音声コマンドを用いた3層アーキテクチャによる遠隔操作システム

後藤和弘・佐藤辰雄・大城英裕*・吉岡孝**・築根秀男***
大分県・産業技術総合研究所研究交流センター・*大分大学・**大分県立芸術文化短期大学・***産業技術総合研究所

Remote Control System with Three-Tier Architecture Using Voice Command

Kazuhiro GOTO, Tatsuo SATO, Hidehiro OHKI*, Takashi YOSHIOKA** and Hideo TSUKUNE***
Oita-AIST Joint Research Center, *Oita University, **Oita Prefectual College of Arts and Culture, ***AIST

要旨

音声コマンドを用いた3層アーテクチャによる移動ロボットシステムを開発した．クライアントにおいてマイクから入力された音声コマンドを，ネットワーク経由でアプリケーションサーバへ転送して音声認識を行うことにより，クライアント端末の性能や種別に依存することなく，ネットワーク上のどこからでも，音声コマンドによって移動ロボットの遠隔操作が可能となる．本システムでは，操作対象をロボット以外に置き換える場合やアプリケーションサーバ等のハードウェアを更新した場合，また，音声認識エンジン等のソフトウェアを更新した場合でも，クライアント端末のソフトウェアを変更することなく，ユーザがシステムを利用できる．実験によって開発したシステムを評価するとともに，操作対象を容易に拡張できることを示す．

1．はじめに

オフィスや家庭において人を支援するロボットを想定し，ユーザへの負担が少なくロボットと自然なコミュニケーションを行うためにジェスチャや音声によるマンマシンインタフェースが研究されている[1]~[5]．ロボットと人が直接対話するシステムでは，ロボットにマイクと音声認識機能を実装する必要がある[2][3]．また，ネットワーク経由で音声による遠隔操作を行うには，マイクと音声認識機能を持たせたクライアントで認識を行い，その結果をもとにサーバへ文字列等の制御指令を送信する手法が提案されている[4][5]．音声認識には認識エンジンのソフトウェアや認識処理で参照する語彙データを記憶するための大容量のディスク資源や認識処理のための高性能なCPU資源などが要求される．また，ユーザインタフェースを音声のみで実現する目的で動作要求への応答や各種情報の通知などを音声で出力するには，音声ファイルや音声合成エンジンが利用できるものの，音声認識エンジンと同様にディスク資源やCPU資源が必要となる．しかし，音声認識エンジンや音声合成エンジンが特定のプラットフォーム向けに開発されている場合には，クライアント端末の種別やOS等が制限されてしまう．また，ネットワーク経由での音声による遠隔操作システムを構築するために，すべてのクライアントへ音声認識エンジンや音声合成エンジンをインストールすることは困難である．

そこで本研究では，クライアントの性能や種別等に大きく依存することなく，またネットワーク上のどこからでも音声コマンドを用いて移動ロボットやパンチルトカメラを遠隔操作することを目標とし，3層アーキテクチャによる遠隔操作システムを提案する．

2．遠隔操作システム

2.1 システムの概要

遠隔操作システムの概要をFig.1に示す．マイクを接続したクライアント，アプリケーションサーバ，パンチルトカメラを搭載した移動ロボットをネットワークに接続して，システムを構成する．クライアントは，マイクから入力された音声コマンドをアプリケーションサーバへ送信する．アプリケーションサーバは受信した音声コマンドをもとに音声認識を行い，対応する制御コマンドをロボットへ送信する．ロボットは受信した制御コマンドにもとづいて，移動やカメラのパンチルト制御を行う．

ネットワーク上でシステムの機能を分散する手法に3層アーキテクチャがある．3層アーキテクチャは一般的にプレゼンテーション層，アプリケーション層，データ層の3つで構成され，拡張性や柔軟性，保守性に優れるという特長をもつことから，ネットワーク環境におけるデータベースシステムなどに採用されている．プレゼンテーション層では主に情報の入出力を行い，画面の表示機能やマウス等による入力機能をもつ．アプリケーション層はクライアントでの表示内容の加工やデータベースへのアクセスなど，システムの中で主要な機能を実行する．

