

# 博士論文

スマートフォンセンシングの  
適用領域拡大に関する研究

高木 雅

## 内容梗概

情報技術の進歩は社会の有り様を大きく変化させてきた。今や情報通信産業は我が国で最大の市場規模を持つ産業分野であり、その生産額は年間 100 兆円に迫る勢いである。とりわけ、携帯電話網を含めたスマートフォンの進化は、先進国・発展途上国を問わず世界中の人々のライフスタイルに直接的な影響を与えてきた。演算能力の向上や携帯電話網の高速大容量化もさることながら、多彩な機能を取り込み利用シーンを広げたことが、スマートフォンの地位を確固たるものとした。今日のスマートフォンは、通話や SMS、Eメール、Web ブラウザといった携帯電話機の基本機能に加えて、デジタルカメラ、音楽プレーヤー、携帯ゲーム機、非接触 IC 決済カード、生体認証装置としての機能を併せ持つ。このように多彩な機能を手のひらサイズに凝縮したスマートフォンは、まさに情報通信技術の結晶である。

スマートフォンに搭載された多彩かつ高性能なセンサデバイスに着目し、低コストかつ高品質なセンシングを実現するため、多くの研究が行われてきた。とりわけ、ユーザの協力を得て日常生活の中でセンシングを行う「参加型センシング」の概念は画期的であり、商用サービスにも影響を与えた。身近な例では、Google マップに表示される道路や店舗の混雑情報は、数多の Android スマートフォンから収集したユーザの位置情報の統計データに基づくものである。一方で、参加型センシングには、ユーザの行動範囲によってセンシング可能な場所や時間が限定される、パケット通信量やスマートフォンの消費電力の増加のためユーザに敬遠されやすい、といった問題点もあった。

近年になって、電子回路の製造コストが低下し、MVNO が携帯電話回線を非常に安価で提供するようになったことで、大量のセンサノードをインターネットに直結し、クラウド上にセンサデータを集積することが現実的なコストで可能となった。しかし、多くの電子回路にはレアメタルや有害物質が含まれるため、廃棄時にも資源回収と環境汚染対策のための社会的コストが発生する。それゆえ、環境負荷の観点からは、いたずらに機器を増やすより汎用的な機器を繰り返し利用することが望ましい。他方で、スマートフォンは回線契約と一体化した販売形態ゆえに製品寿命に比して買い替えサイクルが短いのが、購入から 2 年が経過してもセンサノードとしては依然として高性能である。従って、中古スマートフォンをセンサノードとして活用できれば、廃棄物の削減という観点では一石二鳥である。しかしながら、スマートフォンは多機能ゆえに消費電力を抑えることが難しく、従来、電池駆動で長期間のセンシング活動を行うことは難しいとされてきた。

これらの背景を踏まえて、本論文では中古スマートフォンを活用した環境に優しいセンシング活動を実現するため、以下の 3 つの技術レイヤで提案を行う。

### 1. ハードウェア

まず、中古スマートフォンに「第二の人生」を与え、センサノードとして活用するための技術として、遠隔操作でのスマートフォンの電源投入を実現する USB ドン

グル型デバイスを提案する．ここでは，精密農業などの用途で必ずしも常時のセンシングが必要でないことに着目し，スマートフォンを間欠動作させることで商用電源のない環境でも長期間のセンシングを可能にする．この手法では，電源オフ状態の Android 端末の USB 端子に所定の電圧を印加すると電源状態管理ソフトウェアが起動することに着目し，マイクロ波無線給電技術を用いて必要な電力を供給する．これにより，家電のリモコンのような遠隔操作でのスマートフォンの電源投入を可能とし，待機電力をゼロに抑えて数ヶ月単位での電池駆動を実現する．

## 2. OS・プラットフォーム

次に，Android 端末を間欠動作させる際の時間的・エネルギー的なオーバーヘッドを削減するため，Android デバイ스에搭載された Linux カーネルを活用してセンシングを行う手法を提案する．本手法では，温度センサやタッチパネルの静電容量センサだけでなく，Bluetooth や USB 通信をサポートし，センサノード等の外部機器との連携も可能とした．また，本手法を適用した場合でも Android OS の機能は完全に温存でき，必要に応じて Linux カーネルと Android OS を選択的に起動できる．

## 3. アプリケーション

さらに，音に着目したセンシング技術のアプリケーションを2つ紹介する．

1つ目は，交通事故から歩行者を守るための接近車両検知技術である．ここでは，走行時の高い静粛性ゆえに事故が多発している電気自動車およびハイブリッド車に焦点を当て，走行用モータが発するスイッチング雑音を機械学習で検出することで，環境雑音や車種の違いにロバストな車両検知を実現する．

2つ目は，マルチコプタの積載重量を飛行音から推定する技術である．マルチコプタは，急速に低価格化が進み幅広い用途での活躍が期待されるが，我々の生活空間を縦横無尽に飛び回るため，周囲の人と衝突する危険性がある．そこで，ロータの回転に起因する飛行音から機械学習で積載重量を推定し，積載重量が多く危険性の高いマルチコプタを選択的に検知する．

いずれの手法も，インフラとして一切特別な設備を必要とせず，ユーザが所持するスマートフォンだけで処理が完結するため，ユーザが自衛のために今すぐ利用できる点が特徴的である．ただし，これらのセンシング技術は少なからずスマートフォンの消費電力を増加させるため，センシング専用の中古スマートフォンとユーザの手元のスマートフォンで役割分担を行うことが望ましい．

なお本論文では，現時点で普及台数が最も多く低価格化が進んだスマートデバイスの代表として「スマートフォン」という言葉を用いるが，本論文で提案する技術は，将来スマートフォンに替わる新たなスマートデバイスが登場した場合にも応用が可能と考えられる．

# 目次

---

第1章	序論	1
1.1	本研究の背景	2
1.2	関連技術と先行研究	3
1.2.1	参加型センシング	3
1.2.2	スマートフォンの省電力化	4
1.2.3	携帯電話網とLPWA	5
1.2.4	センサノード	7
1.3	本論文の構成	8
第2章	マイクロ波給電を応用した遠隔操作での電源投入	9
2.1	まえがき	10
2.2	要素技術	11
2.2.1	USB給電による起動トリガ	12
2.2.2	マイクロ波給電による無線起動トリガ	13
2.2.3	マイクロ波給電による選択的な無線起動トリガ	14
2.2.4	マイクロ波給電のコントローラ	16
2.2.5	イベントドリブンでの起動トリガ	17
2.2.6	エナジーハーベスティングモジュール	18
2.2.7	Androidアプリの高速起動	18
2.3	応用例	20
2.3.1	乳幼児や高齢者の遠隔見守り	20
2.3.2	インタラクティブ広告	22
2.3.3	ジオフェンス	23
2.3.4	防災(1): がけ崩れの検知	24
2.3.5	防災(2): 有毒火山ガスの検知	25
2.4	検討事項	26
2.4.1	スマートフォンの機種依存	26
2.4.2	起動トリガの無線信号の到達距離	26
2.4.3	中古スマートフォンの電池容量	27
2.4.4	既存の省電力モードとの消費電力比較	29
2.4.5	セキュリティの確保	30
2.5	関連研究	31
2.6	あとがき	32



<b>第 3 章</b>	<b>Linux カーネルを活用した起動時間の短縮</b>	<b>33</b>
3.1	まえがき	34
3.2	Android OS の構造	35
3.2.1	Linux カーネル	35
3.2.2	ネイティブライブラリ	36
3.2.3	Java 仮想マシン	37
3.2.4	アプリケーション・フレームワーク	37
3.2.5	システムアプリ	37
3.2.6	メンテナンス用ユーティリティ	39
3.3	要素技術	40
3.3.1	起動モードセクタ	41
3.3.2	ディスプレイの制御	43
3.3.3	Bluetooth Low Energy 通信	44
3.3.4	USB インタフェース	44
3.3.5	ボタン入力	44
3.4	消費電力の計測	45
3.5	電池寿命の推定	45
3.6	検討事項	47
3.6.1	スマートフォンの機種依存	48
3.6.2	フレームワークの導入コスト	48
3.6.3	さらなる省電力化	49
3.7	関連研究	49
3.8	あとがき	50
<b>第 4 章</b>	<b>音に着目した機械学習によるセンシング技術</b>	<b>51</b>
4.1	まえがき	52
4.2	電気自動車の接近検知による交通事故の予防	53
4.2.1	背景	53
4.2.2	電気自動車の走行音	55
4.2.3	車両検知システム	58
4.2.4	性能評価	61
4.2.5	検討事項	64
4.2.6	実装	68
4.2.7	関連研究	69
4.2.8	まとめ	70
4.3	マルチコプタの積載重量の推定	72
4.3.1	背景	72
4.3.2	マルチコプタの飛行音	74
4.3.3	積載重量の推定	77
4.3.4	性能評価	79
4.3.5	検討事項	80
4.3.6	関連研究	82

4.3.7	まとめ	83
4.4	マイクの周波数特性に関する機種依存の影響評価	84
4.4.1	背景	84
4.4.2	周波数特性の計測	84
4.4.3	Android 端末 100 機種 of 周波数特性	86
4.4.4	センシング精度への影響評価	91
4.4.5	まとめ	94
4.5	あとがき	96
<b>第 5 章</b>	<b>結論</b>	<b>97</b>
5.1	本論文のまとめ	98
5.2	今後の検討課題	99
5.2.1	スマートフォンの機種依存	99
5.2.2	中古スマートフォンならではのアプリケーション開拓	99
	謝辞	100

# 目次

1.1	我が国の主な産業の市場規模（名目国内生産額）の推移	2
2.1	提案するフレームワークの概要	11
2.2	無線起動トリガのためのレクテナ USB ドングル	13
2.3	(a) マイクロ波給電源 (Powercast TX91501)	
	(b) 到達距離の計測環境	
	(c) 送電距離と起動トリガの所要時間の関係	14
2.4	選択的起動トリガを実現するオンオフ変調方式	15
2.5	選択的起動トリガを実現するための拡張モジュールの回路図	16
2.6	選択的起動を行うためのオンオフ変調のタイミング・ダイアグラム	17
2.7	イベントドリブン方式の USB ドングルの回路図	18
2.8	WakeDroid が中古スマートフォンの性能を引き出す 4 つの典型的な利用シナリオ	20
2.9	(a) 遠隔みまもりのデモ実験の配置図	
	(b) スチール棚の上に設置した見守りシステムのプロトタイプ	21
2.10	(a) 文具店を想定したインタラクティブ広告のプロトタイプ	
	(b) 置換されたブートロゴ	
	(c) 置換されたブートアニメーション	
	(d) 起動プロセスの完了後は、あらかじめ指定した Android アプリが画面表示を引き継ぐ	22
2.11	(a) 家畜管理システムの概要	
	(b) 家畜管理システムのプロトタイプ	23
2.12	(a) 低コストな土壌水分センサを拡張モジュールに採用した	
	(b) 崖に設置されたがけ崩れ検知のプロトタイプ	
	(c) 人工的に発生させた小規模ながけ崩れの検知に成功した	
	(d) システムは周囲の写真を撮影し、管理者に送信する	24
2.13	有毒ガス検知システムのプロトタイプ	25
2.14	中古スマートフォンの内蔵バッテリーの放電電圧の時間変化	28
3.1	提案コンセプト	34
3.2	提案手法と通常の Android OS で利用可能なスマートフォンの機能構成	35
3.3	Android OS のアーキテクチャ概要	36
3.4	Nexus 5 上で Linux カーネルのみを起動した場合の消費電力の推移	46
3.5	Nexus 5 上で Android OS を起動した場合の消費電力の推移	46
3.6	Nexus 5 を充電する際のロジックボードの消費電力	46

3.7	マイクロ波照射によるスマートフォンの起動間隔と移動履歴の送信間隔を 変化させた時の電池寿命の比較	47
4.1	提案するシステムのコンセプト	53
4.2	ガソリン車とハイブリッド車の騒音レベルの比較	55
4.3	計測環境の風景	56
4.4	電気自動車やハイブリッド車の走行音のスペクトログラム	57
4.5	モータの回転速度と各周波数における音圧レベルの関係	58
4.6	提案する車両検知システムのフローチャート	59
4.7	録音データから学習用データセットを切り出す手順	60
4.8	車両が最接近するまでに1度でも車両を正しく検知できた確率	61
4.9	車両が最接近するまでに1度でも車両を正しく検知できた確率	62
4.10	車両が最接近するまでに1度でも車両を正しく検知できた確率	63
4.11	複数の車両が同時に接近するシナリオでの検知結果	65
4.12	Android アプリケーションとして実装した我々の車両検知システムのデモ インストレーションの様子	68
4.13	提案するシステムのコンセプト	72
4.14	マルチコプタの飛行と姿勢制御のメカニズム	74
4.15	マルチコプタへの貨物の積載方法	75
4.16	マルチコプタの飛行音のスペクトログラム	76
4.17	提案する積載重量推定システムのフローチャート	78
4.18	周波数特性の計測環境と配置図	84
4.19	スピーカー GENELEC 1029A の周波数特性（データシート [92] から抜粋）	86
4.20	iPhone 5s で録音したホワイトノイズのスペクトル分布と包絡線	86
4.21	iPhone 5s と 100 機種 of Android 端末で録音したホワイトノイズのスペク トル分布と包絡線（端末番号#01～#25）	87
4.22	iPhone 5s と 100 機種 of Android 端末で録音したホワイトノイズのスペク トル分布と包絡線（端末番号#26～#50）	88
4.23	iPhone 5s と Android 端末 100 機種で録音したホワイトノイズのスペクト ル分布と包絡線（端末番号#51～#75）	89
4.24	iPhone 5s と Android 端末 100 機種で録音したホワイトノイズのスペクト ル分布と包絡線（端末番号#76～#100）	90
4.25	ゲイン調整後のスペクトル分布の iPhone 5s との差分（端末番号#01～#50）	92
4.26	ゲイン調整後のスペクトル分布の iPhone 5s との差分（端末番号#51～ #100）	93
4.27	各スマートフォンの録音データに相当する擬似的な入力でのマルチコプタ 積載重量の推定精度の度数分布	94

# 目 次

1.1	主要な LPWA の通信規格（セルラー系）	6
1.2	主要な LPWA の通信規格（非セルラー系）	7
2.1	Android OS のリファレンスモデル 7 機種における，USB 端子からの起動トリガに必要な電圧とエネルギー量	12
2.2	各機種で認識可能な最短のオン/オフのシンボルの長さ	15
2.3	Android OS の種々のディストリビューションを実行する Nexus 5 における，電源投入からセンシングアプリ起動までの所要時間の比較	19
2.4	中古スマートフォンの電池容量	29
2.5	Android スマートフォン向けのセキュリティパッチの配信状況	31
3.1	Java 言語と C 言語でのベンチマークプログラムの処理速度の比較	38
3.2	Android OS における特権パーミッションの例	39
3.3	Nexus 5 用の Android 5.1.0 (build LMY47I) のファクトリーイメージを構成するファイル群	41
3.4	Android 5.1.0_r1.0 のソースコード内の <code>init.rc</code> で定義されているイベントトリガの一覧	42
4.1	2016 年の新車販売台数の上位 30 車種のうち，25 車種で EV/PHV/HV モデルが提供されている	54
4.2	理想的な環境下で日産リーフを異なる速度で走行させた場合の検知精度	61
4.3	3 つの計測環境で日産リーフを走行させた場合の検知精度	62
4.4	理想的な環境下で 4 車種を走行させた場合の検知精度	63
4.5	学習データ数が誤検知率と見落とし率に与える影響の評価	66
4.6	録音データの窓幅と検知精度の関係	66
4.7	車両接近判定ロジックの実行間隔を変化させた時の Nexus 5 の消費電力	67
4.8	Phantom 2 Vision+ のロータの回転数	75
4.9	積載重量から予想される基本周波数と実際の基本周波数	77
4.10	k-NN でのマルチコプタの積載重量の推定精度	79
4.11	k 近傍分類における近傍標本数 $k$ を変化させた時の判定精度	80
4.12	サポートベクタマシンでのマルチコプタの積載重量の推定精度	80
4.13	ランダムフォレストでのマルチコプタの積載重量の推定精度	81
4.14	J48 決定木でのマルチコプタの積載重量の推定精度	81
4.15	CART 決定木でのマルチコプタの積載重量の推定精度	81
4.16	計測対象のスマートフォン	85

4.17 各スマートフォンの録音データに相当する擬似的な入力でのマルチコプタ 積載重量の推定精度と iPhone 5s との差分 . . . . .	95
---	----

# ■ 第1章

---

序論

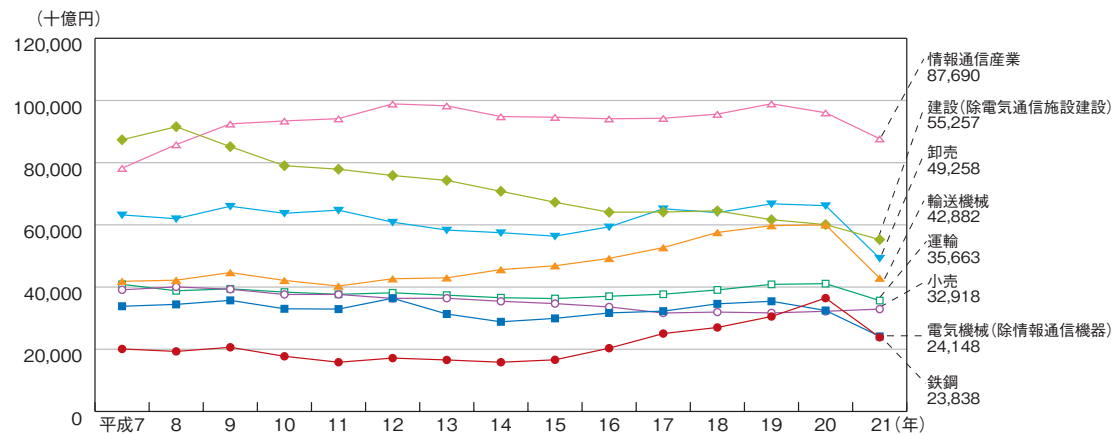


図 1.1: 我が国の主な産業の市場規模（名目国内生産額）の推移 [1]

## 1.1 本研究の背景

情報技術の進歩は社会の有り様を大きく変化させてきた。今や情報通信産業は我が国で最大の市場規模を持つ産業分野であり、その生産額は年間 100 兆円に迫る勢いである（図 1.1）。とりわけ、携帯電話網を含めたスマートフォンの進化は、先進国・発展途上国を問わず世界中の人々のライフスタイルに直接的な影響を与えてきた。演算能力の向上や携帯電話網の高速大容量化もさることながら、多彩な機能を取り込み利用シーンを広げたことが、スマートフォンの地位を確固たるものとした。今日のスマートフォンは、通話や SMS、Eメール、Web ブラウザといった携帯電話機の基本機能に加えて、デジタルカメラ、音楽プレイヤー、携帯ゲーム機、非接触 IC 決済カード、生体認証装置としての機能を併せ持つ。このように多彩な機能を手のひらサイズに凝縮したスマートフォンは、まさに情報通信技術の結晶である。

スマートフォンに搭載された多彩かつ高性能なセンサデバイスに着目し、低コストかつ高品質なセンシングを実現するため、多くの研究が行われてきた。とりわけ、ユーザの協力を得て日常生活の中でセンシングを行う「参加型センシング」の概念は画期的であり、商用サービスにも影響を与えた。身近な例では、Google マップに表示される道路や店舗の混雑情報は、数多の Android スマートフォンから収集したユーザの位置情報の統計データに基くものである。一方で、参加型センシングには、ユーザの行動範囲によってセンシング可能な場所や時間が限定される、パケット通信量やスマートフォンの消費電力の増加のためユーザに敬遠されやすい、といった問題点もあった。

近年になって、電子回路の製造コストが低下し、MVNO が携帯電話回線を非常に安価で提供するようになったことで、大量のセンサノードをインターネットに直結し、クラウド上にセンサデータを集積することが現実的なコストで可能となった。しかし、多くの電子回路にはレアメタルや有害物質が含まれるため、廃棄時にも資源回収と環境汚染対策のための社会的コストが発生する。それゆえ、環境負荷の観点からは、いたずらに機器を増やすより汎用的な機器を繰り返し利用することが望ましい。他方で、スマートフォンは回線契約と一体化した販売形態ゆえに製品寿命に比して買い替えサイクルが短い。購入から 2 年が経過してもセンサノードとしては依然として高性能である。従って、中古スマー



トフォンをセンサノードとして活用できれば、廃棄物の削減という観点では一石二鳥である。しかしながら、スマートフォンは多機能ゆえに消費電力を抑えることが難しく、従来、電池駆動で長期間のセンシング活動を行うことは難しいとされてきた。

## 1.2 関連技術と先行研究

本論に入る前に、本研究での提案内容に関係する技術および先行研究についてまとめる。ここで紹介する先行研究の中には、本稿での提案内容と相補的な関係にあるものも含まれており、我々の手法と組み合わせることでスマートフォンの機能性をさらに高めたり、電池駆動時間を伸ばしたりできる可能性がある。

### 1.2.1 参加型センシング

参加型センシング (Participatory Sensing) とは、人々が日常生活の中で利用しているモバイル機器をセンサノードとして活用し、低コストかつ高品質なセンシングを実現する技術である。ここでは、モバイル機器の持つ演算能力や通信帯域といった余剰能力に着目し、日常生活の中で参加ユーザの負担にならない範囲でセンシングを行う。ユーザに求められる明示的な負担としては、指定された場所への移動や指定されたタスク (写真撮影など) の実行が挙げられ、暗黙的な負担としては、モバイル機器の消費電力やパケット通信量の増加などが挙げられる。近年は、モバイル機器としてスマートフォンを用いる「スマートフォンセンシング」が主流であるが、もともとは、従来型の携帯電話機や PDA を想定して提案されたものであった。スマートフォンが主流となった背景には、アプリケーションの開発や配布が容易であることに加え、演算能力が高いこと、多彩なセンサを搭載すること、インターネットに常時接続すること、などセンサノードとしての優れた特徴がある。なお、Burke らが参加型センシングのコンセプトを初めて提唱したのは 2006 年のことであり、iOS や Android OS を搭載した近代的なスマートフォンが登場する前年のことであった [2]。

**フレームワーク** スマートフォンが登場して参加型センシングに注目が集まると、アプリケーション実装のためのフレームワークが多く提案されるようになった [3, 4, 5]。2010 年には、センシング品質と消費電力のバランスを取るためのフレームワーク Jigsaw が提案された [6]。Jigsaw は、マイクや加速度センサの入力状況が芳しくない場合に信号処理を簡略化したり、移動状況やユーザの行動パターンに基づいて GPS による位置測位を間引いたりすることで、センシング品質を保ちつつ消費電力の削減を図る。2012 年には、アプリケーション実装を補助するだけでなく、センシング活動の主催者と参加者の間を取り持ちセンシングタスクの分配を補助するフレームワーク Medusa が提案された [7]。Medusa では、センシングタスクを XML 形式で簡潔に記述できるため、主催者はアプリ開発に関する専門的な知識を持つ必要がない。2013 年には、所望のセンサや通信モジュールが他のアプリによって利用されるタイミングに合わせてセンシングやデータ送信を行うことで、スマートフォンの消費電力を抑えるフレームワーク Piggyback CrowdSensing (PCS) が提案された [8]。特に、LTE/3G 通信モジュールに関しては、通信終了後に消費電力の高い状

態 (Radio Tail) が続くことが知られており, 利用機会を集約することで大幅な省電力化を実現できる.

**プライバシーの配慮** 参加型センシングでは, センサノードが常にユーザに密着した状態でセンシングを行うため, ユーザのプライバシーが侵害される懸念がある. この点については, 参加型センシングが提唱された当時から問題視されており, センサデータに意図的にノイズを混入させる, 複数ユーザのデータから算出した統計値のみを報告する, といった手法が提案されてきた [9, 10]. 2015年には, 悪意あるユーザが意図的に誤ったデータを報告した場合に, センサデータの時間的・空間的な相関を調べることで, 真のセンサ値と同時に各デバイスの信頼度を推定する手法が提案された [11]. また, この手法に準同型暗号を組み合わせることで, センサデータを暗号化したまま統計処理する手法も提案された [12].

**インセンティブ設計** 他方で, ユーザの協力を取り付けるための適切なインセンティブ設計は, 現在も重要な研究課題として残されている. 要求する報酬金額が最も少ないユーザにセンシングタスクを割り当てる逆オークション方式は多く提案されているが, ユーザ間の競争により報酬が低下すると, 参加型センシング自体のユーザ数が減少してしまうという問題がある [13, 14]. また, 近年はゲーム理論を応用したインセンティブ体系も提案されているが, 広く使用されるには至っていない [15].

### 1.2.2 スマートフォンの省電力化

スマートフォンは従来の携帯電話機より高機能である反面, 消費電力が多く電池駆動時間が短いことが知られている. Carroll らの報告によると, スマートフォンの消費電力の大部分は液晶ディスプレイと通信モジュールによるものであり, CPU や RAM の消費電力は約 1 割に過ぎない [16]. このうち, 液晶ディスプレイの消費電力はバックライトが大部分を占めており, 液晶素子の透過率向上 [17] や有機 EL ディスプレイへの換装など抜本的な対策を除くと削減の余地は少ない. 一方, 通信モジュールについては, CPU や GPU と一体化した SoC (System on Chip) の一機能として実装されており, 最大手ベンダである Qualcomm によると SoC 全体の定格電力は減少傾向にある. 改善の余地が比較的大きい通信モジュールについては, 多くの省電力化手法が提案されてきた. この分野の先行研究は, 通信タイミングの制御手法と通信パラメータの最適化手法, 低電力な位置測位手法の 3 種類に大別される.

**通信タイミングの制御** この分野では, 遅延耐性が高くユーザへの影響が少ないバックグラウンド通信や画面消灯中の通信に着目し, 大胆なスケジューリングを行う提案が支配的であった [18, 19]. これに対して, 同一の基地局に接続しているスマートフォン同士を自律分散的に協調させ, 平均スループットの向上と消費電力の削減を同時に実現する LoadSense は画期的であった [20]. この手法では, 基地局からの信号の S/N 比を手掛かりとして基地局の負荷状況をオーバーヘッドなく推定した上で, 各スマートフォンの通信タイミングがなるべく重ならないようにバックグラウンド通信のスケジューリングを行う. これにより, 各スマートフォンにおける平均スループットが向上し, 通信時間が短くなる分だけ消費電

力の低減に繋がる。この手法は、対応端末と非対応端末が混在しても、効果を発揮できるのが特徴的であった。

**通信パラメータの最適化** この分野では、携帯電話網の無線アクセス区間における接続管理ロジックに着目した手法が多く提案されてきた [21, 22, 23]。LTE/3G 網では無線アクセス区間の接続確立・切断時のオーバーヘッドが大きいため、一連の通信が完了した後も一定時間は後続トラフィックに備えて接続を維持する。この間、LTE/3G 通信モジュールは消費電力の高い状態 (Radio Tail) のままとなる。これに対し、トラフィックの発生源である各アプリは自身が行う通信の開始タイミングや終了タイミングを正確に予測できることに着目し、アプリが OS に対して通信タイミングを通知する仕組みを構築することで、Radio Tail 部分の消費電力を抑える手法として TOP が提案された [21]。ただし、この手法はアプリ開発者による対応が必要であり、既存のアプリ資産を活用できないという問題がある。また、この問題点を解決する手法として、アプリからのシステムコールの発行パターンから通信タイミングを予測し、Radio Tail 長を調整する RadioJockey も提案された [23]。

**位置測位** GPS を用いた位置測位は消費電力が多いことが知られており、位置情報を頻繁に取得する場合には GPS モジュールの消費電力が無視できないほど大きくなる。一方で、携帯電話の基地局情報を用いる方式は測位精度が数百 m~数 km とごく粗いため、より正確な位置情報が必要な用途には適さない。近年では、Wi-Fi アクセスポイントの SSID と電波強度のリストから位置情報を推定する手法も商用化されているが、モバイルルータなど移動式のアクセスポイントが学習データに含まれると誤作動を起こす可能性がある。この問題に対し、消費電力を抑えつつ高精度な測位を実現する手法が提案されてきた [24, 25, 26]。これらの手法では、GPS 測位における信号処理をパイプラインに分割し、最も計算負荷の高い部分をクラウドサーバにオフロードすることで、消費電力を最大 80% 削減できる。

### 1.2.3 携帯電話網と LPWA

あらゆる場所でのインターネット接続を可能とする携帯電話網は、センシング活動を支える重要な技術の 1 つである。2010 年代前半まで、携帯電話網は高速・大容量化を目指して進化を続けてきたが、IoT 技術が注目を集め M2M 通信の需要が高まる中で、低遅延・高信頼・省電力・低コスト・超多元接続など多様な目標が設定されるようになった。他方で、アンライセンスバンドを用いて低消費電力・広域カバレッジ・低ビットレートの通信を実現する LPWA (Low Power Wide Area) と呼ばれる新しい通信規格が登場し注目を集めている。

**携帯電話網** 総務省の調査報告によると、日本における携帯電話網の人口カバー率は 2013 年 11 月現在で 99.97% であった [27]。これほど広域なカバレッジ持つ商用の無線アクセスサービスは、後述する LPWA が登場するまで衛星電話を除いて他に存在しなかった。また、衛星電話はデータ通信が非常に低速で利用料金が高額であることから、野外における

表 1.1: 主要な LPWA の通信規格 (セルラー系)

通信規格	周波数帯	帯域幅	通信速度	カバレッジ
LTE Cat.M	LTE 周波数帯	1.4 MHz 幅	1 Mbps	~100 km
NB-IoT	LTE 周波数帯	180 kHz 幅	250 kbps	~40 km
EC-GSM-IoT	LTE/GSM 周波数帯	200 kHz 幅	240 kbps	~40 km

センシング活動では携帯電話網が重用されてきた。特に近年は、MVNO の台頭によりパケット単価が低下し大量調達も容易となったため、幅広い分野で盛んに利用されている。

携帯電話網は、約 10 年の周期で新たな多元接続方式を導入して世代交代を繰り返してきた。1980 年代に登場した第 1 世代携帯電話 (1G) は、アナログ変調方式で音声通話機能のみを提供し、FDMA 方式で多元接続を実現していた。一方、1990 年代に登場した 2G では、デジタル変調を採用したことで 28.8 kbps (PDC の場合) のパケット通信が可能となり、TDMA 方式で多元接続を実現していた。日本で採用された PDC は既に停波しているが、海外で採用された GSM は現在でも広く利用されている。2000 年代になると、多元接続に CDMA 方式を採用した 3G が登場し、インターネット接続を念頭にパケット通信の高速化が図られた。その後も改良が続けられ、W-CDMA は当初の 384 kbps から最終的に 42 Mbps まで、CDMA2000 は当初の 144 kbps から最終的に 9.2 Mbps まで高速化された。2010 年代には、多元接続に OFDMA 方式を採用した 4G が登場し、回線交換方式による音声通話を廃止するなどの効率化により、パケット通信のさらなる高速化が図られた。複数の周波数帯を束ねるキャリアアグリゲーション技術や 3.5 GHz 帯の導入により、最大スループットは 788 Mbps に達した。2020 年に向けて仕様策定が進む 5G では、多元接続に NOMA 方式を導入し、Massive MIMO アンテナを活用した事実上の空間多重化により、Gbps クラスのスループットを実現する。

**LPWA (セルラー系)** 携帯電話網の広域なカバレッジを活かしつつ、M2M 用途の特殊な需要を満たすため、既存の携帯電話網向け通信規格をベースとした LPWA 規格が 3GPP により提案されている (表 1.1)。LTE Cat.M と NB-IoT は、いずれも 4G の LTE をベースとした通信規格であり、基地局からのページング受信間隔拡大により省電力化を、周波数帯域幅の縮小と変調方式の単純化により通信モジュールの低コスト化を、同一信号の複数回送信と変調方式の単純化によりカバレッジの拡大を実現する。両規格とも、既存の LTE 周波数帯を使用し、LTE 通信の一部タイムスロットやガードバンドを活用して共存する。なお、LTE Cat.M には移動耐性があり、NB-IoT には移動耐性がない、という違いがある。一方、EC-GSM-IoT は、2G の GSM をベースとした通信規格であり、基地局からのページング受信間隔拡大により省電力化を図る。いずれの通信規格も、通信モジュールを乾電池で数年間の運用可能とすることを目標としている。

仕様策定中の 5G においても、M2M 用途を見据えて、低遅延・高信頼・超大量接続・低コスト・省電力といった目標が定められており、LPWA に相当する通信規格の登場が予想される。5G 版 LPWA では、NOMA 方式の特性により送信パケットの衝突が許容されるため、非セルラー系 LPWA の課題である連続送信に関する制約が解消する可能性がある。

表 1.2: 主要な LPWA の通信規格 (非セルラー系)

通信規格	周波数帯	帯域幅	通信速度	カバレッジ
Wi-SUN	915 MHz/2.4 GHz 帯	0.2~1.2 MHz 幅	800 kbps	~1 km
LoRaWAN	915 MHz 帯	125~500 kHz 幅	20 bps	~10 km
SIGFOX	915 MHz 帯	100 kHz 幅	100 bps	~50 km
Sony's LPWA	915 MHz 帯	200 kHz 幅	80 bps	~270 km

**LPWA (非セルラー系)** 他方では、携帯電話網向け通信規格をベースとせず、アンライセンスバンドを活用する LPWA 規格が、携帯電話業界外から提案されている (表 1.2)。カバレッジ拡大のために送信出力を高めた場合、日本では電波法により連続通信が規制されるため、カバレッジ半径 10 km 以上の通信規格の実効スループットは最大でも 100 bps に留まる。

Wi-SUN は NICT 主導で開発されたスマートメータ向けの通信規格であり、物理層は IEEE 802.15.4g として標準化されている。LPWA 規格としては高速な通信が可能であるが、カバレッジ半径が最大 1 km と狭いのが難点である。一方、LoRaWAN や SIGFOX はカバレッジ半径が最大 10 km~50 km と広い反面、パケットサイズが 10 バイト前後と非常に小さく利用シーンが厳しく限定される。また、Sony's LPWA は見通し 100 km 以上のカバレッジを持ち、時速 100 km での高速移動中にも安定して通信できる利点を持つが、ダウンリンクなしの片方向通信という制約が加わる。

4 章で提案するような機械学習を用いたセンシング技術では、最新車種や最新機種に対応するために定期的な分類器の配信が必要となるが、ファイル受信に 1 MB あたり丸 1 日以上も掛かる LPWA 回線は実用的とはいえない。

#### 1.2.4 センサノード

スマートフォンなどのモバイル機器に注目が集まる以前から、専用ハードウェアを用いたセンシング技術が広く研究されてきた。近年では、Raspberry Pi や BeagleBoard といったシングルボードコンピュータや Arduino マイコンが普及したことで、誰でも手軽にセンサノードを実装できるようになったが、以前はセンサノード専用のハードウェアが主流であった。このようなセンサノードは、今日でも多くの研究に活用されている。

**MICA シリーズ** MICA は Crossbow Technology によって開発された無線センサネットワークのテストベッド端末であり、無線センサネットワークに関する研究分野ではデファクトスタンダードとして広く活用されている [28]。MICA, MICA2, MICAz と順次改良が加えられ、後に IRIS Mote や NeoMote に発展した。MICAz は、MPU に ATmega128L を採用しており、Tiny OS で動作し、ZigBee 通信を利用できる。

**TelosB** Telos-B はオープンソースの無線センサネットワーク端末であり、MICAz に次いで広く利用されている [29]。MPU として省電力マイコン MSP430 を採用しており、消費電力は MICAz の 75 mW に対して、50 mW と低消費電力である。MICAz と同じく

Tiny OS で動作し、ZigBee 通信を利用できる。なお、製造は Crossbow Technology が担当している。

いずれも単三乾電池2本で駆動できるほど省電力である反面、画像や音声を取り扱うには処理能力が不足しており、信号処理用のハードウェアが別途必要となる。実際に、MICAz を利用して銃撃音と弾丸通過時の衝撃波から狙撃手の位置を割り出す研究事例では、音声信号を処理するために別途 FPGA 回路を搭載している [30]。

### 1.3 本論文の構成

本論文は以下の5つの章により構成される。

- 第1章 序論
- 第2章 マイクロ波給電を応用した遠隔操作での電源投入
- 第3章 Linux カーネルを活用した起動時間の短縮
- 第4章 音に着目した機械学習によるセンシング技術
- 第5章 結論

第2章では、中古スマートフォンに「第二の人生」を与え、センサノードとして活用するための技術として、遠隔操作でのスマートフォンの電源投入を実現する USB ドングル型デバイスを提案する。ここでは、精密農業などの用途で必ずしも常時のセンシングが必要でないことに着目し、スマートフォンを間欠動作させることで商用電源のない環境でも長期間のセンシングを可能にする。

第3章では、Android 端末を間欠動作させる際の時間的・エネルギー的なオーバーヘッドを削減するため、Android デバイスに搭載された Linux カーネルを活用してセンシングを行う手法を提案する。本手法を適用した場合でも Android OS の機能は完全に温存でき、必要に応じて Linux カーネルと Android OS を選択的に起動できる。

第4章では、音に着目したセンシングアプリケーションとして、交通事故から歩行者を守るための接近車両検知技術とマルチコプタの積載重量を飛行音から推定する技術の2つを提案する。これらの手法は、インフラとして一切特別な設備を必要とせず、ユーザが所持するスマートフォンだけで処理が完結するため、ユーザが自衛のために今すぐ利用できる点が特徴的である。

最後に、第5章で本論文の内容をまとめる。

## ■ 第2章

---

マイクロ波給電を応用した遠隔操作での電源投入

## 2.1 まえがき

スマートフォンは我々の生活に欠かせない便利な存在であるが、携帯電話の回線契約と一体化した販売形態ゆえに、製品寿命に比して買い替えサイクルが短い。2012年以降、全世界で累計60億台ものスマートフォンが出荷されており、買い替えサイクルを平均2年と仮定すると、30億台以上のスマートフォンが既に廃棄されたか、または、机の引き出しの中に眠っている計算になる。1台あたり100g強のスマートフォンであっても、積み上げれば30万トン以上の廃棄物となり、6,000トン以上のレアメタルが含まれる。他方で、Raspberry Piの累計出荷台数は2016年に1,000万台に到達したが、Raspberry Piの製造にもレアメタルが必要であり、Raspberry Pi自体も将来のいずれかの時点で廃棄物となる。このような環境負荷を考慮するならば、いたずらに新たなデバイスを製造するよりは、中古スマートフォンなどの既存機器を繰り返し利用の方が望ましい。

低価格でロバストなセンサノードはユビキタスコンピューティングの実現に不可欠である。Telosのような専用センサノードは廉価であるが、機能や性能に厳しい制約があり信号処理などの用途には適さない[29]。一般に、電子機器の調達コストを低減するにあたり最も有効な手段は、大量生産された汎用的な製品を使うことであり、スマートフォンはこのような汎用製品の代表格である。ほとんどのスマートフォンは、カメラやマイク、加速度センサなどの高品質なセンサを搭載しており、防水筐体を備えるものも珍しくない。Wi-FiとBluetooth, GPS, 3G通信など最低限の機能を備えたスマートフォンの小売価格は、定期契約なしでも10ドル台まで低下している。たとえば、スマートフォンとしてはCPU性能が低く、RAMやストレージの容量が少ないとしても、無線センサノードとしては十分に高性能である。携帯電話回線の月額料金についても低廉化が進んでおり、米国のある携帯電話事業者は月額3ドルで30分の無料通話と30通の無料SMSを提供している。また、極端な例では、月額料金なしで500MBのパケット通信を提供する事業者も存在する。さらに、IoT機器に特化した事業者も出現しており、ごく低速ながら低価格で通信可能なSIMカードを大量調達できる。

これまでも、スマートフォンを様々なセンサデータを集約するためのハブとして利用する検討が数多くなされてきた[31, 32]。高度に発達した携帯電話網のおかげで、スマートフォンを用いたセンシングシステムを農村地域や森林地帯に導入することが理論上は可能であるが、電池駆動時間の厳しい制約がネックとなり実用性は乏しい。従来のスマートフォンセンシングに関する提案では、対象のスマートフォンがユーザによって毎日充電されることを暗黙の前提としてきた。しかしながら、中古スマートフォンをセンシング専用機器として利用する場合には、充電や電池交換のコストは決して無視できず、SIMカードや端末自体の調達コストを上回る恐れすらある。従って、スマートフォンの電池駆動時間の制約を解消することは、商用電源を利用できない野外でのセンシング活動の新しい可能性を開く重要な鍵となる。

本章では、特に商用電源の利用しにくい環境において専用センサノードとして稼働するスマートフォン向けに、外付け電源管理システム「WakeDroid」を提案する。WakeDroidは、Androidスマートフォンを充電器に接続した際に自動的に起動する電源状態管理ソフトウェアを拡張し、USB端子に接続するUSB Dongle型デバイスとして電源管理機能を実現する。WakeDroidのUSB Dongleには2種類あり、一方は、無線電力伝送のためのレクテナを備え、遠隔操作によるAndroidスマートフォンの電源投入を実現する。他方は、



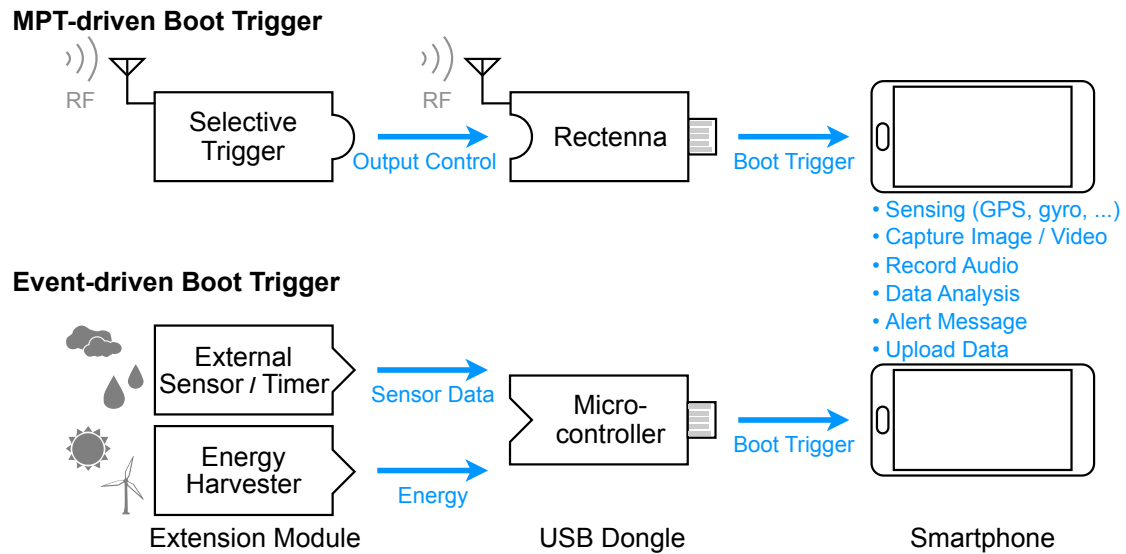


図 2.1: 提案するフレームワークの概要。WakeDroid は 2 種類の USB ドングルを用いてスマートフォンの起動プロセスを開始させる。一方は、マイクロ波無線給電技術を応用したもので、外部から無線で起動指令を受け取る。他方は、内蔵するマイコンがスマートフォンを起動するタイミングを判断する。

乾電池とマイコン、センサを搭載し、そのセンサ値や電池残量に応じて、スマートフォンの起動間隔を調整する。このようにオンデマンドでのスマートフォンの電源投入を実現することで、スマートフォンを電源オフ状態で待機させることを可能とし、センシング活動における電池駆動時間の厳しい制約を大きく緩和する。また、WakeDroid は、中古スマートフォンに低コストで環境にやさしいセンサノードとしての「第二の人生」を与え、草の根的なセンシング活動を支援する一助となる。

本章の構成は以下の通りである。まず 2.2 節で、WakeDroid を支える 7 つの要素技術を紹介し、次に 2.3 節で、4 つの典型的な応用例を交えて適用対象の広さを実証する。さらに 2.4 節で、スマートフォンの機種依存や無線信号の飛距離について議論した後、2.5 節で関連研究を紹介し、2.6 節で本章の内容をまとめる。

## 2.2 要素技術

図 2.1 に WakeDroid の概要を示す。WakeDroid は、対象のスマートフォン、USB ドングル、拡張モジュールの 3 つの要素から構成される。USB ドングルとしては、マイクロ波無線給電方式 (MPT-Driven) とイベントドリブン方式 (Event-Driven) の 2 種類があり、前者の拡張モジュールとして選択的起動モジュールが、後者の拡張モジュールとして外付けセンサとエネルギーハーベスタがある。ユーザは目的に合う USB ドングルと拡張モジュールを組み合わせることが可能であり、これら 3 つの要素が協調することで高品質かつ低消費電力なセンシング活動を実現する。WakeDroid は以下の 7 つの要素技術から構成される。

表 2.1: Android OS のリファレンスモデル7機種における, USB 端子からの起動トリガに必要な電圧とエネルギー量. USB 端子に印加された電圧がこの値に達すると, スマートフォンは起動プロセスを開始する. エネルギー量は概ねミリジュールのオーダーに収まっており, 無線給電やボタン電池で十分に給電できる.

