社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

# rospeex：クラウド型音声コミュニケーションを実現するROS向けツールキット

杉浦　孔明†　　堀　智織†　　是津　耕司†

† 情報通信研究機構
〒 619-0289 京都府相楽郡精華町光台 3-5
E-mail: †{komei.sugiura,chiori.hori,zettsu}@nict.go.jp

**あらまし**　音声コミュニケーション可能なロボットにおいては，ロボット上の計算機によって音声処理が行われることが多い．しかしながら，そのようなスタンドアロン型の機構では，高精度な音声認識や高品質な音声合成が難しい．そこで本研究では，多言語音声インタラクションを可能とするクラウド型ツールキット rospeex を提案する．rospeex により，多言語の音声認識と非モノローグ音声合成が ROS（Robot Operating System）上で可能になる．
**キーワード**　クラウドロボティクス，音声認識，音声合成，ROS

# rospeex: A Cloud-based Spoken Language Communication Toolkit for ROS

Komei SUGIURA†, Chiori HORI†, and Koji ZETTSU†

† National Institute of Information and Communications Technology
3-5 Hikaridai, Seika, Soraku, Kyoto 619-0289
E-mail: †{komei.sugiura,chiori.hori,zettsu}@nict.go.jp

**Abstract**　Stand-alone speech recognition/synthesis engines deployed on robots have drawbacks in accuracy and quality compared with cloud-based speech engines. In this paper, we propose “rospeex”, a novel cloud-based speech communication toolkit for ROS (Robot Operating System). It provides robots with high-performance speech recognition/synthesis capabilities with easy-to-use APIs. We have released the cloud-based service, so that researchers can use it without any cost/authentication.
**Key words**　cloud robotics, speech recognition, speech synthesis, ROS

## 1. はじめに

スマートフォンを始めとする種々のデバイスに音声インタフェースが導入され，広く一般に認知されるようになってきた [1,2]．音声対話システムの分野では，開発者が容易に利用できるツールキットも公開されている（例えば [3]）．一方，人とロボットのインタラクションでは，高性能な音声認識・合成を容易に利用できる状況ではない．ロボットとの高度な音声インタラクションを可能とするためには，音声処理とロボティクスの深い知識を要求されるのが現状である．

本研究では，家庭やオフィス内で「テーブルの上のペットボトルを取って」などの音声対話を行うサービスロボット開発を想定する．代表的なタスクとしては，ロボカップ@ホーム [4] タスクが挙げられる．このようなタスクでは，開発コストを低減できることから，RT ミドルウェアや ROS（Robot Operating System）などのミドルウェアの利用が一般的になってきている．ミドルウェアに対応するソフトウェアとして，音声認識・対話・合成を可能にするモジュールが提供されているものの，スタンドアロン型を前提とするものが多い [5,6]．しかしながら，ロボットに搭載できるコンピュータにはストレージやメモリの制限があるため，高性能な音声認識・合成は難しい．一方，クラウド型の高性能な音声認識・合成 API を提供する企業もあるが，ロボット開発者を想定したものでない場合が多い．

そこで本論文では，ROS 上で利用可能なクラウド型音声コミュニケーションツールキット “rospeex” を提案する．音声認識および音声合成機能をクラウド化することで，音響モデルや言語モデルなどの大規模な資源をロボット上に搭載する必要がなくなり，ハードウェアを簡略化することでコストを低減できる．クラウド型音声認識・合成サービスを通じて，NICT で開発されたエンジン(注1) をユーザは利用可能である．また，他のクラウド型音声認識・合成サービスに切り替えて利用すること

（注1）：rospeex 以外では，音声翻訳サービスとして既に 1000 万発話を処理している [5].