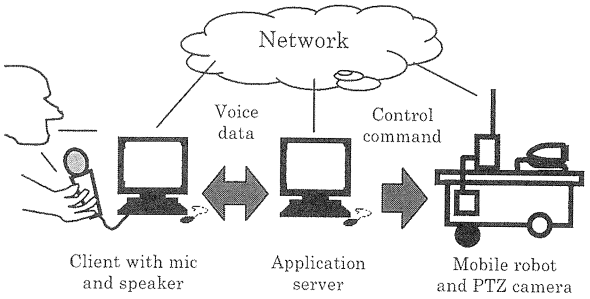

Fig.1 Overview of the system

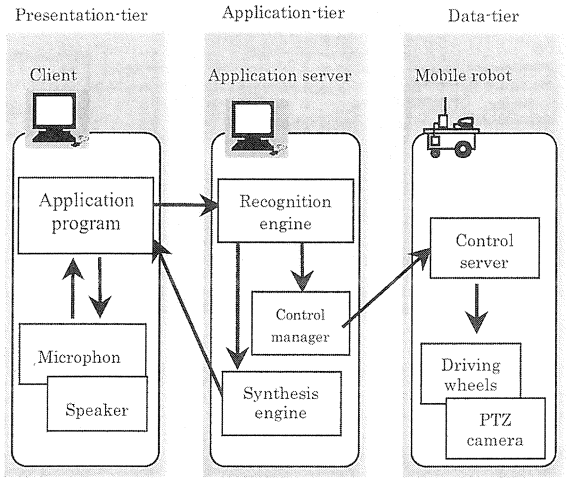

Fig.2 System configuration and data flow

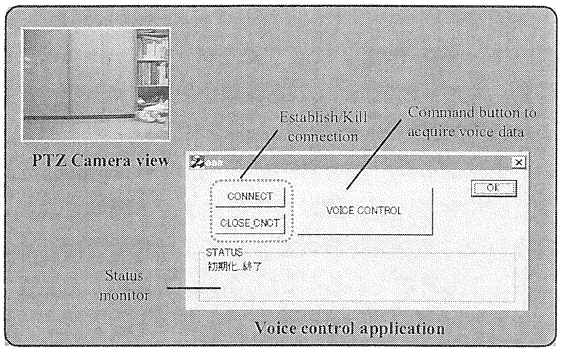

Fig.3 User interface on a client

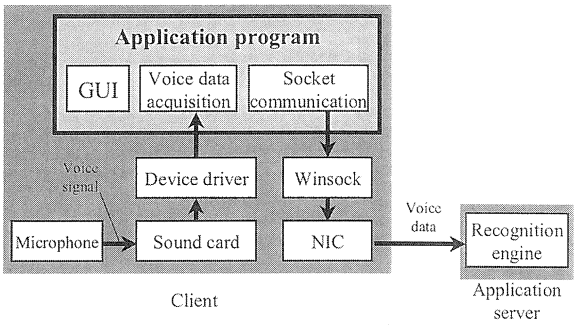

Fig.4 System configuration on a client

データ層ではデータベースシステムなどによりデータを管理する．このアーキテクチャをロボットシステムに適用することで，クライアント端末の性能に大きく依存することのない遠隔操作システムを構築できる[6]．本システムでは，Fig.2 に示すようにプレゼンテーション層にクライアント，アプリケーション層にアプリケーションサーバ，そしてデータ層には移動ロボットをそれぞれ配置する．Fig.2 の矢印は処理の流れを表し，以下の手順で遠隔操作を行う．

(1)クライアントはマイクから入力された音声をサンプリングし，アプリケーションサーバへ送信する．

(2)アプリケーションサーバには，音声認識エンジン，音声合成エンジン，コントロールマネージャの 3 つのアプリケーションプログラムを実装し，受信した音声コマンドを認識して，対応する制御コマンドをコントロールマネージャへ送信する．また，応答内容を音声合成エンジンで生成し，クライアントへ送信する．コントロールマネージャは移動ロボットの操作状況を管理し，ロボットへの接続要求を受信した際に他のユーザが操作中でなければ，ロボットの操作を許可する．

