

ロボットアームの複数命令のラベル付けによるユーザの 指示数の削減

北川 翔也¹ 大沢 恭平² 串田 高幸¹

概要：東京工科大学の Cloud and Distributed Systems Laboratory(CDSL) では 6 軸ロボットアームがある。ロボットアームを動かしているシステムはクライアントからパルス変換サーバー、ESP32 の順番で命令を送信している。クライアントはユーザが入力した命令をパルス変換サーバーに送信する。また、クライアントはユーザが送信した命令から Client.log を生成する。Client.log はパルス変換サーバーに送信される。課題はクライアントからパルス変換サーバー間の連続した命令が 1 つのまとまりとして扱われておらず、ラベル付けもされていないため、ユーザは同じ連続した命令を毎回個別に入力する必要がある。その結果、ユーザがロボットアームに与える指示の回数が増加する。提案では、クライアントがパルス変換サーバーに命令を送信するときに生成される Client.log から頻出する命令の連続パターンを抽出し定義されたルールをもとにラベル付けを行う。これにより、クライアントからパルス変換サーバー間の命令ステップを削減する。評価実験ではクライアントからパルス変換サーバー間の命令のステップ数の削減率を評価する。本提案を適用したあとのクライアントからパルス変換サーバー間の命令ステップ数を提案の適用前の命令ステップ数と比較する。基礎実験では白色の箱に入っている 9V 乾電池を透明の箱に移す際に必要な命令ステップ数を 3 パターン測定した。パターン 1 を基準としてパターン 2 とパターン 3 の命令の一致率を分析した。パターン 1 とパターン 2 ではパターン 1 のステップ数が 11 でパターン 2 のステップ数が 15 であり 7 個が共通で一致率が約 63.64%、パターン 1 とパターン 3 ではパターン 1 のステップ数が 11 でパターン 3 のステップ数が 14 で 7 個が共通で一致率が約 63.64%であった。また、最初と最後に使用する PIN 番号と角度が一致することが分かった。

1. はじめに

背景

ロボットアームは自律的に、あるいは指示に従ってタスクを実行できる機械的な知能エージェントである [1]。組立、搬送、検査、加工や倉庫のピッキング、医療、教育に利用されている [2]。倉庫のピッキングではベルトコンベアで流れてきたボックスに対し、命令された商品を別のボックスからピックし箱詰めを行う。ロボットアームはサーバーに対し 3D カメラ画像、接近センサーのデータをサーバーに送信する。サーバーは受信データをもとに命令計画を立て、ロボットアームに送信する。

ロボットアームには単腕ロボットと双腕ロボットがある。双腕ロボットの活用事例として 1 回の動作で 2 つの化粧品をピックアップする化粧品包装や IT パーツのパッケージがある [3,4]。ロボットアームの制御方法では、ジェ

スチャーを用いたロボット制御や音声制御がある [5,6]。

Internet of Things(IoT) は状況や環境の変化に応じて応じ行動し、情報・データ・リソースを自動的に組織化し共有する能力を持つ、オープンで包括的なインテリジェントオブジェクトのネットワークと定義されている [7]。IoT の事例としてスマート農業、倉庫管理、スマート工場、スマートヘルスケアがある [8]。

Cloud and Distributed Systems Laboratory(CDSL) には 6 軸ロボットアームがある。6 軸ロボットアームにはサーボモーターをもちいている。ロボットアームを動かしているシステムはクライアントからパルス変換サーバー、ESP32 の順番で命令を送信している。クライアントはユーザが入力した命令をパルス変換サーバーに送信する。ESP32 は Wi-Fi と Bluetooth を内蔵したマイクロコントローラである。PCA9685 は PWM 出力ができる I²C 接続のサーボモーターコントローラである。また、クライアント、パルス変換サーバー、ESP32 で生成されたログを貯蓄し、ログサーバーにログを送信する。パルス変換サーバーからの命令を PCA9685 に送信する IoT 機器としてパルス変換サー

¹ 東京工科大学コンピュータサイエンス学部
〒192-0982 東京都八王子市片倉町 1404-1

² 東京工科大学大学院バイオ・情報メディア研究科コンピュータサイエンス専攻 クラウド・分散システム研究室

バーに接続する。また、ログを生成し、パルス変換サーバーに送信する。ロボットアームに対する命令には、特定の命令の後に必ず別の決まった命令が続く連続したパターンが存在する。これらの命令列が1つのまとまりとして扱われていない。また、ラベル付けもされていない。クライアントはユーザが送信した命令から Client.log を生成する。Client.log はパルス変換サーバーに送信される。