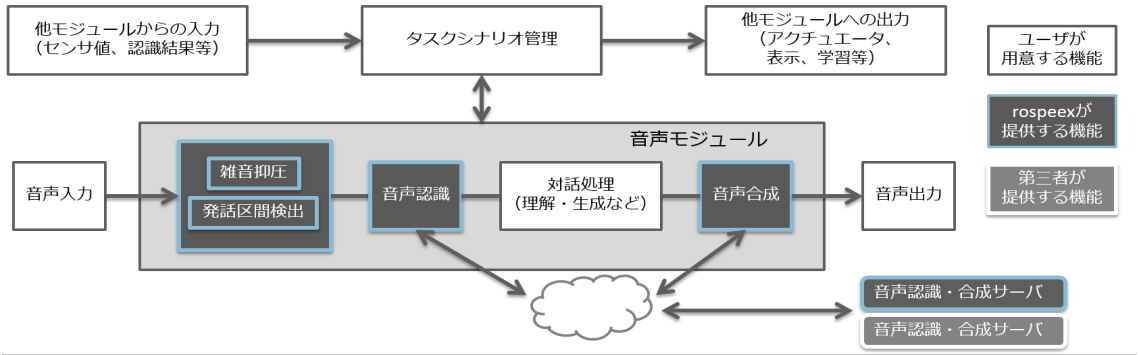


図 1　rospeex の構成の概略

も可能である．

クラウドロボティクスやクラウドネットワークロボティクスなどの分野では，物体認識，知識共有，機械学習などのためにクラウドコンピューティングを用いるアプローチが提案されている [7–9]．本研究はこれらと関連するが，ロボットの音声コミュニケーションに主眼を置く点が異なる．また，HARK [6] や OpenHRI [5] などミドルウェアに対応した音声コミュニケーションツールでは，内部的に Julius [10]，Festival [11]，Open-JTalk [12] などスタンドアロン型のエンジンを用いている．これらのエンジンは機能的には複数言語の音声認識・合成が可能であるが，言語モデルの入れ替えなどをロボット開発者自身が行う必要がある．一方，提案手法では，次節で説明するように言語やボイスフォントの変更を簡単に行うことができる．

提案手法の独自性は以下の 2 点である．

- ROS に対応したクラウド型音声認識・合成ツールキットを構築する．
- 音声合成として非モノローグ音声合成 [13] を用いる．

## 2.　rospeex の機能

rospeex が提供する機能と想定する標準的な構成を図 1 に示す．ROS のディストリビューションとしては，Groovy Galapagos を想定する．発話理解（言語理解），対話制御，応答生成については，ユーザが記述するものとする．

### 2.1　雑音抑圧および発話区間検出

ロボットとの音声対話において特徴的な難しさは，push-to-talk 方式を前提とできないことである．スマートフォン型音声インタフェースなどでは，音声の入力前にボタン等のインタフェース（「OK，XXX」のように音声の場合もある）を用いることが多い．これに対し，例外はあるものの，ロボットではそのようなインタフェースを前提とすることはできない．したがって，発話区間検出（Voice Activity Detection; VAD）が重要になる．さらに，マイクと話者の距離が大きいことが多いため，雑音抑圧が必要である．ロボカップ@ホームのような高騒音環境では，雑音抑圧を使用しなければ正確な発話区間検出はほぼ不可能である．これがヘッドセットを前提としたロボット以外のシステムとの大きな違いである．以下，本論文では，ツールキットの使用者を「ユーザ」，ロボットとの対話者を「話者」と略す．

rospeex では，雑音抑圧と発話区間検出はネットワーク上サーバで行わない設計としている．これらをサーバで処理するとネットワーク由来の遅延によりリアルタイム性の確保が難しくなるためである．また，一般的に発話区間検出の精度はそれほど高くないため，後段の処理でロボット名を含む発話のみ受け付けるなどの工夫が必要である．

図 2 に rospeex の音声波形インタフェースを示す．図中の青で示す部分は発話区間として検出された部分である．このように話者に波形をリアルタイムにフィードバックすることで，声が小さいなどの問題を話者自身が気づくことができる．一方，波形表示が提示されない場合，ロボットの反応から誤りの原因（発話区間検出の失敗か音声認識の誤認識か，など）を推定することは難しい．図のインタフェースでは，必要ない場合に発話区間が検出されないよう，発話区間検出機能を無効にすることも可能である．

