ネットワーク共用デバイスを用いた Ephemeral CPS の構成法に関する研究

Study on Ephemeral CPS Construction Using Shared Devices in the Network

2020 年 2 月

野口　博史

Hirofumi NOGUCHI

ネットワーク共用デバイスを用いた Ephemeral CPS の構成法に関する研究

Study on Ephemeral CPS Construction Using Shared Devices in the Network

2020 年 2 月

早稲田大学大学院　創造理工学研究科

野口　博史

Hirofumi NOGUCHI

目次

図目次

表目次

第1章　序論

1.1 はじめに

情報通信の進化により，社会に大きな変革が起こっている．インターネットは 1960 年代の誕生以来, 世界中のあらゆる情報の共有を可能にしてきた. いまや, インターネットには, 世界各地の気象情報やマイナースポーツの試合速報，さらには，個々人の趣味や，つぶやきといったパーソナルな情報まで，ありとあらゆる情報が公開されている．そして，近年，インターネットはさらに進化し，IoT (Internet of Things) が急速な普及拡大を見せている．IoT とは，あらゆるモノをインターネットに接続し，それらが発信するデータを収集・分析した後に，モノを最適な状態に導くようにフィードバックする一連のシステムを示す言葉である．もはやインターネットに流通するのは人が発信する情報コンテンツだけではない. 家電や事務機器，自動車，製造機械，建設機械，さらには小型センサといった多種多様なモノがインターネットで結ばれつつあり，それらが発する情報を高度に処理してもたらされる高価値なサービスが誕生し始めている．このような時代において，サービスの提供形態や利用者の思想そのものが大きく変わりつつある．

本論文は，あらゆるモノがネットワークを介してつながる時代におけるシステムの在り方を扱うものである．特に，様々な社会課題の解決に期待が寄せられている CPS (Cyber-Physical System) に関して，技術革新のみならず，ビジネスモデルやサービス要件の変化にも触れて論ずる．

序論となる本章では，IoT や CPS を取り巻く近年の社会動向と，そこから予想される将来の IoT の姿を述べる. さらに, 本研究が目指す将来の CPS の提供形態と要件を示した後に, 本研究の目的を述べる．

1.2 IoT に関する動向と期待

本節では，近年の情報通信サービスの潮流である IoT に関して，世界的な動向および，日本における社会課題解決への期待と取り組みを述べる．

1.2.1 IoT に関する世界の動向

2020 年時点において，家屋やオフィスといった日常の生活環境や，工場などの製造現場には，ネットワークにつながる機器が増加し始めている．Cisco Systems が公開している IoT

に関する予測[1]によれば，既に 200 億以上のデバイスがネットワークにつながっており，2022 年にはその数は 285 億を超えると言われている．図 1.1 は，同社が公開している世界の IoT デバイス数の増加予想である．最も成長が著しいカテゴリは M2M (Machine to Machine) であり，2022 年には全接続デバイスのうち 51%以上を占める予想である．また，図 1.2 は, 同社が公開する業種ごとの M2M 接続数の増加予想である. 2022 年の世界の M2M 接続数は，2017 年の 2.4 倍になると言われている．そして，ネットワーク接続デバイスの増加とともに，IP (Internet Protocol) トラフィックの大幅な増加が予想されている．表 1.1 は 1992 年から 2022 年までのインターネットトラフィックの変遷，図 1.3 は 2017 年から 2022 年までの IP トラフィックの予測である．インターネットが急成長を続けてきたことが分かるとともに，今後も成長を続けていくことが予想されている．2022 年の全世界の IP トラフィックは月間 396EB (exabyte) に達することが予測されており，インターネットを介して膨大なデータが流通する世界が到来する．

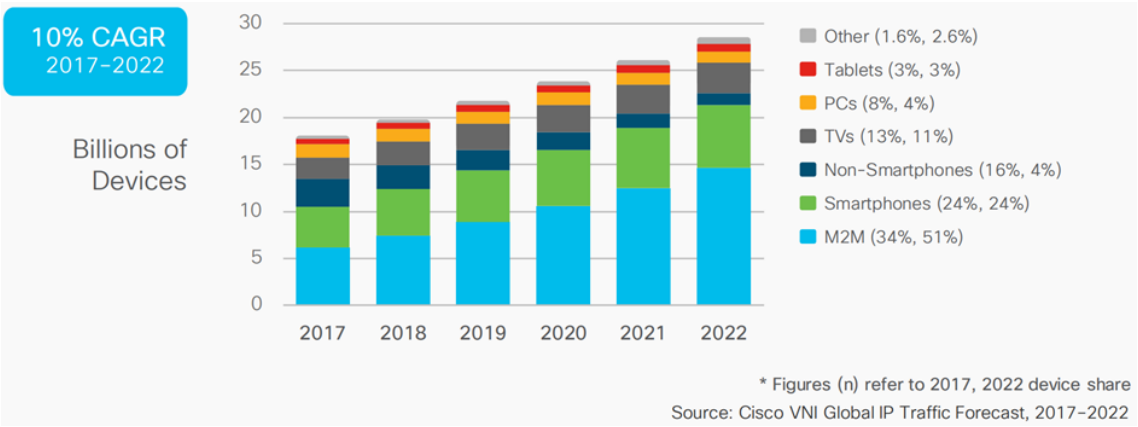


図 1.1　世界的なデバイス数と接続数の増加（出展　Cisco VNI 2018[1]）

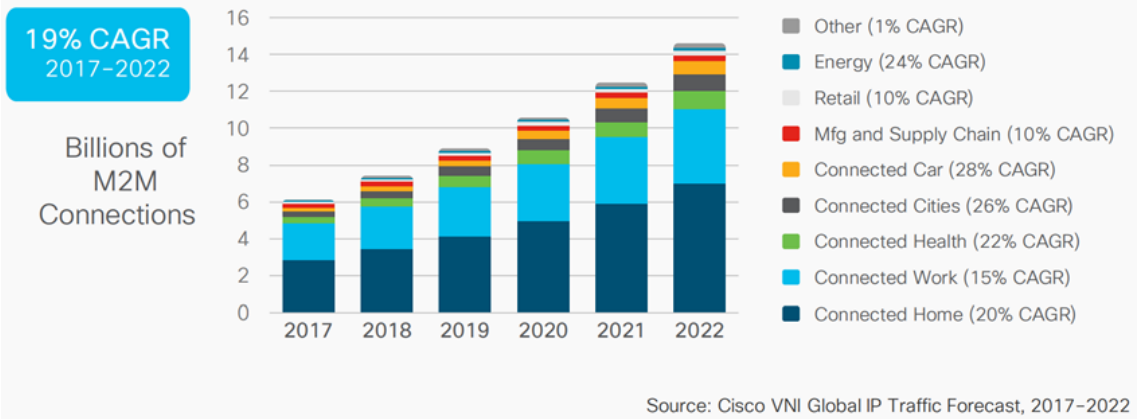


図 1.2　全世界における M2M 接続の増加（業種別）（出展　Cisco VNI 2018[1]）

表 1.1 インターネットトラフィックの変遷（出典 Cisco VNI 2018[1]）

年	世界のインターネットトラフィック
1992 年	100 GB/日
1997 年	100 GB/時
2002 年	100 GB/秒
2007 年	2,000 GB/秒
2017 年	46,600 GB/秒
2022 年	150,700 GB/秒

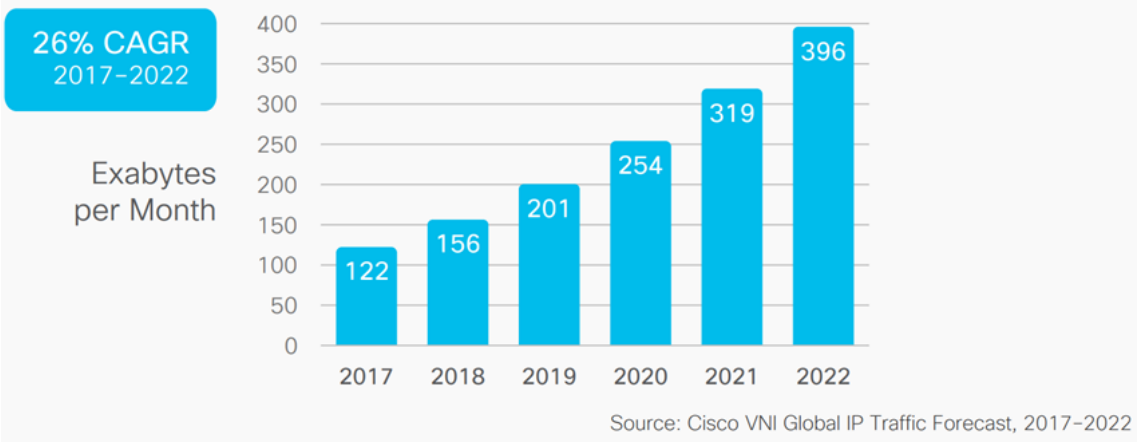


図 1.3 世界の IP トラフィック予測（出展 Cisco VNI 2018[1]）

また，McKinsey は，世界の IoT の市場規模が，2025 年に 11.1 兆ドルに到達すると予測し，その市場として，Home, Offices, Factories, Retail environments, Worksites, Human, Outside, Cities，Vehicles の 9 つを挙げている[2]．つまり，人間の生活，労働に関わるほぼ全ての環境が該当する．実際に，ホーム IoT[3][4], ファクトリーIoT[5], 電力 IoT[6], ヘルスケア IoT[7][8], 農業 IoT[9][10]，シティ IoT[11][12]，ビークル IoT[13]等，IoT に関する多数の研究開発およびサービス事例が既に公表されている．

以上をまとめると，IoT の普及に合わせてインターネットの需要は増加の一途をたどっており，また，IoT 市場は，今後エネルギーや交通，医療，介護など非常に多岐にわたる事業ドメインに拡大していくことが予想される．

1.2.2 日本における IoT への期待

日本においても IoT に関する顕著な動きがある．内閣府は第 5 期科学技術基本計画[14]において，日本が目指すべき未来社会として Society 5.0 [15]という構想を提唱した（図 1.4）．

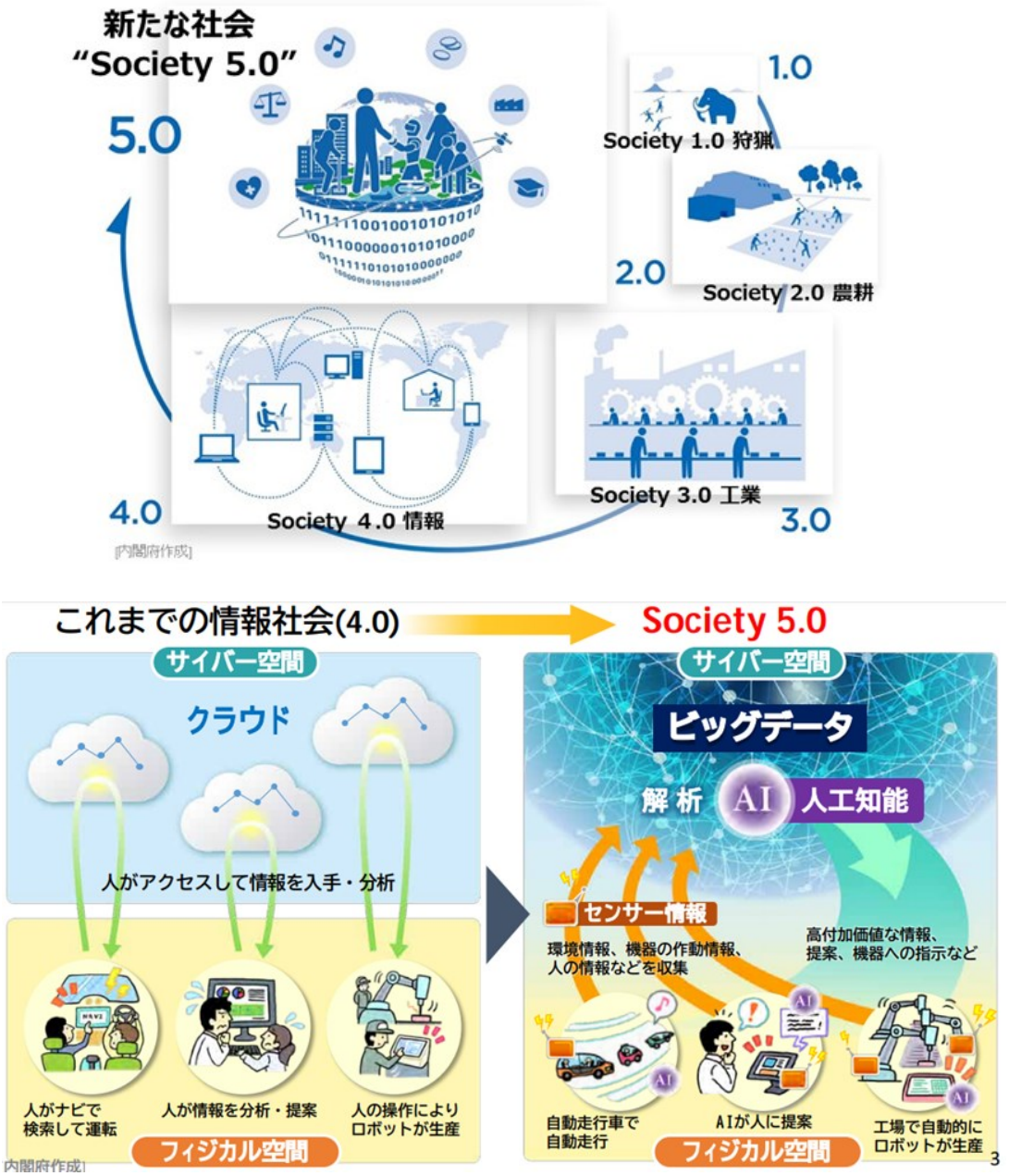


図 1.4 Society 5.0 構想（出展 内閣府 Society 5.0 [15]）

内閣府によれば，Society 5.0 は，狩猟社会（Society 1.0），農耕社会（Society 2.0），工業社会（Society 3.0），情報社会（Society 4.0）に続く新たな社会であり，「サイバー空間（仮想空間）とフィジカル空間（現実空間）を高度に融合させたシステムにより，経済発展と社会的課題の解決を両立する，人間中心の社会（Society）」とされている．その対象領域は，交通，医療，介護，ものづくり，農業，食品，防災，エネルギーと多岐に渡り，官民一体となっての研究開発，環境整備が進められている[16][17][18].

Society5.0 が対象としている，サイバー空間とフィジカル空間の融合を扱うシステムはサイバーフィジカルシステム（Cyber-Physical System, CPS）と呼ばれる．CPS には，実環境からの大量のデータ収集を可能にする IoT が重要な要素とされている[19]．今後，IoT の普及拡大により大量のデバイスがネットワークにつながる将来には，日本はもちろん世界中で多くの CPS の運用が予想される．

1.3 情報通信サービスに関する社会的変化

IoT を取り巻く動向はこれまでに述べた，ネットワーク接続デバイスの増加やサービスの普及拡大だけではない．近年，情報通信サービスそのものにパラダイムシフトが起こっている．IoT の普及に合わせて，新たなビジネスモデルやサービス利用者の思想の変化が生まれている．本節は，このような情報通信サービスに関する二つの顕著な社会的変化を述べる．

1.3.1 データの横断的利用

近年，異なる事業者，サービスが保有するデータの相互利用の期待が高まっている．例えば，内閣府による戦略的イノベーション創造プログラム（SIP）[20]において，ビッグデータ・AI (Artificial Intelligence) を活用したサイバー空間基盤技術推進委員会[21]は，「国，地方公共団体，民間などで散在するデータを連携させ，ビッグデータとして扱い，分野・組織を越えたデータ活用とサービス提供を可能とするため，関係府省庁で整備が進められている分野ごとのデータ連携基盤やその他の様々なデータを相互に連携させる分野横断のプラットフォーム『分野間データ連携基盤』を実現する.」と唱っている．また，同じく SIP におけるフィジカル空間デジタルデータ処理基盤推進委員会[22]は,「企業が保持している情報（生産管理データ，在庫管理データ，勤務情報，等）や公共情報（気象情報，交通情報，カレンダー情報，等）等と重ね合わせることで企業の生産管理や需要予測に基づく適応制御等を可能とするとともに，業種分野間の共通要素を抽出しながらの状況把握と蓄積等により，サイバー空間と連動するための情報を提供する.」と唱っている．このように，これまで個々の事業者や自治体に閉じていたデータの横断的利用を推進する取り組みが政府主導で進められている．

また，McKinsey の調査[2]によれば，IoT デバイスが生成するデータの大半は利用されずに廃棄されているという事実がある．これは，現行の多くの IoT サービスが事故検知や異常検知などを目的とするイベント型であり，平時のデータは監視目的にしか使われていないことに起因する．つまりデバイスを稼働させるエネルギーとネットワーク帯域に関して現行の IoT サービスの多くは非効率であると言える．分野・組織を超えたデータの連携活用には，このような現行使用されていないデータの利用機会を増加させ，IoT インフラの効率的な活用という効果も期待できる．

さらに，ビッグデータ分析技術の発展もデータの横断的利用を促進させる要因である．これまでは，データから目的の分析結果を獲得するためには，専門家による緻密な分析が求められていた．そのため，多種類のデータを扱うには高度な知識と分析能力が求められていた．ところが，近年著しい発展を見せている深層学習に代表される AI 技術によって，多種類，大量のデータから，相関関係や適切なシステム制御値を簡易に求められるようになりつつある．したがって，例えば，気象データと購買データのように従来は一緒に扱うことがなかったデータを組み合わせて分析を行う障壁が低くなっており，横断的なデータ利用の需要は今後ますますの増加を見せていくと思われる．

1.3.2 サブスクリプションサービスの台頭

近年の消費者の思想として，モノからコトへの支払いへの変化，つまり，モノを所有するのではなく，一時的に使用するという考え方が主流になりつつある．このような，思想を体現したビジネスモデルがサブスクリプションサービスである．サブスクリプションサービスとは，その名が示す通り定期購読型のサービスである．特定の商品や機能を購入して永続的に保有する従来のサービスと比べ，導入と管理にかかる費用を抑えられる．既に様々な業界でサブスクリプションサービスは普及拡大を見せている．例えば，Web コンテンツ業界では，音楽や書籍，動画といった大量のコンテンツを契約期間内に定額で提供するサービスが数多く存在する．また，ソフトウェア業界においては，定期的な更新が必要なセキュリティソフトウェアなどがサブスクリプションサービスとして提供されている．さらに，現在最も普及しているサブスクリプションサービスの一つはクラウドコンピューティングサービスである．IaaS (Infrastructure as a Service) と呼ばれるコンピュータリソースを貸し出すクラウドコンピューティングサービスでは，必要な時にだけコンピュータリソースを借用し，使用した時間やリソース量に応じた料金を支払う形態をとっている．コンピュータを所有するよりもはるかに安価かつ迅速にコンピュータリソースを使用できる．また，ソフトウェア機能を API (Application Programming Interface) として提供し，その使用回数に応じた課金を行うサービスも普及している．

最も有名なクラウドコンピューティングサービスの一つである Amazon Web Service [23] は，多数のサブスクリプションサービスを提供している．同サービス群の中で IaaS に位置づけられる Amazon Elastic Compute Cloud (Amazon EC2) は，インターネット経由で契約が完了次第，ただちにコンピュータリソースが提供される．料金は，コンピュータの使用時間に応じた従量課金となる．また，API として提供されるサービスの一つには，Amazon Rekognition Image がある．これは，機械学習を用いた画像分析機能を API として提供し，画像を分析した回数に応じて課金するサービスである．本サービスを利用することで，高価な機械学習ソフトウェアや実行環境の購入が不要になる．

Microsoft Azure [24]も著名なクラウドコンピューティングサービスである．Amazon Web Service と同様に，Virtual Machines という名の IaaS サービスや機械学習機能を提供してい

る．機械学習サービスの課金形態が細分化されており，単純な API の呼び出し回数ではなく，使用する機能の数に応じて課金される．例えば，画像分析サービスである Computer Vision API では，一つの画像に対して，物体抽出，顔認識という 2 種類の分析機能を使用する場合に，それぞれに料金が課金される．

さらに，同じく代表的なクラウドコンピューティングサービスの一つである Google Cloud [25]も上記と同様のサービスを提供している．例えば，画像分析サービスである Cloud Vision API は，Microsoft Azure と同様に，分析画像数ではなく分析内容に応じて課金される．

このように，クラウドコンピューティングサービスは，大きな市場シェアを持つ各社が横並びに商品を揃えており，その課金形態も一般化されている．今後も多数のクラウドコンピューティングサービスがサブスクリプションサービスとして提供されることが予想される．

以上のように，サイバー空間で提供される情報通信サービスにおいては，既にサブスクリプションサービスが一般消費者に受け入れられ普及している．

フィジカル空間を対象にするサービスにおいても，近年サブスクリプションサービスが増加している．洋服やコーヒーメーカ，ウォーターサーバといった日用品を提供するサブスクリプションサービスが既に存在する．特に，世界的に拡大を見せているサービスの一つに，カーシェアリングがある．自動車を所有するのではなく，共用の車を一時利用することで低価格化を実現している．自動車に関する潮流としてさらに，MaaS (Mobility as a Service) [26]と呼ばれる構想がある．これは，電車や航空機，バスといった様々な移動手段を統一的に扱い，移動という一つのサービスとすることで最適化を図るという概念である．例えば，異なる運営団体が提供する移動手段を跨った最適な移動ルートの提案や，それらを統合した運賃設定と課金が行われる．MaaS の普及により，切符や定期券による支払い，複数人での乗合いという，元来サブスクリプションサービスとしての特性を持っていた公共交通機関の需要が今後高まることが予想される．

また，これらのサブスクリプションサービスと密接な関係を持つのがシェアリングという概念である．管理組織や共同体が，個人の遊休資産や共有のモノを管理し，必要な時にだけ個人に貸し出すことで，個人は費用を折半する概念である．クラウドコンピューティングサービスや MaaS も見方によればシェアリングのもとに成り立っているサービスと言える．近年，若者を中心にシェアリングの思想は社会的な拡大を見せており，平成 30 年度版情報通信白書[27]によれば，シェアリングの仲介サービスであるシェアリングエコノミーの経済効果は増加の一途をたどっている．今後，共有されるモノの種類や数が増加していくことが予想される．

以上に述べた，サブスクリプションサービスの台頭を鑑みると，将来の IoT サービスは，共有のデバイスやコンピュータを一時利用する提供形態が一般的になると予想される．

1.4 オープン IoT の到来

ここまでに述べた IoT に関する動向と情報通信サービスに関する社会的変化から，将来の IoT の姿を予想する．IoT デバイスの増加，および多様な環境での利用からは，目的や要求品質が異なる多様なサービスの登場が予想される．また，データの横断的利用とサブスクリプションサービスの台頭からは，データやデバイスを共用化することへの需要が高まることが予想される．以上より，将来の IoT とは，ネットワークでつながったデバイスやコンピュータ，データが多数のサービスにオープンに提供され，共用される世界と予想する．本研究では，このような将来の IoT を“オープン IoT”と呼ぶ．

次項より，オープン IoT の効用と，それを前提とした将来の CPS を述べる．

1.4.1 オープン IoT の効用

図 1.5 は，現行の IoT と将来のオープン IoT の比較である．現行の IoT サービスの多くは，一つのサービス事業者が占有のデバイスとアプリケーション，コンピュータを用意するサイロ型のシステム構成をとっており，デバイスの敷設，維持管理に関わる費用が大きい．一方で，オープン IoT は，環境内に既に設置されている共用デバイスと，オープンなソフトウェアを組み合わせて一時的なシステムを構成するものであり，一つのサービスあたりの費用負担が少ない．また，広範囲に設置された大量の共用デバイスを使用できるため，大規模にサービスを展開することが可能である．オープン IoT の効用を以下に示す．

・安価なサービス運用

デバイスやコンピュータを所有せず，あくまで一時的に利用するため，クラウドコンピューティングサービスと同様に，それらの敷設と維持管理にかかる費用が多数の事業者に分散されることになり，一つのサービスあたりの運用コストを安価に抑えられる．

・物理的・論理的に広範囲に存在するデバイス，データの利用

物理的に広範囲に設置されたデバイスの使用やデータ収集が可能である．例えば，市や県といった広域に分散したセンサを活用したサービスが即座に開始できる．また，自治体や工場といった，業種を跨った論理的に広い範囲からのデータ収集が可能である．

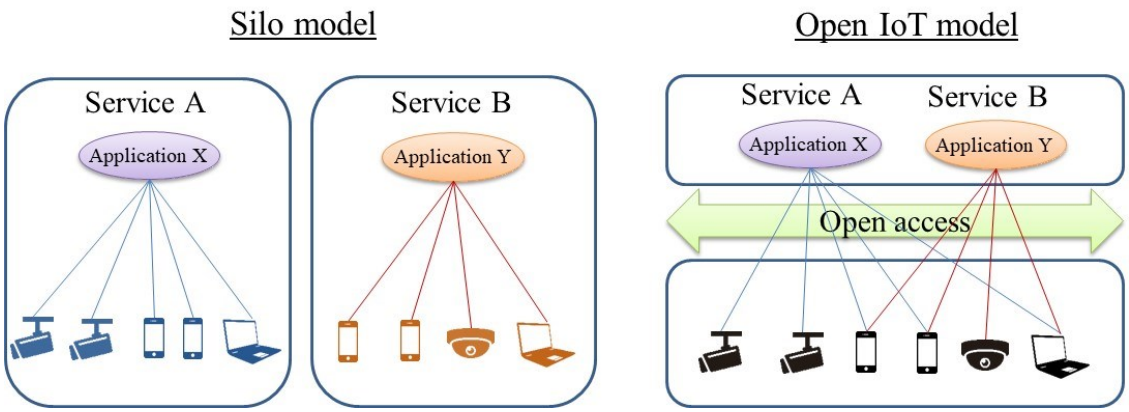


図 1.5 サイロ IoT とオープン IoT の対比
(©IEEE 主著論文 3 より引用)

オープン IoT が可能にするサービスの一例として，街頭の監視カメラや通行人が所有するカメラを利用した子供の見守りサービスがある．現行のサイロ型 IoT は，子供の通学路全域に見守り専用のカメラを設置して管理することが必要なため，莫大な費用がかかる．一方で，オープン IoT は，平時は店舗の監視やドライブレコーダーとして稼働しているカメラを必要な時にだけ使用するため，複数サービス間で費用を分担し，初期投資を削減できる．また，既にネットワークに接続されているデバイスを利用するため，迅速にサービスを開始できる．

1.4.2 オープン IoT による CPS の普及拡大への期待

オープン IoT がもたらす効用を CPS の観点からより詳細に述べる．CPS もオープン IoT の恩恵を享受するシステムである．CPS の重要な要件の一つにフィジカル空間のデータ収集がある．オープン IoT の到来によって，物理的広範囲に分散した共用センサによるデータ収集が可能になり，現行の CPS よりも高精度な制御，複雑な処理が行えるようになる．

オープン IoT の恩恵はこれだけではない．CPS において，フィジカル空間上のデバイスの敷設，維持管理費用は導入障壁の一つであった．財務局による調査（財務局調査による「先端技術（IoT，AI 等）の活用状況」について[28])によれば，IoT，AI，ロボット，クラウドコンピューティング，ビッグデータのうち，企業が活用済みの先端技術として最も回答が多いのは，クラウドコンピューティングである．一方で，活用したくてもできない優先度の高い先端技術として，ロボット，AI，ビッグデータが上位に挙がっている．その理由として費用対効果が低いと回答した企業の割合が高い．商用利用されているサービスロボットや製造ロボットの費用を見ると，例えば，ソフトバンク社のサービスロボットである Pepper [29] の 2020 年時点における家庭用販売価格は，本体 20 万円弱であり，別途に月額約 27,600 円のサポート費用がかかる．等身大のサービスロボットにしては安価とも言えるが，クラウド

コンピューティングサービスの最小価格帯が数万円以内であることと比較して高価である．

また，総務省による，「我が国の ICT の現状に関する調査研究[30]」には 100 件以上の IoT サービス事例が記載されているが，フィジカル空間を扱うサービスはわずか数 10 件程度である．さらに，その大半はスマートフォンへの通知といった力学的作用を伴わないサイバー空間の延長と言えるアクションのみを扱ったものである．

以上より，ロボットに代表されるアクチュエータと，それを利用するサービスは未だ普及に至っていないことが分かる．オープン IoT によるデバイスの共用化によってロボット等のアクチュエータの導入費用が軽減されることで，フィジカル空間へのアクションを伴う多様な CPS の普及拡大が期待できる．

1.4.3 IoT，CPS に関するフレームワークの現状

オープン IoT における CPS は，環境内に設置されている共用デバイスと，オープンなソフトウェアを組み合わせて一時的に構成されるシステムであると考えられる．このような CPS の構成を考える上で，既存の IoT，CPS のフレームワークの現状を分析する．IoT のサービスシステムや CPS の構築・運用を支援するための複数のフレームワークが研究開発されている．以下，IoT サービスシステムのフレームワークと，フィジカル空間を対象とする代表的なシステムであるロボットのフレームワークを順に分析する．

はじめに，IoT サービスシステムのフレームワークを分析する．IBM 社が提供する IBM Node-RED [31]は，IoT サービスシステムのビジュアルプログラミングが行えるフレームワークである．本フレームワークを用いることで，複数のソフトウェアと機能を組み合わせて構成されるシステムを直観的なインタフェースを用いて構築できる．一方で，本フレームワークは，あくまでも人手による設計と検証の支援を目的としたものである．

Stack4Things [32] [33]は，オープンソースのクラウドコンピューティングプラットフォームとして著名な OpenStack [34]をベースとするデバイス管理用のフレームワークである．OpenStack は，データセンタ等において，多数のコンピュータ上に，多数の仮想的なコンピュータを作成し，提供するためのコンポーネント群で構成されるオープンソースソフトウェアである．Stack4Things は，OpenStack が有する，認証やユーザインタフェースといったクラウドコンピューティングを運用するための基本機能を用いて，センサとアクチュエータといったデバイスを使用者に対して抽象化する機能を提供する．このような抽象化は，オープン IoT の共用デバイスを活用する上で有用であるが，本フレームワークは，抽象化されたデバイスを組み合わせたサービスシナリオやシステムの動作論理の構築は扱っておらず，CPS の構成には人手による設計と検証を必要とする．

次に，ロボットのフレームワークを分析する．Ubibot [35]は，複数のデバイスを組み合わせて堅牢なロボットシステムを構築するためのフレームワークであり，複数のプロジェクトが進行している．オープン IoT の共用デバイスの管理，活用に一部有用ではあるが，オープン IoT の特徴の一つである，使用可能なデバイスが動的に追加，削減されることは想定し

ておらず，対応するための機能を有していない．

Nishio 等が提案する UNR-PF [36]は，ロボット等のネットワークに接続されたデバイスの利用を目的とするプラットフォームである．デバイスの場所と状態を管理するローカルプラットフォームと，複数のローカルプラットフォームを一元管理するグローバルプラットフォームで構成され，グローバルプラットフォームがサービス要求を受け付けると，ローカルプラットフォームが適切なデバイスを予約して使用を開始する．ネットワーク内の共用デバイスから適切なデバイスを選択して使用するための有用な機能を有しているが，デバイスの具体的な制御，処理はサポートしていない．

Ren-v [37]は NTT が開発したロボット，センサ，アプリケーション連携サービスのためのプラットフォームである．サービスの状態とアクションが記述されたスクリプトを GUI (Graphical User Interface) で作成する機能を提供しており，システムの簡易な設計を可能にする．サービス開発者の負担を軽減するものではあるが，あらかじめデバイスを指定したうえで，人手でサービスシナリオを設計することを前提としており，共用デバイスの利用において想定される，多種多様なデバイスの動的な組み込みをサポートしていない．

ROS (Robot Operating System) [38]は，オープンソースのロボットソフトウェアフレームワークである．複数のデバイスを連携させるために必要となる，インタフェース変換，デバイス管理および，シミュレータといったツールを提供している．また，複数デバイスを publish /subscribe モデルのネットワークで疎結合に結ぶ機能を提供しており，ネットワークに分散するデバイスを用いて大規模なシステムを構築することができる．ネットワークに分散する複数の共用デバイスを組み合わせることに有用であるが，前述の他フレームワークと同様に，デバイスの制御論理は開発者が設計することを前提としており，システムを構成するデバイスの組み合わせが動的に変更されることを想定したものではない．

以上の通り，既存の IoT，CPS のフレームワークは，いずれも人手による設計と検証に基づき，指定されたデバイスの使用を前提としたものである．これらのフレームワークは，共用デバイスを用いた CPS を構成する上で大きな障壁がある．

CPS のようにフィジカル空間を扱うシステムは，動的に変化する環境の影響を受けるため，システムの処理結果を保障するには，元来，人手と時間をかけた設計と検証が必須である．オープン IoT の共用デバイスを用いるということは，設置される環境が様々かつ，機能や性能が多様な不特定多数のデバイスを組み合わせるということであるため，これら既存のフレームワークを単純に適用するだけでは，オープン化による恩恵を享受する以前に，設計と検証にかかるコストを大幅に増大させてしまう．

つまり，ネットワークに接続された共用デバイスを活用し，オープン IoT の恩恵を享受するには，CPS の構成法にパラダイムシフトが求められる．

1.5 Ephemeral-Cyber-Physical System

本研究では，オープン IoT の恩恵を享受する革新的な CPS として，「Ephemeral-Cyber-Physical System」を提案する（図 1.6）．（以降 E-CPS と表記する．）E-CPS とは，ネットワークに接続された共用デバイスを一時的に利用して，オンデマンドに構成される CPS である．

Ephemeral とは，英語で「束の間の，短命」を意味する言葉である．クラウドコンピューティングサービスにおいて，Ephemeral storage と呼ばれるデータ保存領域がある．これは，共用コンピュータ上に仮想コンピュータを起動している短期間にのみ構成され，仮想コンピュータの停止と同時に削除されるデータ保存領域である．E-CPS は，本コンセプトを CPS に適用したものである．E-CPS は，サービス要求に応じて，ネットワークに接続された多数の共用デバイスの中から，必要なものを一時的に組み合わせて構成される CPS であり，サービス終了と同時に解体され，使用していた共用デバイスを開放するものである． CPS を構成する上で従来必須であった人手によるデバイスの指定や，デバイスの組み合わせに応じたシステム設計，検証を排し，クラウドコンピューティングサービスのように，簡易，迅速，安価，かつ大規模スケールに構成される CPS である．

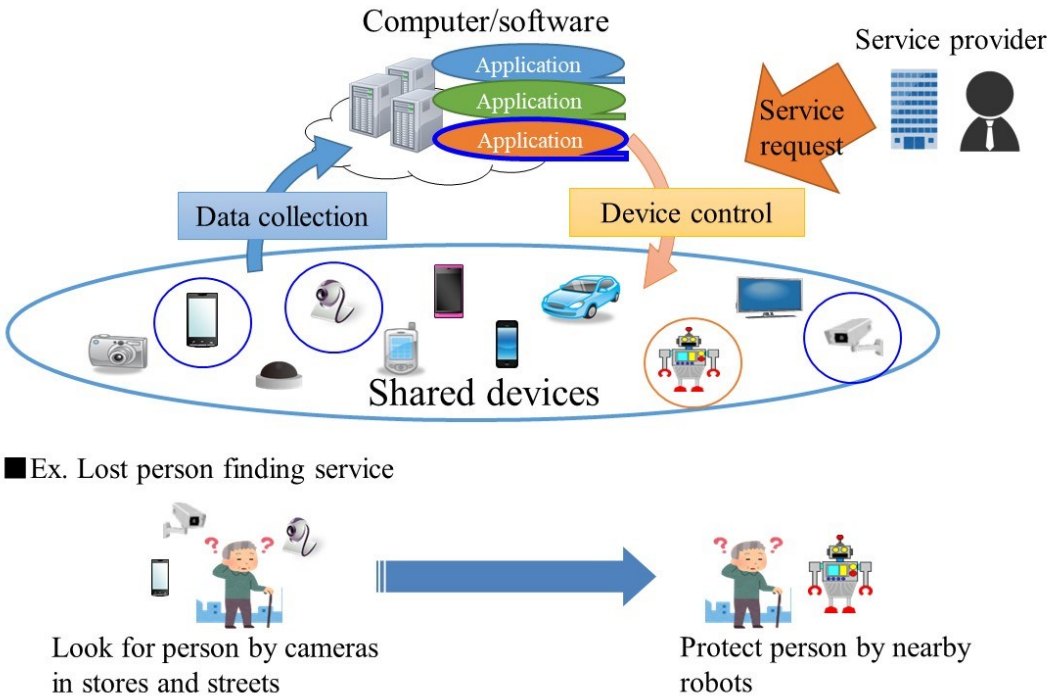


図 1.6 Ephemeral-Cyber-Physical System の概略

1.5.1 E-CPS の効用

E-CPS を実現することで，様々なサービスへの効用が期待できる．具体的なサービス領域ごとに，現行の CPS から E-CPS に移行することで向上する価値を表 1.2 に整理した．本表に示す現行の CPS による価値の事例は，society 5.0 公開資料[15]を参考にした．

例えば，防災に関連するサービスとして，近年，街頭の監視カメラ映像を手がかりに逃走犯を逮捕した事例がある．現在は，カメラ映像データの回収と映像確認を人手で行っているため，発見までに時間を要している．E-CPS では，広範囲に設置された多数所有者の共用カメラからネットワークを介して高速にデータを収集し，さらに，データに応じた解析処理までを自動で行うことで，逃走犯を迅速に発見することが可能になる．さらに，必要なデバイスを一時的にシステムに組み込むことができるため，例えば，逃走犯の進路を塞ぐために，特定の瞬間にだけ信号機や自動ドアを制御することや，巡回中の警備ロボットを派遣するといった，状況に応じたオンデマンドのシステム拡張も可能である．

また，他の例として，農業に関連するサービスでは，周辺区域のセンサを一時利用することで，農地周辺の広範囲リアルタイムセンシングによる農地の天候や気温の予測が行える．さらに，収穫時期等の短期間にのみサービスを利用することができるため安価に導入が行える．そして，栽培，収穫用の機器が汎用化された将来には，それらを周辺センサと迅速に連携させ，農作業を低コストに自動化することが期待される．

このように，E-CPS は，Society 5.0 が対象とする様々な領域において，サービスの低コスト化と迅速な提供，高度化に大きな効用をもたらすものである．

表 1.2 E-CPS による価値向上の事例

ドメイン	現行の CPS による価値の事例	E-CPS による価値向上の事例
交通	・好みに合わせた観光ルートの提供，天気や混雑を考慮した最適な計画 ・自動走行で渋滞なく，事故なく，快適に移動 ・カーシェアや公共交通の組み合わせでスムーズに移動 ・高齢者や障がい者でも自律型車いすを用いて一人で移動	・店舗や街頭に設置されたカメラ映像を利用した，**広範囲，リアルタイム**な渋滞予想 ・自動車，スマートフォン，ウェアラブルデバイス等の**周辺デバイスを迅速に連携**させた移動ナビゲーション
医療・介護	・ロボットによる生活支援・話し相手などによる快適な生活 ・リアルタイムの自動健康診断，病気の早期発見 ・生理・医療データの共有による最適治療 ・医療・介護現場でのロボットによる介護支援	・レンタルカメラや公共カメラを用いた，家族の外出期間に限定した**安価**な要介護家族の見守りサービス

ドメイン	現行の CPS による価値の事例	E-CPS による価値向上の事例
ものづくり	・ニーズに対応したフレキシブルな生産計画・在庫管理 ・AI やロボット活用，工場間連携による生産の効率化，省人化，熟練技術の継承（匠の技のモデル化），多品種少量生産 ・異業種協調配送，トラック隊列走行による物流の効率化 ・特注品を安価，納期遅れなく入手	・レンタルロボットとオープンソフトウェア，設置済みセンサデバイスを組み合わせることによる，短期生産ラインの**低コスト運用**
農業	・ロボットトラクタなどによる農作業の自動化・省力化，ドローンなどによる生育情報の自動収集，天候予測や河川情報に基づく水管理の自動化・最適化などによる超省力・高生産なスマート農業 ・ニーズに合わせた収穫量の設定，天候予測などに併せた最適な作業計画，経験やノウハウの共有，販売先の拡大などを通じた営農計画の策定 ・消費者のニーズに合わせた農作物の自動配送	・特定の季節や天候に限定した**短期契約型の安価**な農業 IoT サービス ・周辺区域の天候など，農地外の**広範囲のリアルタイムセンシング情報を利用**した雨天予測と作業最適化
食品	・アレルギー情報や個人の嗜好に合わせた食品の提案による利便性向上 ・冷蔵庫の食材の自動管理，過不足無い発注・購入による食品ロスの削減 ・家族の嗜好や健康状態などに合わせた料理の提案による快適な食事 ・在庫の最適管理，ニーズに対応した発注による経営改善	・街頭カメラ映像を利用した人流把握による，**リアルタイム**の食品需要予測，移動販売車の派遣，配送ルートの最適化
防災	・個人のスマホに避難情報が提示され，安全に避難所まで移動 ・アシストスーツや救助ロボットにより被災した建物から救助 ・避難所にドローンや自動配送車により救援物資を配送	・災害発生時に，店舗や街頭のカメラ映像を利用した**広範囲，リアルタイム**人物捜索，防災スピーカーを利用したパーソナルな情報配信
エネルギー	・的確な需要予測や気象情報を踏まえた多様なエネルギーの使用による環境負荷低減 ・水素製造や電気自動車（EV)等を活用したエネルギーの地産地消，地域間での融通	・電源車等の移動可能な発電デバイスの最適配備，システムへの組み込みによる，**効率的**なエネルギー運用

1.5.2 E-CPS の要件

本項は，E-CPS の実現に向けて，その要件を整理する．まずは一般的なサイバーフィジカルシステムの要件を述べた上で，共用デバイス利用に関する E-CPS の特徴的要件を示す．

Khaitan 等の調査論文[39]によると，CPS の要件とは，セキュリティ，信頼性，QoS (Quality of Service)，リアルタイム性の 4 つである．それぞれの詳細を以下に述べる．

・セキュリティ （Security）

CPS は，ロボットや自動車といったフィジカル空間に存在するデバイスに対する制御を扱うため，物理的な安全性を確保するために強固なセキュリティが求められる．例えば，Cardenas 等[40]は，CPS に対する攻撃には，フェイク情報の混入，DoS 攻撃，デバイスへの物理攻撃があると述べている．CPS をこのような多数脅威から守るために対策を施すことが必要である．

共用デバイスを利用する場合には，さらに考慮すべきことがある．Roman 等[41]は，共用デバイスの利用における，データ送信元，サービス事業者，情報処理システム等の複数エンティティ間の認証とアクセスコントロール，データ管理の必要性を述べている．また，Atzori 等[42] は，IoT デバイスの認証とデータ一貫性保障の課題について言及している．一般的なコンピュータの認証は，認証用サーバとメッセージを交換することで行われるが，IoT のデバイスのなかには，このようなメッセージを扱うことができないものも存在する．Qiu 等[43]は，性能や機能が均質でないヘテロデバイスを扱う上では，通信プロトコルがセキュリティ上の重要な要素であると述べている．

また，セキュリティに関連して，プライバシー保護も重要な要件である．我々の身の回りのセンサが生成するデータには多くのプライバシー情報が含まれる．プライバシー保護の手段として，例えば，Wickramasuriya 等[44]は，カメラの撮影映像から人物などの特定の情報を抽出して匿名化することを述べている．

・信頼性（Reliability）

多くの CPS は 24 時間稼動することが求められる．天候や気温が変化する屋外などの物理的負荷が高い環境においても，障害を起こさずにシステムを安定稼動できることが求められる．信頼性を高めるための手段は多数あるが，例えば，Abad 等[45]は，分散した要素で構成される電力システムの信頼性を高めることを目的に通信障害への対策を述べている．

共用デバイスで構成されるシステムの信頼性を高めることはさらに難しい．性能が異なるヘテロデバイスを組み合わせて構成されるシステムであり，かつ，十分な事前検証を行う時間が与えられないためである．

・QoS

デバイスやコンピュータを結ぶネットワークの通信品質の担保も CPS の重要な要件である．なぜなら，サービスの実行性能や品質は通信品質に大きく依存するためである．特に，高いリアルタイムを要求するサービスにおいて通信品質の担保は必須である．

共用デバイスを利用する場合には，システムを構成するデバイスやコンピュータが大規模かつ広域なネットワークに分散して配備されているため，それらを結ぶ通信状態を管理することが必要になる．Feng 等[46]は，アプリケーションに応じて様々異なる，通信遅延やパケットロスの許容量に対応するには，システムアーキテクチャ，通信プロトコル，CPU やメモリサイズ，ネットワーク帯域，電力といったリソースの管理が課題になると述べている．また，Balasubramanian 等[47]は，CPS の QoS の実現には，CPU とネットワークリソースを統合的に扱うリソース配備方式が必要であると述べている．

・リアルタイム性 （Real-time-ness）

高いリアルタイム性が要求される CPS は多い．例えば，高速に移動する自動車を対象とする ITS (Intelligent Transportation Systems) では，ミリ秒単位の高速な処理が求められる．その他，イベント処理を扱うシステムにおいても，高いリアルタイム制御が求められるものが多い．処理を高速化する手段として，例えば，Kang [48]等は，システム処理遅延の原因となるデータへのアクセス時間に着目して，高速処理に適したデータベースアーキテクチャを述べている．

共用デバイスで構成されるシステムは，デバイスの性能が様々であり，かつ，ネットワーク上の広域に分散したデバイスを利用するため，高いリアルタイム性を満たすことがより難しい．高いリアルタイム性を満たすには，デバイスとコンピュータの処理性能，ネットワーク距離を考慮したソフトウェア実行環境の選択等を行い，システムを最適化することが必要となる．

以上に示した従来から存在する CPS の要件に加えて，E-CPS を実現するには，さらに以下の 4 つの要件が存在する（図 1.7）．

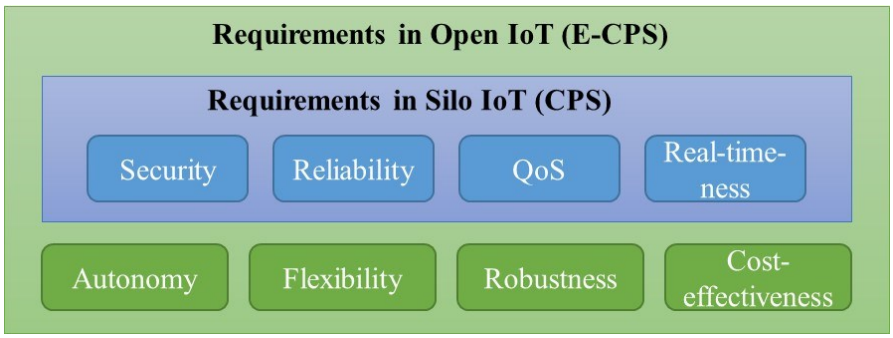


図 1.7 E-CPS の要件

・自律性（Autonomy）

人手による設計や検証を行うことなく自動的にシステムを構成できることが必要である．E-CPS はネットワークに接続された多種多様なデバイスを用いて構成されるシステムであるため，環境やデバイスの組み合わせに応じた設計や検証を全て人手によって行うには，多大な時間と労力がかかる．クラウドコンピューティングのように遠隔からの単純な指示のみで自律的にシステムを構成できることが必要である．

・柔軟性（Flexibility）

ネットワークに接続された多様なデバイスを組み合わせて柔軟にシステムを構築するには，特定のサービスやデバイスのみに有効な機能を可能な限り排除し，汎用性が高い機能で構成される必要がある．例えば，環境内のカメラ映像を用いたドローンの飛行制御サービスでは，カメラとドローンの種類や性能を変更する場合でも，それらを連携させる機能には変更が生じないことが必要である．

