

# ホームネットワークシステムにおける家電状態を利用した 音声操作インタフェースの改善

榊原 弘記<sup>†</sup> 中村 匡秀<sup>††</sup> 井垣 宏<sup>††</sup> 松本 健一<sup>†</sup>

<sup>†</sup> 奈良先端科学技術大学院大学 情報科学研究科 〒630-0192 奈良県生駒市高山町 8916-5

<sup>††</sup> 神戸大学 大学院 工学研究科 情報知能学専攻 〒657-8501 神戸市灘区六甲台町 1-1

E-mail: <sup>†</sup>{hiroki-s,matumoto}@is.naist.jp, <sup>††</sup>{masa-n,igaki}@cs.kobe-u.ac.jp

あらまし ホームネットワークシステム (HNS) には、多様なネット家電やセンサ機器が接続されるため、従来用いられてきた個別機器の専用リモコンに代わる新しい操作方法が求められている。特に既に実用例も多い音声操作インタフェースは今後の普及が期待されている。HNS に音声操作インタフェースを適用する際には、(1) ユーザが事前に音声コマンドを覚えておかなければならないため機器やその操作種類が増えると学習コストが増加する、(2) 音声認識エンジンの誤認識により、ユーザの意図しない機器操作が実行される可能性がある、といった問題点が考えられる。そこで本稿では、HNS のための対話型操作インタフェースを提案する。提案手法では、システムがユーザとの対話を通して、機器名、操作名、パラメータの順番でユーザが段階的にコマンドを組み立てる段階的コマンド構築機構を導入することで、ユーザが事前に記憶していなければならない音声コマンドの削減を図る。さらに、操作対象の家電が現状態で実行可能な操作のみをシステムが提示・解釈する家電状態を利用したコマンド提示・解釈機構を採用することで、実行不可能な音声コマンドの解釈を減らし、結果として誤認識を削減することができる。これらの手法を適用した提案インタフェースと従来手法の比較実験を行った結果、学習コストや誤認識率が減少したことがいくつかの指標により確認できた。

キーワード ホームネットワークシステム、音声認識、音声リモコン、対話型インタフェース、家電機器状態管理

## Improving Voice Command Interface to Operate Home Appliances Using Appliance States

Hiroki SAKAKIBARA<sup>†</sup>, Masahide NAKAMURA<sup>††</sup>, Hiroshi IGAKI<sup>††</sup>, and Ken-ichi  
MATSUMOTO<sup>†</sup>

<sup>†</sup> Graduate School of Information Science, Nara Institute of Science and Technology  
8916-5, Takayama, Ikoma, Nara 630-0192 Japan

<sup>††</sup> Department of Computer Sciences and Systems Engineering, Graduate School of Engineering, Kobe  
University Rokko-dai 1-1, Nada-ku, Kobe, 657-8501 Japan

E-mail: <sup>†</sup>{hiroki-s,matumoto}@is.naist.jp, <sup>††</sup>{masa-n,igaki}@cs.kobe-u.ac.jp

**Abstract** Various appliances are connected to Home Network System(HNS) ubiquitously, so the novel user interface is required to control HNS instead of remote controllers. Especially voice interface, there have already been examples of practical use, so popularization is expected. However, user has to remember voice command to use conventional voice interface, so the learning cost should be very high to operate HNS. In this paper, we propose novel interactive voice interface for HNS. In the proposed method, we introduce gradual command construction structure and command giving / interpretation structure using appliances' state.

**Key words** Home Network System, voice recognition, voice remote controller, interactive voice interface, home appliance state management

## 1. はじめに

ユビキタス技術の進歩に伴い、近い将来、宅内のあらゆるモノがネットワークに接続されることが予想されている [1]。ホームネットワークシステム (HNS) は、宅内の様々な家電やセンサをネットワークに接続し便利で快適なサービスを実現する次世代のユビキタスアプリケーションの枠組みとして注目されている。家電をネットワークに接続することで、宅内外からの遠隔監視や遠隔操作、さらには複数機器の連携や外部ネットワークリソースとの融合など、様々な付加価値サービスの登場が期待されている [2]。