機種名	OS バージョン	印加電圧	エネルギー
Nexus S	4.1.2	3.76 V	0.9 mJ
Galaxy Nexus	4.3	4.00 V	20.9 mJ
Nexus 4	5.1.1	3.92 V	0.3 mJ
Nexus 5	4.4.4	4.23 V	0.4 mJ
Nexus 6	6.0	4.54 V	639 mJ
Nexus 5X	7.0	2.80 V	0.05 mJ
Nexus 6P	7.0	2.81 V	0.04 mJ

### 2.2.1 USB 給電による起動トリガ

7つの要素技術の中でとりわけ重要な技術は, スマートフォンの USB 端子を刺激することで対象のスマートフォンを電源オフ状態から起動させる技術である. Android 端末にはバッテリー充電中の電源状態管理のためのソフトウェアが備わっており, 電池残量が著しく少ない場合を除けば, 端末に充電器が接続されると自動的に起動する. このソフトウェアは Linux カーネル上で動作するものであり, そのソースコードは Android Open Source Project (AOSP) 内の `/system/core/charger/charger.c` として公開されている. 本手法では, この電源状態管理ソフトウェアを改変することで, USB 端子に所定の電圧が印加された場合に自動的に Android OS を起動させる. 同様の自動起動機能は Android OS 標準のブートローダにも備わっているが, この機能は Android OS のリファレンスモデルである Nexus/Pixel シリーズでしか利用できない. これとは対照的に, 我々の手法は後に 2.4.1 節で詳細を述べるように, 事実上あらゆる Android 端末に適用が可能である.

我々は, Android OS のリファレンスモデル7機種を用いて, 起動トリガに必要な USB 端子への印加電圧と必要となるエネルギー量を計測した. 具体的な機種名を挙げると, Nexus S, Galaxy Nexus, Nexus 4, Nexus 5, Nexus 6, Nexus 5X, Nexus 6P の7機種である. 計測実験では, スマートフォンの USB 端子にソースメータ (Keithley SMU 2450) を接続し, 印加電圧と印加時間を調整してスマートフォンが起動する最低限度を調べ, その際にソースメータから供給したエネルギー量を計測した. 表 2.1 に示したように, 起動トリガに必要な最低電圧は機種により異なる. この差異は, ロジックボード上に実装された電源管理 IC の違いに起因するものであるが, 世界最大のスマートフォン向け CPU ベンダである Qualcomm は CPU と電源管理 IC をセットで提供しているため, CPU 型番が同じスマートフォンでは必要な電圧値も同程度となる. 必要なエネルギー量は機種により大きく異なるが, 必要な印加時間が高々1秒と短く概ねミリジュールのオーダーに収まるため, 無線給電やボタン電池で十分に給電可能である.

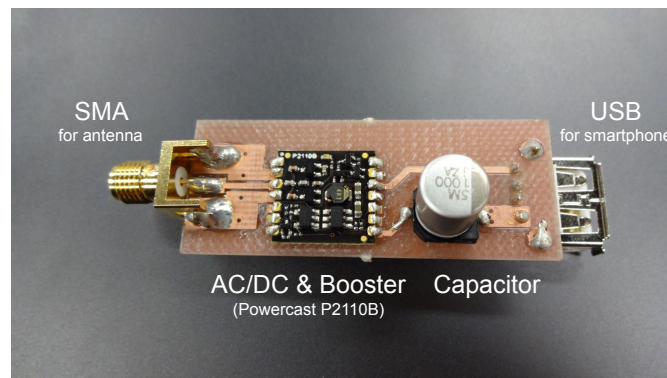


図 2.2: 無線起動トリガのためのレクテナ USB ドングル。この USB ドングルは、給電用のマイクロ波を受信して整流・昇圧し、接続されたスマートフォンの USB 端子に所定の電圧を印加する。

### 2.2.2 マイクロ波給電による無線起動トリガ

この節では、遠隔操作でスマートフォンの電源投入を可能とする USB ドングル型デバイスを紹介する。この外付けドングルは、低消費電力であるだけでなく、対象のスマートフォンに十分な電圧を供給できる必要がある。また、可能な限り低コストでありつつ、指定したスマートフォンを選択的に起動できる能力が求められる。これらの要件を満たすため、ここでは 915 MHz 帯のマイクロ波給電技術を応用して、無線信号による起動トリガを実現する方式を提案する。この方式のユニークな点は、USB ドングル自身が無線給電により駆動し、USB ドングルに内蔵電池が不要な点である。

我々は、図 2.2 に示すように、SMA 端子、整流回路、1 mF のキャパシタ、USB 端子を搭載した USB ドングルを実装した。この USB ドングルは、マイクロ波給電源からの電波をアンテナで受信し、整流して取り出したエネルギーをキャパシタに蓄える<sup>1</sup>。キャパシタの電圧が規定値 (1.25 V) を超えると、電圧が規定値 (1.02 V) を下回るまでの間キャパシタに蓄えたエネルギーを取り出し、これを 5 V に昇圧して USB 端子に印加する。整流昇圧および電源管理の機能は Powercast P2110B を用いて実装しており、整流回路の変換効率は入力 3 dBm の時に 60% 超であった [33]。

無線電力伝送には幾つかの方式が存在するが、本研究でマイクロ波方式を採用したのは、アンテナと送信電力を変更することで到達範囲を制御しやすいことに加え、既存の高出力 RFID リーダや Wi-Fi ルータで給電源を代用できるという利点もあるためである。マイクロ波方式の欠点は、同じ周波数帯を用いる無線通信と干渉することであるが、無線給電と無線通信を両立する技術は SWIPT と呼ばれており広く研究されている。周波数帯と送信出力、到達距離については、後に 2.4.2 節で詳しく議論する。

ここでは、Nexus 5X と Powercast 給電キットを用いて、無線起動トリガ信号の到達距離の計測を行った [33]。送信周波数は 915 MHz、送信電力は送信アンテナ利得 (8 dBi) を除いて 0.48 W とした。実験場所は武田先端知ビル 1 階ロビーである。図 2.3 に、送電距離と起動トリガの所要時間の関係を示す。我々の手法では、マイクロ波給電源から 2 m

<sup>1</sup>必要となる最低の入力電力は -14 dBm (=0.04 mW) である。

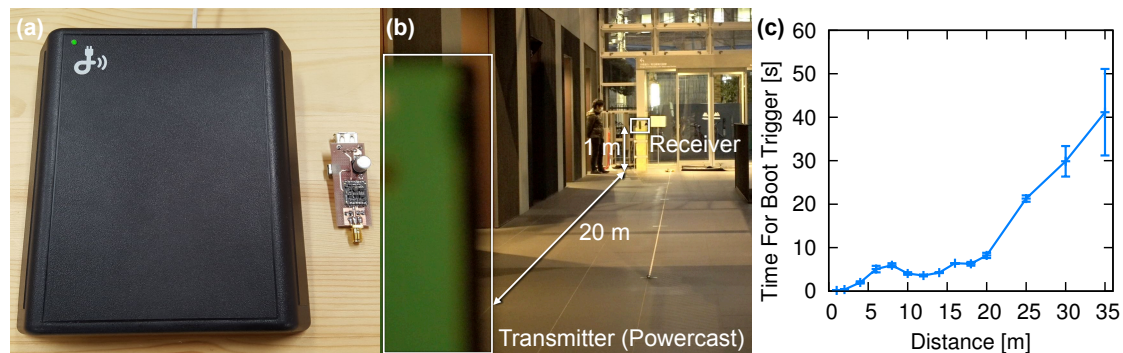


図 2.3: (a) マイクロ波給電源 (Powercast TX91501). (b) 到達距離の計測環境. (c) 送電距離と起動トリガの所要時間の関係. 給電源から 35 m 以内にあるスマートフォンの起動プロセスを 1 分以内に開始させることができる. 所要時間が距離に対して単調増加しないのは, 壁や床で電波が反射するためである.

以内にあるスマートフォンをほぼ瞬時に, 35 m 以内にあるスマートフォンを 1 分以内に起動させることが可能である. しかしながら, 送電距離が長くなるにつれ, 電波強度が不安定となるため所要時間のバラつきが大きくなる. 送信電力を向上させた場合の送電距離の限界については, 後に 2.4.2 節で詳しく議論するが, 送電距離については改善の余地がある.

### 2.2.3 マイクロ波給電による選択的な無線起動トリガ

前節で紹介した無線起動トリガ技術では, 電波の到達範囲内にある全てのスマートフォンに対して一括で起動プロセスの開始を促すことができたが, 用途によっては特定のスマートフォンだけを起動できた方が便利である. そこで我々は, 前述の電源状態管理ソフトウェアにさらなる変更を加えることで, 給電側で指定したスマートフォンのみを選択的に起動する機構を提案する. この機構により, 指定されたスマートフォン以外では, 受信した電力を内蔵バッテリーの充電に充てることが可能となる. マイクロ波給電方式で伝送できる電力は, スマートフォンを常時起動させておくには少なすぎるものの, スマートフォンが電源オフ状態であれば電力を少しずつバッテリーに蓄えることが可能である. 本機構の設計にあたり, 以下の 5 つの要件を設定した.

- 要件 1: マイクロ波給電源の修正が最小限であること.
- 要件 2: ID を幅広いスマートフォンで認識可能であること.
- 要件 3: 端末の収容台数に応じて ID 長を柔軟に変更できること.
- 要件 4: 給電停止時間が最小限であること.
- 要件 5: 複数の端末を一括して指定できること (マルチキャスト).

要件 1 を満たしつつ, 受信側の USB ドングルを可能な限りシンプルな構成とするため, 変調方式としては単純なオンオフ変調を採用した. Bluetooth や ZigBee といった近距離

表 2.2: 各機種で認識可能な最短のオン/オフのシンボルの長さ。オン状態とオフ状態とで大きな非対称性が見られる。

機種名	OS バージョン	“オン” シンボル	“オフ” シンボル
Nexus S	4.1.2	570 ms	2 ms
Galaxy Nexus	4.3	570 ms	4 ms
Nexus 4	5.1.1	230 ms	4 ms
Nexus 5	4.4.4	330 ms	7 ms
Nexus 6	6.0	350 ms	25 ms
Nexus 5X	7.0	1150 ms	22 ms
Nexus 6P	7.0	900 ms	21 ms

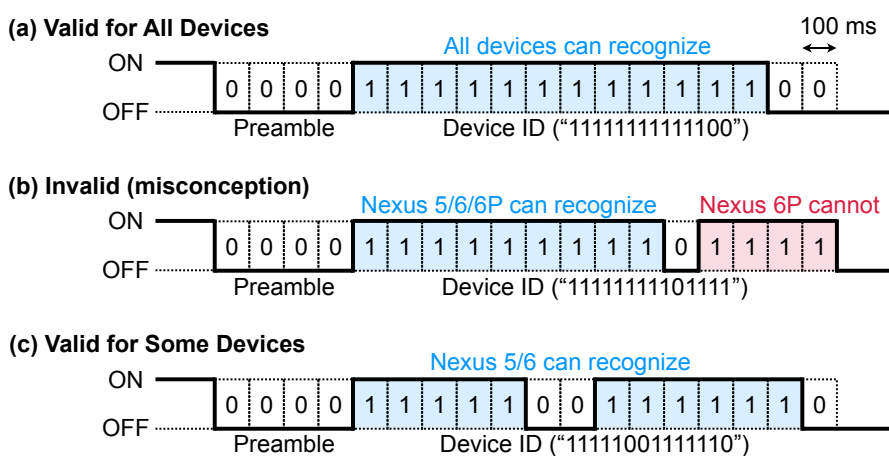


図 2.4: 選択的起動トリガを実現するオンオフ変調方式。特定のデバイスで認識できないIDパターンが存在するため、受信側の端末に合わせて適切なIDを割り振る必要がある。

無線通信規格を用いる場合と比べて、我々の方式では受信機の製造コストを最小限に抑えられる。また、スマートフォン側に特別なハードウェアは必要なく、電源状態管理ソフトウェアが給電のオンオフパターンを監視し、給電側で埋め込まれたIDを読み取る。要件2に関する予備実験により、認識可能なシンボル長はスマートフォンの機種により異なることが明らかとなった（表 2.2）。この差異は、USB 端子からの給電電圧を安定させるためのキャパシタの大きさに起因するものと考えられる。最も反応の遅い端末に合わせたスロット長では、送達時間が長くなりすぎるため、ここではオフ状態の認識下限時間を基準に余裕を持たせてスロット長を 100ms に設定し、受信側の端末で認識可能な ID パターンを選択的に割り当てることとした（図 2.4）。この割り当てロジックは、ID パターンの長さだけでなく、他の端末への給電時間も考慮する（要件 3, 要件 4）。また、要件 5 を満たすため、マルチキャスト IP に相当するグループ ID を導入した。

このようなデバイス選択ロジックを実装するにあたり問題となるのは、給電波のオンオフが対象のスマートフォンに直接伝わらない点である。すなわち、USB ドングル内で整流された電力が一時的にキャパシタに蓄えられ、直接 USB 端子に印加されないために、スマートフォン側では給電波のオンオフを直接検知できないのである。この問題を解決する



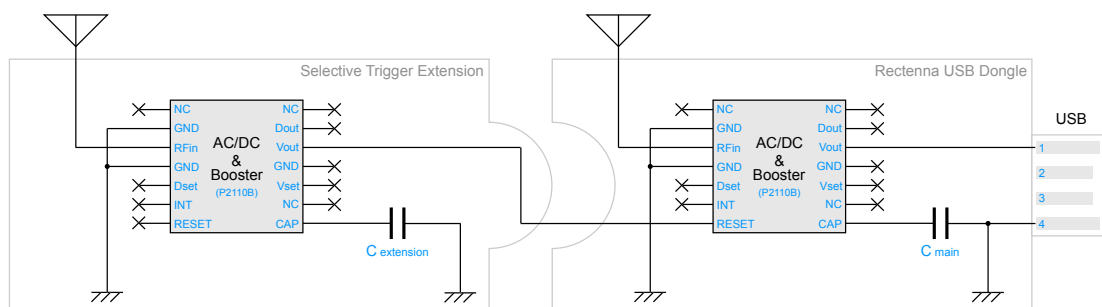


図 2.5: 選択的起動トリガを実現するための拡張モジュールの回路図。基本設計は USB ドングル自体とよく似ているが、キャパシタの容量が異なる。給電のオンオフを最小限の遅延時間で出力に伝えるため、拡張モジュールのキャパシタ容量は USB ドングルのキャパシタ容量より遥かに小さい必要がある。

ため、我々は選択的起動を実現するための拡張モジュールを開発した。図 2.5 に、拡張モジュールの回路図を示す。このモジュールは、それを取り付ける先の USB ドングルとよく似ているが、キャパシタの容量を小さくすることで入力への応答性を高めてある。出力は、USB ドングルのリセットピンに接続されており、USB ドングルの出力を制御する。具体例を挙げて動作を説明すると、無線信号がある時には、拡張モジュールの出力が High になり、USB ドングルの出力は抑制されてキャパシタに電荷を蓄える。一方、無線信号がない時には、拡張モジュールの出力は Low となり、USB ドングルはキャパシタに蓄えた電荷を放出し USB 端子に電圧を印加する（図 2.6）。拡張モジュールのキャパシタを十分に小さく設計するならば、出力のスイッチング遅延時間 ( $t_c$ ,  $t_d$ ) は短く、タイムスロット  $T$  と比べて無視できる水準となる。他方で、キャパシタの放電継続時間も短くなるため、給電オフ状態が長く続かないよう ID パターンを設計する必要がある。この条件を満たす限り、給電の有無によって USB ドングルの出力をほぼ直接的に制御可能であり、マイクロ波給電源をスイッチングすることで、USB 端子への電圧印加をオンオフ変調できる。

#### 2.2.4 マイクロ波給電源のコントローラ

前節で紹介した無線起動トリガを利用して遠隔操作でオンデマンドの電源投入を行うには、ユーザからの指令を受信してマイクロ波給電源の電源管理を行うコントローラが必要となる。また、センシング現象の発生確率を予測するために、天気予報や渋滞情報などを Web サイトからスクレイピングすることも有用である。このような目的を達成するための最適解の 1 つは、Raspberry Pi のような GPIO ピンを備えたボードコンピュータを利用することである。ここでは、前節で述べた条件に従ってデバイス ID を生成し、そのパターン通りにマイクロ波給電源のスイッチングを行うプログラムを実装した。ユーザが入力すべき情報は、機種名と認識可能な最短のシンボル長、各機種の台数である。生成された ID パターンは、スイッチング制御のためのプログラムに読み込まれ、マイクロ波給電源の電源線に噛ませたリレーを GPIO ピンから制御することで、給電パターンに反映される。なお、我々の予備実験では、リレーの動作によって生じる遅延は 10 ms 未満であり、前節で定めたスロット長と比べても十分に短いことが明らかとなっている。

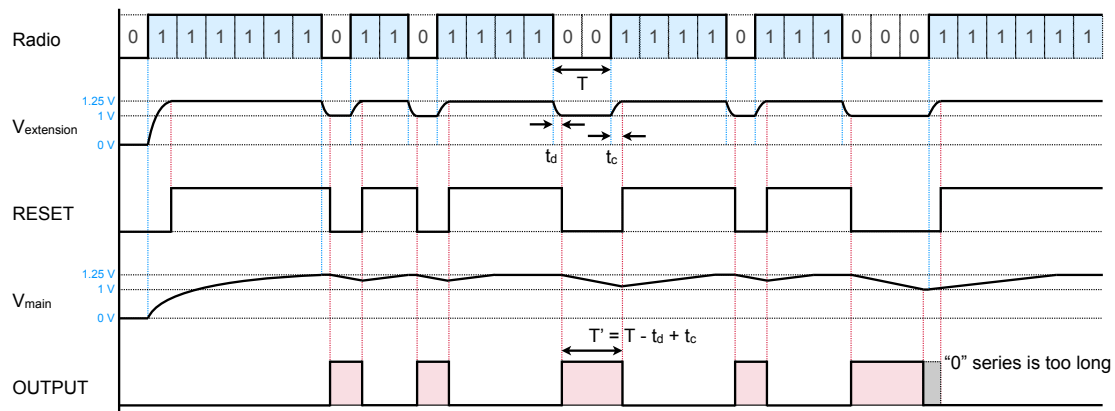


図 2.6: 選択的起動を行うためのオンオフ変調のタイミング・ダイアグラム.  $T$  と  $T'$  の差異を抑えるため、充放電時間 ( $t_c$ ,  $t_d$ ) に影響する拡張モジュールのキャパシタは十分に小さく設計する必要がある. この条件を満たす限りにおいては、給電波の有無として伝わってきた符号を反転する形で USB 端子に電圧を印加できる.

### 2.2.5 イベントドリブンでの起動トリガ

2.2.2 節で提案したマイクロ波無線給電方式の USB ドングルと対になる形で、イベントドリブン方式の USB ドングルを提案する. このドングルはセンサで周囲の状況を監視し、スマートフォンの起動プロセスを開始させるタイミングを決定する. センシング活動によって必要となるセンサが異なるため、ここではセンサ部分をロジックボードとは別の拡張モジュールとして分離してある. このモジュール化により、ユーザは各自の目的に見合ったセンサを利用できる. また、コネクタと配線ルールさえ遵守すれば、拡張モジュールの自作も可能である.

ここでは、ロジックボードを実装するにあたり、8pino と呼ばれる小型の Arduino 互換マイコンボードを採用した [34]. 8pino はスリープ時の待機電力が非常に小さく、乾電池で数年間の駆動が可能であることが主な採用理由である. 図 2.7 にイベントドリブン方式の USB ドングルの回路図を示す. この USB ドングルは 2 本の乾電池で駆動し、それぞれ拡張モジュールとスマートフォンに接続するための 2 つのインタフェースを持つ. また、2 つの DC-DC コンバータを内蔵し、8pino に 3.3 V を、スマートフォンに 5 V の電源を供給する. 拡張モジュールとのインタフェース部分には、RS-232C 端子を通して 7 本の配線 ( $V_{cc}$  ピン,  $V_{power}$  ピン, GND ピン, 4 本の I/O ピン) が通る. I/O ピンを 4 本としたのは、原始的なセンサが 3~4 本程度の端子を持つことと、8pino の GPIO ピン数の制約によるものである.

イベントドリブン方式の典型的な利用シナリオとしては、USB ドングルで拡張モジュールのセンサ値を 1 時間に 1 回の頻度で確認し、1 週間に 1 回程度の頻度でスマートフォンを起動して詳細なセンシングを行う、というものである. ディープスリープモードでの待機電力を 0.4 W, 1 秒間のセンシング中の消費電力を 25 mW, スマートフォンの起動トリガに必要なエネルギーを 0.05 mJ, USB ドングルの電源として単五電池 2 本 (3 V, 900 mAh) を想定すると、推定電池寿命は 276 日となる.

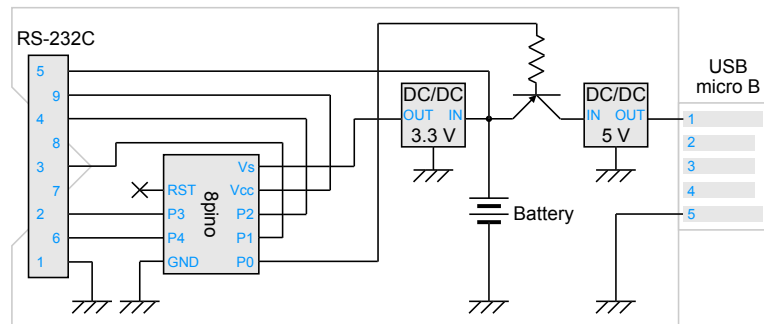


図 2.7: イベントドリブン方式の USB ドングルの回路図. USB ドングルは乾電池で駆動し、それぞれ拡張モジュールとスマートフォンに接続するための2つのインタフェースを持つ。

### 2.2.6 エナジーハーベスティング モジュール

イベントドリブン方式の USB ドングルの電池寿命を伸ばすための拡張モジュールとして、我々はエナジーハーベスティングモジュールを実装した。ここで提案するのは、太陽電池とリチウムポリマー電池を内蔵し、昼夜を問わず USB ドングルに給電を行う拡張モジュールである。巨大な太陽電池パネルは風で吹き飛ばされる危険性があることから、スマートフォンより一回り小さい  $10\text{ cm} \times 5\text{ cm}$  の太陽電池を採用した。晴天時の最大出力は  $300\text{ mW}$  程度である。出力は  $V_{power}$  ピンを通して RS-232C 端子経由で USB ドングル側に供給され、乾電池と共に 8pin を駆動する。

後に 2.3.5 節で詳細を述べる有毒ガス検知のシナリオでは、このエナジーハーベスティングモジュールが重要な役割を果たす。我々が採用したガスセンサは、センサ部を適温に保つために常時  $40\text{ mW}$  の電力を消費する。これを単三電池 2 本 ( $3\text{ V}$ ,  $2700\text{ mAh}$ ) で駆動すると、わずか 8.4 日で電池が切れてしまう。ここにエナジーハーベスティングモジュールを追加すると、電池寿命は飛躍的に向上する。太陽電池の平均出力を最大出力の 20%、すなわち  $60\text{ mW}$  と仮定すると、ガスセンサの消費電力を賄えるだけでなく、余剰電力をスマートフォンの充電に回すことができる。これにより、非常事態に備えて待機中のスマートフォンを満充電の状態に長期間保つことができる。

### 2.2.7 Android アプリの高速起動

Android OS の起動プロセスを短縮することは、スマートフォンの間欠駆動時の電力的なオーバーヘッドを削減し、電池寿命を向上させるために重要である。そこで我々は、起動トリガから Android アプリとして実装されたセンシングロジックの実行開始までの所要時間を短縮する技術についても開拓を行った。Android OS には、OS の起動完了や電池残量の低下といった端末上の重要なイベントを、端末上の全アプリに通知する Broadcast Intent という仕組みが備わっている。OS の起動完了時には `BOOT_COMPLETED` という名前の Broadcast Intent が配信され、バックグラウンドで常駐するアプリが起動する重要な契機となっている。しかしながら、`BOOT_COMPLETED` の Broadcast intent は、OS 起動直後にアプリを実行するための最適な手段でないことが我々の調査により明らかとなった。より



表 2.3: Android OS の種々のディストリビューションを実行する Nexus 5 における，電源投入からセンシングアプリ起動までの所要時間の比較．ホーム画面に指定されたアプリは，起動完了を全アプリに通知する `BOOT_COMPLETED` の Broadcast Intent 配信より，遥かに早いタイミングで起動される．

ディストリビューション名	OS バージョン	ホーム画面	<code>BOOT_COMPLETED</code> 配信
AOSP (Pure Android)	4.4	22 秒	40 秒
	5.0	30 秒	39 秒
	5.1	29 秒	36 秒
	6.0	35 秒	42 秒
CyanogenMod	4.4	24 秒	27 秒
	5.1	26 秒	32 秒
	6.0	33 秒	39 秒
SlimROM	4.4	23 秒	36 秒
	6.0	30 秒	42 秒

具体的には，`BOOT_COMPLETED` の Broadcast intent に先んじて，ホーム画面として設定されているアプリが起動されることが判明した．

表 2.3 は，Android OS の種々のディストリビューション，OS バージョンにおける，電源投入から `BOOT_COMPLETED` の Broadcast intent 配信まで，または，電源投入からホーム画面アプリの実行までの所要時間をまとめたものである．Android OS のユーザインタフェースは，ホーム画面も含めて Android アプリとして実装されており，アプリ開発者は独自のホーム画面を作成できるため，センシングアプリを最も早く起動する方法は，当該アプリをホーム画面として実装することである．なお，この際に必ずしも一般的なホーム画面としての機能を全て実装する必要はない．この発見により，Google 純正の Android 4.4 を実行する Nexus 5 では，OS の起動時間を実質的に 18 秒短縮し，電源投入 1 回あたり 13.7 J のエネルギーを節約できる．また，Android 4.4 が Android 5.x や 6.x より軽量であり，起動プロセスが短いことも実験で確認された．これは，Android 4.4 が，発展途上国に多く見られる安価で低機能のデバイスをサポートするために設計されたバージョンであることと符合する．起動プロセスの長さは間欠駆動時の消費電力に直結する重要な要素であるため，我々はプロトタイプ実装において可能な限り Android 4.4 を採用することとした．

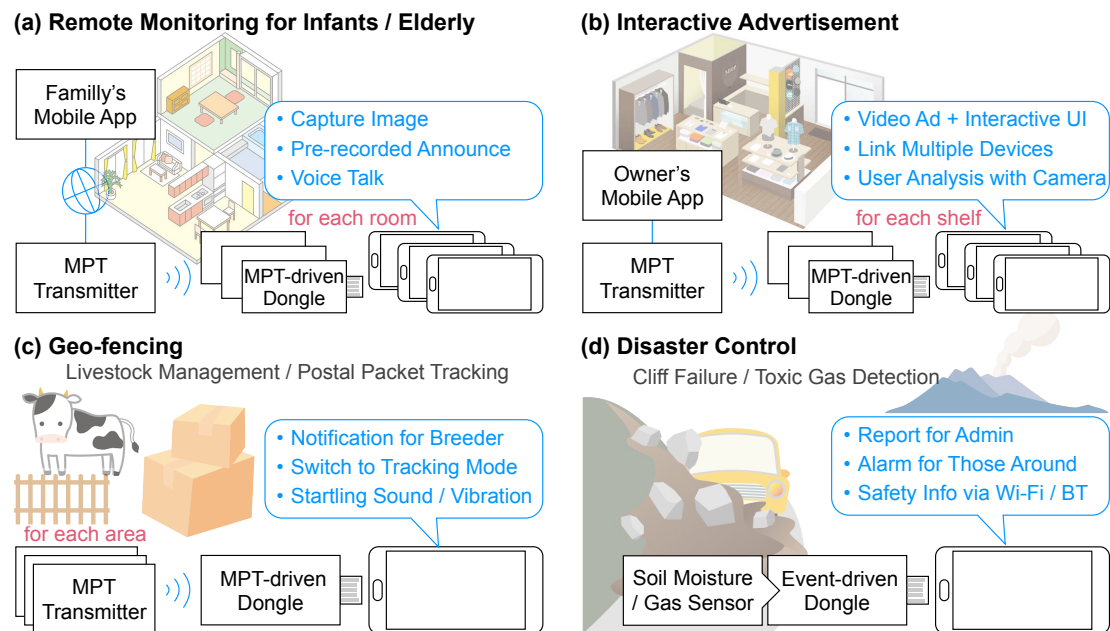


図 2.8: WakeDroid が中古スマートフォンの性能を引き出す 4 つの典型的な利用シナリオ。ここでは、個人や零細企業、地方自治体など予算の限られたユーザを想定している。(a), (b) では、複数のスマートフォンと複数のマイクロ波無線給電方式 (MPT-Driven) の USB ドングルを使用し、(c) では、複数のマイクロ波給電源を使用する。(d) では、イベントドリブン方式 (Event-Driven) の USB ドングルを用いる。(a), (b), (d) では、周囲のユーザとのインタラクションが生じる。

## 2.3 応用例

WakeDroid は、中古スマートフォンに専用センサノードとしての「第二の人生」を与えることで、個人や零細企業の草の根的なセンシング活動を支援することを目標としている。従って、我々が想定するユーザは、個人や予算の限られた零細企業、地方自治体、非営利団体、ボランティア団体などである。WakeDroid は幅広い用途に利用できるよう設計されており、本節では、WakeDroid と組み合わせることで中古スマートフォンの残存能力を有効活用できる 4 つの典型的な利用シナリオを紹介する (図 2.8)。

### 2.3.1 乳幼児や高齢者の遠隔見守り

留守番中の乳幼児や独り暮らしの老人を遠隔から見守るシステムには幅広い需要がある。高齢者の場合、持病の発作や転倒による骨折を素早く検知することは重要である。しかしながら、プライバシーの懸念により商用サービスの利用を躊躇う人が少なくない。ここで求められる要件は、低コストで、プライバシーに配慮しつつ、遠隔地の家族がオンデマンドでセンシング機能を実行できる、というものである。これらの要件を満たすため、我々は中古スマートフォンと WakeDroid を組み合わせた遠隔見守りシステムを提案する。機の引き出しの中に眠る中古スマートフォンは、調達の手間なく使えるだけでなく、初期費用がゼロという利点がある。

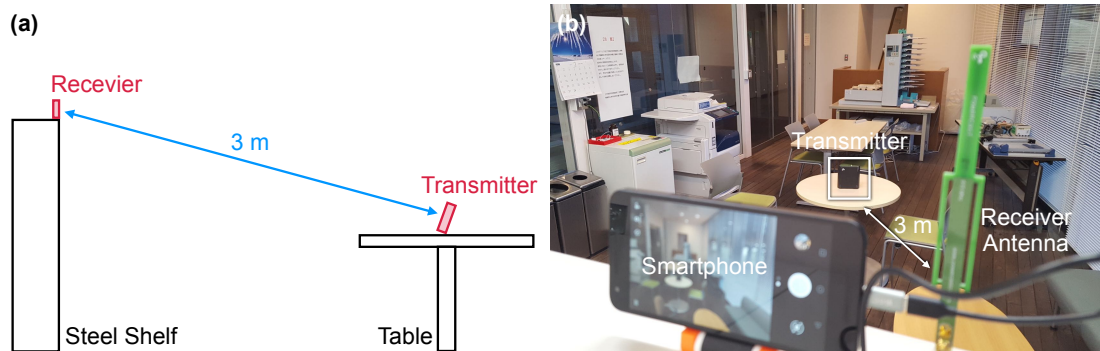


図 2.9: (a) 遠隔みまりのデモ実験の配置図. マイクロ波給電源から受信側 USB ドングルまでの距離は 3 m であった. (b) スチール棚の上に設置した見守りシステムのプロトタイプ.

本システムは、スマートフォンの起動を判断する情報源、マイクロ波給電源とそのコントローラ、図 2.2 に示したマイクロ波給電方式の USB ドングル、部屋ごとに設置されたスマートフォン、から構成される。情報源としては、Wi-Fi 電波を用いた行動認識技術 [35] を用いることで、見守り対象の異常な動きを検知できる。Wi-Fi 電波を用いた行動認識はプライバシーの侵襲度が低い反面、実際に何が起きているのかを正確に検知できない。例えば、見守り対象が動かなくなった場合に、ベッドで寝ているだけなのか、ソファに座ってテレビを見ているのか、持病の発作により倒れたのか、の判別ができない。そこで、状況を確認するために、本システムはマイクロ波給電源に通電し部屋に設置されたスマートフォンを起動する。ここで、Wi-Fi 電波を用いた行動認識技術で対象ユーザの居場所がある程度分かっているならば、見守り対象の居る部屋に設置されたスマートフォンだけを選択的に起動することも可能である。起動後は、スマートフォンで部屋の様子を撮影し、遠隔地の家族に送り届ける。必要に応じて、事前に録音した音声で見守り対象とコミュニケーションを取ったり、遠隔地の家族と音声通話を始めたりできる。画像認識の技術と組み合わせれば、異常状態を検知して自動的に救急車を呼ぶことも可能となる。また、遠隔地にいる家族のスマートフォン上のアプリから、マイクロ波給電源のコントローラにセンシング指令を送信し、オンデマンドで見守り対象の状態を確認することもできる。このように、我々のシステムでは、初期コストを抑えつつ、高性能で拡張性の高いサービスを提供できる。

図 2.9 は、室内に設置されたプロトタイプの様子である。ここでは、棚の上にスマートフォンを設置することで、十分に広い視野角を得ている。我々のデモ実験では、マイクロ波給電源から 3 m 離れた位置に USB ドングル付きのスマートフォンを設置した。給電開始から平均 38.5 秒で室内の様子を撮影し、平均 65.0 秒で撮影画像を家族に送り届けることに成功している。なお、撮影した画像サイズは  $4032 \times 3024$  ピクセルであり、ファイルサイズはおよそ 2.6 MB であった。撮影画質と応答速度の優先度によっては、撮影画像サイズを調整する余地がある。いずれにせよ、先進国の都市部における救急車の到着待ち時間が 10 分程度であることを踏まえると、この遅延時間は十分に実用的である。

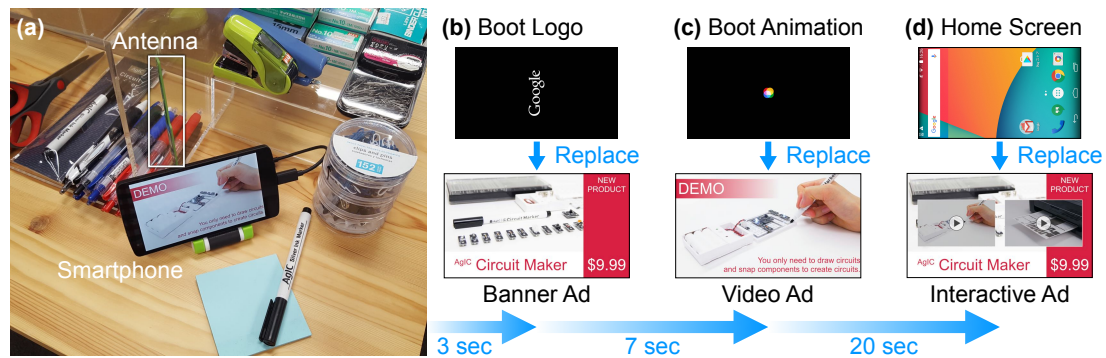


図 2.10: (a) 文具店を想定したインタラクティブ広告のプロトタイプ. ここでは, Android OS の (b) ブートロゴと (c) ブートアニメーションを広告画像に置き換えることで, 起動トリガから広告表示までのタイムラグを最小限に抑えている. (d) 起動プロセスの完了後は, あらかじめ指定した Android アプリが画面表示を引き継ぐ.

### 2.3.2 インタラクティブ広告

小売産業では店頭設置型の広告システムが注目を集めている. 大規模なショッピングモールでは, フロアマップや広告コンテンツを表示する大型ディスプレイの普及が進んでおり, 小規模から中規模の小売店では, 6 インチ前後のディスプレイを搭載した動画再生機が一般的となりつつある. しかしながら, 前者のような大規模商業施設向けのシステムは初期費用が高く, 後者のような小中規模店舗向けのシステムはユーザとのインタラクションがなく広告効果が薄い. ここでの要件は, 低コスト, 小型かつインタラクティブな UI を持つデバイスで, 目障りな電源配線が不要で, 状況に応じたマルチメディアコンテンツを表示できること, である. 小規模店舗をサポートするため, 我々は WakeDroid を用いた新しい広告システムを提案する.

ここでは, WakeDroid と中古スマートフォンを組み合わせることで, 低コストでインタラクティブな広告デバイスを実現する. このシステムでは, 無線起動トリガにより顧客の目の前にあるデバイスを選択的に起動できるだけでなく, ブートロゴやブートアニメーションを広告画像に差し替えることで, ほぼ遅延なく広告コンテンツの再生を開始できる. さらに, 複数のデバイスを協調させて同じタイミングで動画コンテンツを再生するなど, 効果的な宣伝が可能となる. OS 起動完了後は, タッチパネルを介してインタラクティブなマルチメディア広告を提供でき, さらに, 顔認識や音声認識の技術を組み合わせることで, 顧客属性の分析も行える.

図 2.10 は, 文具店を想定したインタラクティブ広告のプロトタイプである. 実装にあたり, スマートフォンとして Nexus 5 (Android 4.4) を採用し, OS のイメージファイルの 1 つである `imgdata.img` に含まれるブートロゴと, Android OS のソースコードのうち `/system/media/bootanimation.zip` に配置されているブートアニメーションを, 自作の広告画像に差し替えた. 後者の zip ファイルには, ブートアニメーションの各コマの画像と, 各コマの長さやループ回数を指定するメタ情報が格納されている. 我々の評価実験では, 起動トリガから 3 秒後にコンテンツの表示を開始し, 30 秒後にはインタラクティブな操作が可能であった.



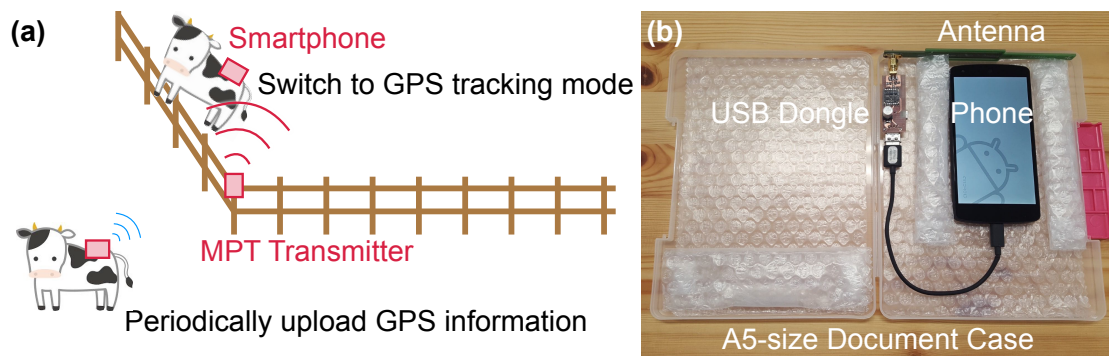


図 2.11: (a) 家畜管理システムの概要の概要. 管理対象の家畜にスマートフォンと WakeDroid ドングルを取り付け, 飼育場の外縁部にマイクロ波給電源を設置しておく. 家畜が飼育場の外縁に近づくと, WakeDroid はスマートフォンを起動して飼育者に通知する. (b) 家畜管理システムのプロトタイプは A5 サイズの書類ケースに収まる.

### 2.3.3 ジオフェンス

ジオフェンスは郵便小包や貨物コンテナといった移動体の追跡に有望な技術である. その追跡対象は無生物に限らず, 乳幼児やペット, 家畜といった自発的に動き回るものも対象となり得る. とりわけ家畜は一頭あたり数十万円と比較的高価であり, ひとたび家畜が飼育場の柵を破壊して脱走すると, 他の家畜が連鎖的に脱走する恐れもあり, 畜産家にとっては大きな経済的損害となる. 家畜の脱走の予兆を検知し, 万が一の脱走後に位置追跡を行うシステムには少なからず必要があるが, 手頃なソリューションが存在しないのが現状である. これは主に, GPS モジュールの消費電力が大きいため電池の消耗が激しく, 日々の充電作業が不可欠で手間が掛かる, という技術的な問題に起因すると考えられる. また, アクティブ RFID タグを用いて予兆を検知する方法も考えられるが, 脱走後の追跡ができない点が問題となる.

