

AI アシスタントとロボット教材を活用した 手話通訳ロボットの製作

河野 翔哉 *

日浅 茉奈人 *

福村 薫登 *

田中 大介 **

Development of a Robot for Sign Language Interpreting using an AI Assistant and Robotics Education System

Syoya KOUNO*

Manato HIASA*

Masato FUKUMURA*

Daisuke TANAKA**

Today, AI assistant is used not only in applications of smartphones but in various fields and industries such as SNS, IoT, medical and education. But, deaf people can't use interactive AI assistant by voice because it is a dialogue only by voice. Therefore, we develop a robot which can translate voice into sign language. This paper shows the overview and components of the robot and the validation result.

1. はじめに

現在、AI アシスタントはスマートフォンのアプリケーションだけではなく、SNS や IoT、医療、教育などの様々な分野、業界に普及している。その処理性能も CPU などのハードウェアの性能向上や音声・画像・自然言語に対する性能を格段に向上させるディープラーニングなどの機械学習により、非常に高いものと言える。また、それらは現在も成長を続け、さらなる高機能化が予測され、さらなる普及、需要増加は想像に難くない。しかし、音声による対話型の AI アシスタントは音声のみの対話であることが多いため、聴覚障害者の利用は非常に困難である。そこで簡単にシステムの構築が可能であり、ロボット学習に用いられている「Lego Mindstorms EV3」(以下、EV3 と略す) を使用して、聴覚障害者との対話が可能な手話通訳ロボットを開発した。これにより、市販のロボットハンドを用いるよりも安価に、AI アシスタントを通じた聴覚障害者との対話を可能とするロボットが開発できた。

本稿では、この開発したロボットのシステム構成と、動作結果について報告する。

ばれる、アマゾンウェブサービス (以下、AWS と略す) で提供されている、必要なときに必要な処理を自動で行ってくれるサービスがあり、それによりサーバレスで簡単にシステムを構築できる。このことは、ユーザビリティの向上に役立つことから、本研究では Alexa を採用した。

以降の章では、このシステムの構成について説明する。

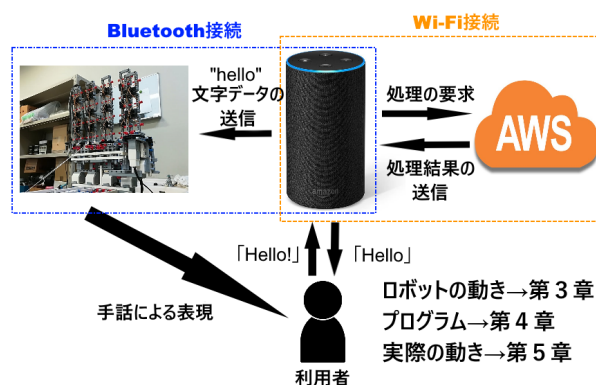


図1 システム構成

2. 提案システム

本研究において開発した、手話通訳ロボットの全体の処理の流れを図1に示す。AI アシスタントには Amazon が開発した「Alexa」を利用した。Alexa には「AWS Lambda」と呼

3. ロボット本体

手話は様々な言語において利用されているが、今回開発したロボットでは英語を対象言語とする手話通訳を行う。この理由として、日本語の手話にすると、腕や両手が必要になり

令和2年12月23日受付 (Received Dec. 23, 2020)

* 新居浜工業高等専門学校 電子制御工学科 (Department of Department of Electronics and Control Engineering, National Institute of Technology (KOSEN), Niihama College, Niihama, 792-8580, Japan)

** 新居浜工業高等専門学校 機械工学科 (Department of Mechanical Engineering, National Institute of Technology (KOSEN), Niihama College, Niihama, 792-8580, Japan)

複雑な動きも多いため、実現が困難であることが挙げられる。これに対して、英語であれば指の形と手首の動きだけで表すことが可能であるため、英語を採用した。以降、この英語の手話の表現のために必要なロボットのハードウェア的な構成について説明する。