### 2.2　クラウド型音声認識・合成

rospeex は複数のクラウド型音声サービスに接続可能であり，それらを切り替えて使用できる．本節では，NICT が提供する音声認識・合成サービスについて説明する．これらは，ROS を経由せずに単体としても利用可能である．現時点では，学術研究目的に限り無償・登録不要で公開している．本サービスでは，JSON ファイルをインタフェースとする．ユーザが用いるプログラミング言語には依存しないため，C++や Python など各種のプログラミング言語を利用可能である(注2)．

音声認識サービスを用いるために，サーバに送る命令（JSON ファイル）の例を図 3 に示す．サーバの基準 URL は，`http://rospeex.ucri.jgn-x.jp/nauth_`

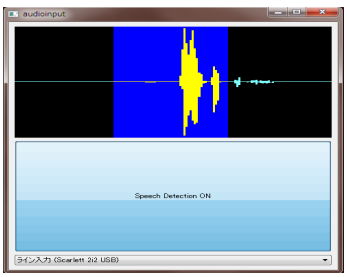


図 2　リアルタイム波形表示インタフェース

(注2)：サンプルコードを `http://komeisugiura.jp/software/nm_tts.html` から入手可能である．

json/jsServices/XXX である．ただし，URL の「XXX」部分はサービスごとに異なり，音声認識では「VoiceTraSR」，音声合成では「VoiceTraSS」である．サーバは認識結果を JSON 形式のテキストとして返送する．同様に，合成サービスでは図 4 のような JSON ファイル（図 4 参照）をサーバに送る．合成された wav ファイルは Base64 エンコードされ，JSON ファイルとして返送される．

```
{ "method":"recognize",
  "params":[
   "ja",
   { "audio":"Base64 エンコードされた wav 音声",
     "audioType":"audio/x-wav",
     "voiceType":"*"
   } ] }
```

図 3　音声認識命令

```
{ "method" : "speak",
  "params" : [
     "ja",
     "UTF-8 エンコードされた合成対象文字列",
     "*",
     "audio/x-wav"
  ] }
```

図 4　音声合成命令

### 2.3　ロボット開発者向けの API

rospeex の対象とするユーザはロボット開発者であるため，音声認識・合成を簡単に使用できることが望ましい．前述したように，NICT のサーバではインタフェースに JSON を用いている．rospeex では，開発者が 1 行程度の記述で音声認識・合成を利用可能な API を用意している．Python や C++であれば，10 行程度で簡単な対話（時刻の問い合わせなど）を行う関数を記述することができる．

対話管理にマークアップ言語（VoiceXML など）を利用するソフトウェアと異なり，rospeex は対話管理の簡単なインタフェースを用意していない．これは，想定ユーザとして，複雑な対話管理を必要としないロボット開発者を念頭に置いたためである．一方，ROS 上で Python や C++で開発したソフトウェア資産があれば，rospeex と簡単に組み合わせることが可能であるという利点がある．また，現状では，音源定位などの音響処理は統合されていない．しかしながら，HARK [6] など音響処理を扱うモジュールが提供されているので，rospeex の前段に容易に組み込むことが可能であると考えられる．

### 2.4　非モノローグ音声合成

ロボットのコミュニケーション機能の開発においては自然な音声合成が求められているが，一般的な音声合成器は人-ロボット対話に最適化されている訳ではない．そのため，ロボットの合成音声は自然さや親しみやすさに欠け，抑揚が適切でないためユーザが質問されたことに気づかないなどの事態も起こり得る．提案手法では，ロボットとの対話に特化して開発されたボイスフォントを利用可能である．本ボイスフォントは，非モノローグ HMM 音声合成 [13] により生成される．サービスロボットへの応用を想定した被験者実験を行い，ベースライン手法に比べて品質が優れるという結果を得ている．実験の詳細は，[13] を参照されたい．なお，上述した音声合成サービスから複数のボイスフォントを利用可能であるが，非モノローグ音声合成が可能なボイスフォントは 1 種類のみである．

## 3. 実　　験

### 3.1　実験設定

本実験では，処理速度の面から提案システムの実用性を評価する．これは，クラウド型の処理速度が実用的な範囲でなければ，スタンドアロン型を用いることが合理的であるためである．主な評価は，音声合成サービスについて行う．音声認識サービスの処理速度および精度については，[2] が詳しい．また，音声合成の品質評価については，[13] を参照されたい．実験に用いたサーバの CPU は Intel 製 X5690（12 コア，3.47GHz），メモリは 200GB であった．以下の 2 種類の実験を行った．