・頑強性（Robustness）

フィジカル空間を扱うシステムは環境の変化に追従できることが必要である．従来の CPS は，サービス環境に応じた対策が事前に講じられていた．一方で，E-CPS は，あらゆる環境がサービス対象と成り得るため，事前に全ての事象に備えることは難しく，システムが高い頑強性を備えたものであることが必要である．例えば，音声による道案内を行うサービスでは，騒音発生時には，イヤホンや街頭スピーカーの音量を自動的に大きくすることが求められる．さらに，共用デバイスは，サービス事業者が意図しない移動やネットワーク切断が起こり得る．そのような事態が発生した場合に代替のデバイスを探してシステムに動的に組み込むことが必要である．

・経済性（Cost-effectiveness）

経済性とは，低コストつまり，サービスに必要な最小限のデバイスのみでシステムが運用される必要があることを意味する．スピーカーや照明といったデバイスは，数と出力ボリュームを大きくすれば作用する範囲を広げることは可能であるが，過剰な使用は無駄なエネルギーを消費し，運用コストを高めてしまう．また，デバイスの共用を前提とする E-CPS において，特定のサービスが必要以上の数のデバイスを占有することは，他のサービスの使用機会を喪失することになるため許容されない．サービス実行上の必要最小限のデバイス数と出力ボリュームによって，低コストにシステムを運用できることが必要である．

上記に述べた，E-CPS の特徴的要件に関して，比較のため，クラウドコンピューティングサービスと現行の CPS の充足性，および，E-CPS が目指す姿を表 1.3 に整理した．

表 1.3 E-CPS の特徴的要件と現行システムの要件充足性

○:充足，×:非充足，△:限定的に充足

	クラウドコンピューティングサービス	現行の CPS	E-CPS
自律性	○ Web 申請を受け， 自動でコンピュータや API を提供	×	○ Web 申請を受け， 自動で CPS 機能を提供
柔軟性	○ 多様なコンピュータと API を提供	×	○ 多様なデバイスと 汎用ソフトウェアを提供
頑強性	×	△ 頑強性を有するが 設計，検証が必須	○ 環境変化に自律的に適応
経済性	○ 共用コンピュータを用いたサブスクリプション型の提供により安価	×	○ 共用デバイスを用いたサブスクリプション型の提供により安価

まず，クラウドコンピューティングサービスは，高い自律性と柔軟性，経済性を満たしている．例えば，Web からサービス利用を申請するだけで，コンピューティング環境や API の迅速な利用が可能である．API は汎用的なものが揃えられており，それらを組み合わせることで様々なサービスへの柔軟な対応が可能である．多数ユーザがコンピュータを共用するため，リソースを最大効率で利用でき，安価に提供される．

次に，現行の CPS は，サービスや環境に応じた個別設計，検証を経て構成されることが前提であるため，Web による遠隔操作のみでサービス提供が完結することはない．デバイスやソフトウェアはサービスに固有のものが多く使われるため，汎用性は乏しい．また，頑強性は満たすものの，環境に応じた設計や事前検証を行うことが前提である．デバイスの敷設やシステムの維持管理に費用がかかるとともに，サービスとデバイスが括り付けであるため，常に最大負荷を想定した設備投資が必要であり，総じてコストが高くなり，高価である．

最後に，E-CPS は，クラウドコンピューティングと現行の CPS 双方の要件を有するものである．クラウドコンピューティングのような高い自律性と汎用性が求められるとともに，共用デバイスの利用による高い経済性，さらには現行の CPS 以上の頑強性を要件とする．

1.6 本研究の目的

これまでに述べた背景を踏まえ，本研究は，オープン IoT の到来により増大するネットワーク内の共用デバイスを一時的に利用して，オンデマンドに構成される革新的 CPS である，E-CPS の構成法を確立することを目的とする．

E-CPS は，ネットワークに接続された不特定多数のデバイスを動的に組み合わせるという性質から，現行の CPS に無い要件を有しており，実現には技術課題の解決が必須である．

次章以降に，E-CPS を構成する機能と技術課題，その解決に向けた取り組みを示す．

1.7 本論文の構成

本論文は，6 つの章で構成される．図 1.8 に全体構成を示す．

序論となる本章では，本研究を取り巻く技術的，社会的背景と，本研究の目的を述べた．具体的には，IoT に関する世界的な動向である，デバイス数とネットワークトラフィックの増加と，日本における Society5.0 の取り組み，および，情報通信サービスを取り巻く社会的変化を述べた．これらの背景を元に，IoT の将来予想として，複数のサービス事業者が，ネットワークを介してデータやデバイスを共用するオープン IoT を示し，オープン IoT の恩恵を享受する新たな CPS の形態として，Ephemeral-Cyber-Physical System (E-CPS) を提案した．E-CPS は，ネットワークに分散する共用デバイスを組み合わせて，クラウドコンピューティングサービスのように，オンデマンドかつ安価に提供される CPS である．大規模ネットワークに分散する多種多様なデバイスを組み合わせて構成されるため，現行の IoT やロボットには無い要件を有しており，実現には，新たな構成法が求められる．以上より，革新的な CPS である E-CPS の構成法の確立を本研究の目的とした．

第 2 章では，本研究の前提となる E-CPS の設計指針と構成機能を示し，関連する既存の取り組みに対する分析から，本研究が取り組む領域と技術課題を示す．はじめに E-CPS の設計指針として，多様なサービスとデバイスに対する汎用性を備えたシステム構成を提案する．さらに，構成機能に関して，ネットワークに分散する大量のデバイスの中から適切なものを選択し，制御するという，E-CPS 固有の性質に基づいて既存の取り組みを分析し，問題点を明らかにする．そして，E-CPS の構成機能のうち本研究が扱う領域を定め，本研究が取り組む技術課題として，大規模ネットワークからのデータ収集，ネットワーク内の多種デバイス識別，多様な組み合わせのデバイスの自律制御の 3 つを提示する．

第 3 章では，第 1 の技術課題である「大規模ネットワークからのデータ収集」を取り上げ，広域に分散する大量のセンサが生成するデータの中から必要なものを発見して収集する手法を提案する．本手法は，IoT における問題の一つであるネットワークトラフィックの膨大な発生を防ぎつつ，大規模ネットワークに分散するデータのリアルタイム検索を可能

にする．エッジコンピューティング，クラウドコンピューティングとの机上比較および，ネットワークトラフィック削減効果および，検索時間に対する評価を示す．

第 4 章では，第 2 の技術課題である「ネットワーク内の多種デバイス識別」を取り上げ，通信情報からデバイスの機種を識別する手法を提案する．本手法は，データ収集，通信特徴量抽出，類似度算出という三段階の処理で構成され，デバイスの種類やネットワーク環境に応じて，処理ごとの拡張が可能である．提案手法の評価は，実用化を目指す観点から，提案手法に関する基本的な検証とフィージビリティ確認のための予備実験および，識別性能の評価実験，処理性能の測定まで実施した．

第 5 章では，第 3 の技術課題である「多様な組み合わせのデバイスの自律制御」を取り上げ，ネットワークに接続された多様なデバイスを，サービス目的に応じて自律制御する手法を提案する．E-CPS における，多様な種類と設置条件のデバイスを扱う上での高い複雑性という問題点に対して，機械学習に基づく手法を採用する．提案手法は，機械学習を利用することで，デバイスやソフトウェアの組み合わせを意識した人手によるシステム設計と検証を不要にする．シミュレーションと実機を用いた実験により，複数のセンサとアクチュエータが混在する環境において，提案手法が，サービス目的に応じたデバイスの自律制御に有効であることを示す．

最後に，第 6 章では，本研究の結論を述べる．はじめに本研究の成果と E-CPS の要件に照らし合わせた技術的な到達点を述べ，さらに，今後の展望として，発展に向けた課題と注目すべき技術領域，本研究成果の社会導入に向けた指針を示す．

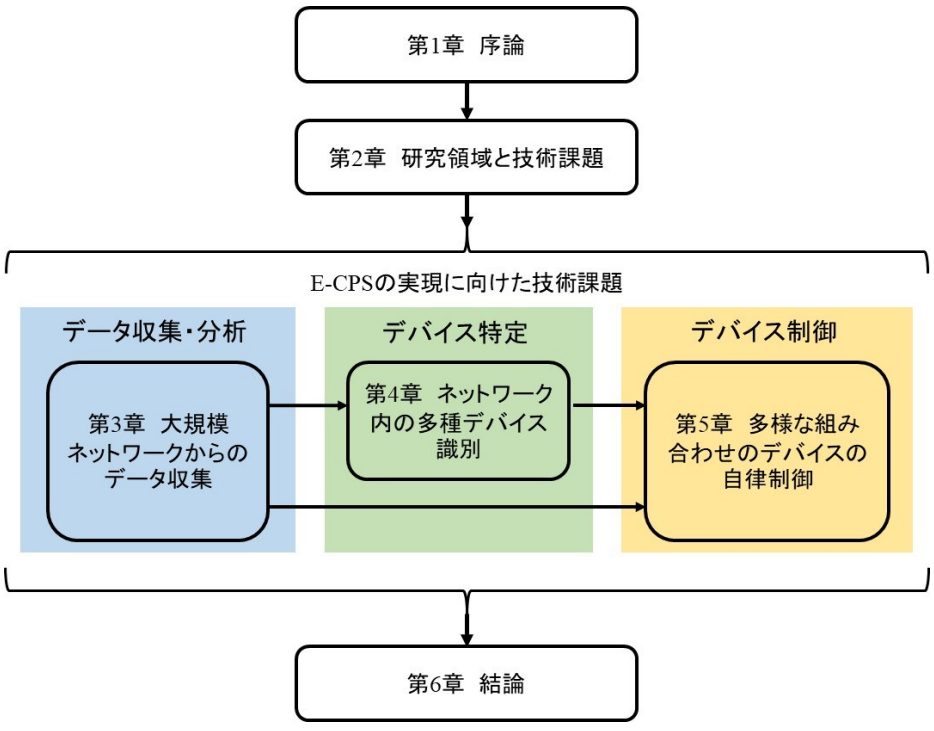


図 1.8 本論文の構成

第2章 研究領域と技術課題

2.1 はじめに

本章では，E-CPS の設計指針となるシステム構成と，その構成機能，および，本研究が取り組む技術課題を示す．はじめに E-CPS のシステム構成を提案し，その後，構成機能ごとに関連する既存の取り組みを分析する．最後に，E-CPS の構成機能のうち，本研究が取り組む領域と技術課題を述べる．

2.2 E-CPS のシステム構成

第 1 章で述べた E-CPS の要件－ Security，Reliability，QoS，Real-time-ness，Autonomy，Flexibility，Robustness，Cost-effectiveness －に従い，本研究は，サービス事業者に対してデバイスを意識させずに構成される，高い自律性を備えたシステムの実現を目指す．また，日々増加する多様なサービスとデバイスへの柔軟な対応も必須とする．

これら要件を満たすために，E-CPS の設計指針として，図 2.1 に示す構成を提案する．本構成は，サービス事業者に対して，抽象化された機能や抽象的なサービス目的を指定するだけでシステム構成指示が可能なインタフェースを提供する．また，デバイス固有のコマンドやデータ形式を変換するアダプタを備えることで，多様なデバイスの相互接続を可能にする．そして，デバイスやデータの選択・制御に関わる汎用機能と，サービスとデバイスの種類に依存する固有機能とを分離することで，増加するサービスとデバイスに対する拡張性を満足する．

以上のシステム構成のもと，さらに，システム構成要求を受けてからデバイスの制御に至るまでの一連の処理を，扱う情報の具体性から三つのレイヤ― コンテキストレイヤ，プランニングレイヤ，コントロールレイヤ ―に分割した．

最も抽象的な情報を扱い，サービス事業者に対するインタフェースを持つのが，コンテキストレイヤである．本レイヤは，サービス事業者からのシステム構成要求を受け付け，要求に対応するシステムの構成要素を特定する．サービスごとに必要なデバイスの種類やソフトウェアはサービスカタログと呼ぶ統一の形式で定義して管理される．例えば，火災通知サービスを例に挙げると，火災映像を撮影するカメラデバイス，映像から火災を判断するソフトウェア，火災発生を通知するサイネージやスマートフォンがサービスの構成要素として定義され，登録される．サービスの事業者や利用者は，本サービスカタログの単位でサービスの開始，つまり E-CPS の構成を指示する．本処理の段階では，サービスを構成するデバ

イスの種類だけが指定される．

二番目に抽象的な情報を扱うのがプランニングレイヤである．本レイヤは，ネットワークに分散する大量のデバイスの中から，適切なものを選択し，その組み合わせに応じた適切な制御値を定める．このような機能は，E-CPS に固有なものであり，その実現に向けて特に重要なものである．したがって本研究では，特に本レイヤを構成する機能を基幹機能と呼ぶ．火災通知サービスの例では，火災検知用の画像データを収集するカメラと，避難対象地域のサイネージや通行人のスマートフォンといったデバイスが，サービス事業者が指定せずとも状況に応じて自動的に選択され，適切に制御される．また，本レイヤでは，以降の章に詳細を述べるデータ分析アプリケーションを用いて処理を行う．データ分析アプリケーションとは，センサデータを分析し，データの性質やサービスの実行状態を定量的に評価するソフトウェアである．基幹機能と独立して本ソフトウェアを充実させることで多様なサービスとデバイスに対応する．

最も具体的な情報を扱い，デバイスを直接操作するのがコントロールレイヤである．本レイヤは，デバイスの種類や機種に対する柔軟性，拡張性を保障するための機能として，デバイス固有のコマンド体系や，データ形式を，デバイスに依らない統一的なものに変換する機能を持つ．デバイスの変更や追加による影響を本機能に閉じて対応することで，基幹機能の汎用性を保つことができ，システム全体として高い柔軟性を備えることができる．

さらに，これらレイヤの横断的な処理として，コンピュータやセンサ，アクチュエータといったリソースの使用状況を管理するリソース管理機能を独立した機能として設ける．

以上の機能を用いて，サービス事業者からシステム構成要求を受けてシステムを構成，運用するまでの一連の処理を自動化する．次節より，基幹機能に含まれない周辺機能および，基幹機能に関連する既存の取り組み事例を順に述べる．

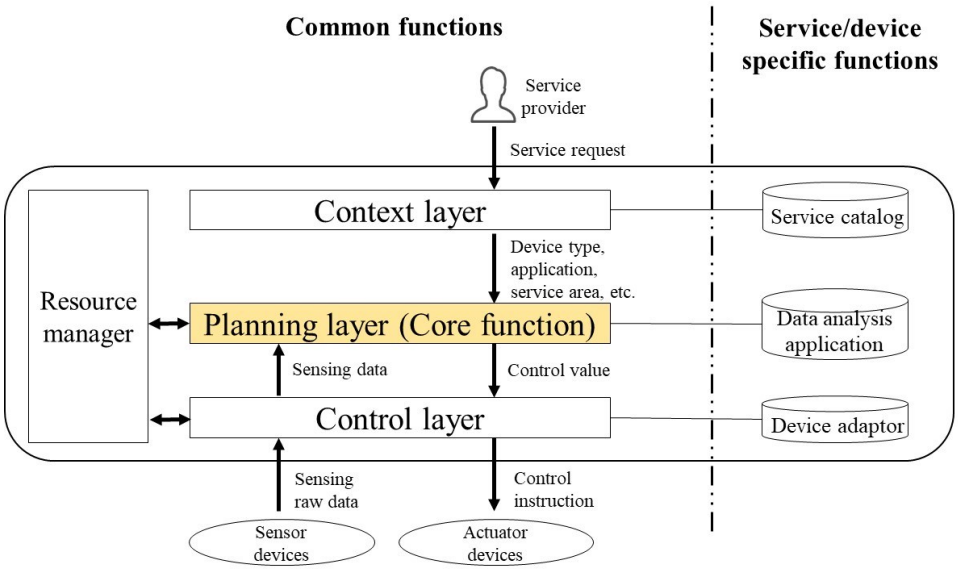


図 2.1　E-CPS のシステム構成

（©IEEE　主著論文 1 の Fig. 2 を一部修正）

2.3 周辺機能に関する既存の取り組み

本節では，E-CPS の基幹機能に含まれない周辺機能に関する既存の取り組み事例を述べる．はじめに，リソース管理に関する事例を述べ，次にソフトウェアやデバイスの相互接続に関する事例を述べる．

2.3.1 リソース管理に関する既存の取り組み

共用リソースを用いて多数のサービスを同時実行するには，実行環境のコンピュータやネットワークといったリソースを管理することが必要である．リソース管理に関する有力な取り組みとして，1.4.3 項に述べた，クラウドコンピューティングの構成技術がある．OpenStack は，大量のコンピュータとその上に構築される仮想コンピュータの管理に有用な機能を備えている．例えば，flavor と呼ばれるフォーマットで仮想コンピュータに割り当てる CPU コア数やメモリサイズを指定することが可能である．また，コンピュータの残存リソースを可視化する機能や，今後のサービス要求に備えて特定のリソースを予約する機能も提供している．さらに，サービスごとの論理的なネットワークを自動構築する機能も提供している．OpenStack を利用することで，インフラとして必要なリソースの確保とシステム構築までの一連の処理を全て自動化することが可能である．OpenStack を始めとするクラウドコンピューティングのリソース管理技術は，現在も盛んに研究開発が進められており，本研究が目指す E-CPS の実現においても活用が期待される．

2.3.2 ソフトウェアやデバイスの相互接続に関する既存の取り組み

デバイスごとに固有なコマンド体系や API，プロトコルの差異を吸収して，ソフトウェア間，デバイス間の相互接続を実現するために，いくつかの取り組みが進められている．

Web of Things (WoT) [49]は，既存の Web 技術やプロトコルを利用して，ソフトウェアやデバイスを組み合わせた IoT サービスの提供を目指すコンセプトである．Web の標準化団体である W3C [50]によって標準化が進められている．WoT を具現化するための研究として，デバイス間の相互接続性を担保するゲートウェイ[51]や，ネットワーク内のデバイスを Web ベースのツールで可視化，検索するシステム[52]が研究されている．

また，OneM2M [53]，AllJoyn [54]，GotAPI [55]，ECHONET [56]といった，デバイス製造者やサービス事業者が参画する標準化の取り組みがある．これらの標準は，デバイス操作に必要なインタフェースや，通知情報のフォーマットを規定しており，ネットワーク管理者によるデバイスの一元管理や，ソフトウェアの再利用性向上に有用である．しかしながら，現状は，団体ごとに標準化が進められており，デバイス製造者ごとに準拠する標準が異なるため，真に統一的な標準が存在しない．したがって，当面は，各デバイスの相互接続には，仲

介用のソフトウェアやゲートウェイ装置を設ける必要がある．標準化が十分に進んだ将来には，デバイスの相互接続には特別な機能は不要になると考えている．

2.4 基幹機能に関する既存の取り組みと課題分析

本節では，ネットワークに分散する大量のデバイスの中から適切なものを選択し，制御するという E-CPS の基幹機能に関する技術課題を分析する．はじめに，一般的な CPS の処理の流れに沿って，データ収集・分析と，デバイス制御に分けて問題を抽出する．

まず，データ収集・分析では，大規模ネットワークに分散する膨大なセンサが生成するデータの中から，必要なデータをリアルタイムかつ低コストに発見して収集することが課題である．オープン IoT においては，利用可能なセンサとデータの候補が膨大であり，かつ，データがリアルタイムに生成され続けるため，従来の CPS や Web 検索のデータ収集手法では対応できない．

次に，デバイス制御では，多種多様なデバイスの中から制御対象のデバイスを特定したうえで，環境変化に対する頑強性を備えつつ，低コストに制御を行うことが求められる．様々な持ち主のデバイスが共用されるオープン IoT においては，必ずしも全てのデバイスの仕様が登録されていない．適切な制御対象デバイスを選択するためには，まず，どのような種類や機種のデバイスがネットワークに接続されているかを把握することが必要である．また，制御対象のデバイスを選択した上で，それらを適切に制御することも必要である．例えば，複数のスピーカーを用いて行う火災通知サービスでは，スピーカーと聴衆の位置関係や，スピーカーの性能に応じた適切な音量を設定することが求められる．また，別の例として，環境内に任意配置されたカメラ映像を利用したドローン制御では，設置場所が異なる複数のカメラ映像を用いたドローンの自己位置把握，飛行制御が求められる．

以上の通り，E-CPS の基幹機能には，オープン IoT の共用デバイスを用いるという性質上，センサデータ収集，デバイス特定，デバイス制御という 3 つの問題がある．これらの問題について，関連する既存の取り組みから導かれる技術課題を次項より順に述べる．

2.4.1 センサデータ収集に関する既存の取り組みと技術課題

一般的な IoT のフレームワークは，デバイスの仕様や設置場所を手動で登録して一覧化した上で，データ収集を行うデバイスをサービス事業者が指定する方法をとる．例えば，IoT の標準フレームワークである OneM2M は，リソース管理機能として，デバイスの種類や場所を情報として登録し，管理者が参照できるようにしている．同様に，標準化プロトコルの一つである Core [57] [58]は，デバイスにメタ情報を付与して管理する機能を有する．標準フレームワークの一つである AllJoyn は，マルチキャストドメインネームシステムである DNS-base [59]を扱っており，特定の属性のデバイスを対象にした検索が行える．また，Web 検索

システムである UDDI (Universal Description, Discovery and Integration) [60]もデバイスに付与したメタ情報を用いたデバイス管理を可能としている．

しかしながら，これらの手法は，E-CPS のデータ収集には適していない．E-CPS のデータ収集には，共用デバイスとして公開されている監視カメラや通行人のスマートフォンといったデバイスがリアルタイムに発信する膨大な総数のデータの中から必要なものだけを収集することが求められる．例えば，特定人物の映像を使用するサービスにおいては，カメラデバイス自体を指定する需要は少ない．対象人物を写しているカメラは，人物やデバイスの移動によって常に変化するため，“対象人物を写しているカメラ”，もしくは，“対象人物の映像”といったコンテキスト（意味情報）で指定できることが重要である．つまり，E-CPS のデータ収集には，大規模なネットワークに存在する多数のデバイス全てが候補であるがゆえに，そのものを指定するのではなく，コンテキストでデバイスやデータを指定することが求められる．

また，収集するデータの性質としては，カメラ映像などのリアルタイムストリーミングデータが主になることが想定される．主流の Web 検索エンジン[61] [62] [63] [64]の多くは，大規模にデータを収集して大量のインデックスを作成する仕組みをとるため，このようなデータをリアルタイムに扱うことには適していない．

以上より，E-CPS が求める大規模ネットワークからのデータ収集には，既存の IoT，CPS のデータ収集や Web 検索とは異なる新たな手法が求められる．

2.4.2 デバイス特定に関する既存の取り組みと技術課題

E-CPSを構成するIoTデバイス（以下，単に「デバイス」という）には，従来のコンピュータ機器とは異なる三つの特徴がある．一つ目は多種多様性である．カメラや温度センサ，スピーカー，ディスプレイなど様々なデバイスがあり，ハードウェア性能や使用する通信プロトコルが異なる．二つ目は，モバイルデバイスが多いという移動性である．代表的なモバイルデバイスであるスマートフォンや，スマートウォッチなどのウェアラブルデバイスは所有者の移動によって接続先のネットワークが動的に変化する．三つ目は，デバイス数の爆発的増加による膨大性である．第1章で述べた通り，ネットワークにつながるデバイスの数は200億を超えるため，これらを人手で登録して管理することは現実的でない．以上より，デバイスを特定するには，多種多様なデバイスへ適用可能かつ，複数ネットワークの移動にも対応可能な，自動化された手法が求められる．

ネットワーク接続されたデバイスの性質を把握する技術がある．UPnP (Universal Plug and Play) [65]は，デバイスが同一ネットワーク上の他の機器に対して，機種名や製造元といった自身の情報を通知するプロトコルである．デバイスの種類を把握して管理するのに有用なプロトコルであるが，全てのデバイスが対応しているわけではない．また，ディスクリプションとしてデバイスの情報が記述されるが，その記述方法には規定がなく，製造会社ごとに表記が異なるため，個別の解釈が必要である．同様に，SNMP [66]と NETCONF [67] [68]もネッ

トワークに接続されているデバイスを把握するためのプロトコルであるが，やはり全てのデバイスが対応しているわけではない．

また，前節に示した，OneM2MやCoreといったデバイス管理の機能を有する標準フレームワークもデバイスの特定に有用なものではあるが，現状，それらの標準に対応しているデバイスが限定的であること，また，ローカルネットワークごとの手動での登録が必要であることを鑑みるに，E-CPSのデバイス特定の要件を満たしていない．

さらに，デバイスの処理の特性や挙動から，種類や機種，個体を推測する技術がある．例えば，Web finger print を利用するアプローチがある．Web finger print とは，Web ページのレンダリング処理の実行時間，表示特性，ページの遷移傾向といった，アプリケーションやユーザに固有のブラウザの挙動である．Web Finger print を用いて，デバイスのハードウェア仕様を高精度に特定する研究例[69][70]がある．しかしながら，ブラウザを操作できるデバイスに適用が限られるため，単純な機能しか持たないセンサやネットワークカメラなどの多くの IoT デバイスには適用できない．また，Hardware finger print を利用するアプローチ[71]がある．Hardware finger print とは，デバイスのハードウェアに依存する特徴である．例えば，無線デバイスのハードウェアの微細な個体差に由来する無線強度の違いや，カメラのレンズの歪みから生じる映像の歪みなどが該当する．一般的に，これらの Hardware finger print の検出には特殊な計測装置が必要，もしくは，検出可能な環境が限られるため，やはり膨大かつ多様な環境で用いられるデバイスの特定には適さない．

以上の通り，E-CPS のデバイス特定に求められる，性能や機能が異なる多種デバイスへの対応，複数ネットワークを跨った管理，自動化という全ての要件に対応する実用段階の技術はなく，特定に必要となるデバイスの性質を識別する新たな手法が必要である．

2.4.3 デバイス制御に関する既存の取り組みと技術課題

IoT サービスシステムやロボットなどの複数のセンサとアクチュエータから構成されるシステムの制御には様々な手法がある．最も一般的なのは，大まかな入出力の情報の関係規則を設計した上で，シミュレーションや実機検証によって，細かいパラメータを設定する手法である．例えば，移動ロボットの制御では，ロボットに搭載されているカメラを用いてリアルタイムに取得する映像を入力として，適切な移動経路を算出し，最終的に車輪を可動するモータの制御値を出力する．計算過程としては，カメラ映像を元にした自己位置の算出し，目的の移動量に対応するモータの回転量を算出する．この場合には，まず，カメラと車輪の位置関係，各計算処理の情報の流れと計算式を設計したうえで，計算処理遅延や車輪の摩擦等の実測値を計測して制御値の修正を行う．

また，フィジカル空間を扱うシステムは，出力結果が環境に左右される．例えば，暖房機器は目標の室内温度が同じ場合にも，外気温に応じて適切な出力が異なる．車輪を持つ移動ロボットでは，床面の摩擦に応じて同じ回転量を満たすためのモータ出力が異なる．このような環境差分への頑強性を実現するためにフィードバック制御が多く用いられる．フィー

ドバック制御とは，ある制御結果における目標値と実際の値とを比較し，差分を元に制御値の調整を行う手法である．指標とする物理量はモータの回転角や電流量などシステムによって様々ではあるが，機械，電気システムにおける基本的な手法である．以上が，フィジカル空間を扱うシステムの一般的な制御方法である．

1.4.3 項に示した，IoT サービスやロボットのフレームワークは，このような制御則の設計を支援する機能を持つ．例えば，Node-RED や，Ren-v は，複数のデバイスや機能間のデータ入出力の設計を視覚的に行える GUI を提供している．また，ROS は，ハードウェア形状や物理特性をシミュレーションするツールを提供しており，開発者が実機を製作する前に様々な条件でシステムを検証することができる．

しかしながら，これらの IoT サービスやロボットのフレームワークでは，E-CPS の構築には対応できない．E-CPS は，ネットワークに接続された共用デバイスを一時的に利用して構成するシステムである．デバイスは所有者の意志でネットワークへ接続もしくは，切断されるため，サービス提供空間に設置されているデバイスの数や種類は頻繁に変化する．つまり，同じサービスを行う場合にも，都度，機種や数，設置位置が異なるデバイスを組み合わせなければならない．これは，固定されたデバイスで構成される現行の IoT サービス，ロボットシステムとは大きく異なる要件である．システム構成要素の変動性の観点から，古典的なロボットと，汎用ロボット，E-CPS の比較を表 2.1 に示す．まず，古典的なロボットは，特定のハードウェアと特定のソフトウェアが一体となって構成される．システムの変動要因はなく，工場から出荷されたそのままの状態で使用される．次に，製造現場での普及が進んでいるアームロボットなどの汎用ロボットは，ハードウェアは固定されたものであるが，ソフトウェアを用途に応じて変更，調整できる．使用環境や具体的な把持物体などに応じて細部の設定を投入してから使用する．最後に，E-CPS のハードウェアは複数のデバイスで構成され，また，ソフトウェアはサービスに応じて選択される．その組み合わせは膨大であるため，事前に全ての組み合わせに対するシステム設計や設定は困難である．

以上より，E-CPS のデバイス制御には，システムを構成するデバイスやソフトウェアの組み合わせが膨大かつ変化するなか，事前の設計や設定無しに，自律的にデバイスを制御する新たな手法が求められる．

表 2.1 システム構成要素のバリエーション

	従来型ロボット （掃除ロボット等）	汎用ロボット （アームロボット， ヒューマノイド）	E-CPS
ハードウェア	固定	固定	多様なデバイスで構成
ソフトウェア	固定	サービスに応じて変動	サービスに応じて変動
タスク	固定	多様	多様
通信環境	内部通信 （組み込み機器）	内部通信 （組み込み機器）	外部通信 （遅延，ジッタが変動）
デバイス間距離	固定 （組み込み機器）	固定 （組み込み機器）	変動
インタフェース （コマンド，API）	固定	外部インタフェースは ソフトウェアに応じて変動 内部インタフェースは ハードウェアごとに固定	外部インタフェース，内部 インタフェース共に変動

2.5 本研究の取り組み領域と技術課題

ここまでに述べた通り，E-CPS の実現に向けて，大規模ネットワークからのデータ収集，多種デバイスの識別，多様なデバイスの組み合わせに対する自律制御が重要な技術課題である．図 2.2 に現行の CPS との比較を改めて示す．本研究は，E-CPS の基幹機能に関するこれらの技術課題を扱う．また，サービス定義やデバイスの相互接続といった基幹機能に含まれない周辺機能に関しては，本章に述べた既存の取り組みに対する調査，分析の通り，標準化等の取り組みが既に進められていること，および，現状でも費用をかければ既存技術の範疇で実現可能なことから，本研究の対象からは除外した．

また，本研究は，E-CPS の基幹機能を，ルータやサーバといったネットワーク設備に実装することを想定したアプローチをとる．本アプローチには複数の利点がある．まず，デバイスに対するソフトウェアのインストールを最低限とすることでデバイスの種類や性能に寄らずに適応が可能である．これは，多種多様な IoT デバイスを扱う上で大きな利点である．また，デバイスの共用に必要な機能をネットワーク設備が担うことで，デバイス製造者や所有者をデバイスの設定にかかる負担から解放できる．理想的には，デバイスをネットワーク

に接続するだけで自動的に共用デバイスとして公開されることを目指す．最後に，通信遅延や通信帯域の観点からも，デバイスとコンピュータ間の物理的距離は短いことが望ましく，デバイスに物理的に近いネットワーク設備を各機能の実行環境として用いることには大きな利点がある．以上に述べた本研究の対象領域と取り組む技術課題の俯瞰を図 2.3 に示す．

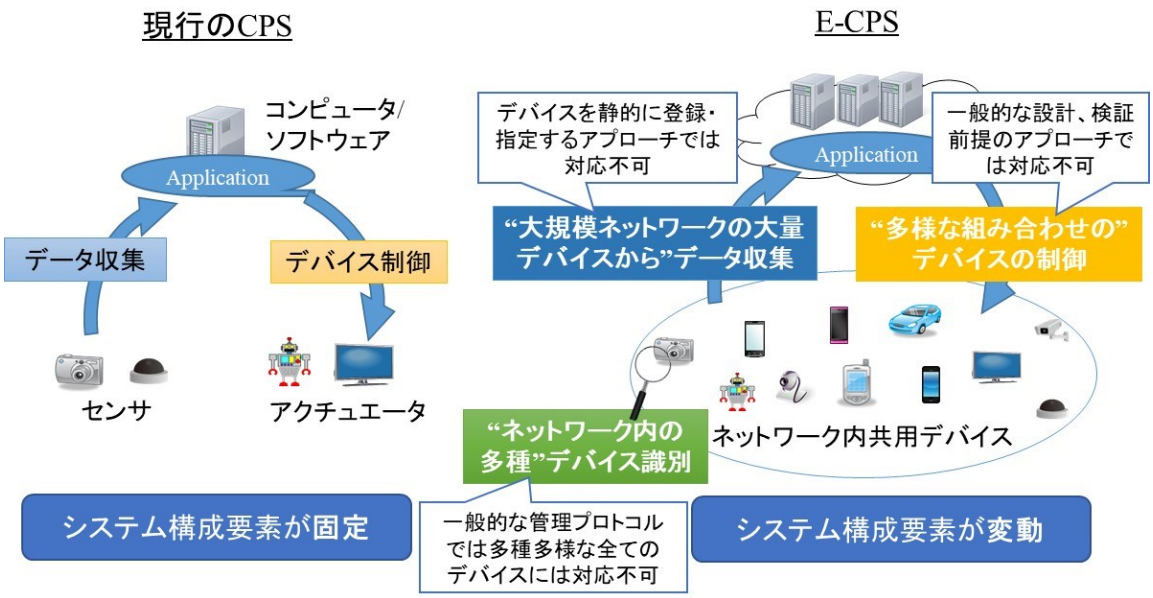


図 2.2　現行の CPS と E-CPS の機能要件比較

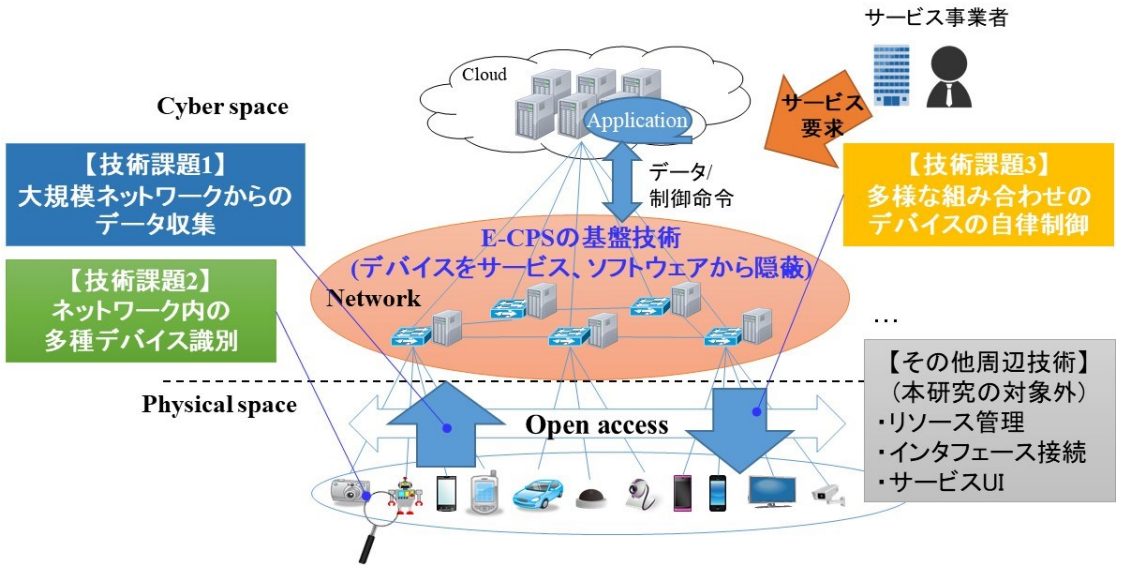


図 2.3　本研究の対象領域と取り組む技術課題

2.6 第 2 章のまとめ

本章では，本研究が取り扱う領域と技術課題を明確にするために，E-CPS のシステム構成と機能を示し，さらに，関連する既存の取り組みの分析から技術課題を導出した．

E-CPS のシステム構成として，デバイスやデータの選択・制御に関わる汎用機能と，サービスとデバイスの種類に依存する固有情報とを分離し，また，デバイス制御に至るまでの処理を，扱う情報の性質から，コンテキスト・プランニング・コントロールの三つのレイヤに分割した構成を示した．そして，プランニングレイヤに関して，ネットワークに分散する大量のデバイスからサービスに応じた適切なものを選択して制御するという E-CPS の特徴に基づいて既存の取り組みを分析し，本研究が取り組む技術課題を以下に設定した．

本研究は，大規模ネットワークからのデータ収集，ネットワーク内の多種デバイス識別，多様な組み合わせのデバイスの自律制御の 3 つの技術課題に取り組む．

以降，第 3 章から第 5 章にかけて，各技術課題に対する取り組みを順に示す．

第3章　大規模ネットワークからのデータ収集

3.1 はじめに

本章では，E-CPS の実現に向けた技術課題の一つである，大規模ネットワークからのデータ収集を取り上げ，広域に分散する大量のセンサが生成するデータから，必要なデータをリアルタイムに発見，収集する手法を提案する．はじめに，E-CPS のデータ収集に関する詳細要件を整理した上で，本研究のアプローチを示す．そして，データ収集に関する提案手法を示した後に， IoT の既存アーキテクチャとの机上比較および，実験システムによる評価を示し，最後に考察を述べる．

3.2 要件とアプローチ

E-CPS のデータ収集には，センサを指定するのではなく，必要なデータの意味情報（コンテキスト）を指定して収集することが求められる．本研究では，ネットワークに接続されるセンサがリアルタイムに生成するデータをライブデータと名付け，E-CPS におけるライブデータ収集要件を以下に整理した．

要件 1：ライブデータの高速な発見

E-CPS のサービスにおいてデータの鮮度は重要な要素である．リアルタイム性を要求するサービスにおいて古いデータの価値は低い．例えば，動く物体を映像で追跡するサービスでは，追跡対象を映している瞬間の映像には価値があるが，対象が通り過ぎた後の過去の映像には価値がない．ライブデータをその価値が失われる前に，発見できることが求められる．

要件 2：多様なコンテキストへの対応

様々なサービスの実現には，収集対象データを多様なコンテキストで指定できることが求められる．例えば，特定人物の捜索サービスでは，捜索対象の顔を指定する場合もあれば，捜索対象の衣服の色を指定する場合もある．また，自動車の捜索においても同様に，車の色や形状といった指定を行う場合もあれば，ナンバープレートの数字を指定する場合もある．このような，サービスごとに異なる様々なコンテキストに対応したデータ収集を行える必要がある．

要件 3：多様なデータ形式への対応

センサデータには，カメラが生成する画像データやストリーミングデータ，温度センサや距離センサが生成するテキスト形式のデータなど，様々な形式が混在する．これら多様なデータ形式に柔軟に対応した検索が行えることが求められる．

以上の要件を満たすデータ収集を行う上で，さらに考慮すべき問題がある．ライブデータを扱う上で最も大きな問題の一つは，膨大なデータ量に伴う通信コストである．ライブデータの多くはカメラ映像などの大容量ストリーミングデータと予想される．さらに，第 1 章で述べたように，IoT デバイスの数は増加の一途をたどっているため，ネットワークにつながる全てのセンサが生成するデータの総量は膨大なものになると予想される． 2020 年時点におけるインターネット回線サービスの通信帯域の主流は，マスユーザ向けサービスでは 1 Gbps であり，一部法人向けサービスでも 10 Gbps である．既存のネットワークインフラにおいて，全てのライブデータをクラウドコンピューティング環境に集約して処理することが不可能であることは明白である．したがって，ライブデータの取り扱いには，ネットワークインフラの制約を考慮し，必要最小限のデータのみを指定して収集することが必須である．

以上を踏まえ，大規模ネットワークからのデータ収集には，ネットワークに分散する大量のライブデータに対する検索が必須である．以降の節では，ライブデータ検索に関する関連研究，および提案手法を示す．

3.3 ライブデータ検索の関連研究

ライブデータ検索に関連する先行研究を述べる．要件 2 に対応して，リアルタイムに生成されるセンサデータやセンサをコンテキストで指定して発見する先行研究がある．Elahi 等[72]は，環境内に設置されたセンサを対象に，特定のコンテキストに合致するデータの発生を予測する手法を提案している．しかしながら，正確性は保障されておらず，また，周期的に発生するデータのみに有効な手法である．Perera 等[73]は，コンテキストによるセンサ検索の手法を提案しているが，事前にデバイスの属性情報を定義して登録することを前提としており，多様なコンテキストに対応するためのスケール性を有さない．

また，Semantic Web は，デバイスをコンテキスト情報で定義することによって，検索と連携を可能にする技術である．例えば，Pfisterer 等[74]や Kotis 等[75]は，IP アドレスではなく種類や設置場所といった属性情報を用いてデバイスを検索するシステムを提案している．Semantic Web は，異なる定義情報の関係性をオントロジーから自動補完して，デバイスの相互接続性の獲得を目指したものであり，Web of Things を具現化する技術としても期待されている．しかしながら，前述の研究例と同様に，デバイスの属性情報を事前に定義して登録することが必要であり，要件 1 の高速なデータ発見および，要件 2 の多様なコンテキスト

への対応を満たさない.

また，コンテキストによるデータ検索をネットワーク機能によって実現する情報指向型ネットワーク[76] [77] [78]と呼ばれる研究分野がある．これは，ICN (Information Centric Network)，もしくは，CCN (Contents Centric Network) と呼ばれる．インターネットを含む既存のデータ通信は，IP ルーティングによって実現されている．接続先のコンピュータとデータの所在を IP アドレス，URL (Uniform Resource Locator) という形で指定すると，ネットワークの中継装置は，その情報をもとに自律分散的にデータまでの通信経路を探索する．情報指向型ネットワークは，このような既存のルーティングの枠組みを変革するものである．情報指向型ネットワークでは，所在情報の代わりにコンテンツの名前を指定して，ネットワーク内のデータにアクセスする．所在という，それ自体に意味を持たない情報ではなく，コンテキストを用いた直接的な情報を用いたルーティングを行うことで，所在とコンテキストを対応付ける機能を別途に用意する必要がなくなり，ネットワークの自律分散特性と相乗して，効率的なデータ管理と通信が行える．しかしながら，実装には既存のネットワーク装置を全て刷新することが必要であり，普及には長期の時間を要する．また，ソフトウェア処理を前提とする仕組みであるため，従来の通信よりも計算負荷と時間を要することから，要件 1 に対応する，大規模ネットワークにおける，大量データのリアルタイム処理に対して懸念が残る.

以上のように，ライブデータに対する柔軟な検索を扱う先行研究はあるものの，そのまま E-CPS のデータ収集の全要件を満たすものはない.

本研究は，広域に分散する大量のセンサが生成するライブデータを検索する手法として，具体化した三つの要件を満足する手法を提案する．そして，提案手法は，ライブデータを扱う上で生じる膨大なデータ通信コストを最小化するためにネットワーク分散型のアーキテクチャをとる．一般的に，分散型アーキテクチャは集約型アーキテクチャと比較して，集約効果が得られない分だけ処理効率が劣るというデメリットがあるが，提案手法は次節に示す構成をとることで，トレードオフを解決する.

3.4 ライブデータ検索に関する提案手法

ライブデータ検索の要件を満たし，かつ，膨大な通信コストをともなわない，ネットワーク分散型ライブデータ検索手法を提案する. 図 3.1 は, 提案手法の全体像である. 本手法は, ライブデータバッファとリソースリゾルバという二つの機能で構成される.

ライブデータバッファは広域に分散するローカルネットワークに具備される機能であり，同一ネットワーク内のセンサが生成するデータを一定時間蓄積する．そして，ライブデータバッファは，センサデータに対する分析・加工処理を行い，ローカルネットワークをつなぐ広域のネットワークに大容量のデータが転送されることを防ぐ．図 3.2 は，ライブデータバッファにおける処理の詳細を示したものである．ネットワークに接続された全ての共用デ

バイスは，同一ネットワークのライブデータバッファにデータを送信し，それらのデータは一定時間ライブデータバッファに蓄積される．ライブデータバッファは，一定期間が経過したデータを随時削除し，一定のデータ蓄積容量を維持する．

そして，ライブデータバッファは，蓄積されたデータに対する分析処理を行う．多様なコンテキストで収集データを指定するために，様々なデータ分析アプリケーションをライブデータバッファ上で実行する．一方で，コンテキストに対応するデータ分析アプリケーションは，サービスの種類数だけ存在するため，常に想定される全てのデータ分析アプリケーションをライブデータバッファ上で実行することはコンピュータ性能の観点から不可能である．したがって，本手法は，データ検索を行うわずかな時間にだけ，データ分析アプリケーションをライブデータバッファに配信して実行する形態をとる．以降，動的に配信する分析アプリケーションをクエリフィルタと呼ぶ．

次に，提案手法のもう一つの機能であるリソースリゾルバは，データ検索要求を受け付け，対応するクエリフィルタをライブデータバッファに配信する機能である．リソースリゾルバは，ローカルネットワークに存在するライブデータバッファの一覧情報を持ち，ライブデータバッファの負荷や過去のデータ傾向から配信先を限定する．例えば，既に多数のクエリフィルタが実行中であり高負荷になっているライブデータバッファを配信先から除外する，もしくは，過去に同様のコンテキストのデータを発見したライブデータバッファを選択するといった判断を行う．また，リソースリゾルバは，検索要求をクエリフィルタと結びつけるクエリトランスレータと呼ぶ機能と連携する．クエリトランスレータは，検索コンテキストとデータ分析アプリケーションの対応関係を記した辞書データを管理する．

なお，クエリフィルタは，ダウンロードアプリケーション，VM (Virtual Machine) [79], Linux container [80]等の既存技術を用いて実装することが可能である．本手法は，ソフトウェアの原本をクラウドコンピューティング等で集約管理できるため，ソフトウェアアップデートに関する利点もある．高度なデータ分析を行う AI ソフトウェアの開発スピードは目覚ましいため，ローカルネットワークの大量の総数のコンピュータのソフトウェアバージョンを意識することなく，常に最新バージョンを適用できることは管理稼働の大幅な削減になる．

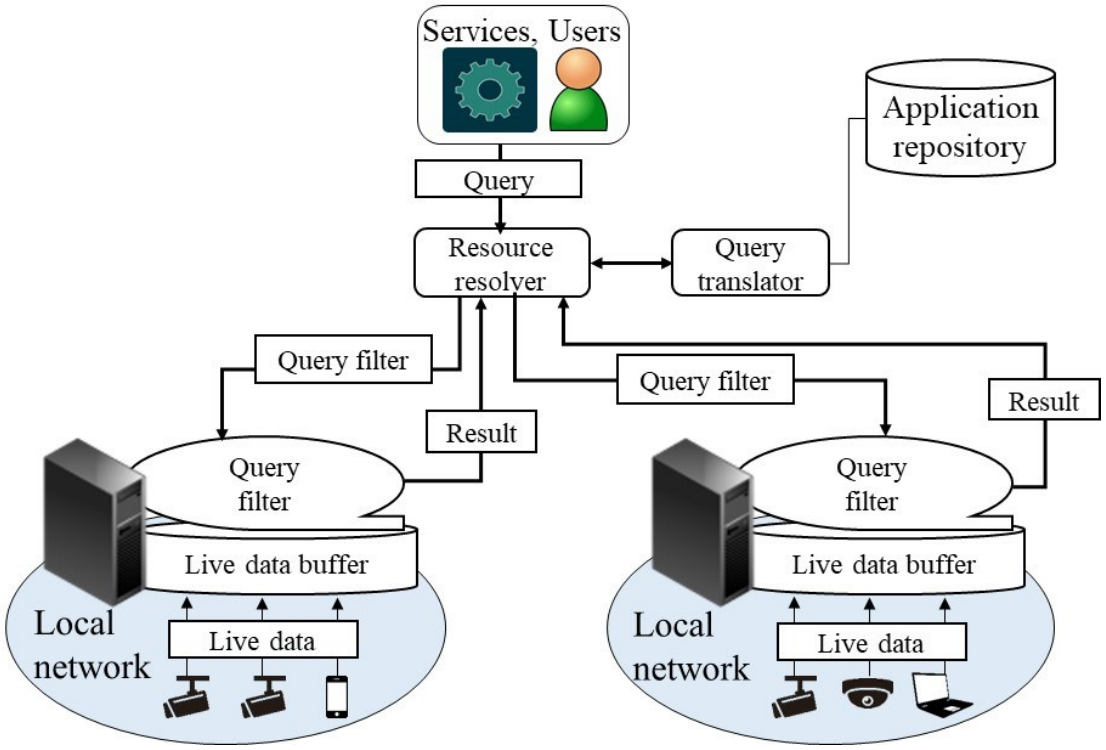


図 3.1 ライブデータ検索手法

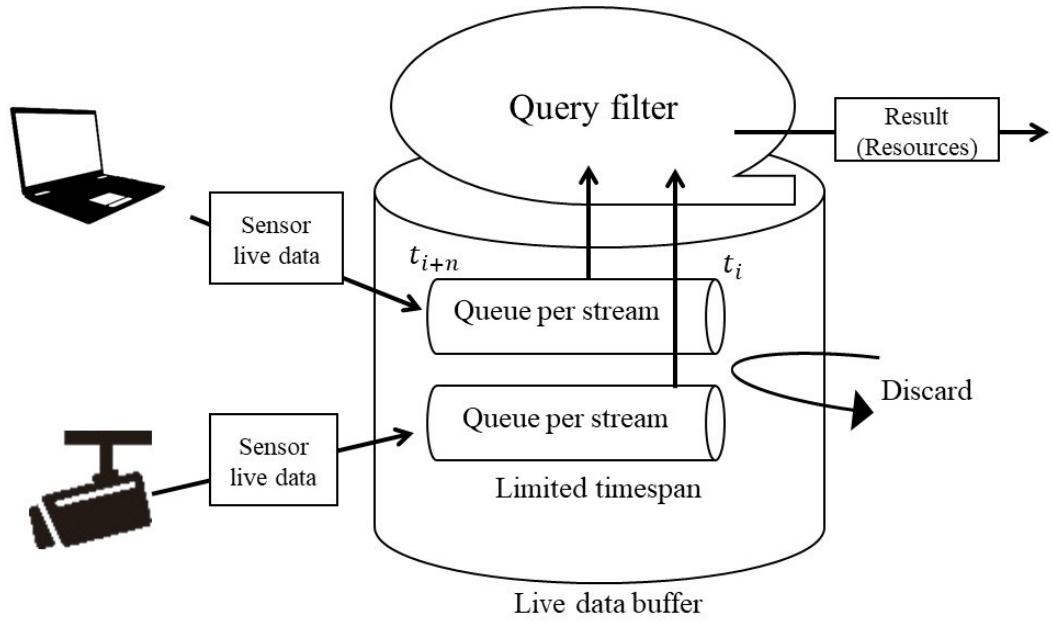


図 3.2 ライブデータバッファの構造
(©IEEE 主著論文 3 の Fig.3 を一部修正)

3.5 提案手法の評価

提案手法と既存のIoTアーキテクチャとの比較および，実験システムを用いた評価により提案手法の有効性を示す.