3-1 ロボット本体の全体

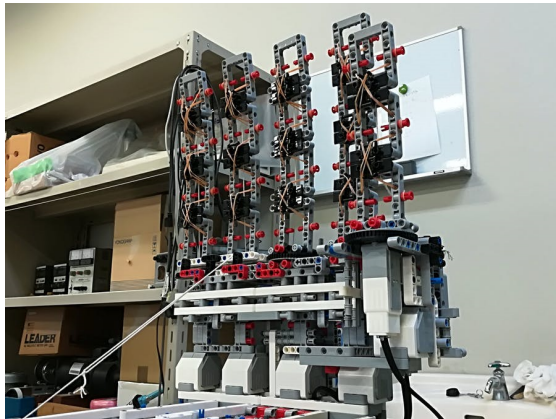


図2 ロボットの全体像

ロボットの全体像を図2に示す。このロボットは、ロボット教材であるEV3を用いて作成した。EV3には基本セット・応用セットと呼ばれる部品セットが用意されており、今回は基本セットと応用セットをそれぞれ3セット用いて作成した。また、EV3のセットの他、市販の輪ゴム、タコ糸で関節部分を駆動させている。一般的に、人間の手と同じ自由度のロボットハンドは高価であるが、本システムはそれより安価に構成することができる。

以下にセット内で使用した主なパーツを示す。

- ・ 大型タイヤ 8 個
- ・ L モータ 6 個
- ・ M モータ 2 個
- ・ インテリジェントブロック EV3 3 個

図2に示したような、手を完全に開いており、垂れていない状態がこのロボットの初期状態である。

3-2 インテリジェントブロック EV3

Alexaからのデータ受け取りやモータ、センサーなどの制御を行うコンピュータである。入力ポートと出力ポートがそれぞれ4つあり、センサーは入力ポート側、モータなどは出力ポート側にケーブルを使って接続することで制御が可能になる。今回はPythonを使用するために元から入っているオペレーティングシステム(以下、OSと略す)ではなく、「ev3dev」と呼ばれるOS[1]を使用した。ロボットに取り付けているインテリジェントブロックを図3に示す。



図3 インテリジェントブロック

それぞれのインテリジェントブロックは以下のものを制御している。

1. 人差し指、中指、薬指
2. 親指、小指
3. 手首

EV3側のプログラムの説明は第4章2節に記載している。

3-3 LモータとMモータ

EV3のキットには「Lモータ」と「Mモータ」と呼ばれるモータが梱包されている。両方が有している特徴として、1[deg]単位で制御が可能であることが挙げられる。異なる点は、Lモータは回転速度が遅い代わりにトルクが大きく、Mモータはトルクが小さい代わりに回転速度が速いことである。このことから力が必要な関節にはLモータを使用し、逆に速度を求める関節にはMモータを使用した(図4参照)。

3-4 指の動き

指の関節部分となる場所には輪ゴムを取り付けており、指の先からLモータに接続されているタコ糸で指を張ってこの状態にしている。表現するアルファベットによって後部に取り付けられているモータが回転し、張っているタコ糸を緩ませて指や手首を曲げ、手話を表現している。この構成は、人間の手の腱の動きを参考にした。後部の写真を図4に示す。

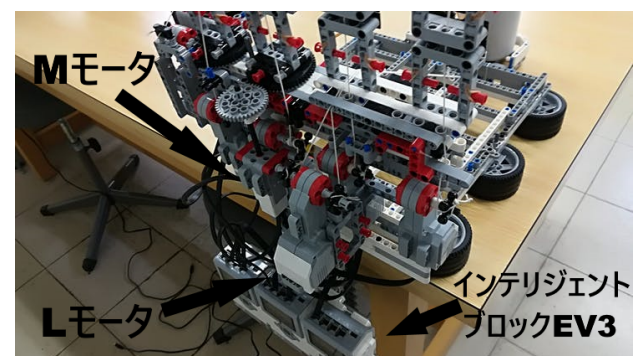


図4 ロボットの後部

手を動かす範囲を確保するためモータやインテリジェントブロックなどは後部に配置し、可動部分には十分なスペース

を取っている。

輪ゴムは指の先、人間で例えると第 1 関節から曲がるようにするため、上から順にかかっている輪ゴムの力を弱くしている。こうすることでタコ糸が緩んだとき、最初に先の関節部分が曲がる。指部分の写真を図 5 に示す。