WakeDroid には, 無線トリガ信号の到達距離の限界という形で, ジオフェンスの特徴がもともと備わっている. 具体的には, WakeDroid はスマートフォンがマイクロ波給電エリアに入った時にしかスマートフォンを起動できない. さらに, マイクロ波給電ごとに異なる ID を割り当てておくことで, スマートフォンはどの給電エリアに入ったのかを検知できる. 従来の GPS 方式のジオフェンスと比べると, スマートフォン自体の待機電力も位置特定に必要な電力も桁違いに少なく, それゆえ電池寿命を飛躍的に向上できる.

図 2.11 に家畜管理システムのプロトタイプを示す. このプロトタイプは, スマートフォンとマイクロ波給電方式の USB ドングル, 衝撃吸収用のクッション材とプラスチック製の筐体から構成される. パッケージ全体の重量は約 300 g であり, 寸法は 225 × 175 × 25 mm と小型である. 管理対象の家畜にこのパッケージを取り付け, 飼育場の外縁部にマイクロ波給電源を設置しておく. 家畜が飼育場の外縁に近づくと, WakeDroid はスマートフォンを起動して飼育者に通知する. また, 音や振動, フラッシュの点滅により, 家畜を驚かせることも可能である. 管理対象の家畜が脱走したことを検知すると, 位置追跡モードに切り替わり携帯電話網を介して現在地を通知し続ける. これにより, 家畜が飼育場の外に出てしまった場合でも, 飼育者は家畜の行方を追うことができる.

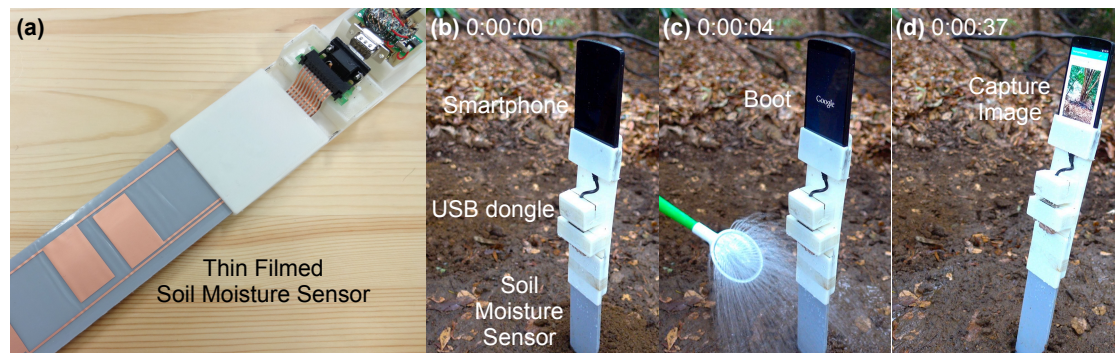


図 2.12: (a) 我々は [36] で開発された低コストな土壤水分センサを拡張モジュールに採用した. (b) 崖に設置されたがけ崩れ検知のプロトタイプ. (c) 小規模ながけ崩れを人工的に発生させるため手作業で水撒きを行ったところ, 提案システムは土壤水分の増加を検知した. (d) その後, 提案システムは周囲の写真を撮影し管理者に送信する.

### 2.3.4 防災 (1) : がけ崩れの検知

自然災害予防の分野において, 土壤水分量は土砂災害の予兆を検知するための有用なパラメータである. がけ崩れの発生前には, 土壤水分量が急速に増加した後, 直前になって飽和し一定の値を示すことが知られている. しかしながら, 商用電源や電話回線の整備されていない山間部において, 複数の地点に土壤水分センサを設置して土砂災害の監視網を構築することは容易ではない. 実際に, このような用途に適した安価なセンサデバイスはこれまで存在しなかった. そこで我々は, WakeDroid と中古スマートフォンを用いた新しいがけ崩れ検知システムを提案する.

このプロトタイプは, 図 2.12(a) に示した低コストな土壤水分センサを搭載した拡張モジュール, イベントドリブン型の USB ドングル, スマートフォンから構成される. USB ドングルは定期的に土壤水分量を測定し, 土壤水分が急速に増加した場合にのみ, スマートフォンの起動プロセスを開始させる. スマートフォンは周囲の状況を撮影し, 地面の傾きを測定することで, がけ崩れの発生状況を確認する. このデータは携帯電話網を通して地域の防災管理者に自動的に送信される. 本システムは, 商用電源や電話回線を必要としないため, 山間部でも設置が容易である. また, センサ 1 ユニットの製造コストは数千円であり, 予算の限られた地方自治体でも容易に導入可能である.

図 2.12 は, 崖に設置されたがけ崩れ検知のプロトタイプである. 我々の評価実験では, 崖に土壤水分センサを差し込み, 水を掛けることで人工的に小規模ながけ崩れを発生させた. 実験では, 水を掛けてから 4 秒後には水分の増加を検知し, 40 秒後には周囲の様子を撮影した画像を管理者に送り届けることに成功した. がけ崩れの発生を検知した場合, Wi-Fi 通信や Bluetooth 通信, 音声アナウンスで周囲にいる人に避難情報を配信することも可能である. このように, 周囲の人々とのコミュニケーションを取ることができる点は, スマートフォンならではの特徴である.

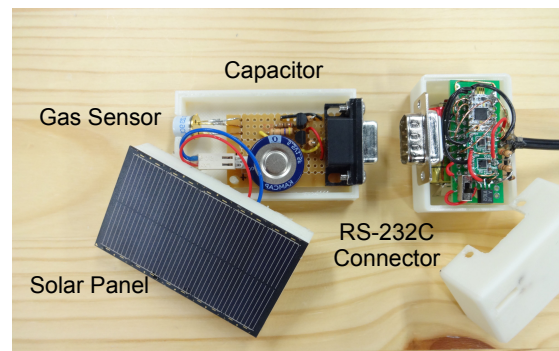


図 2.13: 有毒ガス検知システムのプロトタイプ. このプロトタイプは,  $\text{H}_2\text{S}$  ガスセンサ, 小型の太陽電池とキャパシタ, イベントドリブン型の USB ドングル, スマートフォンから構成される.

### 2.3.5 防災 (2) : 有毒火山ガスの検知

災害予防の別のアプリケーションとしては, 山岳地帯における登山者保護のための有毒火山ガスの検知が挙げられる. 火山ガスには, 硫化水素 ( $\text{H}_2\text{S}$ ) や一酸化炭素 ( $\text{CO}$ ) といった有毒ガスが含まれており, これらの有毒ガスは空気より重いため窪地に溜まりやすいという特徴がある. 致死性があり非常に危険なこれらのガスは透明で無味無臭であるため, 登山者がその存在に気付くのは難しい. この問題を解決するため, 我々は WakeDroid に有毒ガスセンサを拡張モジュールとして組み合わせた有毒ガス検知システムを提案する. このシステムへの要件は, 低コストであり, 周囲の登山者に警告情報を伝えることができ, 商用電源や電話回線のない山岳地帯にも容易に設置できること, である.

図 2.13 に有毒ガス検知システムのプロトタイプを示す. このプロトタイプは,  $\text{H}_2\text{S}$  ガスセンサ, 小型の太陽電池とキャパシタ, イベントドリブン型の USB ドングル, スマートフォンから構成される. ガスセンサで周囲のガス濃度を定期的に測定し, 有毒ガスが検知された場合には, USB ドングルがスマートフォンの起動プロセスを開始させる. スマートフォンが起動すると, 携帯電話網を介して管理者にアラートを送信し, 周囲の登山者への警報アラーム音をバッテリーが切れるまで鳴らし続ける. これと同時に, Wi-Fi 通信と Bluetooth 通信で周囲の登山者に避難情報を配信する. 前節のシナリオと同様に, このシステムもインフラのない山岳地帯にも容易に設置でき, 周囲の人々とのインタラクション機能が強みである. USB ドングルは太陽電池とキャパシタにより駆動され, 余剰電力は電源オフ状態で待機するスマートフォンの内蔵電池を満充電に保つために利用される.

このアプリケーションでは, 安全上の理由から実際の山岳地帯にプロトタイプを設置することができなかった. 代わりに, 実装に用いたガスセンサが検知できるエタノールを用いて, 評価試験を行った. この評価実験では, プロトタイプの隣に脱脂綿を置き, そこにエタノールを滴下した. 我々のプロトタイプは, 10 回の試行において平均 47.3 秒でガス成分を検知し, 70 秒後にはアラーム音を発し始めた. また, 最大の遅延時間は 111 秒であった. 製造コストについては, センサ 1 ユニットあたり数千円に抑えられており, 予算の限られた地方自治体でも容易に導入可能である.

## 2.4 検討事項

前節では、WakeDroid が様々なセンシングタスクに適用できる高いポテンシャルを持つことを、複数の応用例を交えて説明した。この節では、WakeDroid の完成度をさらに高めるために検討すべき事項を紹介する。ここでの最大の懸念事項は、Android スマートフォンの多種多様なハードウェアに起因して生じる機種依存の問題である。

### 2.4.1 スマートフォンの機種依存

我々はこれまで、Android OS のリファレンスモデル 7 機種と、Android Open Source Project (AOSP) [37] で公開されている OS のソースコードを用いて、WakeDroid の開発を行ってきた。AOSP のソースコードは、適用対象の Android 端末からハードウェアドライバを抽出して所定の位置に配置し、いくつかのビルドパラメータを調整することで、あらゆる Android 端末に適用可能である。ここで、適用対象のスマートフォンに求められる要件は、シャットダウン操作に必要なカーネル領域での root 権限へのアクセスと、カスタマイズした OS イメージのインストールに必要なブートローダアンロックの 2 点である。WakeDroid を様々な機種に対応させるためのより簡便な手法は、CyanogenMod [38] や MIUI [39] といった Android OS の派生ディストリビューション (カスタム ROM) を活用することである。これらのカスタム ROM は、既にそれぞれ 1,000 万人以上のユーザを獲得しており、様々なデバイス上で実際に動作している。WakeDroid では、機種依存のある API や機能を利用しないため、原理原則としてはこれらのカスタム ROM と同様に 1,000 万台以上の Android デバイスで動作可能である。

これとは別の懸念事項は、OS バージョンによる違いである。とりわけ、Android 5.0 で導入された SELinux は、システム領域における近年では最大の変更点である。SELinux は通常ユーザだけでなく管理者ユーザや root ユーザの特権までも制限するため、安易には root 権限を利用できなくなる可能性がある。この点について調査したところ、Android OS ソースコードに含まれる SELinux の定義ファイルを編集することで、制限を回避できることが明らかとなった。

### 2.4.2 起動トリガの無線信号の到達距離

2.2.2 節で紹介した無線起動トリガの技術は、対象のスマートフォンへの電源ケーブルの配線を不要とし、デバイス配置の自由度を大きく向上させた。しかしながら、無線信号の電波伝搬に起因する制限は依然として残っている。WakeDroid では無線信号を起動指令の送達だけでなく、USB ドングルへの電源供給にも使用するため、通常の無線通信用途と比べて電波の到達範囲の問題は重大である。この節では、無線起動トリガ信号の到達範囲の上限を明らかにする。

電波の到達距離を左右する要素には、大まかに言って以下の 3 つがある。まず 1 点目として、我々が利用できる周波数帯は、免許が不要な 2.4 GHz 帯や 915 MHz 帯に限られる。より低い周波数帯では、伝搬損失が小さくなり、到達距離が長くなる傾向がある。しかしながら、ほとんどの商用無線機器は ISM バンドを採用しているため、ISM バンドを利用するならば RFID リーダや Wi-Fi アクセスポイントなどの既存機器でマイクロ波給電源



を代用したり、受信回路の実装に既製品を流用したりでき、コストを抑えられるという利点がある。次に2点目として、多くの国において無線電波の送信出力は法令により規制されている。日本では、ISMバンドのうち915 MHz帯については、通信用途では送信アンテナ利得を除いて1 W、それ以外の高周波利用設備では50 Wが上限と定められている。WakeDroidで用いる無線電波は主にUSB Dongleへの給電用途であり、ごく単純なスイッチング機構を備えるものの、事実上は変調機能を持たないため占有する周波数帯域はごく狭い<sup>2</sup>。従って、高周波利用設備に該当すると考えられるため、法令上許される送信出力は50 Wが上限となる。これに加えて、送信機の周辺に存在しうる人体や動物、電子機器の防護についても検討する必要がある。ここで、携帯電話網のLTE基地局が900 MHz帯で最大30 W（送信アンテナ利得を除く）の電波を放射していることを踏まえると、アンテナの周囲に人や動物の立ち入らない環境下であれば、30 Wまでは安全に利用可能と言える。さらに3点目として、無線電波の伝搬損失は、送信機からの距離の2乗に反比例する。この問題については、送信側・受信側の双方に指向性アンテナを用いることで緩和できるが、代償として配置の自由度が低下してしまう。ここでは、高指向性アンテナの配置調整の難しさを考慮し、配置の自由度を尊重するため、極端に指向性の高いアンテナは使用しないこととした。

2.2.2節で述べた給電波の到達距離の計測結果に基き、送信機の出力を0.48 Wから30 Wに向上させた場合、計算上の到達距離は278 mとなる。送信出力を50 Wにした場合は、358 m離れたスマートフォンを起動できることになる。これは、Wi-Fiアクセスポイントのように単一の送信機で、一軒家や小規模な農場をカバーするのに十分である。

### 2.4.3 中古スマートフォンの電池容量

中古スマートフォンの内蔵バッテリーは使い古されており、新品のバッテリーより電池容量が幾分少ない。実効的な電池容量は電池駆動時間に直接的に影響を与える重要な要素であるため、我々は中古スマートフォンのバッテリーを用いて実効容量の計測実験を行った。この検証実験では、購入から3年以上が経過した中古スマートフォンの内蔵バッテリー10個を使用した。なお、ここでNexusシリーズのスマートフォンを使用しなかったのは、内蔵バッテリーを取り外せない仕様の機種が大半であり、ロジックボードの影響を除きつつ計測を行うのが難しいためである。

この実験では、対象のバッテリーに12 Ω抵抗を接続し、放電時間と出力電圧の変化を測定した。抵抗値を12 Ωとしたのは、スマートフォンの画面を消灯した状態でLTE通信を行う際の消費電流が0.3 A前後であり、これに近い電流量を確保するためである。ここでは、電池の実効容量を、満充電したバッテリーが出力電圧3.0 Vを下回るまでに放電できる容量、と定義する。なお、閾値を3.0 Vに設定したのは、リチウムイオン電池の終端電圧が2.5 Vであるにも係わらず、3.0 Vを下回った時点で自動的にシャットダウンするAndroid端末が存在するためである。

表2.4と図2.14は、公称容量と実効容量の比較である。ほとんどのバッテリーが、3年経過後も公称容量の80%以上の電池容量を維持しており、公称容量に対する実効容量の比

<sup>2</sup>長さ0.1秒の矩形波と考えると、±1 kHzの範囲に送信電力の99.9%が含まれる。

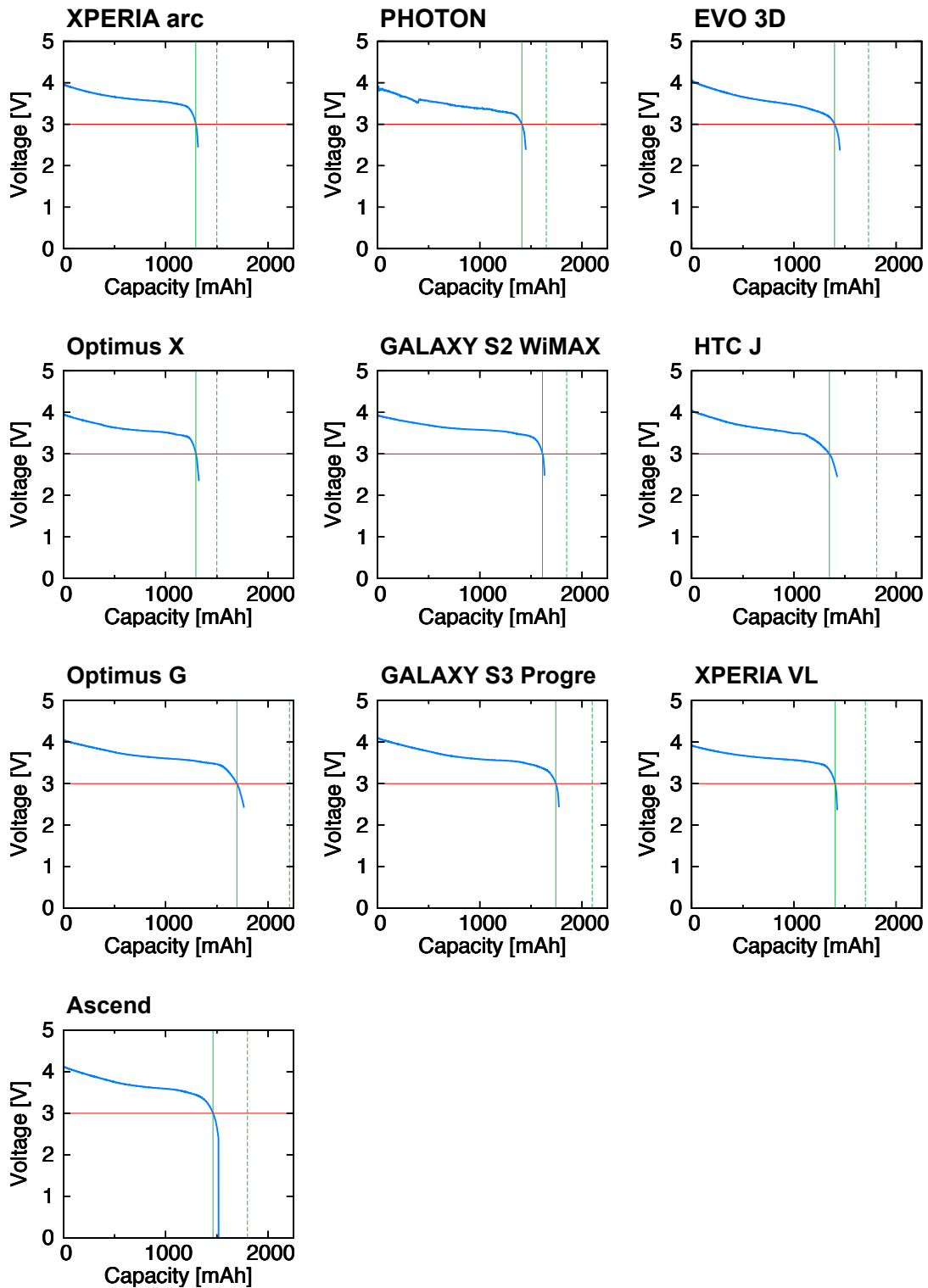


図 2.14: 中古スマートフォンの内蔵バッテリーの放電電圧の時間変化. 緑線は実効容量を点線は公称容量を示す. 購入から3年以上が経過した中古スマートフォンの内蔵バッテリーであっても, 概ね公称容量の約80%までは安定的に電圧を出力できる.

表 2.4: 中古スマートフォンの電池容量. 購入から3年以上が経過したスマートフォンであっても, 公称容量の82.2%の電池容量を発揮できる.

機種名	メーカー	発売日	実効 / 公称容量	比率
Xperia arc SO-01C	SONY	2011/03	1294 / 1500 mAh	86.3%
PHOTON ISW11M	Motorola	2011/10	1411 / 1650 mAh	85.5%
EVO 3D ISW12HT	HTC	2011/10	1398 / 1730 mAh	80.8%
Optimus X IS11LG	LG	2012/01	1295 / 1500 mAh	86.3%
GALAXY S2 WiMAX ISW11SC	Samsung	2012/01	1613 / 1850 mAh	87.2%
HTC J ISW13HT	HTC	2012/05	1347 / 1810 mAh	74.4%
Optimus G LGL21	LG	2012/11	1696 / 2210 mAh	76.7%
GALAXY S3 Progre SCL21	Samsung	2012/11	1743 / 2100 mAh	83.0%
Xperia VL SOL21	SONY	2012/11	1402 / 1700 mAh	82.5%
Ascend HW-01E	Huawei	2012/11	1463 / 1800 mAh	81.3%

率は平均82.2%であった. この計測結果より, 中古スマートフォンであっても電池駆動のセンサノードとして高いポテンシャルを持つことが確認された.

#### 2.4.4 既存の省電力モードとの消費電力比較

WakeDroidを用いることで, 通常の中古スマートフォンを「ノーマリーオフ」のセンサノードとして間欠的に動作させることが可能となるが, Android OSを起動したりシャットダウンしたりする際の電力的なオーバーヘッドが少なからず存在する. この節では, 提案手法の電力的なオーバーヘッドと, 機内モードや緊急時モードといったAndroidスマートフォンに備わっている既存の省電力モードでの待機電力とのバランスを検証する. 言うまでもなく, 損益分岐点はセンシングタスクのデューティサイクルに依存し, 待機時間が長いほど我々の提案手法が有利になる. もちろん, OS起動中の消費電力を最小化するために, 既存の省電力モードを我々の手法と組み合わせることも可能である.

ここで我々は, Google純正のAndroid 4.4をインストールしたNexus 5で, Android OSの起動プロセス, Android OSのシャットダウンのプロセス, スリープモード, 機内モードでの消費電力を測定した. この実験では, Nexus 5を分解して内蔵バッテリーを取り外し, バッテリー接続端子に直流安定化電源から4.2 Vの電圧を印加しつつ, マルチメータ(Keysight 34410A)で電流量を計測した. なお, ここで印加電圧を4.2 Vとしたのは, リチウムイオン電池の満充電時の最大出力電圧に合わせたためである. 我々の実験結果によると, 起動・シャットダウンによる電力的オーバーヘッドは, 1サイクルあたり平均70.5 Jであった. 一方, スリープモードでの待機電力は0.25 W, 機内モードでは0.04 Wであった. これらの結果より, デューティサイクルが4.7分より長い場合には, スリープモードで待機するよりWakeDroidを用いて間欠駆動する方が省電力となる. また, 機内モードと比較した場合は, 29.4分より長いデューティサイクルで提案手法が有利となる.

緊急時モードについては, Samsung Galaxy S6 edge SCV31とSONY Xperia M4 Aqua Dual E2363を用いて電池寿命の調査を行った. 両モデルとも, 通常モードと緊急時モー

ドの切り替えに10秒以上掛かり、緊急時モードではサードパーティ製のアプリを起動できないなど、実用性は決して高くない。さらに、Galaxy S6 edgeでは、消費電力が大きいためかカメラも利用できない。これらの機種は内蔵バッテリーを取り外せず正確な消費電力を測定できないため、ここでは消費電力の下限を見積もるためにメーカ公称の電池寿命から消費電力を割り出した。Galaxy S6 edgeの緊急時モードでの公称電池寿命は17.5日であり、平均消費電力は23.5 mWとなる。従って、提案手法が有利となるデューティサイクルの損益分岐点は50.0分である。Xperia M4 aquaについては、公称電池寿命が22.8日であり、損益分岐点は70.4分である。これらの損益分岐点は、我々の想定する利用シナリオでのセンシング間隔に比して十分に短く、提案手法は従来の省電力モードと比べても省電力である。

また、Huaweiなど幾つかのメーカは、スマートフォンを電源オフ状態で待機させ、指定した時刻に再び起動する機能を実装している。しかしながら、このタイマー起動機能はWakeDroidの優位性を脅かすものではない。まず、このタイマー起動機能はユーザや外部デバイスからのセンシング指令を認識し対応することができない。次に、起動時刻のスケジュール設定は柔軟でなく、1日または1週間以内に限定される。さらに、このタイマー起動機能の実装には、特定のCPUと特別なハードウェア構成が必要である。我々の知る限りでは、世界最大のスマートフォン向けCPUベンダQualcommが製造するSnapdragonシリーズのCPUや、世界最大のスマートフォンメーカSamsungが製造するExynosシリーズのCPUではサポートされていない。幾つかの中堅CPUベンダはこのタイマー起動機能をサポートしているものの、特定のCPUと特別なハードウェア構成が必要である。Antutuの調査によると、QualcommのSnapdragonシリーズとSamsungのExynosシリーズを合わせた世界シェアは2016年現在で70%を超えており、タイマー起動機能は幅広い機種に備わった一般的な機能とは言えない状況である[40]。

#### 2.4.5 セキュリティの確保

ネットワークに接続されたあらゆる情報機器は、常にセキュリティ上の脅威に曝されている。センサノードとして活躍する中古スマートフォンも例外ではなく、携帯電話網の packets 通信やWi-Fi通信を介して攻撃を受ける可能性がある。また、ネットワークに接続されていない機器でも、リムーバブルメディアやBluetooth接続を介して攻撃を受けた事例が報告されている[41, 42]。

センサノードとして利用されるスマートフォンは、通常用途のスマートフォンとは異なりアプリストアを利用しないため、マルウェアアプリが混入する可能性は低いものの、OSレベルの脆弱性は避けられない。このような脆弱性を修正しセキュリティを確保するため、各OSベンダはセキュリティパッチを定期的に配信している。Android OSの場合、リファレンスモデルであるNexus/Pixelシリーズに対しては、発売後3年間に渡って月例パッチやOSアップデートが配信されている。一方で、一般のAndroid端末に対するパッチ配信は製造メーカに委ねられており、その履行率は必ずしも高くない(表2.5)。また過去には、販売チャネルを提供する携帯電話事業者が、端末の性能不足を理由に国内でのOSアップデートの配信を拒否した例も報告されている。

他方で、Android OSの派生ディストリビューション(カスタムROM)の開発コミュ

表 2.5: Android スマートフォン向けのセキュリティパッチの配信状況.

機種名	メーカー	発売日	最終パッチ配信
GALAXY S5 SC-04F	Samsung	2014/05	2017年8月
Xperia Z3 Compact SO-02G	SONY	2014/11	2017年4月
Galaxy S6 edge SC-04G	Samsung	2015/04	2017年4月
ARROWS NX F-04G	富士通	2015/05	2017年2月
Xperia Z5 SO-01H	SONY	2015/10	2017年9月
Galaxy S7 edge SC-02H	Samsung	2016/05	2017年6月
Xperia XZ SO-01J	SONY	2016/11	2017年10月
V20 PRO L-01J	LG	2017/02	2017年8月

ニティでは、古いハードウェアを広くサポートする動きが見られる。例えば、Lineage OS (旧 CyanogenMod) は、2011年2月発売の GALAXY S2 や 2012年9月発売の Xperia V から、2017年2月発売の LG G6 に至るまで、非常に幅広い端末に対して Android 7.1 ベースの OS イメージを提供している。従って、メーカーによるサポートが既に打ち切られた端末については、純正の Android OS を使い続けるより、むしろサードパーティ製のカスタム ROM を導入した方が安全性を高く保ちやすい。

センサノードとして活躍する中古スマートフォンについても、セキュリティを確保するためには OS を最新状態に保つことが望ましい。より具体的には、センサノードに転用する際にカスタム ROM の最新バージョンを導入し、数ヶ月ごとの充電作業の際に OS アップデートを行うことが好ましい。また、緊急性の極めて高いパッチについては、携帯電話網を介して OTA アップデートを行うことも可能である。

## 2.5 関連研究

この節では、スマートフォンを無線センサノードとして活用するための既存のアプローチを紹介する。これらの提案のほとんどは、対象のスマートフォンが所有者とともに行動し、日々の生活の中で所有者が定期的にスマートフォンを充電することを暗黙の前提としている。これとは対比的に、我々の提案する WakeDroid では、対象のスマートフォンがセンシング用途のみに使用されることを想定している、という大きな違いがある。しかしながら、これらの既存手法の中には、我々の手法と組み合わせることで、スマートフォンの機能性を高めたり、電池駆動時間を延長したりできるものも含まれる。

本稿の冒頭でも指摘した通り、参加型センシングは Burke らが初めて提唱した概念である [2]。彼らは携帯電話機の持つ余剰能力に着目し、その能力をセンシング活動に充てることを提案した。スマートフォンを含め携帯電話機を用いた参加型センシングは、多くのユーザを低コストで集められるが、カバーエリアが確率的で確実性に欠けるという問題がある。Medusa は参加型センシングにおいて、主催者と参加者の間を取り持ち、センシングタスクの分配を補助するフレームワークである [7]。Medusa には対象のスマートフォンを電源オフ状態から起動する機能がないため、WakeDroid と組み合わせることで相補的な役割を果たせる。Medusa と同様のアプローチを取る研究は幾つかあり、いずれも WakeDroid

とは相補的關係にある [4, 5].

Piggyback Crowd Sensing (PCS) はセンシングタスクの実行に起因するスマートフォンの消費電力の増加を抑える手法である [8]. PCS は, 所望のセンサが他のアプリによって利用されているタイミングでセンシングを行うことで, センサの消費電力増を実質ゼロにする. 我々の想定シナリオにおいても, 複数のタスクを1台のスマートフォンでこなす場合には, この技術を応用できる.

位置測位に関しては, LEAP などの低電力 GPS 測位システムを適用することで, 消費電力を抑えつつ高精度な測位を実現できる [24]. LEAP は GPS 信号処理のパイプラインを分割し, 最も計算負荷の高い部分をクラウドサーバにオフロードすることで, 消費電力を最大 80%削減する. LEAP 以外にも, 同様のアプローチで位置測位を行う手法が提案されている [25, 26]. 前述の通り, WakeDroid では給電波にエリア ID を埋め込むことで, GPS と比べて桁違いに少ない電力で位置特定が可能である.

これに加えて, スマートフォンのグラフィックス [43], ストレージ I/O [44], ヘテロジニアスなマルチコア CPU [45] などに着目した省電力化手法が提案されており, WakeDroid と組み合わせることで, さらなる消費電力の削減を期待できる.

## 2.6 あとがき

本章では, 特に商用電源の利用しにくい環境において専用センサノードとして稼働するスマートフォン向けに, 外付け電源管理システム「WakeDroid」を提案した. WakeDroid は, Android スマートフォンを充電器に接続した際に自動的に起動する電源状態管理ソフトウェアを拡張し, USB 端子に接続する USB ドングル型デバイスとして電源管理機能を実現する. WakeDroid の USB ドングルには2種類あり, 一方は, 無線電力伝送のためのレクテナを備え, 遠隔操作による Android スマートフォンの電源投入を実現する. 他方は, 乾電池とマイコン, センサを搭載し, そのセンサ値や電池残量に応じて, スマートフォンの起動間隔を調整する. 我々の評価実験では, マイクロ波給電方式の USB ドングルを用いることで, 出力 0.48 W のマイクロ波給電源から 35 m 離れた地点にあるスマートフォンの起動プロセスを開始させることに成功した. また, イベントドリブン方式の USB ドングルを用いることで, 9ヶ月もの間スマートフォンを待機させておき, 必要な時に利用可能であることを示した. このように, WakeDroid は中古スマートフォンにセンサノードとしての「第二の人生」を与えることで, スマートフォンセンシングのコンセプトを大幅に拡張する.

## ■ 第3章

---

# Linux カーネルを活用した起 動時間の短縮

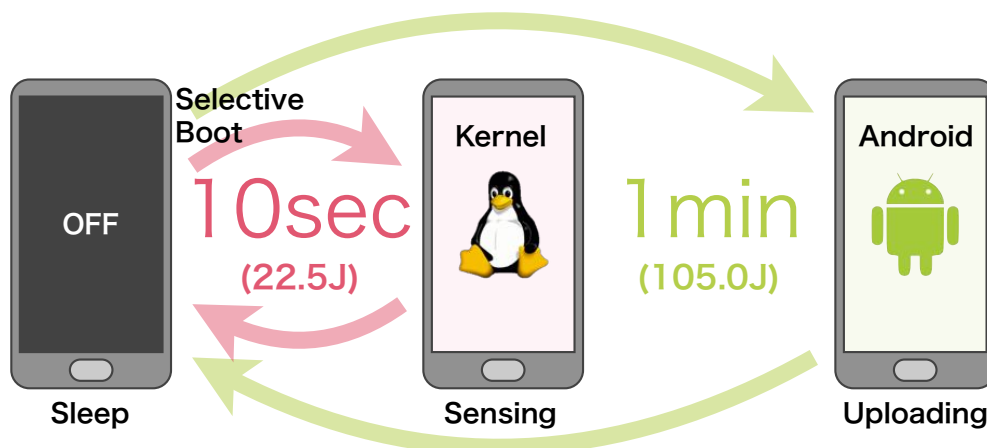


図 3.1: 提案コンセプト. 本手法では, Android OS の基礎部分に採用されている Linux カーネルのみを使用することで, スマートフォンの起動プロセスを大幅に短縮する. これにより, 起動プロセスで消費されるエネルギーを大幅に削減し, 間欠駆動時の電池寿命を飛躍的に向上させる.

### 3.1 まえがき

前章では, 中古スマートフォンを電源オフ状態で待機させることで電池駆動時間を飛躍的に向上させ, スマートフォンセンシングのコンセプトを大きく拡張する技術の提案を行った. しかしながら, スマートフォンのように高機能な OS で動作するデバイスを間欠駆動させる際には, OS を起動したりシャットダウンしたりする時間的, 電力的なオーバーヘッドが生じる. 本章では, このオーバーヘッドを可能な限り短縮する手法の検討を行う.

本章で提案するのは, Android OS の基礎部分に採用されている Linux カーネルを活用し, スマートフォンの起動およびシャットダウンのプロセスを大幅に短縮する技術である. これを前章で提案した WakeDroid と組み合わせることで, ごく短い時間で完了するセンシングタスクを, 間欠駆動するスマートフォン上で実行する際のエネルギー効率を改善する. 本手法を適用することで, OS の起動プロセスにおける消費エネルギーを 75% 以上削減でき, 間欠駆動時の電池寿命を 4 倍強に延長できる. これにより, スマートフォンを商用電源を得られない環境に設置し, 内蔵バッテリーの電力のみで運用する場合でも, ごく単純なセンシングタスクを数ヶ月に渡って繰り返し実行することが可能となる. また, 本手法では, 外部センサとの連携に使用できる Bluetooth や USB インタフェースもサポートしている. さらに, より高度な処理を行うために, Linux カーネルのみを動作させる状態と, Android OS を完全に起動した状態を自由に行き来させることもできる.

本章の構成は以下の通りである. まず 3.2 節で Android OS を構成する各要素について説明した後, 3.3 節で本提案手法を構成する要素技術を 5 つ紹介する. 次に 3.4 節で通常の Android OS を起動する場合と Linux カーネルのみを起動する場合の消費電力の計測結果を紹介する. さらに 3.5 節で典型的なセンシングシナリオに沿って電池寿命の推定を行い, 我々の提案手法の有用性を示す. その後 3.6 節でスマートフォンの機種依存などについて議論した後, 3.7 節で関連研究を紹介して, 3.8 節で本章の内容をまとめる.



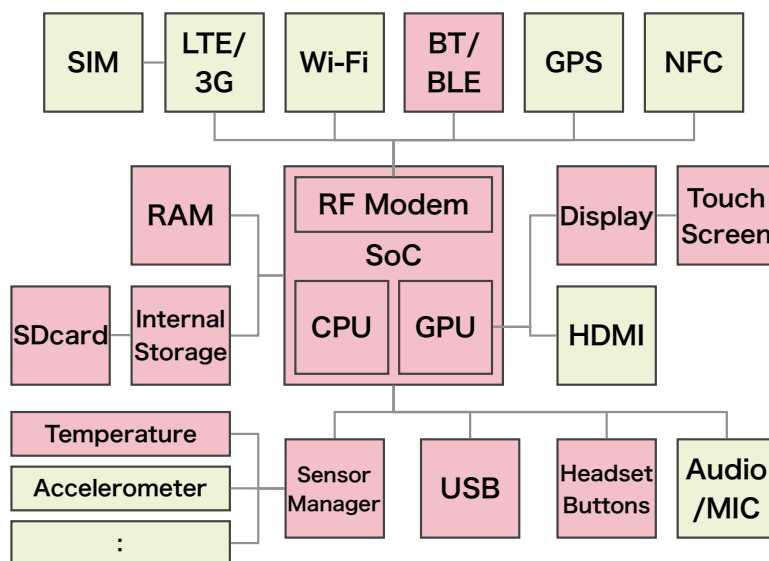


図 3.2: 提案手法（赤）と通常の Android OS（赤+緑）で利用可能なスマートフォンの機能構成。提案手法では、センシング活用に必要十分な機能セットとして、Bluetooth や USB インタフェース、ディスプレイ、ストレージ、センサなどをサポートする。

## 3.2 Android OS の構造

Android OS はスマートフォンやタブレット端末向けの OS プラットフォームであり、IDC の統計によるとスマートフォン市場におけるシェアは 2017 年初頭に 85.0% に達しており、2 位の iOS (14.7%) を大きく引き離している。両 OS を比較すると、iOS が垂直統合モデルを採用して特定のハードウェアに最適化されているのに対し、Android OS は多種多様なハードウェアに適合するように設計されており、リファレンスモデルとして Nexus/Pixel シリーズが提供されている。また、Android OS はソースコードが公開されており改変しやすいことから、モバイル分野の研究においては Android 端末が広く利用されてきた。

図 3.3 に示すように、Android OS は Linux カーネル、ハードウェア抽象化レイヤ (HAL)、ネイティブライブラリ、Java 仮想マシン、アプリケーション・フレームワーク、システムアプリ、および、2 種類のメンテナンス用ユーティリティから構成される。これらの構成要素は、原則として Apache 2.0 ライセンスでソースコードが公開されており、自由に改変して手持ちの端末にインストールすることがユーザの権利として公式に認められている [37]。ただし、Gmail や Google Maps といった Google のプロプライエタリなアプリケーションや一部のハードウェアドライバについては、ソースコードが非公開とされており、バイナリのみが提供されている。

### 3.2.1 Linux カーネル

OS の最も基礎的な部分を司る Linux カーネルについては、Android 4.0 未満ではバージョン 2.6.x、Android 4.0 以上 7.0 未満ではバージョン 3.x、Android 7.0 以上ではバージョン 4.x をベースに、独自のカスタマイズを施したものが採用されている。Android 4.0 未

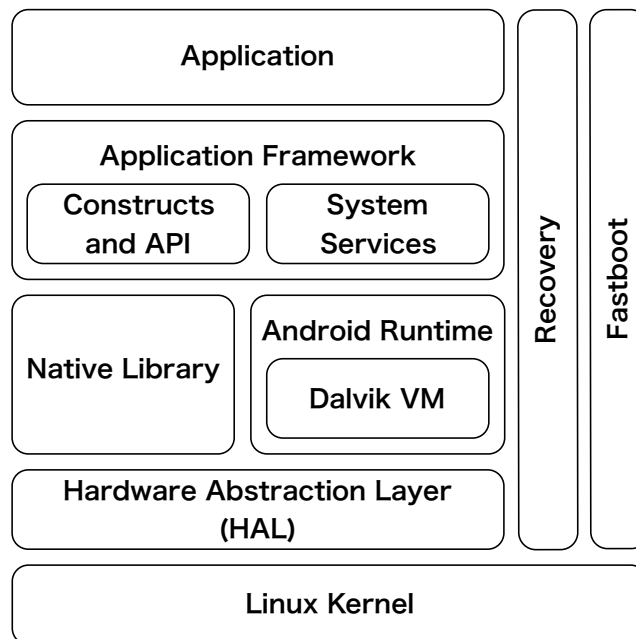


図 3.3: Android OS のアーキテクチャ概要.

満では、ディスク容量の厳しい制約に対応するため、cp や mv といった基本的なコマンドさえも削除されており、Linux 端末として満足に利用することは難しい。また、Android 5.0 以上では SELinux が有効化されており、root 権限を濫用できないよう制限が加えられている。

Linux カーネルはメモリ管理やプロセス管理を司るだけでなく、後述するようにアプリ毎に個別の UID を割り当てることでアプリ間の独立性を確保する役割も担っている。また、Android OS の起動プロセスにおいては、各種ドライバの読み込み、各種ハードウェアの初期化を行った上で、Android フレームワークの init プロセスを呼び出す役割を担う。さらに、端末を電源オフ状態で充電器に接続した際に表示される画面表示を司るのも Linux カーネルの役割である。

### 3.2.2 ネイティブライブラリ

ネイティブライブラリは、OpenGL や WebKit, SQLite, SSL といった、アプリケーション・フレームワークを支える基本的な機能を提供する。これらのライブラリは C/C++ 言語で実装され、Linux カーネル上のネイティブ環境で実行されるため、後述するアプリケーション・フレームワークより処理が高速な傾向がある。アプリ開発者は、Android NDK (Native Development Kit) を用いることで、アプリケーション・フレームワークを通すことなく、直接ネイティブライブラリを利用できる。

### 3.2.3 Java 仮想マシン

Java 言語で記述されたアプリの実行環境としては、Android 4.4 以下では Dalvik VM、Android 5.0 以上では ART (Android RunTime) が採用されている。いずれも、Android OS のために独自開発された仮想マシンであり、低メモリ環境に最適化されている、これらの仮想マシンは、複数のインスタンスが並行して動作できるよう設計されており、Android アプリはそれぞれ独立した仮想マシン上で実行される。Linux カーネル上では、各アプリに対応する UID が作成され、その UID を用いて各仮想マシンを実行することで、アプリ間の独立性やファイルのアクセス管理といった機能を実現している。

旧来の Dalvik VM が JIT (just-in-time) コンパイラを採用し、アプリの実行時にコンパイル処理を行っていたのに対して、ART は新たに AOT (ahead-of-time) コンパイラを採用し、アプリのインストール直後および OS のアップデート直後にコンパイル処理を行う。その際、コード全体を見渡した極めて強力な最適化が行われるため、アプリの実行速度が大きく向上している。

表 3.1 に、Java 言語と C 言語でのベンチマークプログラムの処理時間の比較を示す。このベンチマークプログラムには、空間計算量の多い整数演算処理として素数の探索が、時間計算量の多い浮動小数点演算処理として円周率の数値計算が含まれている。なお、ソースコードは可能な限り共通化されており、gcc でのコンパイル時には O3 オプションで最適化を行った。Java 言語と C 言語で処理時間を比較すると、Dalvik VM を採用する Android 4.4 以下の機種では、C 言語が Java 言語より平均で 3.3 倍高速であった。一方、ART を採用する Android 5.0 以上の機種では、実行速度に大きな差は見られず、円周率の数値計算においては一部の機種で C 言語の性能が Java 言語の性能を下回る逆転現象が見られた。

### 3.2.4 アプリケーション・フレームワーク

アプリケーション・フレームワークは、アプリ開発者に対して様々な Android API を提供する役割を担う。このフレームワークは Java 言語で実装されており、画面表示からネットワーク通信、アプリ間連携に至るまで、幅広い機能を提供する。後述するシステムアプリの実装にも全く同じフレームワークが利用されており、アプリ開発者は全ての機能を利用できる。なお、端末ベンダによるアプリケーション・フレームワークの変更は黙認されており、API が Android SDK での規定通りに動作しない端末も稀に存在する。また、Android OS のバージョンによっても API の仕様が微妙に異なるため、アプリ開発者がバージョン毎の違いを管理しやすいように、Android の各バージョンにはアルファベット順のコードネームが付与されている。

### 3.2.5 システムアプリ

Android OS には、ホーム画面アプリや電話アプリ、ブラウザアプリ、設定アプリといったシステムアプリが含まれている。これらのアプリは、前述のアプリケーション・フレームワークを用いて実装されており、原則として特権的な機能は使用していないため、サードパーティ製のアプリで置き換えることができる。唯一の例外は設定アプリであるが、これはサードパーティ製のアプリからコールできない Android API が一部存在し、設定ア

表 3.1: Java 言語と C 言語でのベンチマークプログラムの処理速度の比較. Java 言語での処理速度を基準とした C 言語での処理速度を倍率で示す.

