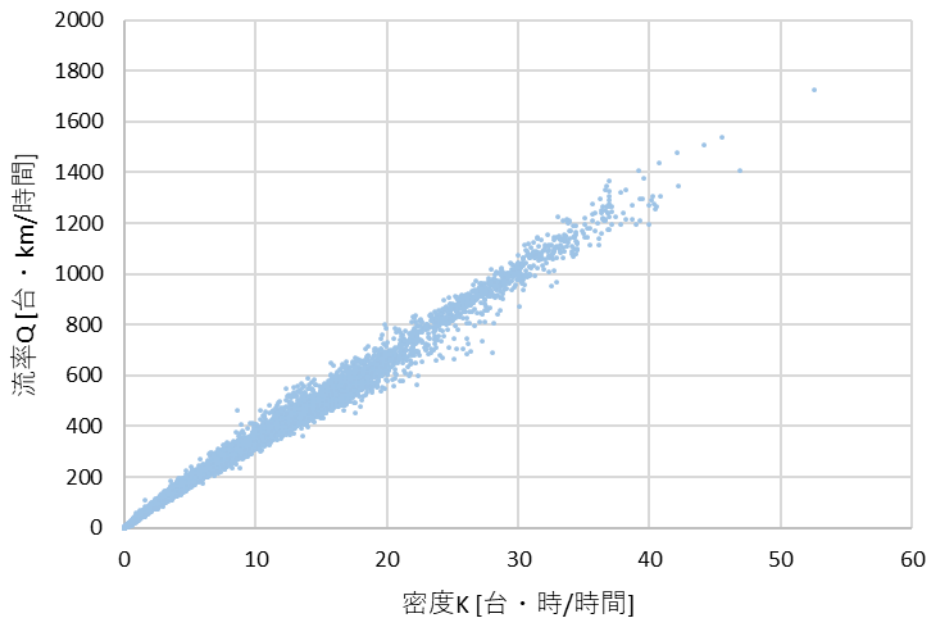


図 13 分析エリア 2 の ETC2.0 プローブ MFD

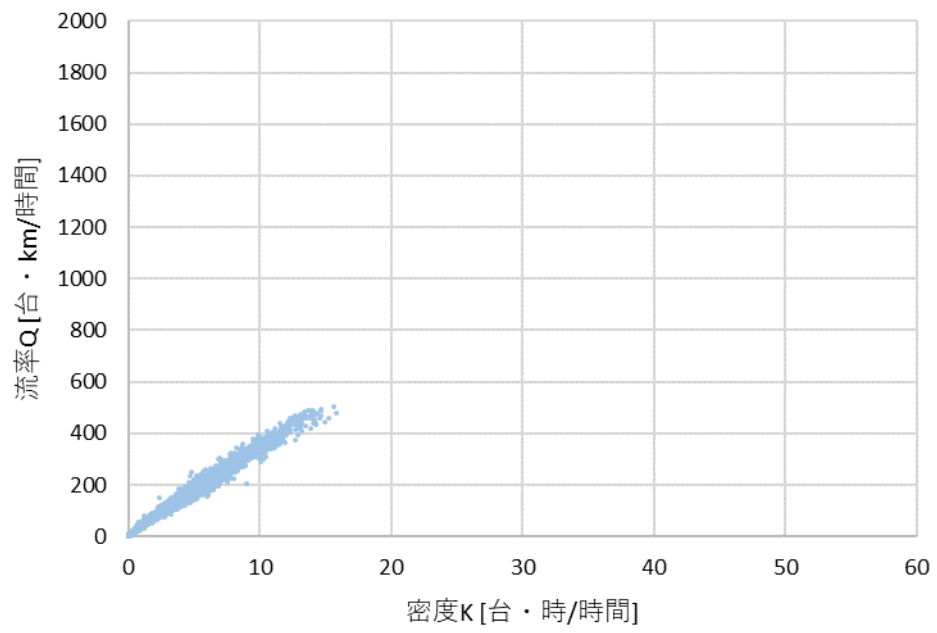


図 14 分析エリア 3 の ETC2.0 プローブ MFD

4-3 バスプローブデータによる MFD の描画

本節ではバスプローブデータから流率と密度を算出し、MFD を描画するまでの手順について述べる。

4-3-1 MFD の算出式

ETC2.0 プローブデータと同様に、奥原ら¹⁰⁾の算出式を用いて1時間あたりに分析エリア内を走行するバス車両の軌跡データを集計し、MFD を描画する。プローブエリア流率 Q_t は式(5)で示すように、時間帯 t におけるバス車両 i の走行距離 D_{it} の総和である。走行距離の単位は km とする。プローブエリア密度 K_t は式(6)で示すように、時間帯 t におけるバス車両 i の走行時間 O_{it} の総和である。走行時間は時間の単位で集計する。また、 a_t は時間帯 t に分析エリア内を走行したバス車両の集合である。

$$Q_t = \sum_{i \in a_t} D_{it} \quad (5)$$

$$K_t = \sum_{i \in a_t} O_{it} \quad (6)$$

Q_t : 時間帯 t におけるバスプローブ流率 [台・km/時間]

K_t : 時間帯 t におけるバスプローブ密度 [台・時/時間]

D_{it} : 時間帯 t におけるバス車両 i の走行距離 [km]

O_{it} : 時間帯 t におけるバス車両 i の走行時間 [時]

a_t : 時間帯 t に分析エリア内を走行したバス車両の集合

4-3-2 分析エリア内を走行する車両の判別

分析エリア内を走行するバス車両を抽出するため、分析エリアの境界線の緯度と経度を取得する。分析エリアは同じであるため、ETC2.0 プローブ車両の抽出で使用した緯度と経度を使用する。

バスプローブデータが分析エリア内に含まれるか、分析エリアのポリゴンを用いて内外判定をする。そこで、分析エリアの境界線の緯度と経度、バスプローブデータの補正位置情報・緯度と補正位置情報・経度を比較した。

4-3-3 流率と密度の算出

分析エリア内に含まれるバスプローブデータを抽出し、バスプローブデータの通過時刻を参考に1時間ごとのデータに分類した(表17)。バスプローブデータの系統名(正式名称)と車番が一致するデータをバス車両1台の1トリップとみなし計算する。

走行時間はバスプローブデータの一番遅い通過時刻から一番早い通過時刻を引くことによって求める。走行距離はバスプローブデータの距離を足し合わせることによって求める。速度は走行距離を走行時間で割ることによって求める。バス車両ごとに求めた速度と走行時間、走行距離を表18に示す。最後に分析エリア内のバス車両の走行時間と走行距離をそれぞれ足し合わせ流率と密度を算出した(表19)。

表17 分析エリア内のバスプローブデータの例

系統内連番	停留所コード	出発時刻(予定)	車両コード(予定)	車載端末コード(予定)	系統番号	系統名(正式名称)	停留所名(正式名称)	停留所名(MC用)	車番	通過時刻
2	3	6:06	4087	356793060139177	G6	棧橋→比島→医大→	棧橋通三丁目	棧橋通三丁目	き402	6:07
3	4	6:07	4087	356793060139177	G6	棧橋→比島→医大→	棧橋通二丁目	棧橋通二丁目	き402	6:08
4	5	6:08	4087	356793060139177	G6	棧橋→比島→医大→	棧橋通一丁目	棧橋通一丁目	き402	6:09
5	6	6:08	4087	356793060139177	G6	棧橋→比島→医大→	土佐高校通	土佐高校通	き402	6:09
6	7	6:09	4087	356793060139177	G6	棧橋→比島→医大→	梅の辻	梅の辻	き402	6:11
2	3	6:30	4088	356793060139185	B1	棧五→上二→鳥越	棧橋通三丁目	棧橋通三丁目	き403	6:31
3	4	6:31	4088	356793060139185	B1	棧五→上二→鳥越	棧橋通二丁目	棧橋通二丁目	き403	6:32
4	5	6:32	4088	356793060139185	B1	棧五→上二→鳥越	棧橋通一丁目	棧橋通一丁目	き403	6:32
5	6	6:32	4088	356793060139185	B1	棧五→上二→鳥越	土佐高校通	土佐高校通	き403	6:33
6	7	6:33	4088	356793060139185	B1	棧五→上二→鳥越	梅の辻	梅の辻	き403	6:34
13	3	6:42	2064	356793060139052	B1	長浜営→はりま→上	棧橋通三丁目	棧橋通三丁目	あ908	6:47
14	4	6:43	2064	356793060139052	B1	長浜営→はりま→上	棧橋通二丁目	棧橋通二丁目	あ908	6:48
15	5	6:44	2064	356793060139052	B1	長浜営→はりま→上	棧橋通一丁目	棧橋通一丁目	あ908	6:48
16	6	6:44	2064	356793060139052	B1	長浜営→はりま→上	土佐高校通	土佐高校通	あ908	6:49
17	7	6:45	2064	356793060139052	B1	長浜営→はりま→上	梅の辻	梅の辻	あ908	6:49
2	3	6:29	2065	356793060139060	P2	棧→十津→種崎	棧橋通三丁目	棧橋通三丁目	あ909	6:31
3	4	6:30	2065	356793060139060	P2	棧→十津→種崎	棧橋通二丁目	棧橋通二丁目	あ909	6:31
4	5	6:31	2065	356793060139060	P2	棧→十津→種崎	棧橋通一丁目	棧橋通一丁目	あ909	6:32
5	6	6:31	2065	356793060139060	P2	棧→十津→種崎	土佐高校通	土佐高校通	あ909	6:32
6	7	6:32	2065	356793060139060	P2	棧→十津→種崎	梅の辻	梅の辻	あ909	6:33
5	6	5:58	2066	356793060138328	K4	棧→バ→潮三	土佐高校通	土佐高校通	あ910	6:00
6	7	5:59	2066	356793060138328	K4	棧→バ→潮三	梅の辻	梅の辻	あ910	6:01
2	3	6:39	2037	356793060138112	P2	棧→十津→種崎	棧橋通三丁目	棧橋通三丁目	あ704	6:40
3	4	6:40	2037	356793060138112	P2	棧→十津→種崎	棧橋通二丁目	棧橋通二丁目	あ704	6:41
4	5	6:41	2037	356793060138112	P2	棧→十津→種崎	棧橋通一丁目	棧橋通一丁目	あ704	6:41
5	6	6:41	2037	356793060138112	P2	棧→十津→種崎	土佐高校通	土佐高校通	あ704	6:42
6	7	6:42	2037	356793060138112	P2	棧→十津→種崎	梅の辻	梅の辻	あ704	6:42
13	3	6:12	2048	356793060138047	C1	長浜営→はりま→県	棧橋通三丁目	棧橋通三丁目	あ716	6:13
14	4	6:13	2048	356793060138047	C1	長浜営→はりま→県	棧橋通二丁目	棧橋通二丁目	あ716	6:14
15	5	6:14	2048	356793060138047	C1	長浜営→はりま→県	棧橋通一丁目	棧橋通一丁目	あ716	6:14

表 18 バス車両ごとの情報の例

速度[km/h]	走行時間[h]	走行距離[km]
7.250	0.400	2.900
17.143	0.117	2.000
34.000	0.050	1.700
36.000	0.033	1.200
20.250	0.067	1.350
51.000	0.033	1.700
30.000	0.017	0.500
25.500	0.067	1.700
8.318	0.367	3.050
24.000	0.050	1.200
60.000	0.017	1.000
33.000	0.033	1.100
25.500	0.067	1.700
6.818	0.367	2.500
21.429	0.117	2.500
3.346	0.867	2.900
24.000	0.050	1.200
20.400	0.083	1.700
30.000	0.083	2.500
7.655	0.483	3.700
24.000	0.050	1.200
18.000	0.067	1.200
36.000	0.017	0.600
24.000	0.050	1.200
24.000	0.050	1.200
34.000	0.050	1.700
6.000	0.333	2.000
72.000	0.017	1.200
20.400	0.083	1.700
18.000	0.067	1.200

表 19 分析エリア内のバスプローブ流率と密度の例

流率[台・km/時間]	密度[台・時/時間]
0.800	0.033
33.750	1.867
94.900	9.467
87.800	9.717
66.850	4.950
57.400	3.250
60.350	5.633
55.850	3.617
45.650	4.033
65.850	6.200
58.000	5.500
61.750	4.333
71.450	7.383
61.600	4.217
71.500	5.783
53.950	4.617
24.150	3.367
1.050	0.050
0.800	0.033
34.100	1.783
96.500	10.700
84.650	9.367
64.000	4.500
61.000	4.400
58.250	5.417
55.900	4.333
47.850	4.317
63.900	6.583
57.650	5.317
63.800	4.283

4-3-4 MFDの描画

バスプローブデータより算出した流率と密度を用いてエリア0からエリア3までそれぞれのMFDを描画した(図15~18). 縦軸に流率[台・km/時間], 横軸に密度[台・時/時間]をとり, 1点は1時間単位の流率と密度の値を示す.

流率の最大値は分析エリア0が100[台・km/時]程度, 分析エリア1が260[台・km/時]程度, 分析エリア2が80[台・km/時]程度, 分析エリア3が70[台・km/時]程度である. また, 密度の最大値は分析エリア0が13[台・時/時間]程度, 分析エリア1が24[台・時/時間]程度, 分析エリア2が11[台・時/時間]程度, 分析エリア3が8[台・時/時間]程度である.

すべての分析エリアに共通して, データが点状に間隔をあけて分布する状態, 傾きが小さくデータが幅広く分布する状態, データの幅が狭く分布する状態がMFDに存在する. ETC2.0プローブデータと同様, サンプル数が少ないがグラフの頂点付近のデータが観測されているように見える.

しかしながら, すべての分析エリアにおいてMFDの傾きは小さい. これにはバス停での乗降時間や時刻表の時間までの待ち時間など, 走行以外の時間が含まれている. そのため, バス車両が進む距離に対して時間がかかることで傾きである速度が低く出ている.

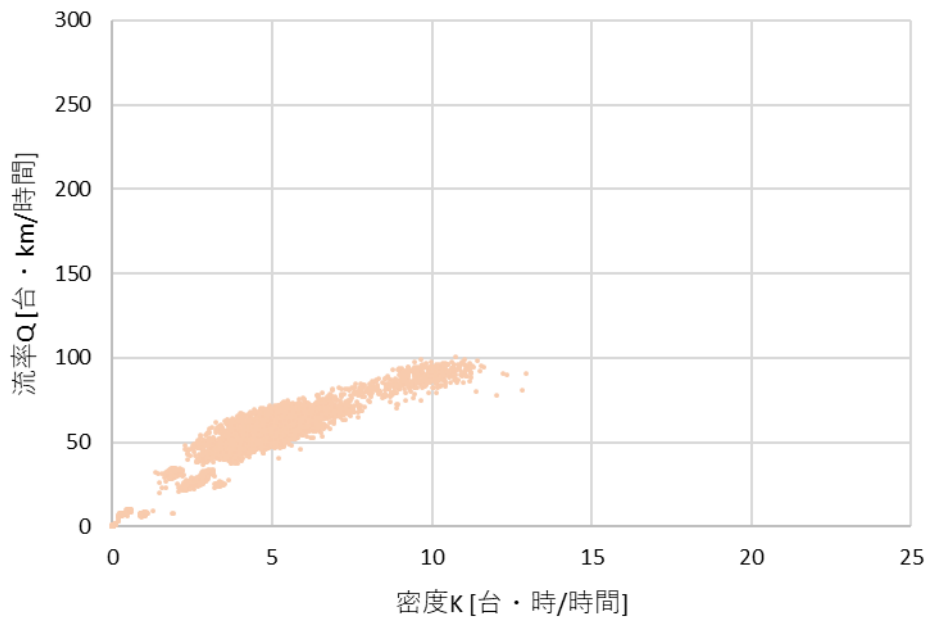


図15 分析エリア0のバスプローブMFD

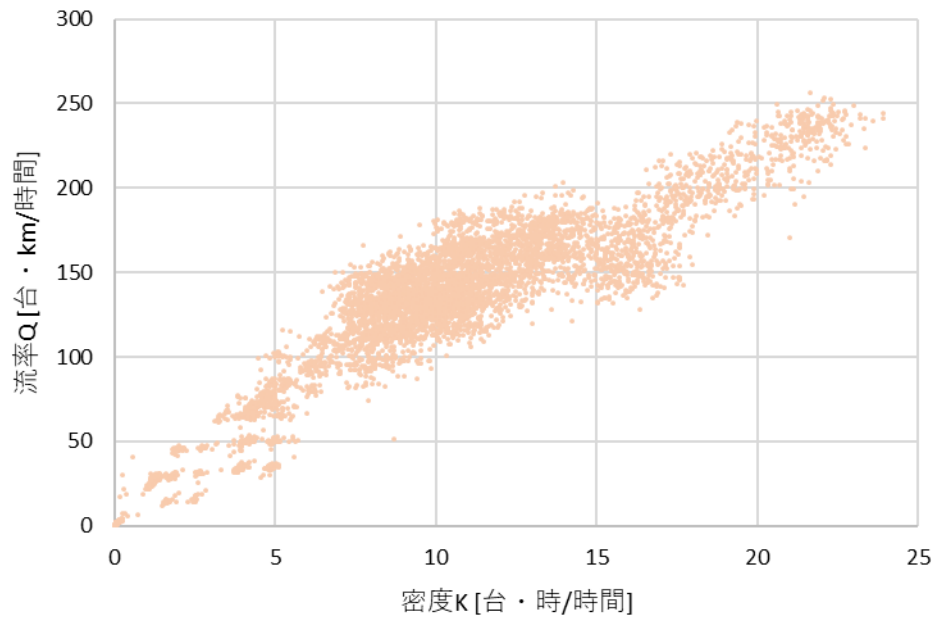


図 16 分析エリア 1 のバスプローブ MFD

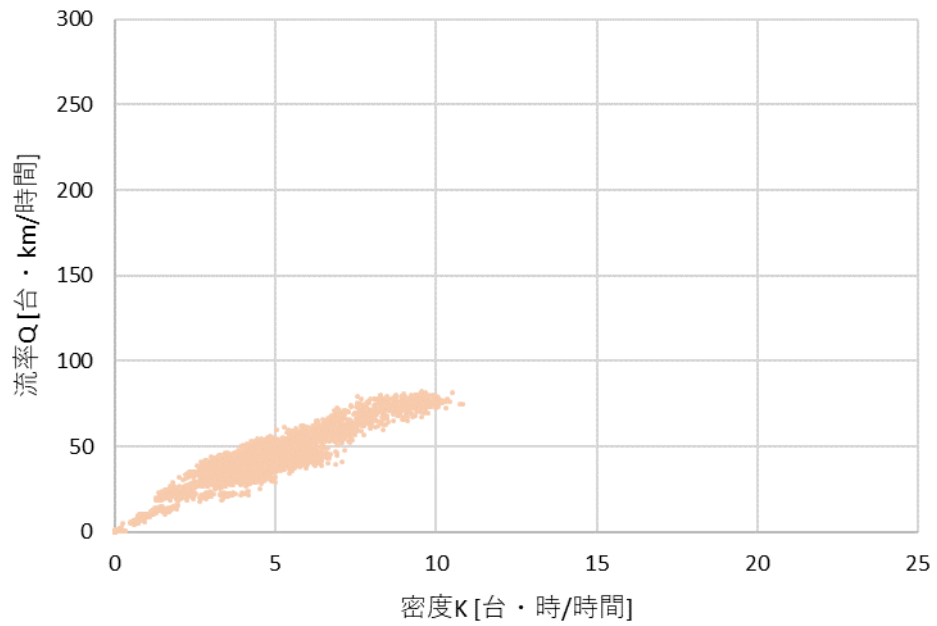


図 17 分析エリア 2 のバスプローブ MFD

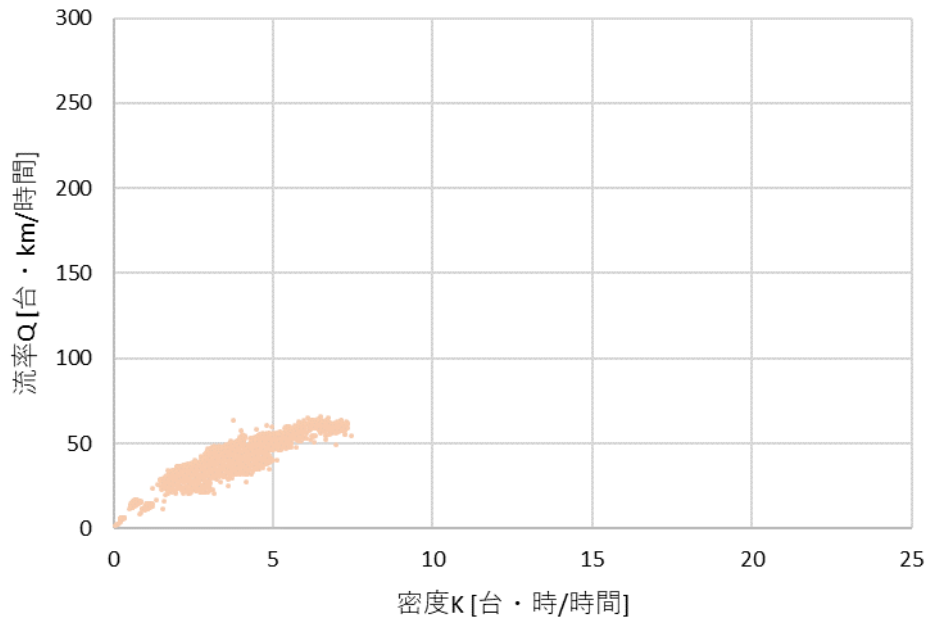


図 18 分析エリア 3 のバスプローブ MFD

5 道路ネットワークの交通流状態の比較

5-1 分析使用データ別の MFD

車両感知器データ・ETC2.0 プローブデータ・バスプローブデータごとにそれぞれ分析エリアの MFD をまとめた(図 19~21)。本研究では、車両感知器で観測されたデータを実際の交通流状態を示す真値とし、ETC2.0 プローブデータやバスプローブデータと比較する。

車両感知器の MFD は分析エリア 0 と分析エリア 1 が約 22000[台・km/時]と同程度の容量であり、分析エリア 0、分析エリア 3 と続く。ETC2.0 プローブの MFD も同様に分析エリア 0 と分析エリア 1 が同程度の容量であり、分析エリア 0、分析エリア 3 と続く。しかしながら、バスプローブの MFD は分析エリア 1 が約 260[台・km/時]と最も容量が大きく、分析エリア 0、分析エリア 2 と続き、分析エリア 3 が最も小さい容量となっている。

バスプローブデータはバス停での観測データであるため、バス車両の運行頻度に依存する。バス車両の運行頻度は交通量から予測される交通需要に合わせて決められていると推察できる。そのため、分析エリア 2 は実際の交通量に対するバス車両の台数が分析エリア 1 に比べて少ないことが分かる。さらに、分析エリア 2 の特徴として JR 鉄道が走行していることが挙げられ、交通需要の分担により分析エリア 1 の MFD の容量との差が生じたと考えられる。