(3) 移動ロボットでは，内蔵コンピュータでコントロールサーバを実行し，アプリケーションサーバからの制御要求を受信すると車輪用モータ，またはカメラのパンチルト用モータを駆動する．

2.2 クライアント

クライアントの主な機能はユーザインタフェースである．データベースシステムでは GUI に関する処理が中心であり，マウスやキーボードの操作情報をサーバへ送信し，処理に対応する画面情報をサーバから受信して表示内容を更新する以外には複雑な計算や処理などは行わない．本システムでは Fig.3 に示すように，ユーザがカメラの映像を見ながら音声コマンドでロボットを操作することを想定し，音声をユーザインタフェースとするために音声の入出力機能とソケットによる通信機能を実装する．なお，カメラ映像については本報告では説明を省略する．ユーザは遠隔操作用アプリケーションのコマンドボタンを押しながら，マイクを通じてロボットへ動作指令を与えることができる．Fig.4 にクライアントの構成を示す．入力された音声コマンドをサウンドカードでサンプリングし，デジタル化された音声データをソケット通信によりアプリケーションサーバ上の認識エンジンへと送信する．また，操作要求に対する応答をアプリケーションサーバから受信した場合には，スピーカで応答内容を再生する．このように複雑な処理や計算を行わないので，PDA などの軽量端末もクライアントとして利用でき，Linux や Macintosh など OS にも依存しない．

2.3 アプリケーションサーバ

2.3.1 音声認識処理

音声認識エンジンには，C 言語のソースがフリーで公開されている大語彙連続音声認識デコーダ Julius を使用する[7]．不特定話者に対応し，単語認識精度は実験室環境で 85～95%程度である．音声の入力にはマイクや音声ファイルのほかに，ネットワーク経由で音声データを受信することもできる．本研究では，あらかじめ登録したロボットの制御コマンドと音声認識結果を比較できるよう

に Julius を改造して使用した. Fig.5 に処理フローを示す. 制御コマンドは Table 1 に示すように，ロボットとの接続や移動，カメラの制御など 10 種類を用意した．認識結果がこれらのいずれかと一致した場合に，対応する制御コマンドをコントロールマネージャへ送信する. Julius とコントロールマネージャ間の通信はソケット通信で実現しているので，これらを異なるコンピュータに分散して実行することも可能である．また，Julius を起動したままで異なるクライアントからの要求を処理できるように，ソケット通信の接続に関する処理を改良した．

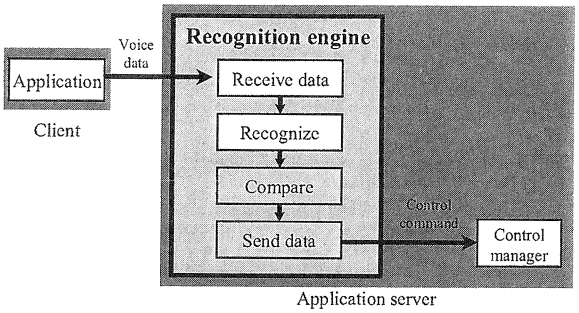

Fig.5 Recognition engine on an application sever

Table1 Registered commands for recognition engine

Classification	Control commands
Connection with a robot	接続，切断
Control of a robot	前進，停止，右，左
Control of a PTZ camera	カメラ上，カメラ下，カメラ右，カメラ左，カメラ正面

2.3.2 音声合成処理

クライアントからの操作要求に対して音声で応答するために，市販の Linux 版日本語音声合成ライブラリを使用してアプリケーションサーバに音声合成機能を組み込む．操作要求に対応させたテキストにもとづいて生成した音声データを，ソケット通信によってクライアントへ送信する．言語辞書や波形辞書をアプリケーションサーバ上にもち，PCM 形式のデータを生成することで，クライアント端末は性能や種別に依存することなく応答内容を再生できる．