機種名	OS	CPU	素数探索	円周率計算
HTC J butterfly HTL21	4.1	APQ8064	2.46 倍	1.91 倍
ARROWS Kiss F-03E	4.1	MSM8960	1.63 倍	4.10 倍
ARROWS ef FJL21	4.1	MSM8960	2.51 倍	5.38 倍
Optimus it L-05D	4.1	MSM8960	2.35 倍	4.63 倍
Optimus LIFE L-02E	4.1	MSM8960	2.27 倍	4.60 倍
MEDIAS X N-07D	4.1	MSM8960	1.81 倍	3.23 倍
MEDIAS U N-02E	4.1	MSM8960	2.72 倍	2.79 倍
MEDIAS W N-05E	4.1	MSM8960	2.81 倍	2.87 倍
VEGA PTL21	4.1	MSM8960	2.43 倍	5.83 倍
GALAXY S3 SC-06D	4.1	MSM8960	2.33 倍	5.36 倍
GALAXY S3 Progre SCL21	4.1	MSM8960	2.41 倍	5.57 倍
AQUOS PHONE SERIE SHL21	4.1	MSM8960	2.70 倍	5.46 倍
Xperia GX SO-04D	4.1	MSM8960	1.90 倍	3.38 倍
Xperia SX SO-05D	4.1	MSM8960	2.13 倍	5.09 倍
Xperia AX SO-01E	4.1	MSM8960	2.44 倍	5.70 倍
Xperia VL SOL21	4.1	MSM8960	2.47 倍	5.56 倍
URBANO L01	4.2	MSM8960	1.90 倍	5.80 倍
ARROWS Z FJL22	4.2	MSM8974	2.55 倍	4.04 倍
G2 L-01F	4.2	MSM8974	2.80 倍	2.79 倍
G Flex LGL23	4.2	MSM8974	2.47 倍	2.88 倍
SH-01F DRAGON QUEST	4.2	MSM8974	2.59 倍	4.28 倍
GALAXY Note 2 SC-02E	4.3	Exynos4412	1.39 倍	6.43 倍
isai LGL22	4.4	MSM8974	2.53 倍	3.22 倍
AQUOS CRYSTAL 305SH	4.4	MSM8926	2.79 倍	6.61 倍
AQUOS SERIE SHL25	4.4	MSM8974AB	2.37 倍	3.74 倍
Xperia Z1 f SO-02F	4.4	MSM8974	2.50 倍	4.68 倍
Xperia Z Ultra SOL24	4.4	MSM8974	1.02 倍	1.17 倍
HTC J butterfly HTV31	5.0	MSM8994	1.20 倍	1.04 倍
GALAXY S4 SC-04E	5.0	APQ8064T	1.10 倍	0.89 倍
GALAXY Note 3 SC-01F	5.0	MSM8974	1.04 倍	0.85 倍
GALAXY J SC-02F	5.0	MSM8974	1.07 倍	0.91 倍
Xperia Z3 SOL26	5.0	MSM8974AC	1.13 倍	0.91 倍
Nexus 5	6.0	MSM8974	1.19 倍	1.25 倍
Nexus 6	6.0	APQ8084	1.61 倍	1.43 倍
Galaxy S6 edge SCV31	6.0	Exynos7420	1.10 倍	1.87 倍

表 3.2: Android OS における特権パーミッションの例

パーミッション名	概要
DUMP	システムの内部状態の取得
FORCE_STOP_PACKAGES	他のアプリの強制停止
GET_APP_OPS_STATS	アプリの操作状況に関する統計情報の取得
HARDWARE_TEST	ハードウェア試験
MANAGE_USB	USB デバイスの設定と許可の管理
MASTER_CLEAR	端末の初期化
MOUNT_FORMAT_FILESYSTEMS	リムーバブルストレージ内のデータ消去
MOUNT_UNMOUNT_FILESYSTEMS	リムーバブルストレージのマウント/マウント解除
READ_LOGS	機密ログデータの読み取り
REBOOT	端末の再起動
WRITE_SECURE_SETTINGS	システムのセキュリティ設定の変更
WRITE_MEDIA_STORAGE	リムーバブルストレージ内の変更/削除
WRITE_APN_SETTINGS	APN 設定の変更

プリがこれを利用しているためである (表 3.2)。これらの API を利用するには、OS のビルド時に付与されたデジタル署名とアプリのコンパイル時に付与するデジタル署名を一致させる必要がある。従って、これらの Android API を利用したい場合には、Android OS を自前でビルドして予め端末にインストールしておく必要がある。

### 3.2.6 メンテナンス用ユーティリティ

Android OS には、3.2.2~3.2.5 節で紹介した構造とは独立に、2 種類のメンテナンス用ユーティリティが含まれている。これらのユーティリティは、OS 本体を起動せずに Linux カーネル上でファイルシステムをマウントし、OS アップデート (システム領域の書き換え) や端末の初期化 (ユーザデータの消去) といった処理を行う。

また、Android の開発元である Google は、Linux カーネルにおける root 権限の取得 (root 化) や、サードパーティ製の OS イメージ (カスタム ROM) のインストールを、ユーザの権利として公式に認めており、メンテナンス用ユーティリティは、これらの操作を行うためのツールとしても機能する。カスタム ROM は、2017 年現在で 40 種類以上が公開されており、カスタム ROM をプリインストールして出荷される端末も出現している [38, 39]。

### Recovery モード

Recovery モードは、PC などの外部機器によるサポートを必要とせず、Android 端末が単独で実行できるメンテナンス用ユーティリティである。適切な権限を持った Android アプリケーションから所定の API をコールすることで、Recovery モードに遷移することが可能であり、Android OS のアップデート適用や Android 端末の初期化などに利用される。多くの Android 端末にプリインストールされている Recovery モードはタッチ操作やボタン

入力を受け付けないが、ClockworkMod や TWRP といったサードパーティ製の Recovery モードは GUI 操作に対応している。これを利用すると、root 権限を取得するツールの実行やカスタム ROM のインストールが可能となる。なお、カスタム ROM のインストールに際しては、後述する Fastboot モードで事前にブートローダアンロックを行っておく必要がある。

### Fastboot モード

Fastboot モードは、USB ケーブルで接続した PC の制御下で実行できるメンテナンス用ユーティリティである。ソフトウェア上のトラブルにより Android OS や Recovery モードで端末を起動できなくなった場合には、Fastboot モードで端末を起動してファクトリーイメージを書き込むことで、端末を工場出荷状態に戻すことができる。特に、Nexus/Pixel シリーズについては、OS バージョン別のファクトリーイメージが公開されていることから、OS のアップグレード・ダウングレードを自由に行える。また、Fastboot モードでブートローダアンロックを行うことで、サードパーティ製の OS イメージを端末にインストールできるようになる。なお、操作元の PC には Android SDK に付属する制御用コマンドをインストールしておく必要がある。

## 3.3 要素技術

前節で紹介した通り、Android OS は Linux カーネル上で動作するミドルウェアであり、起動とシャットダウンには合わせて 1 分前後の時間が掛かるが、これは Linux カーネルの起動後に Dalvik 仮想マシンや種々のシステムサービスの初期化処理を行うためである。Linux カーネルの起動プロセス自体は多くの Linux ベースの OS と類似しており、Nexus 5 の場合、起動プロセスは 10 秒以内に、シャットダウンプロセスは 1 秒以内に完了する (図 3.1)。また、デバイスに搭載された基本的な機能の多くは、Linux カーネルが起動した時点で利用可能である。我々は、このオーバーヘッドの短さに着目し、省電力化を求められる間欠的なセンシング活動に Linux カーネルを活用することを提案する。オーバーヘッドの短さに加えて、Linux カーネルのみを起動した状態では、Android アプリがバックグラウンドで動作することがなく、Android OS の実行時より消費電力が安定することも利点である。起動プロセスの短い Linux カーネルであるが、温度センサや Bluetooth インタフェース、USB インタフェースなどセンシング活動において重要なハードウェアを制御できる。スマートフォン上の全てのハードウェアは Linux カーネル上にマウントされており、カーネルレベルのインタフェース (`/dev/*`, `/sys/devices/*`) を通して Android フレームワークから操作されるため、原理上は全てのハードウェアを Linux カーネル上から利用可能である。本研究では、センシング活動において重要度の高いハードウェアから優先的に順次開拓を行い、現時点で図 3.2 に示す範囲を Linux カーネル上から利用可能としている。

本研究では、Nexus 5 用のファクトリーイメージとそのソースコード (OS バージョン 5.1.0, build LMY47I) を基に、Linux カーネルのみを起動してセンシングを行うフレームワークの実装を行った。ファクトリーイメージは、表 3.3 に示す 5 つのファイルから構

表 3.3: Nexus 5 用の Android 5.1.0 (build LMY47I) のファクトリーイメージを構成するファイル群

ファイル名	概要
boot.img	ブートパーティションのディスクイメージ。このパーティションがないとデバイスを起動できない。このイメージには Linux カーネルと、起動直後に RAM に読み込まれるデータ (ramdisk) が含まれる。
cache.img	キャッシュパーティションのディスクイメージ。この領域には、頻繁にアクセスするアプリのデータやシステムコンポーネントが一時的に保管される。
recovery.img	リカバリーモードのディスクイメージ。なお、リカバリーモードは、初期化やソフトウェアアップデートなどデバイスのメンテナンス操作を行うためのツールである。
system.img	Linux カーネルと ramdisk の部分を除く Android OS 全体のディスクイメージ。このイメージには、Android のミドルウェア部分とプリインストールアプリなどが含まれる。
userdata.img	データパーティションのディスクイメージ。この領域には、インストール済みアプリなどのユーザデータが保管される。

成される。ここでは、Linux カーネル部分が含まれる `boot.img` のみ変更を加えたため、Android OS の機能は完全に温存されており、必要に応じて利用できる。ほとんどの変更点は、Linux カーネルの起動プロセス中に実行される `init.rc` などのブートスクリプトに対するものである。このスクリプトにセンシングタスクを書き足すことで、ユーザは Android デバイスの電源を投入するだけでセンシングタスクを実行できる。スマートフォンの電源投入については、電源ボタンを押下するのが一般的であるが、ここでは前章で提案した USB 端子への電圧印加方式による電源投入を想定する。

ユーザが Android OS の機能や Android アプリの機能を必要とする場合は、ブートスクリプト内で条件分岐して Android OS を起動することも可能である。Android OS 向けの Linux カーネルには、表 3.4 に示すように、`init.rc` 内にいくつかのイベントトリガが用意されている。我々の実装では、ソースコード 3.1 に示すように、`on boot` のイベントトリガで実行されるデーモンを用意し、そこでセンシングタスクを実行することとした。

本節の以下の部分では、本手法を実現する要素技術を順に紹介する。

### 3.3.1 起動モードセレクト

我々のフレームワークでは、状況に応じて Linux カーネルと Android OS を使い分ける。典型的な利用例は以下の通りである。平常時には Linux カーネルのみを起動してセンシングを行い、または、BLE 通信で周囲のセンサノードとデータをやりとりする。観測したい現象の予兆を示すようなセンサデータが得られた時にだけ、Android OS を起動してセンサデータの詳しい分析を行い、または、管理者に警告メッセージを送信する。この機能を

表 3.4: Android Open Source Project (AOSP) [37] で公開されている, Android 5.1.0\_r1.0 のソースコード内の `init.rc` で定義されているイベントトリガの一覧.

イベントトリガ	概要
<code>on early-init</code>	Linux カーネルの起動後, 最初に呼び出されるイベントトリガ.
<code>on init</code>	<code>on early-init</code> の直後に呼び出される, 二番目のイベントトリガ.
<code>on late-init</code>	以下のイベントトリガを呼び出すイベントトリガ.
<code>on post-fs</code>	ファイルシステムが利用可能になった時に呼び出されるイベントトリガ.
<code>on post-fs-data</code>	<code>on post-fs</code> の直後に呼び出されるイベントトリガ.
<code>on boot</code>	Android をブートする準備ができた時に呼び出されるイベントトリガ.
<code>on nonencrypted</code>	暗号化されたファイルシステムが復号された時に呼び出されるイベントトリガ.
<code>on charger</code>	電源オフ状態で充電器に接続された時に呼び出される特別なイベントトリガ.

実現するため, 我々は起動モードセクタの実装を行った (ソースコード 3.1, 3.2).

この起動モードセクタは, ブートスクリプト (`init.rc`) から呼び出される. 条件分岐のための判断材料としては, スマートフォン上のセンサで取得したセンサデータ, 周囲の BLE 機器から受信したデータ, バッテリー残量などのシステム情報を利用可能である. また, 起動モードとしては, Linux カーネルのみの起動, Android OS の起動, リカバリーモードでの起動の 3 つを選択できる. なお, リカバリーモードは, デバイスの初期化やソフトウェアアップデートに有用である.

ソースコード 3.1: `/init.rc`

```

1  import /init.envIRON.rc
2  import /init.usb.rc
3  import /init.${ro.hardware}.rc
4  import /init.${ro.zygote}.rc
5  import /init.trace.rc
6
7  * snip *
8
9  + # trigger the main class daemons
10 on boot
11     class_start main
12
13     * snip *
14
15     ## Daemon processes to be run by init.
16     ##
17 + # register KISS (Kernel-Inspired SenSing) daemon as root user
18 + service kissd /sbin/kiss.sh
19 + class main
20 + user root
21 + oneshot
22
23     * snip *
```



ソースコード 3.2: /sbin/kiss.sh

```
1  #!/system/bin/sh
2
3  # check the length of sensor data
4  length='grep -c '' /data/output.txt'
5
6  # if less than 100 lines, just shutdown
7  if [ $length -lt 100 ]; then
8    /system/bin/reboot -p
9  fi
10
11 # check the battery level
12 battery='cat /sys/class/power_supply/battery/capacity'
13
14 # if less than 10%, just shutdown
15 if [ $battery -lt 10 ]; then
16   /system/bin/reboot -p
17 fi
18
19 # scan BLE devices
20 su -c "setprop_init.svc.bluetoothd_running"
21 sleep 1
22 su -c "hciconfig_hci0_up"
23 su -c "/sbin/busybox_timeout_t_1_s_SIGINT_hcitol_lescan_/data/BLE.
    txt"
24
25 # if device is absent, remote wipe in recovery
26 length='/system/bin/grep AA:BB:CC:DD:EE:FF /data/BLE.txt | /system/bin/
    grep -c '''
27 if [ $length -gt 0 ]; then
28   /system/bin/reboot recovery
29 fi
```

### 3.3.2 ディスプレイの制御

本フレームワークでは、対象となるスマートフォンのディスプレイをタッチスクリーンの機能も含めて制御できる。具体的には、画面に任意の画像を表示し、ユーザや周辺環境からのタッチ操作を受け付けることができる。これにより、我々はデバイスの設定のためのインタフェースやアラート表示を行うことができる。また、防水筐体を持つスマートフォンであれば、タッチスクリーンの機能を応用して、農地などの野外環境において降雨や降霜の検知を行うことも考えられる。

先行研究によると、スマートフォンの消費電力の大部分はディスプレイモジュールによるものである [16]。この先行研究は、スマートフォンが Android OS を起動した状態で日常利用に供されることを暗黙の前提としているが、Linux カーネルのみを起動した状態でもディスプレイモジュール自体の消費電力は大差ない。それゆえ、システム全体の消費電力を抑えるには、ディスプレイモジュールの消費電力を抑えることが非常に重要である。単にディスプレイを消灯するだけでも、理論上は電池寿命を倍増させることができる。我々の予備実験の結果によると、Nexus 5 のディスプレイモジュールは中程度の画面輝度で約 2.2W の電力を消費しており、これはアプリを実行していない待受状態のスマートフォンの消費電力の 90%以上を占める。

液晶ディスプレイを搭載するデバイスについては、バックライトが消費電力に最も大きく寄与するため、Linux カーネル上のインタフェース (/sys/class/leds/lcd-backlight

/brightness) を通してバックライトの消灯制御を行う。一方、有機 EL ディスプレイを採用する機種については、バックライトが存在せず各画素が自発光する仕組みのため、ブートアニメーションを黒一色の画像に置き換えることで、この制御を代用できる。

### 3.3.3 Bluetooth Low Energy 通信

Bluetooth Low Energy (BLE) は、消費電力が非常に少なく優れた近距離無線通信規格の 1 つである。2010 年に Bluetooth 4.0 として標準化されて以来、ほとんどの Android スマートフォンが BLE をサポートするようになった。BLE 通信は、従来の Bluetooth 通信とは異なり、事前のペアリング作業や明示的な接続の確立が不要であり、通信時のオーバーヘッドが少ない。この特徴は、間欠的なセンシング活動にも適するものであり、周囲のセンサノードとの通信に有用である。

本フレームワークで BLE の機能を Linux カーネル上から利用可能とするため、我々は BlueZ プロジェクトから制御コマンド群を導入した [46]。このコマンド群を用いることで、Android デバイス上の Bluetooth インタフェースを明示的に有効化・無効化したり、周辺の Bluetooth 機器を探索したりできる。なお、従来の Bluetooth 規格での機器探索には数秒から十数秒の時間が掛かるが、BLE での機器探索は 1 秒強で完了するため、間欠駆動と相性が良い。

### 3.3.4 USB インタフェース

我々のフレームワークでは、周辺機器との接続に USB インタフェースを用いることもできる。実装面では、Android OS のデバッグ用インタフェースである Android Debug Bridge (ADB) を活用した。このインタフェースは Linux カーネルの起動直後に利用可能となり、USB インタフェース経由でのシリアル通信を受け付ける。このインタフェースを通して、周辺機器は Android スマートフォンのストレージ上のファイルを読み書きしたり、カーネル上のネイティブコマンドを実行したりできる。

Android 4.2.2 以降の Android OS に実装されている ADB インタフェースは、セキュリティ上の理由により接続元の機器をホワイトリスト方式で管理している。接続元の機器がホワイトリストに登録済みかどうかを判定する部分で、周辺機器に対して RSA-SHA1 署名を求めるため、Arduino など計算能力の貧弱なマイコン機器とは相性が悪かった。この仕様は、アプリ開発者のスマートフォンに保存された個人情報への不正アクセスを防ぐためのものであり、専用センサノードとして使用する機器には必要がない。そこで、我々はこの機能を無効化することで、マイコン機器からの接続を容易にした。

### 3.3.5 ボタン入力

我々のフレームワークは、スマートフォン本体に備わった物理ボタンおよび 3.5 mm オーディオジャックに接続されたボタンからの入力も受け付けることができる。スマートフォン用ヘッドセットの配線には 2 種類あり、それぞれ 4 つのボタンをサポートしているが、ここでは両方式のヘッドセットからの入力を認識できる。このボタン入力は、ごく簡素な

周辺機器との意思疎通に応用できる。前節で紹介した USB 通信の場合、周辺機器からスマートフォンに対して直流 5 V の給電が必要であるのに対し、オーディオジャックのボタン入力は所定の端子間の抵抗値を変化させるだけで良く、シリアル通信に対応したハードウェアも必要としない。ボタン入力を応用した通信の最大スループットについては、認識可能な入力シンボル長に依存するため、機種によって異なる。我々の実験結果によると、Nexus 5 の場合、押下状態は 35 ms 以上、押下間隔は 80 ms 以上を必要とする。複数のボタンを同時には認識できないこと、ボタンが 4 種類あることを考慮して、ごく単純なエンコーディングを行った場合、通信速度は約 20 bps となる。

### 3.4 消費電力の計測

我々は、Nexus 5 を用いて Linux カーネルおよび Android OS を起動する際の消費電力の計測を行った。この実験では、Nexus 5 の内蔵バッテリーを取り外し、バッテリー接続端子に直流安定化電源から 4.1 V の電圧を印加しつつ、マルチメータ (Keysight 34410A) で電流量を計測した。なお、ここで印加電圧を 4.1 V としたのは、リチウムイオン電池の満充電時の最大出力電圧に合わせたためである。各条件について 5 回ずつ計測を行い、グラフでは各回の結果を異なる色で示す。

図 3.4 は、Nexus 5 で Linux カーネルのみを起動した場合の消費電力の推移である。平均で 10.7 秒の時間が掛かり、その間の消費電力は平均 2.11 W、起動プロセス全体で消費されたエネルギーは平均 22.5 J であった。図 3.5 は、Nexus 5 で Android OS を起動した場合の消費電力の推移である。平均で 48.2 秒の時間が掛かり、その間の消費電力は平均 1.91 W、起動プロセス全体で消費されたエネルギーは平均 105.0 J であった。これらの計測結果を比較すると、Linux カーネルの起動プロセスは Android OS と比べて遥かに高速であり、消費エネルギーも 1/4 弱に抑えられる。この特徴は、電池寿命の延長のために電源オフ状態での待機を必要とする間欠的センシングにおいて非常に重要である。

さらに、我々は Nexus 5 の充電時の挙動についても調査を行った。この実験では、Nexus 5 の内蔵バッテリーを取り外し、バッテリー接続端子および USB 端子から給電を行った。図 3.6 によると、ロジックボード自体の消費電力が安定するまでの 20 秒間に平均 22.5 J のエネルギーが消費された後、ロジックボードの消費電力は 0.12W で一定となる。このことから、充電時の給電量が 0.12 W を下回る場合には、ロジックボードの消費電力が給電量を上回り、むしろ内蔵バッテリーを放電させてしまうことが分かる。従って、給電量が 0.12 W を下回るような場合には、外部のバッテリーやキャパシタに電力を貯めてから、まとめた分量を給電すべきである。

### 3.5 電池寿命の推定

次に、2.3.3 節で紹介した WakeDroid のジオフェンス機能を用いた貨物追跡アプリケーションを想定し、通常の Android OS を利用する場合と Linux カーネルのみを利用する場合の電池寿命の比較を行う。具体的な利用シナリオとしては、(1) 対象のスマートフォンに取り付けた WakeDroid の USB ドングルにマイクロ波を照射し電源投入する、(2) Android OS または Linux カーネルが起動する、(3) 起動後、5 秒間待機してマイクロ波にオンオフ変調

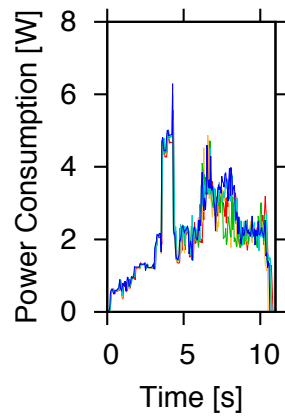


図 3.4: Nexus 5 上で Linux カーネルのみを起動した場合の消費電力の推移. 平均電力は 2.11 W, 合計で 22.5 J のエネルギーを消費した.

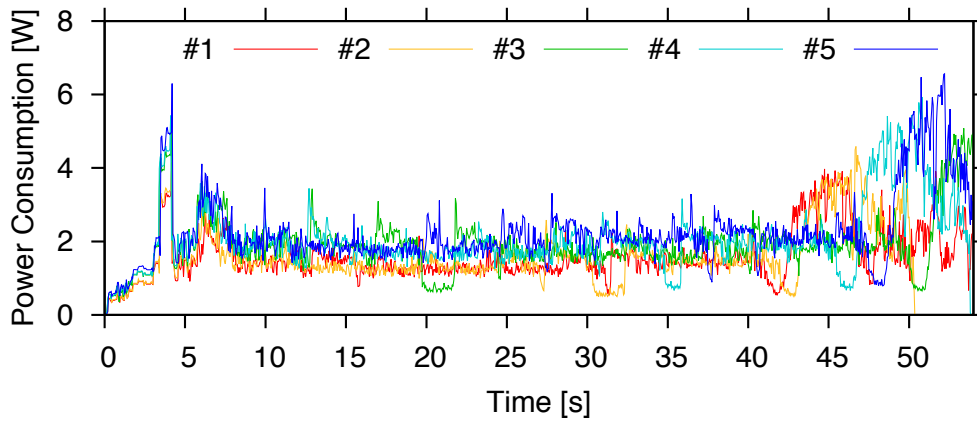


図 3.5: Nexus 5 上で Android OS を起動した場合の消費電力の推移. 平均電力は 1.91 W, 合計で 105.0 J のエネルギーを消費した.

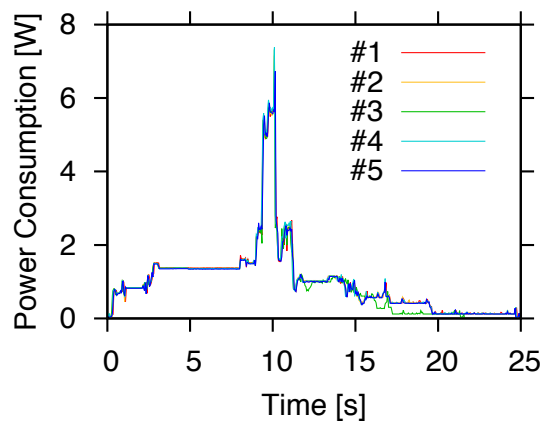


図 3.6: Nexus 5 を充電する際のロジックボードの消費電力. 消費電力が 0.12 W で安定するまでに 20 秒間に, 22.5 J の電力的オーバーヘッドが存在する.

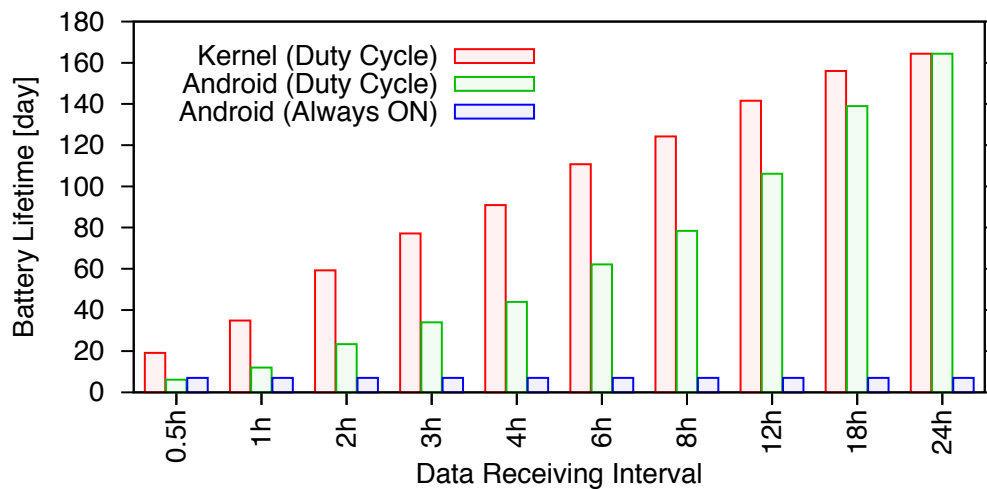


図 3.7: マイクロ波照射によるスマートフォンの起動間隔と移動履歴の送信間隔を変化させた時の電池寿命の比較

で埋め込まれた ID を読み取る, (4) スマートフォンをシャットダウンする, (5) 24 時間に一度 Android OS を起動して移動履歴を送信する, という動作を想定する. ここで, スマートフォンの電池容量を Nexus 5 と同じ  $3.8\text{ V} \cdot 2,300\text{ mAh}$  と仮定し, 電池が 1 年間で完全に自然放電すると想定して電源オフ状態での待機電力は  $2300\text{ mAh} \times 3.8\text{ V} \div (24\text{ h} \times 365) \simeq 1\text{ mW}$  とした. また, マイクロ波給電による内蔵バッテリーの充電は考慮しないこととした.

図 3.7 は, マイクロ波照射によるスマートフォンの起動間隔と変化した時の電池寿命の比較である. 赤色が Linux カーネルを用いて間欠駆動を行った場合, 緑色が Android OS を用いて間欠駆動を行った場合, 青色が Android OS を常時起動した場合の電池寿命である. スマートフォンの起動間隔が 1 時間の場合, Linux カーネルを用いた場合には Android OS を常時起動した場合の 5.0 倍にあたる 35 日程度の電池寿命を期待できる. これは, Android OS を用いて間欠駆動を行った場合の 2.9 倍にあたる. また, 起動間隔が 12 時間の場合は, Android OS を常時起動した場合の 20.2 倍にあたる 142 日程度の電池寿命を期待できる.

### 3.6 検討事項

これまで見てきたように, 我々のフレームワークはスマートフォンを間欠駆動させる際の電力的なオーバーヘッドを大幅に短縮するものであり, スマートフォンセンシングの適用領域の拡大に寄与するものである. しかしながら, さらなる改良のために検討しておくべき事項がいくつか存在する. 最大の懸念事項は, 極めて多様なハードウェアを持つ Android スマートフォンの機種依存の問題である.

### 3.6.1 スマートフォンの機種依存

我々は、本フレームワークの有用性を実証するため Nexus 5 と Android 5.1 の組み合わせで実装作業を進めてきたが、本フレームワークは、原理原則として root 権限の取得とブートローダアンロックが施されたあらゆる Android 端末に対して適用可能である。本手法を様々な機種に対応させる最も簡便な手法は、CyanogenMod [38] や MIUI [39] といった Android OS の派生ディストリビューション（カスタム ROM）を活用することである。これらのカスタム ROM は、既にそれぞれ 1,000 万人以上のユーザを獲得しており、様々なデバイス上で実際に動作している。

このような Android OS の高い移植性を支えているのはハードウェア抽象化レイヤ（HAL）である。HAL は機種によるハードウェアの違いを吸収するための機構であり、アプリケーション・フレームワークとハードウェアドライバの間を取り持つ役割を果たす。これにより、実機から抽出したハードウェアドライバを、カスタム ROM のソースコード内の所定の位置に配置してビルドするだけで、カスタム ROM を様々な機種に対応させることができる。

別の懸念事項は、OS バージョンの違いによる影響である。とりわけ、Android 5.0 で導入された SELinux は、システム領域における近年では最大の変更点である。SELinux は通常ユーザだけでなく管理者ユーザや root ユーザの特権までも制限するため、安易には root 権限を利用できなくなる可能性がある。この点について調査したところ、Android OS ソースコードに含まれる SELinux の定義ファイルを編集することで、回避できることが明らかとなった。また実際に、我々のプロトタイプは Android 5.1 ベースで実装されている。

### 3.6.2 フレームワークの導入コスト

本節では、既存の派生ディストリビューション（カスタム ROM）を活用する前提で、本フレームワークの導入コストについて論じる。ここで必要となる工程は、(1) ビルド環境の構築、(2) カスタム ROM のソースコード取得、(3) ハードウェアドライバの抽出、(4) WakeDroid 機能の実装、(5) ソースコードのビルド、(6) 端末へのインストール、の 6 つである。

まず工程 (1) では、Android Open Source Project (AOSP) の指示に従い、Linux/UNIX 環境に必要なパッケージをインストールする。ハードウェア要件は、250GB 以上の HDD 容量、16GB 以上の RAM 容量、高速なマルチコア CPU、と決して低くはないが、Amazon AWS 等のクラウドサービスを利用すれば容易に調達が可能である。ここで注意すべきは、ビルド対象の Android OS のバージョンによって用意すべき Java Development Kit (JDK) のバージョンが異なる点である。Android 7.0 以降では OpenJDK 8 を用いる。なお、この工程の所要時間は 30 分程度である。

次に工程 (2) では、既存のカスタム ROM のソースコードを GitHub 等のリポジトリから取得する。回線の実効速度やリポジトリの混雑状況によってはダウンロードに時間が掛かるため、その間に工程 (3) を進める。工程 (3) では、AOSP で提供されるツールを用いて、USB ケーブルで接続したスマートフォン実機からハードウェアドライバ等のプロプライエタリなバイナリファイルを抽出する。これらのバイナリファイルは、ダウンロード

が完了したソースコードの所定の位置に配置する。さらに工程 (4) では、我々の用意したソースコードの差分を、カスタム ROM のソースコードにマージする。

いよいよ工程 (5) では、ソースコードのビルドを行って OS イメージを生成する。この工程の所要時間はビルド環境のスペックに大きく依存するが、IST クラウド上の CPU 16 コア、RAM 容量 10GB のマシンを使用した場合で数時間を要した。ただし、ビルドした OS イメージは同一機種スマートフォンに対して使い回すことが可能であるため、各機種と OS バージョンの組み合わせについてビルド作業は 1 回限りで済む。

普及台数の多い機種については、我々が事前にビルドした OS イメージを用意しておくことで、ユーザは工程 (1) から工程 (5) までを省略できる。すなわち、ユーザの実質的な負担は工程 (6) だけということになる。

最後に工程 (6) では、Fastboot モードで端末を起動してブートローダアンロックした上で、ClockworkMod や TWRP といったサードパーティ製の Recovery モードを使用し、事前に SD カード内に配置しておいた OS イメージを書き込む。従前、これらの操作は Android SDK に付属する制御用コマンドで行う必要があったが、近年では GUI で操作可能なツールが出回っている。また、OS イメージを書き込むためのソフトウェアを公式に提供しているメーカーも存在する [47]。なお、この工程は 10~15 分程度で完了する。

### 3.6.3 さらに省電力化

3.3.2 節で述べたように、これまで我々は時間の許す限りディスプレイモジュールの消費電力対策を行ってきたが、まだ改善の余地が残っている。一案として、ディスプレイドライバの読み込みを停止するか、画面解像度を下げることで GPU の負荷を低減することが挙げられる。先行研究によれば、GPU を含むグラフィックモジュールの消費電力は無視できない水準にあり、これを削減することは有用である [16]。

## 3.7 関連研究

スマートフォンを用いたセンシング技術については数多の先行研究が存在するが、いずれも Android OS や iOS といった通常のスマートフォン向け OS で対象のスマートフォンが動作していることを前提としており、この前提を覆して検討を行った事例は我々の知る限り存在しない。すなわち本研究は、従来スマートフォンセンシングという枠組みで捉えられて来なかった全く新しい領域に踏み込んだものである。一方で、Android OS は、オープンソースゆえに有志による派生ディストリビューション (カスタム ROM) の開発が盛んであり、研究用途でもセンシング活動に必要な機能を OS レベルで組み込む提案がなされてきた [48, 49, 50]。本節では、これらの Android OS をベースとして機能拡張を行った研究事例を紹介する。

ユーザの転倒検知など、非常に高いリアルタイム性が求められる用途に Android OS を対応させるための研究としては、Yan らが、Linux カーネル、標準ライブラリ、Java 仮想マシン、Android API の一部を換装しリアルタイム性を持たせた RTDroid を提案している [51]。彼らは、実際の Android 端末を用いて性能評価を行い、優先度の異なる多数のスレッドが共存する状況や、CPU やメモリに高い負荷が掛かる状況でも、タスクの実行時

間を予測可能であることを示した。また、これに先行する研究事例として Kalkov らの提案があるが、Yan らの提案と比べると既存のアプリケーション資産を流用できない、アプリ開発者にメモリ管理などの高い技能を要求するなど、いくぶん実用性の低いものであった [52]。

また、複数のスマートフォンの中でセンサデバイスを共有する取り組みとしては、Rio[53] や mocksensor[54] が挙げられる。これらの研究では、Android SDK の Sensor Manager API を改変し、センサデバイスから取得したセンサ値を Wi-Fi 通信や Bluetooth 通信を介して他の端末に転送することで、当該センサデバイスがあたかもスマートフォン上に存在するかのよう振る舞うことを可能とした。無線通信に起因するセンサ値取得の遅延や消費電力の増加は見られるものの、よほどの低遅延や高信頼性を要求される用途でなければ十分に有用と言える水準である。

### 3.8 あとがき

本章では、Android スマートフォンを間欠的に駆動させる際に問題となる、Android OS の起動およびシャットダウンのプロセスによる時間的、電力的なオーバーヘッドを削減する方法として、Android OS の基礎部分である Linux カーネルのみを起動してセンシングを行う手法を提案した。我々の評価実験によると、本手法を用いることで、デバイスの起動プロセスにおける消費エネルギーを最大 78.5%削減しつつ、対象となるスマートフォンの Bluetooth インタフェースや USB インタフェース、温度センサといったセンシング活動に有用な機能を利用可能とした。これにより、Android OS を用いて間欠駆動を行う場合と比較して、スマートフォンの電池寿命を最大 4.65 倍に拡張することが可能となる。



## 第4章

---

# 音に着目した機械学習による センシング技術

## 4.1 まえがき

21世紀に入り、交通輸送の分野では動力源の変化に起因したパラダイムシフトが発生している。日本自動車工業会によると、エンジンとモータを動力源として備えるハイブリッド自動車は、国内の新車販売台数に占めるシェアが30%に達しており、既に広く普及した当たり前の存在となっている。電気自動車についても、日産自動車が売れ筋のコンパクトカーにEVモデル（ノート e-Power）を設定したことで、本格的な普及期に突入することが確実視されている。一方、鉄道分野では、従来の鉄輪モータ式に替わる超電導磁気浮上式リニアモータカーの技術が確立され、2014年に着工したリニア中央新幹線の開業が2027年に迫っている。他方で、パーソナルモビリティの分野ではセグウェイやNinebot、UNI-CABといった一人乗りの電動車両が、航空分野では急速に低価格化が進んだマルチコプタが注目を集めている。

このような新しいモビリティ装置が登場した背景には、リチウムイオン二次電池の高エネルギー密度化やジャイロセンサの小型化といった技術革新がある。特に、リチウムイオン電池の重量エネルギー密度は、従来のニッケル・カドミウム電池と比べて3~4倍程度、ニッケル・水素電池と比べて2倍程度と高く、機器の小型化・軽量化に大きく寄与する。また、公称電圧が3.7Vと高く出力電圧が安定しているため扱いやすいのも利点である。エネルギー密度の単純比較ではガソリンの1/100程度と遠く及ばないものの、エネルギー変換効率では自動車用ガソリンエンジンの数倍は優れており、乗用車としての航続距離もカタログ値で400kmと実用的な水準になりつつある。また、マルチコプタについても、40分間の連続飛行が可能な製品が登場している。

このような新しいモビリティ装置は、交通渋滞や道路騒音、運転手不足といった社会問題の解決策となり得る一方で、新たな社会問題の原因ともなる。例えば、電気自動車は走行時の静粛性に優れており騒音問題の緩和に役立つが、静音性ゆえに周囲の歩行者が車両の接近に気づきにくく交通事故が多いことが知られている。また、マルチコプタは小型の機体でヘリコプタ由来の高い操縦性を持ち低空域での活用が期待されるが、人間の活動領域に近い空域を飛行することから衝突の危険性がある。

そこで本章では、電気自動車やマルチコプタが発する動作音に着目し、機械学習を用いて電気自動車の接近やマルチコプタの過積載をロバストに検知する手法を提案する。ここで提案する手法は、我々が日常生活で持ち歩くスマートフォン上で実行可能なほど軽量であり、インフラとして特別な設備を必要としないため、ユーザが自衛のために今すぐ利用できる点が特徴的である。

本章の構成は以下の通りである。まず、4.2節では、電気自動車の接近検知による交通事故予防のアプリケーションを紹介する。次に、4.3節では、マルチコプタの積載重量を飛行音から推定する手法を提案する。さらに、4.4節で、スマートフォンに搭載されたマイクの周波数特性に関する調査結果を報告した後、最後に本章の内容をまとめる。

- ① EV/HV quietly approach to pedestrian
- ② Detect a signature sound from EV/HV using smartphone/smart watch
- ③ Show warning messages on wearables

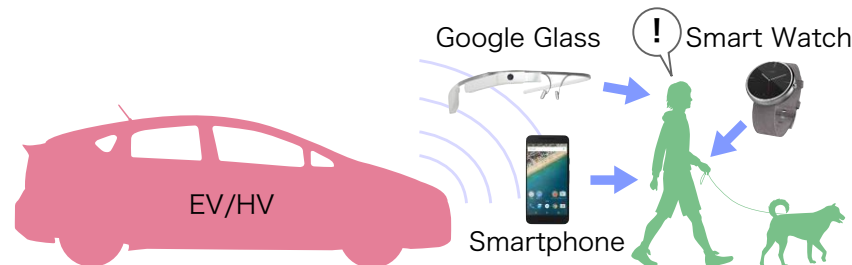


図 4.1: 提案するシステムのコンセプト. 歩行者に接近する電気自動車やハイブリッド車を, 機械学習を用いてスマートフォン上のアプリケーションで検知する.

## 4.2 電気自動車の接近検知による交通事故の予防

### 4.2.1 背景

ハイブリッド車や電気自動車は, もはや我々の社会において特別な存在ではない. 日本では, 2015 年の新車販売台数の上位 30 車種のうち, 25 車種で EV/PHV/HV モデルが提供されており (表 4.1) [55], 著しい燃費性能の向上や販売価格の低下に伴ってますます人気を獲得している. ハイブリッド車を含めて電気自動車が人気を集める理由は, 優れた燃費性能だけでなく高い静粛性にもある. 特に 15 km/h 以下の低速域において, 電気自動車の騒音レベルはガソリン車やディーゼル車と比較して著しく優れている [56] (図 4.2). 優れた静粛性は, 自動車の騒音問題に対する直接的な解決策となる一方で, 皮肉にも新たな社会問題を引き起こした. すなわち, 歩行者や自転車との交通事故の増加である. 米国の国家道路交通安全局 (NHTSA) の統計によると, ハイブリッド車はエンジン車と比較して, 後退時や駐車時など超低速域での対歩行者事故が 2 倍多く, 対自転車事故の増加も報告されている [57]. また, 交通事故の増加だけでなく, 犯罪への悪用も問題となっている. 具体的には, ハイブリッド車の高い静粛性を利用して被害者の後方から忍び寄りバッグを奪い去る, という事件が報告されている [58]. いずれの問題も, 歩行者が車両の接近に気づきにくいことが根本的な原因であり, 早急な解決策が求められる.