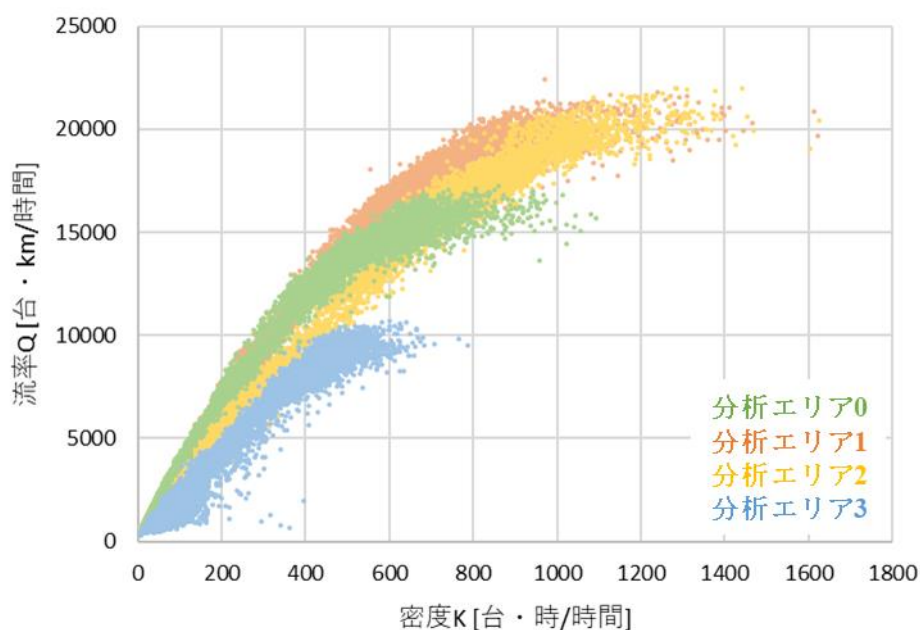


図 19 車両感知器の MFD

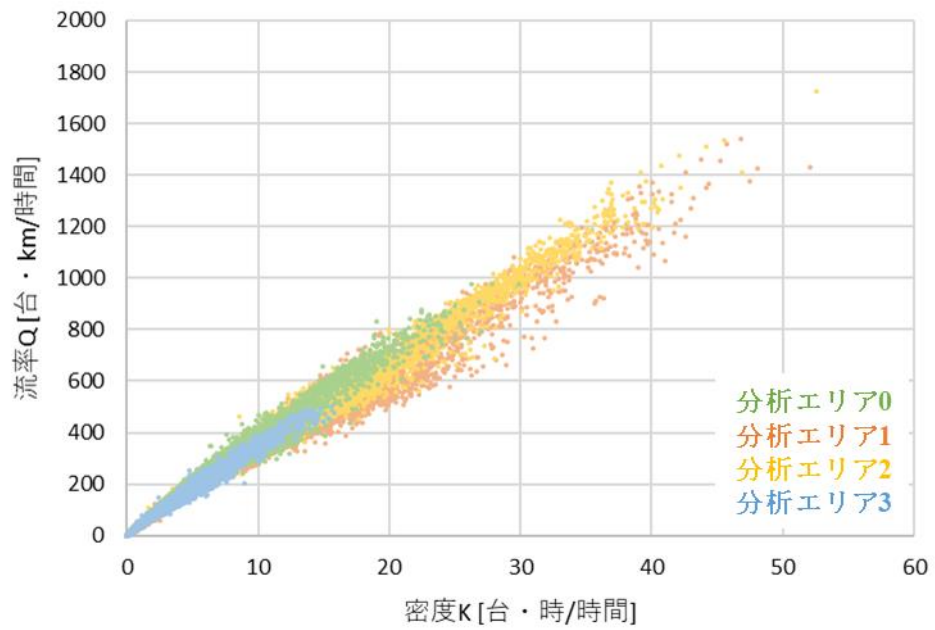


図 20 ETC2.0 プロブの MFD

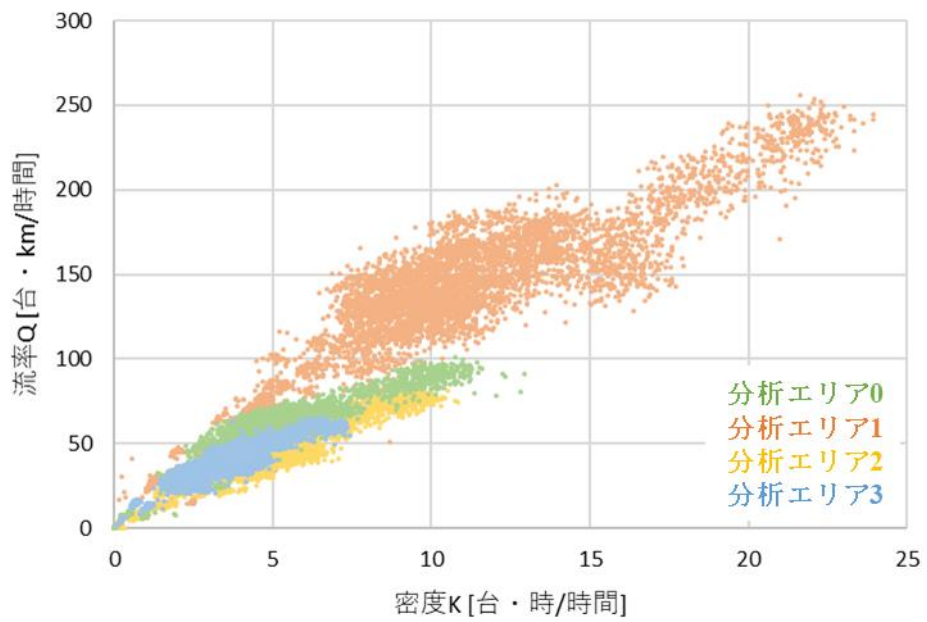


図 21 バスプロブの MFD

5-2 交通流状態の比較指標

ETC2.0 プローブデータとバスプローブデータはサンプルのデータである。そのため、流率と密度については車両感知器の値を大きく下回った。流率は ETC2.0 プローブが車両感知器の 14 分の 1 程度、バスプローブデータが車両感知器の 88 分の 1 程度である。密度は ETC2.0 プローブが車両感知器の 33 分の 1 程度、バスプローブデータが車両感知器の 63 分の 1 程度である。このことから、ETC2.0 プローブデータやバスプローブデータから車両感知器データの流率や密度を把握することは困難である。

そこで本研究では、もう一つの指標である速度に着目した。速度は MFD を構成する点データと原点との傾きにあたる。車両感知器の MFD の速度は分析エリアごとに差が生じている。ETC2.0 プローブの MFD とバスプローブの MFD の分析エリアごと速度に着目すると、ETC2.0 プローブの MFD の速度は分析エリアごと差があまり見られないのに対して、バスプローブの MFD の速度は分析エリアごとに差が見られる。また、車両感知器の MFD とバスプローブの MFD は傾きに変化があり、交通流状態によって速度が変化する。一方で、ETC2.0 プローブの MFD は直線的な分布であるため、交通流状態による速度の変化が少ない可能性がある。そのため、次章以降で分析使用データ別に速度の分析、比較を進める。

6 道路ネットワーク速度の基礎分析

6-1 分析エリア別の速度の時系列変化

速度の時系列変化を分析エリアごとに図 22~25 に示す。MFD の流率を密度で割ることによって 1 時間単位の速度を求めた。期間は 2019 年 10 月 1 日 0 時台から 2019 年 10 月 31 日 23 時台までである。一般的に、交通量調査が行われるのは 9 月から 11 月にかけての秋であり、秋が 1 年の平均的な交通流状態を表現しているため、ここでは 10 月の時系列変化を記載している。

バス車両は深夜から早朝にかけては運行していないため、観測データの時間帯は分析エリアや曜日によって多少異なるが、6 時台から 21 時台は必ず出現する。ETC2.0 プローブ車両についても 1 時台から 4 時台に観測されない日が存在している。

すべての分析エリアに共通して速度の水準は ETC2.0 プローブデータ、車両感知器データ、バスプローブデータの順になっている。すべての分析エリアの平均をとると、ETC2.0 プローブデータの速度の水準は 37.6[km/h]、車両感知器データの速度の水準は 26.6[km/h]、バスプローブデータの速度の水準は 12.1[km/h] である。実際の交通流の速度を示す車両感知器データに比べ、ETC2.0 プローブデータの速度の水準が高いのは、ETC2.0 プローブ車両の台数が少なく、自由流から渋滞流への速度低下を十分に表現できていない可能性があるためである。また、実際の交通流の速度を示す車両感知器データに比べ、バスプローブデータの速度の水準が低いのは、バス停での利用者の乗降時間や時刻表の発車時間までの待機時間などの走行時間以外の要素が含まれるためである。

分析エリア 0 は深夜の自由流の速度について、車両感知器データと ETC2.0 プローブデータの水準が 50~55[km/h] と類似している。また、車両感知器データと ETC2.0 プローブデータの速度の水準が他の分析エリアに比べて近く、6.1[km/h] である。

分析エリア 1 は ETC2.0 プローブデータとバスプローブデータの速度の水準が他の分析エリアに比べて近く、21.4[km/h] である。また、ETC2.0 プローブデータの速度水準が低く、35.3[km/h] である。

分析エリア 2 はデータ同士の速度の折れ線があまり近づくことがなく、車両感知器データは 22.7[km/h]、ETC2.0 プローブデータは 36.2[km/h]、バスプローブデータは 9.3[km/h] とそれぞれの水準を保っている。また、どのデータも日中は速度の振れ幅が小さく、バスプローブデータは他の分析エリアに比べて 2、3[km/h] 小さい。

分析エリア 3 は車両感知器データとバスプローブデータの速度の水準が他の分析エリアに比べて近い位置にあり、その差は 8.9[km/h] である。特に車両感知器データの速度低下が 10[km/h] 近くまで達している。そのため、車両感知器データの速度水準が低く、21.3[km/h] である。

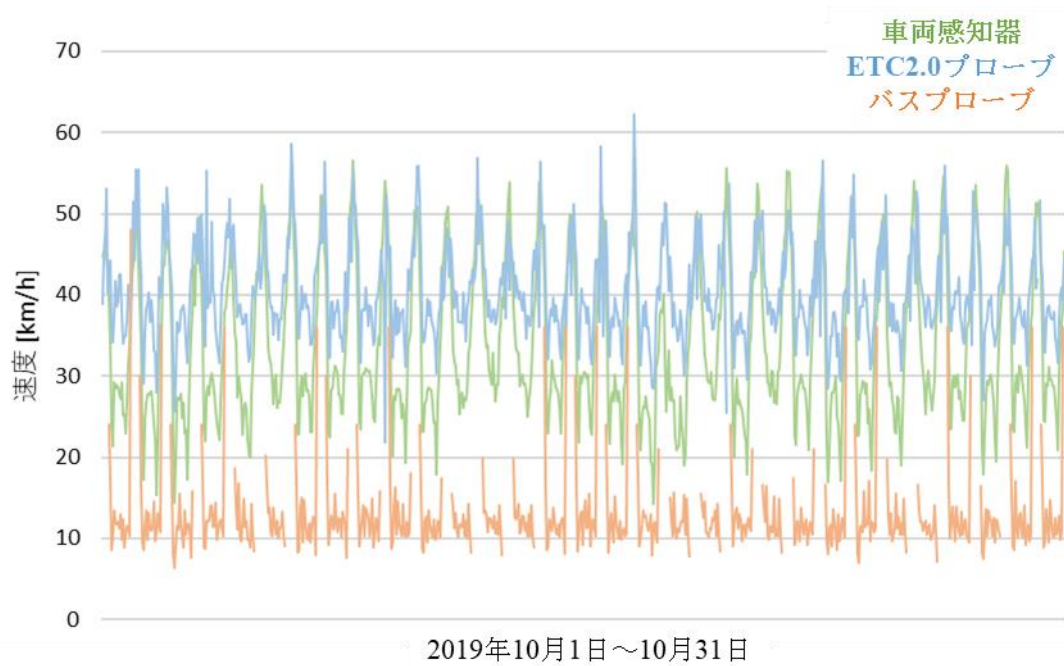


図 22 分析エリア 0 の速度の時系列変化

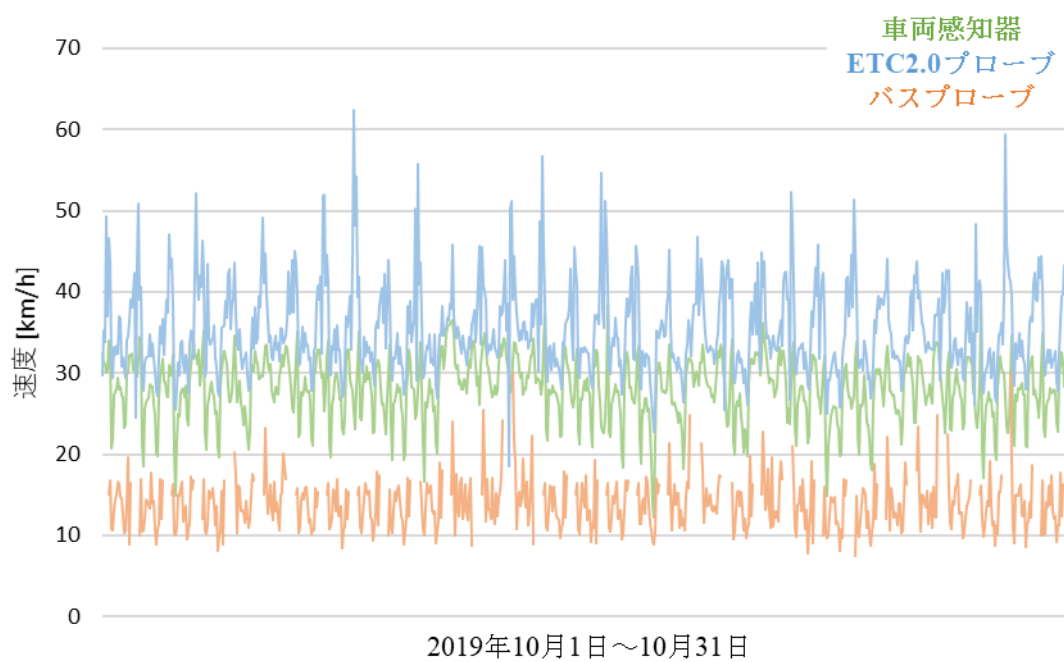


図 23 分析エリア 1 の速度の時系列変化

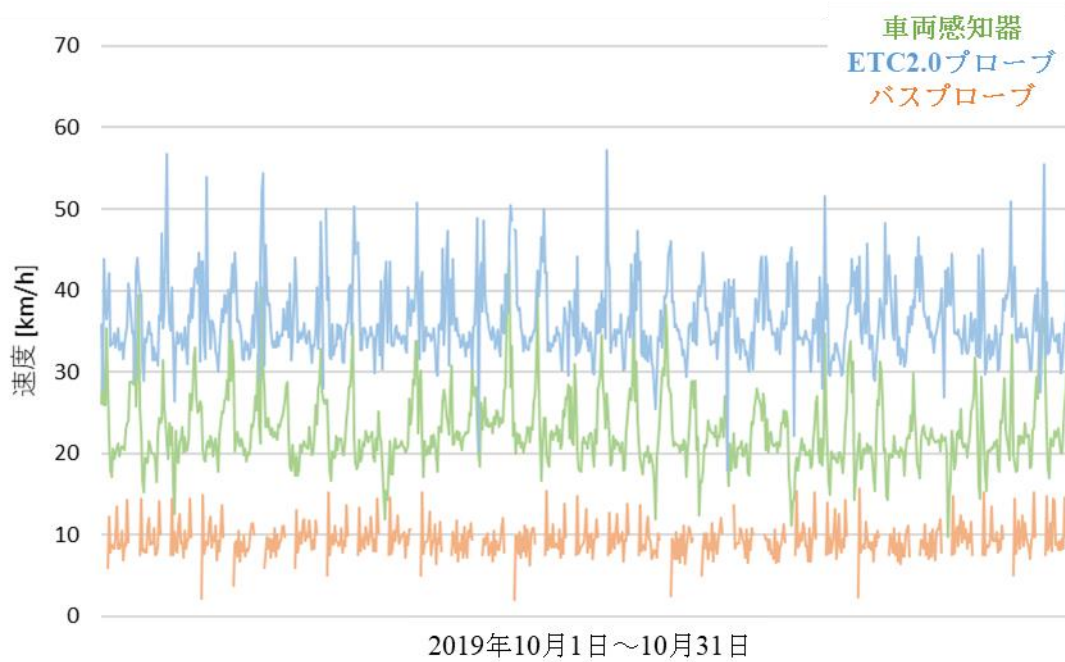


図 24 分析エリア 2 の速度の時系列変化

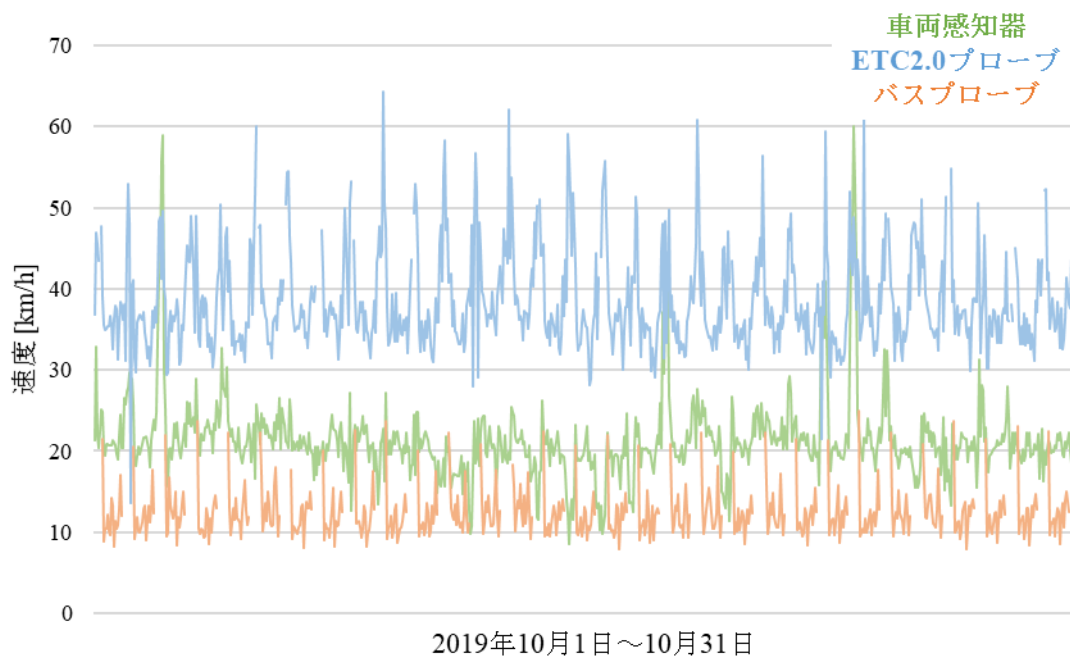


図 25 分析エリア 3 の速度の時系列変化

6-2 分析使用データ別の速度分布

6-2-1 車両感知器の速度分布

分析エリア別の車両感知器の速度分布を図 26～29 に示す。データは 1 年間に観測された 1 時間単位の速度である。

分析エリア 0 は 25～30[km/h]のデータが最も多く、20～25[km/h]、30～50[km/h]はほぼ均等に分布している。図 22 と照らし合わせると、深夜から早朝の交通量が少ない時間帯は 50～55[km/h]の早い速度に分布し、それ以外の時間帯は 25～30[km/h]に多く分布している。

分析エリア 1 は 25～30[km/h]のデータが最も多く、速度の分布が山型であり、20～25[km/h]と 30～35[km/h]が同程度、20[km/h]以下と 35～40[km/h]が同程度となっている。図 23 と照らし合わせると、深夜から早朝の交通量が少ない時間帯に極端に飛び出た速度のデータは見られない。

分析エリア 2 は 25～30[km/h]のデータが最も多く、25～50[km/h]までは階段状に分布している。しかし、20[km/h]以下のデータや 50[km/h]以上のデータがある程度存在する。図 24 と照らし合わせると、20[km/h]前後の速度変化があまりない日中の時間帯と上下に大きく変化する時間帯が存在する。

分析エリア 3 は 20[km/h]以下のデータが最も多く、20～25[km/h]も同程度分布している。この 2 つの階級がデータの大多数にあたる。図 25 と照らし合わせると、車両感知器の速度は他の分析エリアに比べて低速度で、水準は 21.3[km/h]である。