HNS における重要な研究課題のひとつとして、操作インタフェースが挙げられる。HNS はその構成機器の多様性・偏在性から、従来家電で用いられてきた個別機器の専用リモコンや操作パネルといった操作方法では限界があると考えられており、多様な機器制御や新たな機器・サービスの追加にも対応可能な新しい操作インタフェースへの期待が高まっている。

本稿では、HNS に適用可能なインタフェースとして、音声を利用したものに注目する。従来から、様々な機器を音声で操作する方法が提案されているが、それらの従来アプリケーションの多くは、ユーザの発話 (音声コマンド) を機器操作のために必要な API 呼び出しに単純にマッピングする手法 (フルコマンド法と呼ぶ) にもとづいている。フルコマンド法では、ユーザが事前に全ての音声コマンドを覚えておく必要があるため、機器の種類、数が多くなりがちな HNS では、音声コマンド学習コストが非常に高くなる可能性がある。また、システムが正しくコマンドを認識するかどうかは、音声認識エンジンの性能や環境状態に大きく左右される。コマンドの誤認識による不具合や事故を最小限に抑えるためにも、認識精度のさらなる向上に対する取り組みが必要である。

これらの問題を改善するために、本稿では HNS のための新しい対話型コマンドインタフェースを提案する。提案手法は、以下の 2 つのキーアイデアにもとづいている。まず 1 つ目として、段階的コマンド構築機構を提案する。提案法では、必ずしも一度の発話で全ての音声コマンドを指定する必要が無い。システムはユーザとの対話を通して、機器名、操作名、パラメータの順番で、ユーザが段階的かつ正確に音声コマンドを組み立てるための支援を行う。これによりユーザは完全な音声コマンドを覚えていなくとも意図する機器制御を実行することができる。

2 つ目は家電状態を利用したコマンド提示・解釈機構である。HNS 内に配置されるネット家電はそれぞれ状態を持ち、状態に応じた振る舞いを提供する。HNS ではすべての家電の状態をネットワーク越しに取得することができる。したがって、対象の家電が現状で実行可能な操作のみをシステムが提示・解釈することにより、言い間違いや雑音による音声コマンドの誤認識の結果によるありえない動作を削減し、結果としてユーザの意図しない動作を減らすことができる。

また、本稿では、音声認識エンジン Julius for SAPI [3] を用いて、提案手法の実装を行い、現在運用中の HNS (NAIST-

HNS) [2] に組み込み、評価実験を行った。実験では、12 名の被験者に対して従来のフルコマンド法にもとづいた音声操作と提案手法による音声操作を行ってもらい、性能比較を行った。評価実験の結果、提案手法ではフルコマンド法に比べて、コマンド誤認識によるシステムの誤動作が平均 75%削減された。

## 2. 準備

### 2.1 ホームネットワークシステム (HNS)

HNS はネットワーク家電とそれらを制御するホームサーバから構成される。ネットワーク家電は API を公開しており、この機器 API を通じた機器制御や機器動作状態の獲得が可能である。機器 API を呼び出すためには、操作する機器名、実行する操作名、実行に必要なパラメータの 3 つの情報が必要である。本稿ではこの 3 つの情報をあわせて機器操作のためのフルコマンドと呼ぶ。

HNS 機器の操作インタフェースとして代表的なものに各機器の専用リモコンがある。個別機器専用のリモコンは HNS 環境だけでなく、従来の様々な家電に対して最もよく用いられてきたインタフェースである。これらのインタフェースは機器に備わった機能のほぼすべてを利用できるが、操作対象によってはボタンの数が増え、操作が煩雑になりやすい。

このようなリモコンを用いたインタフェースの場合、操作したい機器のリモコンとそのボタン配置を学習しなければならない。そのため、HNS に配備される機器の種類や数が増加すると、ユーザの学習コストが増加し、操作が煩雑になる傾向がある。特に多くの高齢者や障害者が (a) リモコンの液晶表示やボタンラベルの視認が困難である、(b) 機能やボタンの数が多すぎて操作が難しい、とリモコン操作時に感じている [4]。

また、高齢者や障害者だけでなく健康者が利用する際でも、室内が暗い場合リモコンの位置が分かりにくい、料理中で手がふさがっている場合は操作ができない、といった不便さがある。以上のような問題点から、リモコンやパネルといった操作のための特別なデバイスを利用する必要がないインタフェースが現在期待されている。