この問題を緩和するため, ほとんどの電気自動車やハイブリッド車には「車両接近警報装置」が備え付けられている. この装置は, 車外に設置されたスピーカーから警告音を発することで, 歩行者に車両の存在を知らせるものである. しかしながら, この装置は 3 つの重大な問題を抱えている. 1 点目として, ほとんどの車種の警告音はモータ音を模した高周波音であり, 高齢者には聞き取りづらい. 2 点目として, 警告音自体が騒音の原因となり, 電気自動車の利点である高い静粛性を台無しにする. 3 点目として, 警報装置は運転者の判断で無効にできるため信頼性が低い.

本研究では, 電気自動車やハイブリッド車に搭載されたモータユニットが発する高周波音に着目し, 機械学習を用いて車両の接近を検知する手法を提案する. この高周波音には,

表 4.1: 2016 年の新車販売台数の上位 30 車種のうち, 25 車種で EV/PHV/HV モデルが提供されている.

順位	車種	メーカー	販売台数	EV/HV モデルの有無
1	プリウス	トヨタ	248,258	有り
2	アクア	トヨタ	168,208	有り
3	シエンタ	トヨタ	125,832	有り
4	フィット	ホンダ	105,662	有り
5	ノート	日産	102,402	有り
6	ヴォクシー	トヨタ	91,868	有り
7	カローラ	トヨタ	84,770	有り
8	ヴェゼル	ホンダ	73,889	有り
9	セレナ	日産	73,502	有り
10	ヴィッツ	トヨタ	71,909	有り
11	パッソ	トヨタ	70,381	無し
12	デミオ	マツダ	57,320	無し
13	エクストレイル	日産	56,151	有り
14	ノア	トヨタ	54,826	有り
15	ステップワゴン	ホンダ	52,472	有り
16	フリード	ホンダ	52,202	有り
17	ヴェルファイア	トヨタ	48,982	有り
18	ソリオ	スズキ	48,814	有り
19	エスクァイア	トヨタ	44,881	有り
20	シャトル	ホンダ	42,514	有り
21	インプレッサ	スバル	42,423	有り
22	ハリヤー	トヨタ	41,403	有り
23	クラウン	トヨタ	39,813	有り
24	アルファード	トヨタ	37,069	有り
25	オデッセイ	ホンダ	30,858	有り
26	フォレスター	スバル	28,544	無し
27	アクセラ	マツダ	26,342	有り
28	イグニス	スズキ	24,261	有り
29	レヴォーグ	スバル	24,184	無し
30	ランドクルーザー W	トヨタ	23,321	無し

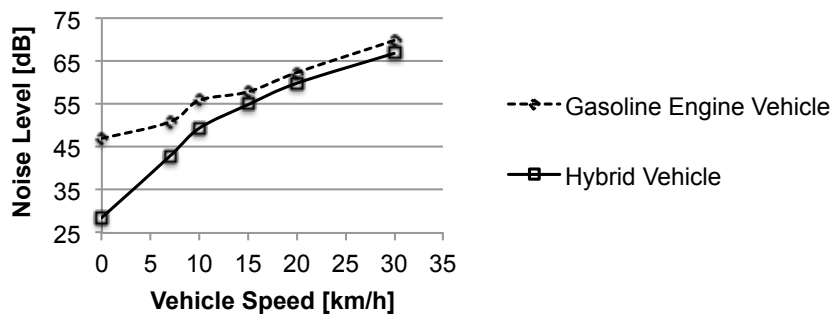


図 4.2: ガソリン車とハイブリッド車の騒音レベルの比較. ハイブリッド車は 15km/h 以下の低速域で特に静粛性に優れている [56].

駆動系の内部構造に起因して生じるスイッチング雑音が含まれており, これを分析することで走行速度, 相対速度といった有用な情報を得られる. また, ドップラー効果による見かけの周波数の変化により, 近づく車両と遠ざかる車両を明確に区別できる. 我々の評価実験では, これらのスイッチング雑音は都市部の大通りにおいても検知可能なほど明瞭である. 人間の聴覚が高周波音に対して比較的鈍感であるのに対して, スマートフォンのような電子機器はこのような高周波音を容易に検知できる. 我々の提案する車両検知ロジックはスマートフォン上でリアルタイムに実行できるほどシンプルかつ軽量である. 本論文では, 修士課程での検討からさらに踏み込んで, 実験車両を国産車 2 車種から輸入車を含む 4 車種に拡充し, 複数車両が同時に存在するシナリオでの性能評価を行う. さらに, 成長期に差し掛かりつつある Android Wear スマートウォッチへのプロトタイプ実装についても紹介する.

本章の構成は以下の通りである. まず 4.2.2 節では, 4 車種の電気自動車を用いて録音した走行音の特徴を紹介し, ここからどのような情報を得られるかを述べる. 次に 4.2.3 節では, 我々の提案する車両検知システムの概要と機械学習を用いた学習フェーズにおける録音データの取り扱いや特徴量について述べる. 4.2.4 節と 4.2.5 節で我々の提案するシステムの性能を多面的に評価した後, 4.2.6 節で, 各種デバイスへの実装の詳細を述べる. 4.2.7 節で関連研究を紹介し, 4.2.8 節で本章の内容をまとめる.

## 4.2.2 電気自動車の走行音

### 車載モータの駆動メカニズム

電気自動車やハイブリッド車を駆動するためのモータには, タイヤホイールの内側に実装できるほどの小型さと, 限られた電力を最大限に活用できる高いエネルギー変換効率が求められる. これらの要件を満たすため, ほとんどの電気自動車やハイブリッド車は永久磁石同期電動機を採用している. この電動機の回転速度は交流電源の周波数によって制御されるが, 電気自動車に搭載されるリチウムイオン電池やニッケル水素電池は直流電源であるため, パルス幅変調 (PWM) で生成した擬似的な正弦波を用いる. この PWM 制御でも高い電力効率が求められるため, 絶縁ゲートバイポーラトランジスタ (IGBT) などの半導体スイッチが用いられる. この時, モータの回転速度に依らず, スwitching 周波



図 4.3: 計測環境の風景

数は一定であることが多い。この高周波電流がモータ内のコイルを流れて磁界の変動を発生させ、磁界の変動がモータ周辺の金属部品を振動させることで、電気自動車に特有の高周波音が発生する。他に、タイヤと路面との摩擦音や車体の風切音も存在するが、低速域においてはモータからの高周波音が支配的である [59]。

### 走行音の録音

電気自動車やハイブリッド車の走行音を分析するため、我々は4車種の車両と図 4.3 に示す4つの測定環境で走行音の録音実験を行った。車両については、(1) 日産リーフ (ZE0), (2) トヨタプリウス PHV (ZVW35), (3) BMW i3, (4) 三菱アイミーブ (HA3W), を採用し、実験中は前述の「車両接近警報装置」を無効とした。実験場所については、理想的な計測環境として、(a) 閑静な墓地, (b) 河川公園の駐車場, 交通事故の発生しやすい環境として、(c) 閑静な住宅街, (d) 交通量の多い大通り, を採用した。先行研究によると、電気自動車が 30 km/h 以上で走行する時は、タイヤと路面の摩擦音や車体の風切音が支配的となり、ガソリン車と同程度の騒音レベルとなることから、5 km/h, 10 km/h, 20 km/h, 30 km/h の4通りで計測を行った [59]。ただし、(d) 交通量の多い大通りでは、交通の妨げとならないよう 30 km/h での計測のみとした。録音機器としては iPod Touch (第4世代) と iPhone 5s を採用し、観測者が腰の前で片手保持した状態で内蔵マイクを用いて録音した。サンプリングレートについては広く用いられる 44.1 kHz と 48 kHz の2種類とし、録音データは非圧縮の 16 bit リニア PCM 形式 (WAVE 形式) で保存した。車両の通過位置は観測者の前方約 1.5 m の地点としたが、通過する向きは奇数回目と偶数回目で異なる。(d) 交通量の多い大通りを除いて、各計測条件毎に最低5回ずつの計測を行った。

### 計測結果

図 4.4 に計測した走行音のスペクトログラムを示す。日産リーフについては、明瞭なスイッチング雑音が 10 kHz 周辺と 20 kHz 周辺に見られる。これとは対照的に、トヨタプリウスのスイッチング雑音は 5 kHz 周辺と 10 kHz 周辺に見られるが不明瞭である。また、BMW i3 では 16 kHz 付近に、三菱アイミーブでは 7 kHz 付近と 14 kHz 付近にスイッチング雑音が見られる。車種によりピーク周波数は異なるが、いずれも 10 kHz 以上の高周波音が主成分である。一方、人間の可聴域は概ね 20 Hz ~ 20 kHz とされるが、個人差が大きいことに加え、加齢により高周波音に対する感度が低下する。それゆえ、特に高齢者には電気自動車やハイブリッド車の走行音を聞き取ることが難しい。



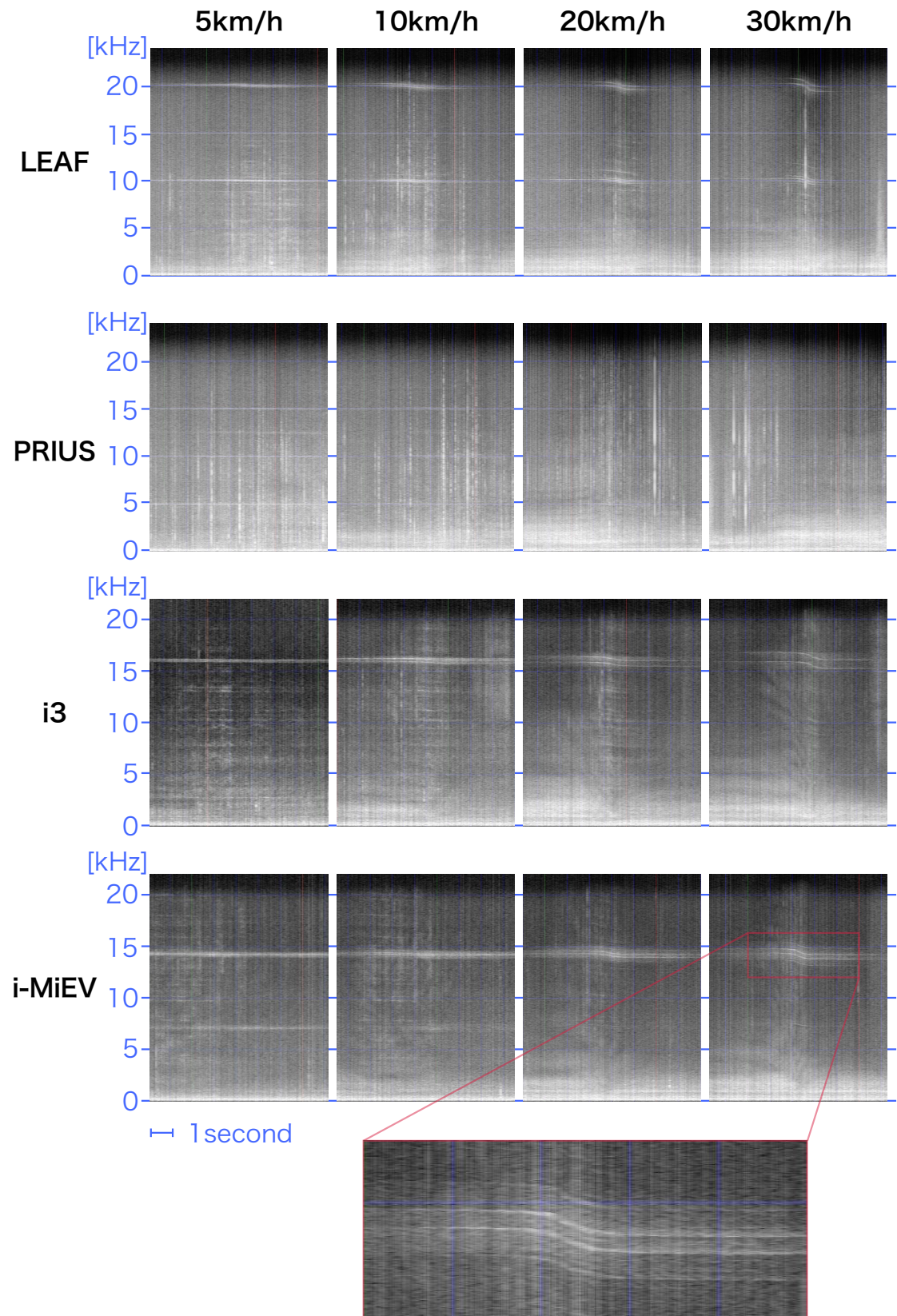


図 4.4: 電気自動車やハイブリッド車の走行音のスペクトログラム。ドップラー効果による見かけの周波数変化を示す S 字カーブが見られる。

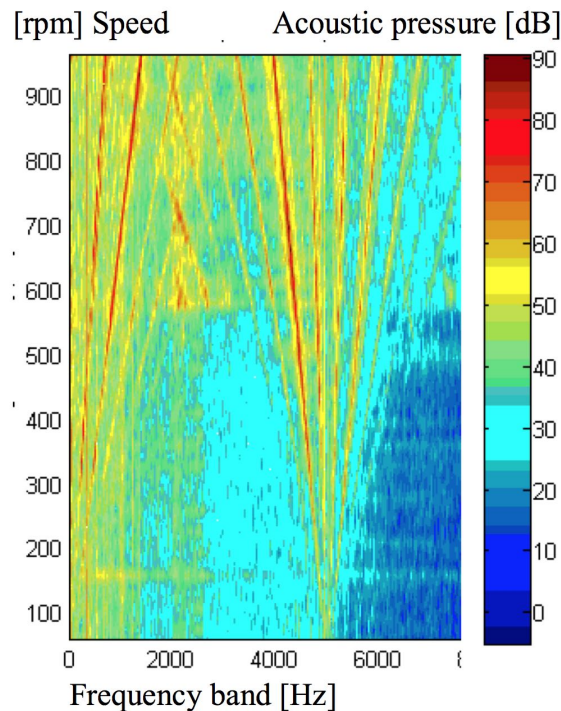


図 4.5: モータの回転速度と各周波数における音圧レベルの関係 [60]

各スペクトログラムに見られる S 字カーブは、車両の移動に伴いドップラー効果で生じた見かけの周波数変化を反映している。ドップラー効果の定義によると、見かけの周波数  $f$  と音源の周波数  $f_0$  の関係は

$$f = \frac{c^2 f_0}{c^2 - v^2} \left\{ 1 - \frac{v^2 t}{\sqrt{c^2 v^2 t^2 + l^2 (c^2 - v^2)}} \right\} \quad (4.1)$$

で与えられる。ここで  $c$  は音速であり、 $v$  は車両の移動速度、 $l$  は最接近時の観測者と車両の距離、 $t$  は時刻を表す。なお、最接近時の時刻が  $t = 0$  である。この関係を利用して、各車速における周波数変化の大きさを事前に予測することは、分類器の作成に用いる特徴量ベクトルの選定に有用である。また、図 4.5 に示す通り、永久磁石同期電動機を PWM 制御した際のスペクトルは複数の側波帯に分かれることが知られており、図 4.4 に示したスペクトログラムでも各 2,3 本のカーブが見られる [60]。これらを踏まえると、電気自動車やハイブリッド車の走行音を分析することで、車速と相対速度を割り出すことが可能である。

### 4.2.3 車両検知システム

計算資源が限られたスマートフォンやスマートウォッチ上でリアルタイムの車両検知を実現するためには、検知ロジックが十分に軽量である必要がある。一方で、環境雑音や録音機器の違いに対してロバストとするためには、特定の車種の特徴に依存してはならない。さらに、スイッチング雑音に含まれる、車速や相対速度といった情報を有効活用できるこ



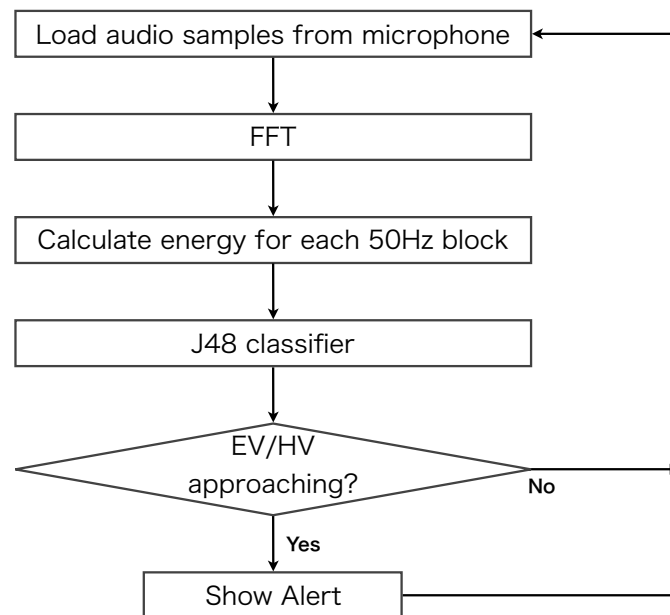


図 4.6: 提案する車両検知システムのフローチャート. 分類器は事前に学習データを用いて構築しておく.

とが好ましい. これらの要件を満たすため, 我々は機械学習によるアプローチを採用した. 提案する車両検知システムの概要を図 4.6 にフローチャートで示す.

### 特徴量ベクトル

機械学習の最初のステップとして, 我々は 4.2.2 節で採取した録音データから特徴量ベクトルの抽出を行った. まず, 移動幅 100 ms のスライディングウィンドウを設定して音声データを一定量ずつ切り取り, 高速フーリエ変換 (FFT) を用いて周波数領域のエネルギー分布を算出する. 次に, 50 Hz 刻みでの平均エネルギーを計算し, 全周波数帯での平均エネルギーで正規化する. なお, 刻み幅を 50 Hz としたのは, 以下の 2 つの条件を考慮したためである. 1 つ目の条件は, 明瞭なスイッチング雑音が見られる 14 kHz 以上の周波数帯において, 5 km/h の差をドップラーシフトにより区別できることである. 14 kHz の音源が 5 km/h で移動する場合のドップラーシフトは約 57 Hz であることから, 刻み幅は 57 Hz 以下に制約される. 2 つ目の条件は, スwitching 雑音のピーク周波数が刻みの境界上に位置し, わずかなドップラーシフトでも, 上下いずれかの刻みに分類されることである. これにより, 近づいて来る車両と遠ざかって行く車両を明確に区別できる. なお, 44.1 kHz サンプリングの録音データと 48 kHz サンプリングの録音データを等しく扱うため, ここでは 22 kHz 以下の周波数帯域のみを抽出の対象とした. それゆえ, 特徴量ベクトルは 440 次元となる. 以降, この 440 次元の特徴量ベクトル 1 つを学習データの 1 インスタンスとして扱う.

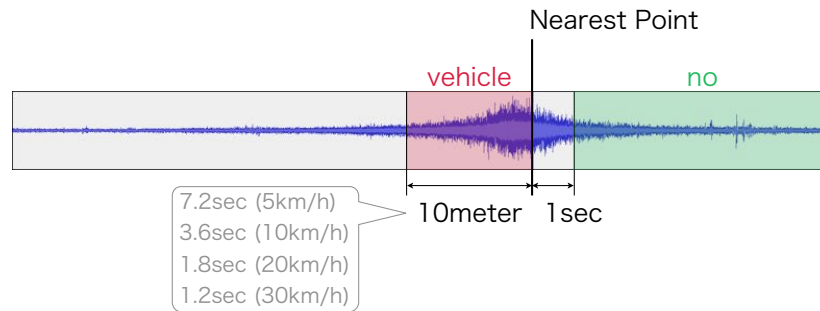


図 4.7: 録音データから学習用データセットを切り出す手順. 走行音の明瞭な最接近の直近 10 m 分を「車両あり」の学習データとし, 周波数変化の少ない最接近 1 秒後以降のデータを「車両なし」の学習データとした.

### ラベリング

次に, 録音データから抽出された特徴量ベクトルに対して, 車種と車速を示すラベルを手作業で付与した. 車両なしの場合も含めて, “no”, “leaf-5”, “leaf-10”, “leaf-20”, “leaf-30”, “prius-5”, “prius-10”, “prius-20”, “prius-30”, “i3-5”, “i3-10”, “i3-20”, “i3-30”, “imiev-5”, “imiev-10”, “imiev-20”, “imiev-30” の全 17 種類である. 車両が近づいて来る部分については, 最接近の直近 10 m 分 (5 km/h の場合 7.2 秒分) のみを「車両あり」の学習データとして採用し, それ以前の区間は走行音が不明瞭であることから除外した. また, 遠ざかる車両が歩行者を傷つける可能性は無いため, 車両が遠ざかる部分の録音データについては, 「車両なし」を示す “no” のラベルを付与した (図 4.7) ただし, ドップラー効果により見かけの周波数が大きく変化する最接近後 1 秒間は除外した. このラベルは, 次節に示す決定木の構築において正解データとして利用される.

### 教師あり学習

教師あり学習を行うため, 我々は WEKA というデータマイニングツールを利用した. 分類器には, C4.5 決定木の Java 実装版である J48 決定木を採用した. 学習データには, ほぼ理想的な環境である, (a) 閑静な墓地, (b) 河川公園の駐車場, での録音データのみを使用した. この学習データには, 全 4 車種 (日産リーフ, トヨタプリウス, BMW i3, 三菱アイミーブ) を, 全 4 車速 (5 km/h, 10 km/h, 20 km/h, 30 km/h) の録音データが含まれる. また, 「車両あり」の学習データ数と「車両なし」の学習データ数を均衡させるため, 後者については間引きを行った. 4.2.4 節では, スライディングウィンドウ幅を 32,768 サンプル, 間引き率を 1/5 に固定して性能評価を行うが, 4.2.5 節で, これらのパラメータの妥当性について検討する.

表 4.2: 理想的な環境下で日産リーフを異なる速度で走行させた場合の検知精度。接近車両の有無の判定精度は 92.8% であり、誤検知率と見落とし率（赤字）はそれぞれ 12.5%、5.2% であった。

Nissan LEAF Parking Lot		Recognition Result				
		no	LEAF 5	LEAF 10	LEAF 20	LEAF 30
Ground Truth	no	87.5%	4.7%	3.7%	2.2%	1.9%
	LEAF 5	2.8%	94.7%	2.3%	0.2%	0.0%
	LEAF 10	6.5%	1.9%	91.2%	0.5%	0.0%
	LEAF 20	6.5%	0.0%	5.6%	85.2%	2.8%
	LEAF 30	13.9%	0.0%	1.4%	2.8%	81.9%

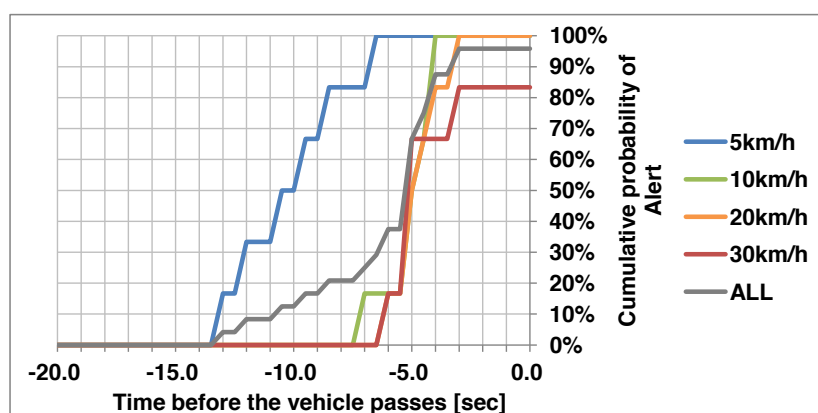


図 4.8: 車両が最接近するまでに 1 度でも車両を正しく検知できた確率。最接近の 3 秒前までに 95% の車両を検知できている。

#### 4.2.4 性能評価

##### 理想環境シナリオ

まず、理想的な環境下における単一車種に対する検知性能の評価を行った。具体的には、日産リーフを用いて (b) 河川公園の駐車場で録音したデータのみを用いて 10 分割交差検証により判定精度を評価した。車速は 5 km/h, 10 km/h, 20 km/h, 30 km/h の 4 種類であり、学習データのインスタンス数は 1,149 個であった。その内訳は、“no” が 321 個，“leaf-5” が 432 個，“leaf-10” が 216 個，“leaf-20” が 108 個，“leaf-30” が 72 個であり、交差検証における標本群分割はランダムとした。

判定精度は表 4.2 に示す通りであり、接近車両の有無の判定精度は 92.8% であった。接近車両が存在しないデータ群に対して「車両あり」と判定した割合を「誤検知率」、接近車両が存在するデータ群に対して「車両なし」と判定した割合を「見落とし率」と定義すると、誤検知率は 12.5%、見落とし率は 5.2% であった。わずかながら隣接する速度区分の誤判定も見受けられるが、交通事故の防止という目的に照らせば大きな問題ではない。

図 4.8 は、車両が最接近するまでに車両を正しく検知できた確率を示したものである。ここでは、雑音等による誤検知の影響を除いた正当な評価を行うため、100 ms 刻みで実

表 4.3: 3つの計測環境で日産リーフを走行させた場合の検知精度。接近車両の有無の判定精度は93.2%であり、誤検知率と見落とし率はそれぞれ11.9%、4.9%であった。

Nissan LEAF 3 Environment		Recognition Result				
		no	LEAF 5	LEAF 10	LEAF 20	LEAF 30
Ground Truth	no	88.1%	4.3%	3.0%	2.1%	2.5%
	LEAF 5	3.8%	94.4%	1.8%	0.0%	0.0%
	LEAF 10	5.6%	4.3%	87.6%	1.5%	1.0%
	LEAF 20	6.1%	0.5%	3.0%	86.4%	4.0%
	LEAF 30	7.1%	0.6%	1.2%	3.0%	88.1%

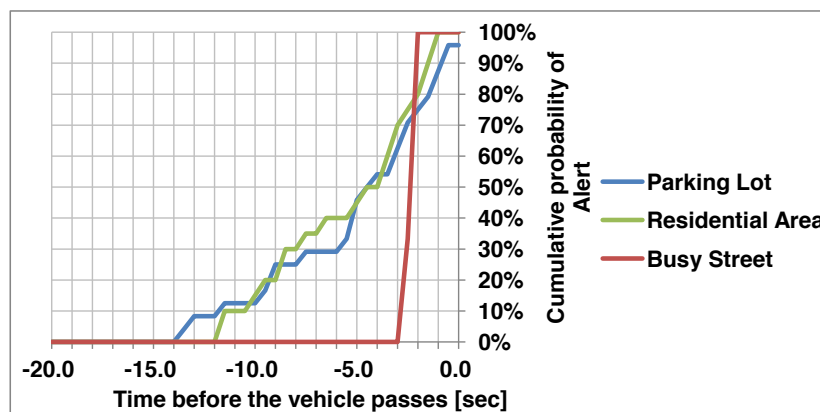


図 4.9: 車両が最接近するまでに1度でも車両を正しく検知できた確率。大通りでも2秒前までに接近車両を検知できている。

行した過去10回の判定で5回以上「車両あり」と判定した場合のみを「検知」とした。図4.8によると、最接近の平均6.4秒前には車両を検知できており、最も早い例では最接近の13.1秒前に車両を検知できた。また、車速が上がるほど検知が遅くなる傾向が見られるものの、全体では3秒前までに95%以上の車両を検知できた。

### 雑音環境シナリオ

次に、環境雑音が検知性能に及ぼす影響の評価を行った。具体的には、日産リーフを用いて、(b) 河川公園の駐車場、(c) 閑静な住宅街、(d) 交通量の多い大通り、の3ヶ所で録音したデータを用いて10分割交差検証により判定精度を評価した。学習データのインスタンス数は2,115個であった。その内訳は、“no”が561個、“leaf-5”が792個、“leaf-10”が396個、“leaf-20”が198個、“leaf-30”が168個であり、交差検証における標本群分割はランダムとした。

判定精度は表4.3に示す通りであり、接近車両の有無の判定精度は93.2%であった。誤検知率は11.9%、見落とし率は4.9%であり、理想環境下とほぼ同じ判定精度を達成している。図4.9は、車両が最接近するまでに車両を正しく検知できた確率を、計測環境別に集計したものである。最接近の平均5.4秒前には車両を検知でき、最も早い例では最接近

表 4.4: 理想的な環境下で4車種を走行させた場合の検知精度。接近車両の有無の判定精度は92.5%であり、誤検知率と見落とし率はそれぞれ7.6%, 6.7%であった。

4 Vehicle Models Semi-ideal Env.		Recognition Result				
		no	LEAF	PRIUS	BMW i3	i-MiEV
Ground Truth	no	92.4%	1.2%	1.5%	2.0%	2.9%
	LEAF	5.8%	92.3%	1.7%	0.1%	0.1%
	PRIUS	7.7%	1.0%	91.2%	0.0%	0.1%
	BMW i3	4.8%	0.3%	0.1%	94.8%	0.0%
	i-MiEV	8.4%	0.0%	0.0%	0.0%	91.6%

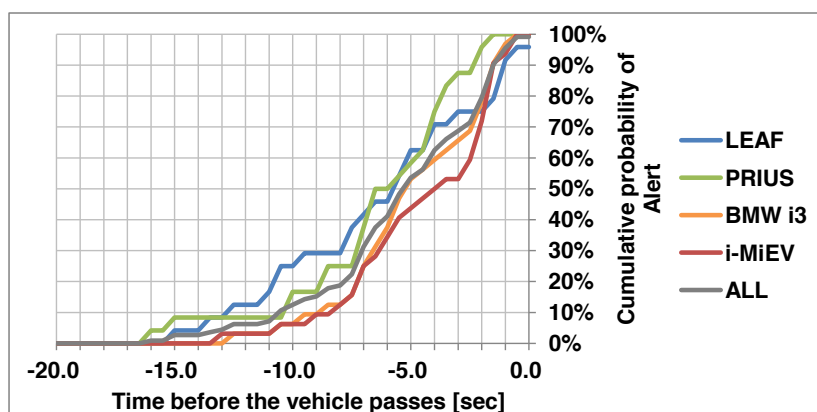


図 4.10: 車両が最接近するまでに1度でも車両を正しく検知できた確率。複数車種の学習データを混在させても、問題なく接近車両を検知できている。

の13.6秒前に車両を検知できた。理想環境下と比較すると検知が若干遅れているが、交通量の多い大通りにおける30 km/hの実験データでさえ、最接近の2秒以上前に車両を検知できており、交通事故の防止に十分な性能である。

#### 車種による違い

さらに、車種の違いが検知性能に及ぼす影響の評価を行った。具体的には、日産リーフ、トヨタプリウス、BMW i3、三菱アイミーブの4車種を用いて、ほぼ理想的な環境である、(a) 閑静な墓地、(b) 河川公園の駐車場、で録音したデータを用いて10分割交差検証により判定精度を評価した。学習データのインスタンス数は7,066個であった。その内訳は、“no”が3,202個、“leaf”が828個、“prius”が828個、“i3”が1,104個、“imiev”が1,104個であり、交差検証における標本群分割はランダムとした。

判定精度は表4.4に示す通りであり、接近車両の有無の判定精度は92.5%であった。誤検知率は7.6%、見落とし率は6.7%であり、単一車種の場合とほぼ同じ判定精度を実現している。また、90%以上の精度で車種を判別できた。図4.10は、車両が最接近するまでに車両を正しく検知できた確率を、車種別に集計したものである。最接近の平均5.6秒前には車両を検知でき、最も早い例では最接近の16.5秒前に車両を検知できた。以上より、複

数車種が混在する場合でも、問題なく接近車両を検知できることが明らかとなった。

### 複数車両シナリオ

最後に、複数の車両が同時に走行する2つのシナリオでの評価性能を行った。1つ目は、一方の車両が3秒の間隔を置いて他方の車両を追走するシナリオであり、2つ目は、観測者の前で2台の車両がすれ違うシナリオである。これら2つのシナリオは、BMW i3と三菱アイミーブを用いて、(a) 閑静な墓地で録音を行った。ただし、性能評価に用いる分類器については、理想的な環境における4車種分の録音データで構築した決定木を使用した。この時の学習データの内訳は、“no”が3,202個、“leaf”が828個、“prius”が828個、“i3”が1,104個、“imiev”が1,104個で、合計インスタンス数は7,066個であった。

図4.11に、複数の車両が同時に接近するシナリオでの検知結果を示す。追走、すれ違いのいずれのシナリオでも、少なくとも1台の車両が接近している場合に検知が集中しており、交通事故防止アプリケーションとして有用である。複数の電気自動車またはハイブリッド車が同時に存在することは珍しくないため、このシナリオで適切な判定をできたことは実用上極めて重要な成果である。また、最下段に掲載した車両が存在しない場合の録音データに関しては、前述の性能評価結果と同じく90%以上の高い判定精度を実現している。時折見られる誤検知も連続的に発生するものではないため、連続する数フレームの判定結果で多数決を取ることで不必要な警報を防ぐことができる。なお、この録音データは、(c) 閑静な住宅街において、立ち止まった状態で録音したものである。

## 4.2.5 検討事項

### 見落とし率の低減

交通事故を予防するための車両検知アプリケーションにおいては、誤検知と比較して見落としのコストが非常に大きい。なぜなら、誤検知は歩行者が振り返って車両の有無を確認すれば済むが、見落としは交通事故の原因となり得る。それゆえ、誤検知率を犠牲にしても、見落とし率を減らすことが非常に重要である。4.2.3節で触れたように、誤検知率と見落とし率のバランスは「車両なし」の間引き率によって調整可能である。

表4.5に、間引き率を1/2から1/100まで調整した時の判定精度を示す。見落とし率と誤検知率のバランスという観点では、「車両あり」と「車両なし」のデータ数が同程度になる1/5とするのが良い。一方、見落とし率の削減を追求するなら、なるべく「車両なし」のデータを間引くことが望ましい。間引き率を1/100に設定することで、見落とし率を3.2%まで削減できるが、誤検知率は増大する。

### 検知精度・遅延・消費電力のバランス

スライディングウィンドウ幅は検知精度を向上させるための重要なパラメータである。4.2.4節では、サンプリングによる遅延時間を1秒未満としつつ可能な限り高い周波数解像度を得るため、窓幅を32,768サンプル(=0.683秒)としていたが、この節では、スライディングウィンドウ幅を短縮する可能性について検討する。



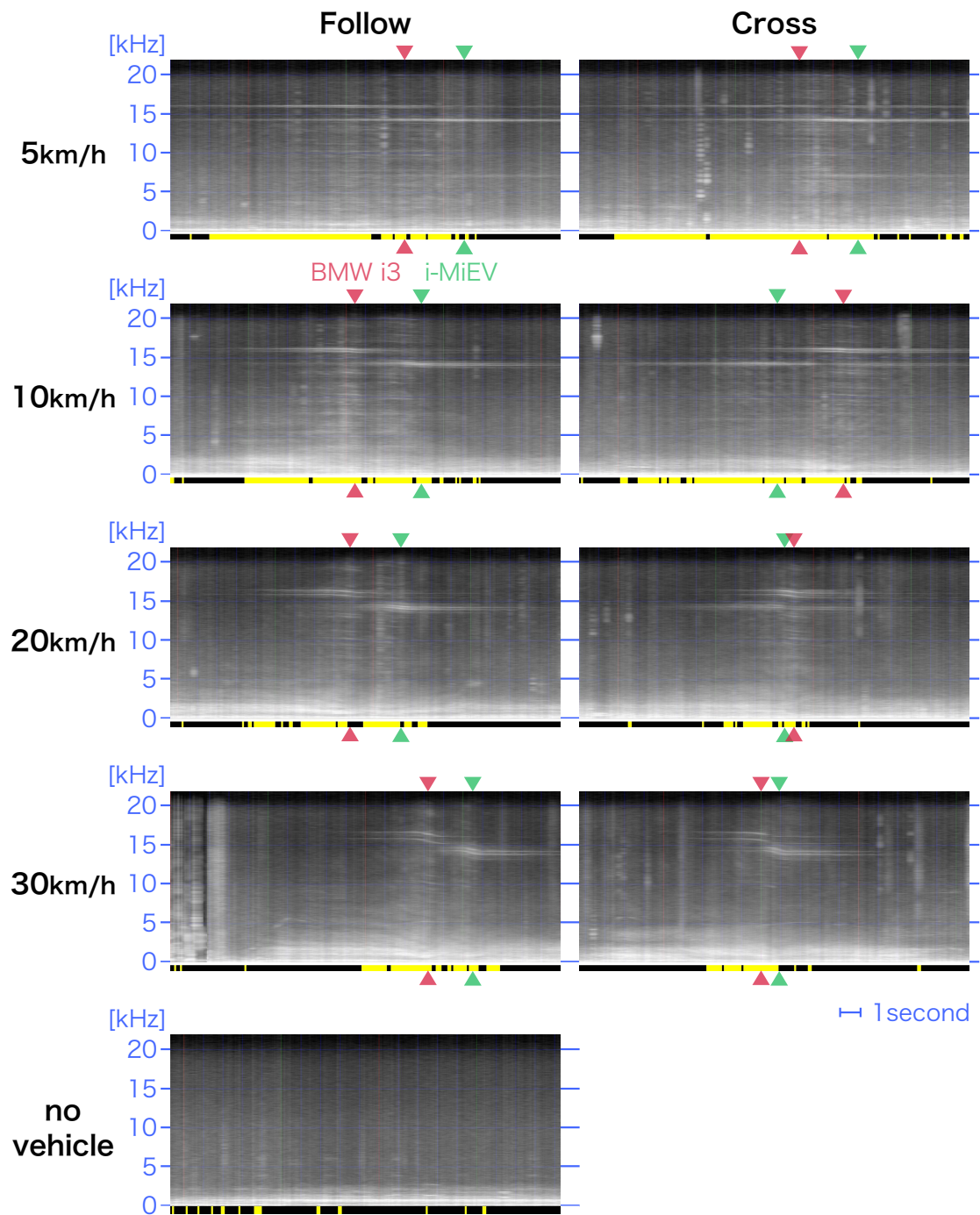


図 4.11: 複数の車両が同時に接近するシナリオでの検知結果. BMW i3 が最接近した時刻を赤印で, 三菱アイミーブが最接近した時刻を緑印で, 我々のアルゴリズムが接近車両を検知した時間帯を黄線で表示している.

表 4.5: 学習データ数が誤検知率と見落とし率に与える影響の評価. 間引き率を 1/5 に設定すると, 見落とし率と誤検知率がバランスする. 間引き率を 1/100 に設定することで, 見落とし率を 3.2%まで削減できるが, 誤検知率が犠牲となる.

Thinning Rate		Recognition Result	
= 1/2		no	EV
Ground Truth	no	96.5%	3.5%
	EV	6.9%	93.1%

no:7744 instances, EV:3864 instances

Thinning Rate		Recognition Result	
= 1/20		no	EV
Ground Truth	no	78.7%	21.3%
	EV	4.7%	95.3%

no:894 instances, EV:3864 instances

Thinning Rate		Recognition Result	
= 1/5		no	EV
Ground Truth	no	92.4%	7.6%
	EV	6.7%	93.3%

no:3202 instances, EV:3864 instances

Thinning Rate		Recognition Result	
= 1/50		no	EV
Ground Truth	no	61.8%	38.2%
	EV	4.0%	96.0%

no:422 instances, EV:3864 instances

Thinning Rate		Recognition Result	
= 1/10		no	EV
Ground Truth	no	85.7%	14.3%
	EV	5.4%	94.6%

no:1674 instances, EV:3864 instances

Thinning Rate		Recognition Result	
= 1/100		no	EV
Ground Truth	no	50.9%	49.1%
	EV	3.2%	96.8%

no:269 instances, EV:3864 instances

表 4.6: 録音データの窓幅と検知精度の関係. 窓幅を小さくすると検知精度は悪化する傾向にあるが, 窓幅を 1,024 サンプルに設定しても, 前後5フレームで多数決を取ることで, 95%弱の判定精度を得られる.