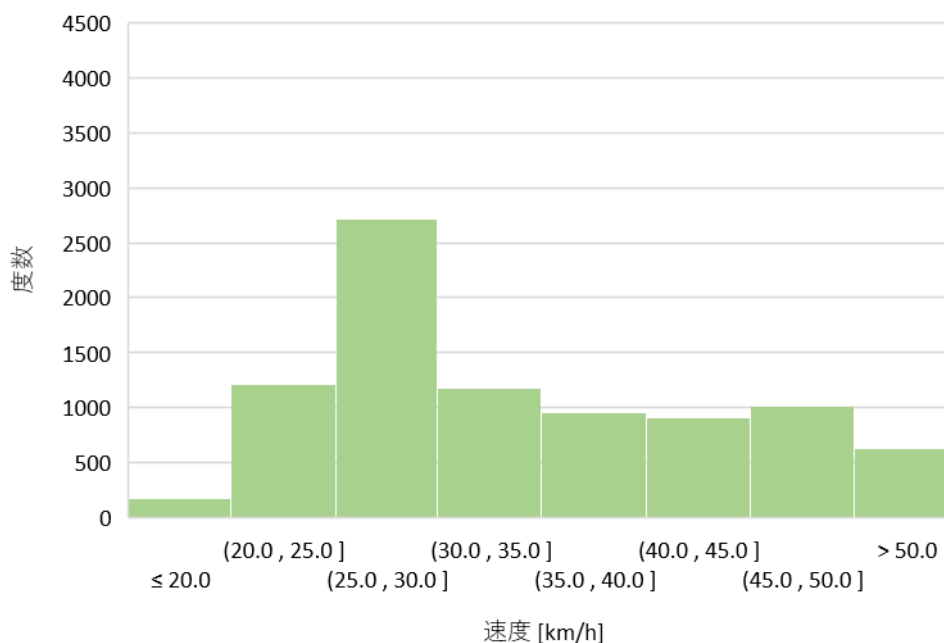


図 26 分析エリア 0 の車両感知器速度分布

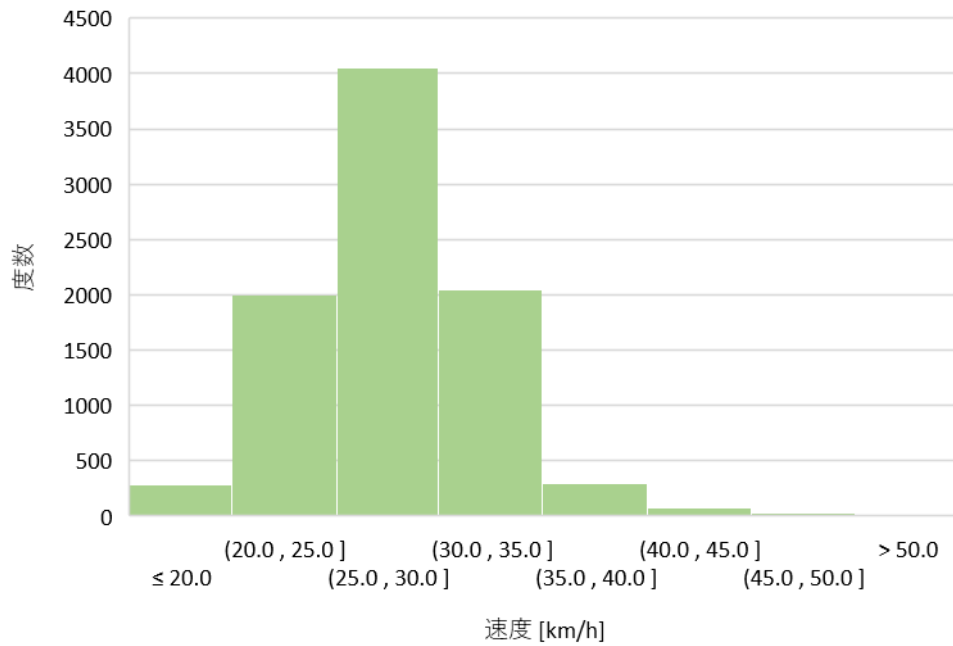


図 27 分析エリア 1 の車両感知器速度分布

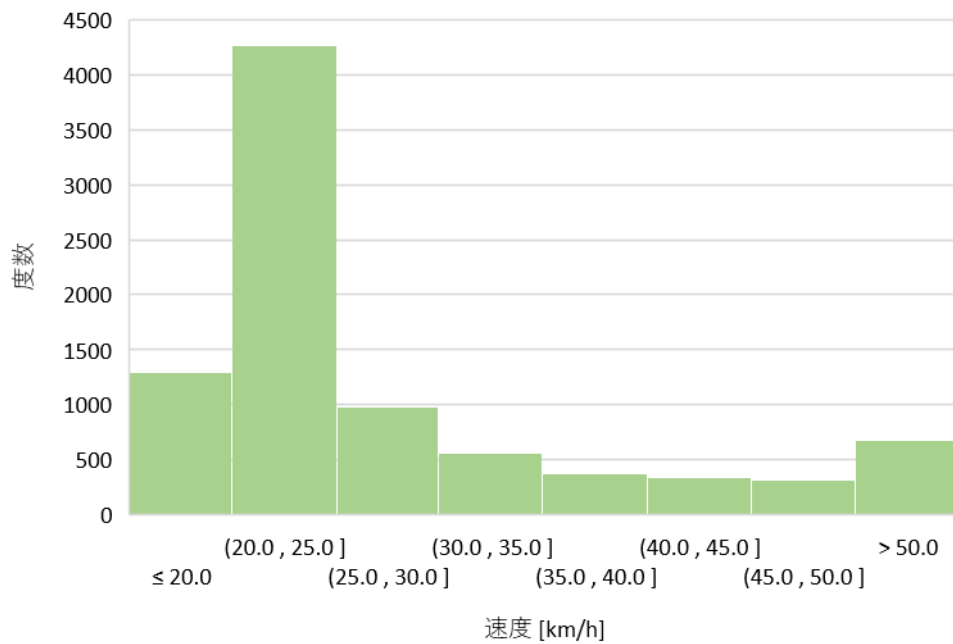


図 28 分析エリア 2 の車両感知器速度分布

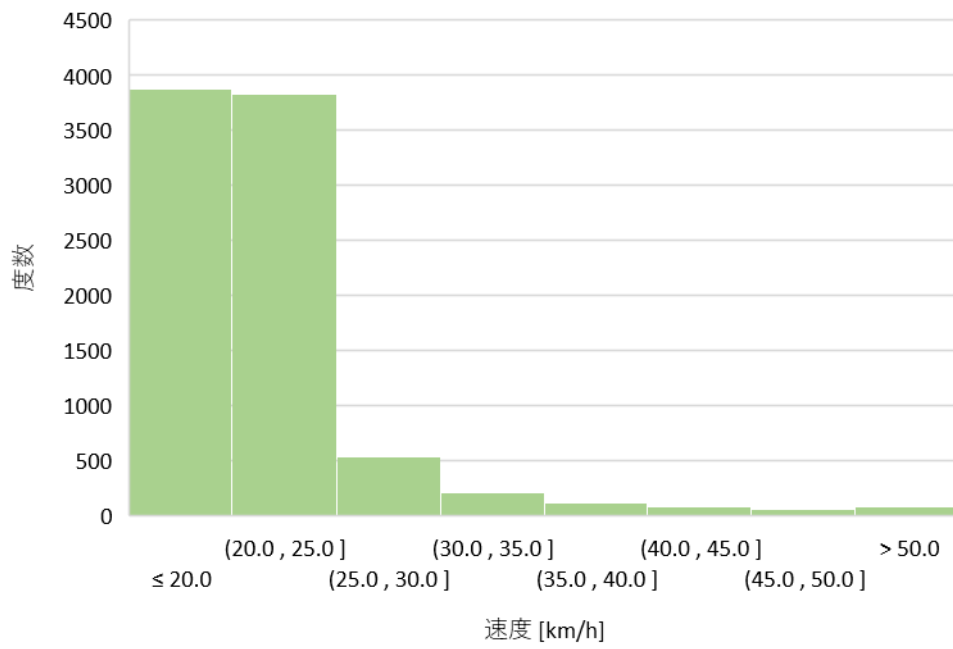


図 29 分析エリア 3 の車両感知器速度分布

6-2-2 ETC2.0 プローブの速度分布

分析エリア別の ETC2.0 プローブの速度分布を図 30～33 に示す。データは 1 年間に観測された 1 時間単位の速度である。車両感知器の速度分布では分析エリアごとに特徴が分かっていたのに対して、ETC2.0 プローブの速度分布ではどの分析エリアも類似した特徴を持つ。

分析エリア 0 は 35～40[km/h]のデータが最も多く、左右非対称の山型である。25[km/h]以下のデータはほぼ見られず、常に速度が高いことが分かる。図 22 と照らし合わせると、ETC2.0 プローブの速度の水準が 40.3[km/h]と他の分析エリアに比べ高く、大きく低下していない。

分析エリア 1 は 30～35[km/h]のデータが最も多く、左右非対称の山型である。分析エリア 0 よりも階級が 1 つ下がって類似した分布である。図 23 と照らし合わせると、50[km/h]以上の速度の出現頻度が低いことが分かる。

分析エリア 2 は 30～35[km/h]のデータが最も多く、25～30[km/h]のデータが少なく絶壁型に近い分布である。図 24 と照らし合わせると、速度の最大値と最小値の幅が 40[km/h]程度と小さくなっている。

分析エリア 3 は 35～40 [km/h]のデータが最も多く、左右非対称の山型である。速度の分布が似ている分析エリア 0 に比べて 30～35[km/h]のデータが多い。図 25 と照らし合わせると、30～40[km/h]の速度で上下に変化している状態が多い。

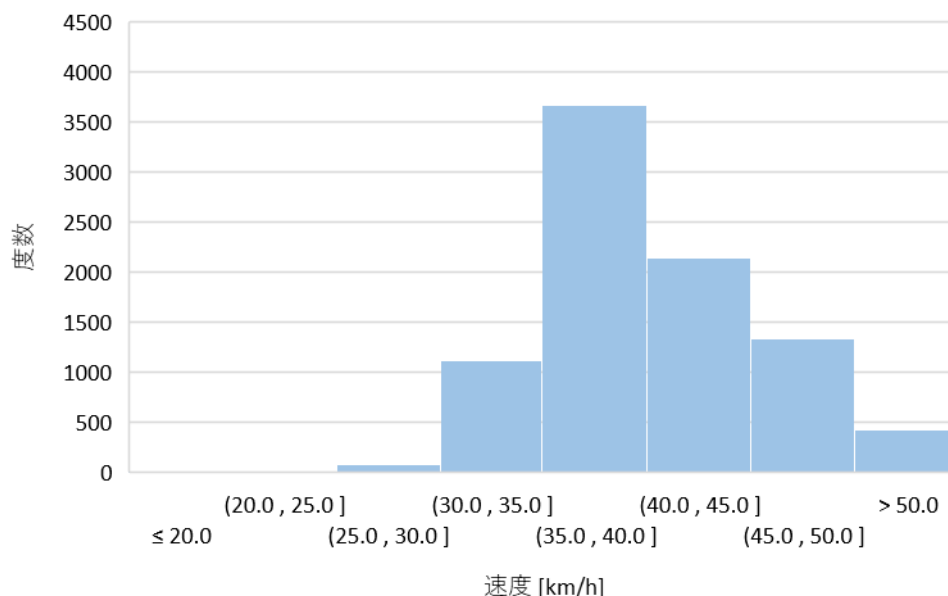


図 30 分析エリア 0 の ETC2.0 プローブ速度分布

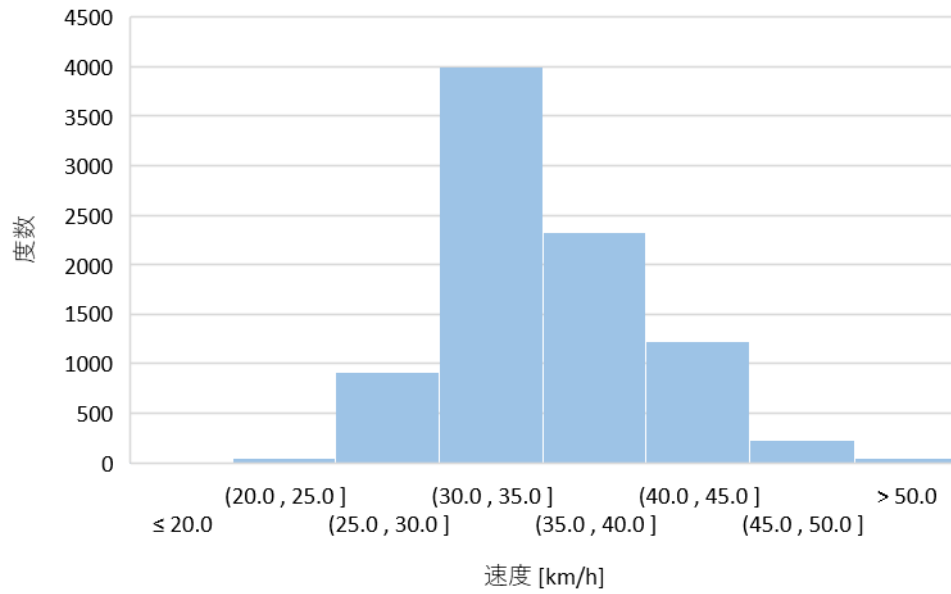


図 31 分析エリア 1 の ETC2.0 プローブ速度分布

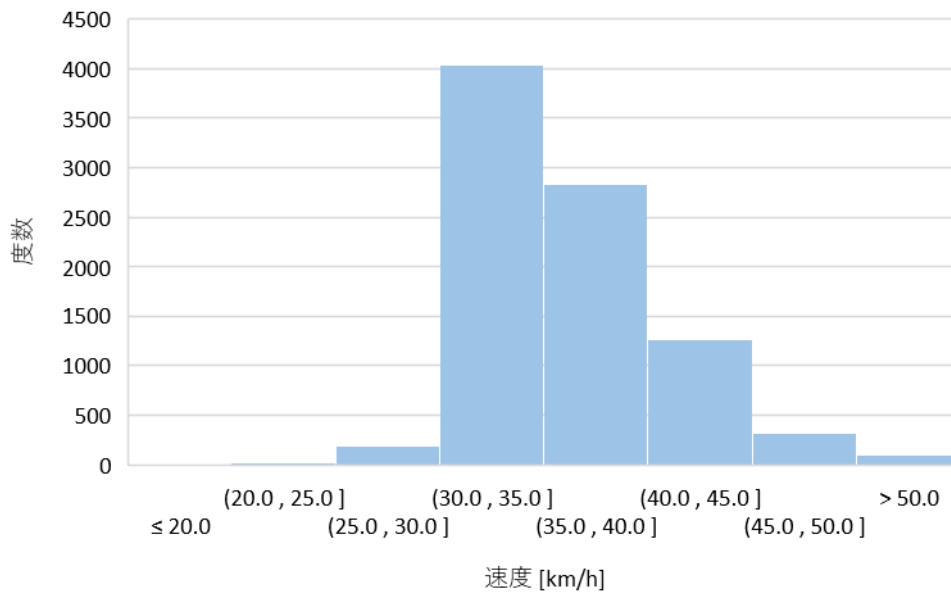


図 32 分析エリア 2 の ETC2.0 プローブ速度分布

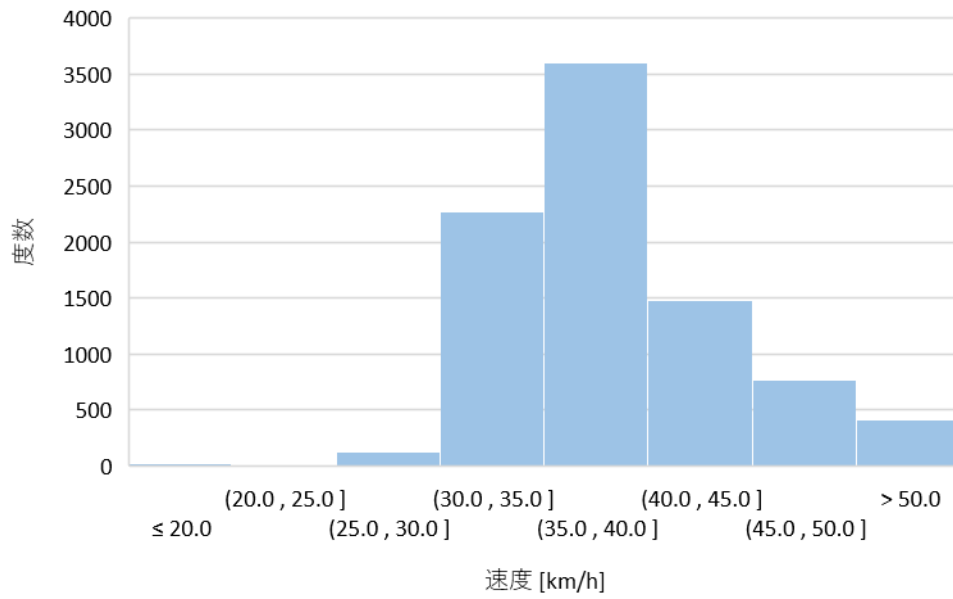


図 33 分析エリア 3 の ETC2.0 プローブ速度分布

6-2-3 バスプローブの速度分布

分析エリア別のバスプローブの速度分布を図 34～37 に示す。データは 1 年間に観測された 1 時間単位の速度である。車両感知器の速度や ETC2.0 プローブの速度に比べて低いことが分布から明らかである。

分析エリア 0 は 10～15 [km/h] のデータが最も多く、他の分析エリアと比べるとそれ以外の速度もある程度分布している。図 22 と照らし合わせると、基本的には低速であるが、30 [km/h] 付近の速度もいくらか観測されている。

分析エリア 1 は 10～15 [km/h] のデータが最も多く、分析エリア 0 の分布に似ているが、次に多いのは 15～20 [km/h] である。図 23 と照らし合わせると、高い位置での速度変化が見られ、速度の水準は 13.9 [km/h] と他の分析エリアに比べて高い。

分析エリア 2 は 10 [km/h] 以下のデータが最も多く、次の 10～15 [km/h] のデータと合わせて大多数を占める。図 24 と照らし合わせると、5 [km/h] 以下のデータがいくつか観測されていることが分かる。

分析エリア 3 は 10～15 [km/h] のデータが最も多く、速度の分布は 30 [km/h] までで低速である。図 25 と照らし合わせると、25 [km/h] 以上の早い速度は観測されておらず、ほかの月においても同様であることが分かる。

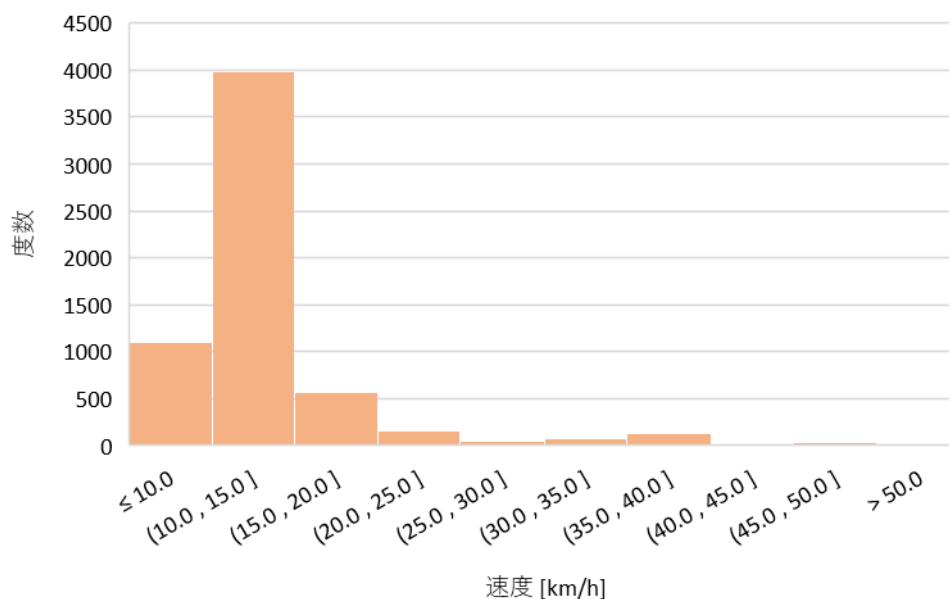


図 34 分析エリア 0 のバスプローブ速度分布

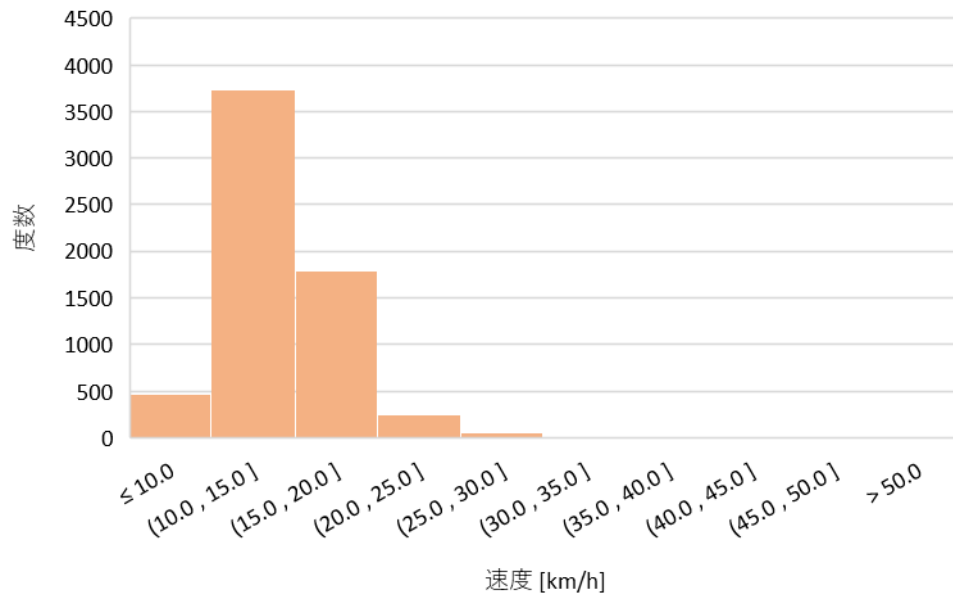


図 35 分析エリア 1 のバスプローブ速度分布

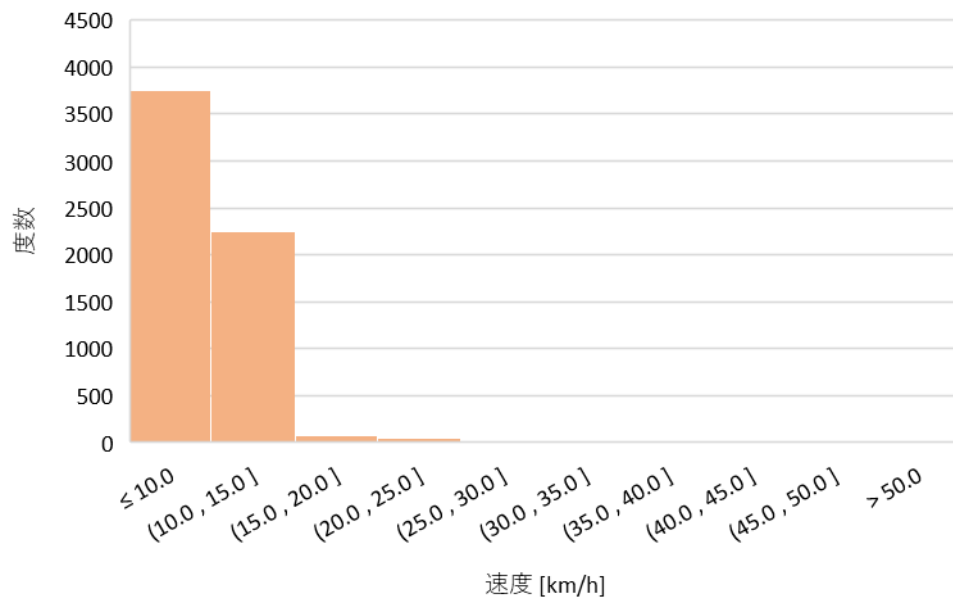


図 36 分析エリア 2 のバスプローブ速度分布

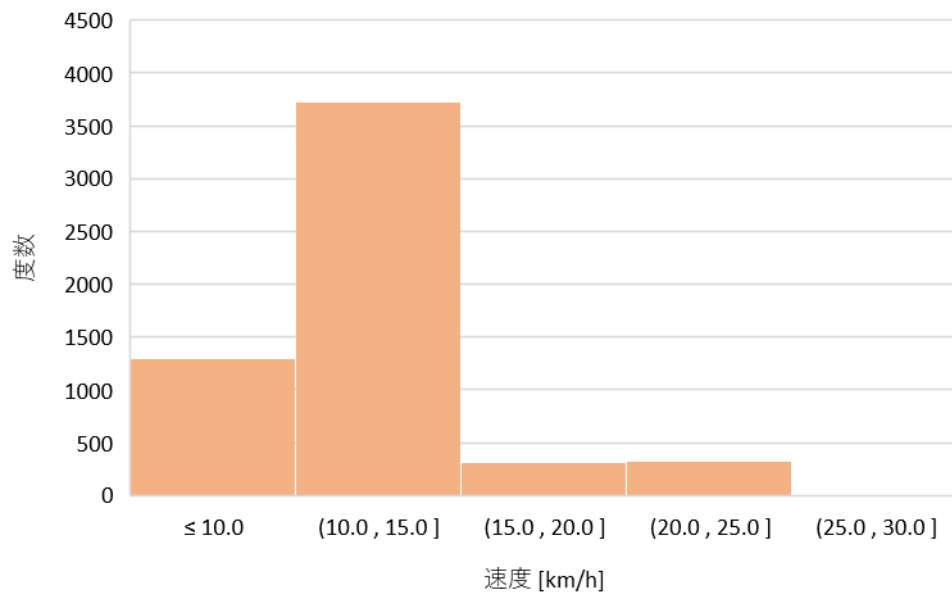


図 37 分析エリア 3 のバスプローブ速度分布

7 折れ線回帰による交通流状態別の速度分析

本章では、前章の速度の基礎的な集計と分析を踏まえて、交通流状態を自由流、混雑流、渋滞流の3つに分け車両感知器の速度がプローブデータにおいても表現されているかを述べる。