### 2.2 音声による機器操作

操作のための特別なデバイスを必要としない次世代の操作インタフェースの一つとして、音声操作インタフェースが注目されている。音声操作インタフェースとは、音声認識エンジンを利用してユーザの発話内容を認識し、認識結果と特定の機器の振る舞いを対応付けることが可能な操作インタフェースである。音声操作インタフェースの実用化の例として、カーナビゲーションシステム [5] や対話型音声情報案内システム [6] などがある。

また、音声を利用した家電操作インタフェースとして、ビデオを録画する操作を音声で行うもの [7] や頭の向きと音声を併用した四肢まひ者向けの家電機器操作インタフェース [8] などが提案されている。

### 2.3 音声操作インタフェースの HNS への適用可能性

既存の音声インタフェース (例えば [9] [10]) を用いて発話された内容を音声コマンドと解釈し、フルコマンドと対応付ける

ことで、HNS での音声操作は可能である。このとき、従来の音声インタフェースでは、ユーザの一度の発話がフルコマンドとして必要な全ての情報を含んでいる必要がある。そのため本稿では従来法をまとめてフルコマンド法と呼ぶ。

フルコマンド法は、既存技術を用いて容易に実装可能である反面、以下のような問題点が想定される。

P1：音声コマンドの増加による学習コスト増大 フルコマンド法では、ユーザは完全な音声コマンドを事前に知っている必要があるが、操作対象機器が増えると音声コマンドの種類・数も増えるため、学習コストが増大してしまう。

P2：認識失敗による意図しない動作の発生 従来のリモコンによる操作と比べると、音声インタフェースには誤認識が発生するため操作の正確性が低い。音声認識を失敗すると、制御される機器の振る舞いがユーザの意図しないものになり、特に HNS では危険な状況やユーザのストレスを引き起こす要因となる可能性がある。

### 3. 提案する HNS 操作インタフェース

#### 3.1 システム要求

2.3 で述べた問題点にもとづき、提案する音声インタフェースが満たすべき要求を以下に述べる。

R1： 機器数が増加しても音声コマンド学習コストを増やさない P1 は、これまでの HNS 操作インタフェース全般にとっての問題点でもある。特に音声インタフェースでは、音声コマンドに含まれるキーワード全てを記憶する必要があるため、学習コストがリモコン操作などよりも高くなる可能性がある。このような状況を避けるためには、機器の数が増えてもユーザが記憶・学習しなければならない音声コマンドの量が変化しないような仕組みが必要である。

R2： 誤認識発生時の信頼性向上と誤認識発生削減を実現する P2 で述べたとおり、音声コマンドの誤認識は避けられないものであるため、ユーザが意図していない機器制御が行われないような機構が必要である。ここで、ユーザ発話認識後の機器制御前に確認を行う機構は、システムの信頼性を向上させるための最もシンプルな解決策のひとつである。また、信頼性の向上だけでなく、HNS から得られる情報にもとづいて誤認識発生を削減するための仕組みを実現することは、家電機器操作時のユーザストレスの削減といった観点からも重要であると考えられる。

#### 3.2 キーアイデア

本稿では、これらのシステム要求に対応するために以下のキーアイデアを実現する。

K1： 対話による音声コマンドの段階的構築支援 ユーザに対して一度に全ての音声コマンドの発話を求めずに、システムとの対話を通して段階的に構築可能なインタフェースを作成する。我々が提案するインタフェースでは、ユーザは必ずしも一度にフルコマンドに必要な全ての情報を音声コマンドとして発話する必要は無い。

ユーザが機器名を発話すると、その機器に応じた操作が提示される。その後ユーザの選択（発話）した操作名に応じて、操作

が受け入れ可能なパラメータを順にシステムがガイドする。このように段階的にフルコマンドの構築を支援することで、ユーザが事前に知っていなければならない音声コマンドに関する知識を大幅に減らすことができる。

K2： 家電の状態に応じたコマンド提示・解釈 HNS 内に配置されるネット家電はそれぞれ状態を持ち、状態に応じた振る舞いを提供する。また、HNS ではすべての家電の状態をネットワーク越しに取得することができる。したがって、対象の家電が現状で実行可能な操作のみを、システムが提示・解釈することにより、ありえない組み合わせのフルコマンドを排除することが可能となる。結果として言い間違い等に伴う音声コマンドの誤認識や意図しない動作を削減することが可能となる。