Window Size		Recognition Result	
= 32768		no	EV
Ground Truth	no	92.4%	7.6%
	EV	6.7%	93.3%

Window Size		Recognition Result	
= 4096		no	EV
Ground Truth	no	85.0%	15.0%
	EV	15.2%	84.8%

Window Size		Recognition Result	
= 16384		no	EV
Ground Truth	no	90.3%	9.7%
	EV	9.0%	91.0%

Window Size		Recognition Result	
= 2048		no	EV
Ground Truth	no	81.9%	18.1%
	EV	16.5%	83.5%

Window Size		Recognition Result	
= 8192		no	EV
Ground Truth	no	88.3%	11.7%
	EV	11.2%	88.8%

Window Size		Recognition Result	
= 1024		no	EV
Ground Truth	no	78.7%	21.3%
	EV	21.0%	79.0%



表 4.7: 車両接近判定ロジックの実行間隔を変化させた時の Nexus 5 の消費電力

状態	消費電力	「待受画面」比	「間隔なし」比
スリープ	0.028 W		
待受画面	0.462 W	(基準)	
間隔なし	1.838 W	398%	(基準)
間隔 0.4 秒	0.771 W	167%	42%
間隔 0.9 秒	0.651 W	141%	35%

表 4.6 によると、窓幅が広いほど高精度となる傾向が見られるが、一方で、音声のサンプリングによる遅延時間が長くなる。逆に、窓幅を 1,024 サンプル (= 0.021 秒) まで短縮しても 80% 弱の判定精度を維持できるため、前後 5 フレーム (= 0.105 秒) で多数決を取ることで、95% 弱の判定精度を得られる。

消費電力に関しては、窓幅が十分に狭ければ間欠動作を行う余地が出て来る。ここでは 30 km/h 以下の低速域で移動する電気自動車の検知を目的としていることから、1 秒毎に車両検知を行えば十分である。仮に、窓幅を 1,024 サンプルに設定して前後 5 フレームで多数決を取る場合、検知システムは 0.1 秒強で動作が完了するため、残りの 0.9 秒弱はスマートフォンの CPU をスリープ状態にして省電力化を図ることができる。ここでは、Nexus 5 で車両接近判定ロジックを実行した際の消費電力の測定を行った。なお、測定中は機内モードに設定した上で GPS を無効化し、画面輝度を最小とした。表 4.7 に判定ロジックの実行状態毎の消費電力を示す。車両接近判定ロジックを常時稼働させた場合には、待受状態と比較して消費電力が 4 倍近くまで増加するが、実行間隔を 0.9 秒まで拡大することで 1.4 倍に抑えることができる。これは常時稼働させた場合の約 35% の消費電力にあたる。

### スマートフォン間での連携

前節で指摘した消費電力の増加を抑制するためには、スマートフォン同士を協調させて役割分担を行い、センシング頻度を必要最小限に抑えることが望ましい。ここでは、Bluetooth Low Energy (BLE) などの省電力な近距離無線通信を用いて、検知状況を複数のスマートフォンで共有することを考える。

本手法で接近車両を検知したスマートフォンは所定の BLE ビーコンを発信し、これを受信したスマートフォンは車両検知アプリを起動して自ら車両検知を始める。一定時間が経過しても接近車両が検知されなかった場合は、車両検知アプリを終了して省電力状態に戻る。BLE ビーコンは、事前のペアリング作業や接続確立が不要であるため、スマートフォン同士のすれ違いぎまでの通信に適している。また、到達距離が 10 m 前後と短いため、近接するスマートフォンのみ指令を与えられるという利点もある。

都市部では、検知対象となる車両の絶対数が多いため、省電力状態への移行時間を適切に設定すれば、複数ユーザのスマートフォンを連鎖的に使用して効果的に車両検知を行える。一方、郊外では、もとより近接するスマートフォンが少ないため連鎖的な使用は難しい。そこで、中古スマートフォンの活用を考える。

まず、交通の要衝となる地点に中古スマートフォンを配置し、商用電源に接続して常時の

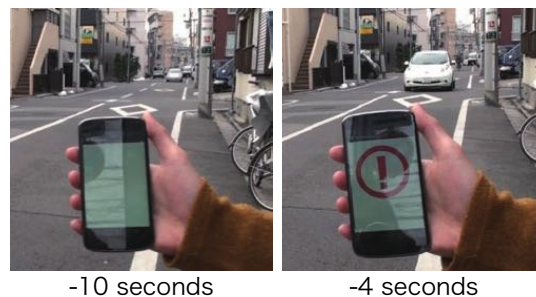


図 4.12: Android アプリケーションとして実装した我々の車両検知システムのデモンストラーションの様子。白い車（日産リーフ）が近づくと警告が表示される。

車両検知を行う。このスマートフォンが接近車両を検知すると、2章で提案した WakeDroid の技術を用いて、周辺に配置された電池駆動の中古スマートフォンを起動し、BLE ビーコンを発信させる。これを受信したユーザの手元にある現役スマートフォンは、車両検知アプリを起動して自ら車両検知を始める。これにより、システム全体での待機電力を抑えつつ、広範囲で必要十分な車両検知を実現する。

また、要衝地点に配置したスマートフォンを用いて、EV/HV の新車種の走行音を収集することも可能である。このスマートフォンは商用電源に接続されているため、電池容量の制約を受けることなく常時の動画撮影を行える。この動画を分析することで、車種、車速および通過タイミングの情報を含んだ走行音データを取得できる。

#### 未知の車種への対応

4.2.4 節で検証した通り、走行音の学習データが与えられた既知の車種については、我々の機械学習によるアプローチで非常に精度良く検知できる。しかしながら、特徴音の異なる未知の車種を検知可能とするためには、当該車種の走行音を学習データとして追加する必要がある。そこで、4.2.6 節で紹介する Android アプリケーションに録音機能を付与して、ユーザが未知の車種に遭遇した際の録音データを収集し、4.2.2 節で録音したデータを含む全ての録音データを用いて決定木を再構築することを考える。多量の録音データを用いる特徴量の抽出および分類器の構築は、スマートフォン上で行うと計算処理に時間が掛かるため、クラウド上で実行することが好ましい。また、特徴量はパラメータを変更しない限り不変であるため、1つの録音データにつき1回のみ行えば良く、抽出した特徴量ベクトルをキャッシュしておくことが有効である。クラウド上で分析処理を行うことで、ユーザ間での学習データの共有が可能となり、構築した分類器を全ユーザに配信することで、新車種の登場に迅速に対応できるようになる。

#### 4.2.6 実装

##### Android スマートフォン

近年のスマートフォンは、我々の車両検知システムをリアルタイムで実行するのに十分な処理性能を有しており、かつ、ほとんどの歩行者がスマートフォンを所持しているため、

十分に活用の余地がある。そこで我々は、提案する車両検知システムを Android アプリケーションとして実装し、住宅街においてデモンストレーション実験を行った。図 4.12 に、Nexus 4 (2012 年製) で車両検知システムが動作する様子を示す。

このデモンストレーションでは、4.2.4 節の環境雑音シナリオで構築した分類器を使用している。図 4.12 の 2 枚の写真は、車両の最接近 10 秒前と 4 秒前に撮影したものであり、白い日産リーフが接近した 4 秒前の時点で画面に警告が表示されていることが確認できる。

### Android Wear スマートウォッチ

スマートウォッチにはマイクとプロセッサ、ディスプレイが搭載されており、スマートフォンと同様に専用アプリケーションを実行可能である。我々は、提案する車両検知システムを Android Wear アプリケーションとしても実装し、実行速度の評価を行った。最新鋭のスマートフォンと比較するとスマートウォッチの計算資源は貧弱であるが、マイクから入力された音声をリアルタイムで処理できた。ここでは、スマートウォッチとして ZenWatch WI500Q を採用し、32,768 サンプル (= 0.683sec) の音声データを 200 ms 以内に処理できることを確認した。Android Wear SDK を利用すると、通知領域に常駐するアプリケーションを実装可能であり、接近車両を検知した場合にバイブレーションでユーザに警告を伝えることが可能である。また、文字盤アプリケーションとして実装して常駐させることも可能である。

### Google Glass への通知

一般に、歩行中にスマートフォンを使用することは、周囲への注意が疎かになり交通事故の原因となることから、好ましい行動ではない。この問題を解決するため、我々は前述の Android アプリケーションと連動してユーザに警告を表示する、Google Glass アプリケーションの実装を行った。すなわち、Android アプリケーションで接近車両を検知した場合に、車両接近を知らせる警告メッセージを Google Glass に表示してユーザに注意喚起する、というものである。Google Glass アプリケーションの開発に際しては、2 種類のアプローチが存在する。1 つ目のアプローチは、Glass Development Kit (GDK) を用いる方法で、カメラやマイクといった低レベルのハードウェアにアクセスしつつ、ネイティブアプリを実装できる。しかしながら、計算資源と電池容量の制約が極めて厳しい Google Glass 上で、長時間に渡って車両検知アプリを実行し続けることは現実的ではない。2 つ目のアプローチは、Mirror API を用いる方法で、Web ベースのサービスと連携してプッシュ通知を送信できる。今回は、車両検知の処理をスマートフォン側に任せられることから、Mirror API を用いて通知を表示する部分のみを実装した。

### 4.2.7 関連研究

本研究は、電気自動車に特有の高周波音に着目し、機械学習を用いて接近車両の検知を行う、我々の知る限りで初めての研究である。本研究の特筆すべき点は、車両側や道路側のインフラ整備を一切必要とせず、歩行者が持つスマートフォンでバックグラウンド実行

が可能であるため、今すぐにでも導入が可能な点にある。この節では、歩行者を交通事故から守るための関連研究を紹介する。これらの提案は、我々の手法と共存が可能であり、併用することで検知精度を向上できる可能性がある点に注意されたい。

車両側において周辺の歩行者や障害物を検知する取り組みは、自動車メーカーを含めて幅広く行われており、すでに商用化されたものも存在する [61, 62, 63, 64]。しかしながら、歩行者側の自衛のための車両検知技術は発展途上にある。

これまで、多くの研究者によって提案されてきたのは、近距離無線通信や携帯電話網を利用した車々間（歩車間）通信システムであり、自動車や歩行者が位置情報を交換することで、互いの接近を検知するものである [65, 66]。これらのシステムは信頼性に優れている反面、車両側と歩行者側の双方に専用機器の設置が必要であり、残念ながら普及には程遠い状況である。言い換えれば、車両側の協力なくしては、歩行者は自分の身を守ることができないという重大な欠点を抱えている。

車両側の設備に頼らない手法としては、画像情報から車両を検知する手法が多く提案されている [67, 68]。Sivaraman らは、動画中の車両を認識し、その移動を追跡する汎用的な学習フレームワークを提案している [69]。また、The cyber-physical bike は自転車の認識に特化したフレームワークである [70]。さらに、Wang らは、スマートフォンのカメラを用いて接近車両をリアルタイムに検知する WalkSafe を提案している [71]。画像処理によるアプローチの長所は、車両の移動方向を推定することで衝突のリスクを評価できる点にある。一方で、短所としては、歩行者が常にカメラの向きを意識し続ける必要がある点や、カメラの消費電力が非常に大きい点が挙げられる。

音声情報から接近車両を検知するというアプローチ自体は、特段新しいわけではない。Tsuzuki らは、LVQ ニューラルネットワークを用いた、スマートフォン向けの車両検知システムを提案している [72]。しかし、この手法は 30 km/h 以上で走行するガソリン車を対象としており、より静粛性の高い電気自動車は検知できないと推定される。加えて、ニューラルネットワークを用いることから、スマートウォッチのような機器には計算負荷が大きい。判定精度に関しては、Tsuzuki らの手法では、30 km/h で走行する車両を最接近の平均 4.3 秒前に検知可能であるが、我々の手法では、同様の状況下で平均 6.5 秒前に検知可能である。

#### 4.2.8 まとめ

本稿では、電気自動車やハイブリッド車に特有の高周波音を手掛かりとして、機械学習で接近車両を検知する手法を紹介した。この高周波音は、車両を駆動するモータの回転速度を制御するために用いられる、PWM 制御のスイッチング操作に起因するものであり、我々の手法はほとんどの商用車に適用可能である。本手法では、正規化した 50Hz 刻みの周波数分布を特徴量として、機械学習により分類器を生成する。我々の評価実験では、接近する車両のみを選択的に検知するだけでなく、車速や車種も区別可能であることが示された。また、環境雑音の違いや車種の違いに対しても十分にロバストであった。理想的な環境における車両の有無の判定精度は 92.8% であり、見落とし率は 5.2% であった。また、歩行者に近づいて来る車両と遠ざかって行く車両を明確に区別できる点も特筆すべきである。最も早い場合では、最接近の 16.5 秒前には車両の存在を検知可能であった。

また、我々はこの車両検知システムを Android アプリケーション、Android Wear アプリケーションとして実装し、Google Glass に警告を表示する連携システムを実装した。計算負荷は十分に小さく、スマートフォンやスマートウォッチ上でも十分にリアルタイム実行が可能であった。

本システムは、車両側や道路側の特別なインフラを必要としないため、歩行者が自衛のために有用だと判断すれば、今すぐにでも導入が可能である。また、既存手法とは異なりカメラやGPSといった電力消費の激しいデバイスを使用しないことから、電池容量の限られたモバイルデバイスやウェアラブルデバイスに適したシステムである。

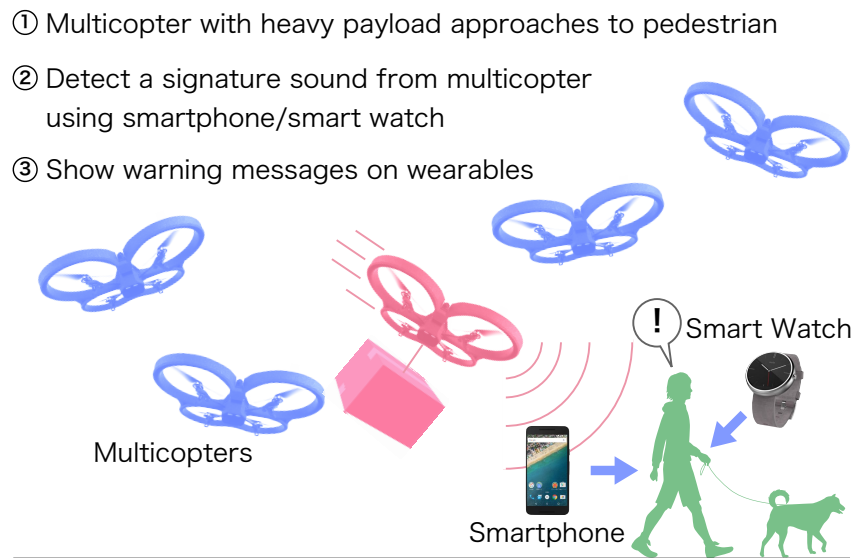


図 4.13: 提案するシステムのコンセプト. 墜落時の危険性に直結するマルチコプタの積載重量を, 機械学習を用いてスマートフォン上のアプリケーションで検知する.

## 4.3 マルチコプタの積載重量の推定

### 4.3.1 背景

モータ駆動式の無人マルチコプタは, 荷物配送から, 施設警備, 農業, 災害調査, 地雷除去に至るまで幅広い産業分野で注目を集めている. 特に荷物配送の分野では, 近年の EC サイトの発達による荷物量の増加に, 慢性的な人手不足といった業界事情が重なり, ラストワンマイルの配送を自動化する手段として大きく期待されている [73]. このような期待の背景には, マルチコプタの持つ極めて高い操縦性がある. すなわち, ヘリコプタのような垂直離着陸やホバリングが可能であり, かつ, ジャイロセンサを用いた自律的な姿勢制御機能を持つため, 高層建築物の多い都市部においても障害物の間を縫って自在に飛び回ることができる. さらに, 従来の航空機より遥かに小型であることから, これまで航空機で実現できなかった地表面に近い空域での活躍を期待できる.

マルチコプタのコンセプト自体は 20 世紀初頭に提案されたものであるが, 当時航空機の動力源として用いられていた内燃機関では各ロータの出力を瞬時に調整することが難しく, また, 姿勢制御技術も未発達であったことから, 飛行可能な機体の実装は長らく困難であった. 1990 年代に入ってリチウムイオン電池が商用化されると, モータ駆動により各ロータの出力を瞬時に調整することが可能となったが, フライホイールを内蔵した機械式ジャイロセンサは小型化・軽量化が難しく, 小型マルチコプタへの搭載は困難であった. 2000 年代に入り, MEMS ジャイロセンサが実用化されてセンサモジュールの小型化が進み, また, スマートフォンにジャイロセンサが標準搭載されたことで大量生産による低廉化が進んだため, 高性能で廉価なマルチコプタが多く商品化されるに至った.

マルチコプタは幅広い産業分野から高い期待を集める一方で, 飛行空域が人間の活動領域に近いことから安全面の懸念が多いのも事実である. マルチコプタの姿勢制御は全ロー

タの出力が均一であることを前提として設計されており、モータの劣化などによる各ロータの不均一な出力低下が致命的な問題に直結しうる。ここでは、現時点で最も普及が進んでいるクアッドコプタを例として、4枚のロータ（時計回りにA,B,C,Dとする）のうち1枚（ロータA）に問題が発生して最大出力が80%まで低下した場合を考える。まず、出力の不均衡による機体の転覆を抑止するため、対角線上に位置するロータCの出力を同程度まで低下させる必要がある。次に、ロータの回転モーメントによる機体の旋回運動を抑止するため、左回りのロータと右回りのロータとで出力の総和を均衡させる必要がある。ここで、マルチコプタは隣り合うロータが互いに逆回転することで回転モーメントを打ち消し合うよう設計されているため、既に出力が80%に低下しているロータAとロータCは回転方向が同一である。従って、回転モーメントを均衡させるためには、両隣に位置するロータBとロータDの出力を同程度まで低下させる必要がある。結果として、全ロータの出力を80%まで低下させる必要に迫られる。このように、クアッドコプタのシステム全体の出力は、最も出力の低いロータにより律速されるため、ロータの動力源が1つでも喪失すると墜落を避けられない。また、ヘリコプタが可変ピッチの回転翼を採用しているのに対して、ほとんどのマルチコプタは固定ピッチの回転翼を採用しているため、主動力の喪失時にオートローテーションによる軟着陸を行うことが原理上できない。さらに、固定翼を持たないため、グライダーのように滑空飛行を行い高度を緩やかに下げることができない。すなわち、従来の航空機とマルチコプタの決定的な違いは、主動力の喪失または出力低下が墜落に直結する点にある。

日本国内では2015年の航空法改正により、私有地を含めて人口集中地区の上空や、建築物から30メートル以内での飛行が原則禁止された。このような法規制はマルチコプタの利便性を損ねるものであり、必要十分な安全性を技術的に担保した上で規制緩和を求めるのが望ましい。そこで本研究では、マルチコプタの飛行音に着目し、機械学習を用いてマルチコプタの積載重量を推定する手法を提案する（図4.13）。これにより、積載重量が多く危険性の高いマルチコプタを選択的に検知できる。

マルチコプタの飛行音には、各ロータの回転によって生じる風切音が含まれており、積載重量の違いによりロータの回転数が増えるため、基本周波数が増える。他方で、鉛直方向の加速度運動によっても基本周波数は増える。水平移動や旋回運動の際には基本周波数が分裂する。また、観測者との相対速度が増えるとドップラー効果により見かけの周波数が増えるなど、外乱要因も多く含まれる。4.2節で紹介した電気自動車の走行音とは対照的に、マルチコプタの飛行音には低周波成分の周期的なピークが多く見られる。このような低周波音は人間にとって聞き取りやすいものの、複数のマルチコプタが飛び交う中で積載重量が多く危険性が高いものを判別することは容易ではない。本研究で提案するマルチコプタの積載重量の推定ロジックは、スマートフォン上でリアルタイムに実行できるほどシンプルかつ軽量であり、ユーザは自衛のために気軽に利用できる。

本章の構成は以下の通りである。まず4.3.2節では、マルチコプタの姿勢制御メカニズムと実機を用いて録音した飛行音の特徴を紹介し、どのような要素が飛行音に影響を与えるかについて論じる。次に4.3.3節では、我々の提案する積載重量推定システムの概要と機械学習を用いた学習フェーズにおける録音データの取り扱いや特徴量について述べる。さらに、4.3.4節と4.3.5節で我々の提案するシステムの性能を多面的に評価する。4.3.6節で関連研究を紹介し、4.3.7節で本章の内容をまとめる。

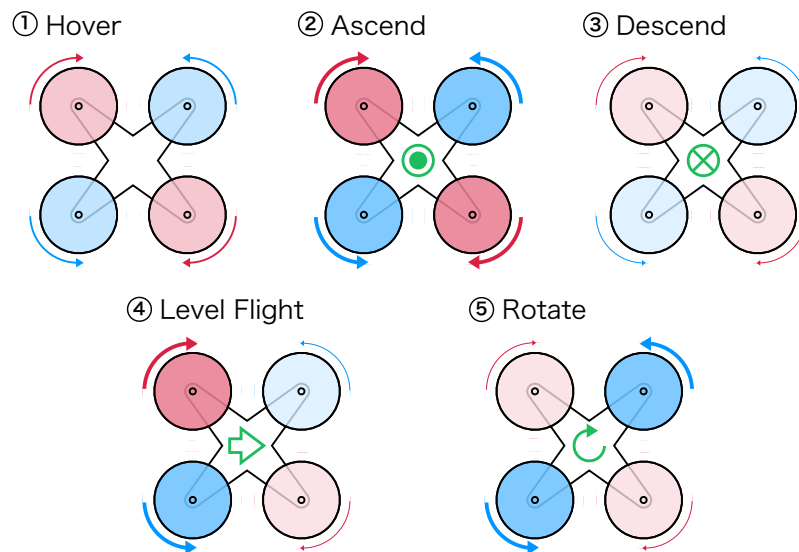


図 4.14: マルチコプタの飛行と姿勢制御のメカニズム

### 4.3.2 マルチコプタの飛行音

#### 飛行と姿勢制御のメカニズム

マルチコプタは3枚以上のロータを搭載した回転翼機であり、各ロータは回転軸の直下に搭載されたモータにより個別に駆動される。ほとんどのマルチコプタは迎角が一定の固定ピッチ翼を採用しており、モータの回転数によって推力を調節し重力に対抗する。各モータの回転速度は、ジャイロセンサや加速度センサの情報およびユーザからの操縦指令に基づいて、フライトコントローラにより個別に制御される。固定翼を持たないマルチコプタは、揚力の全てを回転翼に依存しており、各ロータの推力をバランス良く制御することで、上昇・ホバリング・下降の動作を実現する。また、意図的に推力を偏らせることで機体を傾斜させ、推力を斜め上向きにすることで水平移動を行う。さらに、通常はロータの半数を左回り、半数を右回りとすることで互いの回転モーメントを打ち消しているが、これを意図的に不均衡させることで旋回動作を実現する。従って、鉛直上向きの加速度が大きいほどモータの回転数が上昇し、また水平移動時や旋回時にはモータの回転数が一様でなくなる。ロータの枚数が多いほど制御の自由度が増すが、ロータが4枚のクアッドコプタではほぼ自由度がなく、図 4.14 に示す制御方式に集約される。

#### 飛行音の録音

マルチコプタの飛行音を分析して積載重量推定の手掛かりとなる特徴を見出すため、我々はマルチコプタ実機を用いて飛行音の収録を行った。マルチコプタについては DJI 社の Phantom 2 Vision+ を採用し、実験場所については法令遵守の観点から人口集中地区を避け、埼玉県内の荒川河川敷（秋ヶ瀬公園）を採用した。また、フライトリミット機能を用いてマルチコプタの飛行範囲を半径 20 m 以内、高さ 20 m 以内に制限し、公園内の設備や周囲の利用者に接触しないよう安全に配慮した。録音機器としては iPhone 5s を採用し、





図 4.15: マルチコプタへの貨物の積載方法. 脚部に重量 100 g のカゴを取り付け, その中に 200 g または 400 g の重りを配置した.

表 4.8: Phantom 2 Vision+ のロータの回転数

機体の状態	(A) 回転数	(B) 翼の枚数	(A)×(B)
アイドリング時	2,000 rpm (33.3 rps)	2 枚	66.6 Hz
フルスロットル時	7,925 rpm (132.1 rps)	2 枚	264.2 Hz
図 4.16 のピーク間隔の最大値			≈240 Hz

後述するラベル付け作業に役立つよう飛行の様子を動画で記録しながら音声も同時に収録した. 録音フォーマットは圧縮による劣化の少ない AAC (Advanced Audio Coding) 形式とし, サンプリングレートは 44.1 kHz, 量子化ビット数は 16 bit で統一した.

録音に際しては, 図 4.15 に示すように Phantom 2 Vision+ の脚部に重量 100 g のカゴを取り付け, その中に 200 g または 400 g の重りを配置した. カゴを取り外した場合の積載重量を 0 g とすると, カゴを含めた積載重量は 100 g, 300 g, 500 g の 3 通りである. なお, Phantom 2 Vision+ の公称機体重量は 1,242 g であり, 積載重量 500 g は機体重量の約 40% にあたる.

### 計測結果

図 4.16 に計測したマルチコプタの飛行音のスペクトログラムを示す. 4.2 節で紹介した電気自動車の走行音とは対照的に, マルチコプタの飛行音では低周波成分の周期的なピークが多く見られる. この低周波音はロータの風切音に起因するもので, 基本周波数が (モータの回転数) × (ロータの翼の枚数) に一致する. 表 4.8 は, レーザー式の非接触デジタル回転計 (DT2234B) で Phantom 2 Vision+ のロータの回転数を計測したものであり, アイドリング時で分速 2,000 回転弱, フルスロットル時で分速 8,000 回転弱であった. Phantom

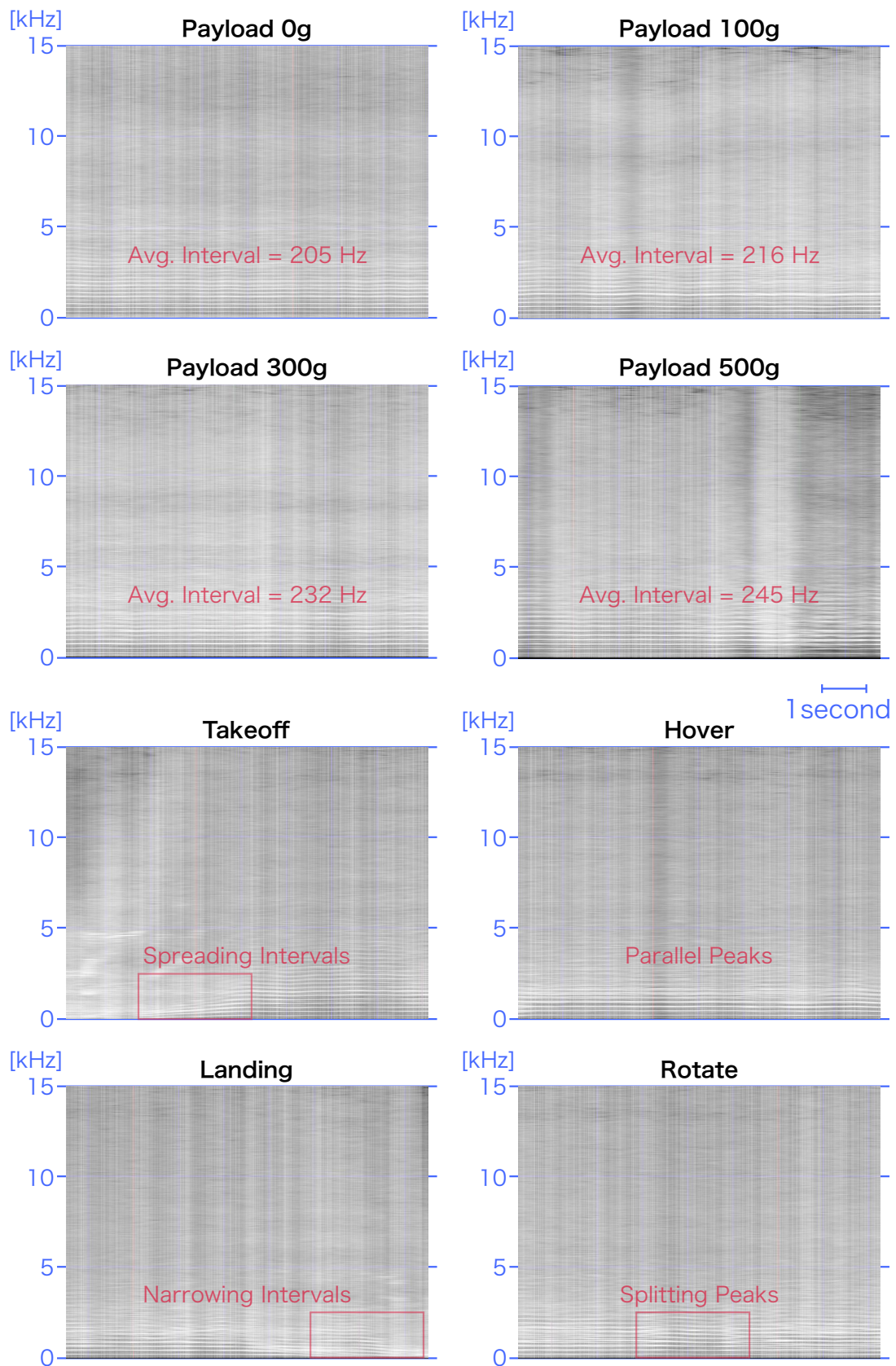


図 4.16: マルチコプタの飛行音のスペクトログラム. 低周波成分にロータの風切音に起因する周期的なピークが多く見られる.

表 4.9: 積載重量から予想される基本周波数と実際の基本周波数

積載重量	0 g	100 g	300 g	500 g
予想値	205 Hz	213 Hz	228 Hz	243 Hz
実測値	205 Hz	216 Hz	232 Hz	245 Hz

2 Vision+のロータは2枚の翼を備えるため、風切音の基本周波数は67~264 Hzとなる。

マルチコプタはロータの推力のみで重力に対抗しており、ロータの静止推力  $T$  は以下の式で求められる。

$$T = D^3 \times P \times N^2 \times K \quad (4.2)$$

ここで、 $D$  はロータの直径 (定数)、 $P$  はロータのピッチ (定数)、 $N$  はロータの回転数 (可変)、 $K$  はプロペラ係数 (定数) である。前述の通り、マルチコプタはロータの推力のみで重力に対抗して飛行するため、積載重量を含めた総重量はロータの回転数の2乗に比例する (表 4.9)。従って、無積載時の機体重量とロータの回転数が既知であれば、ロータの回転数から積載重量を推測できる。

離陸直前のスペクトラムからは、ロータの回転数の増加に伴って周期的なピークの間隔が広がっていく様子が読み取れる。逆に、着陸直後のスペクトラムからは、ロータの回転数の減少に伴うピーク間隔の狭まりが読み取れる。一方、滞空中のスペクトラムでは、上昇中、ホバリング中、下降中のいずれにおいてもピーク間隔がほぼ一定で推移している。他方で、移動・旋回時には所々にピーク周波数の分裂が見られる。これは、図 4.14 に示した飛行制御メカニズムに従って、各ロータの回転数が個別に制御されていることを示すものである。

### 4.3.3 積載重量の推定

計算資源や電池容量に厳しい制約のあるスマートフォン上で、積載重量の推定をリアルタイムで実行するためには、推定ロジックがシンプルかつ軽量であることが求められる。それと同時に、周辺環境の騒音の違いや録音機器の違いに対してロバストである必要がある。さらに、飛行音に含まれる有益な情報を最大限に有効活用できることが望ましい。これらの要件を満たすべく、我々は機械学習によるアプローチを採用した。提案する飛行状態の推定ロジックの概要を図 4.17 にフローチャートで示す。

#### 特徴量ベクトル

機械学習における分類器の構築のため、我々はまず 4.3.2 節で収録した録音データから特徴量ベクトルの抽出を行った。ここでは、まず、移動幅 100ms のスライディングウィンドウを設定して音声データを 32,768 サンプルずつ切り取り、高速フーリエ変換 (FFT) を用いて周波数領域のエネルギー分布を算出する。次に、5 Hz 刻みでの平均エネルギーを集計し、音量の違いを補正するため全周波数帯での平均エネルギーで正規化する。さらに、明瞭なピーク周波数が見られる 2.5 kHz 以下の帯域について、 $\Delta = 5 \sim 1,000$  Hz の範

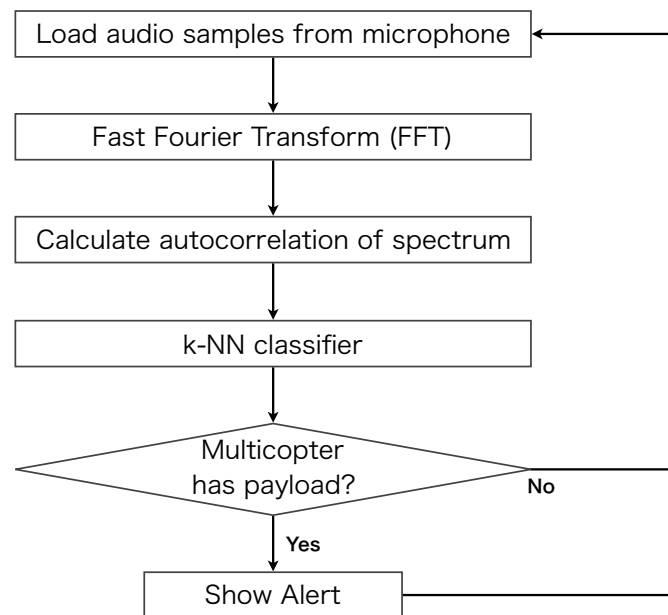


図 4.17: 提案する積載重量推定システムのフローチャート。分類器は事前に学習データを用いて構築しておく。

囲でスペクトルの自己相関を計算し基本周波数を得る。ここでエネルギー分布の集計を 5 Hz 刻みとしたのは、可能な限り細かい粒度でキリの良い数字を選んだためである。32,768 サンプルでの FFT の解像度は、サンプリング周波数が 44.1 kHz の場合で約 1.35 Hz, 48 kHz の場合で約 1.46 Hz, 96 kHz の場合で約 2.93 Hz である。

### ラベリング

次に、録音データから抽出した特徴量ベクトル 1 つ 1 つに対して、マルチコプタの積載重量を示すラベルを手作業で付与した。ここでは、“payload-0g”（積載重量 0 g），“payload-100g”（積載重量 100 g），“payload-300g”（積載重量 300 g），“payload-500g”（積載重量 500 g）の 4 種類のラベルを用意した。これらのラベルは、次節での分類器の構築において正解データとして利用される。

### 教師あり学習

教師あり学習を行うため、我々は WEKA というデータマイニングツールを利用した。後述する通り、分類器の候補として k 近傍分類 (k-NN)、サポートベクタマシン (SVM)、ランダムフォレスト、J48 決定木、CART 決定木を検討し、最終的に判定精度の最も高い k 近傍分類 (k-NN) を採用した。4.3.4 節では分類器として k 近傍分類 (k-NN) を用いて学習パラメータを調整した場合の性能評価を行い、4.3.5 節では他の分類器を用いた場合の性能を紹介する。

表 4.10: k-NN でのマルチコプタの積載重量の推定精度。総合的な判定精度は 82.8%であった。

		Recognition Result			
		0 g	100 g	300 g	500 g
Ground Truth	0 g	100.0%	0.0%	0.0%	0.0%
	100 g	0.2%	87.9%	11.9%	0.0%
	300 g	0.0%	2.9%	72.4%	24.7%
	500 g	1.6%	6.1%	18.6%	73.7%

#### 4.3.4 性能評価

##### k 近傍分類による積載重量の推定

まずは、最も単純な分類器の1つであるk近傍分類を用いて、マルチコプタの積載重量の推定性能の評価を行った。ここでは、1つのフライトデータをテストデータとして選択し、他の全フライトデータを用いて分類器を構築した。次に、テストデータから抽出した特徴量ベクトルを構築した分類器に投入し、出力が正解ラベルと一致するかどうかを確認した。この手順を繰り返し、全てのフライトデータで1回ずつ性能評価を行った。前述した通り、積載重量の区分は4種類であり、学習データのインスタンス数は合計2,210個であった。その内訳は、“payload-0g”が510個、“payload-100g”が530個、“payload-300g”が550個、“payload-500g”が620個、である。

マルチコプタの積載重量の推定精度を表4.10に示す。本手法では、4種類の積載重量を82.8%という高い精度で判別できる。“payload-100g”と“payload-300g”、“payload-300g”と“payload-500g”の間で若干の判定ミスが見受けられるが、積載の有無については99.4%と非常に高い精度で判別できる。

##### 学習パラメータの調整

上記の評価では、k近傍分類における近傍標本数を $k=1$ としたが、 $k$ を変化させた場合の総合性能を表4.11に示す。ここでは概ね $k$ が小さいほど高い判定精度が得られた。これは特徴量ベクトルの混在領域が少なく、境界領域付近での判定にのみ影響が出るためと考えられる。なお、WEKAで設定可能な他のパラメータ（distanceWeighting：距離の重み付けの有無、meanSquared：平均絶対誤差の代わりに平均二乗誤差を用いるか）については、判定性能への影響が見られなかった。



表 4.11: k 近傍分類における近傍標本数  $k$  を変化させた時の判定精度

	k (number of nearest neighbors)									
	1	3	5	7	11	21	51	101	201	501
Total Accuracy (4 Category)	82.8%	82.7%	82.7%	82.7%	82.7%	82.7%	82.7%	82.7%	81.2%	75.5%

### 4.3.5 検討事項

#### k 近傍分類以外の分類器

前節では k 近傍分類を採用した場合の性能評価を行ったが、ここでは他のアルゴリズムでの性能評価を行う。結果として、k 近傍分類の判定精度が最も高かったため、我々は k 近傍分類を採用することとした。

**サポートベクタマシン** まず、分類性能の高い学習モデルとして知られるサポートベクタマシンでの推定性能を表 4.12 に示す。なお、ここでコストパラメータは調整の結果  $c = 2$  に設定してある。k 近傍分類に次いで判定精度が高いものの、“payload-300g” の判定ミスが目立ち、総合的な判定精度は 78.9% であった。また、k 近傍分類と比較すると、“payload-100g” と “payload-500g” といった非隣接クラス間での判定ミスもやや多い。

表 4.12: サポートベクタマシンでのマルチコプタの積載重量の推定精度。総合的な判定精度は 78.9% であった。

		Recognition Result			
		0 g	100 g	300 g	500 g
Ground Truth	0 g	100.0%	0.0%	0.0%	0.0%
	100 g	0.6%	90.0%	9.4%	0.0%
	300 g	0.0%	16.2%	50.4%	33.5%
	500 g	1.6%	9.7%	11.5%	77.3%

**ランダムフォレスト** 次に、決定木を用いた集団学習アルゴリズムとして知られるランダムフォレストでの推定性能を表 4.13 に示す。サポートベクタマシンとは異なり、“payload-500g” の誤判定が目立ち、総合的な判定精度は 75.0% に留まった。ランダムフォレストでは説明変数無作為に選択するため、ピーク周波数の周期を捉えた特徴量が全て説明変数として利用されるとは限らず、ここで情報量の欠落が発生するために精度が伸び悩むものと考えられる。

表 4.13: ランダムフォレストでのマルチコプタの積載重量の推定精度. 総合的な判定精度は 75.0%であった.

		Recognition Result			
		0 g	100 g	300 g	500 g
Ground Truth	0 g	100.0%	0.0%	0.0%	0.0%
	100 g	0.2%	80.8%	11.9%	7.2%
	300 g	0.0%	2.9%	72.4%	24.7%
	500 g	1.6%	6.1%	40.5%	51.8%

**J48 決定木** さらに, C4.5 アルゴリズムの Java 実装版である J48 決定木での推定性能を表 4.14 に示す. ランダムフォレストと同様に, “payload-500g” を “payload-300g” と誤判定する事例が目立ち, 総合的な判定精度は 76.7%に留まった.

表 4.14: J48 決定木でのマルチコプタの積載重量の推定精度. 総合的な判定精度は 76.7%であった.

		Recognition Result			
		0 g	100 g	300 g	500 g
Ground Truth	0 g	100.0%	0.0%	0.0%	0.0%
	100 g	0.2%	87.9%	11.9%	0.0%
	300 g	0.0%	2.9%	72.4%	24.7%
	500 g	1.6%	6.1%	40.5%	51.8%

**CART 決定木** 最後に, ベーシックな二分決定木として知られる CART 決定木での推定性能を表 4.15 に示す. J48 決定木と同様に, “payload-500g” を “payload-300g” と誤判定する事例が目立ち, 総合的な判定精度は 76.7%に留まった.

表 4.15: CART 決定木でのマルチコプタの積載重量の推定精度. 総合的な判定精度は 76.7%であった.