7-1 折れ線回帰モデルを用いた MFD の推定

奥原ら¹⁰⁾の手法を用いて式(7)で示すように、車両感知器データ・ETC2.0プローブデータ・バスプローブデータから算出したそれぞれの流率を目的変数とし、密度を説明変数とする折れ線回帰による MFD の推定をした。

$$y = \beta_1 x + \beta_2(x - p_1)d_1 + \beta_3(x - p_2)d_2 \quad (7)$$

y : 流率 [台・km/時]

x : 密度 [台・時/時間]

p_1, p_2 : 折曲点の密度 [台・時/時間]

$\beta_1, \beta_2, \beta_3$: パラメータ

d_1 : ダミー変数 (0 if $x \leq p_1$)

d_2 : ダミー変数 (0 if $x \leq p_2$)

MFD を3つの交通流状態に分け、密度の小さい順に自由流、混雑流、渋滞流と定義する。交通流状態の境界となる2つの折曲点(p_1, p_2)の組み合わせは0から密度の最大値までを1刻みに設定する。全ての組み合わせの中から自由流の R^2 値が高いもの、混雑流の R^2 値が高いもの、渋滞流の R^2 値が高いものを比較する。高知市において、実際の交通流状態は渋滞流である場合が数少ないと考えられるため、渋滞流の領域が少ないものが最適であるとし、適切な折れ線回帰モデル式を決定する。

7-2 折れ線回帰モデル式

7-2-1 車両感知器の折れ線回帰モデル式

車両感知器の折れ線回帰モデル式を式(8)~(19)に、モデル式と MFD を重ね合わせた図を図 38~49 に示す。自由流の R^2 値が高い場合と混雑流の R^2 値が高い場合の渋滞流の領域の始まりである折曲点 p_2 の差は最大でも 14[台・時/時間]とほとんど変わらないが、渋滞流の R^2 値が高い場合は渋滞流の領域が全体の 99%と多く、MFD がほぼ直線的な式で表現されている。また、分析エリア 0 は渋滞流の直線が MFD の形状から少し離れているのに対して、分析エリア 3 の渋滞流の領域は約 300[台・時/時間]で直線が短く、傾きが 2.8 で横ばいの状態が表現されている。

$$y = 33.6x + 20.8(x - 226)d_1 + 13.0(x - 233)d_2 \quad (8)$$

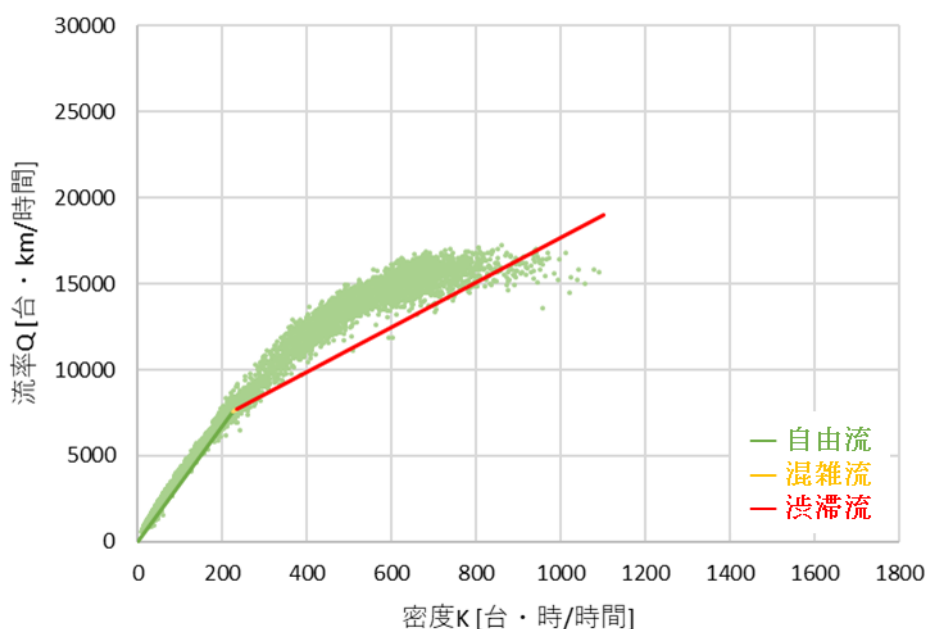


図 38 分析エリア 0 の車両感知器折れ線回帰モデル(自由流の R^2)

$$y = 46.4x + 33.6(x - 12)d_1 + 13.5(x - 226)d_2 \quad (9)$$

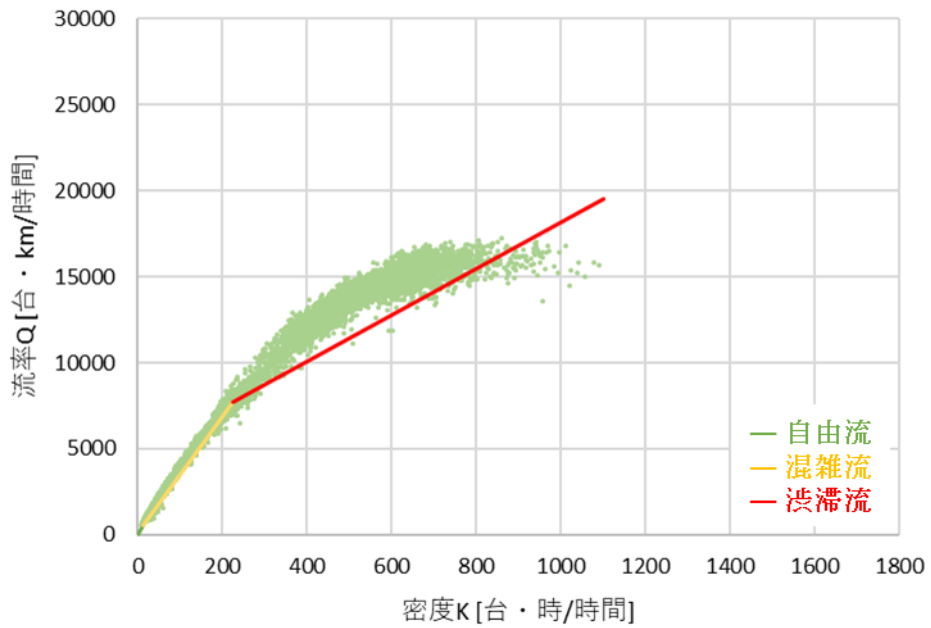


図 39 分析エリア 0 の車両感知器折れ線回帰モデル(混雑流の R^2)

$$y = 46.4x + 45.6(x - 12)d_1 + 21.9(x - 14)d_2 \quad (10)$$

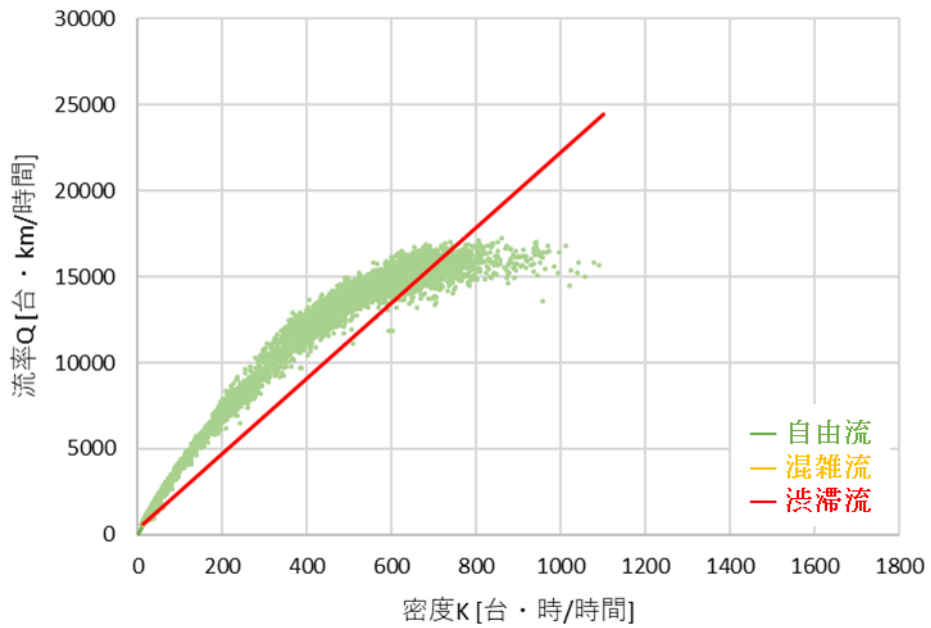


図 40 分析エリア 0 の車両感知器折れ線回帰モデル(渋滞流の R^2)

$$y = 27.4x + 18.1(x - 658)d_1 + 5.9(x - 665)d_2 \quad (11)$$

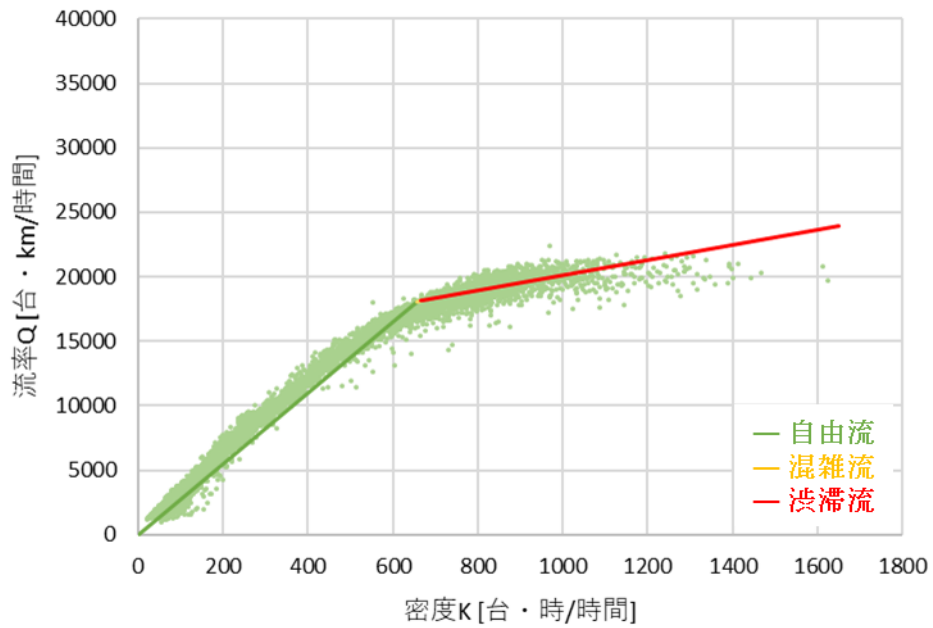


図 41 分析エリア 1 の車両感知器折れ線回帰モデル(自由流の R^2)

$$y = 33.4x + 27.4(x - 24)d_1 + 6.0(x - 658)d_2 \quad (12)$$

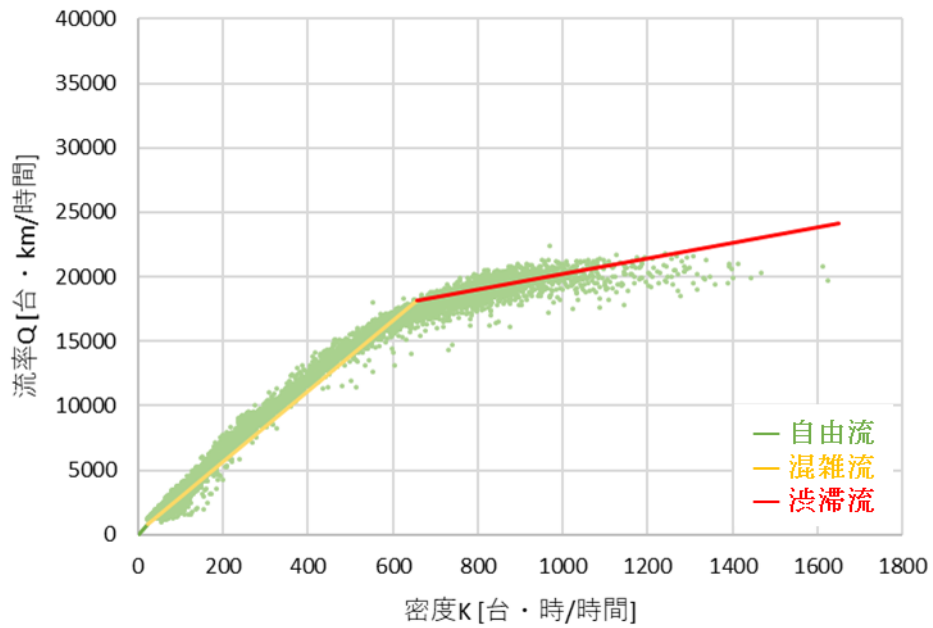


図 42 分析エリア 1 の車両感知器折れ線回帰モデル(混雑流の R^2)

$$y = 33.4x + 31.6(x - 24)d_1 + 21.8(x - 28)d_2 \quad (13)$$

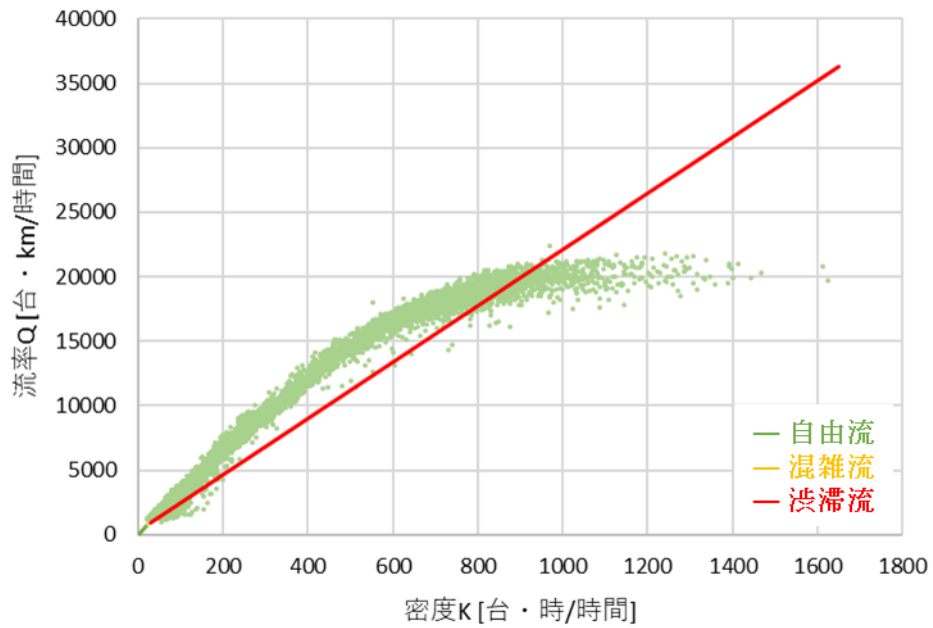


図 43 分析エリア 1 の車両感知器折れ線回帰モデル(渋滞流の R^2)

$$y = 21.5x + 19.9(x - 769)d_1 + 8.7(x - 783)d_2 \quad (14)$$

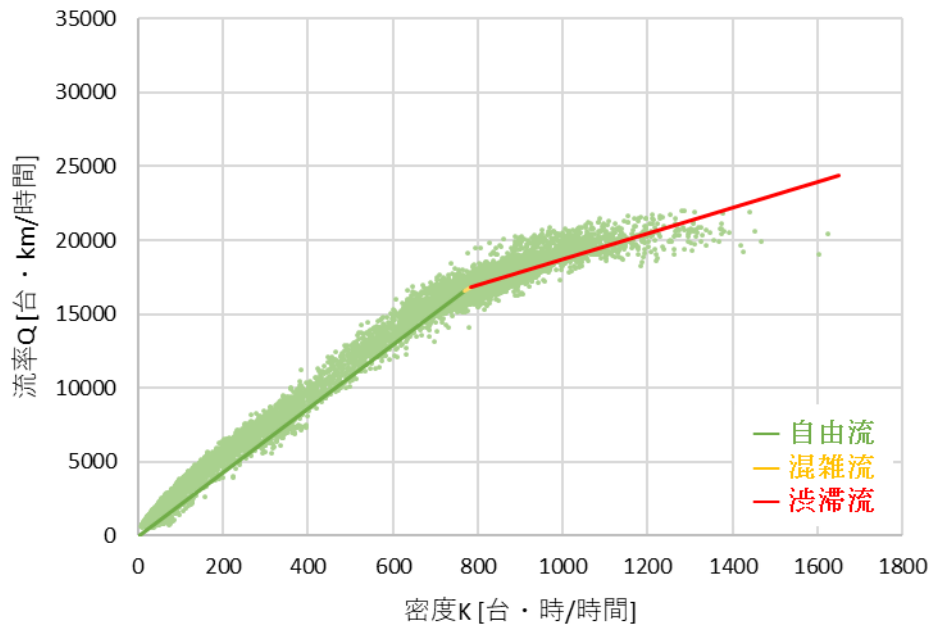


図 44 分析エリア 2 の車両感知器折れ線回帰モデル(自由流の R^2)

$$y = 38.9x + 21.5(x - 8)d_1 + 9.0(x - 769)d_2 \quad (15)$$

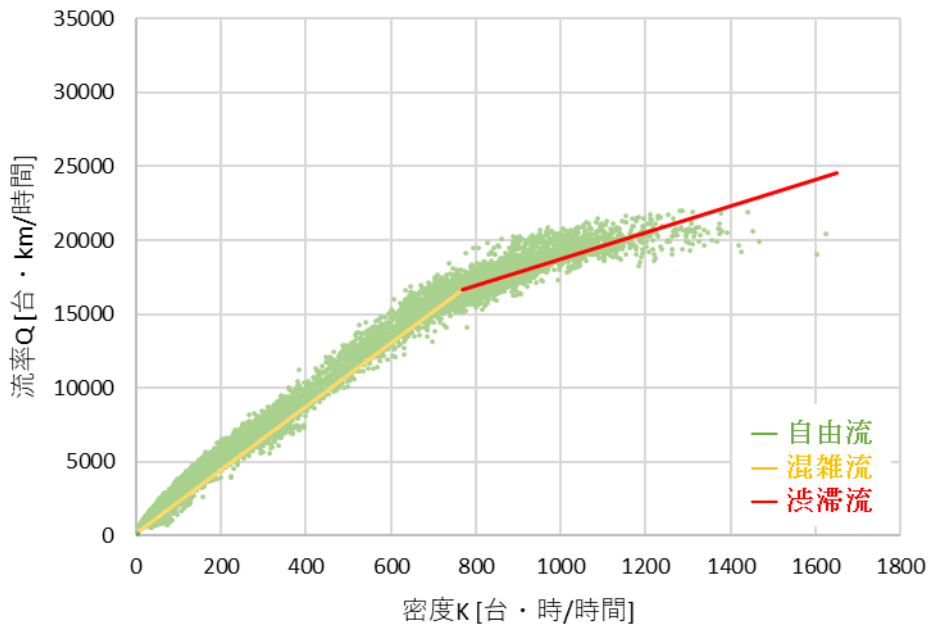


図 45 分析エリア 2 の車両感知器折れ線回帰モデル(混雑流の R^2)

$$y = 38.9x + 38.2(x - 8)d_1 + 18.9(x - 9)d_2 \quad (16)$$

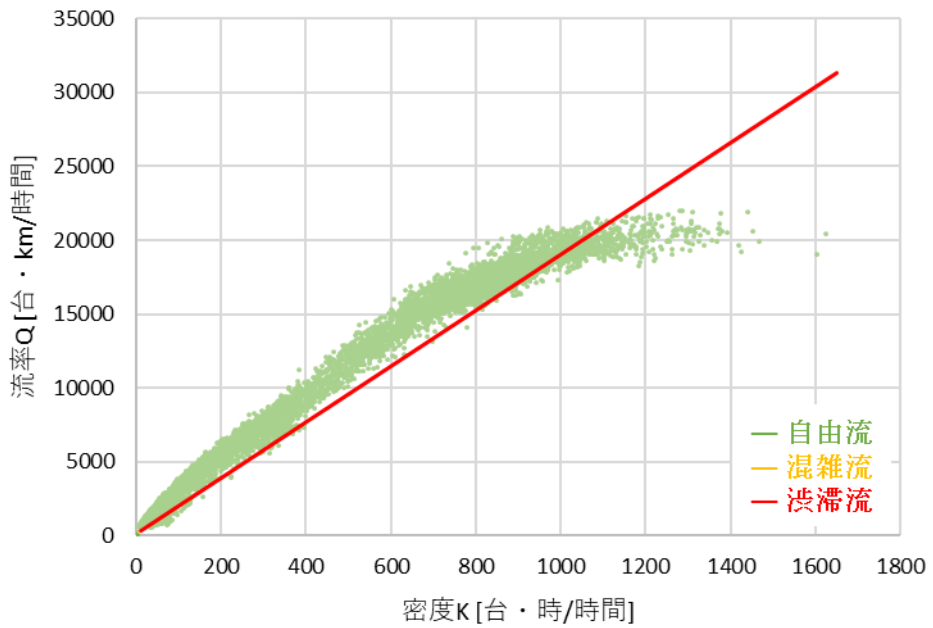


図 46 分析エリア 2 の車両感知器折れ線回帰モデル(渋滞流の R^2)