3.5.1 既存 IoT アーキテクチャとの机上比較

提案するライブデータ検索手法を既存の IoT アーキテクチャと比較した．比較対象は，最も普及しているクラウドコンピューティング，および，近年注目が集まっているエッジコンピューティング[81] [82]とした．エッジコンピューティングとは，デバイスの物理的近傍のコンピュータを用いて，データ分析やデバイス制御を行うアーキテクチャである．デバイスとコンピュータ間の通信遅延が小さくなること，大容量データを広域ネットワークに転送させないという効果がある．本比較におけるエッジコンピューティングとは，アプリケーションインストール済みの一定数のコンピュータを，ローカルネットワークに配備した運用モデルを指す．表 3.1 に比較結果を示す.

まず，広域ネットワークトラフィックについて，提案手法とエッジコンピューティングは共にデバイスのデータをローカルネットワーク内で処理するため，クラウドコンピューティングと比較して小さくなる.

検索条件への柔軟性として，提案手法は，随時必要なアプリケーションをダウンロードする形態をとるため多様な検索条件に対する拡張性を有している．一方，エッジコンピューティングは，特定のアプリケーションを事前にインストールしておくことを前提としており，対応できる検索条件は，そのアプリケーションが扱えるものに限定される．例えば，顔画像を分析するアプリケーションをインストールしている場合には，顔画像に関する検索条件は扱えるものの，映像内の不審行動の検知や音声分析といった異なるデータ分析に対応する検索条件は扱えない．IoT サービスの多様化により，データ検索条件と，対応する分析アプリケーション数の増加が予想される．したがって，大量のローカルネットワークにある全てのコンピュータリソースを，常にあらゆるアプリケーションがインストールされた状態に保つには，アプリケーション転送にかかる通信負担および，管理負担が大きいため，一定数のアプリケーションのみを備えた状態で運用を行うことになる．また，クラウドコンピューティングは，豊富なコンピューティングリソースとアプリケーションを備えているため，提案手法と同様に多様な検索条件への対応が可能である.

以上のように，提案手法は，多様なコンテキストによる柔軟な検索という要件を満たしつつ，ネットワークトラフィックも小さい．一方で，いくつかトレードオフとして生じる問題がある.

一つ目の問題点は，処理性能の保証が難しいことである．ライブデータバッファはローカルネットワーク内の共有コンピュータに実装されるため，同時に複数のクエリフィルタが

実行される．検索要求に応じて様々な組み合わせのクエリフィルタが同時実行されるため，事前に全ての組み合わせの処理負荷を検証することは困難である．対称的に，エッジコンピューティングは，事前にインストールされたアプリケーションのみを実行するため，最大負荷を事前に推定した設備設計が行える．また，クラウドコンピューティングはコンピュータリソースの物理的距離の制約を考慮しないため，そもそもコンピュータリソースをスケールアウトすることが可能であり，リソースの不足が生じない．提案手法に関する本問題は，OpenStack などのリソース管理ツールを適用することで解決が可能と考えている．例えば，OpenStack は，flavor と呼ばれるフォーマットによって VM が使用する CPU コアの数量とメモリサイズを指定できる．さらに，OpenStack は，KVM (Kernel-based Virtual Machine） [83] の CPU ピニング機能と連携することで VM を特定の CPU コアに割り当てることも可能である．このようなツールを適用することで，複数のデータ分析アプリケーションを単一のコンピュータで同時実行する場合にも，それぞれの処理性能を独立して担保することが可能であると考えている．

二つ目の問題点は，汎用コンピュータの使用を前提としつつ，高い性能要求への対応も必要なことである．ライブデータバッファは様々なデータとクエリフィルタを処理するため，特定のアプリケーションに特化したコンピュータを用意することは経済的に非効率である．例えば，映像データの分析処理には，GPU (Graphics Processing Unit) が適しているが，全てのローカルネットワークに高価な GPU サーバを配備するには大きな費用がかかる．したがって，特定のソフトウェアに特化したコンピュータではなく，汎用的なコンピュータの配備が基本となる．一方で，AI 等の高度なデータ分析アプリケーションの実行には，GPU や FPGA (Field Programmable Gate Array) といった特殊なハードウェアの使用が不可欠である．本問題の解決手段として，GPU サーバや大容量ストレージサーバといった複数種類のコンピュータリソースを少数用意し，需要から導かれる効率的なネットワーク集約点に配備することが考えられる．

三つ目の問題点は，クエリフィルタの配信にかかる通信量と時間である．提案手法は検索要求ごとにライブデータバッファにクエリフィルタを配信するため，検索要求数に比例した量のダウンロードトラフィックが発生する．また，ソフトウェアダウンロードに要する時間が検索の所要時間に加わる．これら通信に関わる影響は次項に示す実験により示す．

表 3.1 提案手法の机上比較

〇:優位，×:不利

	クラウドコンピューティング	エッジコンピューティング	提案手法
ネットワークトラフィック	×	〇	〇
検索条件の柔軟性	〇	×	〇
多重処理に対する性能保障	〇 （潤沢リソース）	〇 （最大負荷の事前見積り）	× （ローカルネットワークの共用コンピュータ使用）
ハードウェア依存処理に関する性能保証	〇 （特殊コンピュータも使用）	× （汎用コンピュータを使用）	× （汎用コンピュータを使用）
懸念要素	−	−	クエリフィルタの配信に伴うネットワークトラフィックと遅延

3.5.2 実験システムの仕様

提案手法における，ネットワークトラフィックと検索時間を評価するために，実験システムを開発した．本システムは，複数ネットワークを含む広範囲リアルタイム人物捜索サービスを模擬したものであり，評価用機能の他に，E-CPSによるデータ収集サービスをPoC (Proof of Concept) として具現化した機能を有する．

A. 実験環境

実験環境を図3.3に示す．外観に特徴がある移動オブジェクトを複数のカメラによって撮影し，リアルタイムに生成する撮影映像データの中から検索要求に対応するものをシステムが自動的に選択する．実験環境の詳細は以下の通りである．

・顔写真，色，数字が表面に描かれた自律移動オブジェクト（図3.4)が周回移動する．
・一定間隔に設置されたカメラの撮影映像を常時ライブデータバッファが取得する．
・システムは2つのローカルネットワークと1つの管理ネットワークで構成される．
・ローカルネットワークには，ライブデータバッファとカメラが接続される．
・管理ネットワークには，リソースリゾルバと後述するサービスポータルが接続される．

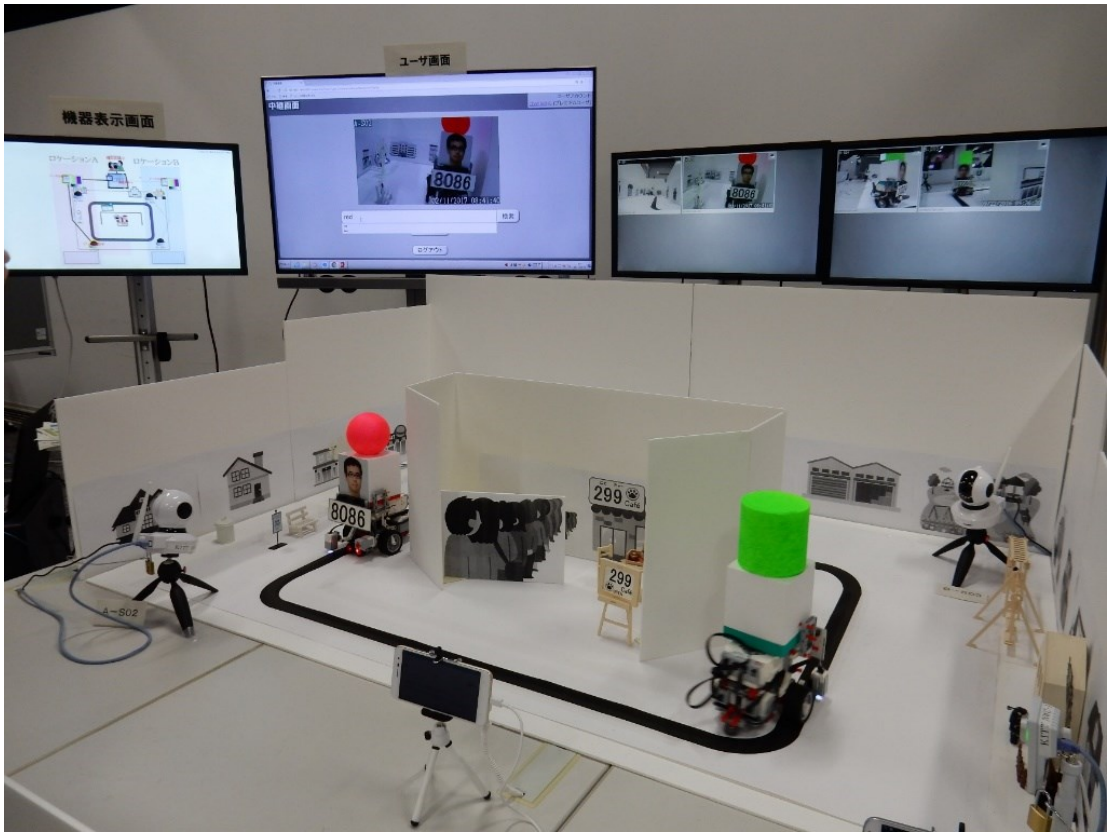


図 3.3　実験環境
(©IEEE　主著論文 3 より引用)

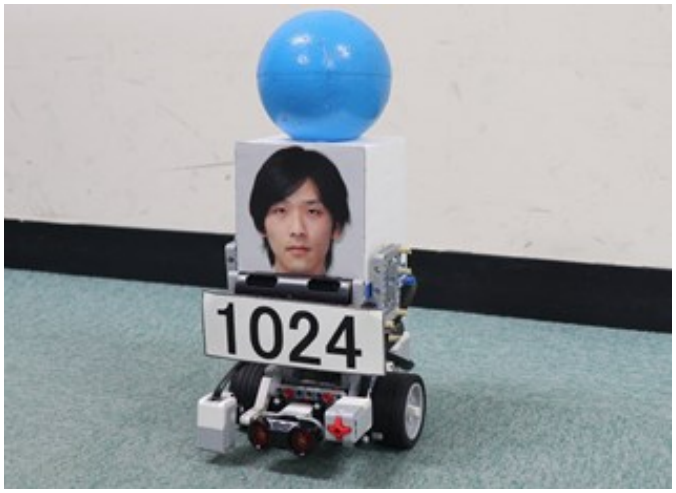


図 3.4　捜索対象の自律移動オブジェクト

B. 実験システム構成

図3.5は実験システムの構成を示したものである．実験システムは，3.4節に示した提案手法を構成する機能の他に，検索条件を受け付けるサービスポータルを備える．以下に実験システムを構成する機能を示す．

・サービスポータル

サービスポータルは，図3.6に示すGUIを介して，数字，名前，色といった検索条件をサービス利用者から受け取り，リソースリゾルバへ検索を指示する．また，サービスポータルは，後述する機能部の検索結果を受け，検索条件に合致するカメラにアクセスし，映像をリアルタイムに表示する．複数のカメラが目的のオブジェクトを同時に撮影している場合には，最も鮮明な（後述するデータ確信度が高い状態を指す）カメラ映像を表示する．検索処理は，終了指示を与えられるまで継続するため，サービスポータルに表示される映像の生成元カメラは，オブジェクトの移動に応じて変化する．

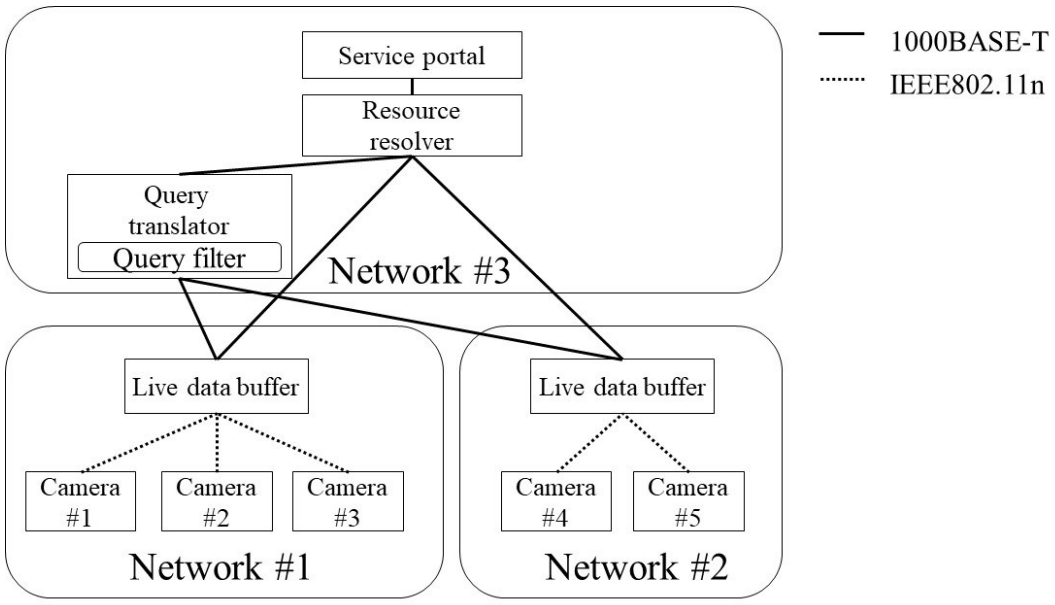


図 3.5　実験システムの構成

(©IEEE　主著論文 3 の Fig.4 を一部修正)

図 3.6　サービスポータル画面
(©IEEE　主著論文 3 より引用)

・リソースリゾルバ

リソースリゾルバは，サービスポータルからの検索指示を受けると，クエリトランスレータによって対応するクエリフィルタを選択し，2つのローカルネットワーク上に配備したライブデータバッファに配信する．ライブデータバッファのデータ分析結果を受け，後述するデータ確信度が最も高い画像を生成したカメラを特定し，そのIPアドレスをサービスポータルに出力する．

・クエリフィルタ

クエリフィルタは，画像から顔認識，色認識，文字認識，および形状認識を行うアプリケーションをまとめて1つのパッケージとしたものであり，ファイルサイズは115 MBである．本ソフトウェアは画像分析の結果として，データ確信度と呼ばれる，画像認識結果の確からしさを示す定量値を出力する．データ確信度は，認識モデルとの差異が小さいほど大きいものとなり，例えば，顔認識においては，正面画像を認識モデルとしたため，正面から全体を撮影した画像ほどデータ確信度が高くなる．

・ライブデータバッファ

ライブデータバッファは，同一ネットワーク内のカメラから1台当たり200ミリ秒周期で9.4 kBの画像を取得する．また，ライブデータバッファは，画像に対してクエリフィルタを適用し，検索条件に合致する画像を発見次第，その生成元カメラのIPアドレスとデータ確信度をリソースリゾルバに通知する．

・ネットワーク

カメラとライブデータバッファ間はIEEE802.11nの無線通信, その他は全て1000BASE-Tの有線通信である.

なお, 実験システムのコンピュータ仕様は表3.2に示す通りである. 本実験では, ライブデータバッファ1つ当たりに2台のコンピュータを使用した. 1台はクエリフィルタ処理を実行し, もう1台はデバイスや他機能との接続等の処理を実行した. また, ライブデータバッファは物理コンピュータ上で実行し, その他の機能は全てVM上で実行した.

表 3.2 実験システムのコンピュータ仕様

Module		CPU (Virtual Machine)	Memory (Virtual Machine)
Service portal		Intel Core i5-6200U CPU @2.30GHz × 2 (1 core)	(2 GB)
Resource resolver		Intel Core i7-6700 @3.40GHz × 8 (2 core)	(12 GB)
Query translator		Intel Core i7-6700 @3.40GHz × 8 (1 core)	(4 GB)
Live data buffer	Query filter	Intel Xeon CPU E5-2620 v3 @2.40GHz × 12	32 GB
	Other	Intel Core i7-6700 CPU @3.40GHz × 8	16 GB

3.5.3 実験システムによる評価

前項で述べた実験システムを用いて, 提案手法のフィージビリティを二つの観点から評価した. 一つ目は, ライブデータをローカルネットワークに閉じて処理することによるネットワークトラフィックの削減効果, 二つ目は, クエリフィルタの配信による検索時間への影響である.

A. ネットワークトラフィックの削減効果

提案手法によるネットワークトラフィックの削減効果を明らかにするために, 実験システムのネットワークトラフィックを測定した. 2つのライブデータバッファを用いた3分間の処理における, ライブデータバッファとリソースリゾルバ間のデータ転送量の累積値を同時使用するカメラ数ごとに測定した. 測定は同一条件で2セット実施し, その平均値を算出した.

図3.7に測定結果を示す. 測定結果より, カメラ数が1の場合に, ライブデータバッファとリソースリゾルバ間の平均ネットワークトラフィックは15 kB/sと算出される. これは, 提案手法における広域ネットワークトラフィックに該当する.

前述の通り, 実験システムでは, ライブデータバッファには, カメラ1台当たり200ミリ秒周期で9.4 kBの画像データが送信される. つまり, カメラとライブデータバッファ間のネットワ

ークトラフィックはカメラ1台当たり47 kB/sである．なお，この数値は，通信パケットを構成するための各種ヘッダによるオーバヘッドを含めないものである．仮に，クラウドコンピューティングを採用し，全画像データを集約して分析するならば，最低でもこれだけのネットワークトラフィックが広域ネットワークに発生することになる．

つまり，提案手法を採用することにより，クラウドコンピューティングと比較して広域ネットワークトラフィックを68%削減できる．なお，本実験システムは，カメラ映像を200ミリ秒周期で画像に変換して取得する形態をとったが，多くのカメラは映像を秒間10～30フレームのストリーミングデータとして送信するため，実際の映像サービスでは，画像圧縮を考慮しても，さらに大量のネットワーク帯域を消費するため，実環境における提案手法の効用はさらに大きいと予想される．

また，カメラ数に応じてネットワークトラフィックが増加する傾向が見られたが，実験システムのクエリフィルタは複数センサデータの分析結果を集約してリソースリゾルバへ送信するため，単純にカメラ数に対する等倍の増加にはならないことが実験結果から確認された．

さらに，データ分析に必要なクエリフィルタの配信によるネットワークトラフィックの増加分を含めて提案手法が優位となる条件を明らかにした．図3.8は，実験システムにおいて，1台のカメラ映像を扱う際の広域ネットワークのデータ転送量の時間累積について，クラウドコンピューティングと提案手法を比較したものである．データ検索要求が60分（3,600秒）以上継続する場合に，提案手法の広域ネットワークのデータ転送量の累積がクラウドコンピューティングを下回り優位であると言える．

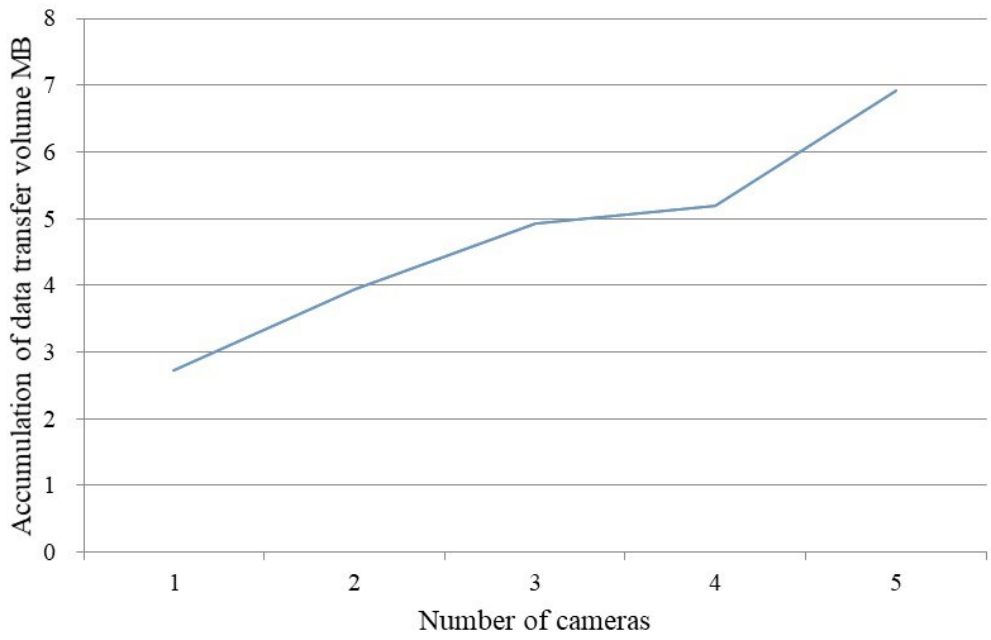


図 3.7　カメラ台数と広域ネットワークのデータ転送量の関係（3 分間の累積）

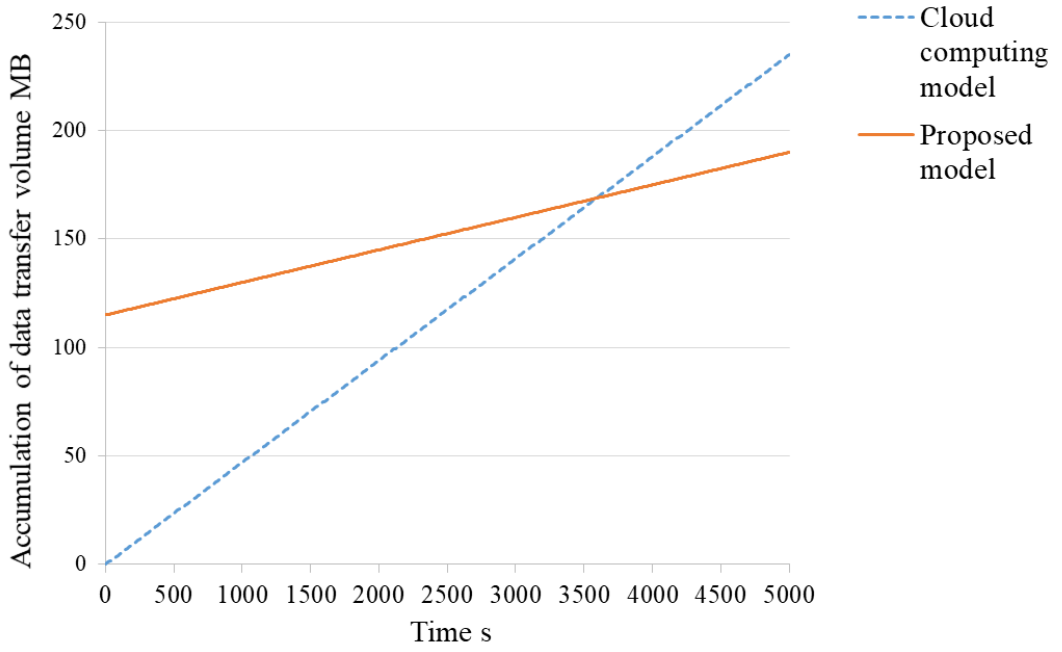


図 3.8 広域ネットワークのデータ転送量の時間累積

このように，提案手法が優位となる損益分岐点は，クエリフィルタの種類および，データ分析対象デバイスの種類と数によって定まる．具体的には，クエリフィルタのデータサイズD (B)，デバイスの送信トラフィックp (B/s)，クエリフィルタの送信トラフィックq (B/s)，デバイス数nを用いて，損益分岐点となるデータ検索要求時間T (s) は以下の式で表される．

$$T = \frac{D}{n(p-q)} \qquad (3.1)$$

例えば，本実験と同一のカメラとクエリフィルタを用いる検索では，損益分岐点となるデータ検索要求時間は，カメラ数に依存し，カメラ数が2の場合には約30分，4の場合には約15分となる．E-CPSは，ネットワークに接続された多数デバイスの使用を前提としており，カメラのような大容量データを扱うデバイスを少なくとも10台以上同時使用することが想定されるため，短時間のサービスを実行するうえでも提案手法の有効性は明瞭である．

なお，デバイスの送信トラフィックが小さいほど，損益分岐点となるデータ検索要求時間は長くなる．E-CPSのサービスは，そのコンセプトから，短期間に提供されるものが多く想定されるため，温度センサなどの送信トラフィックが小さいデバイスについては損益分岐点に到達する前にサービスが終了する可能性がある．したがって，予想されるサービス継続時間と損益分岐点を元に，提案手法とクラウドコンピューティングを選択して使い分けることが必要である．

B. 検索時間

提案手法の検索時間がリアルタイムサービスに許容されるものであるかを明らかにするために，リソースリゾルバが検索要求を受信してから，条件に合致するカメラのIPアドレスを送信するまでの時間を測定した．本実験は，各ネットワークに1つのライブデータバッファと1つのカメラが接続された環境で行った．ローカルネットワークの数は，1つの場合と2つの場合についてそれぞれ実験を行った．同一の条件下で5回測定を行い，中央値を算出した．表3.3に結果を示す．

ローカルネットワークが2つの場合に検索時間は7.86秒であった．人の平均歩行速度を80 m/分（1.33 m/s）と仮定すると，検索完了までに約10 mの距離を移動することになる．カメラの視野角に依存するところはあるが，歩行者を追跡するうえでは十分な検索時間と言える．また，検索要求受付からクエリフィルタ配信までに要した時間は，処理全体の合計時間の約80%を占めていた．クエリフィルタのサイズを最適化することにより短縮されると期待できる．さらに，検索要求受付からクエリフィルタ配信までに要した時間は，ローカルネットワーク（ライブデータバッファ）数による違いがみられた．この原因が，コンピュータ処理負荷によるものか，ネットワーク負荷によるものかを判定するために，クエリフィルタを事前配信し，配信処理を省略した条件で実験を行った．表3.4に結果を示す．配信処理を行わない場合にも，ローカルネットワーク（ライブデータバッファ）数が多い方が処理時間は長い結果となったが，配信を行う場合と比較して，その差は大幅に小さい．以上より，ローカルネットワーク（ライブデータバッファ）数増加による処理時間増加の主な原因は，クエリフィルタ配信に伴うネットワーク負荷の増大によるものと考えられる．

なお，配信済みのクエリフィルタを使用する同一条件の検索では，表3.4に示した通り検索時間が大幅に短縮され，ローカルネットワーク（ライブデータバッファ）数が2であり，かつ，配信済みのクエリフィルタを使用する場合には，検索時間は2.55秒と算出される．以上より，同一条件の検索が頻繁に発生する環境における，歩行者の約3倍の速さと推定される走者や自転車の追跡への可能性が示された．

表 3.3　検索時間の実測値

Sequence	Time (s)	
	1 live data buffer	2 live data buffers
Receive query and distribute query filter	5.32	6.36
Execute query filter	1.04	1.04
Returns the address of the device	0.463	0.463
Total	6.82	7.86

表 3.4 ライブデータバッファ数ごとの処理時間

Sequence	Time (s)			
	Download		Pre-download	
	1 buffer	2 buffers	1 buffer	2 buffers
Receive query and distribute query filter	5.32	6.36	0.923	1.05

3.6 考察と今後の課題

机上比較と実験システムを用いた評価により，広域に分散する共用デバイスからのデータ収集における提案手法の有効性を評価した．

机上比較では，ネットワークの物理配置を意識しないクラウドコンピューティングおよび，ネットワークの物理配置を意識し，デバイス近傍のコンピュータでデータ処理を行うエッジコンピューティングとの比較を行った．エッジコンピューティングの詳細な定義は業界ごとに異なっており，例えばデバイス近傍のコンピュータとして同一ネットワーク内の端末を指す場合もあれば，近傍の通信事業者の設備を指す場合もある．さらに，後者の場合には，通信事業者が同設備を維持管理する運用形態を含めてエッジコンピューティングと呼ぶ場合がある．本評価では，このような通信事業者が人手で管理する設備を用いるエッジコンピューティングを比較対象とした．さらに，エッジコンピューティングの提供サービスは，管理者の判断でコンピュータへ導入され，人手を介して一定の頻度で更改されること，および，多数ある全ての設備に対してあらゆるアプリケーションを配備することは通信費と管理稼働の観点から行わないことを前提とした．以上の前提のうえで，クエリフィルタと呼ぶデータ分析アプリケーションの配信を行う提案手法の有効性を示した．本手法は，エッジコンピューティングと相反するものではなく，エッジコンピューティングの構想に対して，オンデマンド，自動化の観点を加えて具体化したものである．

実験システムを用いた評価では，ローカルネットワークに配備されたコンピュータに対してクエリフィルタを配信，実行する場合のネットワークトラフィックと検索時間を明らかにした．本実験では，ローカルネットワークのコンピュータは，OS (Operating System) が標準インストールされた状態とし，データ分析アプリケーションの実行ファイルをクエリフィルタとして配信した．これは，ローカルネットワークのコンピュータには汎用機能のみを残し，検索条件に対する固有機能は配信によって付与するという本手法の特徴を表す構成である．提案手法が，広域ネットワークトラフィックを削減し，また，デメリットとして生じる処理遅延がリアルタイム性を要求するサービスを阻害しない程度に収まっていることを確認した．

以上の評価の通り，提案手法の有効性を示した．今後，提案手法を実用システムとして展開する上での課題を以下に示す．

まず，クエリフィルタを効率的に配信する必要がある．インターネットなどの多数のネットワークドメインから構成されるネットワークに提案手法を適用する場合，需要が少ないパーソナルなクエリフィルタを全てのライブデータバッファに配信すると，ダウンロードコストとライブデータバッファ処理コストが大きくなり効率が悪いことから，クエリフィルタを配信するライブデータバッファを限定する必要がある．アプローチの一つとして，ライブデータバッファのメタ情報の利用がある．例えば，特定の個人を追跡するサービスの場合，国または地域といった地理的な情報からデータ収集候補のライブデータバッファを事前に絞り込むことができる．このようなメタ情報と検索条件を対応させることで，目的データの発生確率が高いライブデータバッファのみを選択してクエリフィルタを配信する．さらに，図3.9に示すように，過去に収集したデータの統計や検索履歴からライブデータバッファのメタ情報を自動的に生成することを考えている．例えば，検索履歴を元にしてライブデータバッファごとに固有なデータ発生傾向を動的に把握し，メタ情報として登録することで，後続の類似する条件の検索に対して，データ発見確率が高いライブデータバッファを選択することが可能になる．

また，クエリフィルタ自体の改善も課題である．複数サービスのデータ検索要求をまとめて処理することで，ライブデータバッファのデータ分析処理を効率的に行うことができる．アプローチの一つとして，クエリフィルタの階層化が考えられる．移動オブジェクトの検出といった汎用的なフィルタと，特定人物の顔の識別といった特定用途に特化したフィルタを設け，前者のフィルタは，複数サービスの検索要求に対して共通的に実行し，そこで検出されたデータのみを後者のフィルタにかける．この二階層化でクエリフィルタのダウンロードコストとデータ処理コストを削減することが可能である．

最後に，提案手法は，セキュリティおよびプライバシー対策への応用が考えられる．リソースリゾルバを介することで，サービス事業者がデバイスへ直接アクセスすることを防止できる．また，クエリフィルタとしては，データの匿名化，暗号化，マルウェア検出などのセキュリティソフトウェアを適用することも可能である．ローカルネットワーク内のライブデータバッファ上でこれらのセキュリティ対策を行うことで，広域ネットワーク内に機密情報が無防備に流通することを防止できる．そして，要求に応じてソフトウェアを動的に配信するメリットとして，常に最新バージョンのソフトウェアを適用できることに加えて，特定の特徴データのみの匿名化など，個人ごとに調整された条件のソフトウェアを迅速に適用することが可能である．

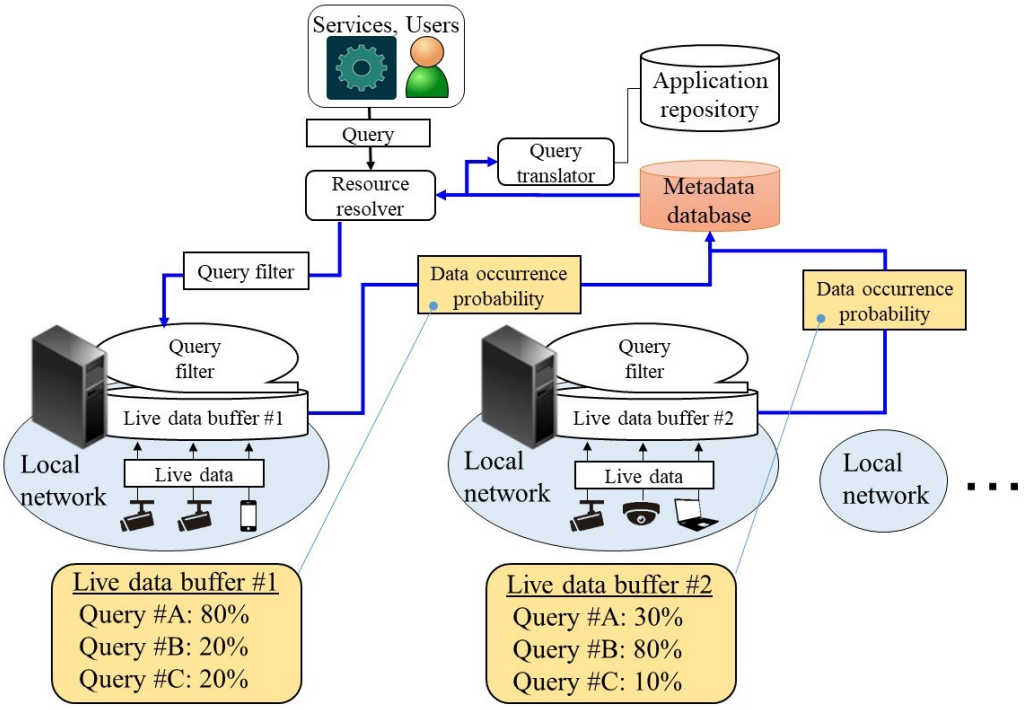


図 3.9 自動生成メタ情報によるデータ検索範囲限定の概略

3.7 第 3 章のまとめ

本章は，E-CPS の技術課題の一つである，大規模ネットワークからのデータ収集について，ネットワーク分散型ライブデータ検索手法を提案した．広域のネットワークに分散するデバイスがリアルタイムに生成するデータをライブデータと名付け，その収集要件として，高速な発見，多様なコンテキストへの対応，多様なデータ形式への対応を挙げた．

提案手法は，データ収集時に問題となる膨大なネットワークトラフィックを解消しつつ，これらの要件を満たすデータ収集を実現する．具体的には，検索要求に対応するデータ分析アプリケーションをローカルネットワークに動的に配信することで，柔軟な検索と広域ネットワークトラフィック削減を実現する．

クラウドコンピューティング，エッジコンピューティングとの机上比較，および，リアルタイム人物捜索サービスを模擬した実験システムによる，ネットワークトラフィック削減効果と検索時間から提案手法の有効性を示した．

最後に，ネットワークトラフィック削減とデータ検索処理効率化に向けた今後の課題として，データ分析アプリケーションを配信するライブデータバッファの限定とクエリフィルタの階層化を挙げた．

第4章　ネットワーク内の多種デバイス識別

4.1 はじめに

本章では，E-CPS の実現に向けた技術課題の一つである，ネットワーク内の多種デバイス識別を取り上げ，通信情報を分析することでデバイスの種類や機種を識別する手法を提案する．はじめに，E-CPS におけるデバイス識別の詳細な要件を整理したうえで，本研究のアプローチを示す．そして，デバイス識別手法の詳細を示した後に，試作システムを用いた評価実験結果と処理性能測定結果を示し，最後に手法改善に向けた考察を述べる．

4.2 要件とアプローチ

E-CPSが求めるデバイス識別には，性能や機能が異なる多様なデバイスへ汎用的に対応可能であり，かつ，接続先ネットワークが頻繁に変化するモバイルデバイスに対応できる手法が求められる．デバイス識別の詳細な要件を以下に示す．

・要件1：デバイスの種類，機種の識別

サービスの実行に必要なデバイスをネットワーク内から特定し，デバイスに応じた適切な制御を行うために，デバイスの種類や機種を識別することが必要である．本研究におけるデバイスの種類とは，カメラやマイク，スピーカーといったデバイスの機能ごとの分類を指す．また，デバイスの機種とは，メーカーや型番で指定される製品単位の分類を指す．

・要件2：デバイスに対して非侵襲

性能や機能が異なる多様なデバイスに対応するために，デバイスへの識別用ソフトウェアのインストールを必須としない手法であることが求められる．また，識別用の処理がデバイス本来の処理に影響を与えないことも求められる．

・要件3：ネットワーク設備やプロトコルに非依存

モバイルデバイスに対応するために，特定のネットワーク設備やプロトコルに依存しない手法であることが求められる．例えば，近接の無線通信を前提とする手法は適応できる状況が限定的であるため望ましくない．

以上の要件を満たすために，本研究では，デバイス性能や機能に依存せず，また，特定のネットワーク設備やプロトコルに依存しない識別を行う手法として，通信情報を分析して，デバイスの種類や機種を識別するアプローチをとる．

まず，通信情報を分析することで，要件1とした，デバイスの種類と機種を特定できる．デバイスの種類と機種に応じて通信の特性は異なり，例えば，カメラは常に大容量の映像データを継続的に送信し続け．温度センサなどの原始的なセンサは，データサイズが小さいセンサ値を定期的に送信する．また，同じ種類のデバイスであっても，送信されるデータサイズと情報は機種によっても異なる．つまり，デバイスの通信情報を分析して固有の特性を見つけることで，デバイスの種類と機種を特定することができると考えられる．

さらに，本アプローチは，要件2と要件3も満たしている．ネットワークに接続してデータを送受信することは，IoTデバイスの共通的な機能であるため，デバイスへの特別なソフトウェアのインストールが不要である．また，通信情報は一般的なネットワークスイッチを介して複製を取得できるため特別な設備も不要であり，また，デバイスに対して識別用の特別な処理や通信を発生させず，デバイス本来の処理に影響を与えない．

本アプローチの概略を図4.1に示す．デバイスから収集した通信情報を蓄積したデータベースを持ち，識別対象デバイスの通信情報と，データベース内の情報を比較することで，識別対象デバイスの種類と機種を識別する．さらに，識別に用いた情報を，識別結果と合わせて随時データベースに蓄積する．

本研究は，ネットワークカメラや温度センサなど，特定の機能に特化したデバイスを対象とする．IoTはこのような単機能のデバイスを中心に普及拡大が進んでおり，E-CPSの主要な構成要素と成り得るためである．これらのデバイスは，パソコンやスマートフォンなどの高機能なデバイスとは異なり，使用状況ごとの通信特性の差はわずかであり，本アプローチが特に有効と考えられる．

また，E-CPSの実現には，少なくとも90%以上の識別正解率が必要と考え，本研究の目標値とする．共用されるネットワーク接続デバイスの全てをシステムに組み込んで利用できることが理想的ではあるが，あらゆる環境に潤沢な数のデバイスが普及することを想定すると，全体の9割程度のデバイスが利用できることでサービスを実行できると考えている．

また，識別する情報としては，少なくともデバイスの機種が必要である．これは，デバイスにネットワークを介して操作指令を送るには，そのデバイスが有するAPIやコマンドを特定することが必要なためである．本研究では，APIやコマンド体系は製造者から一般公開され，機種を特定できればインターネットを介してそれらの情報を入手できることを前提としている．

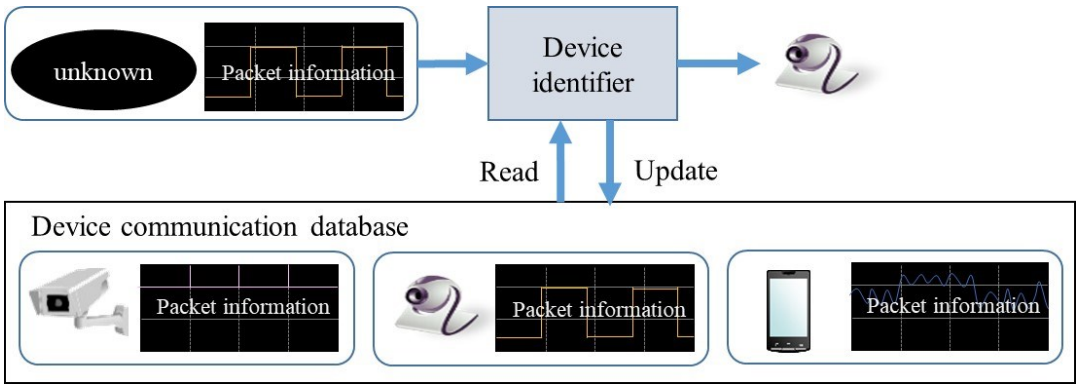


図 4.1 通信情報によるデバイス識別のアプローチ

4.3 通信情報分析の関連研究

本節では，本研究と同様に通信情報を用いてデバイスやソフトウェアを特定する先行研究に関して述べる．

通信情報を分析することで OS やアプリケーションを識別する研究がある．Matsunaka 等[84]と Chang 等[85]は，DNS (Domain Name System) クエリの送信周期や送信先から OS の推測を試みている．これらはデバイスの機種に寄らず適用できる技術であるが，デバイスの種類と機種の識別は扱っていない．

Meidan 等[86]は，TCP パケットから抽出された特徴量を使用してデバイスの種類を識別する方法を提案している．彼らは，プリンタ，テレビ，スマートウォッチなどの 9 種類のデバイスをデータから分類したが，機種の識別は行っていない．Kawai 等[87]は，パケットサイズとパケット到着時間（IAT）の 2 つの要素のみを用いてデバイスの種類と機種を識別する方法を提案している．特定のプロトコルに依存しないため，様々な IoT デバイスに適用できる非常に効果的な手法だが，同一種類のデバイスが多く存在する環境下での評価は行っておらず，そのような条件において高精度に機種を識別できることは示されていない．Dalai 等[88]は，セキュリティ上脆弱なワイヤレスデバイスをネットワークから分離することを目的に，デバイスの種類を特定する手法を提案している．本手法は，ネットワークスキャン中にデバイスが送信するプローブ要求フレームを使用しているため，ワイヤレスデバイスにしか適用できない．また，データをキャプチャするには，識別システムが同一 WiFi エリア内に存在しなければいけないという制約がある．Markus 等[89]は，同様に脆弱なデバイスの隔離を目的に，セキュリティゲートウェイを含む総合的なシステムを提案している．16 種類の通信プロトコルから，識別に有効な 23 種類の特徴量を抽出しているが，その中には HTTP (Hypertext Transfer Protocol) や DNS といったアプリケーションレイヤのヘッダ情報が多く含まれており，汎用的な通信プロトコルの特徴量だけで識別を行うことは追及していない．

以上の関連研究に対して，本研究は，同一種類の多数のデバイスが存在する中で，種類だけでなく機種を高精度に識別することを目指す．また，多種のIoTデバイスへの適用を考慮し，より汎用的な通信情報だけを用いた識別を目指して分析を行う．

4.4 デバイス識別に関する提案手法

デバイスを通信情報から識別する手法を提案する．本手法は，汎用的な処理とデバイスごとに固有な実装が必要な処理が分割されており，多種デバイスへの拡張性を有するものである．図4.2に提案手法の処理の流れを示す．以下，処理の順序に沿って説明する．

A. 通信情報の収集

一つ目の処理は，識別対象のデバイスが送受信する通信情報の収集である．本手法は，ネットワークスイッチのミラーポートを使用し，コピーされた通信パケットを収集する（図4.3）．したがって，デバイス本来の処理には影響を与えない．

通信パケットを取得した後に，パケットからヘッダ情報を抽出する．本手法は，パケットのヘッダ情報を通信の性質を表す最小単位の情報として扱う．図4.4に示すように，TCP/IP通信のヘッダ情報にはパケット長やTCPポート番号といった複数のフィールドがある．これらのヘッダフィールドの値を抽出する．本機能は，通信プロトコルに応じて実装する．

複数のデバイスがネットワーク内に存在する中で，異なるデバイスの通信が混在しないようにするため，MACアドレス（Media Access Control address）等の特定のヘッダフィールドの値の同一性からデバイスの通信パケットを区別する．このとき，モバイルデバイス等の頻繁にネットワークを移動するデバイスへ対応するために，ヘッダフィールドの値とデバイスの対応は一定時間を経過するとリセットする．なぜなら，永続的にユニークなヘッダフィールドはなく，例えば，ローカルネットワークのIPアドレスは，ネットワーク接続のたびに変化することも，異なるローカルネットワーク間で重複することもあるためである．また，MACアドレスはデバイスの個体を識別するために用いられることが多いものの，セキュリティ上の理由から，MACアドレスをネットワークに接続するたびに自動生成するOSがあり，やはり確実な一貫性はない．