2.3.3 コントロールマネージャ

コントロールマネージャは Fig.6 に示すようにメインプロセスと複数の子プロセスで構成され，複数のユーザが同時にロボットを操作できないようにロボットの操作権を管理する．メインプロセスはクライアントからの操作要求を受けつけるとロボットの状態を管理テーブルで確認し，操作が可能であればロボット上で実行しているコントロールサーバへ制御指令を送信する．しかし，既

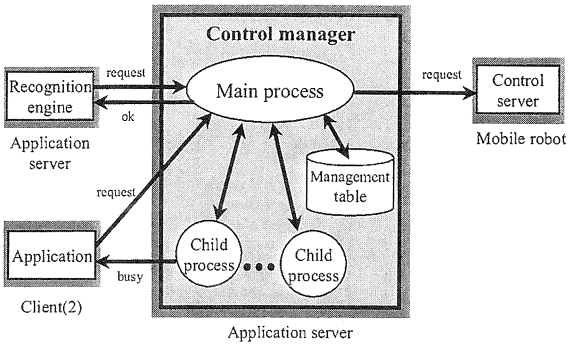

Fig.6 Control manager on an application server

に他のユーザが操作している場合には，子プロセスを通じてロボットの操作を禁止する．

2.4 移動ロボット

移動ロボットには２つの駆動輪と１つの補助輪があり，前進，後退，左右への方向転換が可能である．また，搭載したカメラのパン・チルト・ズーム機能は，シリアルケーブルを通じて制御できる．ロボットに内蔵したコンピュータ上で，モータへ駆動指令を与えるためのコントロールサーバを実行する．コントロールサーバは，ネットワーク経由でコントロールマネージャからロボットの操作要求を受信すると，車輪や PTZ カメラのモータを駆動する．

3．実験

3.1 実験方法

Fig.7 に示すように実験システムを構成し，音声コマンドによる遠隔操作実験を行った．クライアントとアプリケーションサーバはそれぞれ UTP5 ケーブルで 100Mbps 対応のスイッチングハブへ接続した．移動ロボットは 11Mbps の無線 LAN でネットワークへ接続した．マイクから音声コマンドを入力し，クライアントにおける音声コマンドの発声時間，サンプリングされた音声データのサイズを測定する．アプリケーションサーバでは，音声認識の処理時間，および CPU 使用率を測定する．また，クライアントからアプリケーションサーバへの音声デー

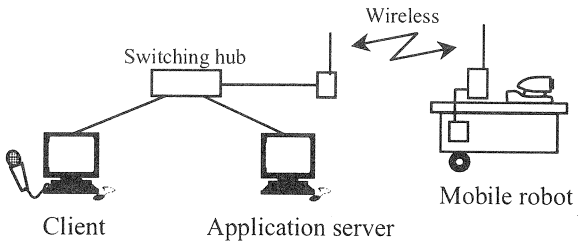

Fig.7 Experimental environment

タの転送時間を測定する．実験に使用したコンピュータの仕様を Table2 に示す．

Table2 Specification of the client and the application server used for the experiments

	Client	Application Server
CPU	Pentium2 350MHz	Pentium3 700MHz
Memory (MB)	128	128
OS	Windows98	Linux2.2.12

3.2 実験結果

Table3 は 5 つの音声コマンドの測定結果で，それぞれについて，発声時間，音声データのサイズ，音声認識の処理時間を 20 回ずつ測定し，その平均値を表している．発声時間が長いほど音声データのサイズは大きくなり，これに伴って認識処理時間も長くなることが分かる．

Table3 Measured results.