$$y = 19.8x + 7.1(x - 480)d_1 + 2.6(x - 485)d_2 \quad (18)$$

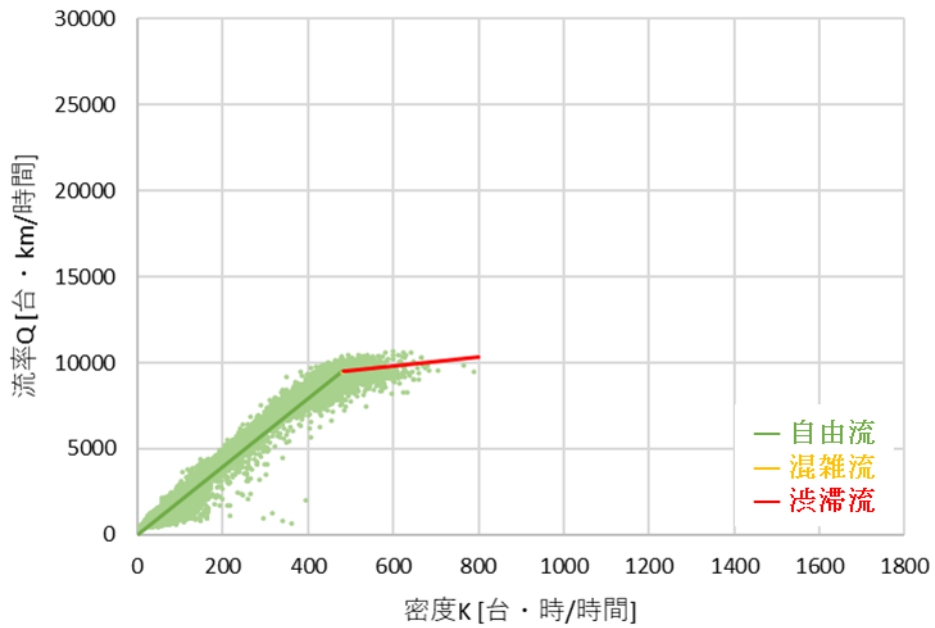


図 47 分析エリア 3 の車両感知器折れ線回帰モデル(自由流の R^2)

$$y = 41.8x + 19.8(x - 6)d_1 + 2.8(x - 480)d_2 \quad (18)$$

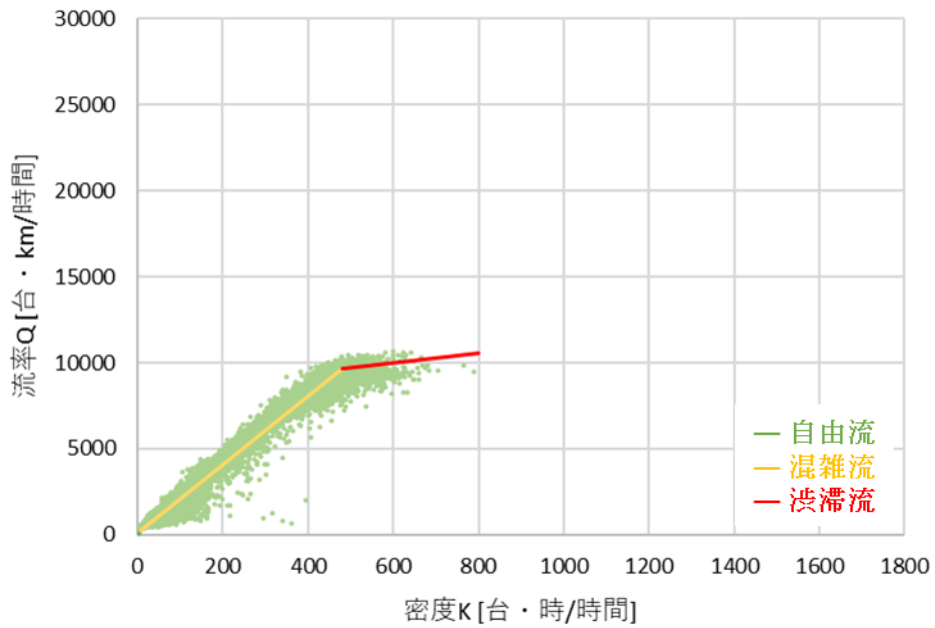


図 48 分析エリア 3 の車両感知器折れ線回帰モデル(混雑流の R^2)

$$y = 41.8x + 38.7(x - 6)d_1 + 18.8(x - 8)d_2 \quad (19)$$

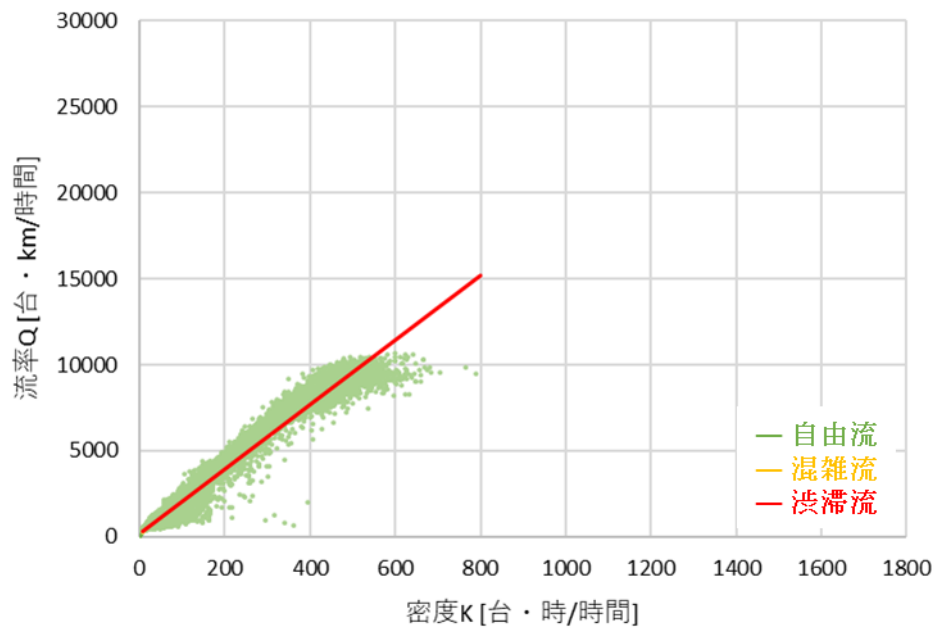


図 49 分析エリア 3 の車両感知器折れ線回帰モデル(渋滞流の R^2)

7-2-2 ETC2.0 プローブの折れ線回帰モデル式

ETC2.0 プローブの折れ線回帰モデル式を式(20)～(31)に、モデル式と MFD を重ね合わせた図を図 50～61 に示す。分析エリア 0 は混雑流の R^2 値が高い場合、折曲点 p_2 が 25[台・時/時間]となり渋滞流の領域が少なくなっている。分析エリア 1 は混雑流の R^2 値が高い場合、折曲点 p_2 が 47[台・時/時間]となり渋滞流の領域が少なく、傾きが 7.6 と横ばいの状態が表現されている。分析エリア 2 は自由流の R^2 値が高い場合と混雑流の R^2 値が高い場合に折曲点 p_2 が 43[台・時/時間]となり渋滞流の領域が少なくなっている。分析エリア 3 は自由流の R^2 値が高い場合と混雑流の R^2 値が高い場合に折曲点 p_2 が 14[台・時/時間]となり渋滞流の領域が少なくなっており、どちらも傾きが 11.8 と横ばいの状態が表現できている。

$$y = 41.8x + 33.0(x - 4)d_1 + 31.2(x - 14)d_2 \quad (20)$$

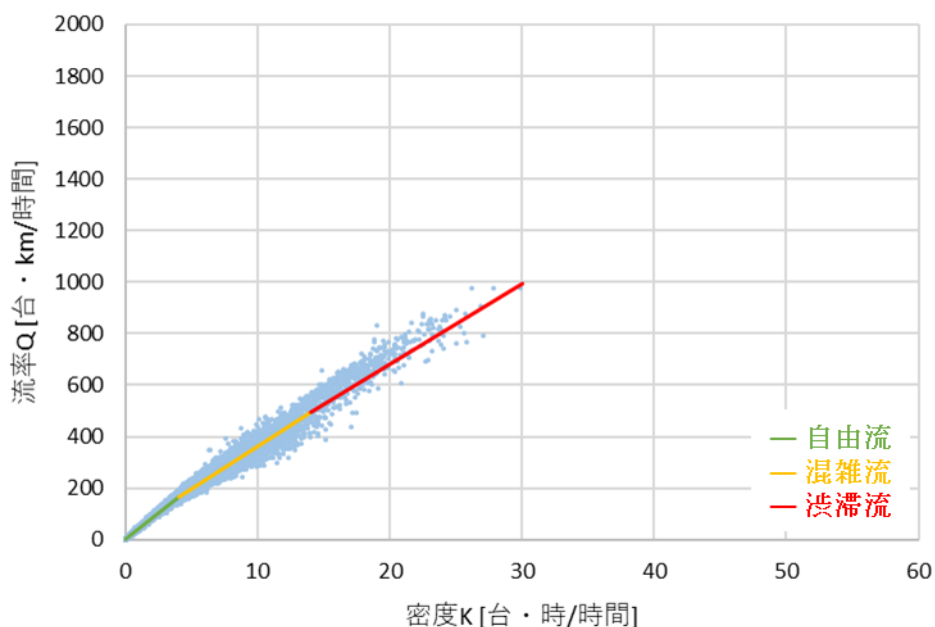


図 50 分析エリア 0 の ETC2.0 プローブ折れ線回帰モデル(自由流の R^2)

$$y = 45.5x + 34.6(x - 1)d_1 + 34.4(x - 25)d_2 \quad (21)$$

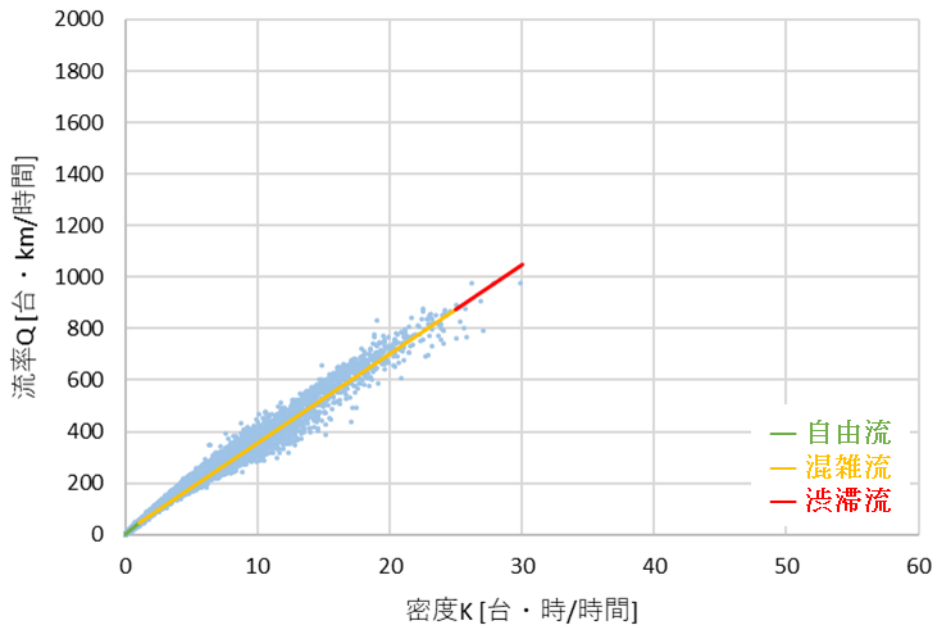


図 51 分析エリア 0 の ETC2.0 プローブ折れ線回帰モデル(混雑流の R^2)

$$y = 45.5x + 43.3(x - 1)d_1 + 34.2(x - 2)d_2 \quad (22)$$

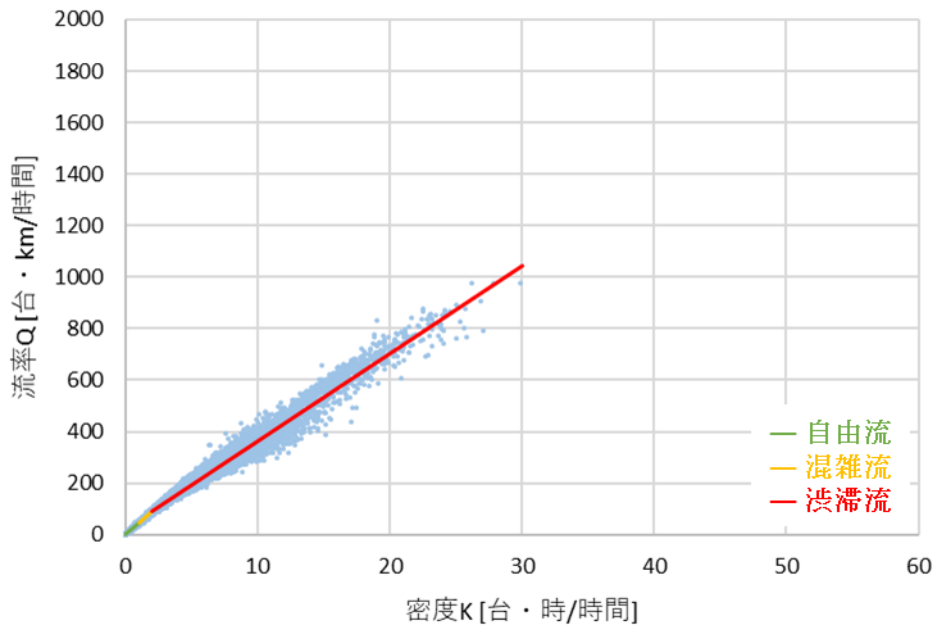


図 52 分析エリア 0 の ETC2.0 プローブ折れ線回帰モデル(渋滞流の R^2)

$$y = 37.4x + 31.3(x - 7)d_1 + 29.9(x - 8)d_2 \quad (23)$$

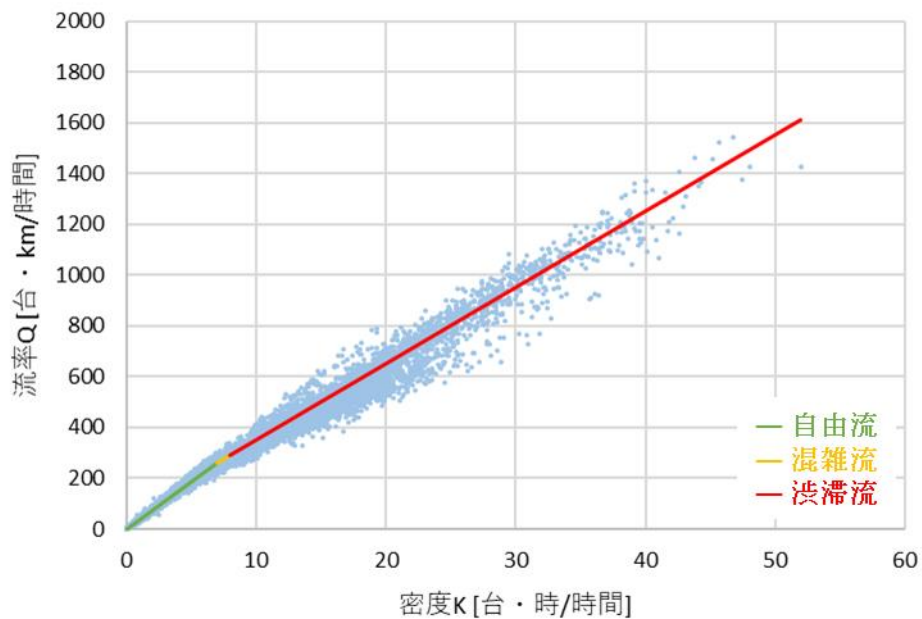


図 53 分析エリア 1 の ETC2.0 プローブ折れ線回帰モデル(自由流の R^2)

$$y = 39.4x + 30.2(x - 1)d_1 + 7.6(x - 47)d_2 \quad (24)$$

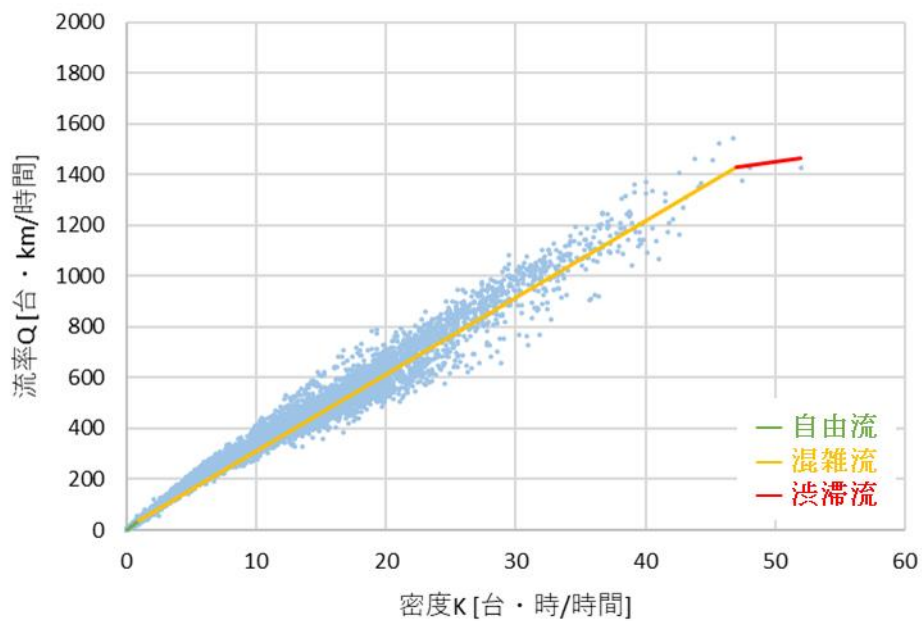


図 54 分析エリア 1 の ETC2.0 プローブ折れ線回帰モデル(混雑流の R^2)

$$y = 39.4x + 35.9(x - 1)d_1 + 29.8(x - 2)d_2 \quad (25)$$

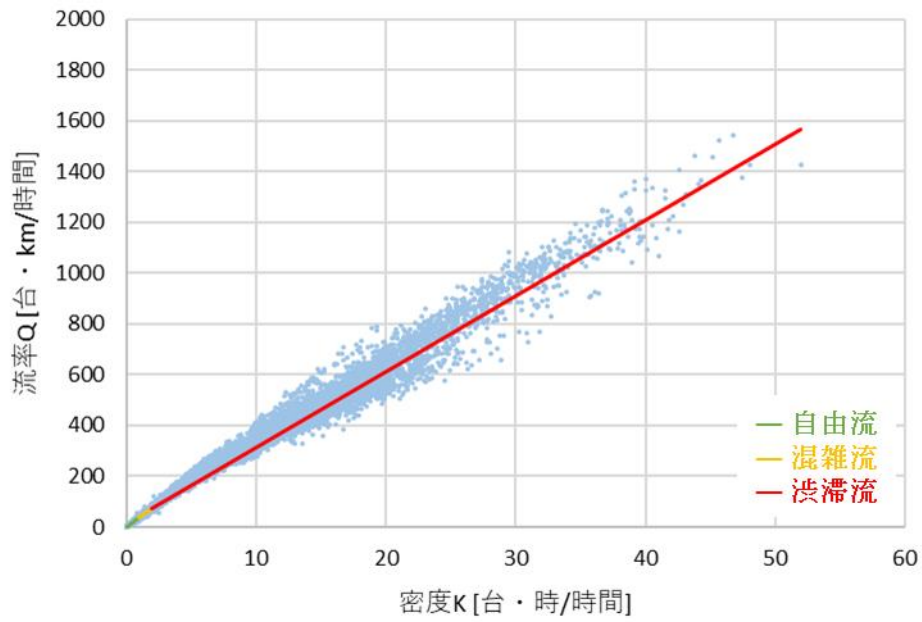


図 55 分析エリア 1 の ETC2.0 プローブ折れ線回帰モデル(渋滞流の R^2)

$$y = 33.0x + 28.2(x - 37)d_1 + 27.8(x - 43)d_2 \quad (26)$$

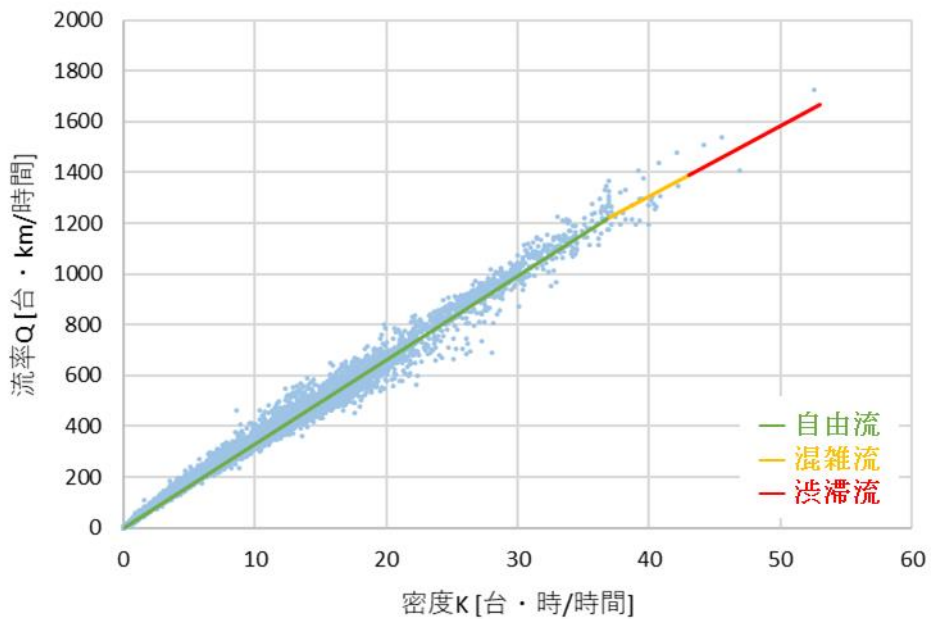


図 56 分析エリア 2 の ETC2.0 プローブ折れ線回帰モデル(自由流の R^2)

$$y = 39.1x + 32.7(x - 1)d_1 + 27.8(x - 43)d_2 \quad (29)$$

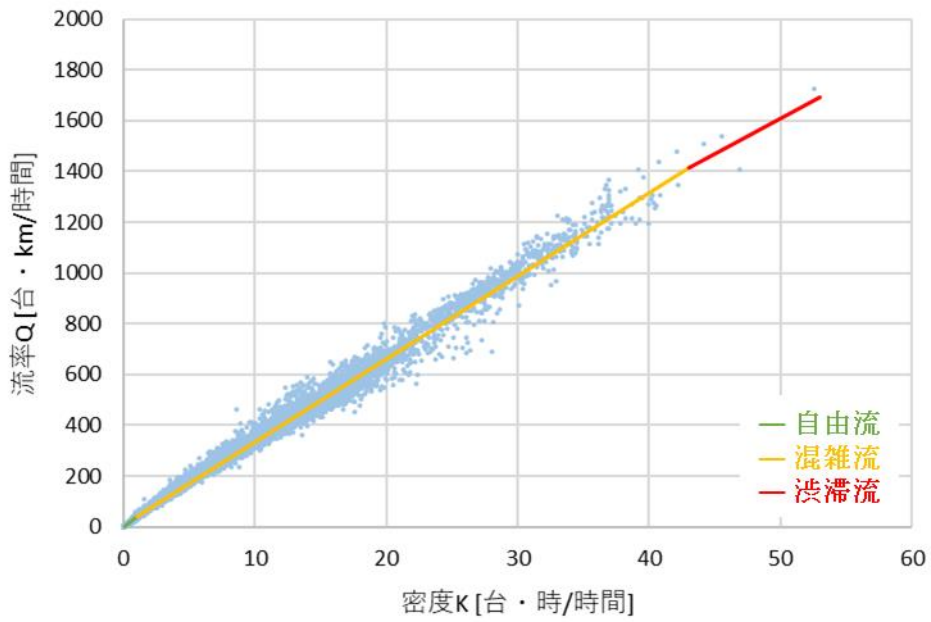


図 57 分析エリア 2 の ETC2.0 プローブ折れ線回帰モデル(混雑流の R^2)