		Recognition Result			
		0 g	100 g	300 g	500 g
Ground Truth	0 g	100.0%	0.0%	0.0%	0.0%
	100 g	0.2%	87.9%	11.9%	0.0%
	300 g	0.0%	2.9%	72.4%	24.7%
	500 g	1.6%	6.1%	40.5%	51.8%

### 未知の機種への対応

ここまで検証してきた通り、飛行音が既知で学習データとして利用可能なマルチコプタについては、我々の手法により非常に精度良く積載重量を推定できる。しかしながら、ピーク周波数の間隔など飛行音の特徴が異なる機種に適用可能とするためには、機種ごとの飛行音を学習データとして追加する必要がある。4.2節で紹介した電気自動車の検知手法とは対照的に、k近傍分類では事前の決定木構築が不要であるため、単に新しい学習データを各スマートフォンに配信するだけで良い。ただし、データ数が増えると最近傍標本の探索に時間が掛かるため、判定に影響が出ない範囲で適宜データの間引きを行うことが望ましい。この処理については、学習データを配信するサーバ側で行えば良い。

### 4.3.6 関連研究

マルチコプタのロータで生じる風切音に着目して、マルチコプタの積載重量を推定する研究は、我々の知る限りこれまで存在しなかった。従って、本研究は従来検討されて来なかった全く新しい領域を切り拓いたものである。この節では、マルチコプタの飛行状態推定や位置推定に関連する研究事例を紹介する。

これまでのマルチコプタに関する研究事例の大部分は、マルチコプタ自体の姿勢制御または自己位置推定を実現するためのものであった [74, 75, 76]。例えば Saied らは、8つのロータを持つオクタコプタにおいて、各ロータの出力変化に対する機体の挙動を分析することでロータ駆動系での障害発生を検知し、墜落しないよう残存ロータの冗長制御を行う手法を提案している [77]。他には、気圧データから飛行高度や鉛直方向の速度を推定する技術も多く提案されてきた [78, 79]。また、我々の手法とは対照的に、マルチコプタがカメラ画像を用いて周囲の物体を認識し追尾する技術は既にいくつか提案されている [80, 81]。これらの技術は、マルチコプタの安全性や機能性を高めるものではあるが、周囲の歩行者が自衛のために用いる技術ではない。

他方で、空間中を飛び回る物体を認識する技術は多く提案されてきた。特に、航空管制の分野ではレーダーが広く用いられるが、マルチコプタのような小型機は投影面積が少なく反射波が少ないために検知が困難であり、また、建築物などの障害物が多い低空域での検知能力には限界がある [82]。同じく電波を用いる方式として、マルチコプタが発する通信電波に着目し、その存在を検知する手法も提案されているが、完全な自律飛行を行うマルチコプタは通信電波を発しないため検知できないという問題がある [83]。電波を用いない方式としては、カメラと画像処理を用いたアプローチも検討されている [84, 85]。これらの画像処理によるアプローチでは、飛行体の進路や速度を正確に推定できるが、カメラの消費電力が非常に大きいためスマートフォン上で検知ロジックを常時実行し続けるのは無理がある。また、カメラの視野角の範囲内にある飛行体しか検知できないのも欠点であり、物陰から不意にマルチコプタが飛び出すような場面では検知が大幅に遅れてしまう。

これに対して、本手法ではマルチコプタの飛行に際して必然的に発生し、周囲の物体にも遮蔽されにくい飛行音に着目した。これにより、飛行体の正確な進路や速度までは検出できないものの、積載重量が多く危険性が高いマルチコプタのみを選択的に検知できるようにしつつ、視野角や消費電力の問題を大幅に改善できた。



また、コピキタスコンピューティングの分野においても、マルチコプタを活用した提案が多くなされてきた [86, 87, 88, 89]. Flying Display は、マルチコプタを用いて3次元空間中にプロジェクタとスクリーンを配置することで、場所の制約を受けることなくユーザーに情報を提示することを可能とする技術である [90]. 一方、EverCopter は、複数のマルチコプタをマグネット吸着式の電源ケーブルで動的に接続することで、バッテリーを増設することなくマルチコプタの滞空時間を飛躍的に伸ばす技術である [91]. これらの技術は、我々の提案と相補的な関係にあり、我々の手法と組み合わせることで、マルチコプタの安全性をさらに向上させることができる。

#### 4.3.7 まとめ

本稿では、マルチコプタの飛行音に着目し、機械学習を用いてマルチコプタの積載重量を推定する手法を提案した。これにより、積載重量が多く危険性の高いマルチコプタを選択的に検知できるようになった。マルチコプタの飛行音には、各ロータの回転によって生じる風切音が含まれており、ロータの回転数に対応した周期的なピーク周波数が見られた。本手法では、2.5 kHz 以下の低周波領域における正規化したエネルギー分布の自己相関を手掛かりとして、機械学習により分類器を生成する。我々の評価実験では、積載重量の多いマルチコプタのみを選択的に検知できることが示された。評価実験における4種類の積載重量の判定精度は82.8%、積載の有無の判定精度は99.4%であった。また、本稿で提案したマルチコプタの積載重量の推定ロジックは、スマートフォン上でリアルタイムに実行できるほどシンプルかつ軽量であった。

本システムは、マルチコプタ側の特殊な装備や周囲の特別なインフラを必要としないため、ユーザが有用だと判断すれば、自衛のために今すぐ利用できる点が特徴的である。また、既存手法とは異なり、電力消費の激しいカメラを使用しないことから、電池容量に厳しい制約のあるモバイル機器に適したシステムである。

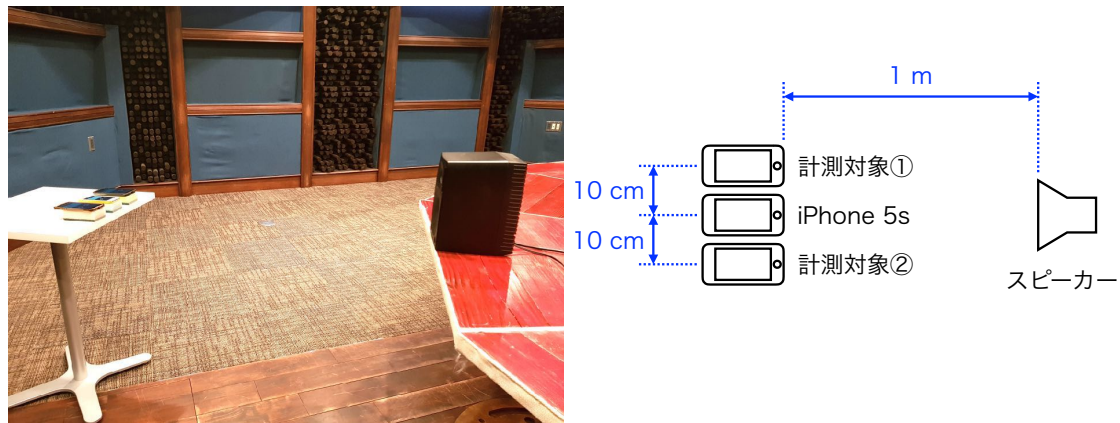


図 4.18: 周波数特性の計測環境と配置図

## 4.4 マイクの周波数特性に関する機種依存の影響評価

### 4.4.1 背景

本章で提案した電気自動車の接近検知手法やマルチコプタの積載重量の推定手法では、走行音や飛行音をスマートフォンの内蔵マイクで録音して捉える必要がある。このような音に着目したセンシング手法の性能は、録音に用いるスマートフォンの内蔵マイクの特性に影響を受ける。一般にマイクの特性には、周波数特性、感度特性、指向性、の3種類があるが、我々のセンシング手法では周波数領域における正規化したエネルギー分布を機械学習の特徴量として採用しているため、感度特性や指向性に比べて周波数特性が重要な指標となる。そこで本節では、これまでの録音実験に用いてきた iPhone 5s と 100 機種 of Android 端末について周波数特性を計測し、その特性の違いがセンシング性能に与える影響を評価する。

### 4.4.2 周波数特性の計測

まず我々は、表 4.16 に示した 100 機種 of Android 端末について、周波数特性を調べるための録音実験を行った。計測対象 100 機種の内訳はスマートフォンが 97 機種、タブレットが 3 機種（端末番号 #08, #09, #10）であり、少なくとも 15 社の製造メーカーが含まれる（#01, #24, #25, #100 の製造メーカーは不明）。録音アプリの制約により Android 4.0 以上の端末に限定されたものの、発売時期が 2010 年～2017 年に幅広く分布する多種多様なハードウェアを用意した。計測環境としては、防音設備が整っていて、外部からの騒音が少なく、音の反響が少ない、DMM.make AKIBA 12 階のジャッジルームを採用した（図 4.18 左）。計測に際しては、高さ約 75 cm の地点にスピーカーを配置し、その真正面の 1 m 離れた地点に計測対象のスマートフォンを配置した（図 4.18 右）。周波数特性の基準データとなる iPhone 5s を挟む形で両脇に 1 台ずつスマートフォン配置した上で、スピーカーから 15 秒間のホワイトノイズを含む音声を再生した。iPhone 5s を含めて 3 台同時に録音を行い、後に録音データの周波数分布の比較を行う。この操作を 50 回繰り返して、全 100 機種分の録音データを採取した。計測に用いたスピーカーは GENELEC 社の 1029A

表 4.16: 計測対象のスマートフォン

#	機種名	OS	#	機種名	OS
01	Priori 3 LTE	5.1	51	isai FL LGL24	5.0
02	HUAWEI P9 lite	7.0	52	isai VL LGV31	5.0
03	Nexus S	4.1	53	LG G2 mini LG-D620J	4.4
04	GALAXY Nexus	4.3	54	Motorola RAZR IS12M	4.0
05	Nexus 4	5.1	55	Motorola RAZR M 201M	4.1
06	Nexus 5	6.0	56	MEDIAS ES N-05D	4.0
07	Nexus 6	7.1	57	MEDIAS X N-07D	4.1
08	Nexus 7 (2012)	5.1	58	MEDIAS U N-02E	4.1
09	Nexus 7 (2013)	6.0	59	Disney Mobile N-03E	4.1
10	Nexus 9	7.1	60	MEDIAS X N-04E	4.1
11	ZenFone 2 Laser ZE550KL	6.0	61	MEDIAS W N-05E	4.1
12	ZenFone Go ZB551KL	5.1	62	VEGA PTL21	4.1
13	BlackBerry PRIV STV100-1	5.1	63	Xperia NX SO-02D	4.0
14	G'zOne TYPE-L CAL21	4.0	64	Xperia acroHD SO-03D	4.0
15	ARROWS X LTE F-05D	4.0	65	Xperia GX SO-04D	4.1
16	Disney Mobile F-08D	4.0	66	Xperia SX SO-05D	4.1
17	ARROWS X F-10D	4.2	67	Xperia AX SO-01E	4.1
18	ARROWS Me F-11D	4.0	68	Xperia A SO-04E	4.2
19	ARROWS X F-02E	4.1	69	Xperia Z1 SO-01F	4.4
20	ARROWS Kiss F-03E	4.1	70	Xperia Z1 f SO-02F	4.4
21	ARROWS Z ISW13F	4.0	71	Xperia A2 SO-04F	4.4
22	ARROWS ef FJL21	4.1	72	Xperia Z3 Compact SO-02G	6.0
23	ARROWS Z FJL22	4.2	73	Xperia X Compact SO-02J	7.0
24	SAMURAI MIYABI FTJ152C	5.1	74	Xperia acroHD IS12S	4.0
25	MUSASHI FTJ161A	5.1	75	Xperia VL SOL21	4.1
26	EVO 3D ISW12HT	4.0	76	Xperia UL SOL22	4.2
27	HTC J ISW13HT	4.1	77	Xperia Z Ultra SOL24	4.4
28	HTC J butterfly HTL21	4.1	78	Xperia Z3 SOL26	5.0
29	HTC J One HTL22	4.4	79	GALAXY S2 SC-02C	4.0
30	HTC J butterfly HTL23	5.0	80	GALAXY S2 LTE SC-03D	4.0
31	HTC J butterfly HTV31	6.0	81	GALAXY Note SC-05D	4.1
32	Ascend HW-01E	4.0	82	GALAXY SIII SC-06D	4.1
33	honor6 Plus PE-TL10	5.1	83	GALAXY Note2 SC-02E	4.3
34	honor8 FRD-L02	7.0	84	GALAXY S4 SC-04E	5.0
35	URBANO PROGRESSO	4.0	85	GALAXY Note3 SC-01F	5.0
36	DIGNO S KYL21	4.0	86	GALAXY J SC-02F	5.0
37	URBANO L01	4.2	87	GALAXY SII WiMAX ISW11SC	4.0
38	INFOBAR A03	4.4	88	GALAXY SIII Progre SCL21	4.1
39	TORQUE G01	4.4	89	GALAXY S5 SCL23	6.0
40	TORQUE SKT-01	4.2	90	Galaxy S6 edge SCV31	7.0
41	Optimus it L-05D	4.1	91	Galaxy S8 SM-G950FD	7.0
42	Optimus Vu L-06D	4.0	92	AQUOS PHONE SH-06D	4.0
43	Optimus LIFE L-02E	4.1	93	SH-01F DRAGON QUEST	4.2
44	Optimus it L-05E	4.2	94	AQUOS PHONE SL IS15SH	4.0
45	G2 L-01F	4.2	95	AQUOS PHONE CL IS17SH	4.0
46	Disney Mobile DM-01G	5.0	96	AQUOS PHONE SERIE SHL21	4.1
47	Optimus X IS11LG	4.0	97	AQUOS SERIE SHL25	5.0
48	Optimus G LGL21	4.0	98	Disney Mobile DM013SH	4.0
49	isai LGL22	4.4	99	YotaPhone 2 YD201	6.0
50	G Flex LGL23	4.2	100	Model RETRO MD-02P	7.0

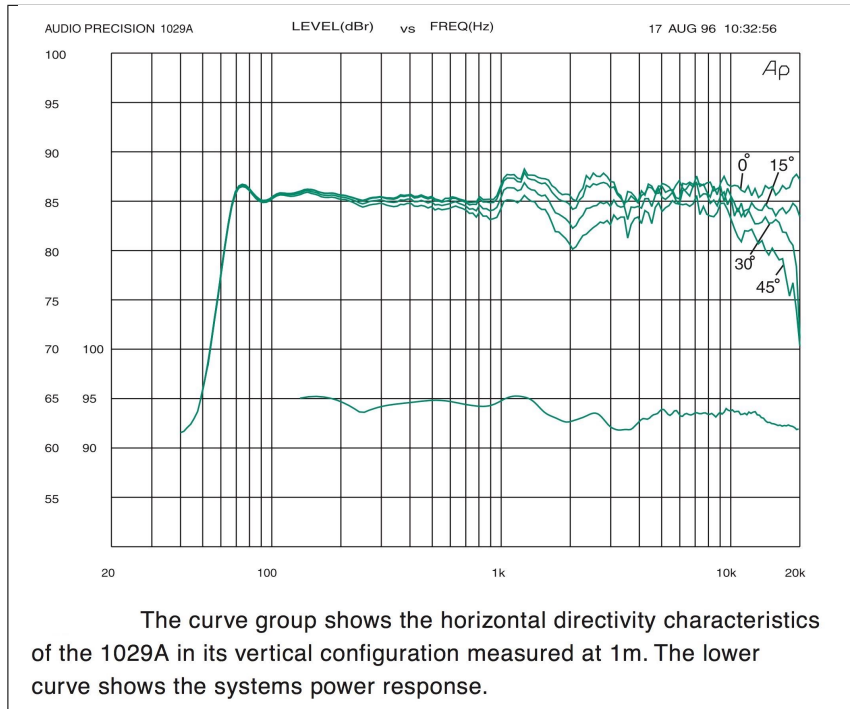


図 4.19: スピーカー GENELEC 1029A の周波数特性 (データシート [92] から抜粋)

であり、図 4.19 に示したデータシートによると [92], 正面から左右に 15 度以内の範囲では周波数特性に大きな違いはない。なお、今回の計測実験の配置では、スピーカーの正面から iPhone 5s の左右に配置したスマートフォンまでの角度は 6 度未満であった。

#### 4.4.3 Android 端末 100 機種別の周波数特性

図 4.20～図 4.24 は、iPhone 5s および 100 機種別の Android 端末で録音した 15 秒間の音声データのスペクトル分布を示したものである。黒色が iPhone 5s のスペクトル分布を、赤色、橙色、緑色、青色、紫色が Android 端末のスペクトル分布を表しており、凡例の番号は表 4.16 に記載した端末番号と対応している。なお、図の右側に示した包絡線は、ケプストラムの 44 次以下の低次成分を抽出したものである。

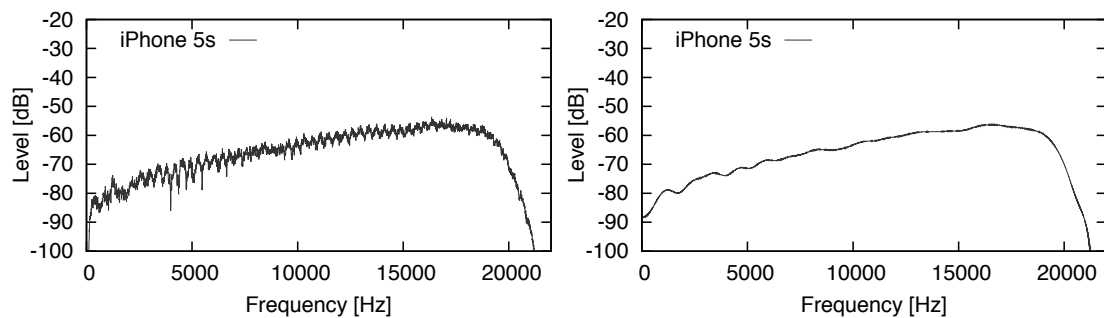


図 4.20: iPhone 5s で録音したホワイトノイズのスペクトル分布と包絡線

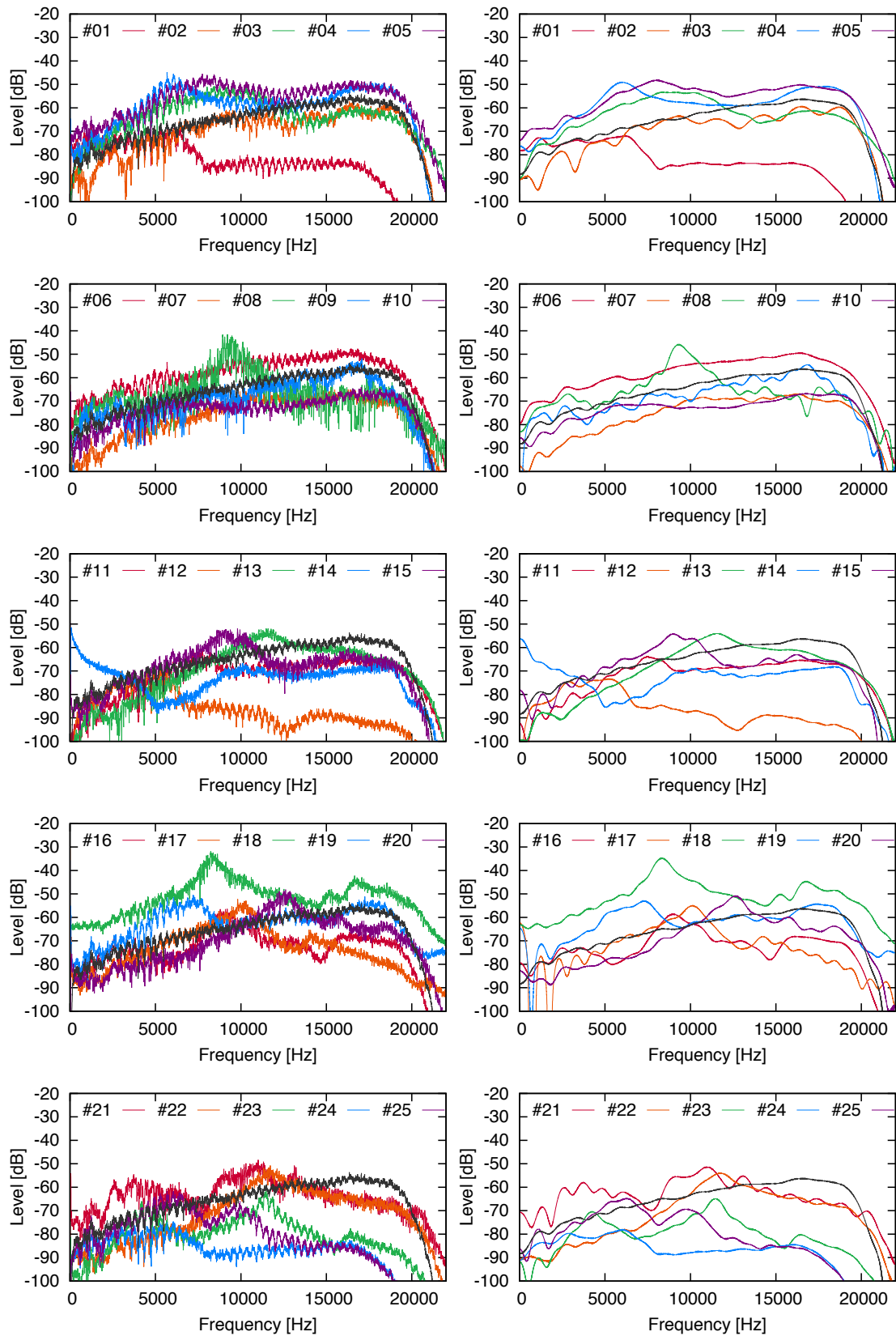


図 4.21: iPhone 5s と 100 機種 の Android 端末 で録音したホワイトノイズのスペクトル分布と包絡線 (端末番号 #01 ~ #25)

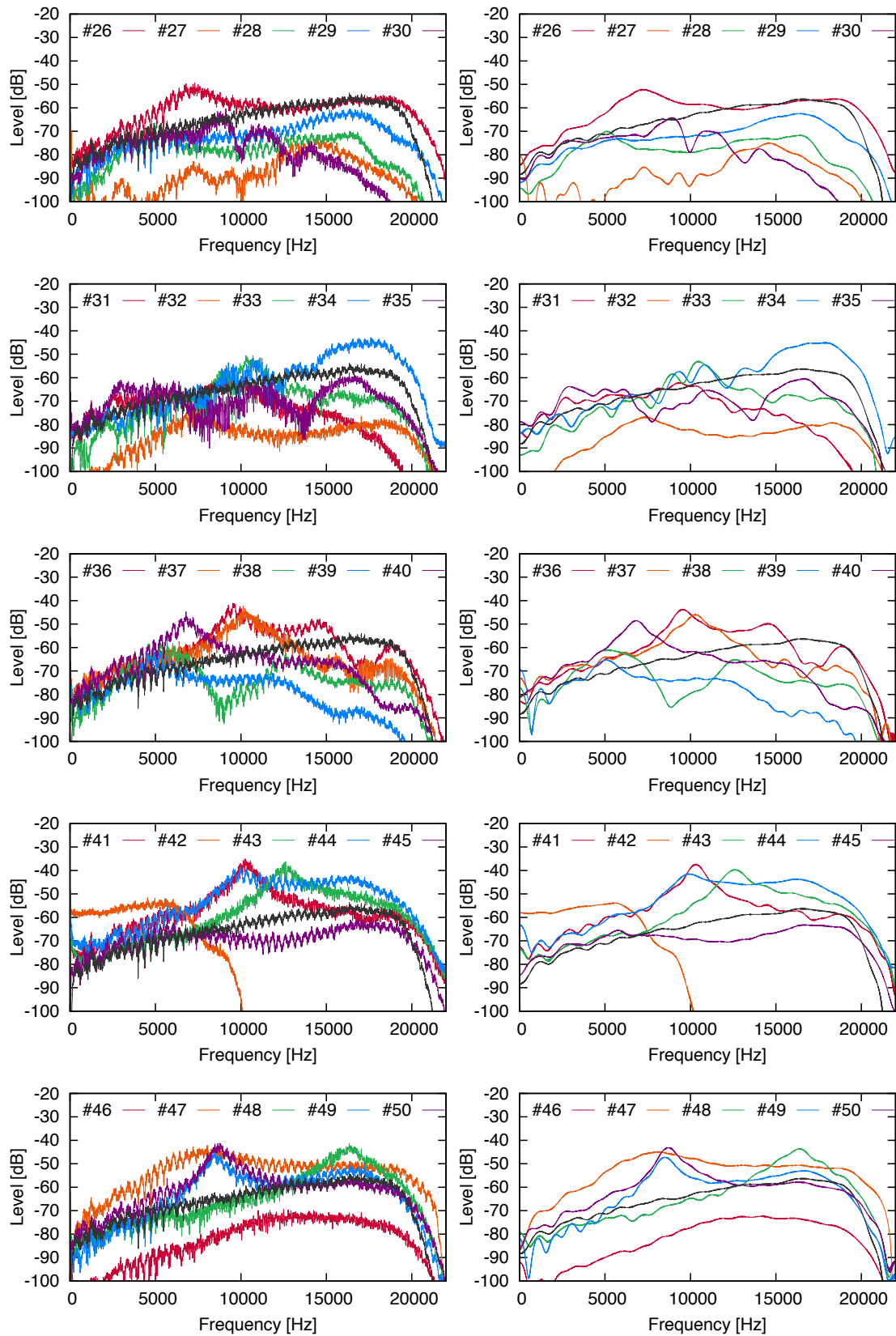


図 4.22: iPhone 5s と 100 機種 の Android 端末 で 録音 し た ホワイ ト ノイズ の スペクトル 分 布 と 包 絡 線 ( 端 末 番 号 #26 ~ #50 )



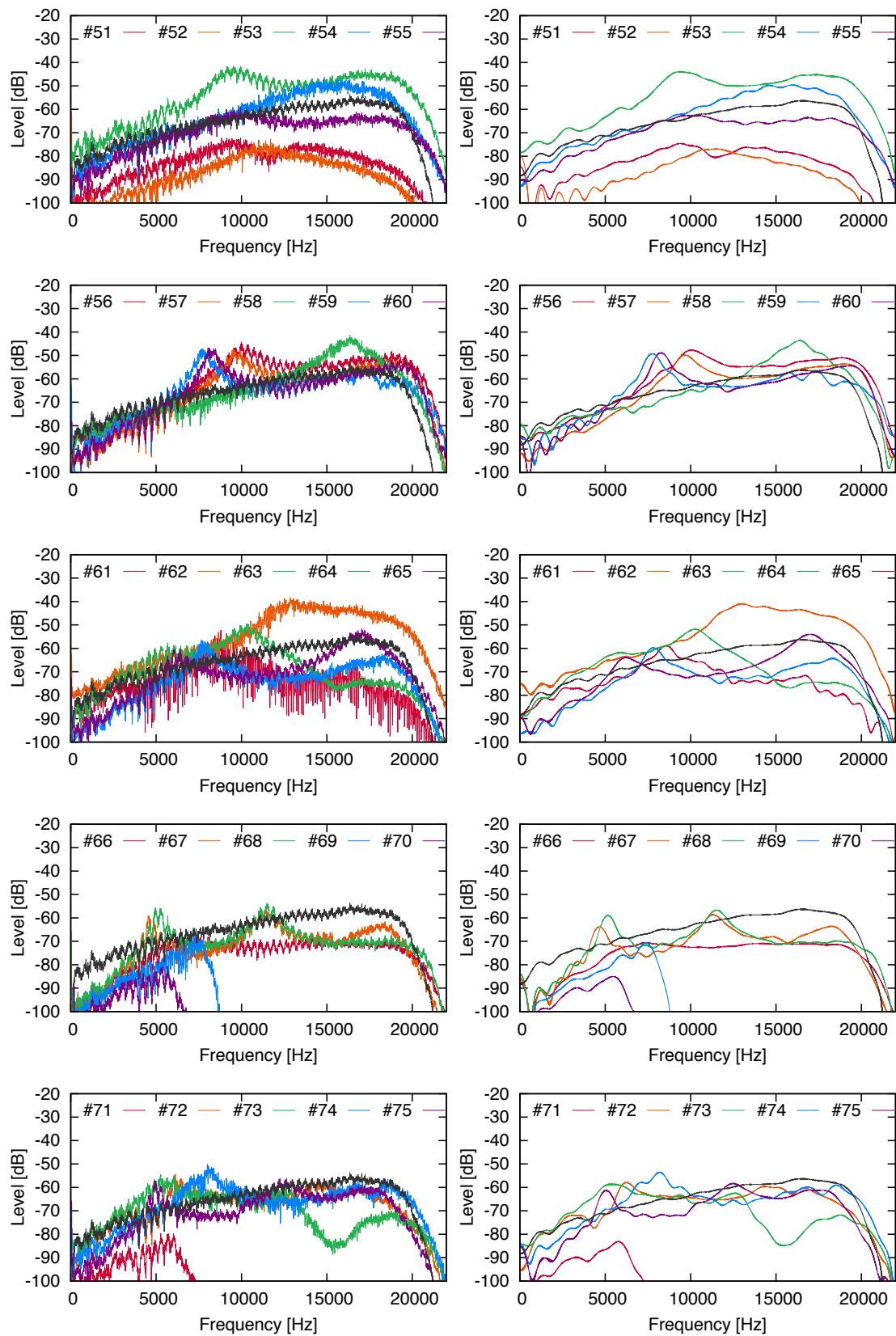


図 4.23: iPhone 5s と Android 端末 100 機種で録音したホワイトノイズのスペクトル分布と包絡線（端末番号#51～#75）

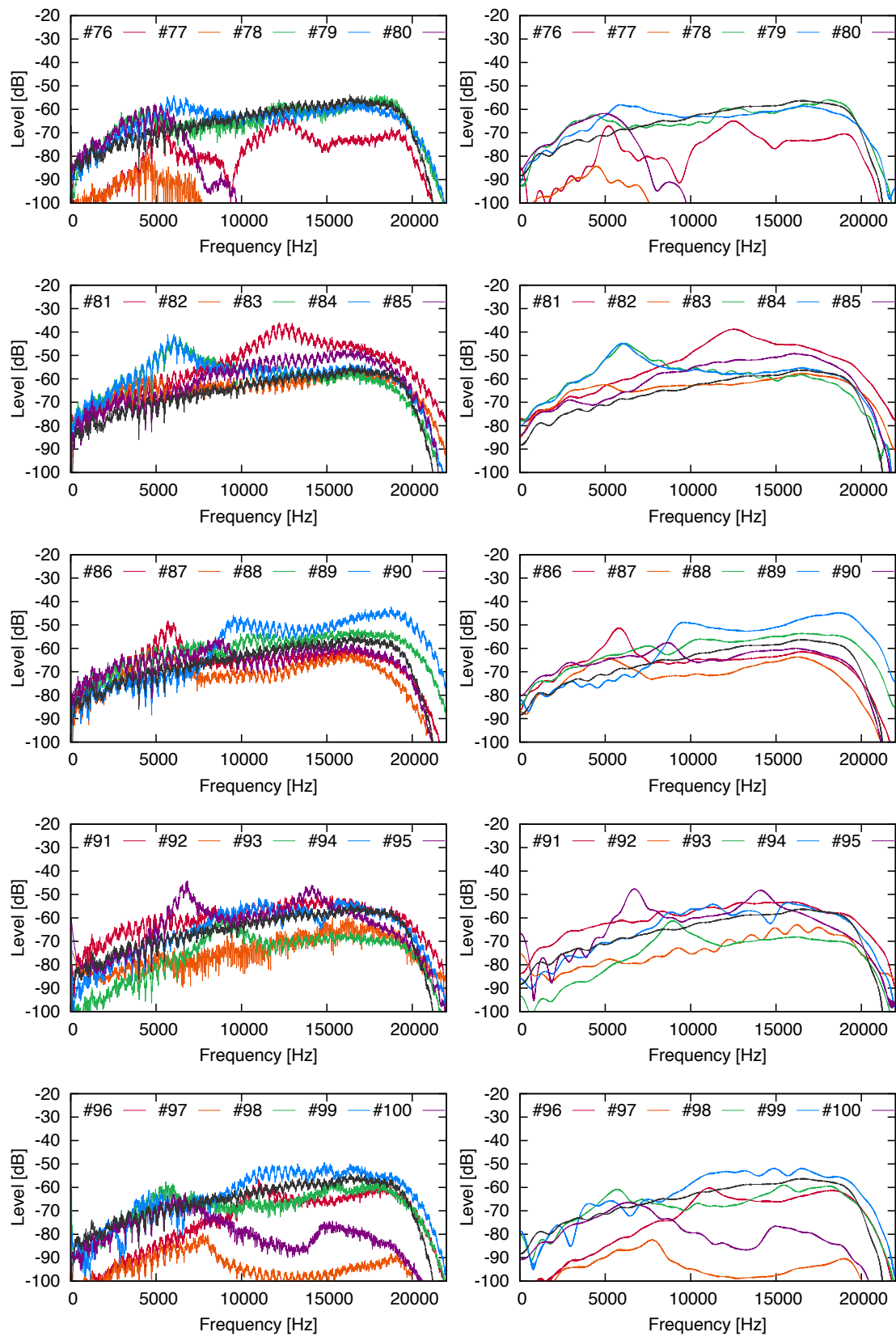


図 4.24: iPhone 5s と Android 端末 100 機種で録音したホワイトノイズのスペクトル分布と包絡線（端末番号#76～#100）



まず、これまでの録音実験に用いてきた iPhone 5s の周波数特性は均一でなく、17.5 kHz 以下の領域では周波数が高いほど感度が高いことが判明した。また、19 kHz を越えると急速に感度が低下する。なお、50 回の計測においてスペクトル分布に目立った違いは見られなかったため、図には 50 回の平均値をプロットした。

次に、Android 端末については機種によって大きな差異が見られる。Nexus 5 (#06) や Nexus 6 (#07) のように iPhone 5s と類似した周波数特性を示す端末もあれば、ARROWS Me F-11D (#18) や G Flex LGL23 (#50) のように特定の周波数帯に感度の鋭いピークを持つ端末もある。また、製造メーカーが同じで発売時期の近い端末同士では類似した周波数特性が見られる例が少なくない。具体的には、ARROWS Kiss F-03E (#20) と ARROWS ef FJL21 (#22), isai LGL22 (#49) と G Flex LGL23 (#50), MEDIAS ES N-05D (#56) と MEDIAS X N-07D (#57), Disney Mobile N-03E (#59) と MEDIAS X N-04E (#60), Xperia AX SO-01E (#67) と Xperia VL SOL21 (#75), GALAXY S4 SC-04E (#84) と GALAXY J SC-02F (#86) が該当する。これらの端末は共通のベースモデルが存在するために、部品や筐体構造が類似しているものと考えられる。一方で、製造メーカーが同じでも発売時期の異なる端末同士では異なる周波数特性が見られる。

なお、Optimus Vu L-06D (#42), Xperia Z1 SO-01F (#69), Xperia Z1 f SO-02F (#70), Xperia A2 SO-04F (#71), Xperia Z Ultra SOL24 (#77), GALAXY S2 LTE SC-03D (#80) の 6 機種については高周波領域のマイク感度が著しく低く、十分な録音音量を確保できなかった。このうち、Xperia シリーズの 4 機種については、単調な入力が続いた場合にノイズキャンセリング機能が作動することが知られている。

さらに、全周波数帯域での平均音量が iPhone 5s と等しくなるようゲイン調整を行い、iPhone 5s のスペクトル分布との差分を取ったものを図 4.25, 図 4.26 に示す。100 Hz 未満の低周波領域と 20 kHz 以上の高周波領域では大きな乖離が見られるものの、100 Hz ~ 20 kHz の範囲では概ね 10 dB 以内の誤差に収まっている機種が多い。100 Hz ~ 20 kHz の区間における平均絶対誤差は、47 機種で 5 dB 未満、40 機種で 5 dB 以上 10 dB 未満、7 機種で 10 dB 以上 15 dB 未満、6 機種で 15 dB 以上であった。

以上より、スマートフォンの機種による内蔵マイクの周波数特性の違いは無視できる水準とは言えず、スマートフォンセンシングにおいては周波数特性の違いに対してロバストな設計や実装が求められる。

#### 4.4.4 センシング精度への影響評価

次に、前節で紹介した周波数特性の違いが我々の提案するマルチコプタの積載重量推定手法の性能に与える影響を評価する。ここでは、iPhone 5s と各スマートフォンの周波数特性の違いを、判定ロジックの入力に適用した状態での性能評価を行う。具体的には、積載重量の推定フローにおいて FFT により算出したスペクトル分布に対し、図 4.25, 図 4.26 に示したスペクトル分布の差分を適用した上で、k-NN 分類器に入力する。なお、分類器の構築においては、周波数特性の違いを考慮せず iPhone 5s での計測データをそのまま用いた。すなわち、純粋に iPhone 5s の録音データで構築した分類器に対して、100 機種の Android 端末での録音データに相当する擬似的な入力を行い、その判定精度を評価する。

図 4.27 および表 4.17 に Android 端末 100 機種分の擬似的な入力を行った際の積載重量

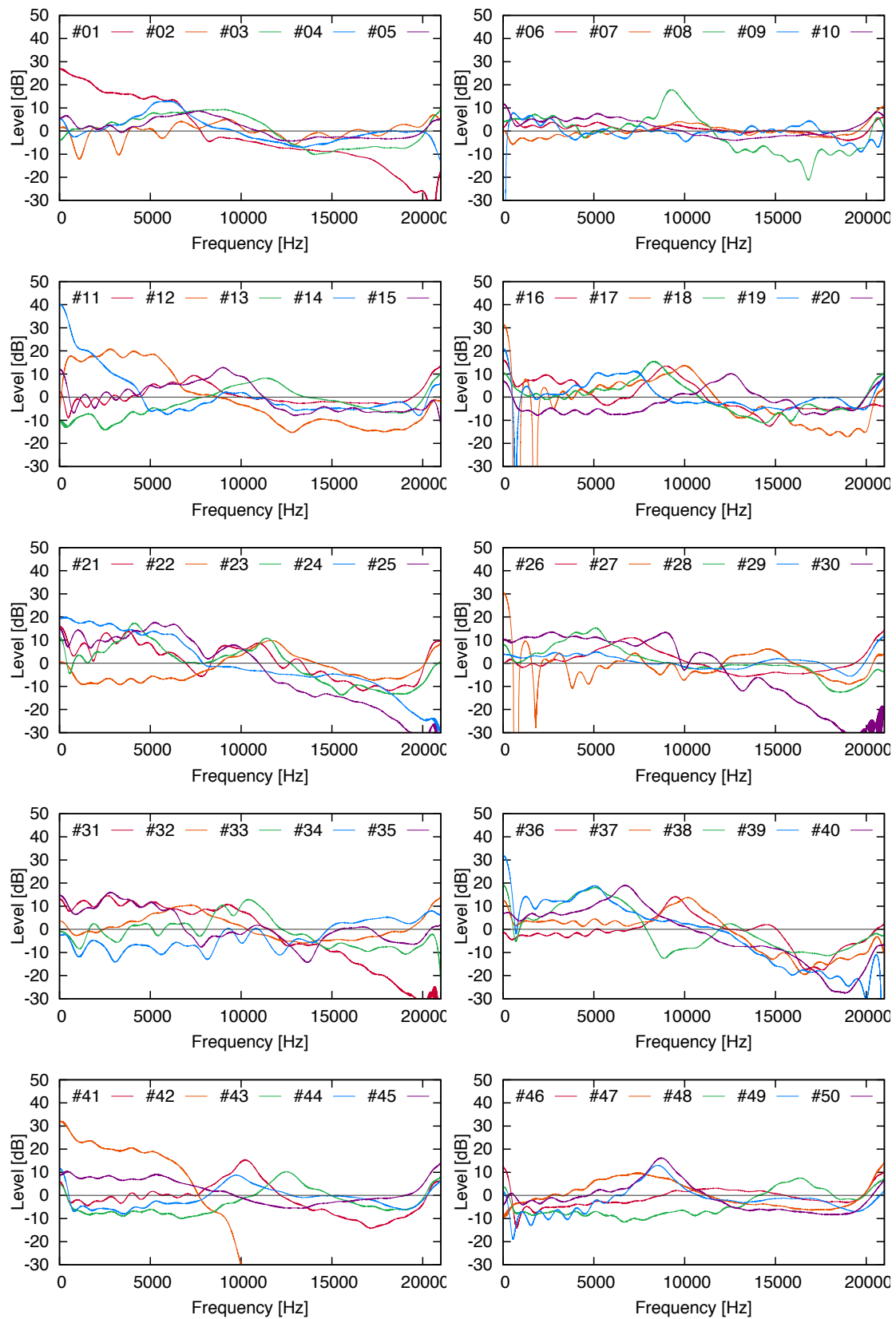


図 4.25: ゲイン調整後のスペクトル分布の iPhone 5s との差分 (端末番号#01~#50)

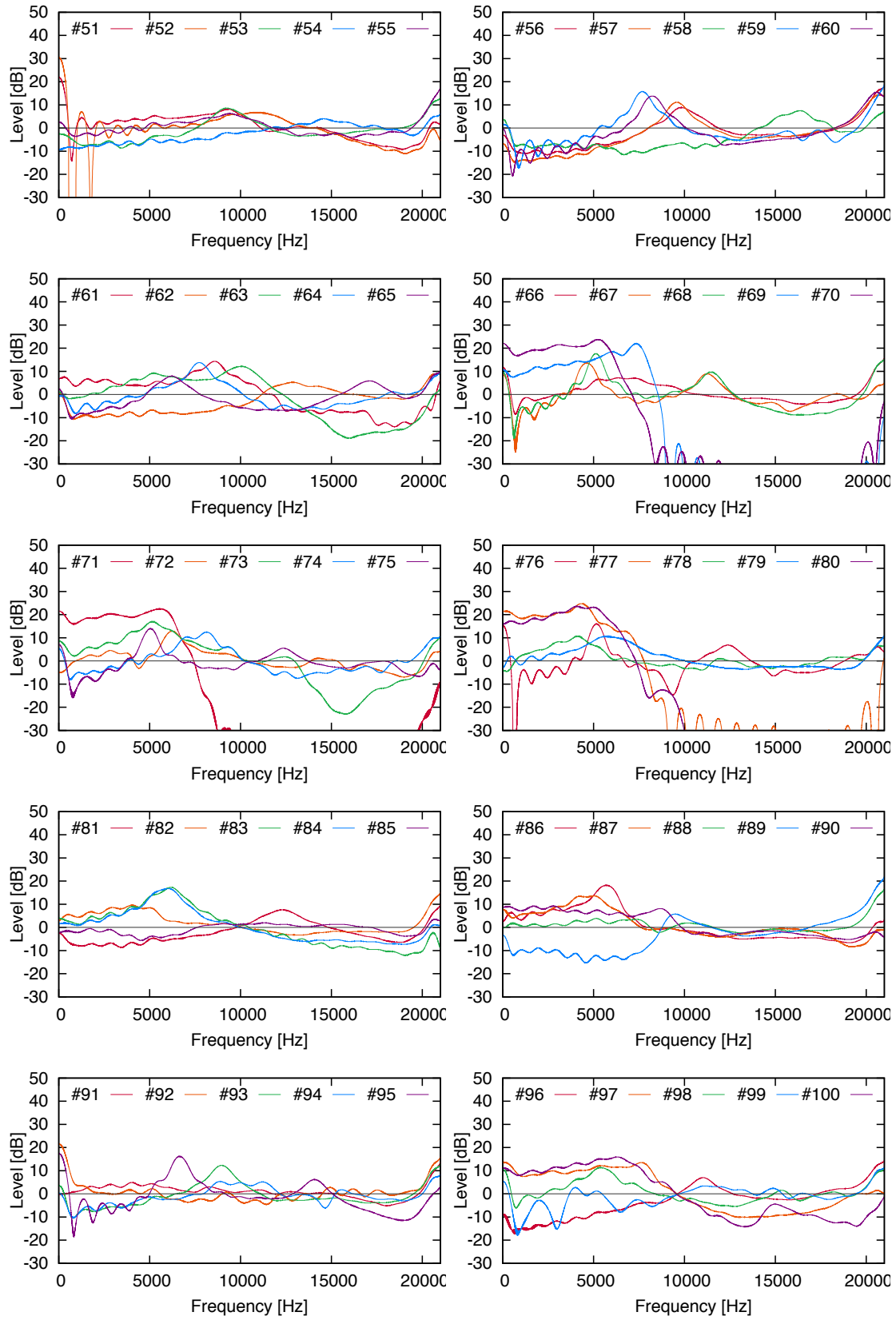


図 4.26: ゲイン調整後のスペクトル分布の iPhone 5s との差分 (端末番号#51~#100)

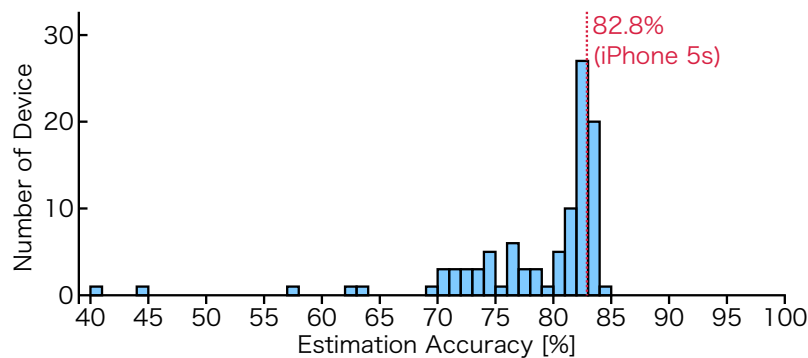


図 4.27: 各スマートフォンの録音データに相当する擬似的な入力でのマルチコプタ積載重量の推定精度の度数分布

の判定精度を示す。判定対象は、4.3.4 節での評価と同じく、積載重量 0 g, 100 g, 300 g, 500 g の 4 種類である。100 機種中 86 機種では判定精度の低下を 10% 未満に抑えられており、うち 63 機種では判定精度が 80% に達するなど、iPhone 5s での録音時と遜色ない性能を期待できる。100 機種での平均判定精度は 78.6% であり、iPhone 5s と比べての精度低下は 4.2% であった。一方で、ARROWS X F-10D (#17)、HTC J ISW13HT (#27)、isai VL LGV31 (#52) の 3 機種では、判定精度が 60% を下回る結果となった。これらの 3 機種のスペクトル分布を確認すると、低周波領域においてスペクトル包絡線が大きく落ち込む部分が見受けられる。これによりマルチコプタの飛行音に含まれる周期的な周波数ピークをうまく捉えることができなかつたと推測される。

#### 4.4.5 まとめ

本節では、スマートフォンの機種により異なる内蔵マイクの周波数特性について、100 機種 of Android 端末での録音データを基に分析し、センシング性能への影響を評価した。感度特性のみならず周波数特性についても無視できるとは言い難い水準の差異が見られたため、スマートフォンセンシングの設計や実装においては、周波数特性の違いを考慮しロバスト性を持たせる必要がある。一方、4.3 節で提案したマルチコプタの積載重量の推定手法においては、積載重量の判定精度に平均で 4.2% の影響が見られたものの、ほとんどの機種においては致命的な影響を及ぼすものではなく、我々のセンシング手法は様々なスマートフォン上で利用可能であることが確認された。