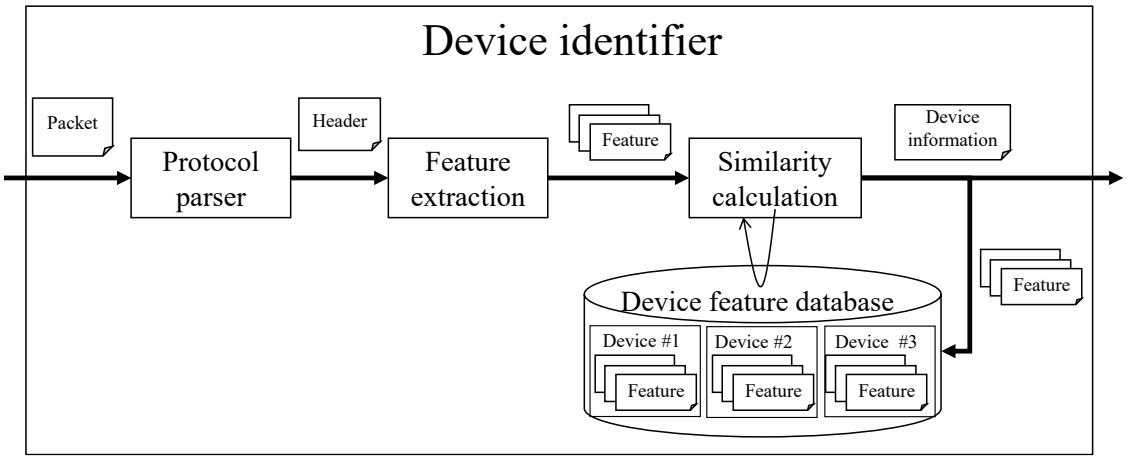


図 4.2 デバイス識別処理の流れ

(©IEEE 主著論文 2 より引用)

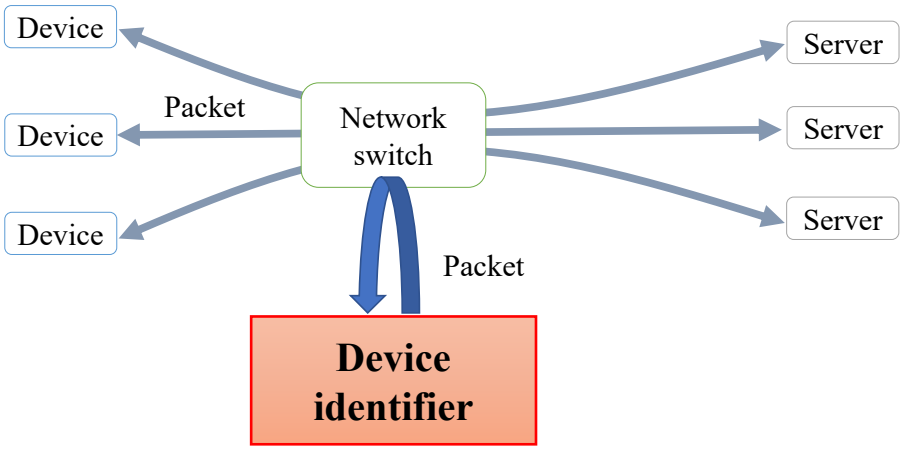


図 4.3 通信情報の収集方法

(©IEEE 主著論文 2 より引用)

TCP/IPパケットの構成例

Ether	IP	TCP	HTTP	Payload

Ethernet IIヘッダ

Destination MAC address	Source MAC address	Type

IPv4ヘッダ

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 bit

Version	IHL	Service type	Total length	
Identification			Flags	Fragment offset
Time to live		Protocol	Header checksum	
Source IP address				
Destination IP address				
Options				

TCPヘッダ

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 bit

Source port			Destination port
Sequence number			
Acknowledgment number			
Data offset	Reserved	Control flag	Window size
Checksum			Urgent pointer
Options			

図 4.4 IP 通信パケットのプロトコルスタック

B. 通信特徴量の抽出

二つ目の処理は，収集，蓄積されたヘッダ情報から一定の時間周期で通信特徴量を抽出することである．本手法は，単一パケットのヘッダ情報ではなく，一定時間に送受信する複数パケットのヘッダ情報を分析して特徴量を求める．なぜなら，パケット長の変動傾向やバーストの有無，使用ポート番号の変更周期性といった特徴こそがデバイスの識別に有用なものと考えているためである． ヘッダフィールドによっては文字列を扱うものがあるが，図4.5に示すように，全てのヘッダフィールドの値を一律に数値化して扱い，特徴量を算出する．本研究では，試験的に数種類の特徴量を用いて識別への有用性を検証した．まず，一定時間の最大値，最小値，平均値，および一次近似曲線の勾配と切片を用いた．最大値，最小値，および平均値は，送受信される通信パケットの統計的な特徴である．また，一次近似曲線の勾配と切片は，特定期間内の通信の時間変動傾向を表す特徴である．さらに，一定時間の収集パケットのヒストグラムも特徴量として扱った．具体的には，ヘッダフィールドごとに，値域をN個の階級に区切り，それぞれの階級に属するパケットの数を特徴量とした．ヒストグラム特徴量の詳細は4.6節にて述べる．

以上の特徴量を次工程の処理である，類似度算出に用いる最小単位の情報とする．

Received time	Header #1	Header #2	Header #3	･･･	Header #n
09:00:02	40	80	128	･･･	2500
09:00:10	40	80	128	･･･	2500
09:00:32	1500	80	128	･･･	2500
09:00:56	40	80	128	･･･	2500
09:00:58	700	63000	64	･･･	8200

Feature	Header #1	Header #2	Header #3	･･･	Header #n
Maximum value	1500	63000	128	･･･	8200
Minimum value	40	80	64	･･･	2500
Average value	464	12664	115.2	･･･	3640
(Slope, Intercept)	(6.8, 248.1)	(630.3, -7254.9)	(-0.6, 135.5)	･･･	(57.1, 1835.5)

図 4.5 ヘッダフィールド情報の特徴量化方法

(©IEEE 主著論文 2 より引用)

C. 通信特徴量の類似度算出

三つ目の処理は，識別対象デバイスの通信特徴量と，データベースに蓄積されている過去に収集したデバイスの通信特徴量との類似性を定量値として算出し，両デバイスが同一であるか，異なるかを判定することである．以降，特徴量の類似性を0～100の範囲の値で示した指標値を類似度と呼ぶ．類似度の算出には複数の方法が取り得るが，最も単純には，式4.1と式4.2に示す計算式を用いて，特徴量のユークリッド距離として算出する．機械学習を用いる応用手法を4.6節にて後述することとし，本節では，ユークリッド距離を用いる算出方法を示す．

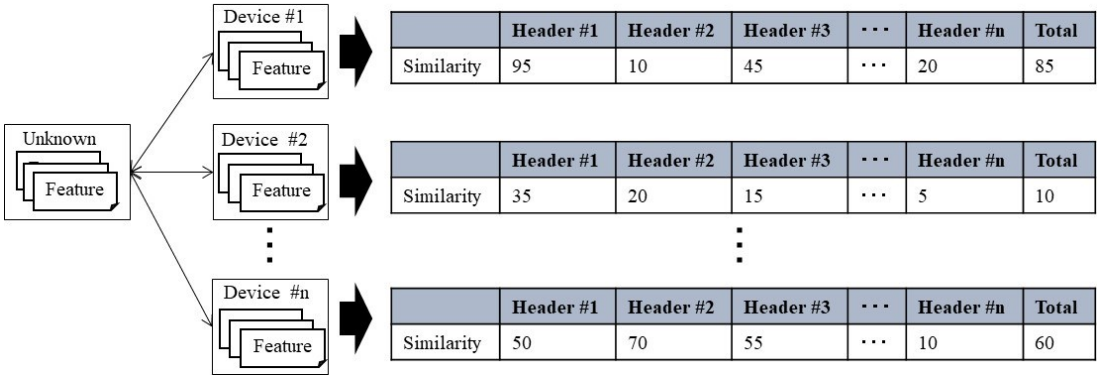
式4.1において，c_iはヘッダフィールドごとの類似度，Δxは識別対象デバイスとデータベース内の比較対象デバイスの特徴量のユークリッド距離，iはヘッダフィールドの番号を示す．あるヘッダフィールドから生成した特徴量について距離Δx_iを求め，比較対象デバイス全体において最も距離が離れているヘッダフィールド$max(\Delta x)$で除して正規化する．つまり，距離が0となる場合に類似度は100となり，最も距離が離れているヘッダフィールドでは類似度は0となる．式4.2において，Cは総合的な類似度，k_iは後述するヘッダフィールドごとの重みである．ヘッダフィールドごとの類似度に重みをかけた後に総和を取り，比較対象デバイスとの類似度とする． 図4.6は，類似度の計算過程を示したものである．

なお，最大値，最小値，平均値といった複数の特徴量を識別に用いる場合には，特徴量ごとに上記計算により類似度を算出した後に，全特徴量の平均を求めて総合的な類似度とする．

$$c_i = \left(1 - \left(\Delta x_i / max(\Delta x)\right)\right) \times 100 \qquad (4.1)$$

$$\mathrm{C} = \sum k_i c_i \qquad (4.2)$$

データベースに蓄積された全ての種類や機種に対する類似度を算出した後，最も類似度が高く，かつ，一定の閾値より高い類似性が認められたものを同一デバイスと判断する．一定の閾値を上回るものが無い場合には，識別対象デバイスを，データベースに情報が無い新規のデバイスと判断する．



	Header #1	Header #2	Header #3	・・・	Header #n	Total
Similarity	95	10	45	・・・	20	85

	Header #1	Header #2	Header #3	・・・	Header #n	Total
Similarity	35	20	15	・・・	5	10

	Header #1	Header #2	Header #3	・・・	Header #n	Total
Similarity	50	70	55	・・・	10	60

図 4.6 デバイス類似度の算出方法

(©IEEE 主著論文 2 より引用)

式4.2において, k_iと示した重みについて説明する. よりユニークなヘッダフィールドほど, 識別に強く反映させる必要がある．例えば，HTTPクライアント端末が送信するパケットの宛先ポート番号は重複することが多い．宛先ポート番号を使用して類似性を計算すると，全てのHTTPクライアント端末の類似性が一律に高くなり，種類や機種のデバイスを正しく識別できない恐れがある．このような問題に対処するために，ヘッダフィールドごとに適切な重みを設定することが有効と考えられる．しかしながら，多種多様かつ，頻繁に新製品が登場するIoTデバイスの識別においては，重みを手動で設定，管理することは困難である．したがって，ユニークなヘッダフィールドを自動的に判断して適切な重みを動的に設定する手法が不可欠である．本研究で提案する動的重み設定法を以下に示す.

式4.3により重みk_iを算出する．本式のv_iは，あるヘッダフィールドにおける，データベース内の全ての比較対象デバイスに対する類似度の分散である．分散が大きいほどユニークなヘッダフィールドと言える．全てのヘッダフィールドの分散を算出した後に，ヘッダフィールドごとの分散の比率をそれぞれの重みk_iとする．つまり，全てのヘッダフィールドの重みの合計は1であり，分散が大きいものほど，大きな重みが設定される.

$$k_i = v_i / \sum v_i \qquad (4.3)$$

4.5 デバイス識別に関する予備実験

提案手法の有効性を評価するために，4.4節に述べた機能を搭載したデバイス識別システムを開発し，2つの予備実験を行った．1つ目は，複数ネットワークカメラが混在する環境において，特徴量として有用なヘッダフィールドを確認した．2つ目は，実際のIoT環境におけるフィージビリティ評価として，工場を模擬した環境における複数種類のデバイスに対する識別を行った.

4.5.1 特徴量として扱うヘッダフィールドの検証

本検証は，識別に用いるヘッダフィールドの検証を目的としたものである．表4.1に示す，メーカーが異なる4機種のネットワークカメラが接続された環境を用いて，使用するヘッダフィールドと識別性能との関係を明らかにした.

本実験における全てのネットワークカメラは，ネットワークスイッチを介して有線接続されており，同一ネットワーク内にある映像視聴PCへ常時映像を配信する状態とした．ネットワークカメラの送信パケットは，ネットワークスイッチのミラーポートを介して収集した.

本項に示す全ての実験において，通信特徴量の抽出周期とデバイス識別周期は等しく30秒とし，特徴量には，ヒストグラム特徴量以外の全てを用いた．本周期で常時デバイスの識別を実行し，最も高い類似度を示すデバイスを判定した．なお，本手法をシステム化して運用

する際には，識別結果に基づいた分類で特徴量が蓄積されるが，本実験では実験条件を単純化するため，使用済み特徴量は識別成否に関わらず全て正解デバイスの特徴量として蓄積した．

以上の条件のもと，特徴量化して扱うヘッダフィールドをネットワークプロトコルの階層ごとに区切り，それぞれを使用した場合の識別性能を確認した．また，ヘッダフィールドに付与する重み付けの効果および，撮影映像に対するパケット長の変動についてもあわせて本項に示す．

表 4.1 ヘッダフィールド検証における使用デバイスと設定

Manufacturer	**Model**	**Resolution**		
AXIS	M1034-W EUR	QVGA (320×180)	VGA (640×360)	HD (1280×720)
CANON	VB-M720F	QVGA (320×180)	VGA (640×360)	HD (1280×720)
VSTARCAM	C7823WIP	QVGA (320×180)	VGA (640×360)	-
I-O DATA	TS-Wrlp	352×200	-	HD (1280×720)

A. 特徴量として扱うヘッダフィールドと識別性能の関係

TCP/IP通信には，図4.4に示した通り，複数のネットワーク階層があり，パケットのヘッダはこの階層に従って構成されている．下位のヘッダほど，異なるデバイス間でも共通的に使用される可能性が高い．通信情報の汎用性の観点で分析を行うために，特徴量として用いるヘッダをネットワーク階層ごとに区切り，デバイス識別への効果を確認した．

ユニーク性と再現性の観点から，識別に使用するヘッダフィールドは，ネットワーク層のパケット長とTTL (Time To Live)，トランスポート層のTCPウィンドウサイズ，およびアプリケーション層のHTTPヘッダのみに限定した．HTTPヘッダについて，より詳細に実験条件を述べると，本実験で使用したネットワークカメラは，表4.2に示すフィールドを使用する．

以上を整理すると，本実験では，識別に用いる特徴量に関して，表4.3に示す3つの条件を扱う．一つ目は，IPヘッダのみを特徴量として使用するもの，二つ目は， TCPヘッダ以下のヘッダを特徴量として使用するもの，三つ目は， HTTPヘッダ以下のヘッダを特徴量として使用するものである．

なお，本実験では，ネットワークカメラの解像度は表4.1における最小に設定し，カメラは静止した同一のオブジェクトを撮影している状態とした．なお，デバイスから映像視聴PCに送信されるパケット（TX）と，映像視聴PCからデバイスに送信されるパケット（RX）はそれぞれ独立した特徴量として扱った．

表 4.2　ヘッダフィールド検証における対象 HTTP ヘッダフィールド

Number	Header	Status	Number	Header	Status
1	Date	Use	21	If-None-Match	-
2	Pragma	Use	22	If-Range	-
3	Cache-Control	Use	23	If-Unmodified-Since	-
4	Connection	Use	24	Max-Forwards	-
5	Transfer-Encoding	-	25	Proxy-Authorization	-
6	Upgrade	-	26	Range	-
7	Via	-	27	Expect	-
8	Trailer	-	28	TE	-
9	Warning	-	29	Location	-
10	Authorization	Use	30	Server	Use
11	From	-	31	WWW-Authenticate	-
12	If-Modified-Since	-	32	Accept-Ranges	Use
13	Referer	Use	33	Age	-
14	User-Agent	Use	34	Proxy-Authenticate	-
15	Accept	Use	35	Public	-
16	Accept-Charset	-	36	Retry-After	-
17	Accept-Encoding	Use	37	Vary	-
18	Accept-Language	Use	38	Warning	-
19	Host	Use	39	Alternates	-
20	If-Match	-	40	Etag	Use

表 4.3 ヘッダフィールド検証における使用特徴量

Features	Header			
	IP Total length	IP TTL	TCP Window size	HTTP
IP ヘッダのみ	Use	Use	-	-
TCP ヘッダ以下のヘッダ	Use	Use	Use	-
HTTP ヘッダ以下のヘッダ	Use	Use	Use	Use

実験結果を以下に示す.

識別に使用した特徴量の種類，デバイスごとの識別再現率を図4.7に，特徴量の種類ごとの混同行列を表4.4～表4.9に示す．表内に記載された数値は，識別された特徴量の割合（%）を示す．本項の実験では，識別性能の指標として識別再現率を扱う．識別再現率とは，あるデバイスの通信情報から生成された全ての特徴量のうち，正しいデバイスとして識別された割合である.

送信パケットを用いる場合の識別再現率は，IP ヘッダのみを用いる場合には，AXIS のみ 40%，TCP ヘッダ以下を用いる場合と HTTP ヘッダ以下を用いる場合には，VSTARCAM が 70%，他は全て 100%であった．また，受信パケットを用いる場合の識別再現率は，特徴量の種類によらず，VSTARCAM が 15%，他は全て 100%であった.

以上より，ヘッダフィールドごとの識別性能を比較すると，送信パケットについては，IP ヘッダのみを用いる場合よりも，さらに上位のヘッダを合わせて利用する場合の方が識別再現率の平均値が向上することを確認した. 一方で, VSTARCAM の識別再現率については，IP ヘッダのみを用いる場合よりも低下しており，デバイスごとに識別に優位なヘッダフィールドには差があることを確認した．また，受信パケットの結果と合わせて，HTTP 以下のヘッダを用いる場合と TCP 以下のヘッダを用いる場合とでは，識別再現率に差は見られなかった.

さらに，受信パケットを特徴量として用いる場合に，VSTARCAMの識別再現率が低い理由を分析した．収集したパケットを個別に分析したところ，AXISとVSTARCAMは，定期的にHTTP応答パケットを受信していることが判明した．そして，VSTARCAMのみ特徴量抽出周期よりも長い周期でicslapパケット(Microsoft開発プロトコルのパケット)を受信していた．つまり，蓄積されたVSTARCAMの特徴量には，icslapパケットが含まれるものと，含まれないものが混在していた．したがって，後者の特徴量を識別する際に，二種類の特徴量が混在して蓄積されたVSTARCAMよりも，AXISとの類似度の方が高く算出されていた.

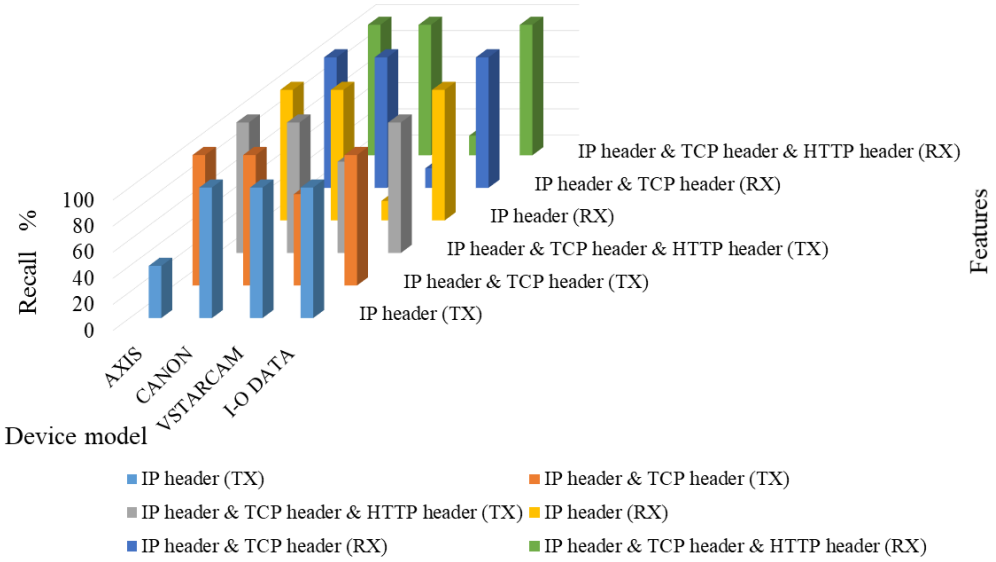


図 4.7 特徴量と機種ごとの識別再現率

表 4.4 IP ヘッダ（TX）を用いた識別の混同行列

		Recognized class			
		AXIS	CANON	VSTARCAM	I-O DATA
True class	AXIS	40	35	15	10
	CANON	0	100	0	0
	VSTARCAM	0	0	100	0
	I-O DATA	0	0	0	100

表 4.5 TCP ヘッダ以下のヘッダ（TX）を用いた識別の混同行列

		Recognized class			
		AXIS	CANON	VSTARCAM	I-O DATA
True class	AXIS	100	0	0	2
	CANON	0	100	0	0
	VSTARCAM	0	30	70	0
	I-O DATA	0	0	0	100

表 4.6 HTTP ヘッダ以下のヘッダ（TX）を用いた識別の混同行列

		Recognized class			
		AXIS	CANON	VSTARCAM	I-O DATA
True class	AXIS	100	0	0	0
	CANON	0	100	0	0
	VSTARCAM	0	30	70	0
	I-O DATA	0	0	0	100

表 4.7 IP ヘッダ（RX）を用いた識別の混同行列

		Recognized class			
		AXIS	CANON	VSTARCAM	I-O DATA
True class	AXIS	100	0	0	0
	CANON	0	100	0	0
	VSTARCAM	85	0	15	0
	I-O DATA	0	0	0	100

表 4.8 TCP ヘッダ以下のヘッダ（RX）を用いた識別の混同行列

		Recognized class			
		AXIS	CANON	VSTARCAM	I-O DATA
True class	AXIS	100	0	0	0
	CANON	0	100	0	0
	VSTARCAM	85	0	15	0
	I-O DATA	0	0	0	100

表 4.9 HTTP ヘッダ以下のヘッダ（RX）を用いた識別の混同行列

		Recognized class			
		AXIS	CANON	VSTARCAM	I-O DATA
True class	AXIS	100	0	0	0
	CANON	0	100	0	0
	VSTARCAM	85	0	15	0
	I-O DATA	0	0	0	100

B. ヘッダフィールドの重み付け効果検証

ヘッダフィールドに対する重み付けに関して，新規デバイス判定に関する効果の検証を行った．本実験では，送信パケットから抽出した，IPヘッダのパケット長，TTL，TCPヘッダのウィンドウサイズ，およびHTTPヘッダを特徴量として使用した．

図4.8は，デバイスごとの識別に用いた全特徴量に関する，正解デバイスとの類似度の最小値，および正解以外のデバイスとの類似度の最大値を重み付けの有無それぞれの場合について示したものである．新規デバイスの判定は，これら2つの値に挟まれる範囲に閾値を設けることで行う．

実験結果より，重み付けを行わない場合，AXISの正解デバイスとの類似度の最小値は，正解以外のデバイスとの類似度の最大値よりも下回っている．つまり，新規デバイス判定閾値を設けることができない．一方で，重み付けを行う場合には，VSTARCAMを除く，3機種について，正解デバイスとの類似度の最小値が，正解以外のデバイスとの類似度の最大値を上回っている．つまり，これら3機種の全特徴量について，対応する蓄積データが無い場合には，新規デバイスとして判定することができる．具体的には，類似度74～80%の範囲に閾値を設けることが可能である．以上より，新規デバイス判定においてヘッダフィールドの重み付けが有効である可能性が示された．なお， VSTARCAMは，表4.6に示した通り誤識別が発生していることから，閾値が設けられないことが明らかである．

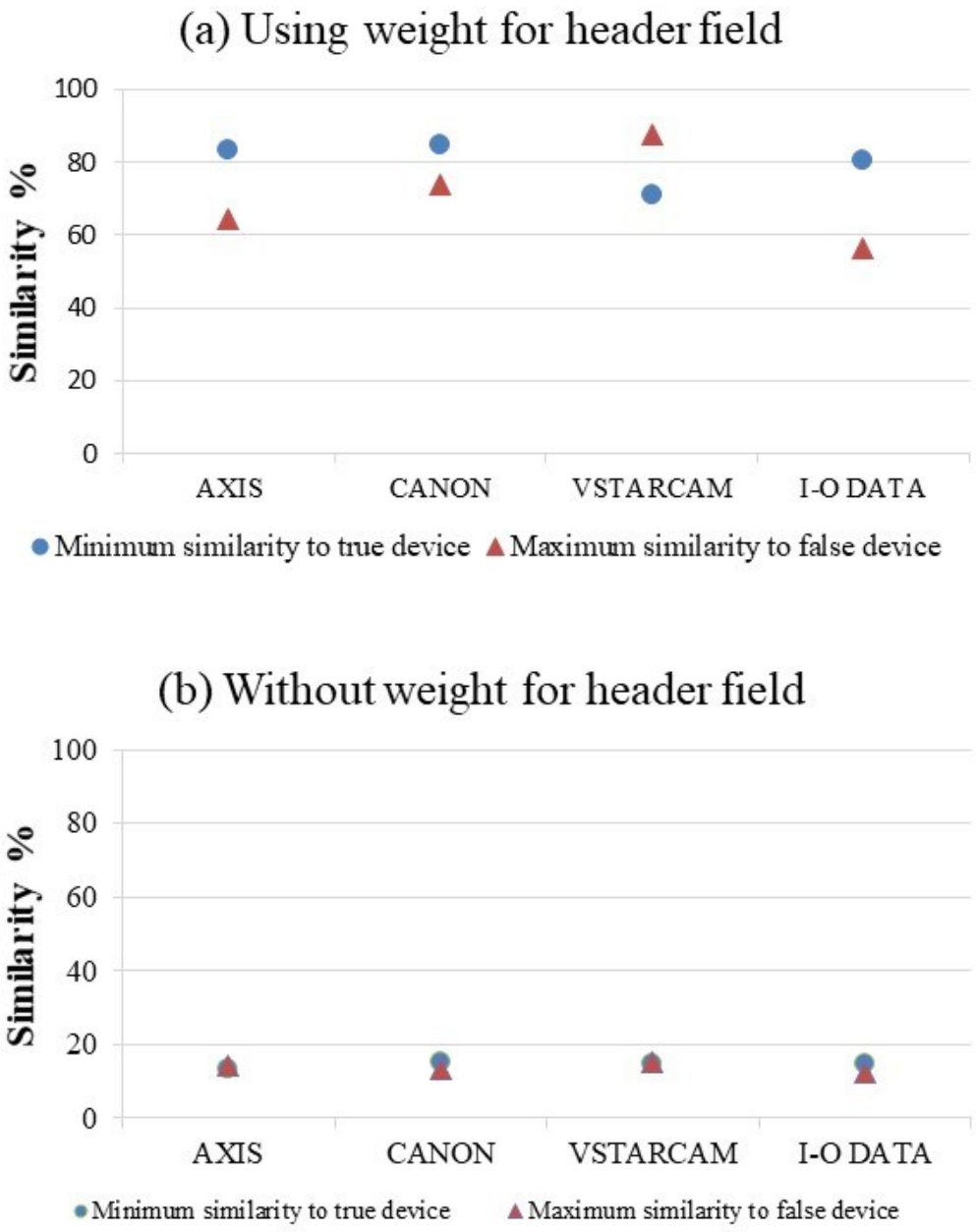


図 4.8 ヘッダフィールドの重み付け有無による類似度の比較

C. ネットワークカメラのパケット長の変動測定

ネットワークカメラのパケット長は，同一設定の同一機種であっても，撮影画像によって変化する．したがって，使用環境の変更等により撮影画像が変化すると，パケット長から得られる特徴量も変化してしまい，機種を正しく識別できなくなる恐れがある．このような撮影画像の変化による影響を検証するため，機種と解像度の組み合わせで構成される9条件について，撮影対象の色が様々に変動する環境において，解像度の設定ごとに送信パケットのパケット長の変動範囲を測定した．実験結果を図4.9に示す．

機種ごとに，パケット長の範囲を異なる解像度で比較すると，CANONのVGAとQVGAには一定範囲の重複が見られたが，AXISの範囲の重複は微小であった．また，9条件全てのパケット長の値域を比較すると，同時に重複するのは，最大でも5条件である．以上より，ネットワークカメラのパケット長は撮影画像に応じて変動するものの，その変動範囲はデバイスごとにユニークであるため，撮影画像が変化する環境下においても識別に有効である可能性が示された．

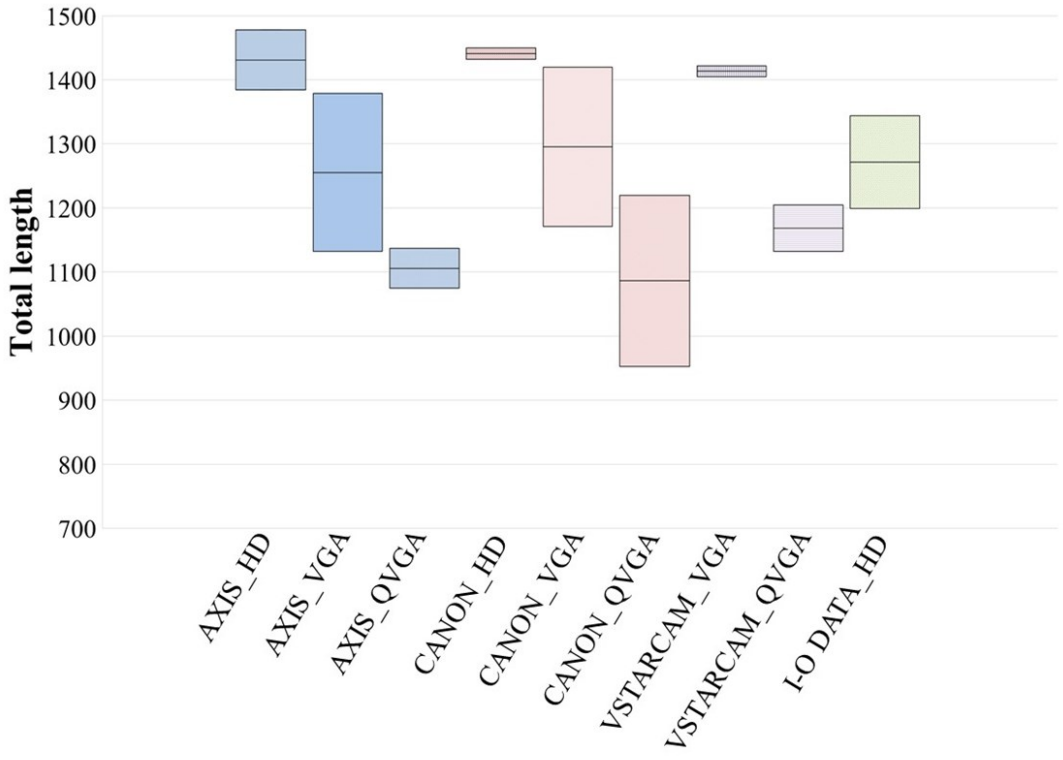


図 4.9　ネットワークカメラの機種，解像度ごとのパケット長の変動範囲

4.5.2 工場模擬環境におけるフィージビリティ評価

実際のIoT環境における提案方法のフィージビリティ評価および，識別性能評価の予備実験として，工場の生産ラインを模擬したネットワーク環境で実験を行った．図4.10は，実験環境のネットワーク構成を示したものである．ネットワークには，PLC (Programmable Logic Controller)，パトランプ，ネットワークカメラ（AXIS M1034-W EUR），PLCを制御するためのPC，パトランプを制御するためのPC，およびカメラ映像視聴用PCが接続されている．これらのデバイスを生産ラインにおける通常稼働状態とし，特徴量の蓄積と識別を並行して行った．本実験では，いずれのPCも特定の単機能しか持たないため，それぞれを異なる種類のデバイスとして扱った．

提案手法を実装した識別用サーバをネットワークスイッチのミラーポートに接続し，各デバイスの送信パケットを収集した．本予備実験では，特徴量として，パケット長，TTL，およびTCPウィンドウサイズを使用し，特徴量抽出周期は10秒とした．

予備実験の結果を表4.10に示す．デバイスごとに識別再現率をみると，カメラは92%，カメラ映像視聴用PCは16%，それ以外のデバイスは全て100%となった．

カメラ映像視聴用PCの識別再現率が低い理由を分析した．同デバイスの送信パケットを確認したところ，RTSP (Real Time Streaming Protocol) とHTTPの2種類のパケットを同量かつ特徴量抽出周期よりも長い周期で送信していた．したがって，特徴量抽出の10秒分の収集パケットの中に，常に一方の種類のパケットだけが含まれることになり，データベースに二種類の特徴量が同量ずつ蓄積されることになった．識別対象の特徴量は，どちらかの特徴量であるため，データベース内の特徴量の半分と常に一致せず，低い類似度が算出された．したがって，10秒よりも長い特徴量抽出周期を用いることが必要である可能性が判明した．

また，特異な現象が認められたため原因を分析した．パトランプ制御用PCは，特徴量を正しく識別できているものの類似度が低く算出される場合があった．対応する時間の通信パケットを確認したところ，本周期にだけIGMP (Internet Group Management Protocol) パケットが含まれていた．IGMPは，IPネットワーク上でマルチキャストを行うためのプロトコルである．ネットワークに接続されたデバイスには，デバイス本来の通信以外にもネットワーク管理のための通信が発生することがある．このようなデバイスのユニーク性を表現しない通信が識別へ影響を与えることを防ぐために，特徴量から除外することが必要と考えられる．特に，IGMPのような低レイヤのプロトコルは，IPパケットのヘッダフィールドに含まれているプロトコル番号を見ることで特定が可能であり，ヘッダフィールドの情報だけでプロトコル単位での除外が可能である．これにより更なる性能向上が期待できることが判明した．

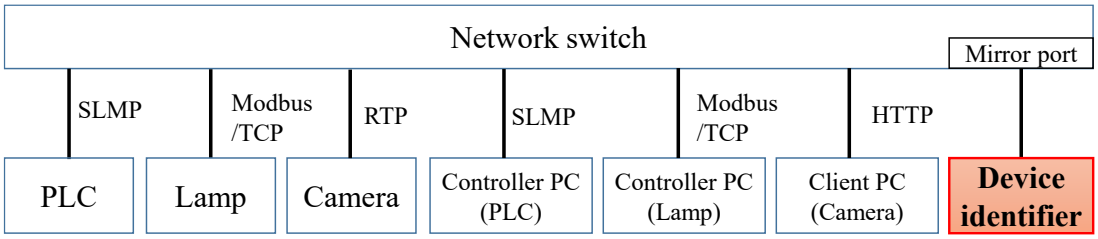


図 4.10 工場模擬環境のネットワーク構成
(©IEEE 主著論文 2 の Fig.12 を一部修正)

表 4.10 工場模擬環境における識別結果の混同行列

		Recognized class					
		PLC	Lamp	Camera	Controller PC (PLC)	Controller PC (Lamp)	Client PC (Camera)
True class	PLC	100	0	0	0	0	0
	Lamp	0	100	0	0	0	0
	Camera	0	8	92	0	0	0
	Controller PC (PLC)	0	0	0	100	0	0
	Controller PC (Lamp)	0	0	0	0	100	0
	Client PC (Camera)	4	0	0	60	20	16

4.6 提案手法の改善とデバイス識別性能評価実験

4.4 節に示した提案手法の識別性能評価の予備実験結果を改善するのを目的に，新たな通信特徴量抽出処理を開発し，複数の類似度算出処理に適用して，その有効性を確認する実験を行った．特に類似度算出処理には，識別への高い効果が期待される機械学習を適用した．本節では，はじめに，11 機種のネットワークカメラの機種識別に対する性能評価を述べ，その後，より細かい粒度である設定単位の識別に関する性能評価を述べる．

4.6.1 特徴量抽出処理と類似度算出処理の改善手法

4.4節に述べた，一定時間の最大値，平均値，最小値といった統計量よりも情報量が多い，ヒストグラム特徴量を導入する．ヒストグラム特徴量とは，ヘッダフィールドごとに値をN個の階級に区切り，特徴量抽出周期ごとに収集したパケットについて，それぞれの階級に属する数を表現したものである．つまり，M種類のヘッダフィールドを特徴量として用いる場合には，特徴量一つ当たりのデータ次元数は，N×Mとなる．

次に，類似度算出処理に関しては，機械学習を適用する．4.5 節に示す基礎実験では，ユークリッド距離を用いて特徴量の類似性を算出したが，より高精度に分類を行うには，機械学習の適用が有効であると考えられる．代表的な機械学習アルゴリズムである，ナイーブベイズ（NaiveBayes），線形サポートベクタマシン（Linear SVC），ロジスティック回帰（Logistic Regression），ランダムフォレスト（Random Forest）を適用した．本節の実験における機械学習処理は全て，オープンソースの機械学習ライブラリである scikit-learn [90]を用いて実装した．

4.6.2 デバイス機種識別の性能評価

表4.11に示す11機種のネットワークカメラを用いて，ヒストグラム特徴量を用いる場合における，機械学習アルゴリズムごとの識別性能を評価した．また，比較として，4.4節に述べたユークリッド距離による識別手法を用いた実験も実施した．

本実験では，ネットワークカメラの送信パケットのみを収集し，特徴量の抽出を行った．特徴量の抽出周期およびデバイス識別の周期は予備実験結果の知見から30秒とした．特徴量には，パケット長，TTL，およびTCPウィンドウサイズを使用し，ヒストグラム特徴量は表4.12に示す条件で作成した．なお，デバイスの設定はメーカー出荷時の初期状態とした．

以上の実験条件において，11機種のデバイスそれぞれについて，連続する10分間の送信パケットを収集し，30秒周期で抽出した20の特徴量を識別における1クラスとした．1つのデバイスに対して，同様の作業を10回実施し，1機種あたり10クラスで合計200の学習用特徴量を用意した．

このように特徴量を区切り，異なるクラスとして扱ったのは，本手法のシステム運用を考慮したためである．E-CPSのデバイス識別では，共用デバイスのネットワーク接続後，長くとも10分程度の通信パケットから抽出した試験用特徴量を用いて識別を行うことを想定している．そして，識別完了後の試験用特徴量は，次回の識別において学習用特徴量として利用することを考えている．その際に，識別結果に従って既存の学習用特徴量に試験用特徴量を統合してしまうと，誤識別が発生した場合には，学習用特徴量に誤った特徴量が混入してしまう．このような問題を防ぐために，一度の識別で使用した試験用特徴量は生成デバイスごとに，識別結果と対応する新規の識別クラスとすることを考えている．本実験条件は，このような運用形態を想定し，10分間分の学習用特徴量が複数存在する状態を再現したものである．

学習用特徴量と同様に，連続する10分間の送信パケットを収集し，20の試験用特徴量を用意した．試験用特徴量が同一機種から生成された特徴量のいずれかのクラスに分類されることを正解とした．本実験は，類似度算出手法の優劣を評価するために，全特徴量数に対する正しく識別された特徴量数の割合である識別正解率を評価指標とした．図4.11に識別正解率を示し，図4.12にデバイスごとの識別再現率を示す．

類似度算出手法に，線形サポートベクタマシン，ロジスティック回帰を用いた場合に，識別正解率は99.5%となった．なお，ナイーブベイズが最も識別正解率が低く，78.2%であった．表4.13～表4.17に各条件における混同行列を示す．

表 4.11　デバイス機種識別の性能評価実験における使用デバイスと設定

Number	Manufacturer	Model	Resolution	FPS
#1	CANON	VB-M720F	320×180	30
#2	VSTARCAM	C7823WIP	HD (1280×720)	20
#3	ELECOM	NCC-EWF100RWH	HD (1280×720)	10
#4	HIKVISION	DS-2CD2032F-I	FHD (1920×1080)	25
#5	PANASONIC	WV-SPN310	HD (1280×720)	30
#6	BOSCH	NIN-50022-A3	FHD (1920×1080)	25
#7	I-O DATA	TS-Wrlp	HD (1280×720)	25
#8	I-O DATA	TS-Wrlp/e	HD (1280×720)	25
#9	I-O DATA	TS-Wrlc	640×480	15
#10	I-O DATA	TS-Wlc2	640×480	15
#11	PLANEX	CS-QP50F	FHD (1920×1080)	15

表 4.12　ヒストグラム特徴量のビン設定（11 機種の識別実験）

	Bin [1]	Bin [2]	Bin [3]	Bin [4]	Bin [5]
Total length	0–300	301–600	601–900	901–1200	1201–
TTL	0–50	51–100	101–150	151–200	201–
TCP window size	0–2000	2001–4000	4001–6000	6001–8000	8001–

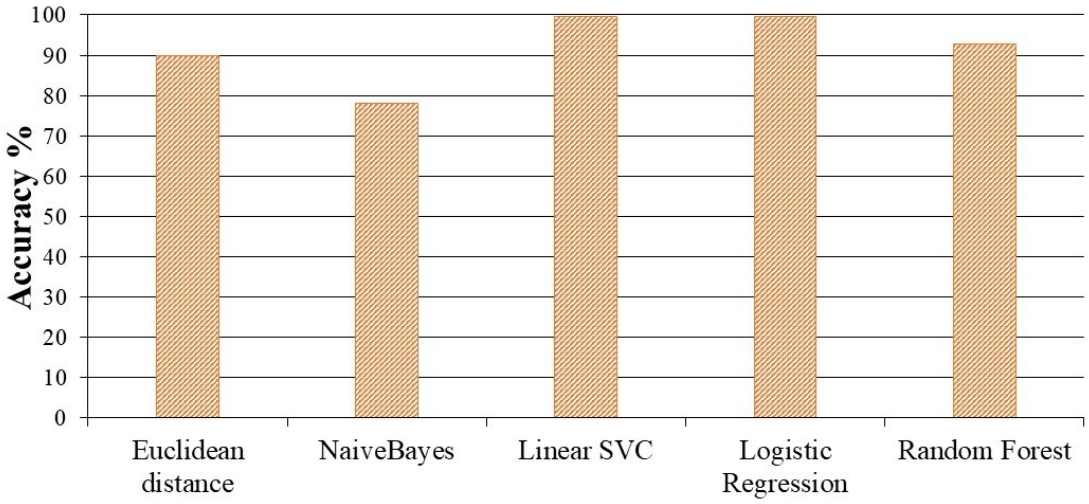
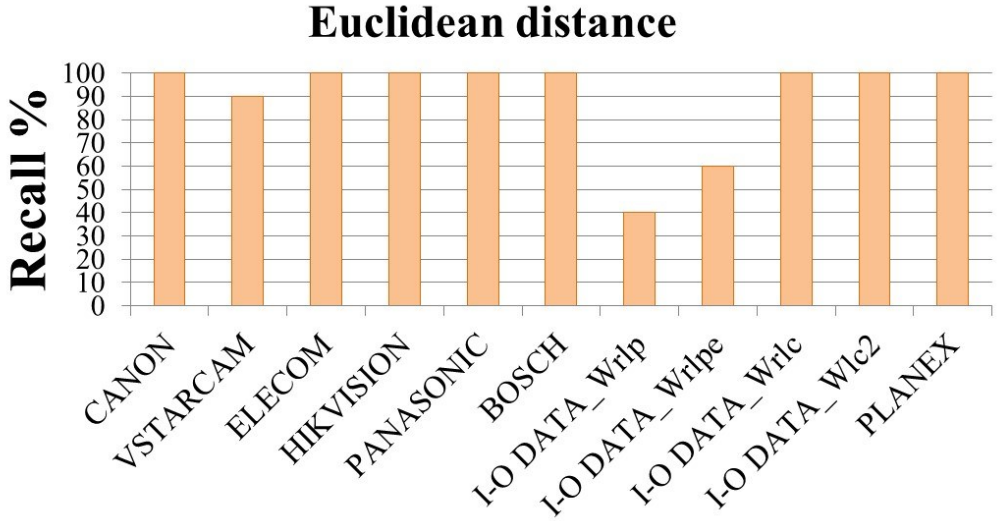
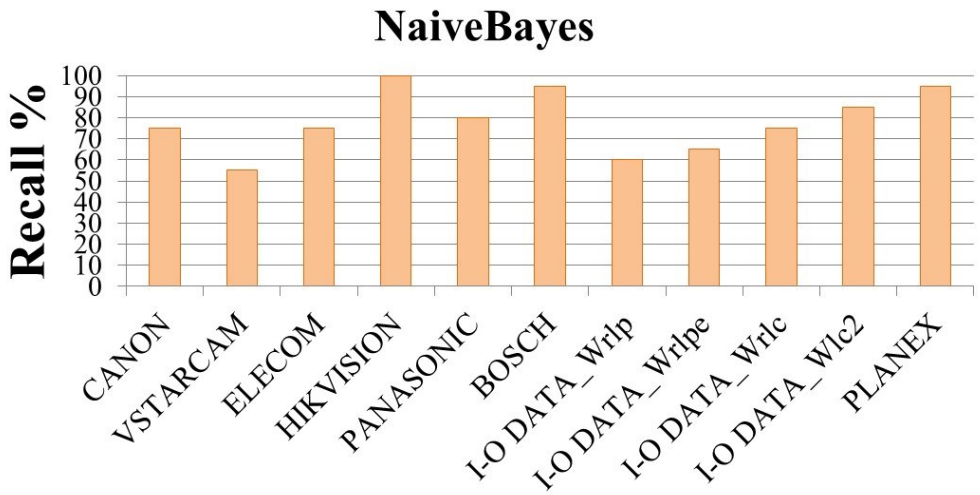
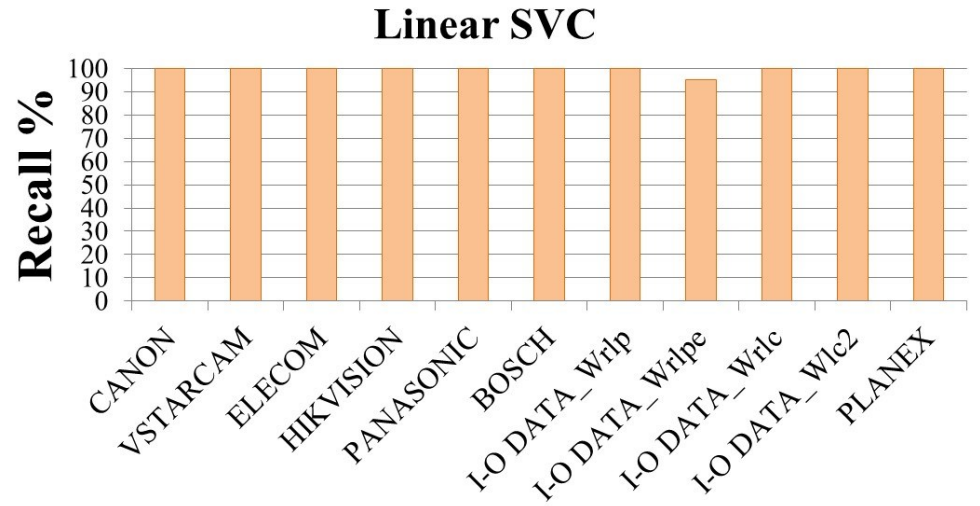
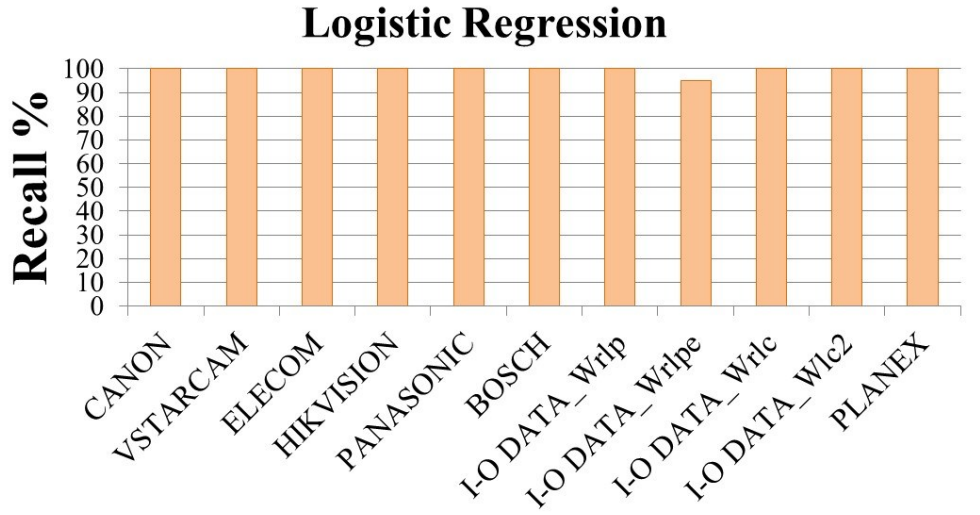