Voice Command	Duration of command [sec.]	Data size [kbytes]	Recognition time [sec.]
前進	1.14	34.7	1.7
左	0.94	28.6	1.3
右	0.85	25.7	1.2
カメラ左	1.58	48.5	2.2
カメラ右	1.71	52.9	2.5

Fig.8 は 5 つの音声コマンドについて音声データの転送時間を 20 回ずつ測定した結果を表していて，横軸は音声データのサイズ，縦軸は転送時間である．結果から，同一の音声コマンドでは，転送時間のばらつきが小さいことがわかる．また，5 つの音声コマンドの転送時間は 30~50msec の範囲にあり，音声データのサイズによる大きな差は見られなかった．

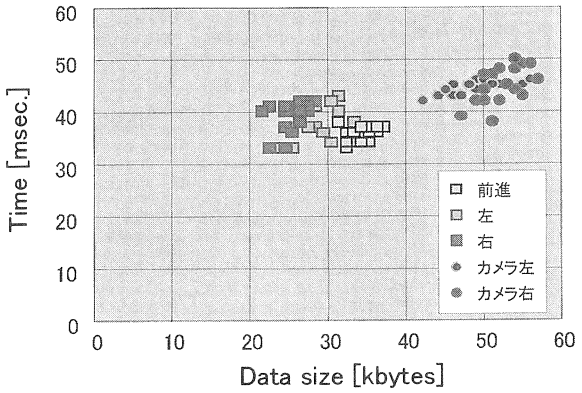

Fig.8 Recognition time and data size of voice data.

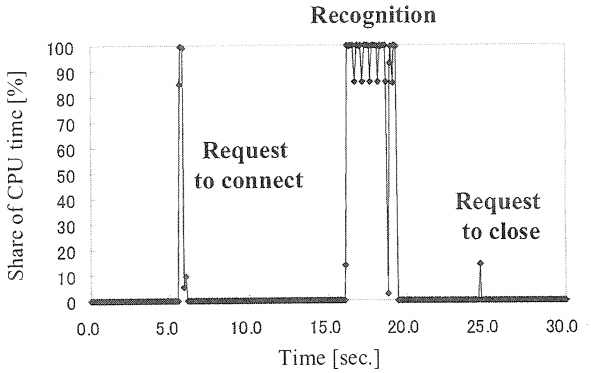

Fig9 Share of CPU time on the application server.

アプリケーションサーバにおける CPU 使用率の測定結果を Fig.9 に示す．クライアントからの接続要求時，音声認識処理時，クライアントとの切断時には CPU 使用率が高く，特に音声認識処理では数秒間にわたって 100%近く CPU を占有していた．

これらの結果から，音声データの転送時間は，音声コマンドの発声時間や音声認識の処理時間よりも十分に短く，システム全体としての処理時間に大きな影響を与えない．また，音声認識のような負荷の大きい処理をアプリケーションサーバでを行うことは，クライアント端末における処理の負荷軽減になる．音声認識の処理時間は，語彙を限定した辞書データの使用や，認識エンジンの動作パラメータを調節することなどによってさらに短縮できると考えられる．

4．システムの拡張

音声コマンドによってテレビやビデオデッキを遠隔操作できるようにシステムを拡張し，センターフェア 2001 で紹介した．システム構成の概要を Fig.10 に示す．テレビやビデオデッキの操作には市販のマルチリモコンを使用し，組込み用 Linux である uClinux を搭載した小型モジ