(a)　サービスロボットタスク

想定する用途に近いテストセットを用い，処理時間を計測した．ロボカップ@ホーム [4] で用いられたプロンプトから 150 文を抽出し，テストセットとして用いた．表 1 に例文を示す．各文を 5 秒おきにサーバに送出し，音声合成させた．

(b)　実証実験タスク

一般ユーザの音声合成サービス利用データを分析した．2013/9/6 から 2013/9/16 までのアクセス記録を分析し，実際の利用における処理時間を分析した．

表 1　サービスロボットタスクにおけるテストセットの例

| |
|---|
| これから綿菓子を作ります，準備が整ったら合図して下さい． |
| すみません，物体をつかめません． |
| すみません．飲み物が見つかりません． |
| はい，分かりました． |
| ペットボトルを覚えました． |
| ペットボトルを把持します． |
| リビングルームまで移動します，いいですか？ |
| 時間になりました． |
| 次の人は正面に立って下さい． |
| 他の所で探してみます． |
| 物体を取ってきます，いいですか？ |
| 綿菓子が完成しました． |

### 3.2　結果 (a):サービスロボットタスク

表 2 の第 2 列に処理時間および RTF の中央値を示す．表 2 において，リアルタイムファクター（RTF）は以下で定義される．

$$\mathrm{RTF} = \frac{T_p}{T_o} \tag{1}$$

ここに，$T_p$ と $T_o$ は，それぞれ処理時間と合成音声の長さを表す．表 2 より RTF は 0.1 程度であるので，合成された音声の長

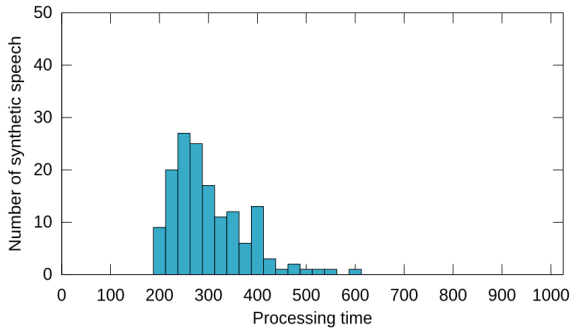


図 5　サービスロボットタスクにおける処理時間のヒストグラム

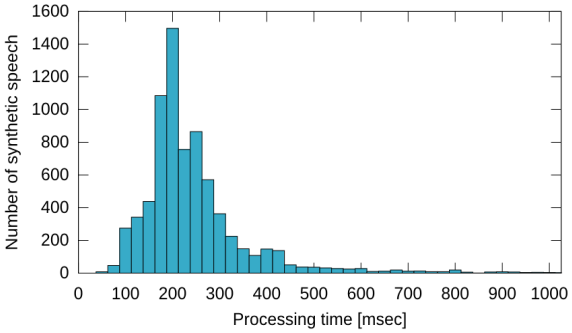


図 6　実証実験タスクにおける処理時間のヒストグラム

さは処理時間の約 10 倍であったことがわかる．

図 5 にサーバが音声合成に要した時間の分布を示す．処理時間は 200 から 400 [msec] の間に集中している．処理時間が比較的長いものは，2 文に分割できるものが多い．例えば，表 1 中の「これから綿菓子を作ります，準備が整ったら合図して下さい.」が挙げられる．本実験では，実際のユーザがしばしば分割せずに送信することから，このような文をテストセットから除かず，そのまま用いている．

表 2　サーバの処理時間．(a) サービスロボットタスク，(b) 実証実験タスク．

| タスク | (a) | (b) |
|---|---|---|
| ユニーク IP 数 | - | 813 |
| 合成処理数 | 150 | 7556 |
| 処理時間（中央値）[msec] | 270 | 203 |
| RTF（中央値） | 0.0977 | 0.115 |