表 4.17: 各スマートフォンの録音データに相当する擬似的な入力でのマルチコプタ積載重量の推定精度と iPhone 5s との差分

#	機種名	精度	差分	#	機種名	精度	差分
01	Priori 3 LTE	82.8%	-0.1%	51	isai FL	72.9%	-9.9%
02	HUAWEI P9 lite	82.4%	-0.4%	52	isai VL	44.8%	-38.0%
03	Nexus S	82.8%	0.0%	53	LG G2 mini	78.6%	-4.2%
04	GALAXY Nexus	83.5%	0.6%	54	Motorola RAZR	81.8%	-1.1%
05	Nexus 4	82.0%	-0.8%	55	Motorola RAZR M	76.4%	-6.5%
06	Nexus 5	82.0%	-0.8%	56	MEDIAS ES	73.3%	-9.5%
07	Nexus 6	82.4%	-0.4%	57	MEDIAS X	76.5%	-6.4%
08	Nexus 7 (2012)	82.8%	-0.1%	58	MEDIAS U	81.7%	-1.1%
09	Nexus 7 (2013)	81.5%	-1.3%	59	Disney Mobile	70.5%	-12.3%
10	Nexus 9	83.4%	0.6%	60	MEDIAS X	72.2%	-10.7%
11	ZenFone 2 Laser	82.2%	-0.6%	61	MEDIAS W	83.1%	0.2%
12	ZenFone Go	81.6%	-1.3%	62	VEGA	74.5%	-8.3%
13	BlackBerry PRIV	81.5%	-1.3%	63	Xperia NX	82.8%	-0.1%
14	G'zOne TYPE-L	80.6%	-2.3%	64	Xperia acroHD	79.2%	-3.7%
15	ARROWS X LTE	81.3%	-1.6%	65	Xperia GX	74.6%	-8.3%
16	Disney Mobile	83.3%	0.5%	66	Xperia SX	71.9%	-10.9%
17	ARROWS X	40.6%	-42.2%	67	Xperia AX	72.1%	-10.7%
18	ARROWS Me	76.8%	-6.1%	68	Xperia A	70.3%	-12.6%
19	ARROWS X	63.8%	-19.0%	69	Xperia Z1	83.0%	0.1%
20	ARROWS Kiss	77.6%	-5.2%	70	Xperia Z1 f	83.3%	0.4%
21	ARROWS Z	82.2%	-0.7%	71	Xperia A2	83.3%	0.4%
22	ARROWS ef	76.9%	-6.0%	72	Xperia Z3 Compact	82.9%	0.1%
23	ARROWS Z	84.2%	1.4%	73	Xperia X Compact	82.5%	-0.4%
24	SAMURAI MIYABI	82.9%	0.1%	74	Xperia acroHD	76.6%	-6.3%
25	MUSASHI	83.1%	0.3%	75	Xperia VL	75.6%	-7.2%
26	EVO 3D	80.9%	-1.9%	76	Xperia UL	62.2%	-20.6%
27	HTC J	57.2%	-25.6%	77	Xperia Z Ultra	82.8%	0.0%
28	HTC J butterfly	83.3%	0.5%	78	Xperia Z3	82.4%	-0.4%
29	HTC J One	82.0%	-0.8%	79	GALAXY S2	82.9%	0.0%
30	HTC J butterfly	83.0%	0.2%	80	GALAXY S2 LTE	83.1%	0.2%
31	HTC J butterfly	83.7%	0.8%	81	GALAXY Note	76.4%	-6.4%
32	Ascend	80.2%	-2.7%	82	GALAXY SIII	82.5%	-0.4%
33	honor6 Plus	83.7%	0.8%	83	GALAXY Note2	82.9%	0.1%
34	honor8	78.7%	-4.2%	84	GALAXY S4	83.1%	0.2%
35	URBANO PROG...	82.9%	0.1%	85	GALAXY Note3	82.8%	0.0%
36	DIGNO S	83.3%	0.5%	86	GALAXY J	83.0%	0.2%
37	URBANO L01	83.0%	0.2%	87	GALAXY SII WiM...	83.5%	0.7%
38	INFOBAR A03	81.2%	-1.6%	88	GALAXY SIII Progre	81.8%	-1.0%
39	TORQUE G01	78.4%	-4.4%	89	GALAXY S5	69.0%	-13.8%
40	TORQUE	83.0%	0.2%	90	Galaxy S6 edge	83.0%	0.1%
41	Optimus it	80.2%	-2.6%	91	Galaxy S8	82.7%	-0.2%
42	Optimus Vu	82.3%	-0.5%	92	AQUOS PHONE	71.7%	-11.2%
43	Optimus LIFE	74.3%	-8.5%	93	DRAGON QUEST	74.6%	-8.3%
44	Optimus it	74.0%	-8.9%	94	AQUOS PHONE SL	77.6%	-5.2%
45	G2	82.3%	-0.5%	95	AQUOS PHONE CL	73.1%	-9.8%
46	Disney Mobile	71.0%	-11.8%	96	AQUOS PHONE S...	74.6%	-8.3%
47	Optimus X	82.4%	-0.5%	97	AQUOS SERIE	83.3%	0.4%
48	Optimus G	80.3%	-2.5%	98	Disney Mobile	77.5%	-5.4%
49	isai	82.4%	-0.5%	99	YotaPhone 2	71.0%	-11.8%
50	G Flex	82.0%	-0.8%	100	Model RETRO	83.3%	0.4%

## 4.5 あとがき

本章では、高密度リチウムイオン電池やMEMS ジャイロセンサといった近年の技術革新により実用化され、歩行者の新たな脅威となりつつある電気自動車とマルチコプタについて、これらの機体が発する音に着目した接近検知手法および過積載検知手法を提案した。本手法では、機械学習を用いることで環境雑音の違いに対するロバスト性を確保しつつ、音声データを追加することで新機種に対応可能とした。また本手法は、我々が日常生活で持ち歩くスマートフォン上で実行可能なほど軽量であり、インフラとして特別な設備を必要としないため、ユーザが自衛のために今すぐ利用できる点が特徴的である。

本章で提案した技術は、これから急速な増加が予想されるモータ駆動方式のモビリティ装置に対して応用できる可能性が十分にあり、我々の技術は次世代のモビリティ装置と安全に共存するための基礎を切り拓いたと言える。

## ■ 第5章

---

結論

## 5.1 本論文のまとめ

本論文では、中古スマートフォンを活用して低コストで高機能かつ環境負荷の低いセンシング活動を実現するため、以下の3つの技術レイヤで提案を行った。

### 1. ハードウェア

まず、中古スマートフォンに「第二の人生」を与え、センサノードとして活用するための技術として、遠隔操作でのスマートフォンの電源投入を実現するUSB Dongle型デバイスを提案した。ここでは、精密農業などの用途で必ずしも常時のセンシングが必要でないことに着目し、スマートフォンを間欠動作させることで商用電源のない環境でも長期間のセンシングを可能にした。この手法では、電源オフ状態のAndroid端末のUSB端子に所定の電圧を印加すると電源状態管理ソフトウェアが起動することに着目し、マイクロ波無線給電技術を用いて必要な電力を供給する。これにより、家電のリモコンのように遠隔操作でのスマートフォンの電源投入を可能とし、待機電力をゼロに抑えて数ヶ月単位での電池駆動を実現した。

### 2. OS・プラットフォーム

次に、Android端末を間欠動作させる際の時間的・エネルギー的なオーバーヘッドを削減するため、Androidデバイスに搭載されたLinuxカーネルを活用してセンシングを行う手法を提案した。本手法では、温度センサやタッチパネルの静電容量センサだけでなく、BluetoothやUSB通信をサポートし、マイコン等の外部機器との連携も可能とした。また、本手法を適用した場合でもAndroid OSの機能は完全に温存でき、必要に応じてLinuxカーネルとAndroid OSを選択的に起動できる。

### 3. アプリケーション

さらに、音に着目したセンシング技術のアプリケーションを2つ紹介した。

1つ目は、交通事故から歩行者を守るための接近車両検知技術である。ここでは、走行時の高い静粛性ゆえに事故が多発している電気自動車およびハイブリッド車に焦点を当て、走行用モータが発するスイッチング雑音を機械学習で検出することで、環境雑音や車種の違いにロバストな車両検知を実現した。

2つ目は、マルチコプタの積載重量を飛行音から推定する技術である。マルチコプタは、急速に低価格化が進み幅広い用途での活躍が期待されるが、我々の生活空間を縦横無尽に飛び回るため、周囲の人と衝突する危険性がある。そこで、ロータの回転に起因する飛行音から機械学習で積載重量を推定し、積載重量が多く危険性の高いマルチコプタを選択的に検知することを可能とした。

いずれの手法も、インフラとして一切特別な設備を必要とせず、ユーザが所持するスマートフォンだけで処理が完結するため、ユーザが自衛のために必要と判断すれば今すぐにでも利用できる点が特徴的である。ただし、これらのセンシング技術は少なからずスマートフォンの消費電力を増加させるため、センシング専用の中古スマートフォンとユーザの手元のスマートフォンで役割分担を行うことが望ましい。

本論文で提案したこれらの技術は、低コストかつ高性能なセンシングを実現し、個人や零細企業の草の根的なセンシング活動を支援することに資するものである。



## 5.2 今後の検討課題

本研究において、今後検討すべき課題としては以下のものが挙げられる。

### 5.2.1 スマートフォンの機種依存

中古スマートフォンの活用にあたり、ハードウェア、OS・プラットフォーム、アプリケーションのいずれのレイヤにおいても、機種依存への対策は最も重要な検討課題である。各章で既に述べたように、原理原則に基づいた機種依存に対する分析は実施済みであるが、多種多様なハードウェアを持つ Android 端末の世界には、原理原則で説明できないような特殊な機種も存在する。これらの機種 1 つ 1 つについて検証を行うことは容易でないが、実用化の段階においては市場シェア上位の機種について重点的に互換性の検証を行う必要がある。

### 5.2.2 中古スマートフォンならではのアプリケーション開拓

本研究では、スマートフォンセンシングの概念を大きく拡大し、中古スマートフォンに専用センサノードとしての「第二の人生」を与えるための基礎的な技術を開発し、アプリケーションまで一貫した提案を行ってきた。しかしながら、依然として“スマートフォンセンシング＝現役スマートフォンの仕事”という固定観念は拭いきれておらず、中古スマートフォンならではのアプリケーションにはまだまだ開拓の余地がある。より幅広い分野で中古スマートフォンを活用できるよう、アプリケーションの拡大に努めていきたい。

## 謝辞

---

本研究の遂行ならびに博士論文の執筆にあたり，学部4年次から博士課程修了までの6年間に渡って，ご多忙な時期にも手厚くご指導を頂いた指導教員の川原圭博准教授に深く感謝いたします。また，昨年ご退官されるまで5年間に渡り，常に熱心にアドバイスを下さった浅見徹名誉教授に，心より感謝申し上げます。博論審査において幅広い視点から貴重なご指摘を頂きました，瀬崎薫教授，田浦健次朗教授，落合秀也准教授にも感謝を申し上げます。最後に，日々の研究生活でお世話になりました川原研究室の皆様，博士課程への進学を理解してくれた家族にもお礼申し上げます。

## 参考文献

---

- [1] 総務省, “平成 23 年版 情報通信白書 情報通信産業の動向 第 2 節,” <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h23/pdf/n4020000.pdf>, 2011. [Online; accessed 15-Sept.-2017].
- [2] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M.B. Srivastava, “Participatory sensing,” In: Workshop on World-Sensor-Web (WSW ’06): Mobile Device Centric Sensor Networks and Applications, 2006.
- [3] S. Kang, Y. Lee, C. Min, Y. Ju, T. Park, J. Lee, Y. Rhee, and J. Song, “Orchestrator: An active resource orchestration framework for mobile context monitoring in sensor-rich mobile environments,” 2010 IEEE International Conference on Pervasive Computing and Communications (PerCom), pp.135–144, March 2010.
- [4] H. Shen, A. Balasubramanian, A. LaMarca, and D. Wetherall, “Enhancing mobile apps to use sensor hubs without programmer effort,” Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp.227–238, UbiComp ’15, ACM, New York, NY, USA, 2015. <http://doi.acm.org/10.1145/2750858.2804260>
- [5] G. Metri, W. Shi, M. Brockmeyer, and A. Agrawal, “Batteryextender: An adaptive user-guided tool for power management of mobile devices,” Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp.33–43, UbiComp ’14, ACM, New York, NY, USA, 2014. <http://doi.acm.org/10.1145/2632048.2632082>
- [6] H. Lu, J. Yang, Z. Liu, N.D. Lane, T. Choudhury, and A.T. Campbell, “The jigsaw continuous sensing engine for mobile phone applications,” Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, pp.71–84, SenSys ’10, ACM, New York, NY, USA, 2010. <http://doi.acm.org/10.1145/1869983.1869992>
- [7] M.-R. Ra, B. Liu, T.F. La Porta, and R. Govindan, “Medusa: A programming framework for crowd-sensing applications,” Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, pp.337–350, MobiSys ’12, ACM, New York, NY, USA, 2012. <http://doi.acm.org/10.1145/2307636.2307668>
- [8] N.D. Lane, Y. Chon, L. Zhou, Y. Zhang, F. Li, D. Kim, G. Ding, F. Zhao, and H. Cha, “Piggyback crowdsensing (pcs): Energy efficient crowdsourcing of mobile

- sensor data by exploiting smartphone app opportunities,” Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, pp.7:1–7:14, SenSys ’13, ACM, New York, NY, USA, 2013. <http://doi.acm.org/10.1145/2517351.2517372>
- [9] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, “Anonymsense: Privacy-aware people-centric sensing,” Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services, pp.211–224, MobiSys ’08, ACM, New York, NY, USA, 2008. <http://doi.acm.org/10.1145/1378600.1378624>
- [10] M.M. Groat, B. Edwards, J. Horey, W. He, and S. Forrest, “Enhancing privacy in participatory sensing applications with multidimensional data,” 2012 IEEE International Conference on Pervasive Computing and Communications, pp.144–152, March 2012.
- [11] C. Meng, W. Jiang, Y. Li, J. Gao, L. Su, H. Ding, and Y. Cheng, “Truth discovery on crowd sensing of correlated entities,” Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, pp.169–182, SenSys ’15, ACM, New York, NY, USA, 2015. <http://doi.acm.org/10.1145/2809695.2809715>
- [12] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren, “Cloud-enabled privacy-preserving truth discovery in crowd sensing systems,” Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, pp.183–196, SenSys ’15, ACM, New York, NY, USA, 2015. <http://doi.acm.org/10.1145/2809695.2809719>
- [13] J.-S. Lee and B. Hoh, “Dynamic pricing incentive for participatory sensing,” vol.6, no.6, Pervasive Mob. Comput., Dec. 2010. <http://dx.doi.org/10.1016/j.pmcj.2010.08.006>
- [14] L.G. Jaimes, I. Vergara-Laurens, and M.A. Labrador, “A location-based incentive mechanism for participatory sensing systems with budget constraints,” 2012 IEEE International Conference on Pervasive Computing and Communications, pp.103–108, March 2012.
- [15] L. Tianze, W. Muqing, and W. Yuewei, “An game theory based online incentive mechanism for participatory sensing,” 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp.700–705, Aug. 2016.
- [16] A. Carroll and G. Heiser, “An analysis of power consumption in a smartphone,” Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference, pp.21–21, USENIXATC’10, USENIX Association, Berkeley, CA, USA, 2010. <http://dl.acm.org/citation.cfm?id=1855840.1855861>
- [17] Y. Kataoka, H. Imai, Y. Nakata, T. Daitoh, T.M.N. Kimura, T. Nakano, Y. Mizuno, T. Oketani, M. Takahashi, M. Tsubuku, H. Miyake, T.I.Y. Hirakata, J. Koyama, S.

- Yamazaki, J. Koezuka, and K. Okazaki, “56.1: Development of igzo-tft and creation of new devices using igzo-tft,” vol.44, no.1, SID Symposium Digest of Technical Papers, 2013. <http://dx.doi.org/10.1002/j.2168-0159.2013.tb06329.x>
- [18] J. Huang, F. Qian, Z.M. Mao, S. Sen, and O. Spatscheck, “Screen-off traffic characterization and optimization in 3g/4g networks,” Proceedings of the 2012 Internet Measurement Conference, pp.357–364, IMC '12, ACM, New York, NY, USA, 2012. <http://doi.acm.org/10.1145/2398776.2398813>
- [19] F. Xu, Y. Liu, T. Moscibroda, R. Chandra, L. Jin, Y. Zhang, and Q. Li, “Optimizing background email sync on smartphones,” Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services, pp.55–68, MobiSys '13, ACM, New York, NY, USA, 2013. <http://doi.acm.org/10.1145/2462456.2464444>
- [20] A. Chakraborty, V. Navda, V.N. Padmanabhan, and R. Ramjee, “Coordinating cellular background transfers using loadsense,” Proceedings of the 19th Annual International Conference on Mobile Computing & Networking, pp.63–74, MobiCom '13, ACM, New York, NY, USA, 2013. <http://doi.acm.org/10.1145/2500423.2500447>
- [21] F. Qian, Z. Wang, A. Gerber, Z.M. Mao, S. Sen, and O. Spatscheck, “Top: Tail optimization protocol for cellular radio resource allocation,” The 18th IEEE International Conference on Network Protocols, pp.285–294, Oct. 2010.
- [22] F. Qian, Z. Wang, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck, “Profiling resource usage for mobile applications: A cross-layer approach,” Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, pp.321–334, MobiSys '11, ACM, New York, NY, USA, 2011. <http://doi.acm.org/10.1145/1999995.2000026>
- [23] P.K. Athivarapu, R. Bhagwan, S. Guha, V. Navda, R. Ramjee, D. Arora, V.N. Padmanabhan, and G. Varghese, “Radiojockey: Mining program execution to optimize cellular radio usage,” Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, pp.101–112, Mobicom '12, ACM, New York, NY, USA, 2012. <http://doi.acm.org/10.1145/2348543.2348559>
- [24] H.S. Ramos, T. Zhang, J. Liu, N.B. Priyantha, and A. Kansal, “Leap: A low energy assisted gps for trajectory-based services,” Proceedings of the 13th International Conference on Ubiquitous Computing, pp.335–344, UbiComp '11, ACM, New York, NY, USA, 2011. <http://doi.acm.org/10.1145/2030112.2030158>
- [25] P.K. Misra, W. Hu, Y. Jin, J. Liu, A. Souza dePaula, N. Wirstrom, and T. Voigt, “Energy efficient gps acquisition with sparse-gps,” Proceedings of

- the 13th International Symposium on Information Processing in Sensor Networks, pp.155–166, IPSN '14, IEEE Press, Piscataway, NJ, USA, 2014. <http://dl.acm.org/citation.cfm?id=2602339.2602357>
- [26] J. Paek, K.-H. Kim, J.P. Singh, and R. Govindan, “Energy-efficient positioning for smartphones using cell-id sequence matching,” Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, pp.293–306, MobiSys '11, ACM, New York, NY, USA, 2011. <http://doi.acm.org/10.1145/1999995.2000024>
- [27] Google Inc., “140425 [参考資料 6-02] 補足説明資料,” [http://www.soumu.go.jp/main\\_content/000287492.pdf](http://www.soumu.go.jp/main_content/000287492.pdf), 2013. [Online; accessed 15-Sept.-2017].
- [28] J.L. Hill and D.E. Culler, “Mica: a wireless platform for deeply embedded networks,” vol.22, no.6, IEEE Micro, Nov. 2002.
- [29] J. Polastre, R. Szewczyk, and D. Culler, “Telos: Enabling ultra-low power wireless research,” Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, IPSN '05, IEEE Press, Piscataway, NJ, USA, 2005. <http://dl.acm.org/citation.cfm?id=1147685.1147744>
- [30] P. Volgyesi, G. Balogh, A. Nadas, C.B. Nash, and A. Ledeczi, “Shooter localization and weapon classification with soldier-wearable networked sensors,” Proceedings of the 5th International Conference on Mobile Systems, Applications and Services, pp.113–126, MobiSys '07, ACM, New York, NY, USA, 2007. <http://doi.acm.org/10.1145/1247660.1247676>
- [31] H. Shen, A. Balasubramanian, A. LaMarca, and D. Wetherall, “Enhancing mobile apps to use sensor hubs without programmer effort,” Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp.227–238, UbiComp '15, ACM, New York, NY, USA, 2015. <http://doi.acm.org/10.1145/2750858.2804260>
- [32] F. Guerrache, A. Aldabbagh, and E. Kanjo, “Multiple sensor fusion approach to map environmental noise impact on health,” Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, pp.1074–1078, UbiComp '16, ACM, New York, NY, USA, 2016. <http://doi.acm.org/10.1145/2968219.2972710>
- [33] Powercast Corp., “Powercast: P2110B Datasheet,” <http://www.powercastco.com/wp-content/uploads/2016/12/P2110B-Datasheet-Rev-3.pdf>, 2017. [Online; accessed 14-Feb-2017].
- [34] VITRO, “8pino: The minimalist’s Arduino compatible,” <http://8pino.cc/>, 2017. [Online; accessed 14-Feb-2017].

- [35] W. Wang, A.X. Liu, M. Shahzad, K. Ling, and S. Lu, “Understanding and modeling of wifi signal based human activity recognition,” Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, pp.65–76, MobiCom ’15, ACM, New York, NY, USA, 2015. <http://doi.acm.org/10.1145/2789168.2790093>
- [36] Y. Kawahara, H. Lee, and M.M. Tentzeris, “Sensprout: Inkjet-printed soil moisture and leaf wetness sensor,” Proceedings of the 2012 ACM Conference on Ubiquitous Computing, pp.545–545, UbiComp ’12, ACM, New York, NY, USA, 2012. <http://doi.acm.org/10.1145/2370216.2370302>
- [37] Google Inc., “Android Open Source Project,” <https://source.android.com/>, 2017. [Online; accessed 15-Jan-2017].
- [38] CyanogenMod, “CyanogenMod — Android Community Operating System,” <https://www.cyanogenmod.org/>, 2016. [Online; accessed 15-Jan-2017].
- [39] Xiaomi, “Xiaomi MIUI Official Global Site,” <http://en.miui.com/>, 2017. [Online; accessed 15-Jan-2017].
- [40] antutu, “Chip Brand Distribution and Market Shares 2016,” <http://www.antutu.com/doc/107751.htm>, 2017. [Online; accessed 29-March-2017].
- [41] T. Micro, “STUXNET Malware Targets SCADA Systems,” <https://www.trendmicro.com/vinfo/us/threat-encyclopedia/web-attack/54/stuxnet-malware-targets-scada-systems>, 2010. [Online; accessed 26-Oct-2017].
- [42] T. Micro, “BlueBorne: Bluetooth Vulnerabilities Expose Billions of Devices to Hacking,” <https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/blueborne-bluetooth-vulnerabilities-expose-billions-of-devices-to-hacking>, 2017. [Online; accessed 26-Oct-2017].
- [43] H. Han, J. Yu, H. Zhu, Y. Chen, J. Yang, G. Xue, Y. Zhu, and M. Li, “E3: Energy-efficient engine for frame rate adaptation on smartphones,” Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, pp.15:1–15:14, SenSys ’13, ACM, New York, NY, USA, 2013. <http://doi.acm.org/10.1145/2517351.2517364>
- [44] D.T. Nguyen, G. Zhou, X. Qi, G. Peng, J. Zhao, T. Nguyen, and D. Le, “Storage-aware smartphone energy savings,” Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp.677–686, UbiComp ’13, ACM, New York, NY, USA, 2013. <http://doi.acm.org/10.1145/2493432.2493505>
- [45] M.-R. Ra, B. Priyantha, A. Kansal, and J. Liu, “Improving energy efficiency of personal sensing applications with heterogeneous multi-processors,” Proceedings of

- the 2012 ACM Conference on Ubiquitous Computing, pp.1–10, UbiComp '12, ACM, New York, NY, USA, 2012. <http://doi.acm.org/10.1145/2370216.2370218>
- [46] “BlueZ: Official Linux Bluetooth protocol stack,” 2014. BlueZ Project, <http://www.bluez.org/>.
- [47] “Flash tool for Xperia devices,” 2017. Flash tool for Xperia devices, <https://developer.sonymobile.com/open-devices/flash-tool/>.
- [48] A. Gupta, K. Preston, and A. Rodriguez, “Debianrunner: Running desktop applications on android smartphones”. <https://www.ideals.illinois.edu/handle/2142/17374>
- [49] M. Kim and A. Wellings, “An efficient and predictable implementation of asynchronous event handling in the rtsj,” Proceedings of the 6th International Workshop on Java Technologies for Real-time and Embedded Systems, pp.48–57, JTRES '08, ACM, New York, NY, USA, 2008. <http://doi.acm.org/10.1145/1434790.1434798>
- [50] A. Corsaro and DouglasC. Schmidt, “The design and performance of the jrate real-time java implementation,” On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE, eds. by R. Meersman and Z. Tari, vol.2519, pp.900–921, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2002. [http://dx.doi.org/10.1007/3-540-36124-3\\_59](http://dx.doi.org/10.1007/3-540-36124-3_59)
- [51] Y. Yan, S. Cosgrove, V. Anand, A. Kulkarni, S.H. Konduri, S.Y. Ko, and L. Ziarek, “Real-time android with rtdroid,” Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services, pp.273–286, MobiSys '14, ACM, New York, NY, USA, 2014. <http://doi.acm.org/10.1145/2594368.2594381>
- [52] I. Kalkov, D. Franke, J.F. Schommer, and S. Kowalewski, “A real-time extension to the android platform,” Proceedings of the 10th International Workshop on Java Technologies for Real-time and Embedded Systems, pp.105–114, JTRES '12, ACM, New York, NY, USA, 2012. <http://doi.acm.org/10.1145/2388936.2388955>
- [53] A. Amiri Sani, K. Boos, M.H. Yun, and L. Zhong, “Rio: A system solution for sharing i/o between mobile systems,” Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services, pp.259–272, MobiSys '14, ACM, New York, NY, USA, 2014. <http://doi.acm.org/10.1145/2594368.2594370>
- [54] Y. Arakawa, S. Tagashira, and A. Fukuda, “mocksensor: Faking remote sensors as embedded sensors for a functional enhancement of android,” Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, pp.66:1–66:2, SenSys '13, ACM, New York, NY, USA, 2013. <http://doi.acm.org/10.1145/2517351.2517385>



- [55] 一般社団法人 日本自動車販売協会連合会, “新車乗用車販売台数月別ランキング”. <http://www.jada.or.jp/contents/data/ranking.html>.
- [56] Commission on measures for quietness of hybrid vehicles, “Report on measures for quietness of hybrid vehicles,” Government report of Ministry of Land, Infrastructure, Transport and Tourism, Jan. 2010.
- [57] NHTSA, “Incidence of pedestrian and bicyclist crashes by hybrid electric passenger vehicles,” 2009. NHTSA Tech. Rep., DOT HS 811 204.
- [58] 朝日新聞, “静かなプリウスでひったくり 被害者「走行音聞こえず」,” 朝日新聞, April 2010.
- [59] University of California, “Hybrid Cars Are Harder to Hear,” 2008. University of California, Riverside, Newsroom, [http://newsroom.ucr.edu/news\\_item.html?action=page&id=1803](http://newsroom.ucr.edu/news_item.html?action=page&id=1803).
- [60] A. Cassat, C. Espanet, R. Coleman, E. Leleu, L. Burdet, D. Torregrossa, J. M’Boua, and A. Miraoui, “Forces and vibrations analysis in industrial pm motors having concentric windings,” IEEE Energy Conversion Congress and Exposition (ECCE), pp.2755–2762, Sept. 2010.
- [61] T. Gandhi and M.M. Trivedi, “Pedestrian protection systems: Issues, survey, and challenges,” vol.8, no.3, IEEE Transactions on Intelligent Transportation Systems, Sept. 2007.
- [62] F. Bu and C.Y. Chan, “Pedestrian detection in transit bus application: sensing technologies and safety solutions,” Proceedings of IEEE Intelligent Vehicles Symposium, pp.100–105, June 2005.
- [63] Daimler, “Mercedes-Benz TechCenter: Night View Assist Plus”. Mercedes-Benz, Technoligen, [http://techcenter.mercedes-benz.com/en/night\\_view\\_plus\\_with\\_spotlight/detail.html](http://techcenter.mercedes-benz.com/en/night_view_plus_with_spotlight/detail.html).
- [64] Toyota, “Night View”. Toyota Global Site, Technology File, Active Safety, [http://www.toyota-global.com/innovation/safety\\_technology/safety\\_technology/technology\\_file/active/night\\_view.html](http://www.toyota-global.com/innovation/safety_technology/safety_technology/technology_file/active/night_view.html).
- [65] K. David and A. Flach, “Car-2-x and pedestrian safety,” vol.5, no.1, IEEE Vehicular Technology Magazine, March 2010.
- [66] L. Oki Electric Industry Co., “Oki succeeds in trial production of world’s first “safety mobile phone” to improve pedestrian safety,” Press release, May 2007.
- [67] R. Feris, B. Siddiquie, Y. Zhai, J. Petterson, L. Brown, and S. Pankanti, “Attribute-based vehicle search in crowded surveillance videos,” Proceedings of the 1st ACM International Conference on Multimedia Retrieval, pp.18:1–18:8, ICMR ’11, ACM, New York, NY, USA, 2011. <http://doi.acm.org/10.1145/1991996.1992014>

- [68] D. He, C. Lang, S. Feng, X. Du, and C. Zhang, “Vehicle detection and classification based on convolutional neural network,” Proceedings of the 7th International Conference on Internet Multimedia Computing and Service, pp.3:1–3:5, ICIMCS ’15, ACM, New York, NY, USA, 2015. <http://doi.acm.org/10.1145/2808492.2808495>
- [69] S. Sivaraman and M.M. Trivedi, “A general active-learning framework for on-road vehicle recognition and tracking,” vol.11, no.2, IEEE Transactions on Intelligent Transportation Systems, June 2010.
- [70] S. Smaldone, C. Tonde, V.K. Ananthanarayanan, A. Elgammal, and L. Iftode, “The cyber-physical bike: A step towards safer green transportation,” Proceedings of the 12th Workshop on Mobile Computing Systems and Applications, pp.56–61, HotMobile ’11, ACM, New York, NY, USA, 2011. <http://doi.acm.org/10.1145/2184489.2184502>
- [71] T. Wang, G. Cardone, A. Corradi, L. Torresani, and A.T. Campbell, “Walksafe: A pedestrian safety app for mobile phone users who walk and talk while crossing roads,” Proceedings of the Twelfth Workshop on Mobile Computing Systems and Applications, pp.5:1–5:6, HotMobile ’12, ACM, New York, NY, USA, 2012. <http://doi.acm.org/10.1145/2162081.2162089>
- [72] H. Tsuzuki, M. Kugler, S. Kuroyanagi, and A. Iwata, “A novel approach for sound approaching detection,” Neural Information Processing. Models and Applications, eds. by K. Wong, B.SumuduU. Mendis, and A. Bouzerdoum, vol.6444, pp.407–414, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010. [http://dx.doi.org/10.1007/978-3-642-17534-3\\_50](http://dx.doi.org/10.1007/978-3-642-17534-3_50)
- [73] Amazon, “Amazon Prime Air,” <https://www.amazon.com/Amazon-Prime-Air/b?node=8037720011>. [Online; accessed 29-Sept.-2017].
- [74] R. Mahony, V. Kumar, and P. Corke, “Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor,” vol.19, no.3, IEEE Robotics Automation Magazine, Sept. 2012.
- [75] M. Orsag, T. Haus, I. Palunko, and S. Bogdan, “State estimation, robust control and obstacle avoidance for multicopter in cluttered environments: Euroc experience and results,” 2015 International Conference on Unmanned Aircraft Systems (ICUAS), pp.455–461, June 2015.
- [76] S. Wang and Y. Yang, “Quadrotor aircraft attitude estimation and control based on kalman filter,” Proceedings of the 31st Chinese Control Conference, pp.5634–5639, July 2012.
- [77] M. Saied, B. Lussier, I. Fantoni, C. Francis, H. Shraim, and G. Sanahuja, “Fault diagnosis and fault-tolerant control strategy for rotor failure in an octorotor,” 2015

- IEEE International Conference on Robotics and Automation (ICRA), pp.5266–5271, May 2015.
- [78] H. Nakanishi, S. Kanata, and T. Sawaragi, “Measurement model of barometer in ground effect of unmanned helicopter and its application to estimate terrain clearance,” 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, pp.232–237, Nov. 2011.
- [79] P. Gsior, A. Bondyra, S. Gardecki, and W. Giernacki, “Robust estimation algorithm of altitude and vertical velocity for multirotor uavs,” 2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR), pp.714–719, Aug. 2016.
- [80] I.F. Mondragn, P. Campoy, M.A. Olivares-Mendez, and C. Martinez, “3d object following based on visual information for unmanned aerial vehicles,” IX Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control, 2011 IEEE, pp.1–7, Oct. 2011.
- [81] J. Pestana, J.L. Sanchez-Lopez, S. Saripalli, and P. Campoy, “Computer vision based general object following for gps-denied multirotor unmanned vehicles,” 2014 American Control Conference, pp.1886–1891, June 2014.
- [82] L.S. Kalantari, S. Mohanna, and S. Tavakoli, “Detection, identification and tracking of flying objects in three dimensions using multistatic radars,” vol.2, no.6, International Journal of Communications, Network and System Sciences, 2009. <https://doi.org/10.4236/ijcns.2009.26053>
- [83] P. Nguyen, H. Truong, M. Ravindranathan, A. Nguyen, R. Han, and T. Vu, “Matthan: Drone presence detection by identifying physical signatures in the drone’s rf communication,” Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services, pp.211–224, MobiSys ’17, ACM, New York, NY, USA, 2017. <http://doi.acm.org/10.1145/3081333.3081354>
- [84] A. Rozantsev, V. Lepetit, and P. Fua, “Detecting flying objects using a single moving camera,” vol.39, no.5, IEEE Trans. Pattern Anal. Mach. Intell., May 2017. <https://doi.org/10.1109/TPAMI.2016.2564408>
- [85] A. Takeki, T.T. Trinh, R. Yoshihashi, R. Kawakami, M. Iida, and T. Naemura, “Combining deep features for object detection at various scales: finding small birds in landscape images,” vol.8, no.1, IPSJ Transactions on Computer Vision and Applications, Aug. 2016. <https://doi.org/10.1186/s41074-016-0006-z>
- [86] K. Higuchi, Y. Ishiguro, and J. Rekimoto, “Flying eyes: Free-space content creation using autonomous aerial vehicles,” CHI ’11 Extended Abstracts on Human Factors in Computing Systems, pp.561–570, CHI EA ’11, ACM, New York, NY, USA, 2011. <http://doi.acm.org/10.1145/1979742.1979627>

- [87] P. Abtahi, D.Y. Zhao, L.E. Jane, and J.A. Landay, “Drone near me: Exploring touch-based human-drone interaction,” vol.1, no.3, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., Sept. 2017. <http://doi.acm.org/10.1145/3130899>
- [88] B. Kim, H.Y. Kim, and J. Kim, “Getting home safely with drone,” Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, pp.117–120, UbiComp ’16, ACM, New York, NY, USA, 2016. <http://doi.acm.org/10.1145/2968219.2971426>
- [89] J.R. Cauchard, J.L. E, K.Y. Zhai, and J.A. Landay, “Drone & me: An exploration into natural human-drone interaction,” Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp.361–365, UbiComp ’15, ACM, New York, NY, USA, 2015. <http://doi.acm.org/10.1145/2750858.2805823>
- [90] H. Nozaki, “Flying display: A movable display pairing projector and screen in the air,” CHI ’14 Extended Abstracts on Human Factors in Computing Systems, pp.909–914, CHI EA ’14, ACM, New York, NY, USA, 2014. <http://doi.acm.org/10.1145/2559206.2579410>
- [91] Y. Kyono, T. Yonezawa, H. Nozaki, M. Ogawa, T. Ito, J. Nakazawa, K. Takashio, and H. Tokuda, “Evercopter: Continuous and adaptive over-the-air sensing with detachable wired flying objects,” Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication, pp.299–302, UbiComp ’13 Adjunct, ACM, New York, NY, USA, 2013. <http://doi.acm.org/10.1145/2494091.2494183>
- [92] GENELEC, “Data Sheet Genelec 1029A Bi-amplified Monitoring System,” <https://www.genelec.com/sites/default/files/media/Previous%20models/2-way%20loudspeakers/1029A/ds1029a.pdf>. [Online; accessed 30-Nov.-2017].

## 発表文献

---

### 論文誌

- [1] 高木雅, 池内尚史, 角谷和宣, 川原圭博, “マイクロ波給電を応用したスマートフォンの外付け電源管理システム,” 情報処理学会論文誌 (査読中).
- [2] 高木雅, 川原圭博, 浅見徹, “スマートフォンを用いた電気自動車の検知に対する機械学習によるアプローチ,” 情報処理学会論文誌, vol.58, no.10, pp.1617-1627, Oct. 2017. (特選論文)
- [3] 高木雅, 川原圭博, 浅見徹, “アプリ毎のトラヒックとユーザの利用状況を考慮したスマートフォンの通信制御手法,” 情報処理学会論文誌, vol.56, no.3, pp.1121-1131, March 2015. (推薦論文)

### 国際会議

- [4] M. Takagi, Y. Kawahara, and T. Asami, “A Low-Power Sensing Method Using Linux Kernel on Android Devices,” Proc. ACM SenSys 2015, pp.413-414, Seoul, South Korea, Nov. 2015.
- [5] M. Takagi, K. Fujimoto, Y. Kawahara and T. Asami, “Detecting hybrid and electric vehicles using a smartphone,” Proc. ACM UbiComp 2014, pp.267-275, Seattle, United States, Sept. 2014. (Best Paper Nominee Award)

### 特許

- [6] 浅見徹, 高木雅, 川原圭博, “情報処理装置及びプログラム,” 特許第 6161439 号, June, 2017.

### 国内研究会

- [7] 高木雅, 藤本浩介, 川原圭博, 浅見徹, “スマートフォンを用いた電気自動車およびハイブリッド車の接近検知手法,” 情処研究報告, vol.2014-UBI-43(12), pp.1-8, Jul. 2014. (優秀論文賞受賞, 山下記念研究賞受賞)

- [8] 高木雅, 藤本浩介, 川原圭博, 浅見徹, “機械学習によるスマートフォンを用いた電気自動車の接近検知手法,” 信学技報, vol.114, no.65, ASN2014-20, pp.67-68, May 2014. (若手研究奨励賞受賞)
- [9] 高木雅, 川原圭博, 浅見徹, “3G トラヒック削減とプッシュ通知を共存させるアプリ単位の通信制御手法,” DICO2013 論文集, 7F-2, vol.2013, pp.1727-1734, July 2013. (優秀論文賞受賞)
- [10] 池内尚史, 高木雅, 川原圭博, 浅見徹, “マイクロ波給電を用いた Android スマートフォンの遠隔起動トリガに関する実験的検討,” 情処研究報告, vol.2016-UBI-52(10), pp.1-8, Oct. 2016 (学生奨励賞).
- [11] 林直人, 繁田亮, 高木雅, 川原圭博, 浅見徹, “インドの農村部での土壌センシングシステムの実証: 期待と現実,” DICO2016 論文集, 3E-1, vol.2016, pp.565-571, July 2016.

## 国内全国大会

- [12] 高木雅, 川原圭博, “無線 LAN パケットの送信タイミング制御によるオンオフ変調でのセンサノード向け情報の埋め込み,” 信学ソ大, B-18-27, Sept. 2017.
- [13] 高木雅, 池内尚史, 川原圭博, 浅見徹, “Android 端末による間欠センシングのためのマイクロ波給電を用いた選択的遠隔起動トリガ,” 信学総大, B-18-1, March 2017.
- [14] 高木雅, 川原圭博, 浅見徹, “エッジコンピューティング基盤としての中古スマートフォンの機種依存と性能評価,” 信学ソ大, B-18-10, Sept. 2016.
- [15] 高木雅, 川原圭博, 浅見徹, “リカバリモードを用いた省電力なセンシングにおける Android 端末と外部機器の連携手法,” 信学総大, B-18-18, March 2016.
- [16] 高木雅, 川原圭博, 浅見徹, “Android 端末の Linux カーネルを用いた低電力なセンシング手法,” 信学ソ大, B-18-28, Sept. 2015.
- [17] 高木雅, 川原圭博, 浅見徹, “LTE 網の MAC 層パディングを活用した無電力でのセンサデータ送信,” 信学総大, B-15-14, March 2015.
- [18] 高木雅, 川原圭博, 浅見徹, “スマートフォン・トラヒックのアプリ単位での分析,” 信学総大, B-15-7, March 2014.
- [19] 高木雅, 川原圭博, 浅見徹, “ユーザ操作に基づく動的な Idle Timer による 3G 制御信号の削減手法,” 信学ソ大, B-15-2, Sept. 2013.
- [20] 高木雅, 川原圭博, 浅見徹, “アプリ単位の動的な通信制御を用いた Android 端末の 3G 制御信号と消費電力の削減手法,” 信学総大, B-15-8, March 2013.

- [21] 高木雅, 川原圭博, 浅見徹, “スマートフォンの Wi-Fi リンク品質の機種依存性に関する一実験,” 信学ソ大, B-15-9, Sept. 2012.
- [22] 池内尚史, 高木雅, 石毛真修, 角谷和宣, 川原圭博, 浅見徹, “Android スマートフォン遠隔起動トリガの遠隔監視システム,” 信学総大, B-18-2, March 2017.
- [23] 石毛真修, 高木雅, 池内尚史, 川原圭博, 浅見徹, “スマートフォンを用いた間欠的センシングのための外付け電源管理システムの設計,” 情報処理学会全国大会, 5S-06, March 2017. (学生奨励賞受賞)
- [24] 藤本浩介, 高木雅, 川原圭博, 浅見徹, “電気自動車の特徴音を用いたスマートフォンによる車両接近検知手法,” 信学総大, B-15-6, March 2014.
- [25] 藤本浩介, 高木雅, 川原圭博, 浅見徹, “スマートフォンのマイクロフォンを用いた電気自動車接近時の高周波音の計測,” 信学ソ大, B-15-9, Sept. 2013.

## 受賞

- [26] 特選論文, 情報処理学会, Oct. 15, 2017. ([2] による)
- [27] 山下記念研究賞, 情報処理学会, March 10, 2016. ([7] による)
- [28] テレコムシステム技術学生賞, 電気通信普及財団, March 28, 2016. ([5] による)
- [29] 研究会優秀論文賞, 情報処理学会 UBI 研究会, May 11, 2015. ([7] による)
- [30] Best Paper Nominee Award, ACM Ubicomp 2014, Sept. 15, 2014. ([5] による)
- [31] 若手研究奨励賞, 情報処理学会 ASN 研究会, May 30, 2014. ([8] による)
- [32] 優秀論文賞, 情報処理学会 DICOMO 2013, Aug. 19, 2013. ([9] による)
- [33] 学生奨励賞, 情報処理学会 UBI 研究会, May 25, 2017. ([10] による)
- [34] 学生奨励賞, 情報処理学会 全国大会, March 16, 2017. ([23] による)
- [35] 研究科長賞, 東京大学大学院 情報理工学系研究科, May 24, 2015.
- [36] 専攻長賞, 東京大学大学院 情報理工学系研究科 電子情報学専攻, May 24, 2015.
- [37] 優秀卒業論文賞, 東京大学 工学部 電子情報工学科, May 26, 2013.