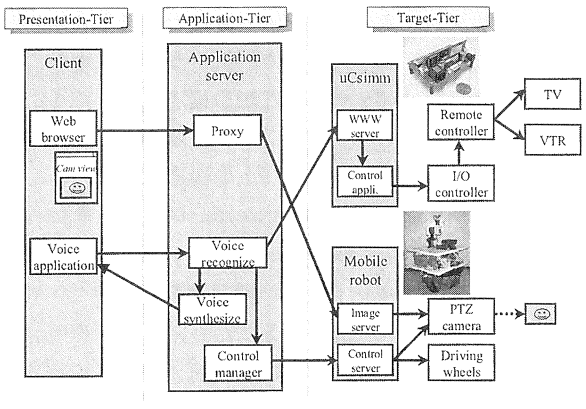

Fig.10 System configuration

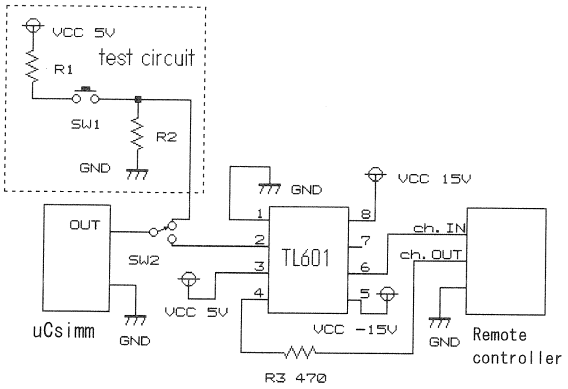

Fig.11　I/O controller.

ュール（uCsimm）でリモコンのボタンを制御する．uClinuxはC言語によるユーザプログラムを開発可能であり，標準で搭載されているWWWサーバを改良して簡易CGI機能を開発した．これにより，WWWサーバからuCsimmの汎用出力ポートの信号を制御し，I/Oコントローラを介してリモコンを操作できる．Fig.11にI/Oコントローラの回路図を示す．また，アプリケーションサーバにおいて，テレビ等の操作に関する音声コマンドを受信した場合にはuClinux上のWWWサーバへHTTP要求を送信するようにJuliusを改造した．このように，本システムでは機能を分散することで拡張が容易であり，クライアント側のプログラムを変更する必要はない．

5．まとめ

本研究では，音声コマンドを用いた3層アーキテクチャによる遠隔操作システムを開発し，音声認識の処理時間や音声データのサイズ，転送時間，CPUの使用率等をもとにシステムを評価した．そして，アプリケーションサーバで音声認識を行うことでクライアントの負荷を軽減しつつ，遠隔操作が可能であることを確認した．さらに，本システムに新たな操作対象を追加する場合でもクライアント上のプログラムの修正や再インストールなどは不要であり，システムの拡張が容易であることを示した．本システムでは，音声入力，ソケット通信などの機能があれば，コンピュータの性能や種別に依存することなく，PDAなどの軽量クライアントも端末として利用できる．アプリケーションサーバ上の機能を複数のコンピュータによって処理を分散させれば，システム全体のパフォーマンス向上も期待できる

今後はインターネットなどの利用環境を想定し，Webブラウザからの利用を検討するとともに，音声等のマルチメディア情報を転送するための標準プロトコルへの対応や，具体的な応用等について検討していきたいと考えている．

参考文献

[1]村嶋照久，久野義徳，島田伸敬，白井良明：人間と機械のインタラクションを通じたジェスチャの理解と学習，日本ロボット学会誌，18，4，pp.590-599, 2000.

[2]山崎信行，安西祐一郎：パーソナルロボットのためのアクティブインタフェースの設計と実装，日本ロボット学会誌，14，3，pp.461-469，1996.

[3] 松井俊浩，麻生英樹，John Fry，浅野太，本村陽一，原功，栗田多喜夫，速見悟，山崎信行：オフィス移動ロボットJijo-2の音声対話システム，日本ロボット学会誌，18，2，pp.300-307, 2000.

[4] 高強，西原主計，吉留忠史，河原崎徳之：福祉用Java-Linux音声コントローラ，日本機械学会ロボティクス・メカトロニクス講演会’01講演論文集，1A1-J5, 2001.

[5] 水川真，神名篤史，松原安彦，安藤吉伸：物理エージェント（PAS）における，音声操作系の基本検討，日本機械学会ロボティクス・メカトロニクス講演会’01講演論文集，2P2-K9, 2001.

[6] Kazuhiro GOTO, Tatsuo SATO, Kazuhiro TSURUOKA and Hideo TSUKUNE: Three-Tier Architecture for Remote Control of a Mobile Robot, Proceedings of International Symposium on Robot, pp.159-164, 2001.

[7] http://winnie.kuis.kyoto-u.ac.jp/pub/julius/