図 4.11　ヒストグラム特徴量と機械学習を用いた識別正解率（11機種）




NaiveBayes
Recall %
100
90
80
70
60
50
40
30
20
10
0
CANON
VSTARCAM
ELECOM
HIKVISION
PANASONIC
BOSCH
I-O DATA_Wrlp
I-O DATA_Wrlpe
I-O DATA_Wrlc
I-O DATA_Wlc2
PLANEX


Linear SVC
Recall %
100
90
80
70
60
50
40
30
20
10
0
CANON
VSTARCAM
ELECOM
HIKVISION
PANASONIC
BOSCH
I-O DATA_Wrlp
I-O DATA_Wrlpe
I-O DATA_Wrlc
I-O DATA_Wlc2
PLANEX


Logistic Regression
Recall %
100
90
80
70
60
50
40
30
20
10
0
CANON
VSTARCAM
ELECOM
HIKVISION
PANASONIC
BOSCH
I-O DATA_Wrlp
I-O DATA_Wrlpe
I-O DATA_Wrlc
I-O DATA_Wlc2
PLANEX

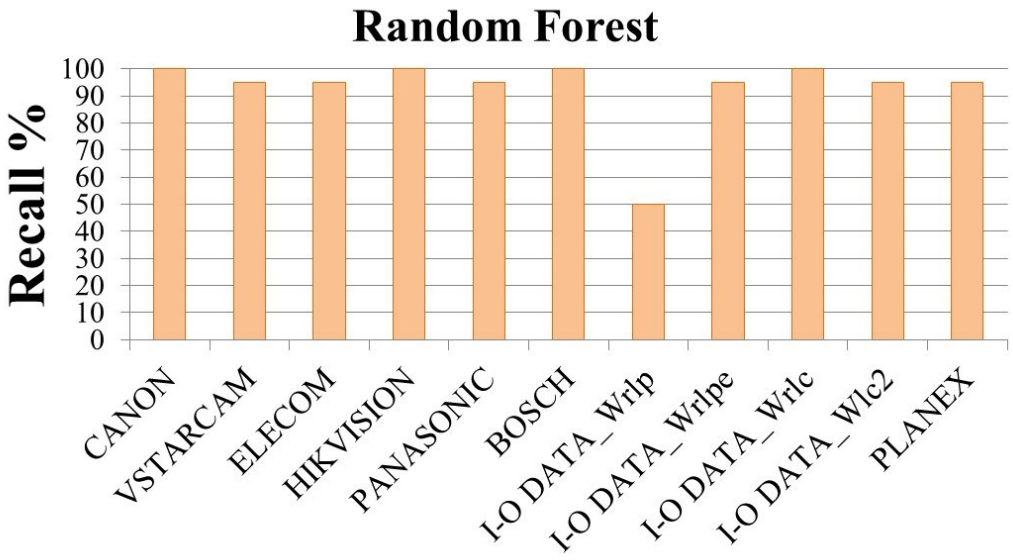


図 4.12 ヒストグラム特徴量と機械学習を用いた識別再現率（11 機種）

表 4.13 ヒストグラム特徴量を用いた識別の混同行列（Euclidean distance）

		Recognized class										
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
True class	#1	100	0	0	0	0	0	0	0	0	0	0
	#2	0	90	0	10	0	0	0	0	0	0	0
	#3	0	0	100	0	0	0	0	0	0	0	0
	#4	0	0	0	100	0	0	0	0	0	0	0
	#5	0	0	0	0	100	0	0	0	0	0	0
	#6	0	0	0	0	0	100	0	0	0	0	0
	#7	0	0	0	0	0	0	40	60	0	0	0
	#8	0	0	0	0	0	0	40	60	0	0	0
	#9	0	0	0	0	0	0	0	0	100	0	0
	#10	0	0	0	0	0	0	0	0	0	100	0
	#11	0	0	0	0	0	0	0	0	0	0	100

表 4.14　ヒストグラム特徴量と機械学習を用いた識別の混同行列（NaiveBayes）

		Recognized class										
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
True class	#1	75	0	0	25	0	0	0	0	0	0	0
	#2	0	55	0	45	0	0	0	0	0	0	0
	#3	0	0	75	25	0	0	0	0	0	0	0
	#4	0	0	0	100	0	0	0	0	0	0	0
	#5	0	0	0	20	80	0	0	0	0	0	0
	#6	0	5	0	0	0	95	0	0	0	0	0
	#7	0	0	0	40	0	0	60	0	0	0	0
	#8	0	0	0	25	0	0	10	65	0	0	0
	#9	0	0	0	25	0	0	0	0	75	0	0
	#10	0	0	0	15	0	0	0	0	0	85	0
	#11	0	0	0	5	0	0	0	0	0	0	95

表 4.15　ヒストグラム特徴量と機械学習を用いた識別の混同行列（Linear SVC）

		Recognized class										
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
True class	#1	100	0	0	0	0	0	0	0	0	0	0
	#2	0	100	0	0	0	0	0	0	0	0	0
	#3	0	0	100	0	0	0	0	0	0	0	0
	#4	0	0	0	100	0	0	0	0	0	0	0
	#5	0	0	0	0	100	0	0	0	0	0	0
	#6	0	0	0	0	0	100	0	0	0	0	0
	#7	0	0	0	0	0	0	100	0	0	0	0
	#8	0	0	0	0	0	0	5	95	0	0	0
	#9	0	0	0	0	0	0	0	0	100	0	0
	#10	0	0	0	0	0	0	0	0	0	100	0
	#11	0	0	0	0	0	0	0	0	0	0	100

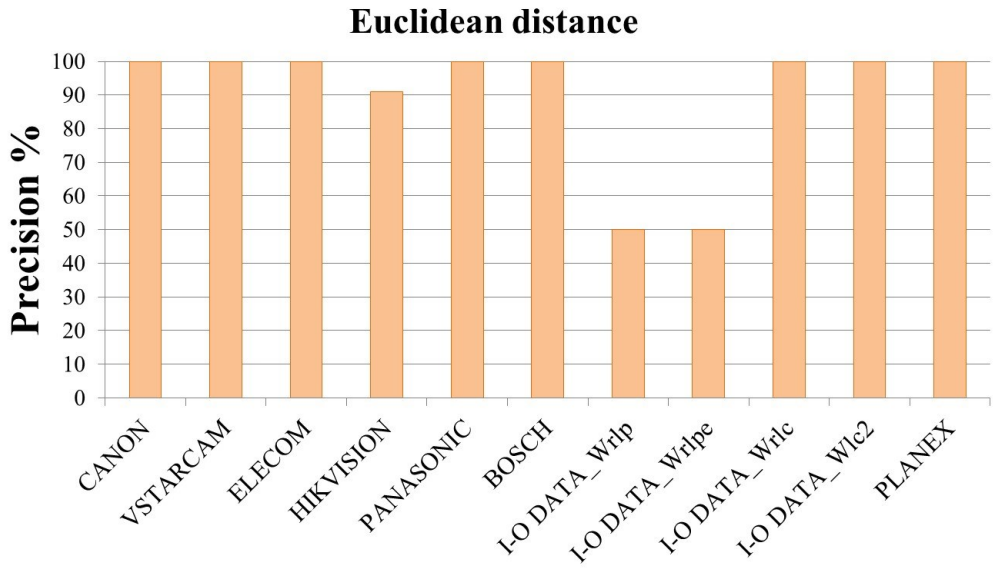
表 4.16　ヒストグラム特徴量と機械学習を用いた識別の混同行列（Logistic Regression）

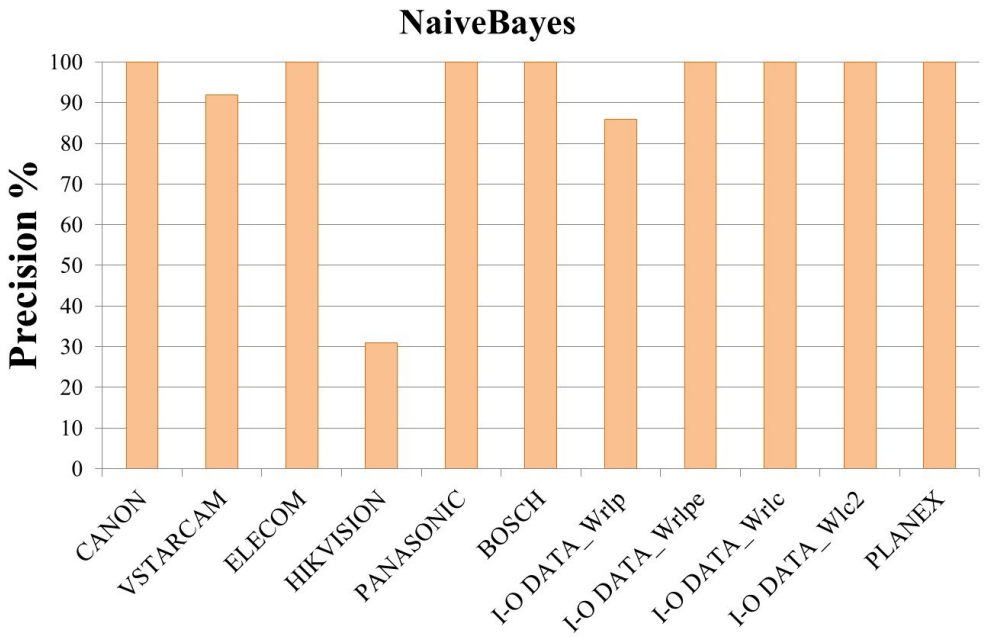
		Recognized class										
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
True class	#1	100	0	0	0	0	0	0	0	0	0	0
	#2	0	100	0	0	0	0	0	0	0	0	0
	#3	0	0	100	0	0	0	0	0	0	0	0
	#4	0	0	0	100	0	0	0	0	0	0	0
	#5	0	0	0	0	100	0	0	0	0	0	0
	#6	0	0	0	0	0	100	0	0	0	0	0
	#7	0	0	0	0	0	0	100	0	0	0	0
	#8	0	0	0	0	0	0	5	95	0	0	0
	#9	0	0	0	0	0	0	0	0	100	0	0
	#10	0	0	0	0	0	0	0	0	0	100	0
	#11	0	0	0	0	0	0	0	0	0	0	100

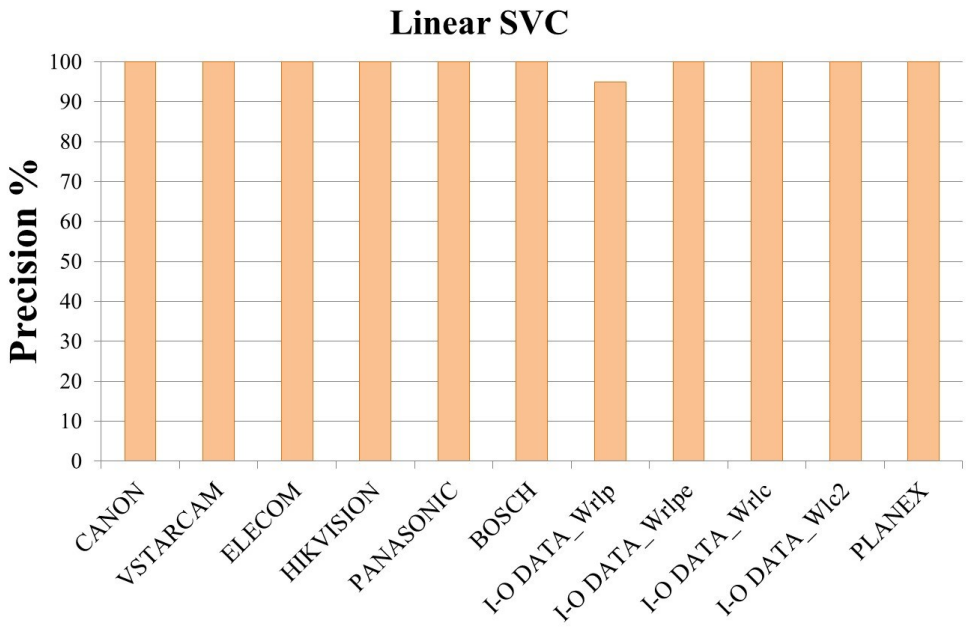
表 4.17　ヒストグラム特徴量と機械学習を用いた識別の混同行列（Random Forest）

		Recognized class										
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
True class	#1	100	0	0	0	0	0	0	0	0	0	0
	#2	5	95	0	0	0	0	0	0	0	0	0
	#3	0	0	95	5	0	0	0	0	0	0	0
	#4	0	0	0	100	0	0	0	0	0	0	0
	#5	0	0	0	5	95	0	0	0	0	0	0
	#6	0	0	0	0	0	100	0	0	0	0	0
	#7	0	0	0	45	0	0	50	5	0	0	0
	#8	0	0	0	5	0	0	0	95	0	0	0
	#9	0	0	0	0	0	0	0	0	100	0	0
	#10	0	0	0	5	0	0	0	0	0	95	0
	#11	0	0	0	5	0	0	0	0	0	0	95

以上の混同行列より，特定のデバイスに誤識別が偏る傾向が認められたため，さらに，デバイスごとの識別適合率を算出した．識別適合率とは，あるクラスに分類された全特徴量のうち，正しい特徴量の割合を示す．図 4.13 に各手法における，デバイスの識別適合率を示す．全ての手法に共通して適合率が 100%となるデバイスは 6 機種，反対にどの手法においても適合率が 100%にならないデバイスは無かった．5 つの手法のうち 4 つの手法において適合率が 100%にならなかった I-O DATA_Wrlp に関して混同行列からさらに分析すると，いずれの手法においても I-O DATA_Wrlpe のみによる誤識別が発生していることが確認された．また，I-O DATA_Wrlpe に関しても，I-O DATA_Wrlp のみによる誤識別が発生していた．以上の結果より，両者の通信は類似しており，それらの識別には高い性能が要求されることが推測される．また，ナイーブベイズを用いた場合の HIKVISION の適合率が低いことに関しても混同行列から分析したところ，9 種類の機種から誤識別が発生していることが確認された．本デバイスの特徴量は，他デバイスに対して汎化した性質を有している可能性が示唆された．一方で，線形サポートベクタマシンとロジスティック回帰では 100%の適合率を示していることから，適切な学習アルゴリズムを適用することで正しい識別が行えることを確認した．




NaiveBayes
Precision %
100
90
80
70
60
50
40
30
20
10
0
CANON
VSTARCAM
ELECOM
HIKVISION
PANASONIC
BOSCH
I-O DATA_Wrlp
I-O DATA_Wrlpe
I-O DATA_Wrlc
I-O DATA_Wlc2
PLANEX


Linear SVC
Precision %
100
90
80
70
60
50
40
30
20
10
0
CANON
VSTARCAM
ELECOM
HIKVISION
PANASONIC
BOSCH
I-O DATA_Wrlp
I-O DATA_Wrlpe
I-O DATA_Wrlc
I-O DATA_Wlc2
PLANEX

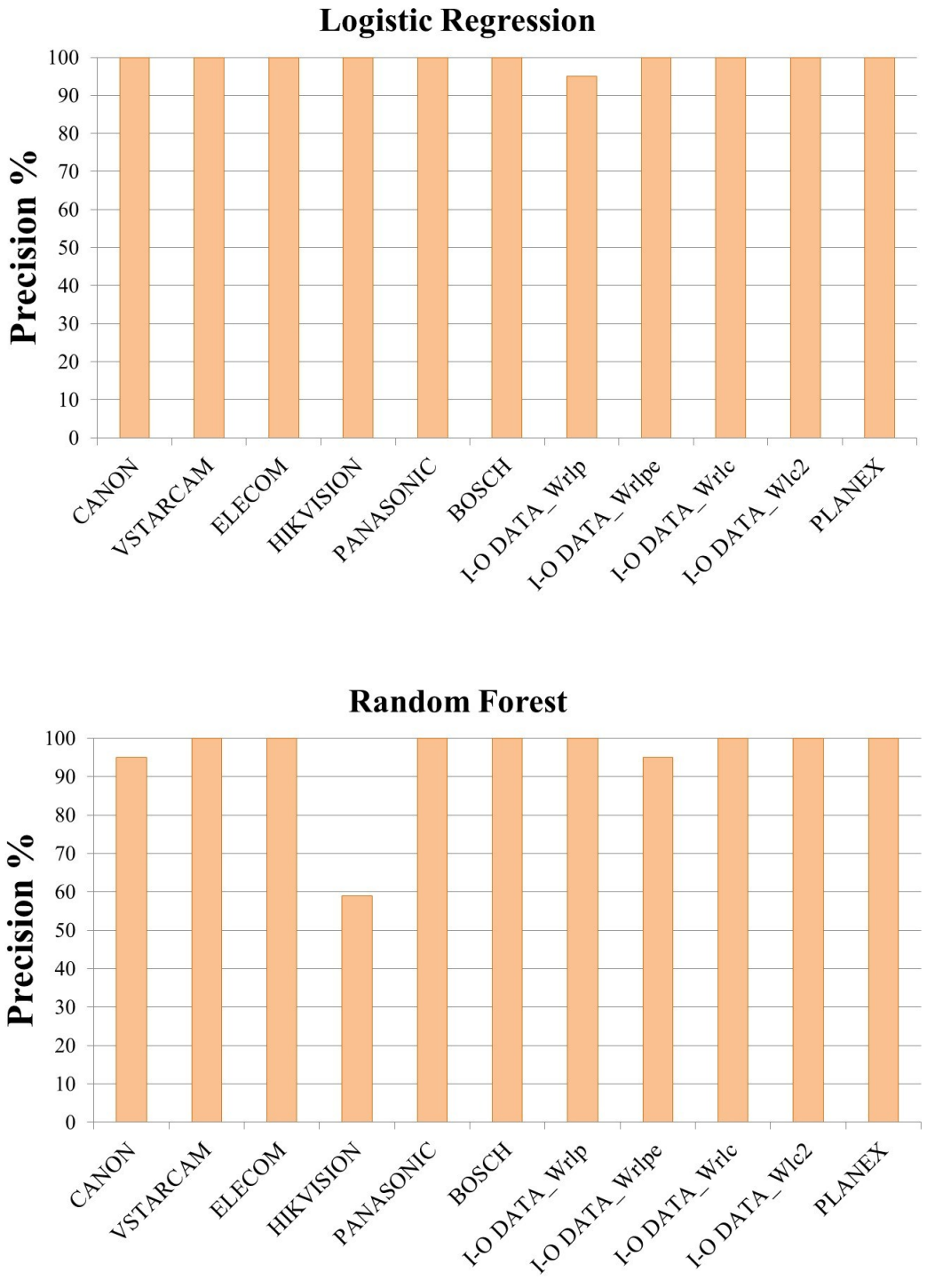


図 4.13　ヒストグラム特徴量と機械学習を用いた識別適合率（11 機種）

4.6.3 デバイス設定識別の性能評価

多種多様なサービスを E-CPS によって実現するにあたり，デバイスを操作する上で不可欠な機種情報に加えて，さらに詳細にデバイスの状態を把握できることが望ましい．デバイスの機種と共にその設定を識別できることで，デバイスの状態を迅速に把握し，システムへの組み込み，設定変更を行うことが可能になる．本項では，機種に加えて，設定情報を含む識別を行う際の性能評価結果を示す．

以下に実験の手順と条件を示す．

6 機種のネットワークカメラに対して様々な解像度とフレームレート (Frames per second, FPS) を設定し，機種と設定の組み合わせから成る 39 種類の異なるデバイスとした．デバイスの設定一覧を表 4.18 に示す．39 種類のデバイスについて，連続する 10 分間の送信パケットを収集し，特徴量を抽出した．特徴量の抽出周期およびデバイス識別の周期は，4.6.2 項に示した実験と同様，予備実験結果の知見から 30 秒とした．つまり，1 デバイス当たりに収集した特徴量数は 20 である．本特徴量を 8 対 2 の割合で学習用と試験用に分割した．

特徴量には，ヒストグラム特徴量および，比較としてヘッダフィールド値の最大値，最小値，平均値を 1 セットとしたものを用いた．ヒストグラム特徴量の条件は表 4.19 に示す通りである．また，類似度算出処理には，機械学習アルゴリズムおよび，比較として 4.4 節に示したユークリッド距離による手法を用いた．

学習用特徴量のみを用いて，ハイパーパラメータの探索手法であるグリッドサーチと 5-fold 交差検証を行い，識別正解率が最も高くなるハイパーパラメータを導出した．本ハイパーパラメータと学習用特徴量を用いて学習モデルを作成し，試験用特徴量を識別することで正解率を算出した．

表 4.18　デバイス設定識別の性能評価実験における使用デバイスと設定

Manufacturer	Model	Resolution	FPS
VSTARCAM	C7823WIP	HD (1280×720)	25, 20, 10
		VGA (640×360)	25, 20, 10
		QVGA (320×180)	25, 20, 10
I-O DATA	TS-Wrlp	HD (1280×720)	25
		400p (720×400)	20
		200p (352×200)	10
ELECOM	NCC-EWF100RWH	HD (1280×720)	30, 20, 15
		VGA (640×360)	30, 20, 10
HIKVISION	DS-2CD2032F-I	HD (1280×720)	25, 20, 10
PANASONIC	WV-SPN310	HD (1280×720)	30, 20, 10
		VGA (640×360)	30, 20, 10
		QVGA (320×180)	30, 20, 10
BOSCH	NIN-50022-A3	HD (1280×720)	30, 20, 10
		768×432	30, 20, 10
		256×144	30, 20, 10

表 4.19　ヒストグラム特徴量のビン設定（設定の識別実験）

	Bin [1]	Bin [2]	Bin [3]	Bin [4]	Bin [5]
Total length	0–300	301–600	601–900	901–1200	1201–
TTL	0–50	51–100	101–150	151–200	201–
TCP window size	0–3000	3001–6000	6001–9000	9001–12000	12001–

各実験条件における識別正解率を図 4.14 に示す．特徴量と類似度算出手法の組み合わせのうち，最も高い識別正解率となるのは，特徴量にヒストグラム，類似度算出に線形サポートベクタマシンを用いる場合であり，76.3%であった．また，最も低い識別正解率となる組み合わせは，特徴量にヘッダフィールドの値の最大値，最小値，平均値のセットを使用し，類似度算出にユークリッド距離を使用した場合であり，32%であった．

さらに，特徴量の種類に着目すると，類似度算出処理にいずれの手法を用いる場合にも，最大値，最小値，平均値をセットとした特徴量より，ヒストグラム特徴量を用いた場合の識別正解率が高く，優位であった．また，類似度算出処理に着目すると，2 種類の特徴量のいずれを用いる場合にも，ユークリッド距離を用いた手法よりも機械学習アルゴリズムを用いた手法が優位であった．

以上より，デバイスの設定識別における改善手法の優位性が示された．

さらに，機械学習アルゴリズムの比較を行った．識別正解率に関して優位であるヒストグラム特徴量を用いる条件に限定して，機械学習アルゴリズムごとの識別正解率を比較すると，最高値の線形サポートベクタマシンと最低値のロジスティック回帰との間に 10.3%の差が生じた．また，デバイスごとの識別再現率の標準偏差を比較した．図 4.15 にデバイスごとの識別再現率の標準偏差を示す．識別正解率に関して優位であるヒストグラム特徴量を用いる条件に限定して標準偏差を比較すると，最低となるロジスティック回帰と最高となるナイーブベイズとの差は 1.6 である．以上より，機械学習アルゴリズムごとの標準偏差の差は識別正解率の差と比較して微小であり，機械学習アルゴリズムの優位性は識別正解率に従うと言える．

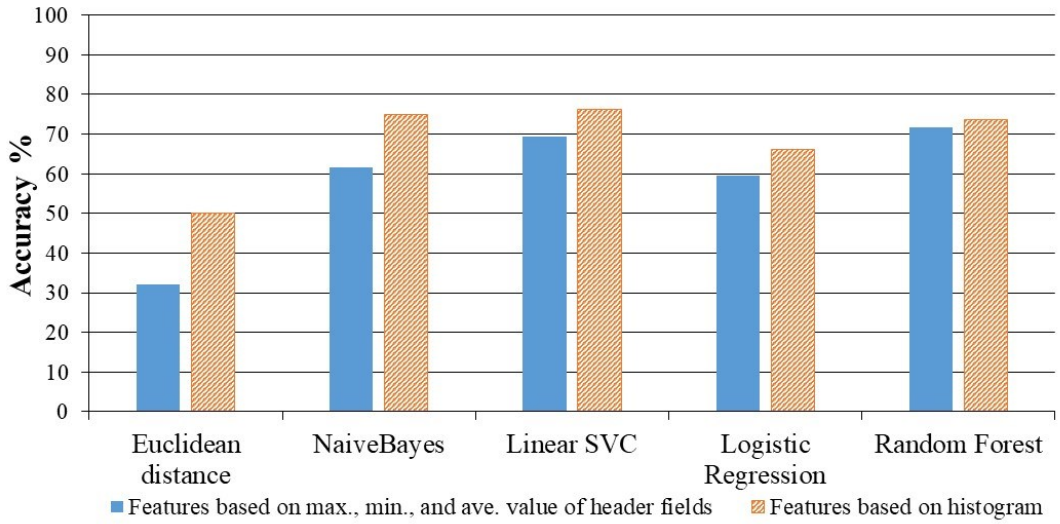


図 4.14 デバイス設定識別における識別正解率

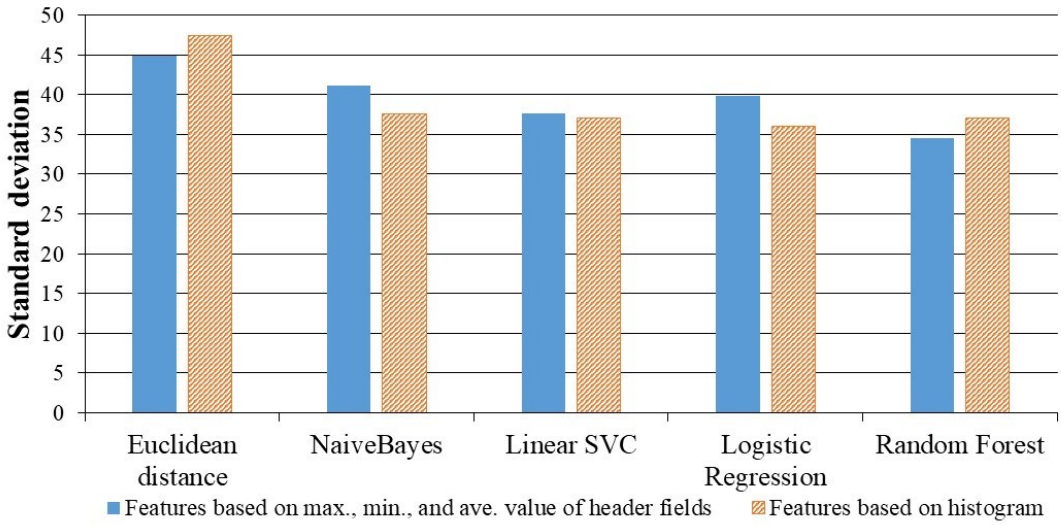


図 4.15 デバイス設定識別における識別再現率の標準偏差

各条件における機種単位の識別正解率を図 4.16 に示す．前述のデバイス設定の識別結果と同様に，いずれの特徴量を用いる場合においても，ユークリッド距離を用いる手法よりも機械学習アルゴリズムを用いた手法が識別正解率において優位であり，最大で 100%の識別正解率を示した．

一方で，特徴量に関しては，ロジスティック回帰のみ，設定識別の実験結果の傾向とは異なり，最大値，最小値，平均値をセットとした特徴量を用いる場合よりもヒストグラム特徴量の識別正解率が 0.64%低くなった．このようにヒストグラム特徴量の方が小さい識別正解率を示したのは，設定識別と合わせて合計 10 の実験条件のうち本 1 例のみであった．

以上の通り，本項に示した実験結果より，類似度算出における機械学習の優位性，および，ヒストグラム特徴量の優位性が示された．ただし，ヒストグラム特徴量については，最大値，最小値，平均値をセットとした特徴量を用いる場合よりもわずかに低い正解率となる場合が認められたため，機械学習のハイパーパラメータのチューニングを含め，今後さらに複数の条件で検証を行うことが必要と考えられる．

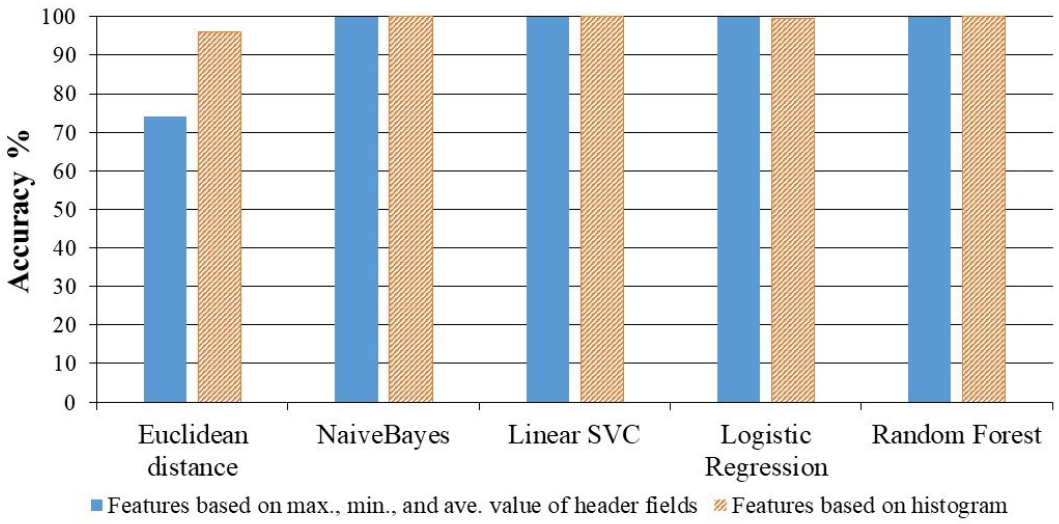


図 4.16　デバイス機種識別における各手法の識別正解率（6 機種）

4.7 識別システムの処理性能測定

提案手法の実用化観点から，デバイス識別処理の実行環境への要求条件，処理時間を明らかにするために，実験で用いた識別システムの処理性能を測定した．本システムは，4.4節に述べた手法に従い，通信情報収集機能部，特徴量抽出機能部，類似度算出機能部の3つで構成され，それぞれを独立したVMとして実装した．VMの性能は全て共通とし，CPUは5コア，メモリサイズは20 GBとした．

まず，特徴量のデータサイズを計測したところ，30秒周期で生成する1デバイスあたりの特徴量のデータサイズは，最大，最小，平均，一次近似曲線の勾配，切片を特徴量として用いる場合には約2.2 kB，ヒストグラム特徴量を用いる場合には0.6 kBであった．仮に前者の特徴量を1時間収集すると，1デバイスあたりの蓄積データサイズは約264 kBであり，10,000台のデバイスを管理すると仮定すると総和は2.6 GBと試算される．市販のハードディスク容量と比較して十分に小さい量であり，大規模なデバイスの管理においても問題がないと言える．

次に，データ収集機能部の処理性能を測定した．図4.17に秒間受信パケット数（pps (packet per second))に対する，秒間ヘッダ抽出パケット数（sps (snapshot per second) と呼ぶ）の実測結果を示す．5,840 ppsまではppsとspsが一致したが，それ以降は，ppsに関わらずspsは一定であった．つまり，本システムのデータ収集機能部がリアルタイム処理を行える最大パケット数は，5,840 ppsである．

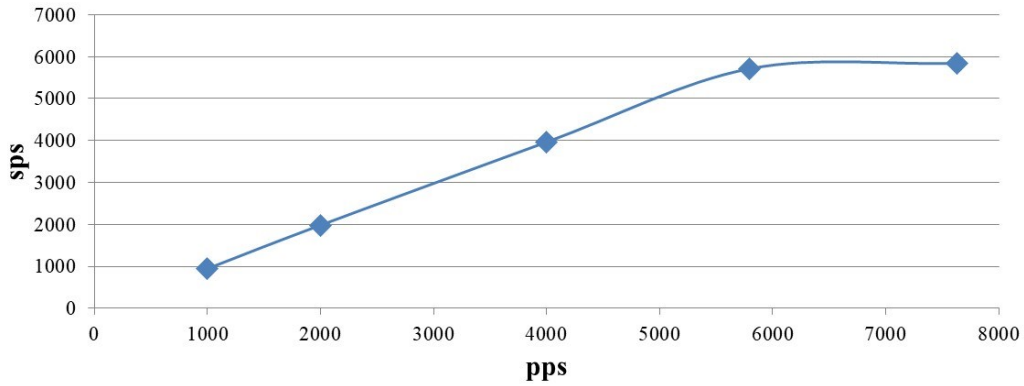


図 4.17 秒間受信パケット数（pps）と秒間ヘッダ抽出パケット数（sps）の関係

さらに，特徴量抽出機能部の処理時間を示す．図4.18は，特徴量抽出周期を10秒とした場合の，デバイス数，spsの総数，および，特徴量抽出処理時間との関係を示したものである．測定結果より，処理時間が，spsの総量ではなく，1デバイス当たりのspsに関係することが確認された．

図4.19は，特徴量抽出周期と，特徴量抽出処理時間との関係を示す測定結果である．本実験では，spsを500に固定した．特徴量抽出周期を長くすることで，特徴量抽出処理時間が長くなること，および，その増加量が正比例しないことを確認した．

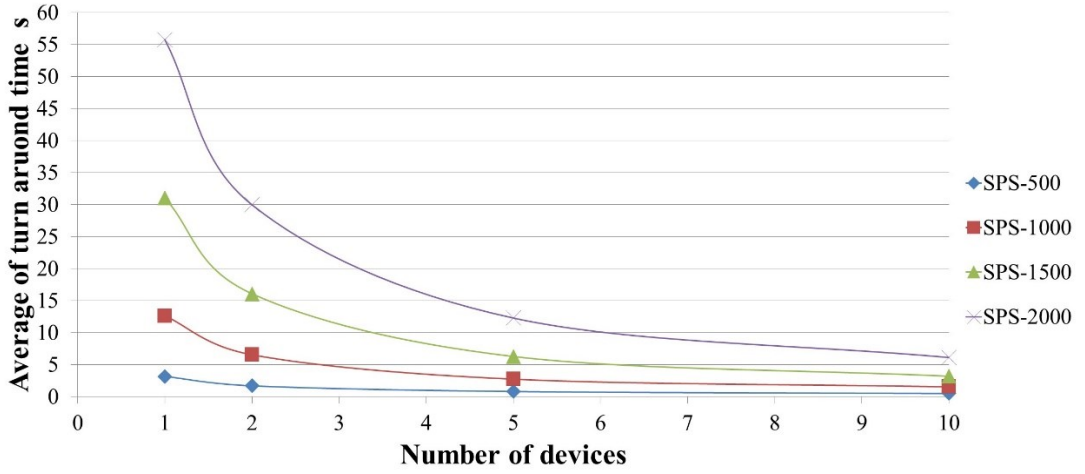


図 4.18　デバイス数，sps 総数と特徴量抽出処理時間との関係

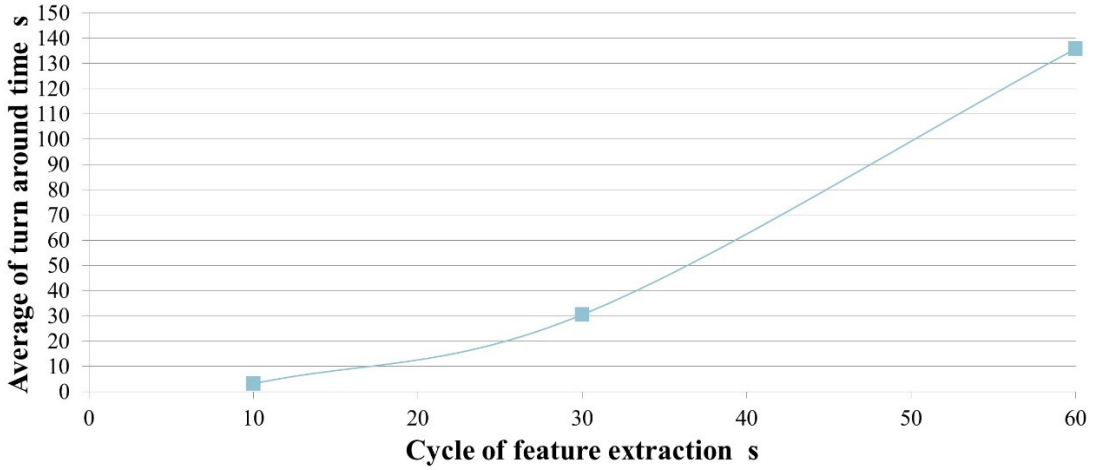


図 4.19　特徴量抽出周期と特徴量抽出処理時間の関係

最後に，類似度算出機能部の処理時間を示す．図4.20は，類似度算出処理時間と，識別候補デバイス数の関係を示す．識別候補デバイス数が2～5の場合には，処理時間にはほぼ差がみられない．一方で，デバイス数が6以上の場合には，デバイス数に応じて一定の比率で処理時間が増加した．

これは，VMのCPUコア数5に対して，それを上回るスレッドが実行されたことが原因と推測される．つまり，VMのCPUコア数と同数以下のスレッドについては，相互のスレッドが干渉せずに理想的な処理が行え，CPUコア数を増加することで処理時間を増加させずに多数デバイスの識別が可能と考えられる．

また，図4.21は，データベース内の学習用特徴量数と類似度算出処理時間の関係を測定したものである．特徴量数の増加にしたがって処理時間が線形に増加することを確認した．

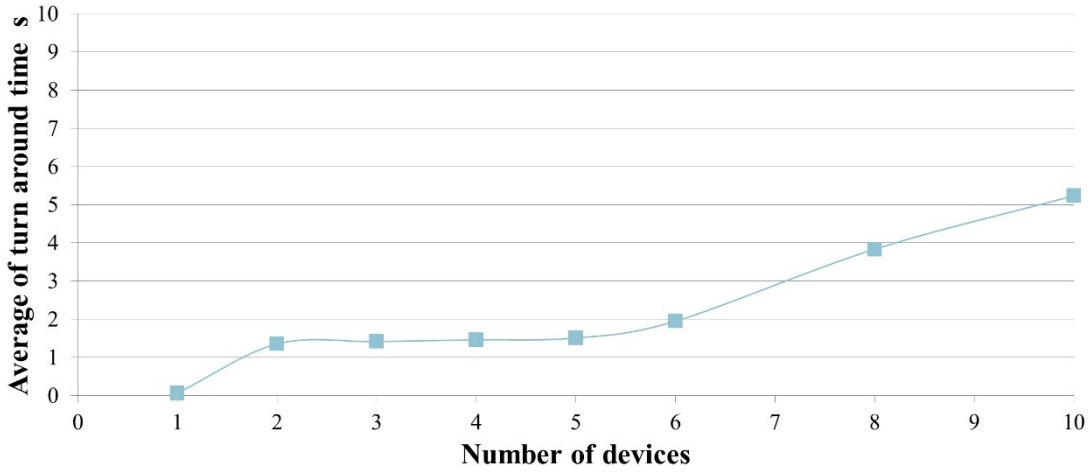


図 4.20　識別候補デバイス数と類似度算出処理時間の関係

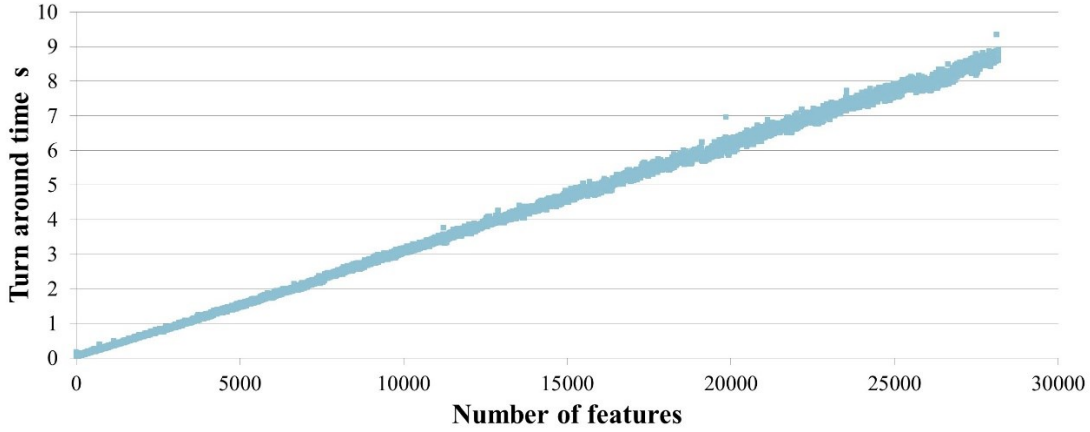


図 4.21　学習用特徴量数と類似度算出処理時間の関係

以上の結果から要点をまとめると，特徴量抽出処理に要する処理時間は，1デバイスあたりの送受信パケット数と特徴量抽出周期によるものの，例えば，特徴量周期を30秒とした場合に，500 spsの通信を行うデバイスに対して31秒で処理が完了した．また，類似度算出処理に要する時間は，実行環境のCPUコア数に依存するが，5コアのCPUを用いた場合には，10台のデバイスに対して5.2秒で処理が完了した．また，学習用特徴量数にも依存して処理時間は線形に増加し，学習用特徴量数が10,000のときに3.1秒であることから，1デバイス当たり100の学習用特徴量を蓄積すると仮定すると，デバイス100台分の学習用特徴量を上記時間で処理することが可能である．以上より，特徴量抽出処理と類似度算出処理はいずれも秒オーダの処理時間であり，実際には複数条件が相互に影響し合い，処理時間に積算されることを鑑みても，100台規模のデバイスについては，合計処理時間が数分に収まることが予想される．以上より，市販される一般的性能のコンピュータを実行環境としても，提案手法が実用上の十分な処理時間を実現できることを確認した．

また，各機能部の処理性能のスケール性に関して述べる．データ収集機能部は，デバイスごとに完全に独立した処理を行っているため，コンピュータ台数を増加することで，リアルタイムに処理できる通信量を増加することが可能である．次に，特徴量抽出機能部については，本実験システムでは本機能がシングルスレッド処理で実装されていることから，マルチスレッド化して負荷分散を測ることで性能向上が図れると想定される．さらに，類似度算出機能部については，マルチコアCPUによる並列処理が行われており，CPUコア数の増加による性能向上が図れると想定される．また，本機能部の処理時間は学習用特徴量数に応じて線形に増加することから，識別に要する時間を短く保つには，学習用特徴量を厳選し，一定量に保つことが必要である．例えば，k近傍法におけるデータ圧縮の考え方を適用し，識別に寄与しないデータを抽出して削除することが有効であると考えている．

4.8 考察と今後の課題

工場模擬環境における予備実験により，本提案手法が6種類のデバイスの内，5種類を90%以上の再現率で識別できることを確認した．また，識別性能の評価実験より，11機種のネットワークカメラを99.5%の正解率で識別できることを確認した．さらに，解像度とフレームレートが異なる6機種のネットワークカメラに関する，39種類の設定の識別正解率は76.3%，機種の識別正解率は100%であった．実験で用いたデバイスはIoTのデバイスの一例ではあるが，機種識別においては，本研究における目標である識別正解率の90%を満足しており，E-CPS実現に向けた提案手法の有効性が示された．一方で，設定の識別正解率は目標値に達しておらず，識別正解率向上に向けた今後の改善が求められる．また，前述の通り，実験で用いたデバイスはIoTのデバイスの一例であり，実環境の様々なデバイスに適用するためにも改善が必要になる．

今後の課題の一つとして，最大の性能を引き出す適切な特徴量抽出周期の設定が不可欠である. 全てのデバイスがカメラのように常時, 再現性があるデータ送信を行うわけではなく，高頻度な通信を行わないデバイスがある．また，本来用途の通信とは別に監視プロトコルのパケットを送受信するデバイスもある. 実験結果より, 特徴量抽出周期を一律に固定すると，それよりも長い周期で発生するパケットの影響を受け，識別精度の低下を引き起こすことを確認した．提案手法をカメラ以外のさまざまなデバイスに適用するには，各環境やデバイスに応じた適切な特徴量抽出周期を設定する必要がある．今後は，特徴量抽出周期を動的に調整する方法を検討する予定である．

また，識別のために適切なヘッダフィールドを選択することも重要な課題である．識別再現率，識別正解率の向上には，ユニークかつ再現性が高いヘッダフィールドのみを識別に使用する必要がある．IoTデバイスのプロトコルと使用環境は多種多様であるため，有用なヘッダを人手による検証ではなく自動的に選択できる必要がある．機械学習におけるハイパーパラメータのチューニングツールのように，探索的な試行によって識別に有用なヘッダフィールドの組み合わせを自動抽出する手法が有効である．

最後に, 提案手法を広域ネットワークの通信機器に実装する際には, NAPT (Network Address and Port Translation) を介したパケットや暗号化されたパケットを用いたデバイスの識別も不可欠な課題である．NAPTを介すると，複数のデバイスが同一のグローバルIPアドレスを用いて通信を行うことになるため，IPアドレス以外の情報から個々のデバイスを区別して識別を行うことが求められる．対応案の一つとして，ポート番号ごとに通信を区別し，各ポートに対応するアプリケーションの種類を識別した上で，その結果を用いてデバイスを推定する方法がある．また，暗号化されたパケットに関しては，IPsec (Security Architecure for Internet Protocol) のトンネルモードのように，パケット全体を暗号化するものについては提案手法の適用対象外であるが，TLS (Transport Layer Security) のような上位層のヘッダフィールド情報のみを暗号化するプロトコルについては，下位層のヘッダフィールド情報を特徴量として扱えるため提案手法の適用対象であり，限定された通信情報だけを用いて高性能な識別を行うために手法の拡張が不可欠である．

4.9 第 4 章のまとめ

本章では，E-CPS の実現に向けた技術課題の一つである，ネットワーク内の多種デバイス識別に取り組み，デバイスが送受信する通信情報を分析することでデバイスの種類や機種を識別する手法を提案した．

本手法は，通信情報の収集，通信特徴量の抽出，類似度の算出という 3 段階の処理で構成され，デバイスの種類やネットワーク環境に応じて，各処理を個別拡張できることを特徴とする．識別性能評価に向けた予備実験では，まず，ヘッダフィールドごとの識別に対する有用性を検証した．基本的には扱うヘッダフィールドが多いほど情報量が多くなり識別に優

位だが，ユニーク性と再現性が低い情報が混ざることで，特定のデバイスの識別再現率が低下する事象も認められた．また，実際の IoT 環境におけるフィージビリティ検証として工場模擬環境における識別では，製造ラインに組み込まれた 6 種類のデバイスのうち 5 種類を 90%以上の識別再現率で識別した．識別性能評価の本実験では，11 機種のネットワークカメラを 99.5%の正解率で識別できることを確認した．さらに，解像度とフレームレートが異なる 6 機種のネットワークカメラに関する，39 種類の設定の識別正解率は 76.3%，機種の識別正解率は 100%であることを確認した．また，実験結果より，ヒストグラム特徴量と機械学習アルゴリズムが識別正解率の向上に有効であることを確認した．なお，実験ごとに最も高い正解率を示した機械学習アルゴリズムが異なることから，識別対象デバイスに応じて，適切なものを選択することが必要と言える．

識別システムの処理性能測定では，提案手法における 3 段階の処理に関する性能限界や処理時間の実測値を示した．市販される一般的性能のコンピュータを実行環境としても，提案手法が 100 台規模のデバイスを数分で識別できる可能性があること，および，実行環境のコンピューティングリソースを増強することで，識別時間を一定に保ちつつ，対応するデバイスの数を増加できる可能性が示された．