#### 3.3 対話による音声コマンドの段階的構築支援

音声コマンドには、機器名、操作名、さらに必要に応じてパラメータ、の 2 つないし 3 つの情報が含まれている。テレビを例に挙げると、テレビのチャンネルを 4 にするには「テレビ、チャンネル、4」と発話しなければならない。また、カーテンの場合は「カーテン、開けて」と発話する必要があるが、こちらはパラメータに相当する情報を必要としない。

図 1 は、従来のフルコマンド法におけるユーザ発話から機器 API 実行に至るまでのフローチャートである。まず受け付けた音声コマンドに機器制御に必要な全ての情報が含まれているか、すなわちフルコマンドであるかどうかを判断する。音声コマンドがフルコマンドとして正しければ、その内容を確認後に機器 API を実行する。必要な情報が含まれていない、間違っている等の理由で正しくないと判断されれば、コマンド入力のやりなおしを受け付ける。

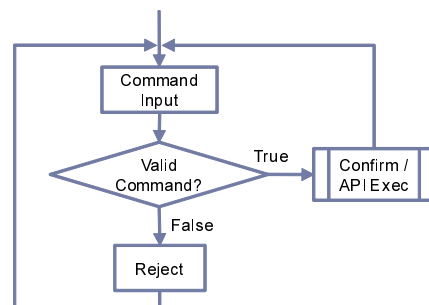


図 1 フルコマンド法操作フロー図

一方、提案する対話型インタフェースでは、音声コマンドが必ずしも最初から完全である必要はない。提案インタフェースは、事前に登録された音声コマンド DB を利用し、不完全な音声コマンドが入力された場合に何が不完全であるかをユーザにフィードバックする。ここで音声コマンド DB は、全ての機器のフルコマンドが登録されている DB であるものとする。ユーザはそのフィードバック内容を確認し、不完全部分の音声コマンドを追加で発話することで完全な音声コマンドを構築し、フルコマンドに対応付けることができる。

図 2 は提案インタフェースのフローチャートを表している。以下にフローチャートに対応した提案インタフェースの処理プロセスを示す。

STEP1: ユーザの発話した内容(発話 A)を音声コマンドとして解釈し、その発話 A で指定されたコマンドが音声コマンド DB に含まれているかどうかの確認を行う。ここで含まれているかどうかの判断は、発話 A が (1) 機器名・操作名・パラメータ、(2) 機器名・操作名、(3) 機器名、のどれかであるという想定において、その発話内容が音声コマンド DB 内の任意のコマンドの一部であるかどうかを検証することで行われる。含まれていない場合は再度ユーザに発話を促し、含まれている場合は STEP2 に移行する。

STEP2: STEP1 における有効性の確認後、音声コマンドの完全性(フルコマンドになっているかどうか)を評価する。発話 A で指定されたコマンドがフルコマンドである場合、STEP10 に移行する。フルコマンドでなければ、STEP3 に移行する。

STEP3: ここでは、発話 A が STEP1 の (2) 機器名・操作名、(3) 機器名、のどちらであるかを確認する。(2) すなわちパラメータが不足している場合は STEP8 に移行し、(3) すなわち操作名とパラメータが不足している場合は STEP4 に移行する。

STEP4: 発話 A に含まれる機器のもつ操作名の集合を音声コマンド DB より抽出後、その操作名を音声でユーザにフィードバックし、STEP5 へ移行する。

STEP5: STEP5 では、STEP4 でフィードバックした操作名にもとづいてユーザが追加の音声コマンド(発話 B)を発話するのを待つ。ここで発話される内容は、(4) 操作名、(5) 操作名・パラメータ、のどちらかであることを想定している。(4)(5) どちらの場合においても、操作名が発話 A に含まれる機器が保持するものであることを音声コマンド DB を用いて確認した後に、STEP6 に移行する。なお、図中の STEP4・5 に相当する部分は、内部に操作名フィードバック、操作名入力、操作名確認の 3 つの処理が存在するが、図では省略した。