$$y = 39.1x + 38.6(x - 1)d_1 + 32.6(x - 2)d_2 \quad (28)$$

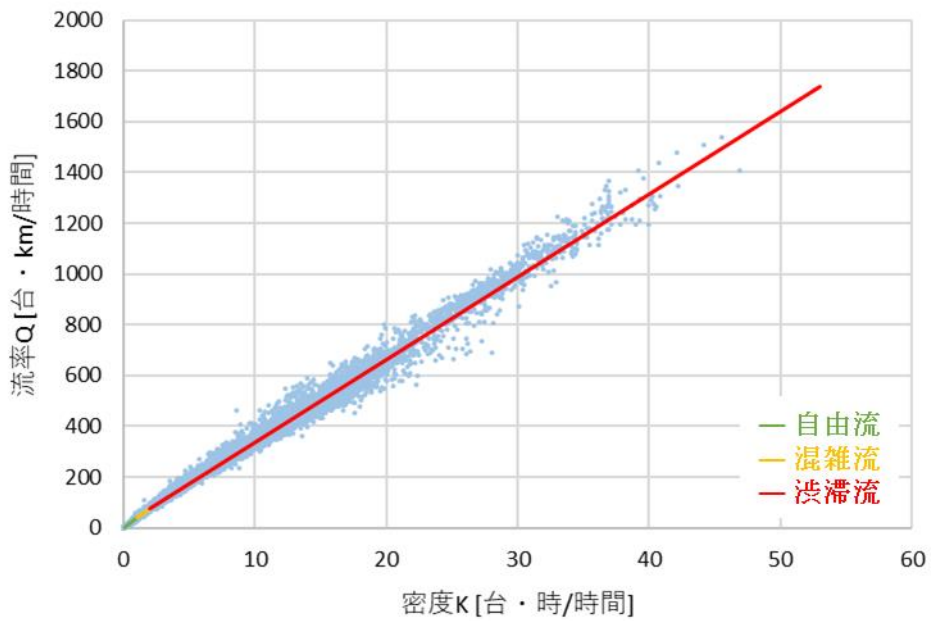


図 58 分析エリア 2 の ETC2.0 プローブ折れ線回帰モデル(渋滞流の R^2)

$$y = 34.4x + 29.9(x - 13)d_1 + 11.8(x - 14)d_2 \quad (29)$$

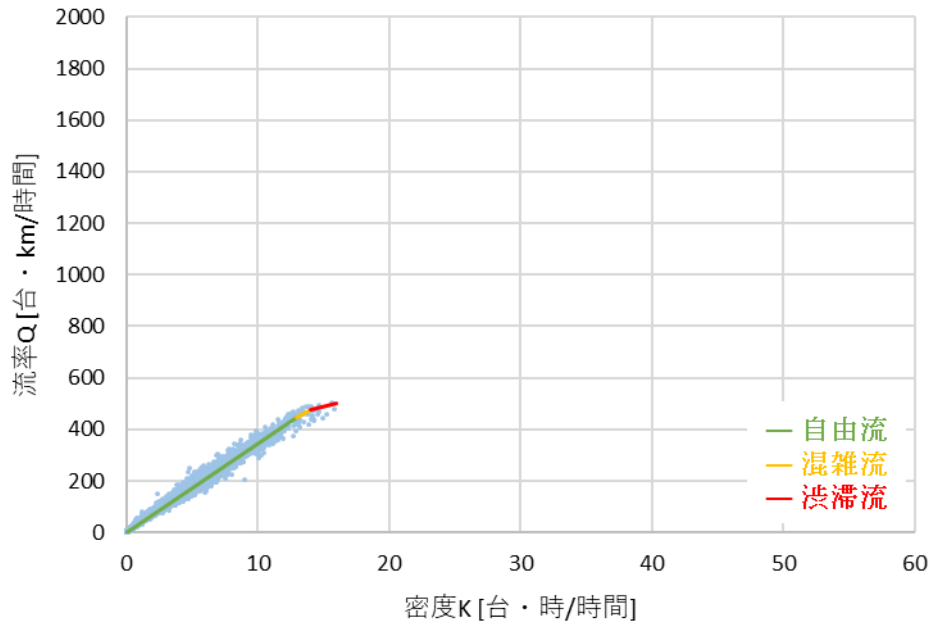


図 59 分析エリア 3 の ETC2.0 プローブ折れ線回帰モデル(自由流の R^2)

$$y = 41.5x + 33.9(x - 1)d_1 + 11.8(x - 14)d_2 \quad (30)$$

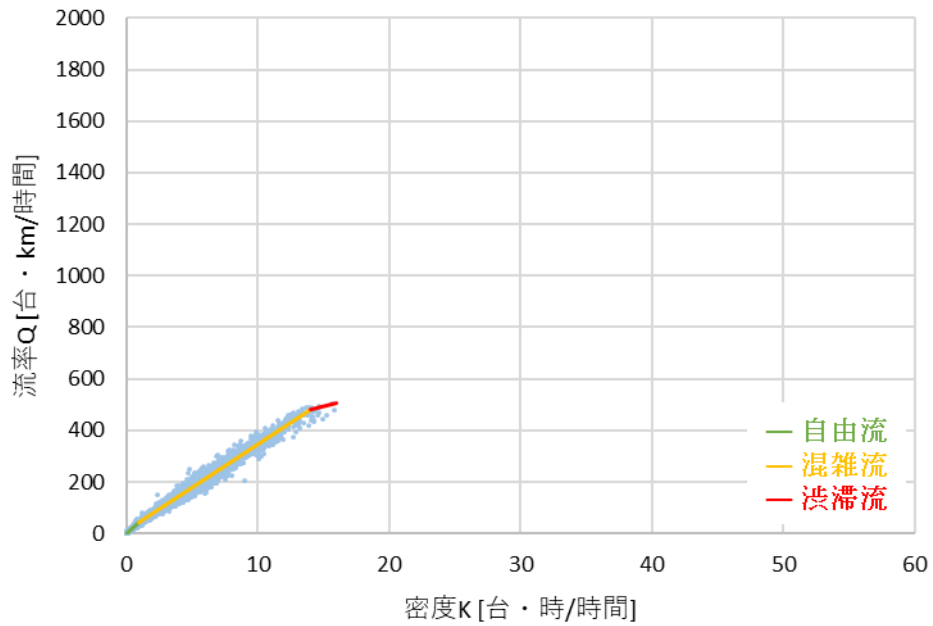


図 60 分析エリア 3 の ETC2.0 プローブ折れ線回帰モデル(混雑流の R^2)

$$y = 41.5x + 36.3(x - 1)d_1 + 33.7(x - 2)d_2 \quad (31)$$

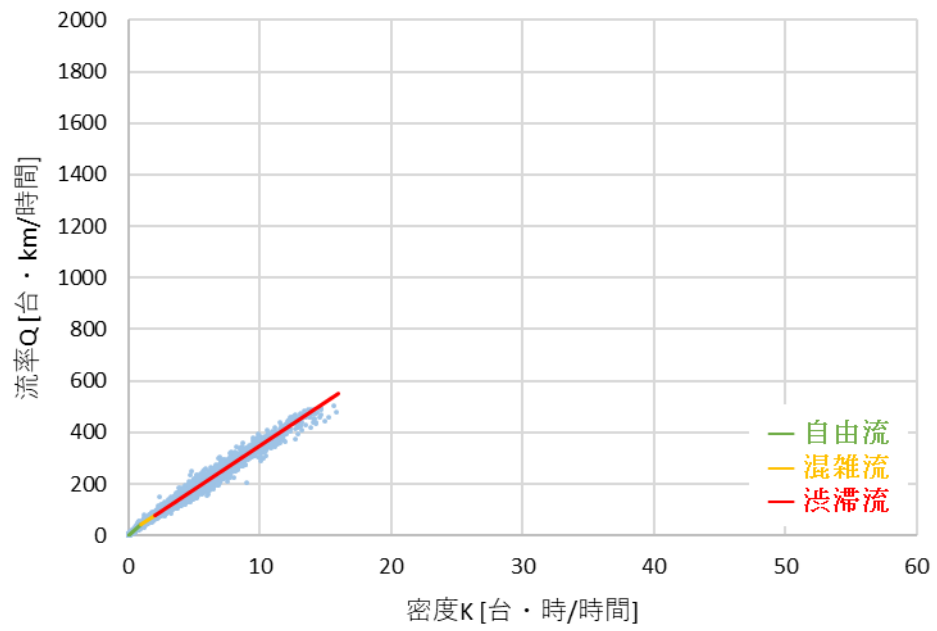


図 61 分析エリア 3 の ETC2.0 プローブ折れ線回帰モデル(渋滞流の R^2)

7-2-3 バスプローブの折れ線回帰モデル式

バスプローブの折れ線回帰モデル式を式(32)~(43)に、モデル式と MFD を重ね合わせた図を図 62~73 に示す。分析エリア 0 は混雑流の R^2 値が高い場合に折曲点 p_2 が 12[台・時/時間]となり渋滞流の領域が少なく、傾きが 3.4 と横ばいの状態が表現されている。分析エリア 1 は自由流の R^2 値が高い場合に折曲点 p_2 が 11[台・時/時間]となり渋滞流の領域が少なくなっている。分析エリア 2 は混雑流の R^2 値が高い場合に折曲点 p_2 が 9[台・時/時間]となり渋滞流の領域が少なく、傾きが 1.7 と横ばいの状態が表現されている。分析エリア 3 は自由流の R^2 値が高い場合と混雑流の R^2 値が高い場合に折曲点 p_2 が 5[台・時/時間]となり渋滞流の領域が少なくなっている。

$$y = 16.1x + 15.5(x - 2)d_1 + 6.5(x - 3)d_2 \quad (32)$$

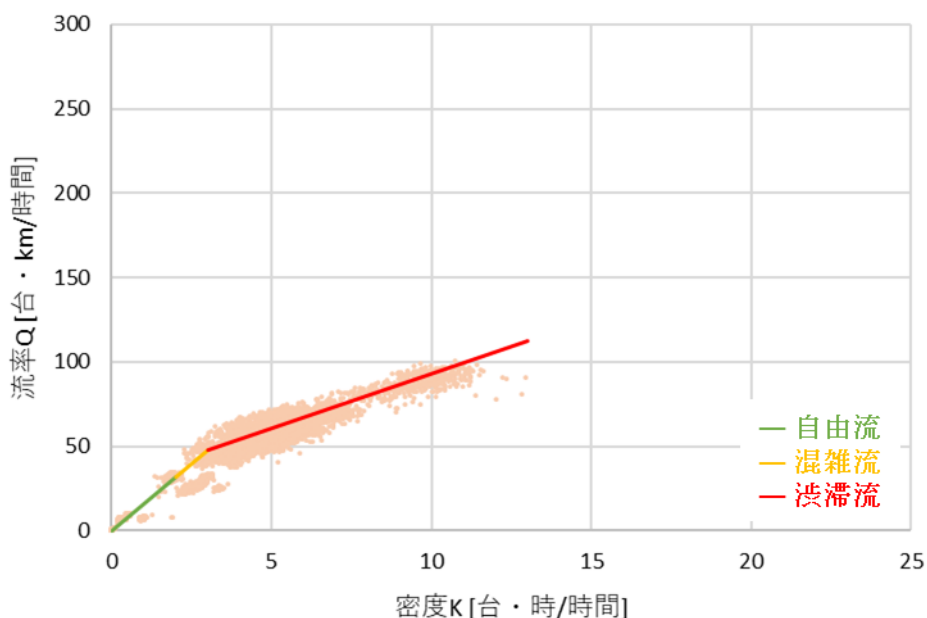


図 62 分析エリア 0 のバスプローブ折れ線回帰モデル(自由流の R^2)

$$y = 10.8x + 6.5(x - 6)d_1 + 3.4(x - 12)d_2 \quad (33)$$

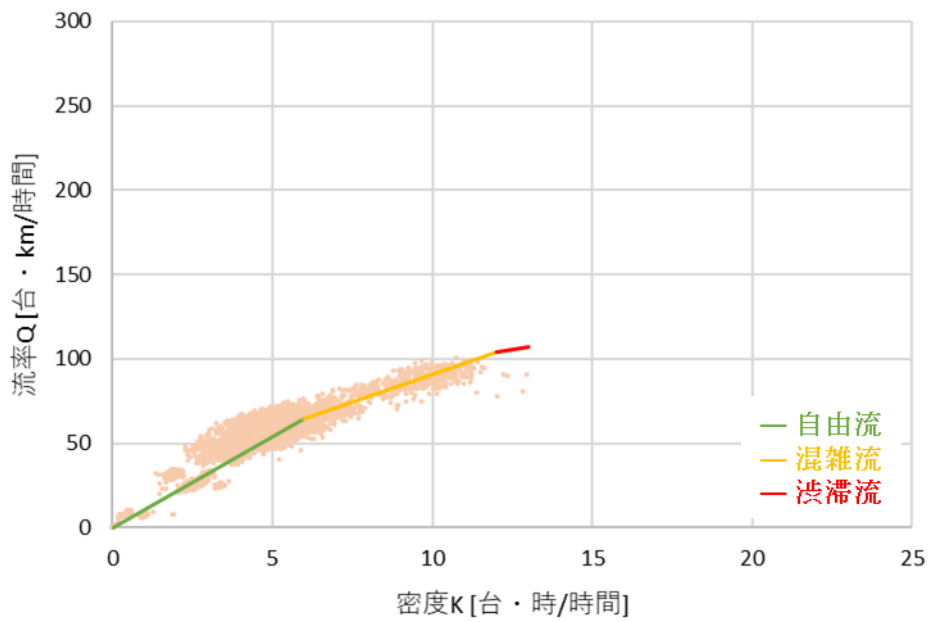


図 63 分析エリア 0 のバスプローブ折れ線回帰モデル(混雑流の R^2)

$$y = 16.1x + 8.8(x - 2)d_1 + 6.3(x - 6)d_2 \quad (34)$$

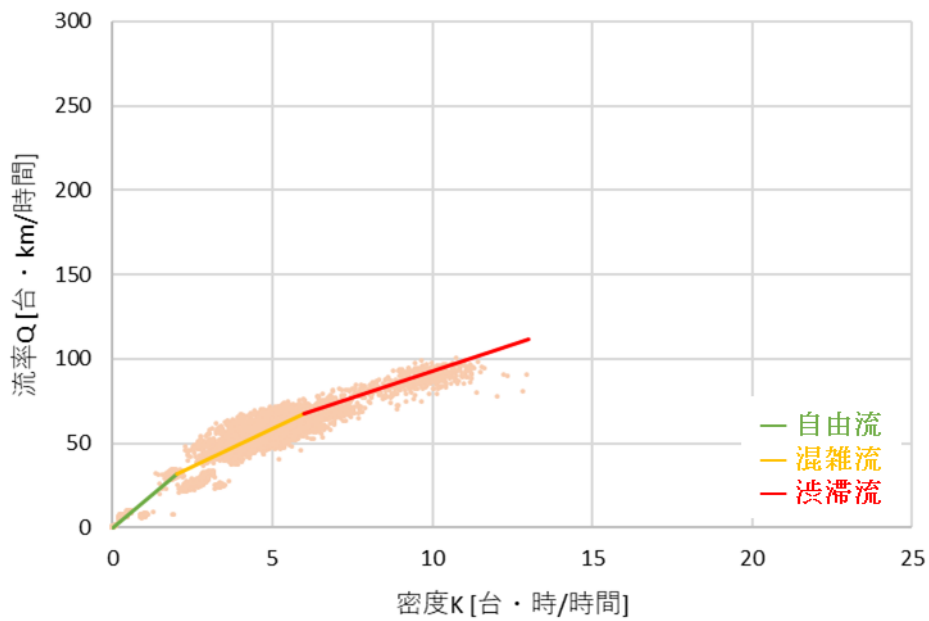


図 64 分析エリア 0 のバスプローブ折れ線回帰モデル(渋滞流の R^2)

$$y = 15.2x + 7.7(x - 9)d_1 + 7.4(x - 11)d_2 \quad (35)$$

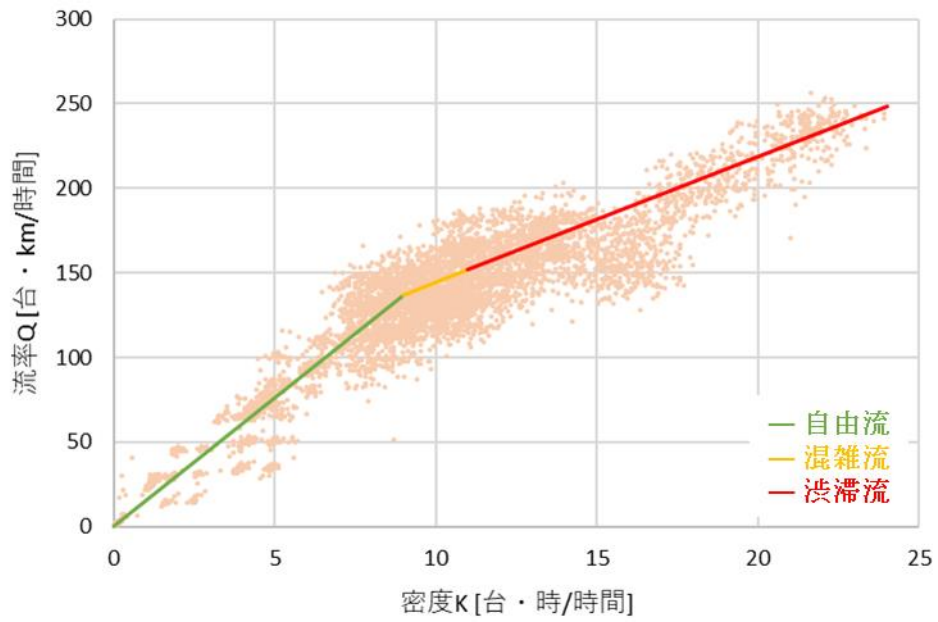


図 65 分析エリア 1 のバスプローブ折れ線回帰モデル(自由流の R^2)

$$y = 23.6x + 15.3(x - 1)d_1 + 7.1(x - 9)d_2 \quad (36)$$

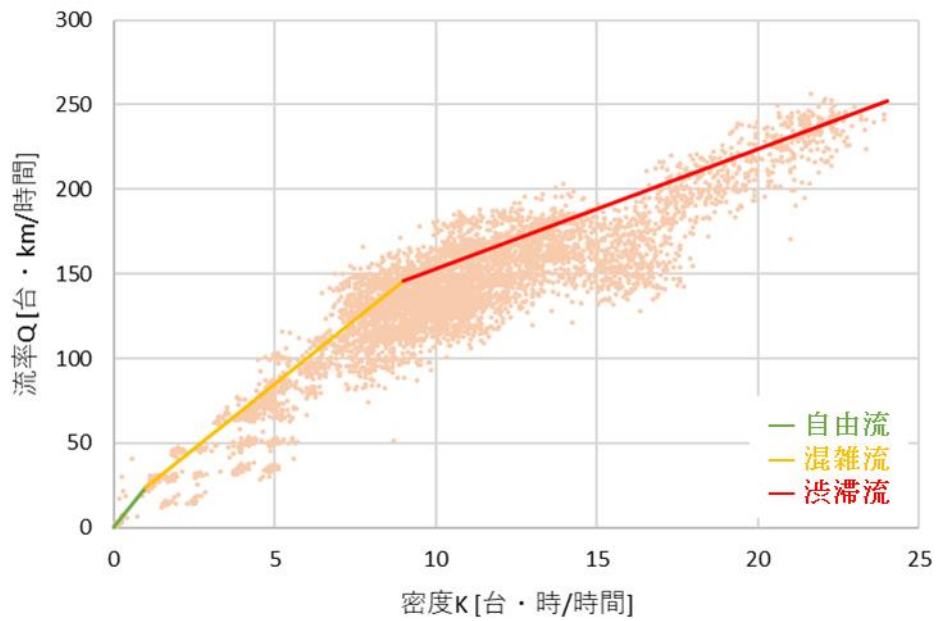


図 66 分析エリア 1 のバスプローブ折れ線回帰モデル(混雑流の R^2)

$$y = 19.3x + 9.6(x - 2)d_1 + 8.3(x - 4)d_2 \quad (310)$$

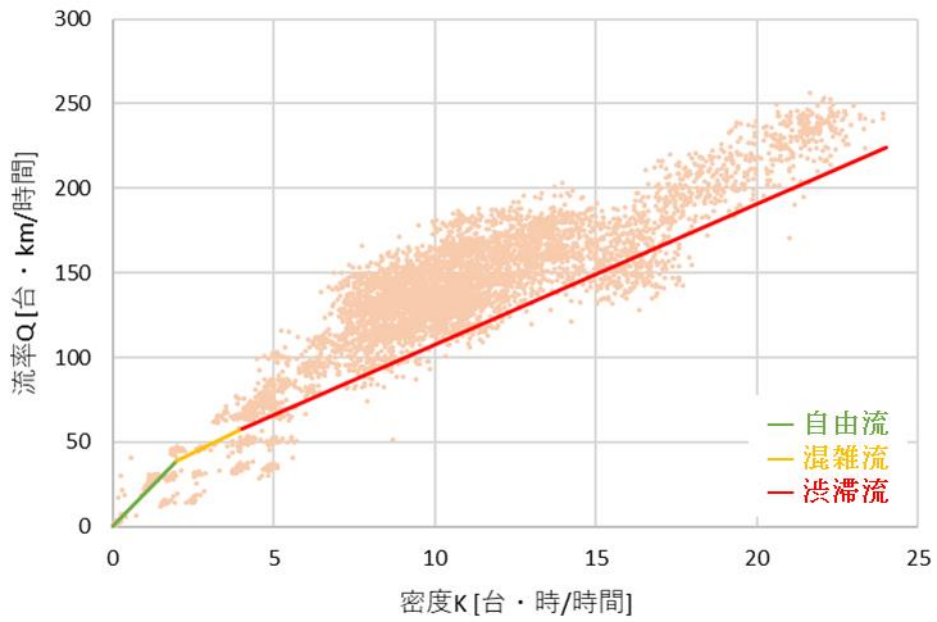


図 67 分析エリア 1 のバスプローブ折れ線回帰モデル(渋滞流の R^2)