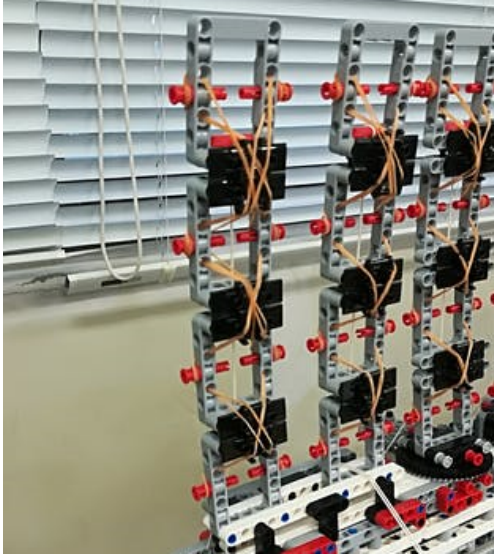


図 5 指の関節部分

掛けている輪ゴムの数を調整することで、力の加減を決めている。また、トルクがある L モータを使用することで、複数の輪ゴムの力に十分耐える事ができ関節を制御している。

中指と親指に当たる部分は少し特殊であり、横方向に指を回転させる必要があるため、中指は下にギアを取り付け M モータで動かし (図 6)、親指は真下に M モータを設置し直接回転するようにした (図 7)。



図 6 中指の後部

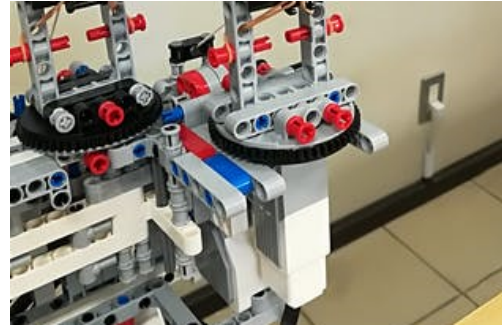


図 7 親指部分

3-5 手首の動き

ロボット全体の角度は、ロボットの土台の先とタコ糸で繋がっている L モータが制御している。ロボットの下部にインテリジェントブロックを 3 台取り付け、すべてのモータの制御をすると同時に重りとしての役割もしている。机の角にロボットを設置することで、机の側面とロボットの下部が当たりバランスが取れ後ろや前に倒れないようになっている。そこに L モータでタコ糸を巻くことでロボット全体が前に傾き、手首の動きを再現している。

4. プログラム

本章では、前章で説明したロボットを動かすためのソフトウェアについて説明する。

4-1 Alexa 側のプログラム

Alexa 側のプログラムは第 2 章でも説明したように、AWS Lambda と呼ばれる AWS で提供されている環境で動作する。また Alexa Gadgets Toolkit[2] と呼ばれる、AWS Lambda と同様 Amazon が提供しているサービスであり、Bluetooth で Alexa と連携できるガジェットの制御を Alexa を通して指示、処理などを行わせるシステムを開発できる環境 [3] があり、これも使用した。