STEP6: ここでは発話 A と B の組み合わせによって指定された機器名・操作名の組がパラメータを必要とするかどうかを音声コマンド DB にもとづいて確認する。パラメータを必要としない組み合わせの場合は STEP10 に移行し、パラメータを必要とする場合は STEP7 に移行する。

STEP7: 発話 B にパラメータが含まれているかどうかを判断する。含まれていた場合、STEP10 に移行し、含まれていなかった場合は STEP8 に移行する。

STEP8: この STEP に到達した時点で、機器名と操作名が指定されている。そのため、ここでは対象の機器名・操作名の持つパラメータの集合をユーザにフィードバックし、STEP9 に移行する。

STEP9: STEP8 でフィードバックされた内容にもとづいてユーザが追加の音声コマンド(発話 C)を発話するのを待つ。この STEP では、発話 C が発話 A もしくは発話 A、B によって指定された機器名・操作名に対応するパラメータであることを確認した後に、STEP10 へ移行する。なお、STEP4・5 と同様に、図中の STEP8・9 に相当する部分は、内部にパラメータフィードバック、パラメータ入力、パラメータ確認の 3 つの処理が存在するが、図では省略した。

STEP10: 発話 A、B、C の内容を組み合わせた後に、その

音声コマンドの内容をユーザにフィードバックする。その後ユーザからの返答が「はい」であれば、その音声コマンドに対応する機器 API の呼び出しを行い、「いいえ」であれば STEP1 に戻る。STEP10 では、内部に操作実行確認フィードバック、操作実行確認入力、操作実行判断、API 呼び出しの 4 つの処理が存在するが、図では省略した。

以上のプロセスが示すように、ユーザは機器名を記憶しておきさえすれば、提案インタフェースによって提示されるメソッド名、パラメータ名の集合のフィードバックにもとづき、音声コマンドを段階的に構築することができる。一方で、既に記憶している音声コマンド(フルコマンド)については、従来のインタフェースと同じプロセスで機器を制御することが可能であるため、 unnecessary プロセスをユーザに強いることは無い。

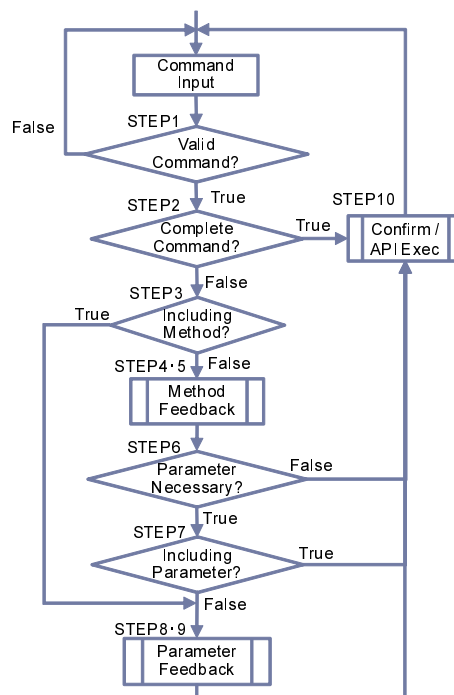


図 2 提案システム操作フロー図

### 3.4 家電の状態に応じたコマンド提示・解釈

HNS に配備される家電はそれぞれ内部状態を持っており、各状態ごとに実行可能な操作、およびパラメータが決まっている。例えば、テレビは電源の状態を持っている。電源がオフの状態では、音量を上げたりチャンネルを選択する操作は実行できず、電源をオンにする操作だけが実行可能である。また、一般的な自動カーテンであれば、「開」「閉」という二つの状態を持っている。「カーテンを開ける」という操作は、カーテンの状態が「閉」の時のみ実行可能である。同様に、「カーテンを閉める」は状態が「開」の時のみ有効である。

このように、各機器が元来持つ操作は複数存在するが、機器の現在状態に着目すれば、そのうち実行可能なものだけに絞り込むことができる。このことを利用すれば、ユーザからの音声コマンドをより高い精度で解釈・実行することが可能となる。さらに、状態に応じて実行可能な操作パラメータのみを 3.3 節で述べたフィードバック機構で提示すれば、ユーザの機能選択