$$y = 10.3x + 9.1(x - 1)d_1 + 6.7(x - 3)d_2 \quad (38)$$

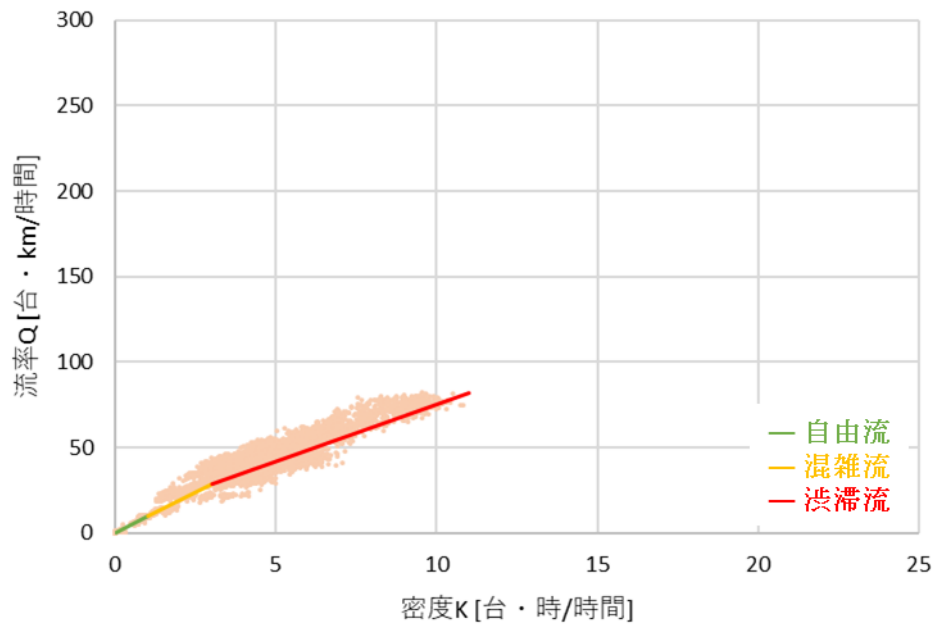


図 68 分析エリア 2 のバスプローブ折れ線回帰モデル(自由流の R^2)

$$y = 10.3x + 7.1(x - 1)d_1 + 1.7(x - 9)d_2 \quad (39)$$

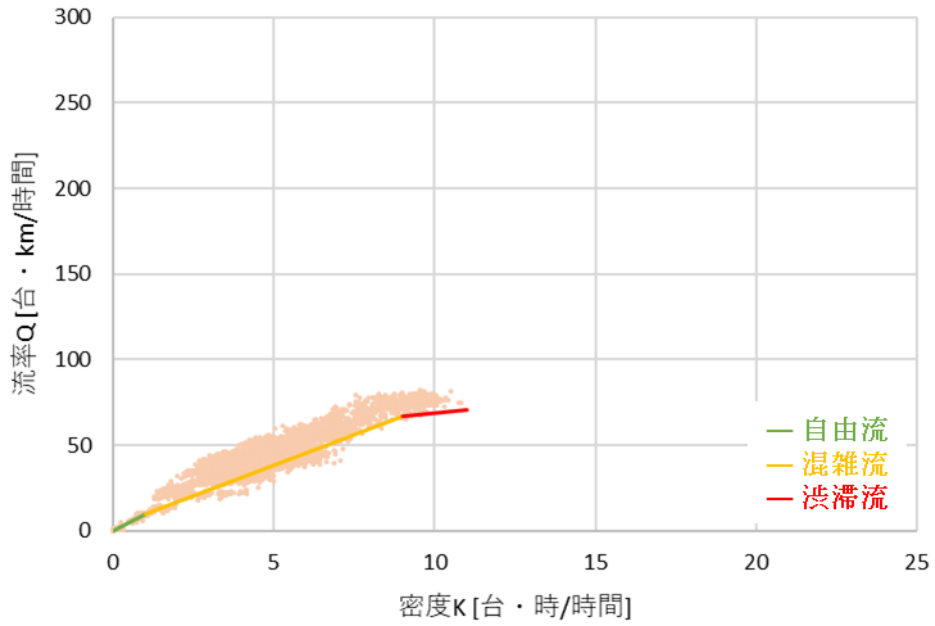


図 69 分析エリア 2 のバスプローブ折れ線回帰モデル(混雑流の R^2)

$$y = 10.3x + 9.1(x - 1)d_1 + 6.7(x - 3)d_2 \quad (40)$$

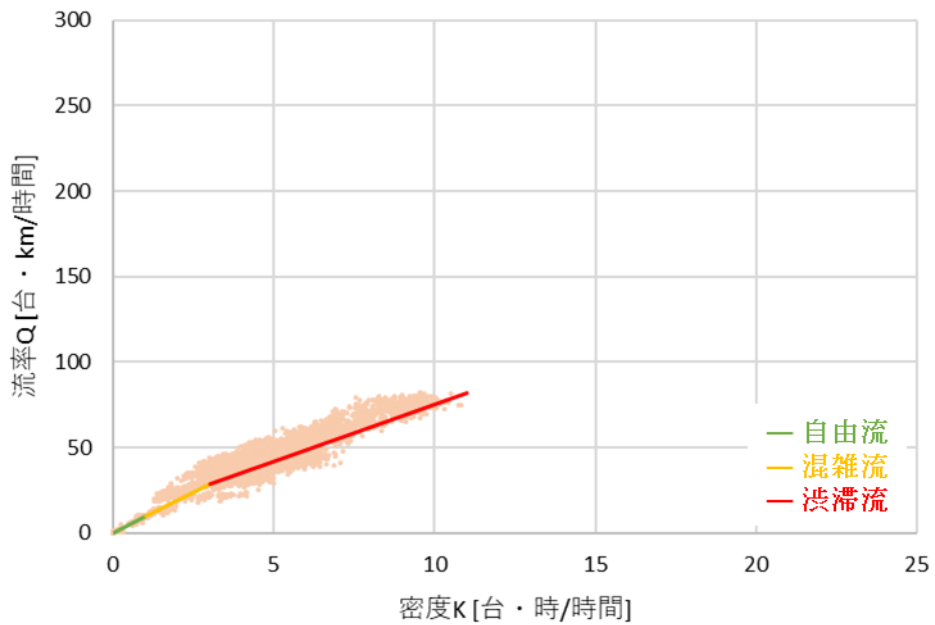


図 70 分析エリア 2 のバスプローブ折れ線回帰モデル(渋滞流の R^2)

$$y = 12.1x + 6.2(x - 2)d_1 + 4.8(x - 5)d_2 \quad (41)$$

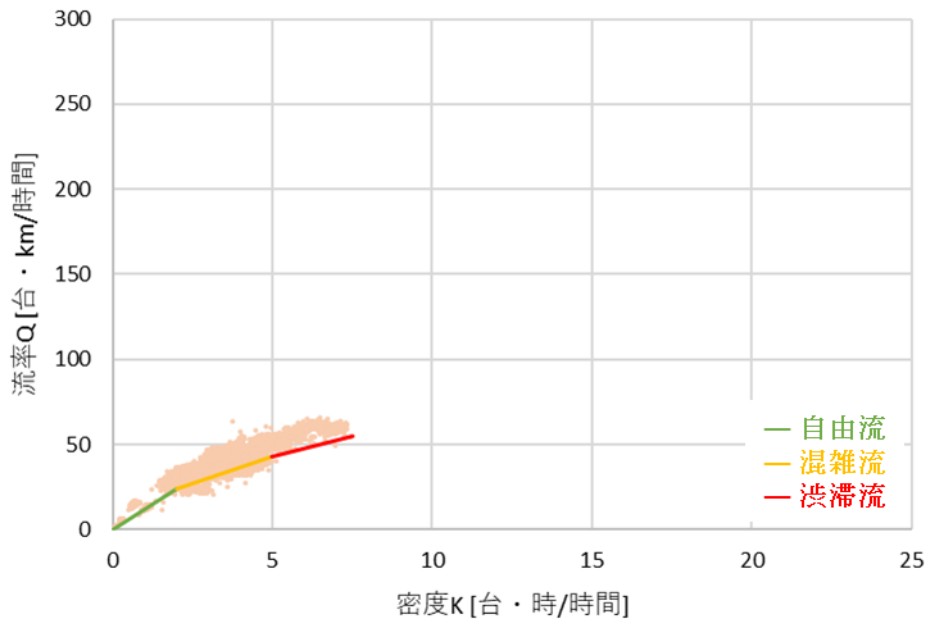


図 71 分析エリア 3 のバスプローブ折れ線回帰モデル(自由流の R^2)

$$y = 14.2x + 6.8(x - 1)d_1 + 4.8(x - 5)d_2 \quad (42)$$

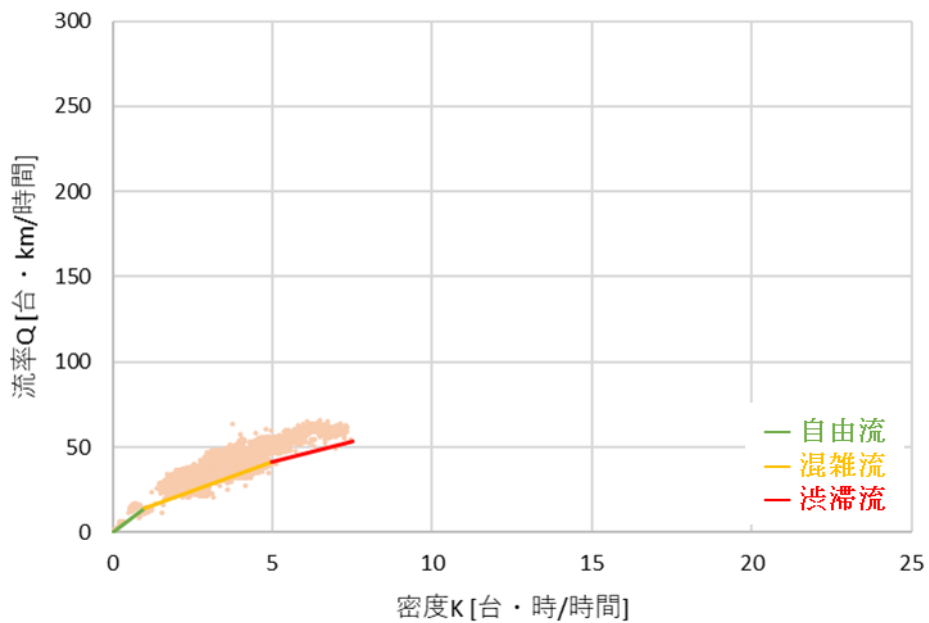


図 72 分析エリア 3 のバスプローブ折れ線回帰モデル(混雑流の R^2)

$$y = 14.2x + 8.7(x - 1)d_1 + 6.6(x - 3)d_2 \quad (43)$$

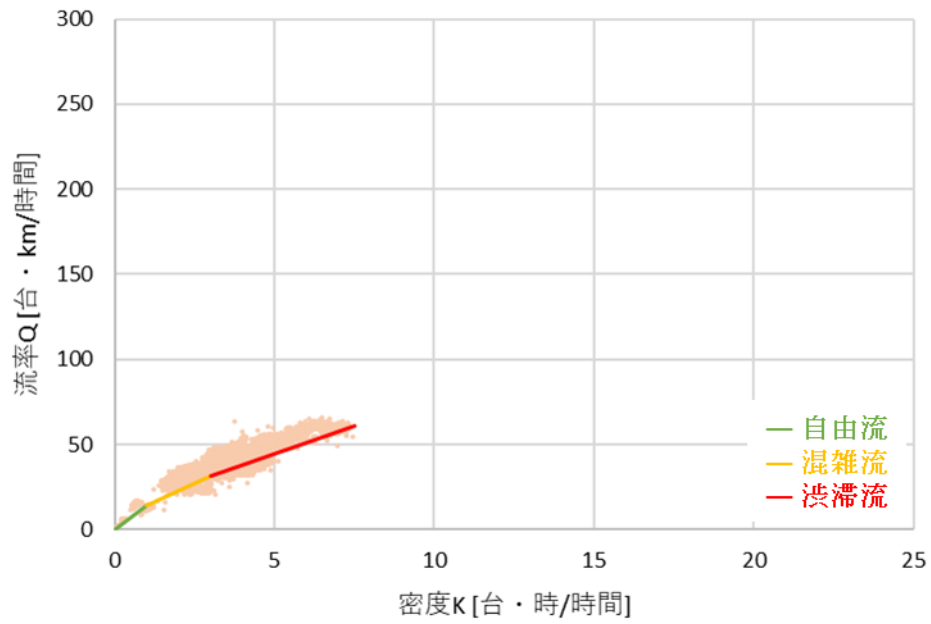


図 73 分析エリア 3 のバスプローブ折れ線回帰モデル(渋滞流の R^2)

7-3 交通流状態別の速度比較

前節の結果から、車両感知器・ETC2.0プローブ・バスプローブのすべての分析エリアにおいて、混雑流の R^2 値が高い場合の折れ線回帰モデル式を採用する。そして本節では、パラメータ β_1 が自由流を代表する速度、パラメータ β_2 が混雑流を代表する速度、パラメータ β_3 が渋滞流を代表する速度として車両感知器と ETC2.0 プローブ、バスプローブの速度を比較する。採用する折れ線回帰モデル式から交通流状態別のパラメータをまとめたものを表 20～22 に示す。

表 20 車両感知器の交通流状態別パラメータ

車両感知器	自由流	混雑流	渋滞流
分析エリア0	46.4	33.6	13.5
分析エリア1	33.4	27.4	6.0
分析エリア2	38.9	21.5	9.0
分析エリア3	41.8	19.8	2.8

表 21 ETC2.0 プローブの交通流状態別パラメータ

ETC2.0プローブ	自由流	混雑流	渋滞流
分析エリア0	45.5	34.6	34.4
分析エリア1	39.7	30.2	7.6
分析エリア2	39.1	32.7	27.8
分析エリア3	41.5	33.9	11.8

表 22 バスプローブの交通流状態別パラメータ

バスプローブ	自由流	混雑流	渋滞流
分析エリア0	10.8	6.5	3.4
分析エリア1	23.6	15.3	7.1
分析エリア2	10.3	7.1	1.7
分析エリア3	14.2	6.8	4.8

7-3-1 自由流と混雑流の速度

分析データ別にまとめた自由流と混雑流の速度比較の結果を表 23～25 に示す。バスプローブの自由流と混雑流の速度は他の 2 つに比べて低速であり、速度の水準が 30[km/h]程度低い。車両感知器と ETC2.0 プローブの速度を比較すると、その誤差は 0.2～6.3[km/h]の範囲である。一方で、車両感知器とバスプローブの速度を比較すると、その誤差は 9.8～35.6[km/h]の範囲である。そのため、バスプローブと比べて ETC2.0 プローブは車両感知器の自由流の速度が表現されていると考えられる。

しかし、ETC2.0 プローブはどの分析エリアも自由流に対する混雑流の速度の比が 0.76～0.84 と高いのに対して、バスプローブは分析エリア 3 について自由流と混雑流の速度比が 1 : 0.48 であり、車両感知器の速度比の 1 : 0.47 と類似している。そのため、分析エリア 3 についてはバスプローブデータが車両感知器の自由流と混雑流の速度比を表現している。

表 23 車両感知器の自由流と混雑流の速度比較

車両感知器	自由流の速度[km/h]	混雑流の速度[km/h]	自由流：混雑流
分析エリア0	46.4	33.6	1 : 0.72
分析エリア1	33.4	27.4	1 : 0.82
分析エリア2	38.9	21.5	1 : 0.55
分析エリア3	41.8	19.8	1 : 0.47

表 24 ETC2.0 プローブの自由流と混雑流の速度比較

ETC2.0プローブ	自由流の速度[km/h]	混雑流の速度[km/h]	自由流：混雑流
分析エリア0	45.5	34.6	1 : 0.76
分析エリア1	39.7	30.2	1 : 0.76
分析エリア2	39.1	32.7	1 : 0.84
分析エリア3	41.5	33.9	1 : 0.82

表 25 バスプローブの自由流と混雑流の速度比較

バスプローブ	自由流の速度[km/h]	混雑流の速度[km/h]	自由流：混雑流
分析エリア0	10.8	6.5	1 : 0.60
分析エリア1	23.6	15.3	1 : 0.65
分析エリア2	10.3	7.1	1 : 0.69
分析エリア3	14.2	6.8	1 : 0.48

7-3-2 自由流と渋滞流の速度

分析データ別にまとめた自由流と混雑流の速度比較の結果を表 26～28 に示す。分析エリア 0 と分析エリア 2 について、ETC2.0 プローブの速度は渋滞流であっても 34.4[km/h]、27.8[km/h]と高いままである。車両感知器とバスプローブの渋滞流の速度に注目すると、車両感知器の速度の最大値は 13.5[km/h]、バスプローブの速度の最大値は 7.1[km/h]とどちらも速度の値は低く、分析エリア 1 と分析エリア 3 については速度の差が 1.1[km/h]、2.0[km/h] であり、類似している。さらに、車両感知器とバスプローブの自由流に対する渋滞流の速度の比は分析エリア 0 が 0.29 と 0.31、分析エリア 2 が 0.23 と 0.17 と類似している。そのため、ETC2.0 プローブよりバスプローブの方が渋滞流の速度、自由流と渋滞流の速度比を表現している。

表 26 車両感知器の自由流と渋滞流の速度比較

車両感知器	自由流の速度[km/h]	渋滞流の速度[km/h]	自由流：渋滞流
分析エリア0	46.4	13.5	1 : 0.29
分析エリア1	33.4	6.0	1 : 0.18
分析エリア2	38.9	9.0	1 : 0.23
分析エリア3	41.8	2.8	1 : 0.07

表 27 ETC2.0 プローブの自由流と渋滞流の速度比較

ETC2.0プローブ	自由流の速度[km/h]	渋滞流の速度[km/h]	自由流：渋滞流
分析エリア0	45.5	34.4	1 : 0.76
分析エリア1	39.7	7.6	1 : 0.19
分析エリア2	39.1	27.8	1 : 0.71
分析エリア3	41.5	11.8	1 : 0.28

表 28 バスプローブの自由流と渋滞流の速度比較

バスプローブ	自由流の速度[km/h]	渋滞流の速度[km/h]	自由流：渋滞流
分析エリア0	10.8	3.4	1 : 0.31
分析エリア1	23.6	7.1	1 : 0.30
分析エリア2	10.3	1.7	1 : 0.17
分析エリア3	14.2	4.8	1 : 0.34

7-4 交通流状態別速度と分析使用データ・分析エリアの関係性に関する考察

7-4-1 交通流状態別速度と分析使用データ

前節の結果からすべての分析エリアにおいて、自由流の速度の把握にはバスプローブデータより ETC2.0 プローブデータの方が有効であることを示唆した。バスプローブデータはバス停での乗降時間や待ち時間を含むため、速度が遅くなる。それに対して、自由流では他の車両の影響を受けずに走行できるため、ETC2.0 プローブデータは台数が少なくても ETC2.0 プローブ車両の速度を全体の車両の代表値としても問題ないと考えられる。

一方で、渋滞流の速度の把握には ETC2.0 プローブデータよりバスプローブデータの方が有効であることを示唆した。ETC2.0 プローブデータは渋滞流において観測される車両は数パーセントであり、密度が実際の交通流より小さい値を取る。速度は傾きにあたるため、同じ流率に対して密度が小さいと傾きが大きくなり、速度が早くなる。

それに対して、バスプローブデータは交通渋滞が発生している道路上ではバス停の到着が遅れ、時刻表の時間に合わせるための停止時間が存在しないことが挙げられる。また、すべての車線の車両が詰まっていると、利用者の乗降のためにバス停で停車しているバス車両を一般車両は追い越すことが出来ず、速度に差が生じないと考えられる。さらに、バス車両が利用者の乗降のためにバス停で停車したとしても、前方で車両が詰まっているとそこに追いつき、どの車両にも平等に待ち時間が生じることも考えられる。以上のことから、車両感知器データの渋滞流の速度を表現できたと考えられる。

7-4-2 交通流状態別速度と分析エリア

前節の結果で分析エリア 1 と分析エリア 3 において、車両感知器データとバスプローブデータの渋滞流の速度が類似していることが明らかになった。また、分析エリア 0 と分析エリア 2 において、車両感知器データとバスプローブデータの自由流と渋滞流の速度の比が類似していることが明らかになった。分析エリア 1 と分析エリア 3、分析エリア 0 と分析エリア 2 のエリアの特徴の違いとして、バスプローブデータが観測された位置が挙げられる。分析エリア 1 と分析エリア 3 は高知市の主要渋滞区間とバス路線が重なる範囲が多いが、分析エリア 0 と分析エリア 2 は主要渋滞箇所とバスプローブデータが観測された位置が重なっている。

つまり、分析エリア 1 と分析エリア 3 は渋滞流を面的に捉えることができるため、渋滞流を走行したバス車両が車両感知器データの速度を表現したと考えられる。しかしながら、バス路線が多く、その分自由流ではバス停での乗降時間や待ち時間を多く含むため、自由流と渋滞流の速度の比を表現できなかったと考えられる。また、分析エリア 0 と分析エリア 2 は渋滞流を点で捉えているため、車両感知器データの速度を表現できなかったが、自由流と渋滞流の速度の比は表現できたと考えられる。

また、前節の結果で分析エリア 3 において、車両感知器データとバスプローブデータの自由流と混雑流の速度の比が類似していることが明らかになった。分析エリア 1 と分析エリ

ア 3 は高知市の主要渋滞区間とバス路線が重なる範囲が多いという共通の特徴を持つが、分析エリア 1 において車両感知器データとバスプローブデータの自由流と混雑流の速度の比は類似しなかった。その理由として、分析エリア 3 の折れ線回帰モデル式はバスプローブの MFD の 3 つのデータ分布の状態に沿って交通流状態が分かっていたのに対し、分析エリア 1 の折れ線回帰モデル式はバスプローブの MFD の 3 つのデータ分布の状態に沿って交通流状態が分かれていなかったことが考えられる。

8 結論

8-1 研究成果

本研究では、車両感知器データや ETC2.0 プローブデータと同様にバスプローブデータを用いて高知市中心部の MFD を描画した。分析エリアを 4 つ定め、分析エリアごとの MFD をそれぞれデータで描画したところ、車両感知器とバスプローブの MFD は分析エリアごとに MFD の傾きである速度に差があることが明らかになった。

速度に着目し、バスプローブデータが実際の交通流状態を示す車両感知器データの速度を表現できているか、折れ線回帰モデルを用いてデータ間で比較した。自由流の R^2 値が高い場合、混雑流の R^2 値が高い場合、渋滞流の R^2 値が高い場合のそれぞれで折れ線回帰モデル式を算出した結果、混雑流の R^2 値が高い場合の折れ線回帰モデル式が実際の交通流状態の出現割合に近いことが明らかになった。

車両感知器と ETC2.0 プローブの自由流の速度の誤差はすべての分析エリアにおいて小さいことが明らかになった。そのため、道路ネットワークの自由流の速度の把握にはバスプローブデータより ETC2.0 プローブデータの方が有効であることを示唆した。

分析エリア 1 と分析エリア 3 において、車両感知器データとバスプローブデータの渋滞流の速度が類似していることが明らかになった。そのため、主要渋滞区間とバス路線の重なる範囲が多いという特徴を持つ道路ネットワークではバスプローブデータを用いることで渋滞流の速度を把握できることを示唆した。

また、分析エリア 0 と分析エリア 2 において、車両感知器データとバスプローブデータの自由流と渋滞流の速度の比が類似していることが明らかになった。そのため、主要渋滞箇所とバスプローブデータが観測された位置が重なっているという特徴を持つ道路ネットワークではバスプローブデータを用いることで自由流と渋滞流の速度の比を把握できることを示唆した。