本ロボットの制御のために、主に以下の 3 種類のプログラムを実装した。図 8 にフローチャートを示す。

- メインプログラム
- パッケージ管理ファイル
- 対話モデル設定ファイル

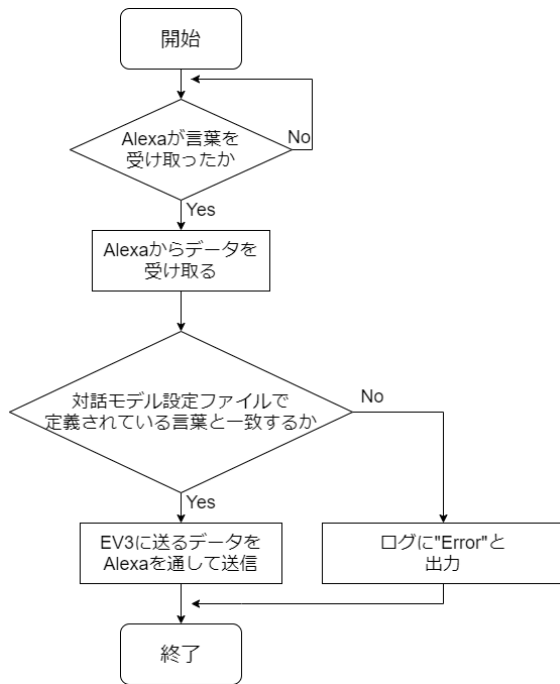


図8 Alexa側のフローチャート

Alexa が言葉を受け取ったか確認し、受け取ったデータを参照し対話モデルファイルで定義された言葉であるか確認する。定義された言葉であればメインプログラム内の関数を使用し、EV3 にデータを送信している。定義されていない言葉であれば例外処理を実行する。

4-1-1 対話モデル設定ファイル

Alexa が聞き取る言語や特定の言葉を定義している。本プログラムでは英語であるかどうかを確認させ、他の言語であれば例外処理に分類する。

4-1-2 メインプログラム

Alexa のメインプログラムは「index.js」、「util.js」によって構成されている。

「util.js」は Bluetooth 接続されているガジェットに送るデータの設定をしている。

「index.js」が最も重要なプログラムである。接続されているガジェットの設定や Alexa から受け取ったデータの処理に必要な関数を定義している。本プログラムでは Alexa から受け取ったデータを AWS Lambda 上で読み取れる形に変換し、そのデータをさらに文字データに変換し3つのインテリジェントブロックに送信する関数を定義した。

4-2 EV3 側のプログラム

EV3 側のプログラミング言語は、安易にモジュールの追加やモータ、センサーの制御が可能な Python を使用した。また、OS には EV3 で Python が実行可能な `ev3dev[1]` を使用した。

4-2-1 EV3 側のフローチャート

図9にEV3側のフローチャートを示す。

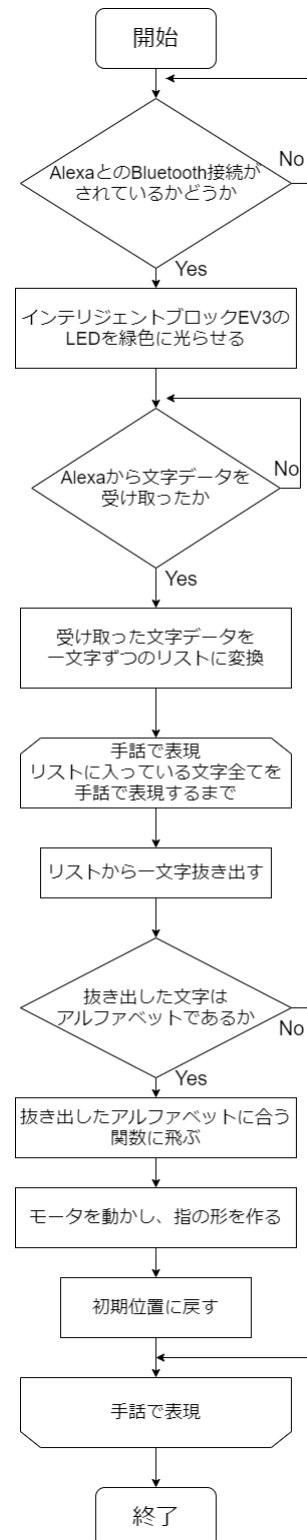


図9 EV3側のフローチャート

まず Alexa との Bluetooth 接続がされているか確認をする。接続が確認後、Alexa を介して文字データを受け取りそれをリスト化し、順番に文字を手話で表現していく。