を効率的に支援できる。また、実行可能な操作が一つしかない場合には、操作の選択プロセスを省略することも可能である。

以上のことを踏まえ、家電の状態に応じたコマンド提示・解釈を行う家電状態モニタ (Appliance Status Monitor) を実現する。家電状態モニタは、HNS に配備される各機器を状態遷移機械 (Finite State Machine, FSM) でモデル化して保持する。家電状態モニタは、操作する家電機器名を与えるとホームサーバに当該機器の現在状態を問い合わせる。次に、取得した現在状態とその機器の FSM モデルを照合し、現状態で実行可能な操作の集合 ( $E$  とする) を取得する。ユーザからの音声コマンドに対応する操作が  $E$  の中に存在しなければ、コマンドを棄却する。また、音声コマンドが不完全な場合、 $E$  を操作可能な候補としてユーザに提示する。また、 $|E| = 1$ 、すなわち、実行可能な操作が 1 つしか無い場合には選択の必要が無いので、ユーザには実行するかしないかの決定を「はい | いいえ」で決定させる確認プロセスに進む。これにより、ユーザのコマンド組み立ての手間を効率化できる。

音声を用いたインタフェースでは、その性質上区別しにくい音声コマンドを誤認識することが多い。例えば、「オン」と「オフ」、「中」と「9」など、発音が紛らわしいコマンドに関しては、システムが非常に高い確率で誤認識してしまう。この認識率は音声認識エンジンの性能に大きく依存し、従来のフルコマンド法ではそれ以上改善を期待できない。

一方、HNS に適用する目的においては、本節で提案した状態を用いたコマンド認識手法を導入することで、音声認識エンジンの性能に頼らない方法で、認識率を向上することができる。結果として、要求 R2 を効果的に達成することができる。

### 3.5 実装

提案した 2 つのキーアイデアにもとづいて、HNS 機器を操作するための音声インタフェースシステムを実装した。提案システムの実装環境を以下に示す。

音声認識エンジン 音声認識エンジンには Julius for SAPI 2.3 [3] を、Speech API には Microsoft SAPI Speech SDK 5.1 を用いた。Julius は認識すべき語彙や文法を自由に定義できることを特徴としている。本稿では、文法 29 種類、語彙 52 種類を登録した。

スピーチエンジン フィードバックを出力するためのスピーチエンジンには、PENTAX 社製の VoiceText Engine [11] を用いた。

実装言語 音声認識エンジンを利用するためのアプリケーションを Visual Studio 2005 上で C++ 言語で実装した。

## 4. 評価実験

### 4.1 実験概要

提案手法の有効性を評価するために、被験者に HNS に接続された家電機器を操作してもらう実験を行った。本実験では、提案手法である対話式のインタフェースと、2.3 で述べたフルコマンド法によるインタフェースの二種類の手法を用いて被験者にタスクを行ってもらった。家電機器を音声で操作するインタフェースはまだ普及しているとは言えず、一般に音声イン

タフェースを使用した経験が少ない、あるいはまったくない人が多数を占めると考えられる。そこで、本稿では音声インタフェースの使用経験がない、もしくは日常的に使用していない 12 名を被験者として実験を行った。

本実験では、実験のタスクとして提案手法及びフルコマンド法で共通の 11 項目のタスクを設定した。フルコマンド法を用いるタスクでは、全ての音声コマンドを示したリストを被験者に渡し記憶してもらう。提案手法を用いるタスクに関しては、機器名を記憶しなければならないが、タスクリストに機器名を明記しているため、コマンドを記憶する作業は発生しない。

### 4.2 実験環境

対象とする HNS として、現在運用中の NAIST-HNS [2] を用いた。操作対象とした機器は、テレビ、DVD プレーヤ、ライト、電動カーテン、空気清浄機、扇風機の 6 種類である。

音声の入力装置としては、Plantronics 社製ヘッドセット Voyager 510 を用いた。この機器は無線接続によるハンズフリーの操作が可能である。

### 4.3 実験手順

本実験では、提案手法を先に行う被験者と従来法を先に行う被験者を半分ずつの各 6 名として実験を行った。先に行う手法で操作を行うことで音声インタフェースによる操作に慣れてしまい、後に行う手法との間で音声操作に対する慣れの度合いが変わってしまう可能性がある。こういった学習効果による影響を少なくするため、各手法を行う順番を変えて実験を行うこととした。