サーボモーターとはシャフトを高精度に制御できる電気モータの一種である。サーボモーターの制御方式は、PWM制御、電流制御、2ループ制御、ファジー制御、プログラマブル制御がある [9]。Pulse Width Modulation(PWM)とは固定周波数と固定新幅を持っており、可変のパルス幅を持つパルス列である。パルスの幅は変調信号によって変化する [10]。

課題

課題はユーザは連続した命令を毎回個別に入力する必要があり、ロボットアームに与える指示の回数が増加している。図 1に連続して命令が続いていること状況を示す。

```
1 35c780ca:2025-07-23 14:35:23.850:  
C_SEND:PIN:0_angle:180  
2 35c791f6:2025-07-23 14:35:28.246:  
C_SEND:PIN:0_angle:0 PIN:1_angle:80  
PIN:12_angle:60  
3 35c77130:2025-07-23 14:35:39.856:  
C_SEND:PIN:11_angle:90  
PIN:1_angle:90 PIN:2_angle:90  
4 35c7d0ca:2025-07-24 14:35:23.250:  
C_SEND:PIN:0_angle:180  
5 35c731f6:2025-07-24 14:35:28.136:  
C_SEND:PIN:0_angle:0 PIN:1_angle:80  
PIN:12_angle:60  
.  
.  
.
```

青色の命令のあとに赤色の命令が
実行されている

図 1: 連続して命令が続いている状況

図 1は Client.log の一部抜粋したものである。また、TraceID, timestamp, C.SEND, 命令の順番で構成されている。C.SEND は Client SEND の略である。PIN:0_angle:180 は PCA9685 で使用している PIN 番号と角度を表している。同様に PIN:0_angle:0 PIN:1_angle:80 PIN:12_angle:60 も PCA9685 で使用している PIN 番号と角度を表している。PIN:11_angle:90 も PCA9685 で使用している PIN 番号と角度を表している。

各章の概要

第 2 章では関連研究について説明する。第 3 章では本稿における提案について説明する。第 4 章では提案手法の実装について説明する。第 5 章では提案手法をもとに実験をし、評価する。第 6 章では本稿の提案手法について議論を行う。第 7 章では本稿のまとめを行う。

2. 関連研究

ロボット工学では深層学習をもちいている [11]。深層学習にはラベル付けがついているデータが大量に必要であるが、時間と手間がかかる。この研究ではラベル付け作業を半自動化する方法を提案している。類似したセンサーデータをクラスタリングしてまとめることでラベル付けをしやすくしている。本稿でもちいている 6 軸ロボットアームではセンサーデータを生成していないため適していない。

知能ロボットが人間からの命令を理解し行動するためにはロボットの関節角度や数値データと人間の表現を対応付ける必要がある。一方で、ロボットの関節角度や数値データは人間にとって理解困難であり、コミュニケーションの障害となっている [12]。この研究はアフォーダンスモデルを用いたアプローチを提案しているこの研究ではロボットの操作動作の結果に対して人間が自然言語のラベルを付与し、SVM を用いて動作とラベルの関係を学習している。だが、使用されているラベルは「push」「tip」「move」の短い単語が中心であり多関節ロボットの一連の動作には適していない。

支援ロボットのユーザは、日常生活の繰り返し動作をロボットに毎回手動で操作しなければならず、操作負担が大きいという課題がある [13]。この研究は Dynamic Movement Primitives(DMP) という軌道学習手法をもちいて、ユーザがロボットを操作した動作を記録する。記録した動作にラベル付けをする。ユーザは呼び出して再実行できる「スキル」として再利用する仕組みを提案している。この研究は軌道ベースでのラベル付けを行っており、命令列に対するラベル付けを目的とする本稿には適していない。

3. 提案

提案方式

本提案では、クライアントが送信した命令から生成される Client.log をもとに頻出する命令の連続パターンを抽出しラベル付けを行う手法を提案する。抽出されたパターンに対して事前に定義された PIN 番号と角度とラベル名のルールを適用してラベル付けを行い、パルス変換サーバー内のデータベースに保存する。提案をもとにつけられたラベルはユーザに送信され、ユーザはラベル名を入力するだけでラベル付けした連続する命令を実行することができる。