最後に，提案手法の識別性能向上と様々なデバイスへの適用に向けた今後の課題として，通信特徴量抽出における，適切な周期の設定とヘッダフィールドの選択，および NAPT や暗号化パケットへの対応を挙げた．

本提案手法は，機種や設定を通知する機能を持たないデバイスへの適用を前提としたものであり，また，使用環境やネットワーク環境に依らず再現性がある通信を行うデバイスに対して有効な手法である．このような性質を持つデバイスには，例えば，ネットワークカメラや測域センサといったセンサ類，工場等に配備される製造機器などの単機能デバイスが挙げられる．特に，ネットワークカメラはセンサであると同時にアクチュエータとしての性質も有するものがあり，例えば，可動機構を有し，遠隔から首振りやフォーカスを行えるものがある．このようなネットワークカメラは，E-CPSが実現する多様なサービスへの利用が期待されるものであり，提案手法の適用対象である．また，提案手法が前提とするネットワーク環境は，通信パケットからデバイス固有のヘッダ情報が得られる環境である．IPsec等のパケット全体を秘匿するプロトコルについては対象としていない．今後，このような通信が主流となった折には，ホームゲートウェイ等のデバイス所有者のインターネット接続機器に対して本提案手法の機能を搭載し，秘匿化以前のパケットを収集することを考えている．このように通信事業者が提供するデバイス近傍の装置に識別機能を搭載することで，広域ネットワークへのデータ漏洩を防ぎつつ，安全性を満たしたデバイスの共有が行えると考えている．なお，このような実装形態は，第3章で述べたデバイス近傍のエッジコンピュータを用いた負荷分散のアーキテクチャにも従うものである．

第5章 多様な組み合わせのデバイスの自律制御

5.1 はじめに

本章は，E-CPS の実現に向けた技術課題の一つである，ネットワークに接続される多様な組み合わせのデバイスに対する自律制御を取り上げ，機械学習と動的な調整によって，事前の設計や設定無くデバイスを自律的に制御する手法を提案する．はじめに，デバイス制御に関する詳細な要件を整理し，本研究のアプローチを示す．その後，提案手法を述べたうえで，シミュレーションと実機を用いた評価実験結果を示し，最後に考察を述べる．

5.2 要件とアプローチ

本章で述べるデバイス制御は，前章までに述べたセンサデータの収集結果および，ネットワークに接続されているデバイスの識別結果を利用する．共用デバイスは，API や外部インタフェースを E-CPS の各機能に公開し，遠隔から自由にアクセス，操作可能な状態にあることを前提とする．E-CPS における，デバイス制御の詳細要件を以下に示す．

・要件 1：デバイスを意識した設計の排除

第 1 章で述べた通り，自律性と柔軟性が E-CPS の要件である．システムを構成するデバイスの設置場所や詳細仕様等をサービス事業者が意識することなく，自律的に適切なデバイス制御が行われることが求められる．

・要件 2：多様なサービスに対する汎用性

E-CPS の対象サービスは多種多様であり，サービスごとに専用の制御論理を準備することはサービス事業者への負担が大きく，E-CPS の普及を阻害する．したがって，汎用的な制御論理とサービス依存機能を独立に拡張できることが求められる．

・要件 3：デバイス変動への適応

共用デバイスは，所有者の使用状況に応じて，システム運用中にネットワーク切断，接続が発生する．このようなデバイス変動に柔軟に適応できることが求められる．例えば，システムを構成するデバイスが減少した場合には，代替のデバイスを発見して処理を引き継ぐこと，反対にデバイスが増加した場合には，サービスに，より有効なデバイスを再選択して制御することが必要である．

・要件 4：システム運用コストの最小化

第 1 章で述べた通り，経済性は E-CPS の要件の一つである．サービスに必要な最小限のデバイス数と出力でシステムを運用する必要がある．過剰なエネルギー消費を抑えること，また，特定のサービスによる必要以上数のデバイスの占有を防止することが求められる．

自律制御の対象となるデバイスには，センサ，アクチュエータ等がある．これらの制御に関して，以上の要件を扱う研究分野がある．

1 つ目は，WSAN (Wireless Sensor and Actuator Network) [91] [92] [93]である．WSAN とは，WSN (Wireless Sensor Network) を起源とする研究分野である．WSN は，ネットワークに接続されている大量のセンサを用いたデータ収集に関する研究分野であり，WSN にアクチュエータ制御を加えたものが WSAN である．WSAN の主要な研究目的は，バッテリー容量や性能が限られた複数小型デバイスに対する，デバイス間通信の効率化と高信頼化である．大量のセンサ，アクチュエータが分散している環境において，それらの作用が漏れなく，重複なく，環境内の全領域を網羅することを目指している．例えば，Chen 等[91]は，アクチュエータとセンサの対応関係を予測し，アクチュエータの適切な作用範囲と出力量を導出する方式を提案している．また，センサとアクチュエータがネットワークに動的に追加されることにも言及している．このような WSAN の機能は，前述の要件 1，要件 3，要件 4 を満たす．一方で，現状の WSAN の研究の多くは，要件 2 の汎用性を満たすことができない．単機能のアクチュエータを対象としており，センサとアクチュエータの対応関係を単純な規則で定義して制御するため，ロボットの移動や，多種多様なデバイスが連携する複雑なシステムに対応するには，現状，システムごとに制御則を設計する必要がある．

2 つ目は，機械学習を用いたデバイス制御である．機械学習は，複数のデータ間の相関関係を，データ指向のヒューリスティックな方法で獲得することができる．CPS に適用することで，センサデータから判断される環境情報と，その環境に応じた適切なデバイス制御値を統計的な計算処理によって直接結び付けることができる．つまり，デバイスやサービスの性質を理解して制御則を考えることが不要であり，要件 1 を満たす．また，カメラが生成する画像データ，温度センサが生成する温度データといった様々なデータに対して，コンテキストを廃して，全て一律に数値化した情報として扱うことが可能であり，要件 2 の汎用性も満たす．要件 3 については，環境とデバイスの関係性を実績に基づいて機械的に算出することで，個々の環境に応じた適切なデバイスと，その制御値を自動的に導出できる可能性がある．要件 4 についても，コストとデバイス制御値との関係を機械的に算出することで，コストを最小化する制御値を自動的に算出できる可能性がある．

以上より，本研究では，機械学習を用いることで自律的，汎用的なデバイス制御を実現するアプローチをとる．なお，本アプローチは，WSAN と融合できるものであり，双方の技術を盛り込んだシステム実装が可能である．次節に，機械学習を用いるデバイス制御の関連研究と，本研究に固有な課題を示す．

5.3 機械学習を用いたデバイス制御の関連研究

機械学習を用いて，大量のセンサ情報からフィジカル空間の状態を把握し，デバイスの制御値を最適化する研究例がある．例えば，Gao 等[94]は，機械学習の手法の一つであるニューラルネットワークを用いて，データセンタの 2 年間のコンピュータ負荷や周囲の温度，空調機の制御状態をもとに，電力消費を最小化する空調制御を実現した．また，センサ，アクチュエータで構成される代表的なシステムはロボットである．強化学習は，複雑環境下におけるロボットのモーションプランニングに適用される．例えば，Kober 等[95]は，強化学習を用いて，ロボットアームによる卓球を可能にした．これらの手法は，大量の試験データを用いることを前提としているため，E-CPS には，単純には適用できない．E-CPS は，ネットワークに接続されている大量のデバイスを候補に構成されるため，事前に全ての組み合わせに対する試験データを収集しておくことはできない．

本研究では，機械学習を利用しつつ，少数の試験データのみで適切なデバイス制御を行うことを目指す．一般的な機械学習の利用目的がシステムの厳格な最適化であることに対して，本研究の最優先目的は，E-CPS を迅速に構成することである．本目的に基づき，システムの最適性と試験データ量とのトレードオフを解決するため，サービスの達成を定義し，少数のデータのみでデバイス制御を行うことを目指す（図 5.1）．

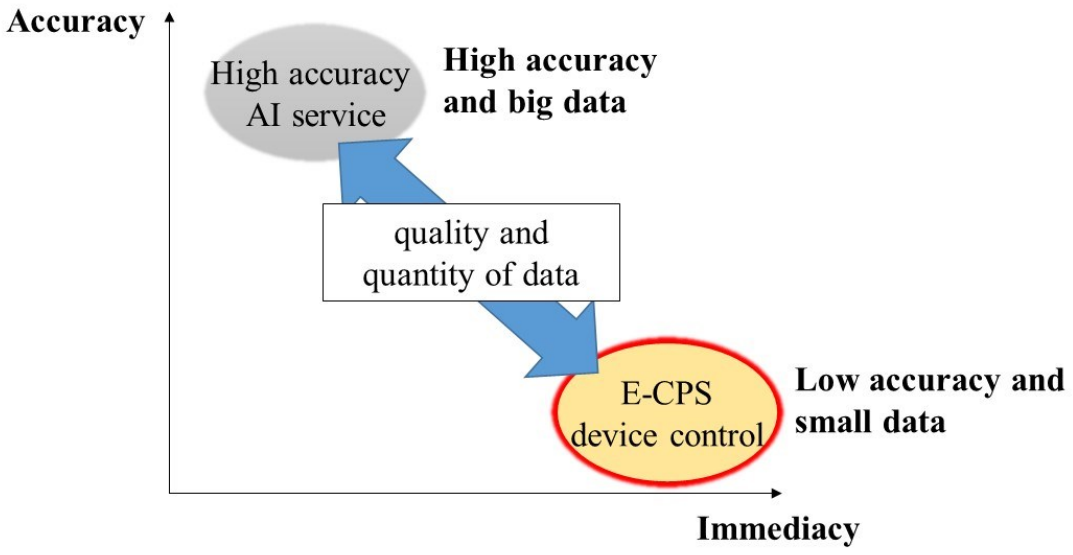


図 5.1 本研究が目指すデバイス制御

5.4 デバイス自律制御に関する提案手法

本節は，E-CPS に求められるデバイス制御を実現するために，事前の設計や設定無く，システムを最適な状態に保つ，機械学習を用いたデバイス自律制御手法を提案する．本節で述べる，「最適な状態」とは，システムが最小コストでサービス目的を達成している状態である．本手法は，設置場所や種類が様々なデバイスを組み合わせたシステムにおいて，サービス目的の達成に必要なデバイス制御値が自明でない状況に対応するためのものである．また，本手法は，サービス目的の達成度合いとデバイス制御値が線形関係を持つシステムを対象とする．以降に，はじめにシステム状態に関する本研究の定義を述べ，その後，デバイス自律制御手法を述べる．

5.4.1 システム状態の定義

本研究におけるシステム状態は，以下の二つの要素で表される．

A. サービス目的達成スコア

サービス目的達成スコア（以下，スコアと呼ぶ）は，サービスの目的がどの程度達成されているかを定量的かつ，共通的に表す指標値である．実際には，サービスごとに目的達成程度を表す指標は異なるため，サービスに対応する特定のソフトウェアとセンサを用いてスコアを算出する．

例えば，避難誘導サービスの場合には，スコアは避難が完了したエリア数に比例する値とし，人物画像認識ソフトウェアとカメラを使用して測定，算出する．また，別の例として，屋内照明を制御するオフィス IoT サービスでは，スコアは特定の場所の光量と視野の鮮明さに比例する値とし，光センサまたはカメラを使用して測定，算出する．

これらのスコアを算出するソフトウェアは，サービスの構成要素の一つとして 2.2 節に述べたコンテキストレイヤのサービスカタログに登録される．サービスごとのスコアの意味は，サービス事業者に公開され，SLA (Service Level Agreement) などの形態で目標スコアが設定されることを想定している．例えば，避難完了人数をスコアとする避難誘導サービスを例にすると，全員の避難完了を保証する必要がある場合には，最大スコアを目標スコアとして設定する．また，別の例として，照度をスコアとする照明サービスでは，多少の暗さと不便を許容できる場合には，後述するシステム運用コストを優先して目標スコアを低く設定することも可能である．

B. システム運用コスト

本研究では，システム運用コストを，システムを構成するアクチュエータの制御値に比例するものと定義する．理想的には，システムの総消費電力が直接コストを表す指標であるが，

デバイスの消費電力を測定できる環境は限られるため，アクチュエータ制御において必ず扱う情報であるアクチュエータ制御値を代替として用いる．例えば，照明デバイスの光量やスピーカーの音量といった出力ボリュームがアクチュエータ制御値である．なお，本研究では，センサの消費電力はシステムの運用コストに含めない．これは，第 3 章に述べた通り，センサは特定のサービスに関わらず，常に稼働していることが前提であるためである．

式 5.1 をシステム運用コストの算出式として定義する．本式において， i はシステムを構成するアクチュエータの番号，val はアクチュエータの制御値を 0〜100 に正規化した値である．本式によって算出される，システムを構成する全てのアクチュエータの制御値の総和をシステム運用コストとする．

$$\mathrm{Cost} = \sum val_i \qquad (5.1)$$

5.4.2 デバイス自律制御手法

本項では，前項に述べたサービス状態に基づいて，システムを構成するデバイスを自律制御する手法を述べる．

本手法は，図 5.2 に示す，初期制御値の算出と制御値の調整という二段階の処理で構成される．一段階目の処理では，機械学習を使用して初期制御値を取得し，二段階目の処理では，非線形最適化問題を解く手法の一つである進化戦略[96]に従ってデバイス制御値を適切な値に調整する．進化戦略の具体的な適用形態については後述する．このような二段階の構成を取る理由は，環境の差異と外乱に対する頑強性を獲得するためである．照明装置の制御を例とすると，適切な照明の明るさは，人の立ち位置と自然光に依存する．サービスに影響するあらゆる要因を全てセンシングし，綿密な学習モデルを作成しない限り，機械学習から得られたデバイス制御値が常に最良であることは保証できない．E-CPS はネットワークに接続された不特定多数のデバイスを用いて迅速に構成される必要があるため，綿密な学習モデルを作成することができない．したがって，サービス実行中のデバイス制御値の調整を不可欠とした．

一定時間の経過，一定の調整回数の到達，といった特定のタイミングで二段階目の処理は中断し，一段階目の処理から再開する．サービス実行中に補充されるデータを用いて学習モデルが更新されるため，時間経過とともに，より適切な制御値が初期制御値として選択され，サービス目的達成に至るまでに必要な調整回数が減少する．

以上の通り，学習による初期制御値算出と，進化戦略に基づく制御値調整を組み合わせることで，デバイスと環境の変化に柔軟に対応することが可能になる．さらに，制御値調整では，サービスを満たす最小のデバイス制御値を探り，コスト削減を行う．提案手法は，従来の機械学習のように大量のデータを事前に準備する必要がなく，E-CPS を稼働しながら必要最小限のデータ収集を行う．処理の詳細を以下に示す．

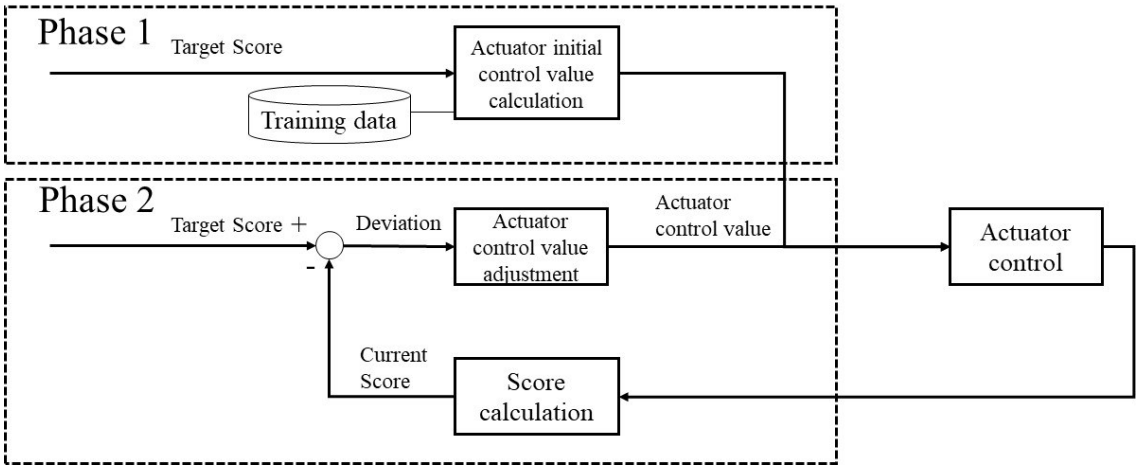


図 5.2　二段階構成の制御値算出処理の流れ
(©IEEE　主著論文 1 の Fig. 3 を一部修正)

A. 初期制御値の算出（一段階目処理)

本処理は，デバイスの設置位置と作用範囲を直接知ることができない状況において，適切な制御値を算出する．以下，環境に作用するデバイスをアクチュエータと表記する．アクチュエータの作用は設置位置によって異なり，また，スピーカーや照明のように作用が重なり合うものもある．このように複雑な，アクチュエータ制御値とスコアの関係を環境ごとに定式化することは困難である．したがって，機械学習アルゴリズムの一つであるニューラルネットワークを使用する[97]．ニューラルネットワークを使用することで，アクチュエータ制御値とスコアの関係を表す学習モデルを機械的なデータ処理で作成し，目標スコアを満たすアクチュエータ制御値を予測できる．本手法は，以下のニューラルネットワークを用いる．

[ニューラルネットワーク条件]
・入力データをスコアとし，その次元数は使用するセンサ数と同一とする
・出力データをアクチュエータ制御値とし，その次元数はアクチュエータ数と同一とする

つまり，センサ数が N，アクチュエータ数が M である場合に，学習データは，N 次元のデータと M 次元のデータのペアで表現される．学習データは，二段階目の処理である制御値調整において常時蓄積する．本ニューラルネットワークに目標スコアを入力すると，対応するアクチュエータ制御値が出力される．

CNN (Convolutional Neural Network) や RNN (Recurrent Neural Network) 等の深層学習[98] [99] [100]に基づくニューラルネットワークは近年の主流であり，多くの研究が行われている．しかしながら，本提案手法では，最も単純なニューラルネットワークである 3 層の MLP (Multilayer perceptron) を使用する．これは，一つのサービスが使用するセンサとアクチュエ

ータの数は多くとも 100 程度であると想定しているためである. 深層学習ベースの手法は，数千を超えるパラメータを処理して正確な最適解を導き出すことが可能だが，多大な計算コストと時間がかかる．提案手法は，後段に制御値の調整を行うため，厳密な最適解よりも短時間で行える軽量の処理が適していることから，単純なニューラルネットワークを用いた．

B. 制御値の調整 (二段階目処理)

本処理の目的は，初期制御値に従ったアクチュエータ制御値では目標スコアが達成できなかった場合，または，アクチュエータの過剰なボリュームを抑制することでシステムコストを削減できる場合に，アクチュエータ制御値を最適化することである．

本処理は，目標スコアと実際のスコアの差が大きいほど大きな調整を行うものとする．なぜなら，これらの差が大きいということは，環境の変化もしくは，学習データ不足によって，初期制御値が有効に働いておらず，大きな調整が必要であると判断できるためである．反対に，差が小さい場合は，現在の状態を乱さない微弱な調整を行う必要がある．このような思想に基づいた調整を進化戦略アルゴリズムの一つである(1+1)-ES アルゴリズムによって実現する．これまでに示した通り，本手法は，サービスに対応するデバイスの制御則が与えられない状況を前提としているため，確率的な選択により最適化を行う進化戦略を用いる．

式 5.2 は，(1+1)-ES アルゴリズムを表したものである．本アルゴリズムは，現在状態に対する増減値を確率的に選択することで，最適解を探索するものである．本手法では，式 5.2 において，x_iに調整前の制御値，x_i'に調整後の制御値を当てはめて用いる．N は σ を分散とする正規乱数である．正規乱数 N によって，σ が大きいほど，制御値の増加値または減少値として大きい値の選択確率が高くなる．

本提案手法は， σ の値をシステム状態に応じて変化させる．式 5.3 において，d は，センサごとの目標スコア（*S_tar(i)*）と現在スコア（*S_cur(i)*）の差の平均を示す．*d_max* は，目標スコアと現在スコアの最大の差，*σ_max* は最大の分散を示し，これらは，システムごとに任意の値を設定する．式 5.4 における，Δi は，各センサから取得した *S_tar* と *S_cur* の差である．式 5.5 における，D は，Δ の正の値の集合を示す．式 5.6 における h は D の要素数である．式 5.6 を用いて，目標スコアに到達しないセンサについてのみ，*S_tar* と *S_cur* の平均差を計算する．以上の式によって，目標スコアと現在スコアの差である d が大きいほど，正規乱数 N における分散 σ が大きくなり，前述の思想に従ったデバイスの制御値の増減量が確率的に選択される．

$$x_i' = x_i + N(0, \sigma^2) \quad (5.2)$$

$$\sigma = \left(d/d_{max}\right) \times \sigma_{max} \quad (5.3)$$

$$\Delta_i = S_{tar(i)} - S_{cur(i)} \quad (5.4)$$

$$D = \{\Delta_i \mid \Delta_i > 0\} \quad (5.5)$$

$$d = \frac{1}{h}\sum(\Delta_i \in D) \quad (5.6)$$

以上の調整処理を，目標スコアを達成するまで繰り返し行う．一回の調整によって *d* が大きくなった場合には，前回のアクチュエータ制御値に戻り，試行を繰り返す．調整を繰り返していくことで，*d* が小さくなり，最終的に *S_tar(i)*に *S_cur(i)* が到達する．

さらに，本提案手法は，目標スコアを達成した後に，同様の計算式を使用してアクチュエータ制御値を調整することによって，コスト削減を試みる．前節で述べた通り，コストはアクチュエータ制御値の合計である． *S_tar(i)*と *S_cur(i)*の負の平均差を使用して，目標スコアの達成を維持しながら，より少ないアクチュエータ制御値を探索する．調整を繰り返すことで，過剰なアクチュエータ制御が減少し，最終的に必要最小限の制御値になることが期待できる．

ここまでに述べた，アクチュエータ制御値の算出，調整アルゴリズムと，その一部であるコスト削減アルゴリズムの擬似コードを以下に示す．

Algorithm 1 Actuator control algorithm

S_tar: target Score
S_cur: current Score
val: actuator control value
n : the number of sensors to be set S_tar
m: the number of actuators to be set S_cur
d_max: the maximum deviation
σ_max: the maximum variance
k: the number of adjustment

```
Repeat
    Obtain val[m] by using neural network with S_tar[n] as input
    for loop = 1 to k
        Obtain S_cur[n] by analyzing sensors data
        Initialize d, and h to zero
        for counter = 1 to n do
            if S_cur[counter] < S_tar[counter] then
                d ← d + S_tar[counter] – S_cur[counter]
                h ← h + 1
            end if
        end do
        if d_pre < d then
            d ← d_pre
            h ← h_pre
            val[] ← val_pre[]
        end if
        d ← d/h
        σ = d/d_max *σ_max
        for counter = 1 to m do
            Obtain x by normal random with σ as the variance
            val[counter] ← val[counter] + x
        end do
        d_pre ← d
        val[] ← val_pre[]
        k ← k + 1
        Only when d = 0, execute cost reduction algorithm (Algorithm 2)
        Control actuators using val[m]
```

```
28:             end loop
29:     until service is ended
```

Algorithm 2 Cost reduction algorithm

```
1:      for counter = 1 to n do
2:          if S_tar[counter] < S_cur[counter] then
3:              d ← d + S_cur[counter] – S_tar[counter]
4:              h ← h + 1
5:          end if
6:      end do
7:      d ← d/h
8:      σ = d/d_max *σ_max
9:      Repeat
10:         for counter = 1 to m do
11:             Obtain x[counter] by normal random with σ as the variance
12:         end do
13:     until sum of x[] < 0
14:     val[counter] ← val[counter] + x[counter]
```

5.5 提案手法の評価実験

シミュレーションおよび，実機デバイスを用いた提案手法の評価を実施した．シミュレーションは，複数センサとアクチュエータの混在環境における提案手法の有効性確認を目的とし，実機デバイスを用いた実験は，実サービスを想定したシステム設計と提案手法のフィージビリティ確認を目的とした．

5.5.2 シミュレーション実験

シミュレーション実験の目的は，複数センサとアクチュエータの混在環境において，提案するデバイス自律制御手法による，目標スコアの達成可否を確認することである．以下に，シミュレーションの仕様と実験結果を示す．

A. シミュレーションの仕様

センサとアクチュエータ，それぞれの作用範囲をシミュレーションモデルとして定義した．作用範囲とは，カメラの視野や照明装置の光の到達範囲といったカバレッジを意味する．シミュレーションモデルは，エアコン，照明，スピーカーといった，作用に重ね合わせがあるものを対象として，次の規則に従って作成した．

[シミュレーション規則]

- センサから計算されるスコアの値域とアクチュエータ制御値の値域は 0〜100 とする
- 全てのアクチュエータは，2 つのセンサに影響する
- 全てのセンサは，同数のアクチュエータから，均等に影響を受ける
- アクチュエータの効果は重ね合わされ，関連する全てのアクチュエータのボリュームが最大のときにスコアは 100 になる

上記の規則に従うと，例えば，5 台のセンサと 5 台のアクチュエータがある場合に，配置は図 5.3 のようになる．表 5.1 は，センサごとのスコアに対するアクチュエータの影響量を 0〜1 の範囲で示したものである．例えば，アクチュエータ#1 の制御値が 100，アクチュエータ#5 の制御値が 50 の場合，センサ#1 のスコアは，それぞれの制御値に影響量 0.5 をかけ合わせたうえで合計した値である 75 となる．また，別の例として，5 台センサと 10 台のアクチュエータがある場合には，各センサは 0.25 ずつの影響量で 4 台のアクチュエータに関連付けられる．本実験は，提案手法の基本的な効果を確認することが目的であるため，このような均等な配置を用いた．センサとアクチュエータの設置位置は固定されており，実験中に関係性が変化しないこととした．

表 5.1 センサ数 5，アクチュエータ数 5 の条件における相関

		Actuator				
		#1	#2	#3	#4	#5
Sensor (Score)	#1	0.50	0.00	0.00	0.00	0.50
	#2	0.50	0.50	0.00	0.00	0.00
	#3	0.00	0.50	0.50	0.00	0.00
	#4	0.00	0.00	0.50	0.50	0.00
	#5	0.00	0.00	0.00	0.50	0.50

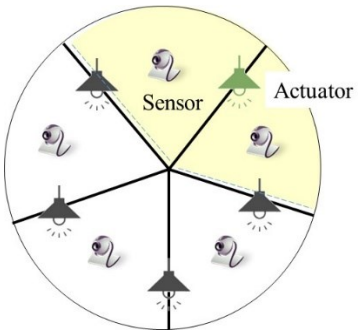


図 5.3 センサ数 5，アクチュエータ数 5 の実験配置
(©IEEE 主著論文 1 より引用)

以上のシミュレーション規則に従い，アクチュエータ数が 5，10，15，センサ数が 5 の実験条件を設定した．シミュレーションには，センサと目標スコアを指定する．これらのセンサ，アクチュエータの関係は，シミュレーションモデルの条件であり，評価対象である提案手法にとっては不明なものである．初期制御値を算出するニューラルネットワークの実装には，scikit-learn の MLP Regressor を使用した．ニューラルネットワークのハイパーパラメータはデフォルト値を用いた．なお，全ての処理は，CPU が 2 コア，メモリサイズが 8 GB の VM 上で実行した．

B. 実験項目と実験結果

(i) 目標スコア達性に関する評価

提案手法により，目標スコアを達成するアクチュエータ制御値の導出が行えることを確認するために，目標スコア達成に要した調整回数を測定した．センサは，センサ#1，#2 に目標スコア 80 を指定する場合，および，センサ#1，#2，#3 に目標スコア 80 を指定する場合の 2 通りの条件を設けた．アクチュエータ数は 5，10，15 の 3 通りの条件を設けた．以上のセンサ数とアクチュエータ数の組み合わせから成る合計 6 通りの条件について各 1 回ずつの実験を実施した．

連続する 20 回の試行を行い，それぞれ初期制御値算出からアクチュエータ制御を実行した．初期制御値算出には，それまでの試行で蓄積した学習データを用いた．つまり，後続の試行では，学習データが蓄積されて，より最適化された初期制御値が選択されることになる．1 回の試行は，現在スコアが目標スコアに到達するか，調整回数が 20 回に到達する時点で終了とした．なお，初期制御値を算出するための学習データの必要最小数を 5 とし，学習データ数がそれ未満の場合には，全てのアクチュエータの初期制御値を 50 とした．本実験では，コスト削減の調整処理は省略した．

実験結果を以下に示す．

図 5.4 に，センサ#1，#2 に目標スコア 80 を指定した場合の各試行と調整数の推移を示す．横軸は 1～20 番目までの実行順に並べた各試行，縦軸は調整回数である．アクチュエータ数ごとに全試行の調整回数の平均をみると，アクチュエータ数が少ない条件から順に，3.0，5.8，7.1 であった．また，20 回未満の調整で目標スコアを達成した試行の割合は，同様の順に 100%，95%，90%，10 回未満の調整で目標スコアを達成した試行の割合は，95%，80%，80%であった．

また，図 5.5 に，センサ#1，#2，#3 に目標スコア 80 を指定した場合の各試行と調整数の推移を示す．同様にアクチュエータ数ごとに全試行の調整回数の平均をみると，アクチュエータ数が少ない条件から順に，5.3，7.0，8.2 であった．また，20 回未満の調整で目標スコアを達成した試行の割合は，同様の順に 95%，90%，95%，10 回未満の調整で目標スコアを達成した試行の割合は，90%，75%，75%であった．

以上の通り，全実験条件において，90%以上の試行が 20 回未満の調整回数で目標スコア

を満たすアクチュエータ制御値の導出が行えることを確認した.

なお, アクチュエータおよび目標スコアを与えるセンサの数が多いほど, 調整回数の平均が大きくなったのは, 関連付けられているアクチュエータが多いほど, 調整によって最適な制御値の組み合わせを獲得できる可能性が小さくなるためと推測されるが, 実験回数が少ないことから本仮説は検証できていない.

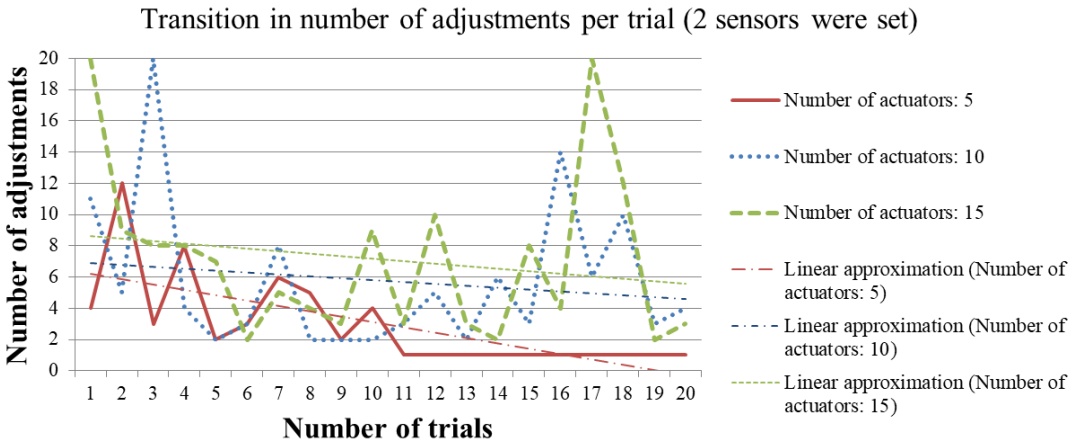


図 5.4 アクチュエータ数ごとの試行回数に対する調整回数の推移（センサ数 2）

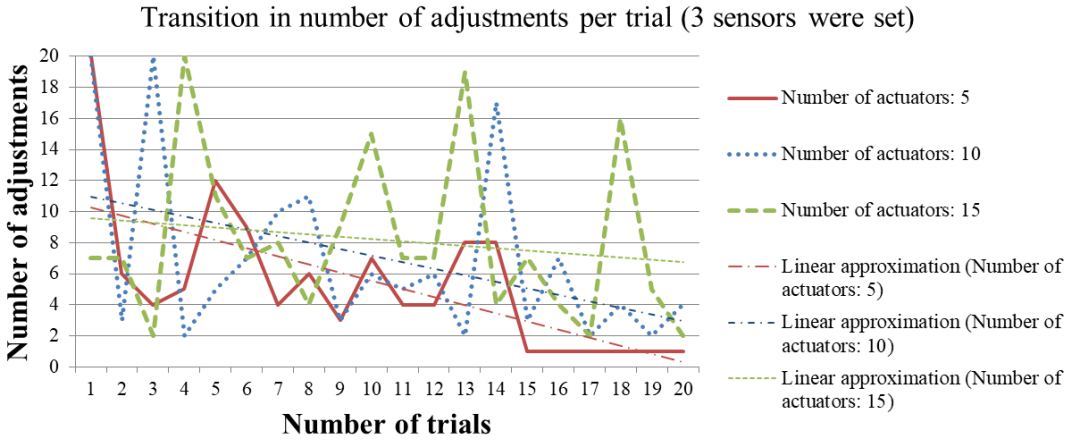


図 5.5 アクチュエータ数ごとの試行回数に対する調整回数の推移（センサ数 3）

(©IEEE 主著論文 1 より引用)

また，いずれの実験条件でも，調整回数の 1 次近似式の勾配は負であったことから，サービスの試行回数が増加するにつれて，必要な調整回数が減少する傾向にあることを確認した．

なお，アクチュエータ数が 15，センサ数が 3 のときの勾配は-0.15 と小さいことから，試行回数に対する調整回数の収束可否を確認するための実験を行った．1 回の試行に対する最大調整回数をこれまでの実験と同様の 20 回とし，試行回数のみを 50 回に変更した実験を 2 セット実施した．図 5.6 に実験結果を示す．アクチュエータ調整値を確率的に求めるという手法の性質上，2 セットの実験には調整回数にばらつきがあるものの，1 次近似式の勾配は共通して負であり，絶対値が小さい方から順に-0.25 と-0.36 であった．前述の 20 回の試行の実験よりも，勾配の絶対値が増加していることから，最適な初期制御値を取得するまでに多数の試行を必要とするが，調整回数は収束に向かうことを確認した．

以上より，提案手法によって，学習データが少なく機械学習による初期制御値では目標スコアを達成できない場合にも，制御値の調整を組み合わせて行うことで目標スコアを達成できることを確認した．また，学習と制御値の調整を繰り返し行うことで，連続する試行に対して，目標スコア達成に必要な調整回数を減少させられることを確認した．

さらに，学習モデルの作成時間と，学習モデルを使用した初期制御値の計算時間を測定した．学習データの次元数が最も大きい 15 個のアクチュエータを用いる場合の学習モデルの作成時間を測定したところ学習データ数が 100 の場合に 0.1 秒，学習データ数が 300 の場合に 0.3 秒であった．また，学習モデルを使用した初期制御値の計算にかかった時間は，0.001 秒であった．以上より，本処理が E-CPS の要件であるリアルタイム性を満たすことを確認した．

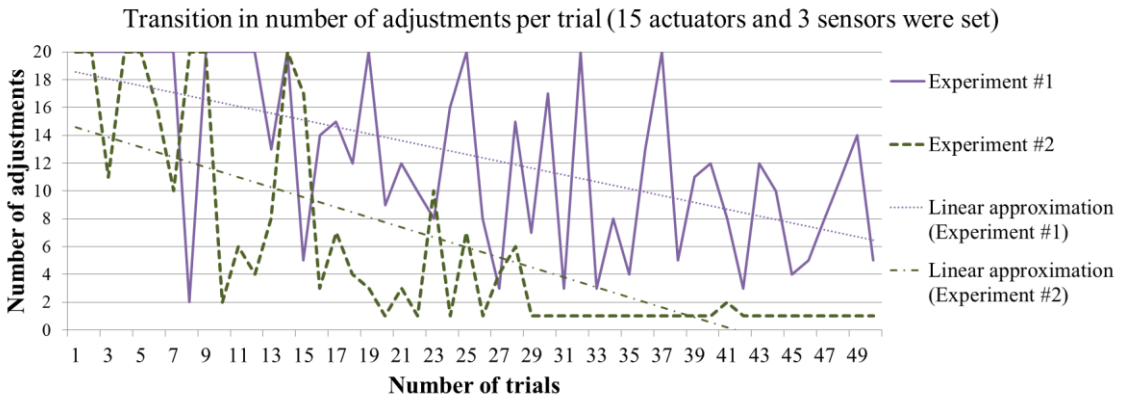


図 5.6　50 回の連続試行に対する調整回数の推移

(©IEEE　主著論文 1 より引用)

(ii) コスト削減に関する評価

提案手法のコスト削減効果を評価するため, 目標スコアを与えるセンサ数が 2, 3 および, アクチュエータ数が，5，10，15 の合計 6 通りの条件について，コスト削減のための調整処理を行う場合と行わない場合，それぞれのコストを算出して比較した．本実験では，各センサに与える目標スコアは 80%とし, 1 試行の最大調整回数はコスト削減のための調整を含めて 20 回とした．各試行の最終アクチュエータ制御値からコストを算出し，20 回の試行の平均値を評価した．

図 5.7 に各実験条件におけるコストの平均値を示す．いずれの実験条件においても，コスト削減調整を行う場合の方がコストの平均値が小さく，センサ数が 3，アクチュエータ数が 5 の条件のときに，削減率は最も大きく 17%であった．

図 5.8 は，一例として，目標スコアを与えたセンサ数が 2，アクチュエータ数が 5 かつ，コスト削減調整を行う条件における，試行ごとの，目標スコアを達成するまでに要した調整回数，コスト削減調整前後のコスト，最終状態のスコア，および最終状態のアクチュエータ制御値を示したものである．なお，本実験における，コスト調整前後のアクチュエータ制御値を見ると，目標スコアに関係しないアクチュエータ#3 に関して，全試行における調整前の制御値の平均は 43，調整後の平均は 34 であり，同じく目標スコアに関係しないアクチュエータ#4 に関して，全試行における調整前の制御値の平均は 28，調整後の平均は 22 であった．以上より，コスト削減調整によって，サービスに関連しないアクチュエータの制御値が減少し，コストを削減できることを確認した．

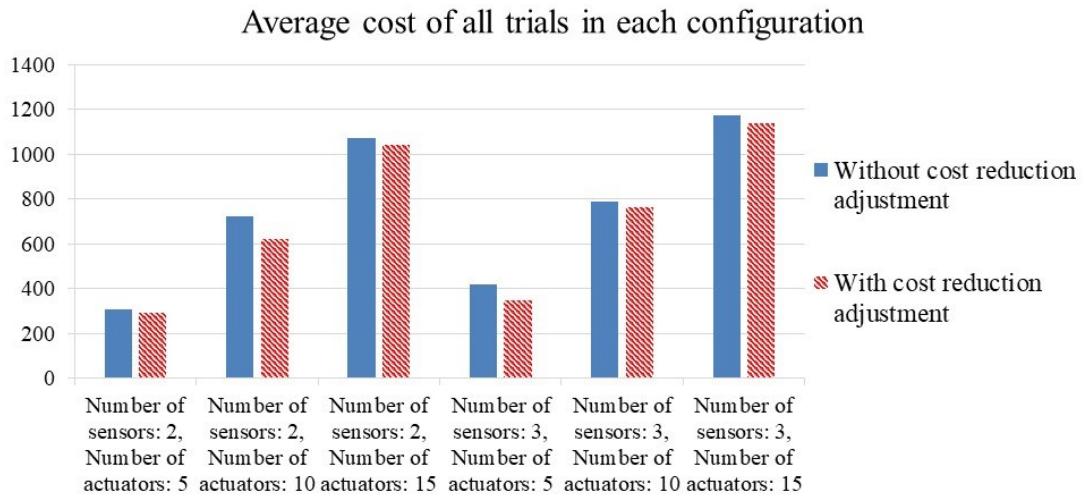
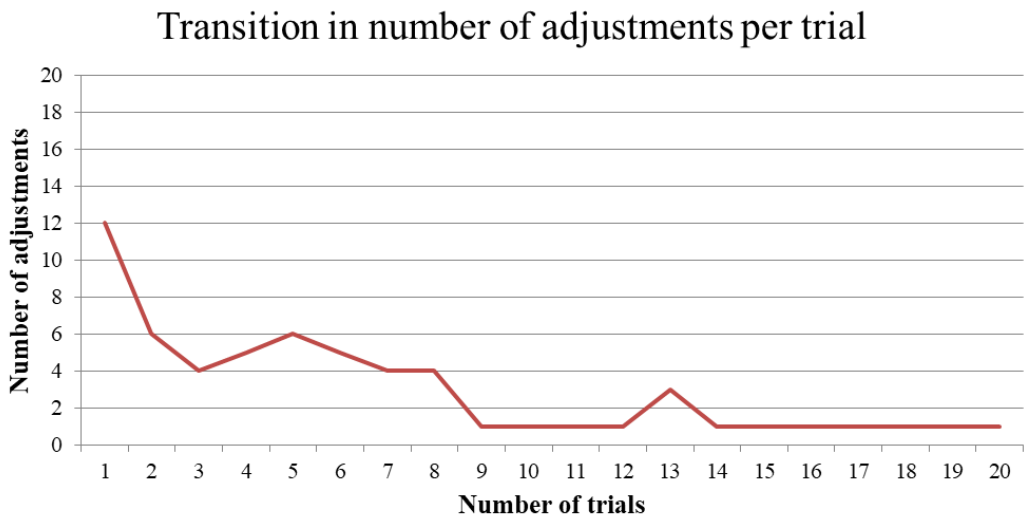
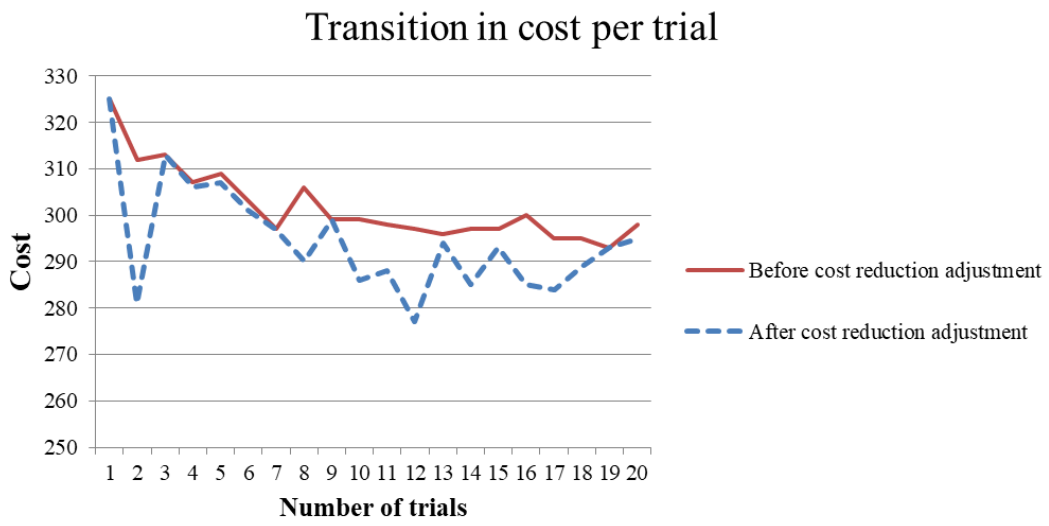


図 5.7　1 試行の平均コストに関するコスト削減調整有無の比較
(©IEEE　主著論文 1 より引用)


Transition in number of adjustments per trial
Number of adjustments
0
2
4
6
8
10
12
14
16
18
20
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Number of trials


Transition in cost per trial
Cost
250
260
270
280
290
300
310
320
330
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Number of trials
Before cost reduction adjustment
After cost reduction adjustment

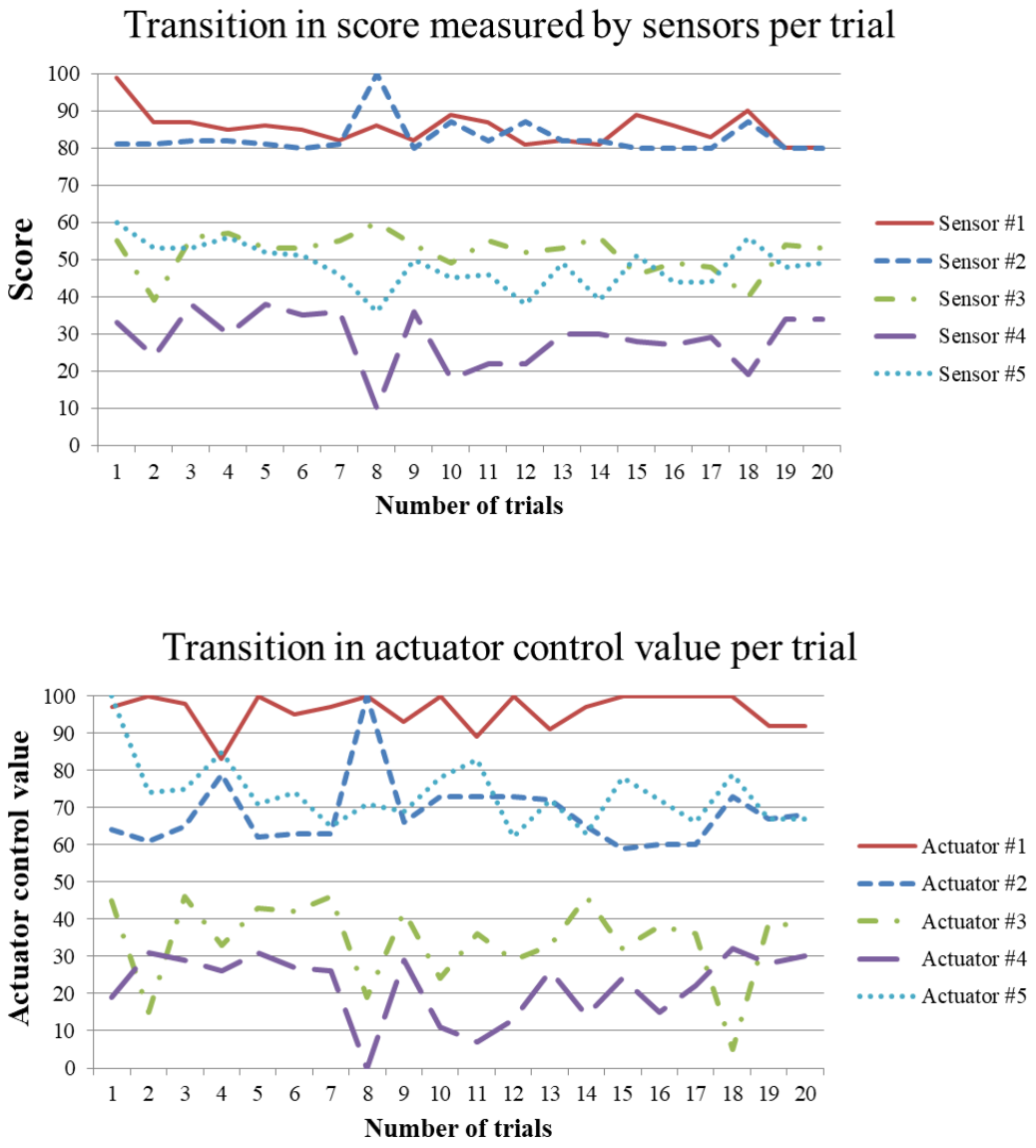


図 5.8　センサ数 2，アクチュエータ数 5 の条件における実験結果
(©IEEE　主著論文 1 より引用)

5.5.3 実機実験

実機実験の目的は，実サービスを想定したシステム設計と提案手法のフィージビリティ確認である．作用に重ね合わせがある複数のアクチュエータを制御するユースケースとして，2 つの照明デバイスとカメラを使用した色相制御実験を行った．以下に，実験条件と実験結果を示す．

A. 実験システムの仕様

2 つの照明デバイスとカメラ，画像認識ソフトウェアから構成されるシステムを設けた．図 5.9 は実験システムの構成を示し，図 5.10 は実験環境を示したものである．2 つの照明デバイスは特定の色相の光を照射し，スクリーンには，重なり合あった色相が投影される．

本実験は，スクリーンに特定範囲の色相が投影されることを目標とし，照明デバイスの色相を制御対象とした．実験システムは，E-CPS の設計指針に従い，センサデータからのスコア算出，アクチュエータ制御値算出およびアクチュエータ制御をそれぞれ独立したソフトウェアで実装した．ソフトウェアの限定的な改修のみで異なる種類のデバイスやサービスへの対応が可能なシステムである．

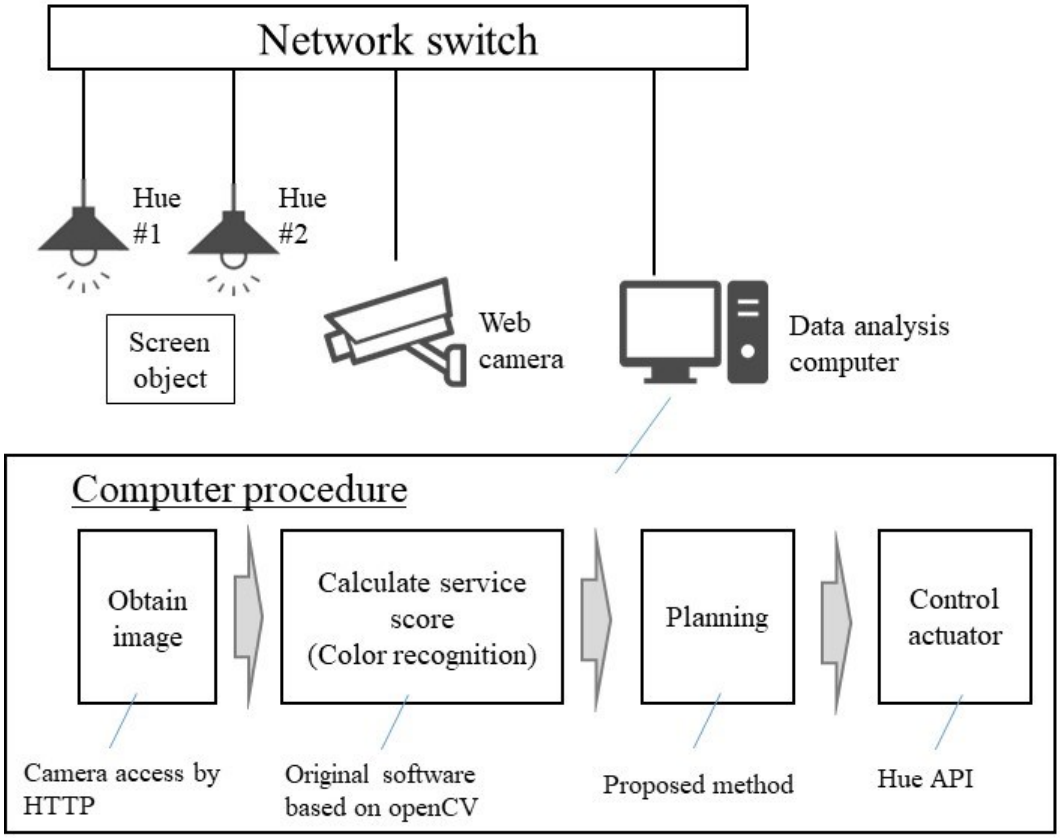


図 5.9 実験システムの構成

(©IEEE 主著論文 1 の Fig. 10 を一部修正)

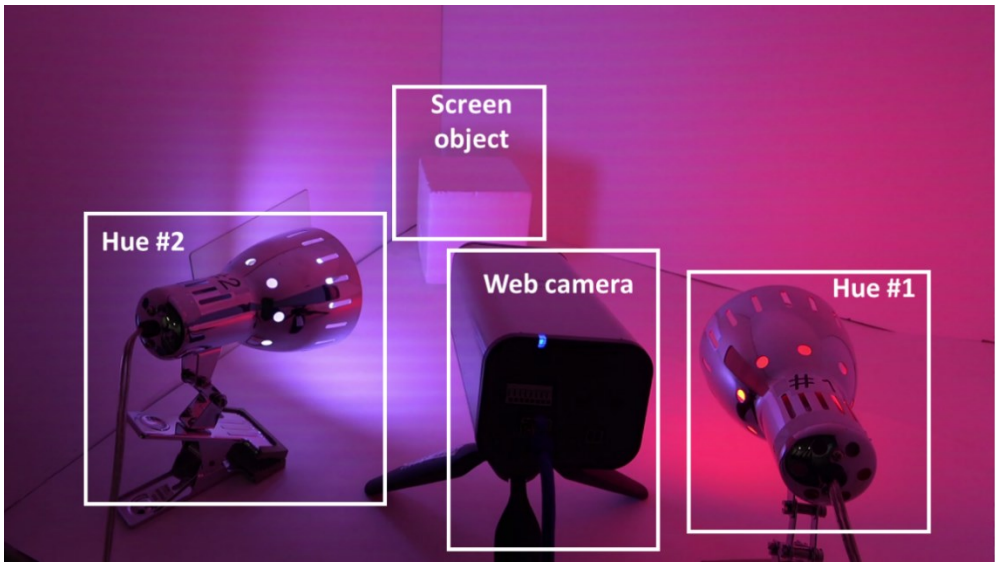


図 5.10 照明デバイスを用いたデバイス制御実験環境
(©IEEE 主著論文 1 より引用)

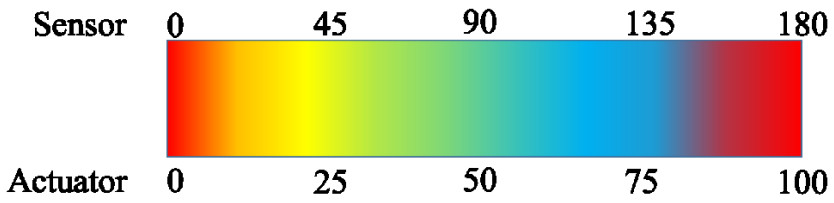


図 5.11 センサ取得値とアクチュエータ制御値に対応する色相

システムの詳細仕様を以下に述べる.