### 3.3　結果 (b):実証実験タスク

表 2 の第 3 列に，処理時間および RTF を含むアクセスの分析結果を示す．ユニーク IP は，NICT 自身からのアクセス数を除いた数である．サービスロボットタスクと同様，RTF の中央値は 0.1 程度であった．

図 6 に，実証実験タスクにおいてサーバが音声合成に要した時間の分布を示す．図より，多くの発話は 400[msec] 以内に処理が終了していることがわかる．ただし，数は少ないものの，処理が 1000[msec] 以上かかる場合も存在する．長文のテキスト入力（ニュースサイトの本文のコピーなど）が行われる場合があったためである．これは，サンプルとしてブラウザからのテキスト入力インタフェースを用意したことにより，容易に長文を入力できたためであると考えられる．最長の入力は 360 文字であり，出力として 26.8[sec] の音声が合成され，処理時間は 3127[msec] であった．ただし，入力は複数の文から構成されていたため，適切に分割すれば全体として待ち時間を大幅に短縮できると考えられる．以上のように，本実験ではネットワーク遅延が考慮されていないものの，実用的に十分な処理時間であったといえる．

## 4.　おわりに

ネットワーク接続を前提とすれば，音声認識・合成処理をクラウド化することで，ロボットの低コスト化が可能である．そこで本論文では，ROS に対応した音声コミュニケーションツールキットを提案した．実証実験の結果，実用的に十分な処理時間であることが示された．また，音声合成の被験者実験では，ベースライン手法に比べて品質が優れるという結果を得ている．本ツールキットの応用としては，サービスロボット・音声対話システムの開発，音声の書き起こし，文書の読み上げ，などが挙げられる．

**謝辞**　本研究の一部は，科研費（若手 (B)24700188，および新学術領域「人ロボット共生学」公募課題 24118710）の助成を受けて実施されたものである．

### 文　献

[1] 河原達也，“音声対話システムの進化と淘汰–歴史と最近の技術動向–,” 人工知能学会誌，vol.28，no.1，pp.45–51，2013.
[2] 松田繁樹，林輝昭，葦苅豊，志賀芳則，柏岡秀紀，安田圭志，大熊英男，内山将夫，隅田英一郎，河井恒，中村哲，“多言語音声翻訳システム“ VoiceTra ”の構築と実運用による大規模実証実験,” 電子情報通信学会論文誌，vol.J96-D，no.10，p.in print，2013.
[3] 大浦圭一郎，山本大介，内匠逸，李晃伸，徳田恵一，“キャンパスの公共空間におけるユーザ参加型双方向音声案内デジタルサイネージシステム,” 人工知能学会誌，vol.28，no.1，pp.60–67，2013.
[4] 杉浦孔明，“ロボカップ@ホームリーグ,” 情報処理，vol.53，no.3，pp.250–261，2012.
[5] 松阪要佐，“OpenHRI,” 日本ロボット学会誌, vol.31, no.3, pp.2–7, 2013.
[6] K. Nakadai, T. Takahashi, H.G. Okuno, H. Nakajima, Y. Hasegawa, and H. Tsujino, “Design and Implementation of Robot Audition System’HARK’—Open Source Software for Listening to Three Simultaneous Speakers,” Advanced Robotics，vol.24，no.5-6，pp.739–761，2010.
[7] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, “Cloud-Based Robot Grasping with the Google Object Recognition Engine,” Proc. ICRA, 2013.
[8] K. Kamei, S. Nishio, N. Hagita, and M. Sato, “Cloud Networked Robotics,” Network, IEEE, vol.26, no.3, pp.28–34, 2012.
[9] M. Tenorth, A.C. Perzylo, R. Lafrenz, and M. Beetz, “The RoboEarth Language: Representing and Exchanging Knowledge about Actions, Objects, and Environments,” Proc. ICRA, pp.1284–1289, 2012.
[10] http://julius.sourceforge.jp/
[11] http://www.cstr.ed.ac.uk/projects/festival/
[12] http://open-jtalk.sp.nitech.ac.jp/
[13] 杉浦孔明，志賀芳則，河井恒，翠輝久，堀智織，“サービスロボットのための非モノローグ HMM による音声合成,” 第 31 回ロボット学会学術講演会資料，pp.2C1–02，2013.