8-2 今後の課題

本研究では、バスロケーションデータをバスプローブデータとして利用し、データクレンジング等の加工をせずに分析を進めた。そのため、自由流や混雑流の速度の水準が車両感知器データや ETC2.0 プローブデータに比べて低くなっていた。そこで、バス利用者の乗降時間や時刻表との時間合わせのための停止時間を取り除きデータを補正することで、自由流や混雑流の速度が他のデータに近づき、交通流状態の把握により有効なデータになると考えられる。

謝辞

本研究を進めるにあたって、多くの方々からの多大なるご指導、ご助言、ご支援を頂戴いたしました。ここに心より感謝申し上げます。

高知工科大学システム工学群の西内裕晶准教授には本研究を遂行するにあたり、終始熱心なご指導を頂きました。深く感謝いたします。

副指導教員である高知工科大学システム工学群の赤塚慎准教授ならびに論文副審査である高知工科大学システム工学群の高木方隆教授には多くのご助言を頂きました。深く感謝いたします。

愛媛大学理工学研究科の坪田隆宏講師には研究方針等、多くのご助言を頂きました。深く感謝いたします。

高知大学教育研究部総合科学系地域協働教育学部門の岡村健志准教授にはゼミの際に多くのご助言を頂きました。深く感謝いたします。

高知県警察本部高知県交通管制センターの皆様には車両感知器データの提供、ご支援を頂きました。深く感謝いたします。

国土交通省四国地方整備局土佐国道事務所の皆様にはETC2.0プローブデータの提供、ご支援を頂きました。深く感謝いたします。

とさでん交通株式会社の皆様にはバスロケーションデータの提供、ご支援を頂きました。深く感謝いたします。

都市・交通計画研究室の皆様にはゼミやカンファレンスの際に多くのご助言を頂きました。深く感謝いたします。

2023年3月10日

吉田 朋恵

参考文献

- 1) 国土交通省：道路交通センサスのデータ収集の現状と課題 資料 3, pp.2, <https://www.mlit.go.jp/road/ir/ir-council/ict/pdf01/03.pdf>, (2023/01/01)
- 2) 一般財団法人 ITS サービス高度化機構：ETC/ETC2.0(DSRC)普及状況 ETC 総合情報ポータルサイト, <https://www.go-etc.jp/fukyu/index.html>, (2023/01/01)
- 3) Daganzo, C. F. : Urban gridlock: Macroscopic modeling and mitigation approaches , Transportation Research Part B, Vol.41, No.1, pp.49-62, 2007.
- 4) Geroliminis, N. and Daganzo, C. F. : Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings, Transportation Research Part B, Vol.42, No.9, pp.759-770, 2008.
- 5) 土生恭祐, 桑原雅夫, 原祐輔 : 交通状態の時空間変化が MFD に与える影響, 土木計画学研究・講演集, Vol.47, 2013.
- 6) 王鵬飛, 赤松隆, 和田健太郎 : Macroscopic Fundamental Diagram における渋滞領域発生メカニズムに関する研究, 土木計画学研究・講演集, Vol.51, 2015.
- 7) 王鵬飛, 和田健太郎, 赤松隆, 杉田正俊, 名古屋利一, 鷲見浩 : 長期間観測データを用いた Macroscopic Fundamental Diagram の特徴づけ : 仙台市および京都市におけるケース・スタディ, 交通工学論文集, Vol.2, No.5, pp.11-20, 2016.
- 8) 酒井高良, 赤松隆 : 首都圏高速道路網における渋滞パターンと MFD の関係性, 土木計画学研究・講演集, Vol.58, 2018.
- 9) 尾高慎二, 吉井稔雄, 奥原瑠依, 神戸信人 : 交通事故が CBD 街路網の MFD 形状に与える影響の分析, 土木計画学研究・講演集, Vol.60, 2019.
- 10) 奥原瑠依, 吉井稔雄, 坪田隆宏 : MFD を用いた道路ネットワーク性能の動的変化解析手法, 土木計画学研究・講演集, Vol.63, 2021.
- 11) 後藤梓, 小木曾俊夫, 牧佑奈, 池田裕二 : ETC2.0 プローブデータの量的代表性に関する分析, 第 16 回 ITS シンポジウム 2018 Proceedings, 2018.
- 12) 桑原雅夫, 吉井稔雄, 堀口良太 : ブロック密度法を用いた交通流の表現方法について, 交通工学, Vol.32, No.4, 1997.
- 13) Open Database License : www.openstreetmap.org/copyright
- 14) GEOFABRIK downloads : <https://download.geofabrik.de/asia/japan/shikoku.html>
- 15) Geocoding : <https://www.geocoding.jp/?q=>

付録

車両感知器番号ごとのデータに分類するコード

```
import pandas as pd
import glob
from pathlib import Path
import csv

date = None

for path1 in glob.glob("D:/感知器/2019年4月分/*.xls"):
    new_path1 = Path(path1)
    year_1 = new_path1.parts[-2]
    data_1 = new_path1.stem

    df = pd.read_excel(path1, header=None, skiprows=[0,1,3,4], usecols=range(3,31),
sheet_name=None)

    for key in df.keys():
        for row in range(int(len(df[key].columns)/4)):
            number = row*4

            df_1 = df[key].iloc[0:289,number:number+4]
            detector_number = str(df_1.iat[0,0])
            detector_number_1 = detector_number[:6]

            df_2 = df[key].iloc[1:289,number:number+4]
            df_2.to_csv("D:/detector/2019_04/" +str(data_1[21:23])+"/" + detector_number_1 +
".csv", mode="a", index = False, header=False)
```

車両感知器ごとに観測情報とリンク情報をまとめるコード

```
import os
import pandas as pd
import numpy as np
import csv
import glob
from pathlib import Path

list_1 =
np.loadtxt(fname="D:/link_length1/detector_link_length1.csv",dtype="unicode",delimiter=",",comm
ents="#",skiprows=1,usecols=None)

for path1 in glob.glob("D:/detector/**/*/*.csv"):
    new_path1 = Path(path1)
    year_1 = new_path1.parts[-3]
    date_1 = new_path1.parts[-2]
    data_1 = new_path1.stem

    for row in list_1:
        if data_1 == str(row[0]):

            df = pd.read_csv(path1, header=None)
            df.columns = ["traffic","occupancy","speed","abnormal"]

            for data in range(24):
                number = data*12

                detector = data_1

                traffic = df["traffic"][number:number+12].sum()

                speed = df["speed"][number:number+12].mean()

                link = row[2]
```



```

        with open("D:/detector_ts/" + str(year_1) + "/" + str(date_1) + "_" +
str(str(data).zfill(2)) + ".csv", "a") as f:
            writer = csv.writer(f, lineterminator = "\n")
            writer.writerow((detector, traffic, speed, link))

```

走行台キロと車両存在台数を算出するコード

```

import pandas as pd
import csv
from pathlib import Path
import glob
import numpy as np

# 読み込み
for path1 in glob.glob("D:/detector_ts/**/*.csv"):
    new_path1 = Path(path1)
    year_1 = new_path1.parts[-2]
    data_1 = new_path1.stem

    data = np.loadtxt(path1, delimiter=',', skiprows = 0, dtype = "unicode")

    # ヘッダー追加
    with open("D:/detector_pn/" + str(year_1) + "/" + str(data_1) + ".csv", "a") as f:
        writer = csv.writer(f, lineterminator = "\n")
        writer.writerow(("p", "n"))

    for row in data:

        # 走行台キロ (流率)
        p = float(row[1]) * float(row[3])

        # 車両存在台数 (密度)
        if float(row[2]) == 0:
            n = 0

```

```

else:
    n = (float(row[1]) / float(row[2])) * float(row[3])

    # CSV 出力
    with open("D:/detector_pn/" + str(year_1) + "/" + str(data_1) + ".csv","a") as f:
        writer = csv.writer(f,lineterminator = "\n")
        writer.writerow((p,n))

```

車両感知器の流率と密度を算出するコード

```

import pandas as pd
import csv
from pathlib import Path
import glob
import numpy as np

# ヘッダー追加
with open("D:/detector_sum.csv","a") as f:
    writer = csv.writer(f,lineterminator = "\n")
    writer.writerow(("q","k"))

# 読み込み
for path1 in glob.glob("D:/detector_pn/*/*.csv"):
    new_path1 = Path(path1)
    year_1 = new_path1.parts[-2]
    data_1 = new_path1.stem

    df = pd.read_csv(path1)

    # 走行台キロの合計 (流率)
    q = df["p"].sum()

    # 車両存在台数の合計 (密度)

```

```

k = df["n"].sum()

# CSV 出力
with open("D:/detector_sum.csv","a") as f:
    writer = csv.writer(f,lineterminator = "\n")
    writer.writerow((q,k))

```

ETC2.0 プローブの内外判定のコード(緯度経度の数値は分析エリア 0 の場合)

```

import pandas as pd
import csv
import datetime
import numpy as np
import glob
from pathlib import Path
from turfpy.measurement import boolean_point_in_polygon
from geojson import Point, Polygon, Feature

df1 =
np.loadtxt(fname="C:/Users/B302/Desktop/list.csv",dtype="unicode",delimiter=" ",comments="#",s
kiprows=0,usecols=None)

hour =
["00","01","02","03","04","05","06","07","08","09","10","11","12","13","14","15","16","17","18","1
9","20","21","22","23"]

for path1 in glob.glob("E:/ETC2.0_data/*//*.csv"):
    data = np.loadtxt(path1, delimiter=',', skiprows = 0, dtype = "unicode")
    for row in data:
        new_path1 = Path(path1)
        year_1 = new_path1.parts[-3]
        OUT_1 = new_path1.parts[-2]
        data_1 = new_path1.stem

```

```

if not row[21] or not row[22]:
    pass

else:
    point = Feature(geometry=Point((float(row[21]), float(row[22])))
    polygon = Polygon([[(133.5054554, 33.55011411),
                        (133.5056731, 33.54891571),
                        (133.5062279, 33.54901495),
                        (133.5152527, 33.5491387),
                        (133.5234711, 33.54900696),
                        (133.5396526, 33.53954529),
                        (133.5405502, 33.53953108),
                        (133.5499249, 33.54470434),
                        (133.554761, 33.54802174),
                        (133.5554223, 33.55122014),
                        (133.5570399, 33.55284598),
                        (133.5540475, 33.55424565),
                        (133.5480006, 33.55561453),
                        (133.5465465, 33.55598399),
                        (133.542653, 33.5563416),
                        (133.5374948, 33.55507811),
                        (133.5300862, 33.55424091),
                        (133.5229824, 33.55383416),
                        (133.5169954, 33.55476076),
                        (133.5150297, 33.55476549),
                        (133.5090177, 33.55207273),
                        (133.5069964, 33.55225983),
                        (133.5054554, 33.55011411)]]])

if boolean_point_in_polygon(point, polygon):
    date_str2 = str(row[6])
    for date in df1:
        if str(date_str2[0:8]) == str(date[2]):
            for hourrow in hour:
                if str(row[6][8:10]) == hourrow:
                    # ファイルオープン#編集

```

```

f = open("E:/ETC2.0_out/" + str(year_1) + "/" + date[1] +
"_" + hourrow + ".csv", 'a')

writer = csv.writer(f, lineterminator='\n')
# 出力
writer.writerow(row)
# ファイルクローズ
f.close()

```

ETC2.0 プローブの走行距離と走行時間を算出するコード

```

import pandas as pd
import csv
from pathlib import Path
import glob
import datetime

# 読み込み
for path1 in glob.glob("D:/ETC2.0_out_1/*/*.csv"):
    new_path1 = Path(path1)
    year_1 = new_path1.parts[-2]
    data_1 = new_path1.stem

    df1 = pd.read_csv(path1,header=None)

    # リスト名の追加
    df1.columns =
["RSU_ID_code","receipt_time","date_of_trip","trip_ID1","vehicle_type","vehicle_use","GPS_time
","serial_number","trip_number","trip_start_time","trip_end_time","trip_completeness","trip_origin
_and_destination_flags","tag_number","longitude","latitude","accumulation_condition","road_type_c
ode","speed","altitude","matching_flag","longitude_after_matching","latitude_after_matching","DR
M_version","secondary_mesh_code","inflow_node","outflow_node","distance_from_inflow_node",
"fix_flag","traffic_survey_basic_section_number","upstream/downstream_code","administrator_cod
e","update_time"]

```

```

# ヘッダー追加
with open("D:/ETC2.0_id_1/" + str(year_1) + "/" + str(data_1) + ".csv","a") as f:
    writer = csv.writer(f,lineterminator = "\n")
    writer.writerow(("speed_kmh","time_h","distance_km"))

# グループ化
for trip_ID,df1_1 in df1.groupby("trip_ID1"and"trip_start_time"and"trip_end_time"):

    # 並び替え
    df1_2 = df1_1.sort_values("serial_number")

    # 速度の平均
    speed = df1_2["speed"].mean()

    # 走行時間の差分
    date_str1 = str(df1_2.iloc[-1]["GPS_time"])
    date_str1_1 = date_str1[:14]
    date_dt1 = datetime.datetime.strptime(date_str1_1,"%Y%m%d%H%M%S")

    date_str2 = str(df1_2.iloc[0]["GPS_time"])
    date_str2_1 = date_str2[:14]
    date_dt2 = datetime.datetime.strptime(date_str2_1,"%Y%m%d%H%M%S")

    time1 = date_dt1 - date_dt2
    time = time1.total_seconds() / 3600

    # 走行距離の算出
    distance = speed * time

# CSV 出力
with open("D:/ETC2.0_id_1/" + str(year_1) + "/" + str(data_1) + ".csv","a") as f:
    writer = csv.writer(f,lineterminator = "\n")
    writer.writerow((speed,time,distance))

```

ETC2.0 プローブの流率と密度を算出するコード

```
import pandas as pd
import csv
from pathlib import Path
import glob

# ヘッダー追加
with open("D:/ETC2.0_sum_1.csv","a") as f:
    writer = csv.writer(f,lineterminator = "\n")
    writer.writerow(("q","k"))

# 読み込み
for path1 in glob.glob("D:/ETC2.0_id_1/*/*.csv"):
    new_path1 = Path(path1)
    year_1 = new_path1.parts[-2]
    data_1 = new_path1.stem

    df = pd.read_csv(path1)

    # 走行距離の合計 (流率)
    q = df["distance_km"].sum()

    # 走行時間の合計 (密度)
    k = df["time_h"].sum()

# CSV 出力
with open("D:/ETC2.0_sum_1.csv","a") as f:
    writer = csv.writer(f,lineterminator = "\n")
    writer.writerow((q,k))
```

バスプローブの内外判定のコード(緯度経度の数値は分析エリア 0 の場合)

```
import pandas as pd
import csv
import datetime
import numpy as np
import glob
from pathlib import Path
from turfpy.measurement import boolean_point_in_polygon
from geojson import Point, Polygon, Feature

hour =
["00","01","02","03","04","05","06","07","08","09","10","11","12","13","14","15","16","17","18","1
9","20","21","22","23"]

for path1 in glob.glob("D:/bus_data/*/*.csv"):
    data = np.loadtxt(path1, delimiter=',', skiprows = 1, dtype = "unicode")
    for row in data:
        new_path1 = Path(path1)
        year_1 = new_path1.parts[-2]
        data_1 = new_path1.stem

        if not row[23] or not row[24]:
            pass

        else:
            point = Feature(geometry=Point((float(row[24]), float(row[23]))))
            polygon = Polygon([[(133.5054554, 33.55011411),
                                (133.5056731, 33.54891571),
                                (133.5062279, 33.54901495),
                                (133.5152527, 33.5491387),
                                (133.5234711, 33.54900696),
                                (133.5396526, 33.53954529),
                                (133.5405502, 33.53953108),
                                (133.5499249, 33.54470434),
```



```
(133.554761, 33.54802174),
(133.5554223, 33.55122014),
(133.5570399, 33.55284598),
(133.5540475, 33.55424565),
(133.5480006, 33.55561453),
(133.5465465, 33.55598399),
(133.542653, 33.5563416),
(133.5374948, 33.55507811),
(133.5300862, 33.55424091),
(133.5229824, 33.55383416),
(133.5169954, 33.55476076),
(133.5150297, 33.55476549),
(133.5090177, 33.55207273),
(133.5069964, 33.55225983),
(133.5054554, 33.55011411]]])
```

```
if boolean_point_in_polygon(point, polygon):
```

```
    for hourrow in hour:
```

```
        if str(row[18][0:2]) == hourrow:
```

```
            # ファイルオープン#編集
```

```
            f = open("D:/bus_out/" + str(year_1) + "/" + str(data_1) + "_" +
hourrow + ".csv", 'a')
```

```
            writer = csv.writer(f, lineterminator='\n')
```

```
            # 出力
```

```
            writer.writerow(row)
```

```
            # ファイルクローズ
```

```
            f.close()
```

バスプローブの走行距離と走行時間を算出するコード

```
import pandas as pd
```

```
import csv
```

```
from pathlib import Path
```

```

import glob
import datetime

# 読み込み
for path1 in glob.glob("D:/bus_out/*//*.csv"):
    new_path1 = Path(path1)
    year_1 = new_path1.parts[-2]
    data_1 = new_path1.stem

    df1 = pd.read_csv(path1,header=None,encoding="SHIFT-JIS")

    # リスト名の追加
    df1.columns =
["Date_of_operation","Service_No","Sequential_number_in_service","System_code","Day_of_the_
week","Flight_No","Temporary_service_flag","Sequential_number_of_extra_service","Sequential_n
umber_in_the_system","Stop_code","Departure_time_tentative","Vehicle_code_tentative","Onboard
_terminal_code_tentative","System_number","System_name_official","Stop_name_official","Stop_
name_MC","Car_No","Passing_time","Delay_time","Distance","GPS_location_information_latitude
","GPS_location_information_longitude","Corrected_location_information_latitude","Corrected_pos
ition_information_longitude","Company_code","Business_office_code","Registrant","Registered_b
usiness_ID","Registration_date","Registration_time_minute_second","Renewer","Renewal_Busines
s_ID","Renewal_date","Renewal_time_minute_second"]

    # ヘッダー追加
    with open("D:/bus_id/" + str(year_1) + "/" + str(data_1) + ".csv","a") as f:
        writer = csv.writer(f,lineterminator = "\n")
        writer.writerow(("speed_kmh","time_h","distance_km"))

    # グループ化
    for trip_ID,df1_1 in df1.groupby("System_name_official"and"Car_No"):

        # 走行時間
        date_str1 = str(df1_1.iloc[-1]["Passing_time"])
        date_dt1 = datetime.datetime.strptime(date_str1,"%H:%M")

        date_str2 = str(df1_1.iloc[0]["Passing_time"])

```

```

date_dt2 = datetime.datetime.strptime(date_str2,"%H:%M")

time1 = date_dt1 - date_dt2
time = time1.total_seconds() / 3600

# 走行距離
distance = df1_1["Distance"].sum()

# 速度の算出
speed = distance / time

# CSV 出力
with open("D:/bus_id/" + str(year_1) + "/" + str(data_1) + ".csv","a") as f:
    writer = csv.writer(f,lineterminator = "\n")
    writer.writerow((speed,time,distance))

```

バスプローブの流率と密度を算出するコード

```

import pandas as pd
import csv
from pathlib import Path
import glob

# ヘッダー追加
with open("D:/bus_sum.csv","a") as f:
    writer = csv.writer(f,lineterminator = "\n")
    writer.writerow(("q","k"))

# 読み込み
for path1 in glob.glob("D:/bus_id/*/*.csv"):
    new_path1 = Path(path1)
    year_1 = new_path1.parts[-2]
    data_1 = new_path1.stem

```

```

df = pd.read_csv(path1)

# 走行距離の合計（流率）
q = df["distance_km"].sum()

# 走行時間の合計（密度）
k = df["time_h"].sum()

# CSV 出力
with open("D:/bus_sum.csv","a") as f:
    writer = csv.writer(f,lineterminator = "\n")
    writer.writerow((q,k))

```

折れ線回帰モデル式を算出するコード

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import linear_model
import csv

#ファイルの名前4個
a =
["D:/detector_sum.csv","D:/detector_sum_1.csv","D:/detector_sum_2.csv","D:/detector_sum_3.csv"
]

#最大値4個
b = [1091.7, 1623.3, 1624.2, 788.4]

count = 0

# 折曲点の組み合わせ
for path,max1 in zip(a,b):

```

```

data = np.loadtxt(fname=path, delimiter=",", skiprows=1)
for i1 in np.arange(0, max1, 1):
    p1 = np.round(i1, 1)

    for i2 in np.arange(0, max1, 1):
        p2 = np.round(i2, 1)

        if p2 >= p1:

            flow1_flow = []
            flow1_density = []
            flow2_flow = []
            flow2_density = []
            flow3_flow = []
            flow3_density = []

            for row in data:
                y = row[0]
                x = row[1]
                list1 = [y,x]

                if list1[1] <= p1:
                    flow1_flow.append(list1[0])
                    flow1_density.append(list1[1])

                elif list1[1] <= p2:
                    flow2_flow.append(list1[0])
                    flow2_density.append(list1[1])

                else:
                    flow3_flow.append(list1[0])
                    flow3_density.append(list1[1])

            # 自由流
            if not flow1_density:

```

```

pass

else:
    arr1_density = np.array(flow1_density)
    arr1_flow = np.array(flow1_flow)

    X1 = arr1_density.reshape(-1,1)
    Y1 = arr1_flow.reshape(-1,1)

    model = linear_model.LinearRegression()
    model.fit(X1,Y1)

    reg_wn = model.coef_
    r2 = model.score(X1,Y1)

    with open("D:/regression_d/regression_"+ str(count) + ".csv","a") as f:
        writer = csv.writer(f,lineterminator = "\n")
        writer.writerow((p1,p2,reg_wn,r2))

# 混雑流
if not flow2_density:
    pass

else:
    arr2_density = np.array(flow2_density)
    arr2_flow = np.array(flow2_flow)

    X2 = arr2_density.reshape(-1,1)
    Y2 = arr2_flow.reshape(-1,1)

    model = linear_model.LinearRegression()
    model.fit(X2,Y2)

    reg_wn = model.coef_
    r2 = model.score(X2,Y2)

```

```

with open("D:/regression_d/regression_"+ str(count) +".csv","a") as f:
    writer = csv.writer(f,lineterminator = "\n")
    writer.writerow((p1,p2,reg_wn,r2))

# 渋滞流
if not flow3_density:
    pass

else:
    arr3_density = np.array(flow3_density)
    arr3_flow = np.array(flow3_flow)

    X3 = arr3_density.reshape(-1,1)
    Y3 = arr3_flow.reshape(-1,1)

    model = linear_model.LinearRegression()
    model.fit(X3,Y3)

    reg_wn = model.coef_
    r2 = model.score(X3,Y3)

with open("D:/regression_d/regression_"+ str(count) +".csv","a") as f:
    writer = csv.writer(f,lineterminator = "\n")
    writer.writerow((p1,p2,reg_wn,r2))

count+=1

```