カメラはスクリーンを撮影し, 映像をネットワーク経由でコンピュータに転送する. コンピュータでは, スコア算出を行う画像認識ソフトウェアを実行し, カメラ画像からスクリーンに投影された色相を判定する.

アクチュエータである照明デバイスには, Philips Hue [101]を使用した. 本製品は, 0～65,535 の範囲で色相を操作できる API を提供している電球型のデバイスである. 本 API を用いることで, ネットワークからリアルタイムに色相を制御することができる. 本実験では, 簡単のために, 0～100 の値域で色相を指定できるようにした.

画像認識ソフトウェアは, OpenCV ライブラリ[102]を使用したオリジナルのソフトウェアである. 本ソフトウェアは, 0～180 の値域で目標とする色相の範囲を設定すると, 観測された色相に応じたスコアを算出する. 目標範囲の中心値の色相が観測された場合にスコア 100, 目標範囲の端の色相が観測された場合にスコア 0 を算出し, 両者の間は線形補間した値を算出する. アクチュエータの制御値と画像認識ソフトウェアそれぞれに対する色相との対応を図 5.11 に示す.

実験は，130〜150（紫色）を目標色相範囲に設定し，目標スコアは 80%とした．つまり，中央値である 140 の色相が観測された場合にスコアは 100 となり，130 未満または 150 を超える色相の場合にスコアは 0 となる．目標スコア 80%を満たすのは，138〜142 の範囲の色相が観測された場合である．初期制御値の算出におけるニューラルネットワークの設定はシミュレーション実験と同様とした．10 回の試行を行い，1 回の試行におけるアクチュエータ調整回数の上限は 10 回とした．本実験では，コスト削減調整は省略した．以上の条件のもとで実験を 2 セット行った．各実験セットの間で学習データは消去した．

B. 実験項目と実験結果

(i) 目標スコア達成に関する評価

はじめに，提案手法により，目標スコアを満たすアクチュエータ制御値の導出が行えることを確認するために，目標スコアを満たすために要した調整回数を測定した．

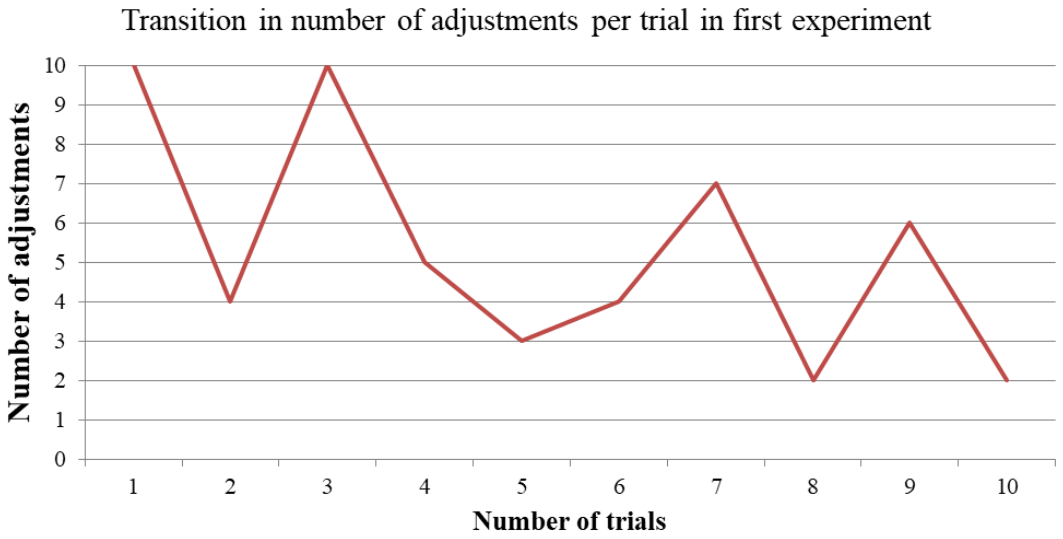
目標スコアに達成するまでに要した各試行の調整回数と各試行における最終アクチュエータ制御値を図 5.12 に示す．

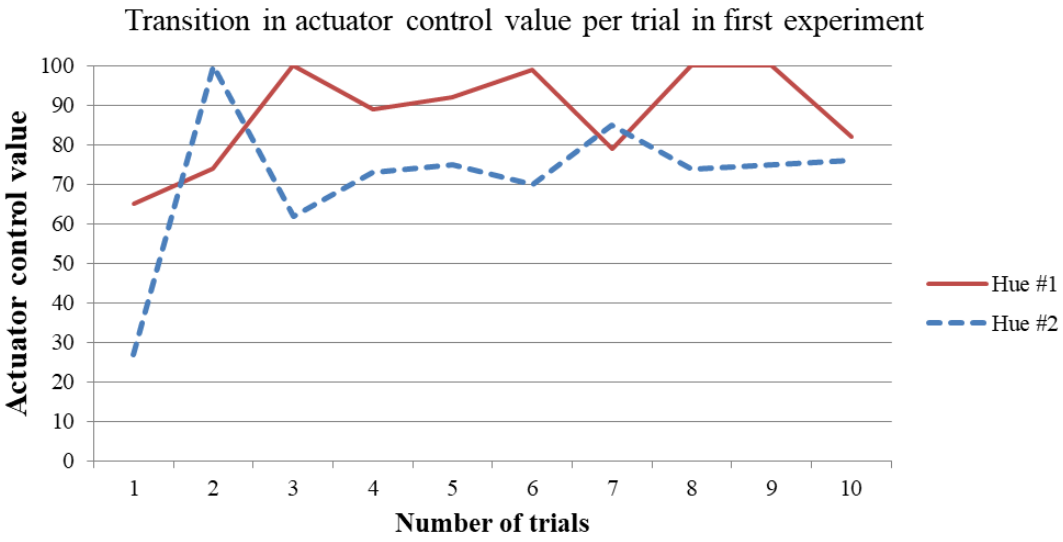
2 セットの実験の総試行に関する調整回数の平均は 3.8 となった．また，10 回未満の調整で目標スコアを達成した試行の割合は 85%，5 回未満の調整で目標スコアを達成した試行の割合は 70%であった．比較のため，アクチュエータ制御値をランダムに 100 セット生成したところ，目標スコアを満たす制御値が選ばれた割合は全体の 8%であった．

さらに，学習データの蓄積による調整回数の減少について分析すると，全 10 回の試行のうち，前半 5 回の試行の 2 セットの実験の平均調整回数は 4.8 回，後半 5 回の平均調整回数は 2.7 回であり，学習データの蓄積が進むことで，より目標スコアを満たす制御値に近い初期制御値が与えられ，調整に必要な回数が減少することを確認した．

また，目標スコアを満たす 2 つのアクチュエータの制御値に関して，二種類のパターンが見られた．一つは，制御値 70（青色）付近と 100（赤色）付近の組み合わせであり，もう一つは両アクチュエータの制御値が 80（紫色）付近の組み合わせである．このように，提案手法を用いることで，解が複数存在する問題に対しても，目標スコアを与えるだけで自律的に適切な制御値の組み合わせを発見できることを確認した．

以上より，実機デバイスを用いたユースケースにおいて，提案手法を用いることで，目標スコアを達成するデバイス制御値を自律的に導出できること，および，学習データの不足により機械学習が算出した制御値では目標スコアを達成できない場合においても，制御値の調整を組み合わせて行うことで目標スコアを達成できることを確認した．


Transition in number of adjustments per trial in first experiment
Number of adjustments
0
1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
10
Number of trials


Transition in actuator control value per trial in first experiment
Actuator control value
0
10
20
30
40
50
60
70
80
90
100
1
2
3
4
5
6
7
8
9
10
Number of trials
Hue #1
Hue #2

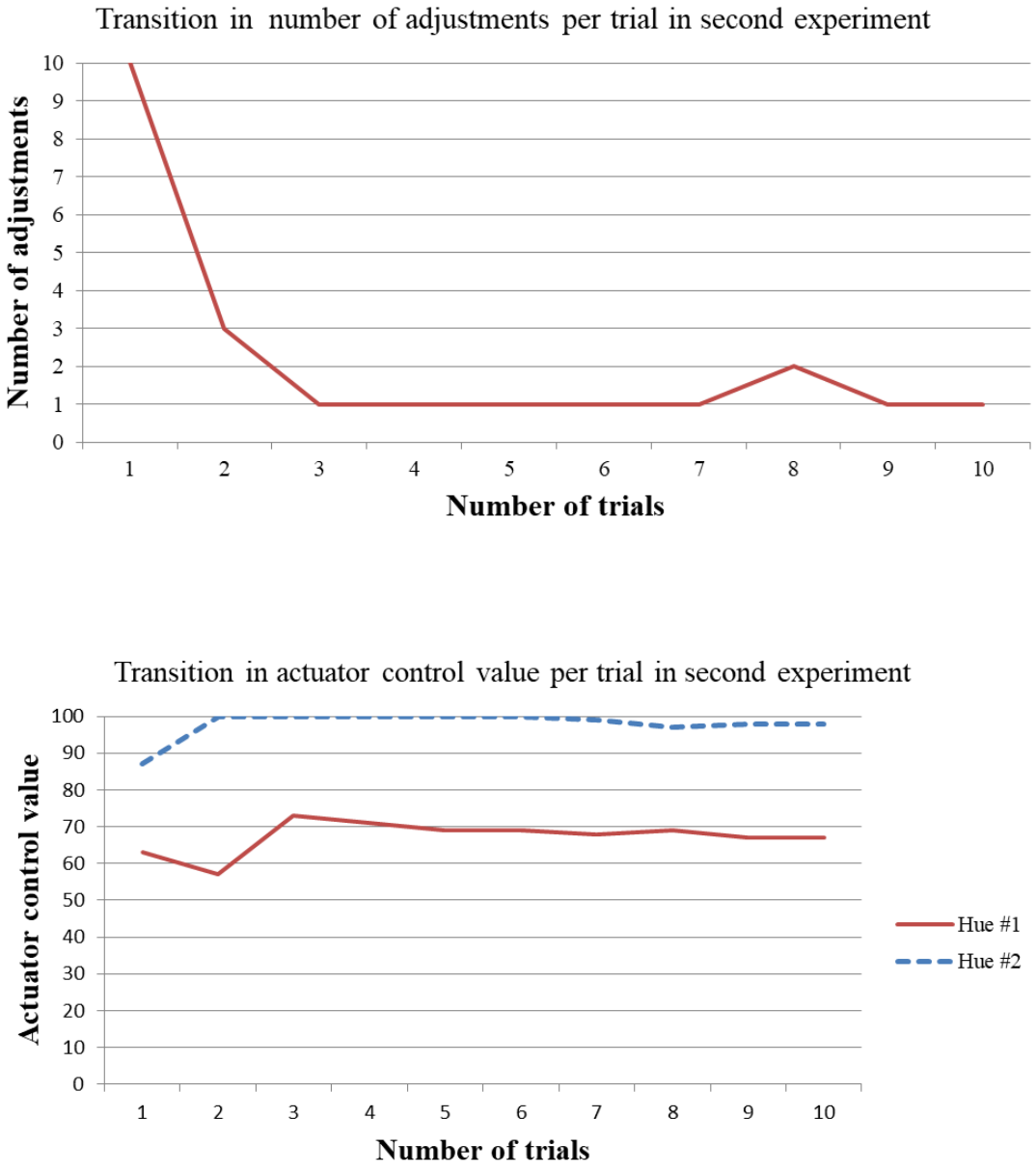


図 5.12 目標スコア達成に関する評価における試行に対する調整回数とアクチュエータ制御値の推移

(©IEEE 主著論文 1 より引用)

(ii) デバイス変動への適応性評価

デバイス位置や作用範囲の変動に対する提案手法の適応性を確認するために，学習データ蓄積後にデバイスの設置状態を変化させたうえで，アクチュエータ制御を継続し，目標スコアを達成するために要した調整回数を測定した．

実験システムにおける，2 つのアクチュエータのうち 1 つに対し，図 5.13 に示す，設置角度が異なる 2 つの状態を設けた．はじめに状態 A において，前述の実験と同様に，目標スコアを達成するためのアクチュエータ制御を行った．試行回数を 20 回とし，試行ごとの最大調整回数は 10 回とした．目標スコアに達成するまでに要した各試行の調整回数と各試行における最終アクチュエータ制御値を図 5.14 に示す．本結果を確認すると，7 回目以降の全ての試行において，初期制御値が目標スコアを満たしている．つまり，状態 A における適切なアクチュエータ制御値を機械学習により獲得している状況である．

次に，照明デバイスの設置角度を変え，状態 B とした．本状態において，状態 A の蓄積済み学習データを使用して同様の実験を継続した．状態 B における試行回数は 5 回として 2 セットの実験を行った．目標スコアを達成するまでに要した各試行の調整回数と各試行における最終アクチュエータ制御値を図 5.15 に示す．2 セットの実験共に，状態 B に遷移後の初回の試行では，初期制御値では目標スコアを達成できず，調整処理が行われている．その後，2 セットの実験において平均 2.5 回の試行を経た後は，目標スコアを満たす初期制御値を連続して算出している． つまり，状態 A において蓄積された学習データは，状態 B に対して完全に同様の効用をもたらすものではないが，状態 B において少数の調整のみで目標スコアを達成することを可能にした．さらに，状態 A の学習データに状態 B の少量の学習データを補充することで，状態 B に適した初期制御値が機械学習によって算出された．

以上より，デバイス設置状態の微細な変化に対して，提案手法が学習データを活用して，目標スコアを少数の試行で達成できることを示し，デバイス変動への適応性を有することを確認した．

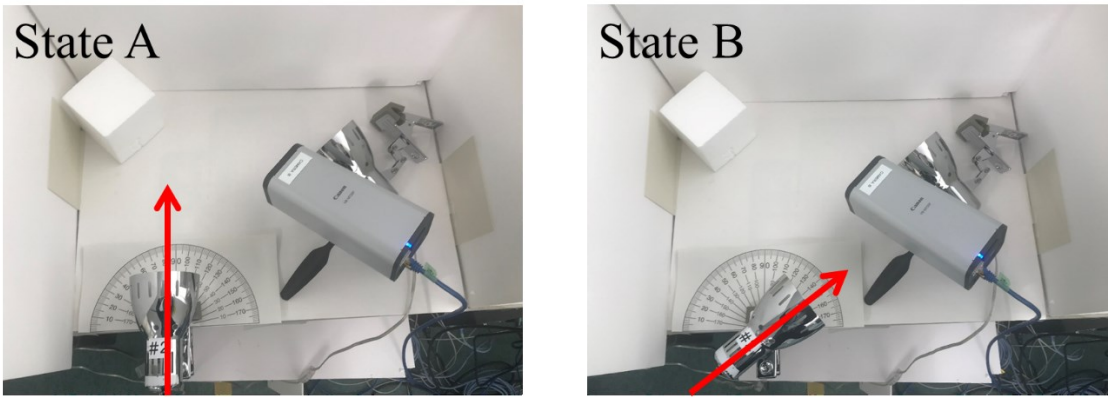


図 5.13 デバイス変動への適応性評価実験環境

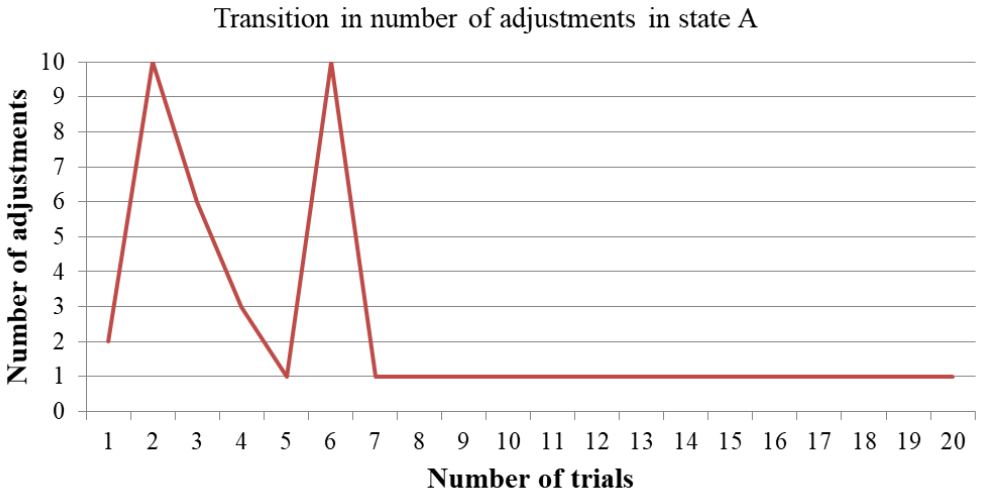
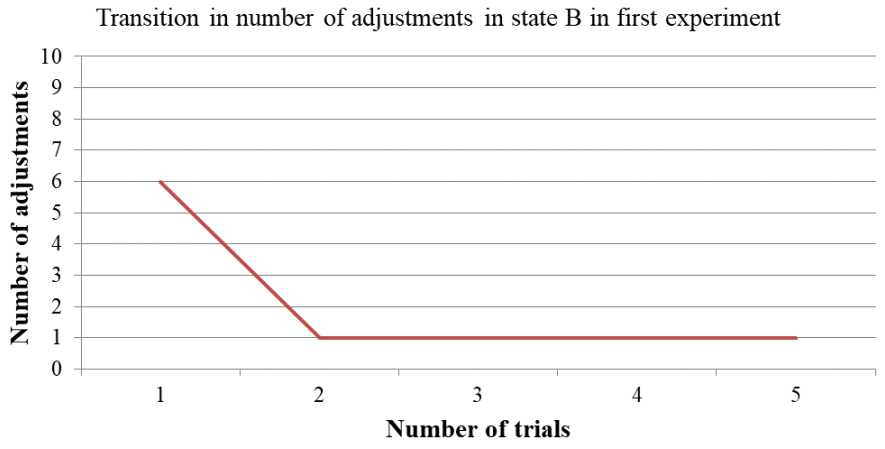


図 5.14 デバイス変動への適応性評価における試行に対する調整回数の推移（状態 A）



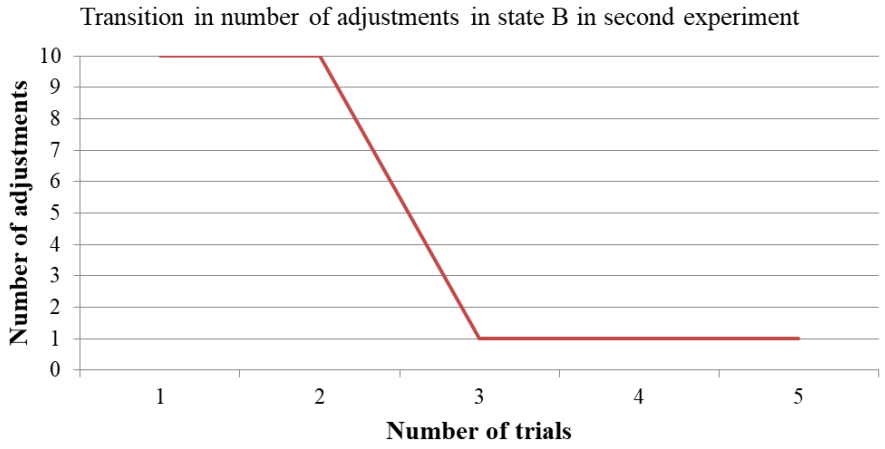


図 5.15 デバイス変動への適応性評価における試行に対する調整回数の推移（状態 B）

5.6 考察と今後の課題

E-CPS の実現に求められる多様な組み合わせのデバイスの自律制御に対する，提案手法の有効性を実験により示した．実機実験に関して，実験結果から判明した必要調整回数の妥当性を考察する．実機実験結果より，目標達成に必要な調整回数の平均は 3.8，10 回未満の調整で目標スコアを達成した試行の割合は 85%であった．また，実機実験システムにおいて，1 回の調整に要する時間は数秒であった．これは，デバイス制御命令が実機の動作に反映されるまでの時間と，カメラ画像の画像認識結果がスコアとして算出されるまでの時間がいずれも微小であったためである．つまり，10 回以内に調整が完了すれば，長くとも数分以内にサービス目的を達成するため，実験結果の調整回数は，E-CPS の迅速なサービス提供を実現する上で妥当と言える．一方で，デバイスやサービスの種類によっては，デバイス制御結果が環境に反映されるまでに長時間を要することがある．例えば，制御対象を暖房機，スコアを室温とする場合には，制御命令の発信からスコア算出を行うまでに一定の待機時間を設ける必要があり，1 回の調整時間は数分オーダとなる．このようなデバイスとサービスを迅速に提供するには，調整回数の削減が課題である．

また，デバイス変動への適応性評価実験結果より，デバイス設置条件が微小に変化するだけで，サービス目的達成に必要なデバイス制御値が変化し，構築済み機械学習モデルの単純利用ができなくなることが確認された．種類や設置条件が多様な組み合わせのデバイスを制御する E-CPS において，事前に構築された学習モデルだけでなく，実際の環境における調整処理が必須であることを示している．また，変動前状態で蓄積した学習データを変動後状態で活用することの有効性を示したが，変動が過大の場合には，蓄積済みの学習データが制御値探索範囲を狭め，適切な制御値の導出を妨げる可能性がある．調整処理を行う中で，蓄積済み学習データ量および，目標スコアと現在スコアの差の程度から，学習モデルの信頼性と流用可否を判断する仕組みが必要である．

以上を踏まえ，今後の展開に向けては下記の課題がある．

一つ目の課題は，移動デバイスへの対応である．実験では，全てのデバイスが固定され，常にネットワークに接続された状態を前提とした．実際の環境では，スマートフォン，ウェアラブルデバイス，コネクテッドカー等の多くの移動デバイスがある．これらの移動デバイスは，デバイス間の物理的な位置関係の変化や，頻繁なネットワークへの接続，離脱が生じる．したがって，スコアとアクチュエータ制御値の関係を表す学習モデルを作成しても，その関係性が頻繁に変化してしまう．変化の程度や頻度が小さい場合には，実機実験で示した通り，制御値を調整し，学習モデルを徐々に修正するアプローチが有効だが，程度や頻度が大きい場合には，異なるアプローチが必要である．今後の課題として，まず第一に，このような移動デバイスへ対応するために，GPS や無線技術によるリアルタイム測位を活用し，位置を元にしたデバイス間の関係性を学習モデルに反映したうえで制御を行うことが挙げられる．

二つ目の課題は，複雑なサービスシナリオへの対応である．E-CPS は，移動ロボットのようなデバイスを用いた複雑なサービスシナリオも対象とする．このようなサービスは，単純なアクチュエータ制御だけでは，適切な動作を生成できない．移動体を含むサービスシナリオに対する状態遷移を考慮した制御を行う必要がある．このようなサービスに対応するために，近年研究が盛んな強化学習[103]のフレームワークが活用できる．強化学習を利用して，データを収集しながら最適な動作を取得することが可能である．一方で，一般的な強化学習は大量の試行を必要とするため，E-CPS のデバイス制御に用いるには拡張が必要である．転移学習やファインチューニングという手法では，任意の環境で学習済みのモデルを他の環境や用途に流用する．本手法に則り，類似したデバイスの組み合わせにおいて生成した学習モデルを適切に流用することで，新たなデバイスの組み合わせに対して，少量の学習データと少数の調整のみで解を導くアプローチが可能であると考えている．

三つ目の課題は，E-CPS の大規模展開に向けた，競合する複数サービスへの同時対応である．共用デバイスを用いるが故に，複数のサービス間でアクチュエータ制御が競合し，単一のサービス目的を達成できたとしても，他のサービスに悪影響が及ぶことがある．したがって，複数のサービスを跨った最適な制御を実行する必要がある．課題解決に向けて二つのアプローチが考えられる．一つ目は，マルチエージェントシステムによる最適解の導出である．マルチエージェントシステムとは，複数のエージェントから構成されるシステムであり，エージェントが相互に交渉することでシステム全体を最適化する．E-CPS のデバイス制御においても，同時に実行される複数のサービスシステムが，相互に目標とするサービススコアとアクチュエータ制御値を通知し合うことで，多数のサービスの調停を自律的に行える可能性がある．二つ目のアプローチは，サービスの構造化である．サービスの性質に基づいて，クラスタリングと階層化を行い，より高次の目的関数を設定することで，競合する要求を調停する．E-CPS は，多数のサービスを扱うことを想定しているため，本アプローチをとる場合には，スコアとアクチュエータ制御値の対応に基づいて，サービスを自動的に階層化する機能が不可欠となる．今後，サービスユースケースを分析し，有効なアプローチを選定して検討を進めていく．

5.7 第 5 章のまとめ

本章では，E-CPS の実現に向けた技術課題の一つである，ネットワークに接続された多様な組み合わせのデバイスに対する自律制御を取り上げ，機械学習と動的な調整による，事前の設計や設定が不要な制御手法を提案し，実験により有効性を示した．

はじめに，E-CPS のデバイス制御の要件を詳細化し，デバイスを意識した設計の排除，多様なサービスに対する汎用性，システム構成デバイスの変動への適応，システム運用コストの最小化，の 4 つの要件を挙げた．

次に，これらの要件を満足するために機械学習を利用するデバイス自律制御手法を提案

した．本手法は，機械学習を用いることで，デバイスやサービスの性質を踏まえた制御則の設計を不要にする．また，機械学習と，動的な調整を組み合わせて行うことで，大量の学習データを前提とせずにデバイスを制御する．

提案手法の有効性をシミュレーションと実機デバイスを用いた実験により示した．均等に配備した 3 台のセンサと 15 台のアクチュエータを想定したシミュレーションにおいて，平均の調整回数が 8.2，20 回未満の調整で目標スコアを達成する試行の割合が 95%であった．また，アクチュエータ制御値に対応するコストを最大 17%削減した．さらに，2 つの照明デバイスとカメラを使用した色相制御実験では，2 セットの実験の総試行に対する調整回数の平均は 3.8，調整回数 10 回未満の調整で目標スコアを達成する試行の割合は 85%であった．さらに，アクチュエータの設置条件を変化させた場合に，平均 2.5 回の試行を経ることで変化後の環境に適した初期制御値を算出できることを確認した．以上より，提案手法によって，学習データが少なく機械学習だけでは目標スコアを達成できない場合にも，制御値の調整を組み合わせて行うことで目標スコアを達成できることを確認した．また，機械学習と調整を繰り返すことで目標スコア達成に必要な調整回数を減少させられること，および，環境の微細な変化に対する適応性を有することを確認した．

E-CPS の適用領域拡大に向けて，ウェアラブルデバイスなどの移動デバイスを用いるシステム，状態遷移を伴う複雑なシステム，および競合する複数サービス調停に対応するための拡張が今後の課題である．

第6章 結論

6.1 はじめに

本章は, 本研究の総括として, E-CPS の実現に向けた研究成果と展望を述べる. はじめに, 本研究の成果および，E-CPS の要件に照らし合わせた提案手法の充足性を述べる．その後, 本研究の展望として，提案手法の発展に向けた課題と，本研究成果の社会導入に向けた指針を示す.

6.2 本研究の成果

本節では，本研究の成果概要，E-CPS の要件に照らし合わせた提案手法の充足性，および各章に示した成果の要点を述べる.

6.2.1 本研究の成果概要

本研究の成果概要を以下に示す.

・E-CPS の提案とシステム構成，課題の明確化

本研究は，ネットワークを介して多種多様なデバイスが共用されるオープン IoT の到来に向け，その恩恵を享受する革新的 CPS である，Ephemeral-Cyber-Physical System (E-CPS) を提案し，実現に向けた技術課題を明らかにした.

E-CPS は，ネットワークに接続された共用デバイスを組み合わせることで，サービス事業者に対して，クラウドコンピューティングのように簡易，迅速，安価に提供される CPS である．既存の IoT やロボット等の CPS は，いずれも人手による設計や検証に基づき指定されたデバイスによる構成を前提としたものであり，ネットワークに接続された大量のデバイスとデータを動的に組み合わせた構成は皆無である．本研究が提案する E-CPS は，現行の CPS にパラダイムシフトをもたらすものである.

さらに，本研究は，E-CPS の実現に必要なシステム構成を示し，データ収集とデバイス制御という CPS の汎用機能を共用デバイスによって実現するための技術課題として，大規模ネットワークからのデータ収集，ネットワーク内の多種デバイス識別，多様な組み合わせのデバイスの自律制御，の 3 つを抽出した．これらのシステム構成と技術課題は，E-CPS の構成法確立に向けた指針を創出するものである.

・E-CPS 構成法の基礎の構築

本研究は，E-CPS の技術課題を解決する 3 つの手法を提案し，その有効性を示した．これらの手法は，E-CPS を実現するための必須技術である．

第 1 の提案は，E-CPS におけるデータ収集を可能にする，大規模ネットワークのライブデータ検索手法である．本手法により，広域ネットワークに分散した膨大な数のセンサが発信するデータの中から，サービス実行に必要なものをリアルタイムに発見することが可能である．指定されたデバイスを使用する従来の CPS より遥かに膨大な数のデバイスからデータを収集するために，既存の IoT サービス事業者が考慮していない物理ネットワーク上の機能配備を扱う手法である．本技術により，全てのデータをクラウドコンピューティングで扱う場合と比較して，広域ネットワークに発生するネットワークトラフィックを 68%以上削減する可能性を示した．

第 2 の提案は，E-CPS におけるデバイス特定を可能にする，通信情報分析による多種デバイスの識別手法である．本手法は，ネットワークに接続されたデバイスを E-CPS の構成要素として選択，制御可能な状態にするために，その機種や設定を識別する．受動的に得られる通信情報のみからデバイスを識別するため，デバイス本来の処理や通信への影響が無く，また，人手を介するデバイスの登録，公開を不要とする．実験により，同一種類のデバイス（ネットワークカメラ）が多数存在する環境下で 99.5%の正解率で機種を識別できることを示した．

第 3 の提案は，E-CPS における多様な種類，設置条件下にあるデバイスの自律制御を可能にする，機械学習を利用したデバイス自律制御手法である．本手法により，ネットワークに接続された不特定多数のデバイスに対して，その組み合わせを意識した人手による設計と検証を伴わずに自律制御させることが可能である．本手法は，機械学習を利用しつつ，動的な調整を組み合わせて行うことで，大量の学習データを前提とせずにデバイスを制御することが特徴である．シミュレーションと実機実験により，少量の学習データのみを用いて，適切なデバイス制御を自律的に行えることを示した．

以上の提案手法をネットワーク設備に実装することで，図 6.1 に示す E-CPS のプロトタイプを実現することが可能と考えている．本システムは，サービス事業者が抽象的なシステム構成要求を与えるだけで，ネットワークに接続された共用デバイスを組み合わせて自動的に構成されるものである．具体的には，ネットワーク設備に配備されたコンピュータは，広域に分散する大量のセンサが生成するデータから必要なものを収集し，さらに，ネットワークに接続された多種の共用デバイスの機種を把握したうえで，それらに対して制御命令を発信する．制御命令は，サービス目的とセンサから収集された環境情報を元に自動的に導出される．

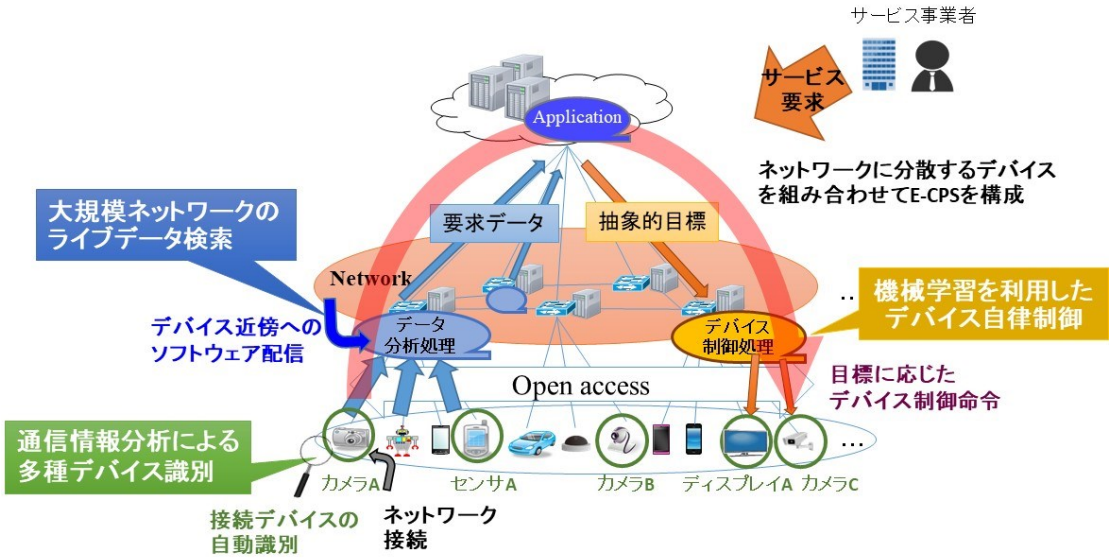


図 6.1 本研究成果による E-CPS の構成例

さらに，以上のプロトタイプをサービスユースケースに照らし合わせて述べる．人物捜索サービスを例とすると，ネットワーク上の近傍のコンピュータは，街頭や店舗といった広域に存在するネットワークカメラの映像データを分析し，当該人物の画像データを発見する．同様のコンピュータはさらに，ネットワークカメラやスマートフォンなどのデバイスが公衆無線 LAN 等に接続されると，通信情報を分析して機種を特定し，E-CPS の構成要素の候補とする．さらに，明るさ等の環境要因から，ネットワークカメラの視聴映像が人物特定に十分な品質ではない場合には，ネットワークカメラの解像度やフレームレートの変更，もしくは，近傍の共用照明デバイスの制御を自律的に行う．

以上の通り，本研究は，ネットワークを介して共用される多種多様なデバイスを一時的に利用して構成される E-CPS の構成法として，広域ネットワークからのデータ収集，多種デバイスの識別，および多様な組み合わせのデバイスの自律制御という，必須技術を提案した．本研究成果は，既存の CPS にパラダイムシフトをもたらし，簡易，迅速，安価な CPS の提供と普及拡大に寄与するものである．

6.2.2 E-CPS の要件に対する提案手法の充足性

本研究が取り組んだ各課題に対する提案手法について，1.5.2 項に述べた E-CPS の要件に照らし合わせた充足性を表 6.1 に示す．

表 6.1 本研究による E-CPS の要件充足性と今後の課題

	提案手法の要件充足事項			解決すべき課題
	大規模ネットワークのライブデータ検索	通信情報分析によるデバイス識別	機械学習を利用したデバイス自律制御	
セキュリティ	・ローカルコンピュータへの最新バージョンアプリケーション配信 ・ローカルネットワーク内に限定したデータ処理	・セキュリティリスクがある機種のネットワーク接続検知	-	・プライバシー保護ソフトウェアの充実
信頼性	・センサ障害時自動代替切り替え	-	・デバイス障害時の制御値自律調整	・デバイス二重化，予約確保等による無中断切り替え
QoS	-	-	-	・ネットワークスライシング，SDN の適用
リアルタイム性	・秒オーダのデータ検索	・100 台規模のデバイスの数分以内の識別	・秒オーダのデバイス制御値算出	・システム全体としての処理時間保障
自律性	・抽象的な検索条件によるデータ検索	・ネットワーク接続デバイスの自動識別	・抽象的なサービス目的による，自律デバイス制御	・サービス記述様式の標準化
柔軟性	・様々なデータ形式，検索要求への対応	・様々な機種，通信プロトコルへの対応	・様々な機種，サービスへの対応	・デバイスインタフェース標準化
頑強性	・ネットワークへのセンサ追加，削減への対応	-	・ネットワークへのデバイスの追加，削減への対応 ・環境変化への対応	・移動デバイスの利用
経済性	・広域ネットワークトラフィックの削減	・汎用ネットワーク装置，汎用サーバの使用	・アクチュエータ出力の自動調整	・データ検索範囲の限定 ・制御最適化に要する試行回数削減

セキュリティに関しては，ライブデータ検索手法におけるローカルネットワークへのアプリケーション配信により，脆弱性対策が施された最新バージョンのアプリケーションを常に使用できる．また，デバイス所有者の要望に応じた匿名化アプリケーションをローカルネットワークに配信して実行することで，データを公開しつつも，プライバシー性が高いデータが広域ネットワークに流通することを防止できる．今後，セキュリティ，プライバシーの更なる向上には，多様なデータ形式と匿名化要件に応えるプライバシー保護ソフトウェアの充実が必要である．また，セキュリティに関して，デバイス識別手法は，セキュリティリスクがある特定機種のネットワーク接続検知にも利用可能である．

信頼性に関しては，ライブデータ検索手法によって，複数のセンサ情報から最も目的に合致するものをリアルタイムに発見することができる．したがって，使用中のセンサに障害が発生した場合にも，代替となるセンサを発見して，データ収集元を切り替えることが可能である．また，デバイス自律制御手法は，デバイス障害時に当該デバイスを除外してサービス目的を達成するデバイス制御値を自律的な調整によって導出することが可能である．今後，交通，医療等の厳格な信頼性が要求されるサービスへの適用に向けては，デバイスの冗長化，予約等を行い，サービス無中断の切り替えを実現することが課題である．

QoS に関しては，本研究の提案手法では明確に扱わなかった．今後，ネットワークスライシング[104]や SDN (Software Defined Network) [105]といったネットワーク制御技術を取り入れることで，個々のサービスにカスタマイズした QoS を提供することが課題となる．

リアルタイム性に関しては，ライブデータの検索手法，デバイス自律制御手法について，秒オーダの高速処理が行えることを確認した．デバイス識別手法に関しても，扱うデバイスとデータ量に応じてコンピュータを拡張することで，数分以内に識別が行える可能性を示した．今後は，通信遅延を含むシステムの総合的な処理時間の保証が課題である．

自律性に関しては，ライブデータ検索手法は抽象的な検索条件，デバイス制御手法は抽象的なサービス目的を与えるだけで処理を行うことが可能である．また，デバイス識別手法は，デバイスをネットワークに接続すると自動的に識別を行う．したがって，サービス事業者は，ネットワークに接続される個々のデバイスを意識することなくシステム運用が可能である．今後は，サービスカタログ等のサービス事業者が扱うサービス記述様式の標準化が課題となる．

柔軟性に関しては，提案手法はいずれも，デバイス機種やデータ形式，通信プロトコルに依存しない基幹機能と，それらに依存する固有機能を分離した実装が可能である．後者の実装にかかる時間と費用を削減し，多種多様なデバイスへの対応をさらに簡易にするためには，デバイスインタフェースの標準化が課題となる．

頑強性に関しては，ライブデータ検索手法およびデバイス自律制御手法は，いずれもネットワークへのデバイスの追加，削減への対応が可能である．両手法とも，ネットワークに接続されているデバイスをリアルタイムに把握してシステム構成要素の候補として扱う．また，デバイス自律制御手法は，デバイス設置条件等の環境変化に対しても制御値を自律的に

調整することで対応可能である．今後は，E-CPS のサービスドメイン拡大に向けて，位置が激しく変化する移動デバイスを用いる場合にも高い頑強性を満たすことが課題である．

最後に，経済性については，ライブデータ検索手法により，データ収集にかかる広域ネットワークトラフィックを削減することが可能である．また，デバイス識別手法の実装は，一般的なネットワークスイッチと汎用コンピュータのみで可能であり設備コストが低い．また，デバイス自律制御手法は，アクチュエータ出力をサービス目的の達成に必要な最小限に自動調整し，電力消費量を最小化できる可能性がある．今後は，ライブデータ検索手法の更なるコスト削減に向けて，検索範囲の限定が課題である．また，デバイス自律制御手法に関しては，最適状態に到達するまでの調整回数を削減し，余剰なアクチュエータ出力を，より早期に抑制することが課題である．

6.2.3 各章に示した成果の要点

以下，各章に示した成果の要点をまとめる．

第 1 章では，IoT に関する世界的な動向である，デバイス数とネットワークトラフィックの増加，日本における Society5.0 の取り組み，および，情報通信サービスを取り巻く社会的変化に基づく IoT の将来予想を示した．複数のサービス事業者が，ネットワークを介してデータやデバイスを共用するオープン IoT を示し，オープン IoT 時代の新たな CPS の形態である，E-CPS (Ephemeral-Cyber-Physical System) を提案した．

第 2 章では，E-CPS のシステム構成と機能，および関連する既存技術を分析し，本研究が扱う領域と技術課題を明確にした．E-CPS のシステム構成として，サービスとデバイスの種類に依存する固有機能と，デバイスやデータの選択・制御に関わる汎用機能とを分離し，また，デバイス制御に至るまでの処理を，扱う情報の性質から三つのレイヤに分割した構成を示した．さらに，広域に分散する多様な共用デバイスの活用という E-CPS 特有の性質から，大規模ネットワークからのデータ収集，ネットワーク内の多種デバイス識別，多様な組み合わせのデバイスの自律制御という，三つの技術課題を示した．

第 3 章では，広域に分散する大量のセンサから目的のデータをリアルタイムに発見する手法を示した．センサがリアルタイムに生成するデータをライブデータと名付け，広域ネットワークへ膨大なネットワークトラフィックを発生させることなく，ライブデータを検索することを実現した．エッジコンピューティング，クラウドコンピューティングとの机上比較と実験により，ネットワークトラフィックおよび，検索時間を評価し，提案手法の有効性を示した．

第 4 章では，ネットワークに接続されたデバイスの種類や機種を，受動的に得られる通信情報から識別する手法を示した．提案手法は，通信情報という IoT デバイスの汎用的情報を用いるため多種デバイスに適用可能な手法である．ネットワークカメラと工場模擬環境における予備実験により本手法のフィージビリティを示した．さらに，提案手法の識別正解率向上に向けて，ヒストグラム特徴量と機械学習の有効性を実験により示した．