3.1 命令の抽出とハッシュ値の導出と保存

Client.log から行単位で順次読み込み、正規表現の $\sim[^\sim:]\sim+:([^\sim:]\sim+):C_SEND:(.\sim+)\$$ をもちいて timestamp と命令を抽出し、抽出した命令に対して xxHash64 関数をもちいてハッシュ値を求める。求めたハッシュ値と timestamp に対し時系列順に 1 から始まる連番を振り、連番、ハッシュ値、命令の順でデータベースに保存する。ハッシュ値を求める理由として、文字数が長い命令を固定長のハッシュ値に変換することで命令の比較処理を簡単にできると、同一の命令に対して常に同一のハッシュ値が生成されるため確実に命令の一致を判定できるためである。また、ユーザから新たに送信されてきた複数の命令に対してもハッシュ値を求める。

3.2 繰り返しパターンの比較とラベル付け候補の出力

「3.1 命令の抽出とハッシュ値の導出と保存」で保存したデータベースから連続したパターンを抽出する。データベースから 1 番目と 2 番目のハッシュ値を選択し、パターンとして一時保存する。この選択したハッシュ値のパターンが 1,2 番以外で連続して存在しているか確認する。存在した場合は 3 番目のハッシュ値をパターンに追加し、再度データベース内で同じパターンが存在するかを確認する。この処理をデータベースの最後まで行う。また、1 番目と 2 番目のハッシュ値パターンは別で存在しているが、3 番目のハッシュ値を追加したパターンが別で存在していないとき、1 番目と 2 番目のハッシュ値パターンをラベル付け候補として出力する。連続した命令の比較の流れをアルゴリズム 1 に示す。入力は、ハッシュ値列 $H = [h_1, h_2, \dots, h_n]$

Algorithm 1: 連続した命令の比較の流れ

Data: ハッシュ値列 $H = [h_1, h_2, \dots, h_n]$
Result: 候補パターン集合 C

```
1  $C \leftarrow \emptyset;$ 
2 for  $i = 1$  to  $n - 1$  do
3    $P \leftarrow [H[i], H[i + 1]], len \leftarrow 2, last \leftarrow \emptyset;$ 
4   while パターン  $P$  が 2 回以上出現 do
5      $last \leftarrow P;$ 
6     if  $i + len \leq n$  then
7        $P \leftarrow P + [H[i + len]], len \leftarrow len + 1;$ 
8     end
9   else
10    break;
11  end
12 end
13 if  $last \neq \emptyset$  かつ  $last \notin C$  then
14    $C \leftarrow C \cup \{last\};$ 
15 end
16 end
17 return  $C;$ 
```

である。これは、クライアントから送信された命令を時系列順にハッシュ化した値が格納されている。1 行目では、候補パターン集合 C を空集合で初期化する。この集合は、最終的に繰り返し出現するパターンを格納するために使用される。3 行目から 15 行目では、外側のループでハッシュ値列の各位置 i について処理を行う。まず、 i 番目と $i + 1$ 番目のハッシュ値を選択して初期パターン P として設定し、パターン長 len を 2 に、最後に有効だったパターン $last$ を空に初期化する。5 行目から 11 行目では、内側の While ループでパターンの延長処理を行う。現在のパターン P がデータベース内で 2 回以上出現する限り、そのパターンを $last$ に保存し、パターンをさらに 1 つ延長する。延長できない場合はループを終了する。13 行目から 15 行目では、最後に有効だったパターンをラベル付け候補として追加する処理を行う。 $last$ が空でない場合、かつそのパターンが候補集合 C に含まれていない場合に、候補パターン集合に追加する。17 行目では、すべての処理が完了した後、候補パターン集合 C を返す。

3.3 ラベル付け

「3.2 繰り返しパターンの比較とラベル付け候補の出力」で繰り返し出現する命令のパターンの文字列に対し事前に定義されたルールをもとにラベル付けを行う。表 1 に定義されたルールを示す。また、ラベルの順番は 1 番目に動作の特徴を 2 番目に PIN0 のラベルを 3 番目に PIN1/PIN2/PIN12 で 4 番目は PIN14 の $0^\circ \rightarrow 180^\circ$ で 5 番目は PIN14 の $180^\circ \rightarrow 0^\circ$ でラベルを結合する。結合するときはラベルの間に半角スペースを入れる。

ポットアームの各 PIN 番号の角度にもとづいてラベルを生成するためのルールの定義を示している。例えば、PIN14 が 0° から 180° に変化した場合は「モノをつかんだ」、PIN0 が $61^\circ \sim 120^\circ$ の範囲にある場合は「正面」、PIN1 が $121^\circ \sim 180^\circ$ の場合は「後ろに動かして」というラベルが付与される。また、同一 PIN が 3 回以上調整された場合は「精密に」、全 PIN が 10° 未満の小刻みな変化をした場合は「ゆっくり」といった動作の特徴もラベル化する。