以下に提案手法を先に行う場合の実験手順を示す。

1. 被験者に実験でどの HNS 機器が操作の対象となるか説明し、使用するマイクについても使い方の説明を行う。
2. 提案手法と従来法の各手法についての説明を行う。
3. 実験タスクのリストを被験者に渡し、順番通りにタスクを行ってもらよう説明する。
4. 提案手法を用いてタスクを行ってもらう。
5. 従来法で用いるコマンドの一覧を 3 分間で記憶してもらう。
6. 従来法を用いてタスクを行ってもらう。

従来法を先に行う場合の実験手順は、提案手法を先に行う場合の順番を変え、1. 2. 3. 5. 6. 4. の順番で実験を行ってもらう。

### 4.4 結果と考察

図 3 に実験結果を箱ひげ図で示す。3.1 で示した要求をシステムが充足しているか評価するために、従来法と提案手法それぞれについて、4 つのメトリクスを評価指標とした。以下に、提案手法が要求を満足しているかどうかを結果をもとに述べる。R1:学習コスト 各手法の学習コストを測る指標として、発話文字数 (M1) と音声コマンド確認回数 (M2) を計測した。M1 は、タスクを開始してから終了するまでにユーザが発話した総文字数を示している。M2 は、従来手法ではタスクを実行するための音声コマンドをユーザが忘れてしまいリストを再確認した回数を示し、提案手法ではシステムが提示した操作に対する音声コマンドをユーザが正しく発話できず確認を求めた回数を示す。



M1 が示す数値はユーザの発話量を示し、値が大きいほど学習コストが高いと言える。また、M2 が示す数値は、従来法ではコマンドのリストを記憶するための学習コストを、提案手法ではフィードバックからコマンドを獲得する際の学習コストをそれぞれ示し、値が大きいほど学習コストが高いと言える。

図 3 上より、M1 では平均値・中央値とも提案手法の方が低い値となった。M2 では、提案手法の方は最大でも 1 回で中央値も 0 だが、従来法では 2 回確認した被験者もあり、中央値も若干高くなっている。これらの結果から、提案手法の方が学習コストが低いと言えるため、要求 R1 を満たしていると言える。

**R2: システムの信頼性** システムの信頼性を測る指標としては、発話失敗回数 (M3) と機器動作間違い回数 (M4) を用いる。M3 は、ユーザの発話がそもそもタスクを実現するための音声コマンドと異なっていた回数を示す。M4 は発話失敗または音声認識の失敗によって音声コマンドがタスクと異なっていた回数を示し、本実験では音声コマンド最終確認時においてユーザが「いいえ」と発話した回数をカウントしている。

図 3 下より、M3 では両手法ともばらつきが大きく回数が多い被験者もいるが、平均値・中央値はさほど高くなくほぼ等しい値となった。M4 では平均値、中央値ともに提案手法の方が低く、最大でも 3 回だが、従来法では最大で 12 回機器動作内容がタスクと異なるケースが現れた。このように、ユーザがそもそも発話を間違えた回数 (M3) が同程度であるにも関わらず、動作間違い回数 (M4) が提案手法のほうが少なくなっている。音声エンジンの誤認識に関しては、どちらの手法でも同じ音声認識エンジンを使っているため、両手法間に差が無いものといえる。結果として、提案手法は従来法に比べ発話失敗や誤認識に対して誤動作が起こる回数が少なく、要求 R2 として挙げたユーザの意図しない操作を削減する仕組みが実現されていると判断できる。