第 5 章では，多様な種類や設置条件の複数デバイスをサービス目的に応じて自律制御する手法を示した．本手法は，機械学習を用いることで，デバイス設置条件や組み合わせを意識した人手による設計と検証を不要にする．本手法が，複数のセンサとアクチュエータが混在する環境においてサービス目的を達成するアクチュエータ制御を行えること，およびシステム運用コスト低減に有効であることをシミュレーションと実機実験により示した．

6.3 本研究の展望

本節では，今後の展望として，提案手法の発展に向けて取り組むべき課題を示し，さらに，本研究の発展に向けて注目する技術領域を明らかにする．最後に，本研究成果の社会導入に向けた指針を述べる．

6.3.1 提案手法の発展に向けた課題

提案手法の発展に向けた課題を表 6.2 に示す．本表に示す方式課題とは，技術の核心に関わる重大な課題であり，研究対象として今後扱うべき課題を示している．実装課題とは，提案手法を実用化する際に考慮すべき性能要件や開発機能を示したものである．また，表 6.2 には，次節に述べる注目する技術領域との関連性を“【 】”内に併せて記した．

表 6.2 提案手法の発展に向けた課題

	大規模ネットワークのライブデータ検索	通信情報分析によるデバイス識別	機械学習を利用したデバイス自律制御
方式課題	・ローカルネットワークのデータ特性を踏まえたデータ検索範囲の絞り込み ・データ分析アプリケーションの構造化	・識別性能の向上【AI】 ・NAPT パケット，暗号化パケットを用いた識別	・移動デバイスを用いた複雑サービスシナリオへの対応【AI】 ・複数サービス要求の競合解消
実装課題	・大量のローカルネットワークとリソースの管理 ・データ分析アプリケーションの充実【AI，マイクロサービス】	・プロトコルパーサの充実 ・大量ネットワークトラフィックの高速処理	・ライブデータ検索手法との連携による制御機能の動的配信，サービスごとの論理的システム分離【仮想化】 ・サービス目的達成スコア算出ソフトウェア(データ分析アプリケーション)の充実【AI，マイクロサービス】 ・デバイスインタフェース標準化

まず，ライブデータ検索手法に関して，今後，世界規模の膨大な範囲を検索対象とするには，ローカルネットワークのデータ特性に応じたデータ検索範囲の限定が必要であり，方式課題である．多数サービスのデータ検索要求に対して，対応するクエリフィルタを常に全てのライブデータバッファに配信すると，ダウンロードにかかる通信コストと，ライブデータバッファのクエリフィルタ処理コストが膨大になってしまう．ライブデータバッファごとに，収集データのコンテキストを分析し，データ発生傾向をもとにデータ検索範囲を限定するアプローチが考えられる．実装課題としては，大量のローカルネットワークのコンピュータリソースを管理する大規模データベースシステムの構築，運用が挙げられる．また，今後，様々なデータ検索要求に対応するためには，データ分析アプリケーションの充実も課題である．

次に，デバイス識別手法に関しては，多種多様なデバイスと通信環境への適用に向けて，識別性能の向上と NAPT や暗号化パケット等の特殊パケットを用いた識別が方式課題である．本研究は，ネットワークカメラを対象に識別性能を評価したが，ネットワークカメラ以外の多種デバイス混在下においても高性能な識別が行える必要がある．今後，識別性能向上に向け，通信特徴量の抽出における，適切な抽出周期とヘッダフィールドを導出する手法を検討する予定である．実装課題としては，識別対象デバイスが使用する通信プロトコルに対応するプロトコルパーサの充実が課題である．例えば，産業用機器などは業界独自の通信プロトコルを用いる事例があり，識別を行うには，対応するプロトコルパーサを用意することが必要となる．また，大量のデバイスが接続されたネットワークにおいて識別を行うには，大量のネットワークトラフィックを高速に処理することが必要であり，通信デバイスやソフトウェアにおける高速パケット処理が課題である．

最後に，デバイス自律制御手法に関しては，多様なサービスに適用範囲を拡大するために，ロボットやコネクテッドカーといった移動デバイスを用いた複雑なサービスシナリオへの対応が方式課題である．本研究は，設置位置が固定された単出力のデバイスのみで構成されるシステムを扱ったが，移動デバイスは，デバイスの位置関係が大規模かつ高頻度に変化するため，それに追従できるよう方式拡張が必要である．また，移動のような状態遷移を伴い，ある時間の出力が後続の出力へ影響を及ぼすシステムへの対応にも方式拡張が求められる．さらに，同時発生する複数サービス要求の競合解消も方式課題である．また，実装課題としては，高信頼，リアルタイムなサービスを実現するために，デバイス制御機能をデバイス近傍のコンピュータに配信し，さらに，サービスごとに論理的に分離されたシステムを構成することが課題である．ライブデータ検索手法や周辺技術との連携による実現が考えられる．また，センサデータからサービス目的達成スコアを算出するソフトウェア（データ分析アプリケーション）の充実および，デバイスインタフェースの標準化も実装課題である．

6.3.2 本研究の発展に向けて注目する技術領域

前項に述べた各技術の方式課題と実装課題より，本研究の発展に向けて特に注目する二点の技術領域を述べる

一点目は，AI 技術である．AI 技術は，デバイスの最適な動作生成，高度なデータ分析といった，E-CPS の様々な要件と機能を向上させる非常に強力なツールである．2000 年代初頭より，第 3 次 AI ブームと呼ばれる世界的社会現象が起きており AI 技術は急速に進化している[106]．今後，AI 技術の動向に注意しつつ，適切に活用，連携させながら研究を進めていくことが必要である．また，AI 技術を利用して多様なサービス需要に応えるために，マイクロサービス化[107]の検討も必要である．サービスの多様化に効率的に追従するには，サービス固有のソフトウェアを開発するのではなく，既存のソフトウェアを細かい粒度で作成し，それらを適宜組み合わせる形態をとることが有利である．一方で，専用に用意されたソフトウェアより性能に懸念が生じるため，適切な組み合わせの選択，システムの総合性能保証が課題である．

二点目は，仮想化技術である．クラウドコンピューティングサービスの多くは，仮想化技術を用いてコンピュータを隠蔽することで，コンピュータの共用を実現している．E-CPS における，多数サービス事業者によるデバイスの共用にも有用な技術である．仮想化技術の歴史は古いが，現在の主流を形成しているのは，コンピュータプロセッサの仮想化技術である Intel の Intel-VT (Intel Virtualization Technology) [108]に始まる流れである．本技術は，業務用コンピュータに限らず，汎用コンピュータにおいても使用される成熟した技術である．近年，さらに，通信機能に関する仮想化技術の研究開発が盛んである．SR-IOV [109]や DPDK [110]といった技術は，これまで課題であった仮想環境下の高速な通信処理を実現している．また，ネットワークの仮想化として，ネットワークスライシングの研究も盛んである．そして，Openstack や VMWare ESX [111]，Docker [112]といった仮想化支援ツールの研究開発も盛んに行われており，データセンタ内の大量のコンピュータと仮想コンピュータの対応管理，仮想コンピュータの移動，自動障害復旧が実現されている．共用デバイスと共用物理ネットワークで構成される E-CPS を安全，高品質に実現するには，これらの仮想化技術を活用し，計算処理，使用データ，ネットワークがサービスごとに論理的に分断されたシステムを構成することが課題である．

6.3.3 本研究成果の社会導入に向けた指針

本項では，これまでに述べた課題を踏まえた本研究成果の社会導入に向けた指針を述べる．E-CPS を実現するために求められる社会システムを経済的観点から述べたうえで，本研究成果の社会導入の構想を示し，最後に，本研究のロードマップを示す．

A. E-CPS のエコシステム

E-CPS の普及には，技術の成熟のみならず，社会システムの形成が不可欠である．図 6.2 に示す，サービス事業者，サービス利用者，デバイス所有者，デバイス製造者，プラットフォーム事業者を取り巻くエコシステムを形成し，それぞれの利得の保証が求められる．

本項で述べる，サービス事業者は，交通，医療といった各事業の IoT のサービスを提供することで収益を獲得する事業者を指し，サービス利用者はその需要者である．また，デバイス所有者とは，デバイスを所有する個人，または事業者を指す．デバイス製造者は，各デバイス製品を開発，販売する事業者である．最後に，プラットフォーム事業者は，サービス事業者とデバイス所有者を仲介する事業者である．

以上のプレイヤーから構成されるエコシステムについて，まずは，既存のエコシステムと共通する部分を述べる．デバイスに関する流れとして，デバイス製造者は，従来通りデバイス所有者にデバイスを販売して代金を徴収する．サービスに関連する流れとして，サービス事業者は，従来通りサービス利用者からサービス利用料金を徴収する．

さらに，E-CPS が想定するデバイスの共有を含むエコシステムには，サービス事業者とデバイス所有者が分離され，両者を取り巻く新たな流れが存在する．デバイス所有者は，デバイスとその生成するデータをサービス事業者に提供し，使用時間やデータ価値に応じた報酬を受け取る．サービス事業者とデバイス所有者の組み合わせは膨大であるため，両者のマッチングや厳格な課金を行うには，中立的な仲介者が必須である．本研究は，このような仲介者をプラットフォーム事業者と呼ぶ．本エコシステムを形成する上で，プラットフォーム事業者が果たす役割は大きい．

なお，本エコシステムを形成することで，サービス事業者は，これまで以上の短期契約，かつ，顧客固有の需要に則したサービスの提供が可能になることが想定される．

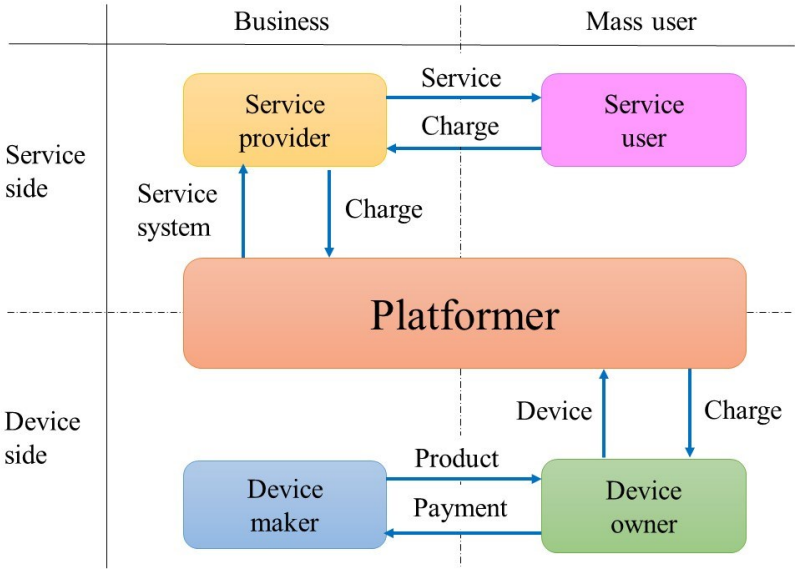


図 6.2 E-CPS が前提とするエコシステム

B. 本研究成果の社会導入に向けた構想

E-CPS の実現に向けて本研究が提案した手法は，前述のエコシステムにおけるプラットフォーム事業者の役割の一部を果たすものである．特に，通信事業者をプラットフォーム事業者とすることを想定し，物理回線を所有する通信事業者の設備に提案手法を具備する展開を考えている．

第 5 世代移動通信システム（5G）に代表される通信インフラの進化は目覚ましく，長期的には通信回線は低遅延，大容量化が進み，データ転送に関する既存の制約の解消が予想される．一方で，流通するデータ量の増加，利用範囲の拡大に伴い，大量のデータを必要な場所へ適切に流通させることへの要望はさらに高まっていくと予想される．このような背景のもと，本技術を通信事業者が保有する，広域に分散配備されたサーバやルータ等の設備に実装することで，情報を遠隔地へ転送するという通信事業者の基本的な役割を進化させ，データ特性や送受信対象デバイス，サービス用途に応じたデータ流通，データ加工を提供することが可能になる．通信事業者がこれらの価値を提供することによって，サービス事業者とデバイス所有者を仲介し，E-CPS のエコシステムを形成することができると考えている．ネットワーク設備への技術導入に向けては，特定事業者による製品開発のみならず，通信の標準仕様へ盛り込むことで幅広く技術を展開できる．一例として，3GPP (3rd Generation Partnership Project) [113]といった通信の標準化団体に参加し，将来のネットワーク設備の機能として本研究成果を盛り込むことが可能である．また，OneM2M 等の IoT の標準化団体に参画し，提案手法の実装に必要な機能をデバイスの標準機能として盛り込むことで，本研究成果の適用範囲拡大が期待できる．以上が本研究成果の社会導入に向けた構想である．

なお，既存のサービスとの関係について述べると，金融システムなどの非常に高い品質が要求されるサービスに関しては，これまで通りのサイロ型のシステム構成を維持し，E-CPS と共存していく想定である．既存の通信回線サービスにおいても，高い信頼性と秘匿性を有する高価な専用線と，ベストエフォートの安価なインターネット回線が共存していることから本想定は妥当と考えている．

また，E-CPS の普及に向けては，デバイスの共用を促進することが必須である．デバイスやデータの共用の現状として，第 1 章に示した通り，一部の事業者間でのデータ共有は既に取り組みが進められている．また，デバイスの共用例として，観光や防災用途にネットワークカメラの映像や撮影角度の制御機能をインターネット上に公開している事例がある．一方で，個人がアクチュエータを含む多様なデバイスを共用する事例は未だ多くない．今後，個人を含むデバイス共用者の数を拡大するには，二つのことが必要と考えている．一つ目は，デバイス共用者へのインセンティブの付与である．提供するデータやデバイスの使用状況を管理し，課金，決済を確実に行えることが必要である．二つ目は，セキュリティとプライバシーの担保である．データとデバイスの種類に基づいて，地域やサービス利用者を限定し，適切な範囲にのみアクセスを許容する仕組みが不可欠である．

以上の課金，アクセスコントロールに関する課題解決に向けても，デバイスの物理的近傍

に配備され，データ転送を担う通信事業者の設備への実装が期待される．本研究の提案手法と併せて，これら機能のネットワーク設備への導入を進めていく必要がある．

C. サービス展開と研究開発ロードマップ

E-CPS の研究と実用化を推進するには，社会動向を踏まえたサービス展開と研究推進が必要である．サービス規模とデバイス多様化の 2 軸から成るロードマップを図 6.3 に示す．

まずは，技術難易度が低く，かつ，早期需要が想定されるサービスで提案手法の早期実用化を行うべきである．例えば，市町村，自治体等の小規模なサービスエリアにおける，Web カメラ，ディスプレイ等の普及済みデバイスを用いた，防犯，広告サービスが挙げられる．

次に，サービス規模拡大の流れとして，将来，世界規模リアルタイムデータに対する検索サービスの需要が予想される．このような広大なサービスエリアを対象にデータ検索を行うために，方式課題として述べたライブデータ検索の方式拡張が必要である．

さらに，デバイスの多様化について，普及には時間を要すると想定されるが，将来はロボットやコネクテッドカーのような，複雑に動作する高度なアクチュエータを活用するサービスの需要が予想される．これらを用いたサービスを実現するために，移動デバイスへの対応や状態遷移を伴うシステムへの対応といったデバイス自律制御の方式拡張が必要である．

サービス需要の拡大とデバイスの普及に合わせて，最終的に，広大な空間に分散する多種多様なデバイスを活用した高度なサービスを展開するべきである．例えば，自動運転等の，高いリアルタイム性とセキュリティ，信頼性が求められるサービスが挙げられる．

以上の通り，E-CPS の普及拡大には，社会動向と照らし合わせながら戦略的なサービス展開と研究開発を進めていくことが必要である．

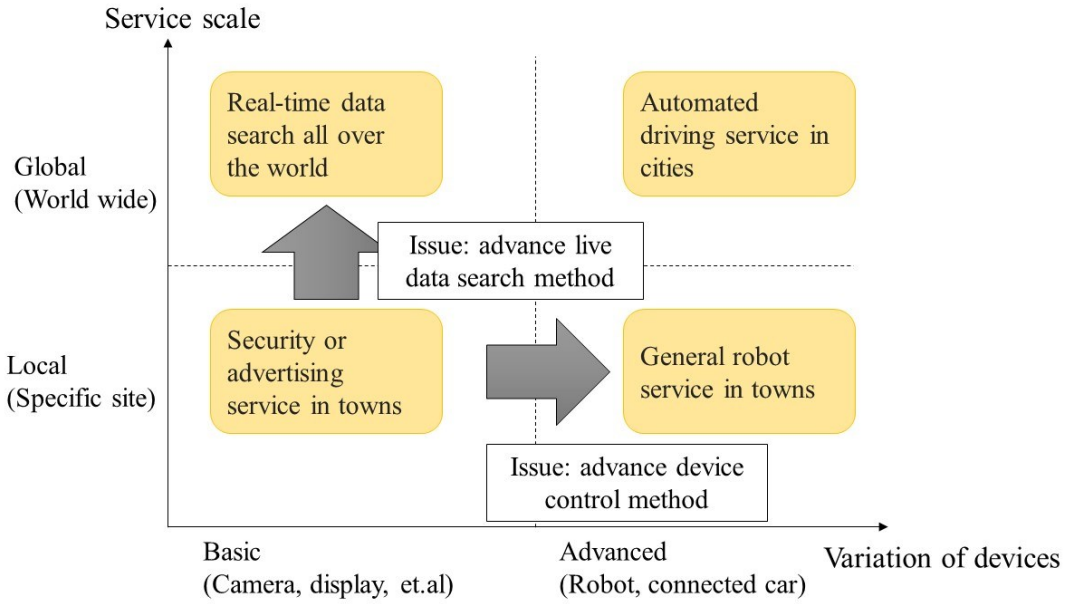


図 6.3 サービス展開と研究開発ロードマップ

參考文獻

[1] Cisco VNI [online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html

[2] J. Manyka, M. Chui, P. Bisson, J. Woetzel, R. Dobbs, J. Bughin, and D. Aharon, “The Internet of Things: Mapping the Value Beyond the Hype,” Technical Report, McKinsey Global Institute, June 2015.

[3] G. Chong, L. Zhihao, and Y. Yifeng, “The research and implement of smart home system based on internet of things,” in Electronics Communications and Control (ICECC) 2011 International Conference, Ningbo, China, pp. 2944-2947, Sept. 2011.

[4] Yin Jie, Yong Pei, Li Jun, Guo Yun, and Xu Wei, “Smart Home System based on IOT Technologies,” in 2013 International Conference on Computational and Information Sciences, IEEE, 2013.

[5] Jay Lee, “Smart factory systems,” Informatik-Spektrum, 38.3, pp. 230-235, 2015.

[6] Stamatis Karnouskos, “Cyber-Physical Systems in the SmartGrid,” in 9th IEEE International Conference on Industrial Informatics, pp20-23, 2011.

[7] Yin Zhang, Meikang Qiu, Chun-Wei Tsai, Mohammd Mehedi Hassan, and Atif Alamri, “Health-CPS: Healthcare Cyber-Physical System Assisted by Cloud and Big Data,” IEEE Systems Journal, vol. 11, no. 1, pp. 88-95, 2015.

[8] Luca Catarinucci, Danilo de Donno, Uca Mainetti, Luca Palano, Luigi Patroono, Maria Laura Stefanizzi, and Luciano Tarricone, “An IoT-Aware Architecture for Smart Healthcare Systems,” IEEE Internet of Things Journal, vol. 2, no. 6, pp. 515–526, 2015.

[9] Ji-chun Zhao, Jun-feng Zhang, Yu Feng, and Jian-xin Guo, “The Study and Application of the IOT Technology in Agriculture,” in 3rd International Conference on Computer Science and Information Technology, vol. 2, pp. 462-465, IEEE, 2010.

[10] Fan TongKe, “Smart Agriculture Based on Cloud Computing and IOT,” Journal of Convergence Information Technology, vol. 8, no. 2, Jan. 2013.

[11] Aditya Gaur, Bryan Scotney, Gerard Parr, and Sally McClean, “Smart city architecture and its applications based on IoT,” Procedia Computer Science, 52, pp. 1089-1094, 2015.

[12] K E Skouby, and P Lynggaard, “Smart Home and Smart City Solutions enabled by 5G, IoT, AAI and CoT Services,” in International Conference on Contemporary Computing and Informatics (IC3I), pp. 874-878, IEEE, Nov. 2014.

[13] Justin M.Bradley and Ella M. Atkins, “Optimization and Control of Cyber-Physical Vehicle Systems,” Sensors, vol. 15, no. 9, pp. 23020-23049, 2015.

[14] 第 5 期科学技術基本計画 [online]. Available: https://www8.cao.go.jp/cstp/kihonkeikaku/5honbun.pdf

[15] society 5.0, 内閣府 [online]. Available: https://www8.cao.go.jp/cstp/society5_0/index.html

[16] 本田敏, 永原正章,“超スマート社会実現に向けた計測自動制御学会の取り組み,”SICE Oukan vol. 12, no. 1, 2018.

[17] 原辰次, 本田敏, “超スマート社会におけるシステム科学技術概論,” 計測と制御 第 55 巻 第 4 号 pp.284-287, 2016 年 4 月.

[18] 貝原俊也, 下原勝憲, “System of Systems コンセプトと超スマート社会,” 計測と制御 第 55 巻 第 4 号 pp.288-290, 2016 年 4 月.

[19] Ivan Stojmenovic, “Machine-to-Machine Communications With In-Network Data Aggregation, Processing, and Actuation for Large-Scale Cyber-Physical Systems,” IEEE Internet of Things Journal, vol. 1, no. 2, Apr. 2014.

[20] 戦略的イノベーション創造プログラム（SIP） [online]. Available: https://www8.cao.go.jp/cstp/gaiyo/sip/

[21] 戦略的イノベーション創造プログラム（SIP）「ビッグデータ・AI を活用したサイバー空間基盤技術」研究開発計画（案） [online]. Available: https://www8.cao.go.jp/cstp/gaiyo/sip/iinkai2/bigdata_1/siryo2-2.pdf

[22] 戦略的イノベーション創造プログラム（SIP） フィジカル空間デジタルデータ処理基盤研究開発計画(案) [online]. Available: https://www8.cao.go.jp/cstp/gaiyo/sip/83kai/siryo2-2.pdf

[23] Amazon Web Service [online]. Available: https://aws.amazon.com/jp/

[24] Microsoft Azure [online]. Available: https://azure.microsoft.com/ja-jp/

[25] Google Cloud [online]. Available: https://cloud.google.com/

[26] 日高洋祐, 牧村和彦, 井上岳一, 井上佳三, “MaaS モビリティ革命の先にある全産業のゲームチェンジ,” 日経 BP 社.

[27] 平成 30 年版 情報通信白書, 総務省 [online]. Available: http://www.soumu.go.jp/johotsusintokei/whitepaper/h30.html

[28] 財務局調査による「先端技術(IoT, AI 等)の活用状況」について [online]. Available: https://www.mof.go.jp/about_mof/zaimu/kannai/201803/sentangizyutu091.pdf

[29] Softbank Pepper [online]. Available: https://www.softbank.jp/robot/pepper/

[30] 我が国の ICT の現状に関する調査研究（2018） [online]. Available: https://www.soumu.go.jp/johotsusintokei/linkdata/h30_01_houkoku.pdf

[31] IBM node-red [online]. Available: https://www.ibm.com/developerworks/ibmi/library/i-running-node-red/

[32] Giovanni Merlino, Dario Bruneo, Salvatore Distefano, Francesco Longo, and Antonio Puliafito, “Stack4Things: integrating IoT with OpenStack in a Smart City context,” in 2014 International

Conference on Smart Computing Workshops, IEEE, 2014, pp. 21-28, 2014.

[33] Francesco Longo, Dario Bruneo, Salvatore Distefano, Giovanni Merlino, and Antonio Puliafito, “Stack4Things: a sensing-and-actuation-as-a-service framework for IoT and cloud integration,” Annals of Telecommunications, vol. 72, no.1-2, pp. 53-70, 2017.

[34] OpenStack [online]. Available: https://www.openstack.org/

[35] Tiantian Zhang, Bo Yuan, Tao Meng, Yinghao Ren, Houde Liu, and Xueqian Wang, “Ubiquitous Robot: A New Paradigm for Intelligence,” International Conference on Intelligent Data Engineering and Automated Learning, Springer, pp. 202–211, 2016.

[36] Shuichi Nishio, Koji Kamei, and Norihiro Hagita, “Ubiquitous network robot platform for realizing integrated robotic applications,” Intelligent Autonomous Systems 12, Springer, Berlin, Heidelberg, pp. 477–484, 2013.

[37] 松元崇裕, 松村成宗, 細淵貴司, 望月崇由, 吉川博, 山田智広, “「R-env:連舞™」クラウド対応型インタラクション制御技術～デバイスを連携させたサービスを簡易に作成するためのアーキテクチャ～,” 第 30 回人工知能学会全国大会, 2016.

[38] ROS [online]. Available: https://www.ros.org/

[39] Siddhartha Kumar Khaitan and James D. McCalley, “Design Techniques and Applications of Cyberphysical Systems: A Survey,” IEEE Systems Journal, vol. 9, no. 2, pp. 350-365, June 2015.

[40] Alvaro A. Cardenas, Saurabh Amin, and Shankar Sastry, “Secure Control: Towards Survivable Cyber-Physical Systems,” in the 28th International Conference on Distributed Computing Systems Workshops, pp. 495-500, 2008.

[41] Rodrigo Roman, Jianying Zhou, and Javier Lopez, “On the features and challenges of security and privacy in distributed internet of things,” Computer Networks, vol. 57, no. 10, pp. 2266-2279, 2013.

[42] Luigi Atzori, Antonio Iera, andGiacomo Morabito, “The Internet of Things: A survey,” Computer Networks, vol. 54, no. 15, pp. 2787-2805, 2010.

[43] Tie Qiu, Ning Chen, Keqiu Li, Mohammed Atiquzzaman, and Wenbing Zhao, “How Can Heterogeneous Internet of Things Build Our Future: A Survey,” IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2011-2027, 2018.

[44] Jehan Wickramasuriya, Mahesh Datt, Sharad Mehrotra, and Nalini Venkatasubramanian, “Privacy Protecting Data Collection in Media Spaces,” in Proceedings of the 12th annual ACM international conference on Multimedia, pp. 48-55, 2004.

[45] Fardin Abdi Taghi Abad, Marco Caccamo, and Brett Robbins, “A Fault Resilient Architecture for Distributed Cyber-Physical Systems,” in IEEE International Conferene on Embedded and Real-Time Computing Systems and Applications, pp. 222-231, 2012.

[46] Feng Xia, Longhua Ma, Jinxiang Dong, and Youxian Sun, “Network QoS Management in Cyber-Physical Systems,” in the 2008 International Conference on Embedded Software and Systems

Symposia, pp. 302-307, 2008.

[47] Jaiganesh Balasubramanian, Sumant Tambe, Aniruddha Gokhale, Balakrishnan Dasarathy, Shriang Gadgil, and Douglas C. Schmidt, “A Model-driven QoS Provisioning Engine for Cyber Physical Systems," Technical Report, 2010 [online]. Available: https://pdfs.semanticscholar.org/d60d/3e91da1b2627127799793aef2fc6d6981be1.pdf

[48] Kyoung-Don Kang and Sang H. Son, “Real-Time Data services for Cyber Physical Systems,” in the 28th International Conference on Distributed Computing Systems Workshops, pp. 483-488, 2008.

[49] Dave Reggett, “The Web of Things: Challenges and Opportunities,” Computer, vol. 48, no. 5, pp. 26-32, IEEE Computer society, 2015.

[50] W3C [online]. Available: https://www.w3.org/

[51] Pratikkumar Desai, Amit Sheth, and Pramod Anantharam, “Semantic Gateway as a Service Architecture for IoT Interoperability,” in IEEE International Conference on Mobile Services, pp. 313-319, 2015.

[52] Benoit Christophe, Vincent Verdot, and Vincent Toubiana, “Searching the 'Web of Things',” in fifth IEEE International Conference on Semantic Computing, pp. 308-315, 2011.

[53] OneM2M [online]. Available: http://www.onem2m.org/

[54] AllJoyn [online]. Available: https://events.static.linuxfound.org/sites/events/files/slides/AllJoynThinLibraryTargetPorting_PeterKrystad_MathewMartineau.pdf

[55] GotAPI [online]. Available: https://en.device-webapi.org/gotapi.html

[56] ECHONET [online]. Available: https://echonet.jp/english/

[57] Z. Shelby, “Constrained RESTful Environments (CoRE) Link Format,” RFC6690, Aug. 2012.

[58] Z. Shelby, M. Koster, C. Bormann, and P. van der Stok, “CoRE Resource Directory draft-ietf-core-resource-directory-09,” Internet-Draft, Oct. 2016.

[59] S. Cheshire and M. Krochmal, “Multicast DNS,” RFC6762, Feb. 2013.

[60] Tamilarasi K. and Ramakrishnan M, “Design of an intelligent search engine-based UDDI for web service discovery,” in Recent Trends in Information Technology (ICRTIT) 2012 International Conference on. IEEE, pp. 520-525, 2012.

[61] J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters,” Communications of the ACM, vol. 51, issue 1, pp. 107-113, Jan. 2008.

[62] J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters,” in 6th conference on Symposium on Operating Systems Design & Implementation, San Francisco, CA, USA, pp. 137-150, Dec. 06-08, 2004.

[63] S. Ghemawat, H. Gobioff, and S. Leung, “The Google file system,” in Proceedings of the nineteenth ACM symposium on Operating systems principles, Bolton Landing, NY, USA, Oct.

19-22, 2003.

[64] A. Toshniwal, S. Taneja, A. Shukla, K. Ramasamy, J. M. Patel, S. Kulkarni, J. Jackson, K. Gade, M. Fu, J. Donham, N. Bhagat, S. Mittal, and D. Ryaboy, “Storm@twitter,” in 2014 ACM SIGMOD international conference on Management of data, Snowbird, Utah, USA, pp. 147-156, June 22-27, 2014.

[65] UPnP [online]. Available: https://openconnectivity.org/developer/specifications/upnp-resources/upnp#architectural

[66] SNMP v3 (RFC 3411 - RFC 3418, STD0062) [online]. Available: https://tools.ietf.org/html/rfc3411

[67] NETCONF (RFC 6241) [online]. Available: https://tools.ietf.org/html/rfc6241

[68] Anuj Sehgal, Vladislav Perelman, Siarhei Kuryla, and Jürgen Schönwälder, “Management of resource constrained devices in the internet of things,” IEEE Communications Magazine, vol. 50, no. 12, pp. 144-149, 2012.

[69] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz, “The web never forgets: Persistent tracking mechanisms in the wild,” in Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Arizona, USA, pp. 674-689, Nov. 2014.

[70] Gunes Acar, Marc Juarez, Nick Nikiforakis, Claudia Diaz, Seda Gürses, Frank Piessens, and Bart Preneel, “FPDetective: dusting the web for fingerprinters,” in 2013 ACM SIGSAC conference on Computer & communications security, Berlin, Germany, pp. 1129-1140, Nov. 2013.

[71] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh, “Wireless device identification with radiometric signatures,” in 14th ACM international conference on Mobile computing and networking, pp. 116-127, ACM, 2008.

[72] B. M. Elahi, K. Römer, B. Ostermaier, M. Fahrmair, and W. Kellerer, “Sensor Ranking: A Primitive for Efficient Content-Based Sensor Search,” in Proceedings of the 2009 international conference on information processing in sensor networks, pp. 217-228, IEEE Computer Society, 2009.

[73] C. Perera, A. Zaslavsky, C. H. Liu, M. Compton, P. Chirsten, and D. Georgakopoulos, “Sensor search techniques for sensing as a service architecture for the Internet of Things,” IEEE Sensors Journal, vol. 14, no. 2, pp. 406-420, 2013.

[74] Dennis Pfisterer, Kay Romer, Daniel Bimschas, Henning Hasemann, Manfred Hauswirth, Marcel Karnstedt, Oliver Kleine, Alexander Kroller, Myriam Leggieri, Richard Mietz, Max Pagel, Alexandre Passant, Ray Richardson, and Cuong Truong, “SPITFIRE: toward a semantic web of things,” IEEE Communications Magazine, vol. 49, issue 11, pp. 40–48, 2011.

[75] Konstantinos Kotis and Artem Katasonov, “Semantic Interoperability on the Web of Things: The Semantic Smart Gateway Framework,” in 2012 Sixth International Conference on Complex,

Intelligent, and Software Intensive Systems, IEEE, 2012.

[76] Van Jacobson, Diana K. Smetters, James D. Thrnton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard, “Networking Named Content,” in Proceedings of the 5th international conference on Emerging networking experiments and technologies, pp. 1-12, ACM, 2009.

[77] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Borje Ohlman, “A survey of Information-Centric Networking,” IEEE Communications Magazine, vol. 50, no. 7, pp. 26-36., 2012.

[78] George Xylomenos, Christopher N. Ververidis, Vasilios A. Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V. Katsaros, and George C. Polyzos, “A Survey of Information-Centric Networking Research,” IEEE Communications Surveys & Tutorials, vol. 16, no. 2, pp. 1024-1049, 2013.

[79] ORACLE Virtual Box [online]. Available: https://www.virtualbox.org/

[80] Linux container [online]. Available: https://linuxcontainers.org/

[81] Y. Jaraweh, A. Doulat, O. AlQudah, E. Ahmed, M. Al-Ayyoub, and E. Benkhelifa, “The future of mobile cloud computing: Integrating cloudlets and Mobile Edge Computing,” in 23rd International Conference on Telecommunications (ICT2016), pp. 1-5, Thessaloniki, Greece, May 2016.

[82] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu, “Edge Computing: Vision and Challenges,” IEEE Internet of Things Journal, vol. 3, no, 5, pp. 637-646, 2016.

[83] KVM [online]. Available: https://www.linux-kvm.org/page/Main_Page

[84] Takashi Matsunaka, Akira Yamada, and Ayumu Kubota, “Passive OS fingerprinting by DNS traffic analysis,” Advanced Information Networking and Applications (AINA), IEEE 27th International Conference On. IEEE, Barcelona, Spain, pp. 243-250., 2013.

[85] Deliang Chang, Qianli Zhang, and Xing Li, “Study on OS fingerprinting and nat/tethering based on DNS log analysis,” in Workshop on Research and Applications of Internet Measurements (RAIM), Yokohama, Japan., 2015.

[86] Yair Meidan, Michael Bohadana, Asaf Shabtai, Juan David Guarnizo, Martin Ochoa, Nils Ole Tippenhauer, and Yuval Elovici, “ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis,” in Symposium on Applied Computing, ACM, 2017, pp. 506-509, 2017.

[87] Hiroki Kawai, Shingo Ata, Nobuyuki Nakamura, and Ikuo Oka, “Identification of communication devices from analysis of traffic patterns,” in 13th International Conference on Network and Service Management, Tokyo, Japan, Nov. 2017.

[88] Asish Kumar Dalai and Sanjay Kumar Jena, “WDTF: A Technique for Wireless Device Type Fingerprinting,” Wireless Personal Communications, vol. 97, no. 2, pp. 1911-1928, 2017.

[89] Miettinen Markus, Samuel Marchal, Ibbad Hafeez, N. Asokan, Ahmad-Reza Sadeghi, and Sasu

Tarkoma, “IoT Sentinel: Automated device-type identification for security enforcement in IoT,” in IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp. 2177-2184, 2017.

[90] Scikit-learn [online]. Available: https://scikit-learn.org/stable/

[91] Jiming Chen, Xianghui Cao, Peng Cheng, Yang Xiao, and Youxian Sun, “Distributed Collaborative Control for Industrial Automation With Wireless Sensor and Actuator Networks,” IEEE Transactions on Industrial Electronics, vol. 57, no. 12, Dec. 2010.

[92] Chenyang Lu, Abusayeed Saifullah, Bo Li, MoSha, Humberto Gonzalez, Dolvara Gunatilaka, Chengjie Wu, Lanshun Nie, and Yixin Chen, “Real-Time Wireless Sensor-Actuator Networks for Industrial Cyber-Physical Systems,” in Proceedings of the IEEE, 104(5), pp. 1013-1024, 2016.

[93] Johan Akerberg, Mikael Gidlund, Mats Bjorkman, “Future Research Challenges in Wireless Sensor and Actuator Networks Targeting Industrial Automation,” in 9th IEEE International Conference on Industrial Informatics, pp. 410-415, IEEE, 2011.

[94] Jim Gao, “Machine learning applications for data center optimization,” Google White Paper, 2014, [online]. Available: https://static.googleusercontent.com/media/research.google.com/ja//pubs/archive/42542.pdf.

[95] Jens Kober, Erhan Oztop, and Jan Peters, “Reinforcement learning to adjust robot movements to new situations,” in IJCAI Proceedings-International Joint Conference on Artificial Intelligence, pp. 2650-2655, 2011.

[96] Thomas Bäck, Frank Hoffmeister, and Hans-Paul Schwefel, “A survey of evolution strategies,” in Proceedings of the fourth international conference on genetic algorithms, pp. 2-9, Morgan Kaufmann Publishers San Mateo, CA, 1991.

[97] Martin T. Hagan, Howard B. Demuth, Mark Hudson Beale, and Orlando De Jesus, “Neural network design,” Martin Hagan, 1996.

[98] Yann Lecun, Yohua Bengio, and Geoffrey Hinton, “Deep learning,” nature, vol. 52, pp. 436-444, 2015.

[99] Jianming Zhang, Chaoquan Lu, Xudong Li, Hye-Jin Kim, and Jin Wang, “A full convolutional network based on DenseNet for remote sensing scene classification,” Mathematical Biosciences and Engineerring, vol. 16, no. 5, pp. 3345-3367, 2019.

[100] Jianming Zhang, Xiaokang Jin, Juan Sun, Jin Wang, and Arun Kumar Sangaiah, “Spatial and Semantic convolutional features for robust visual object tracking,” Multimedia Tools and Applications, 2018.

[101] Philips Hue [online]. Available: https://www2.meethue.com/en-us/p/hue-white-and-color-ambiance-starter-kit-e26/046677530228

[102] OpenCV [online]. Available: https://opencv.org/

[103] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore, “Reinforcement

learning: A survey," Journal of artificial intelligence research, vol. 4, pp. 237-285, 1996.

[104] Xenofon Foukas, Georgios Patounas, Ahmed Elmokashfi, and Mahesh K. Marina, "Network Slicing in 5G: Survey and Challenges," IEEE Communications Magazine, vol. 55, no.5, pp. 94-100, 2017.

[105] Yong li and Min Chen, "Software-Defined Network Function Virtualization: A Survey," IEEE Access, vol. 3, 2169-3536, 2015.

[106] Mehdi Mohammadi, Ala Al-Fuqaha, Sameh Sorour, and Mohsen Guizani, "Deep Learning for IoT Big Data and Streaming Analytics: A Survey," IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 2923-2960, 2018.

[107] Sam Newman (著), 佐藤 直生 (監修), 木下 哲也 (翻訳), "マイクロサービスアーキテクチャ," オライリー・ジャパン 2016.

[108] Intel-VT [online]. Available: https://www.intel.com/content/www/us/en/virtualization/virtualization-technology/intel-virtualization-technology.html

[109] SR-IOV [online]. Available: https://docs.microsoft.com/en-us/windows-hardware/drivers/network/single-root-i-o-virtualization--sr-iov-?redirectedfrom=MSDN

[110] DPDK [online]. Available: https://www.dpdk.org/

[111] Vmware ESXi [online]. Available: https://www.vmware.com/jp/products/esxi-and-esx.html

[112] Docker [online]. Available: https://www.docker.com/

[113] 3GPP [online]. Available: https://www.3gpp.org/

謝辞

本論文は，筆者が，2016 年から現在に至るまでに日本電信電話（株）ネットワークサービスシステム研究所にて従事してきた将来 IoT の研究開発業務および，同社と早稲田大学菅野重樹研究室との共同研究成果をまとめたものです．早稲田大学ならびに日本電信電話（株）の多くの方々にご協力，ご支援を賜りましたことを深く感謝いたします.

はじめに，本論文の執筆にあたり，終始懇切なるご指導，ご鞭撻を賜りました，早稲田大学大学院創造理工学研究科　菅野重樹教授に心より感謝の意を表します.

また，本論文の執筆にあたり，多くのご指導を賜りました，早稲田大学理工学術院高西淳夫教授，岩田浩康教授，奥乃博教授，早稲田大学　藤江正克名誉教授に深く感謝いたします.

そして，本研究の共同研究者として，終始協力を頂きました，日本電信電話（株）池邉隆氏, 同社ネットワークサービスシステム研究所　山登庸次氏, 服部恭太氏, 片岡操氏, 磯田卓万氏，西日本電信電話（株）　出水達也氏，小山工業高等専門学校　干川尚人氏に深く感謝いたします.

最後に, 筆者の研究生活を支えてくださった, 妻, 両親, 妹に心より感謝を申し上げます.

研究業績

項番に○を付したものは，本研究に関する査読付き主著論文，国際会議.

種別	項番	題名， 発表・発行掲載誌名， 発表・発行年月， 著者
論文	○1	Ephemeral-Cyber-Physical System: A Cloud-Like CPS Using Shared Devices in Open IoT, IEEE Systems Journal, Hirofumi Noguchi, Shigeki Sugano (掲載決定)
	○2	Device Identification Based on Communication Analysis for the Internet of Things, IEEE Access, vol. 7, pp. 52903-52912, Apr. 2019, Hirofumi Noguchi, Misao Kataoka, Yoji Yamato
	○3	Distributed Search Architecture for Object Tracking in the Internet of Things, IEEE Access, vol. 6, pp. 60152-60159, Oct. 2018, Hirofumi Noguchi, Tatsuya Demizu, Misao Kataoka, Yoji Yamato
講演（査読付）	1	Evaluation of a Realistic Example of Information-Centric Network Metadata Management, 2019 10th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), Oct. 2019, Tomoki Ito, Hirofumi Noguchi, Misao Kataoka, Takuma Isoda, Yoji Yamato, Tutomu Murase
	2	Network Architecture with Categorizing Metadata by Locality and Lifetime for IoT Database Management, 2018 21st International Symposium on Wireless Personal Multimedia Communications (WPMC), pp. 177-181, Nov. 2018, Tomoki Ito, Misao Kataoka, Hirofumi Noguchi, Yoji Yamato, Tutomu Murase
	3	Transaction Offloading for Access Management to Live Data of IoT in Information-Centric Network, 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE), pp. 287-288, Oct. 2018, Tomoki Ito, Hirofumi Noguchi, Yoji Yamato, Tutomu Murase
	4	Data Management and Packet Transmission Method based on Receivers' Attributes, 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), pp. 186-190, Feb. 2018, Tatsuya Demizu, Hirofumi Noguchi, Naoto Hoshikawa, Misao Kataoka, Yoji Yamato
	○5	Autonomous Device Identification Architecture for Internet of Things, 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), pp. 407-411, Feb. 2018, Hirofumi Noguchi, Tatsuya Demizu, Naoto Hoshikawa, Misao Kataoka, Yoji Yamato
	6	Tacit Computing and its Application for Open IoT Era, 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), Jan. 2018, Misao Kataoka, Naoto Hoshikawa, Hirofumi Noguchi, Tatsuya Demizu, Yoji Yamato
	7	Distributed Live Data Search Architecture for Resource Discovery on Internet of Things, 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), pp. 591-596, Dec. 2016, Takashi Ikebe, Hirofumi Noguchi, Naoto Hoshikawa

種別	項番	題名，　発表・発行掲載誌名，　発表・発行年月，　著者
講演 （査読無）	8	IoT データの重複を考慮した可用性向上手法，電子情報通信学会　ソサイエティ大会，2019 年 9 月，片岡操，野口博史
	9	通信分析と機械学習によるデバイス識別手法に関する検討，電子情報通信学会　総合大会，2019 年 3 月，野口博史，片岡操，磯田卓万，山登庸次
	10	デバイスのサービス有効度を用いた最適配置選択方式，電子情報通信学会　総合大会，2019 年 3 月，片岡操，野口博史，磯田卓万，山登庸次
	11	IoT デバイスのデータ処理におけるネットワーク上のリソース最適配置の提案，電子情報通信学会　総合大会，2019 年 3 月，磯田卓万，野口博史，片岡操，山登庸次
	12	ICN における IoT メタデータ管理手法の実際的評価，電子情報通信学会　情報ネットワーク研究会，2019 年 3 月，伊藤智稀，野口博史，片岡操，磯田卓万，山登庸次，村瀬勉
	13	DB 更新負荷軽減のための IoT データ特性を考慮したメタデータ指向ネットワークアーキテクチャ，電子情報通信学会　情報ネットワーク研究会，2018 年 10 月，伊藤智稀，片岡操，野口博史，山登庸次，村瀬勉
	14	IoT データのローカリティを考慮したメタデータ管理区別，電気・電子・情報関係学会東海支部連合大会，2018 年 9 月，伊藤智稀，片岡操，野口博史，山登庸次，村瀬勉
	15	通信の類似性分析にもとづく IoT デバイス識別手法，電子情報通信学会　ネットワークシステム研究会，2018 年 7 月，野口博史，出水達也，片岡操，山登庸次
	16	ユーザー周辺環境のマルチコンピュータリソース活用に関する検討，電子情報通信学会　ネットワークシステム研究会，2017 年 3 月，野口博史，池邉隆，佐々木潤子
	17	ネットワーク機能仮想化に向けた課題と目指す機能の検討，電子情報通信学会　ネットワークソフトウェア研究会，2017 年 1 月，野口博史，日高直人，浜田信，森谷俊之
	18	Network-AI:さまざまなモノが賢くつながるネットワーク，電子情報通信学会　ネットワークシステム研究会　招待講演，2016 年 9 月，池邉隆，千川尚人，野口博史
	19	クラウドインフラにおけるマルチホスト OS 設定管理方式，電子情報通信学会　ソサイエティ大会，2016 年 9 月，野口博史，森谷俊之
	20	順序保証と復旧処理を考慮した複数 VM 自動起動方式，電子情報通信学会　総合大会，2016 年 3 月，野口博史，森谷俊之
	21	高可用ストリーミングサーバクラスタ構成方式に関する一検討，電子情報通信学会　総合大会，2015 年 3 月，野口博史，藤岡幸，福元健
	22	効率的なクラスタ運用を実現するプロセス配置方式，電子情報通信学会　ソサイエティ大会，2014 年 9 月，野口博史，小川泰文，福元健
	23	広域ネットワークにおける高可用サーバクラスタ構成方式に関する一検討，電子情報通信学会　総合大会，2014 年 3 月，野口博史，森谷俊之，福元健

種別	項番	登録番号, 名称, 発明者
登録特許	1	特許第 6530353 号，ライブデータ検索システムおよびライブデータ検索方法，池邉隆，野口博史
	2	特許第 6495871 号，リソース管理システム、リソース管理サーバ及びリソース管理方法，野口博史，森谷俊之
	3	特許第 6383336 号, サーバ管理装置およびサーバ管理方法, 野口博史, 福元健
	4	特許第 6445985 号，振分システム、振分装置、及び振分方法，出水達也，池邉隆，舩田雅史，野口博史，梅川慎吾
	5	特許第 6204287 号，分散処理方法、処理サーバ、および、プログラム，北野雄大，小西啓介，徐広幸，外山篤史，藤岡幸，野口博史
	6	特許第 6170476 号，ハッシュ評価サーバ、ハッシュ評価システム及びハッシュ評価方法，外山篤史，小西啓介，北野雄大，徐広幸，藤岡幸，野口博史
	7	特許第 6235973 号，サーバ，野口博史
	8	特許第 6343241 号，ストリーミングサーバクラスタおよびそのストリーミング制御方法，野口博史，福元健，藤岡幸，樫本義文
	9	特許第 6251203 号，ストリーミングデータ配信システム、及び、ストリーミングデータ配信方法，藤岡幸，野口博史，金子雅志，福元健
	10	特許第 5848743 号，クラスタシステム，野口博史，岩佐絵里子
	11	特許第 6093319 号，クラスタシステム，野口博史，福元健，堀米紀貴，小川泰文，大谷育生
	12	特許第 5932875 号，サーバ装置およびプログラム，金子雅志，野口博史