3.4 フィードバック

ラベル付けされたパターンは、ユーザに送信され、命令を入力するコマンドライン上で自らの命令がどのようなラベル付けされどのような意味が付与されたかを視覚的に確認することができる。

ユースケース・シナリオ

EC 倉庫におけるベルトコンベアで流れてきた容器に指示された物品を別の容器に移すロボットアームを想定する。図 2 にユースケースシナリオの図を示す。サーバーか

表 1: 定義されたルール

PIN 番号	条件/パターン	ラベル
PIN14	0° → 180°	モノをつかんだ
	180° → 0°	モノを離れた
PIN0	0° ~ 60°	右
	61° ~ 120°	正面
	121° ~ 180°	左
	0° → 180°	右から左に
	180° → 0°	左から右に
PIN1	0° ~ 60°	前に動かして
	61° ~ 120°	上にあげて
	121° ~ 180°	後ろに動かして
PIN2 と PIN12	両方 < 45°	遠く
	45° ~ 100°	真ん中に
	両方 > 100°	手前
動作の特徴	同一 PIN が 3 回以上調整	精密に
	全 PIN が 小刻み変化 (<10°)	ゆっくりと
	PIN13 の変化量 > 30°	向きを変えて

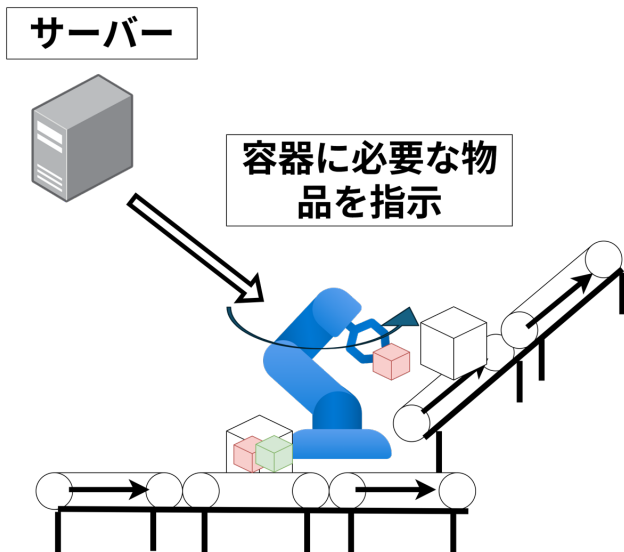


図 2: ユースケースシナリオ

ら必要な物品を指示され、ロボットアームは左から流れてきた容器に入っている赤い箱を別の容器に移している。

4. 実装

図 3 にロボットアームを動作させているシステム構成図を示す。赤矢印はロボットアームの命令経路を示しており、青矢印はログの経路を示している。システム構成図にある機器について以下に説明をする。

クライアント

- 役割：命令を作成，パルス変換サーバーに送信する。

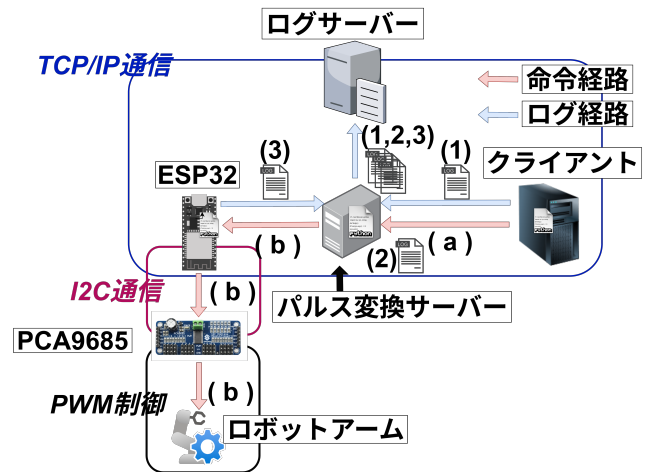


図 3: システム構成図

また、生成したログをパルス変換サーバーに送信する。送信は TCP/IP 通信である。

- 実行ファイルの言語: Python

パルス変換サーバー

- 役割：クライアントから送られてきた角度をパルスに変換する。また、クライアント、パルス変換サーバー、ESP32 で生成されたログを貯蓄し、TCP/IP 通信でログサーバーに送信する役割を持つ。
- 実行ファイルの言語：Python