4-2-2 クラス AlexaGadget

「Alexa Gadgets Toolkit」の API から提供されているクラスが「AlexaGadget」である。class クラス名 (AlexaGadget) と記述することでそのクラスに継承され、初期化メソッドを使用して呼び出しが可能となる。

4-2-3 Bluetooth 接続

Bluetooth 接続には、外部デバイス (スマートフォンやパソコン) による Alexa とのペアリングが行われていれば自動的に接続される。その確認には先程説明したクラス AlexaGadget が必要となる。クラス AlexaGadget で用意されている Bluetooth の接続を確認する関数を記述することで、接続されたデバイス名や状態を出力できるようになる。また、接続が切れたときに実行される関数も用意されている。

4-2-4 データの受け取りと実行

Alexa から文字データを受け取る関数は Bluetooth 接続と同様にクラス AlexaGadget で用意されており、記述し変数に代入することで受け取ることができる。その処理のフローチャートを図 9 に示す。

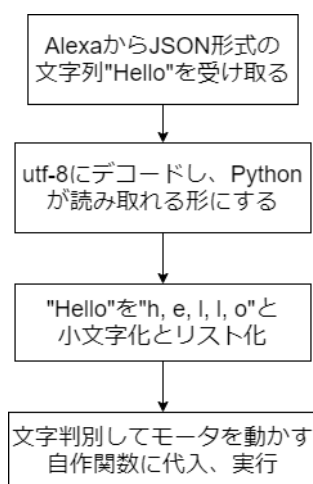


図 9 文字データの処理

あとは、図 9 に示したフローチャート内のループのように一文字ずつ文字を判別し、それに合うモータを動かす関数を実行するだけである。

5. 動作確認

本稿では、Hello という言葉に対して本システムを動作させ、動作を確認した結果を示す。なお、アルファベットすべての文字が実現可能であるが、紙面の都合上割愛させていただく。Hello という言葉を表現する際の動きの流れを図 10 に示す。ロボットは以下の順番で動いている。

1. 利用者が「Hello」と言う。このときはまだ初期位置である
2. "h"の形を作る
3. "e"の形を作る
4. "l"の形を作る
5. "l"の形を作る
6. "o"の形を作る
7. 初期位置に戻る

なお、現在の文字から次の文字の形を作る際、一度初期位置に戻ってから次の文字の形を作っている。図 10 の動作より、発話した音声をアルファベットに分解し、それぞれを表現できることが確認できた。

なお、“j”と“z”は動きを伴って表現するため、他のアルファベットの表現に対して少し特殊である。その動きをそれぞれ図 11 と図 12 に示す。これらの実現のため、指の形を作ったあと、手首を動かして表現している。実際の“j”の表現は小指以外を握って空に“J”と書くが、手首の動きは上下しか再現できていないため上下に動くだけで表現することとした。“z”も同様に実際は人差し指で空に“z”と書いて表現するが、人差し指以外を握って上下に動くだけとした。

6. あとがき

本稿では、AI アシスタントを用いて難聴者との対話を可能にすることを目的として、市販のロボットハンドよりも安価に開発する方法として、ロボット教材として用いられる LEGO Mindstorms EV3 を用いたシステムを開発した。対象言語は英語とし、手話特有の動きも実現できるような関節構成・プログラムを実装して、すべてのアルファベットを表現可能であることを確かめた。

現状のシステムは、比較的安価に構成することができたものの、このロボットには EV3 の基本セット、応用セットをそれぞれ 3 セットに含まれているパーツをほとんど使用したため、とても大きく重量があるため移動が難しい点がある。ま