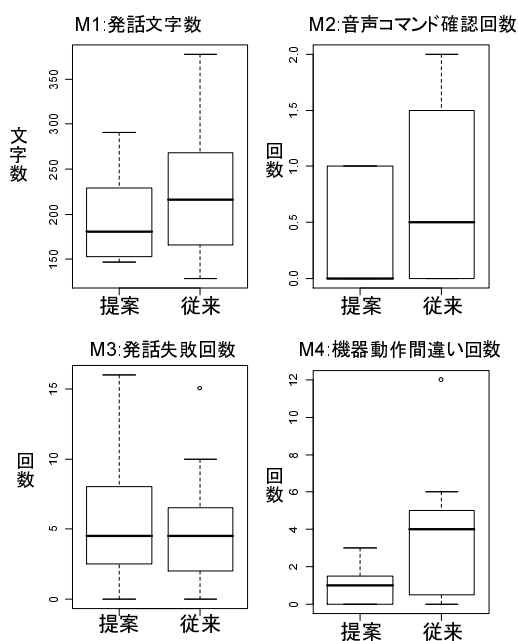


図 3 実験結果

## 5. ま と め

本稿では、2.3 において、現状の音声による HNS 操作インタフェースの問題点として、P1:操作対象機器が増えると音声コマンドの数も増える、P2:音声認識を失敗すると制御される機器の振る舞いがユーザの意図しないものになる可能性がある、の 2 つを挙げ、3.1 でこれらの問題点を解決するための R1, R2 の 2 つの要求を明確にした。これら 2 つの要求を満たすため 3.2 において K1, K2 の 2 つのキーアイデアを提案し、3.5 でキーアイデアを実現する段階的な音声コマンド構築のための対話型インタフェースを実装した。また、4. では提案するインタフェースを評価するための実験として、従来技術を利用した音声操作インタフェースであるフルコマンド法と提案手法とを比較する実験を行い、得られた結果の考察を行った。

今後の研究課題として、さらなる誤動作の低減やより自然な会話に近い対話インタフェースの構築、雑音環境下におけるロバスト性などが挙げられる。家電機器を操作する音声インタフェースは未だ一般的なものとなっていないが、これらの課題を解決することで広く一般に利用されるシステムにより近づくことができると考えられる。

### 謝 辞

この研究は、科学技術研究費 (若手研究 B 18700062, 若手研究 (スタートアップ) 18800060, 基盤研究 B 17300007), および、日本学術振興会日仏交流促進事業 (SAKURA プログラム) の助成を受けて行われている。

### 文 献

- [1] 森川 博之, “ワイヤレスが開くコピキタスネットワーク,” 電子情報通信学会誌 Vol.87, No.5, pp.356-361, 2004.
- [2] Masahide Nakamura, Akihiro Tanaka, Hiroshi Igaki, Haruaki Tamada, and Ken-ichi Matsumoto, “Constructing Home Network Systems and Integrated Services Using Legacy Home Appliances and Web Services,” International Journal of Web Services Research, Vol.5, No.1, pp.82-98, 2008.
- [3] 河原 達也, 李 晃伸, “連続音声認識ソフトウェア Julius,” 人工知能学会誌, Vol.20, No.1, pp.41-49, 2005.
- [4] 共用品推進機構, “不便さ調査データベース,” [http://www.kyoyohin.org/02\\_syougai/index.php](http://www.kyoyohin.org/02_syougai/index.php)
- [5] Pioneer “CYBER NAVI,” <http://pioneer.jp/carrozzeria/cybernavi/function/design/sound.html>
- [6] 鹿野 清宏, 川波 弘道, 西村 竜一, 李 晃伸, “音声情報案内システム「たけまるくん」および「キタちゃん」の開発,” 情報処理学会研究報告, SLP, 音声言語情報処理, Vol.2006, No.107, pp.33-38, 2006.
- [7] 渡辺 裕太, 関口 芳廣, 鈴木 良弥, “ビデオ装置を例とした家電品の音声対話機能について,” 情報処理学会論文誌, Vol.44, No.11, pp.2690-2698, 2003.
- [8] 伊藤 英一, “頭の向きと音声により操作する四肢まひ者向け家電コントローラ,” ヒューマンインタフェース学会論文誌, Vol.4, No.1, pp.25-31, 2002.
- [9] Innotec Systems Inc. “Accenda,” <http://www.innotechsystems.com/accenda/index.html>
- [10] 松政 宏典, 田中 克幸, 滝口 哲也, 有木 康雄, 李 義昭, 中林 稔 義, “情報家電操作における脳性麻痺構音障害者の音声認識評価,” 電子情報通信学会技術研究報告, WIT, 福祉情報工学, Vol.107, No.61, pp.33-38, 2007.
- [11] PENTAX, “VoiceText Engine,” <http://voice.pentax.co.jp/>