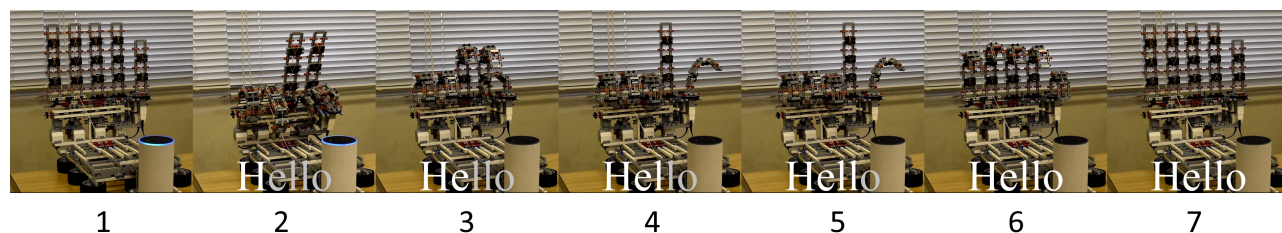


図 10 Hello を手話で表現

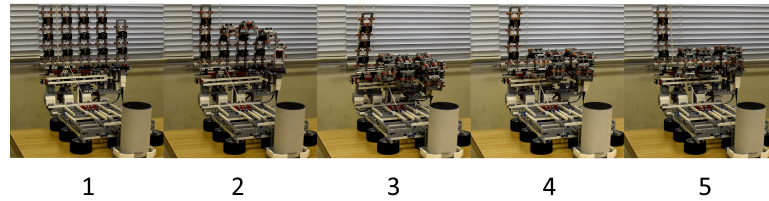


図 11 "j"の動き

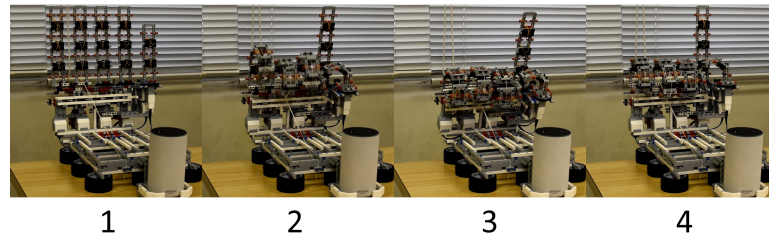


図 12 "z"の動き

た、ある程度のスペースがある机が必須であり、机の角に設置する必要があるため場所にも制限がある。プログラムに関してもさらなる最適化は可能であり、Bluetooth 接続であるためタイムラグが生じる可能性もある。今後はこれらの点の改良を行い、使いやすいシステムを目指したい。

なお、本ロボットを用いて、2019 年 9 月 24 日 (PT: 太平洋時間) から 2020 年 1 月 17 日 (PT) の期間で開催されたコンテスト、「LEGO MINDSTORMS Voice Challenge: Powered by Alexa」に参加した。このコンテストは、Amazon とレゴ社が主催し、hackster.io[4] で開かれ、Alexa 搭載のデバイスと EV3 を用いて革新的なロボットを作るというのがコンセプトであった。本研究で開発したロボットは、早期に作品を提出した先着 100 名に授与される「First 100 Submissions」を獲得した (図 13)。

参考文献

- [1] ev3dev ホームページ
<https://www.ev3dev.org/>
- [2] Alexa Gadgets Toolkit Documentation
<https://developer.amazon.com/ja-JP/docs/alexa/alexa-gadgets-toolkit/understand-alexa-gadgets-toolkit.html>
- [3] EV3 Python API Documentation
<https://python-ev3dev.readthedocs.io/en/ev3dev-stretch/spec.html>
- [4] hackster.io コンテストホームページ
<https://www.hackster.io/contests/alexa-lego-voice-challenge#category-717>
- [5] Can Alexa talk with deaf people?
<https://www.hackster.io/316472/can-alexa-talk-with-deaf-people-d54b0e>

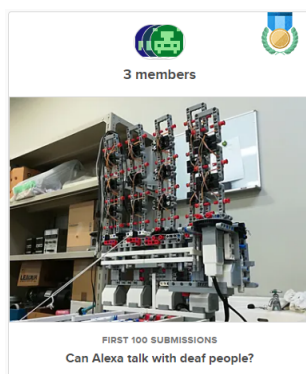


図 13 First 100 Submissions

hackster.io[5] にコンテストで作成したレポートがおりてあり、本稿には示していない詳しい内容を説明している。また、実際に動いている動画も掲載されているため、参照されたい。