ESP32

- 役割：パルス変換サーバーからの命令を PCA9685 に送信する IoT 機器としてパルス変換サーバーに接続する。また、ログを生成し、パルス変換サーバーに送信する。PCA9685 は I²C で制御している。

- 実行ファイルの言語：MicroPython

PCA9685

- 役割：各サーボモーターとジャンパー線で接続し、ESP32 からの命令を伝達する。
- 使用サーボモーター：MG996R を 5 つ、DS3225 を 1 つである。
- 使用している PIN：0,1,2,12,13,14

次に送信している命令 (a) と (b) についての詳細を以下に示す。

システム構成図では、クライアントからパルス変換サーバーに PCA9685 で使用している PIN 番号と角度を送信し、パルス変換サーバーで角度をパルス数値に変換し、ESP32 に PIN 番号とパルス数値を送信している。表 2 にシステム構成図の (a) と (b) の送信内容の詳細を示す。表 2 はクライアントから ESP32 までの送信内容とロボットアームが初期姿勢に動いた場合に必要の命令内容を例として記述している。(a) は PCA9685 で使用している PIN 番号と角度が送信内容である。(b) は PCA9685 で使用している PIN 番号と角度をパルスに変換した数値が送信内容である。

表 2: クライアントから ESP32 までの送信内容 (初期姿勢の場合)

番号	命令の形式	送信内容 (初期姿勢の場合)
(a)	PIN 番号と角度	PIN:0_angle:90 PIN:1_angle:90 PIN:2_angle:90 PIN:12_angle:90 PIN:13_angle:90 PIN:14_angle:0
(b)	PIN 番号とパルス幅	'PIN:0_on:0_off:291', 'PIN:1_on:0_off:235', 'PIN:2_on:0_off:291', 'PIN:12_on:0_off:291' 'PIN:13_on:0_off:291', 'PIN:14_on:0_off:102'

システム構成図におけるログサーバーに送信するクライアント、パルス変換サーバー、ESP32 が生成するログの構文について表 3 に示す。TraceID は 8 桁の 16 進数の文字

表 3: ログの構文

番号	ログ内容
(1)	Trace ID + ":" + timestamp + ":" + "C_SEND" + ":" + (a)
(2)	Trace ID + ":" + timestamp + ":" + "S_RECV" + ":" + (a) Trace ID + ":" + timestamp + ":" + "S_CONV" + ":" + (a) + ":" + (b) Trace ID + ":" + timestamp + ":" + "S_SEND" + ":" + (b)
(3)	Trace ID + ":" + timestamp + ":" + "E_RECV" + ":" + (b) Trace ID + ":" + timestamp + ":" + "E_ACT" + ":" + (b)

列であり、クライアントで生成され、パルス変換サーバー、ESP32 で同じ ID がもちられる。これにより、どこからどこまでが一連の命令であるか判別することができる。timestamp の形式は YYYY-MM-DD HH:MM:SS.mmm である。C_SEND は Client SEND の略であり、クライアントがパルス変換サーバーに送信したことを表す。S_RECV は Server Receive の略であり、パルス変換サーバーにクライアントから命令を受け取ったことを表す。S_CONV は Server Convert の略であり、パルス変換サーバーがクライアントから受け取った命令をパルスに変換したことを表す。S_SEND は Server Send の略であり、パルス変換サーバーが ESP32 に変換した命令を送信したことを表す。

E_RECV は ESP32 Receive の略であり、ESP32 がパルス変換サーバーから命令を受け取ったことを表す。E_ACT は ESP32 Action の略であり、ESP32 が受け取った命令を PCA9685 に送信し、ロボットアームの動作を行ったことを表す。

また、表 4 は表 3 の番号 (1), (2), (3) の送信条件を示している。(1) はクライアントが命令を送信した場合にログ

表 4: 送信条件

番号	送信条件
(1)	クライアントが命令を送信
(2)	クライアントからの命令を受信 パルス変換サーバーが命令の変換を完了 ESP32 に対し命令の送信を完了
(3)	パルス変換サーバーから命令を受信 PCA9685 に対し命令の送信を完了

を送信する。(2) はパルス変換サーバーがクライアントからの命令を受信した場合とパルス変換サーバーが角度をパルスに変換した場合、変換した命令を ESP32 に送信したときにログを送信する。(3) はパルス変換サーバーから命令を受信した場合と PCA9685 に命令を送信した場合にログを送信する。

5. 評価実験

評価実験ではクライアントからパルス変換サーバー間の命令のステップ数の削減率を評価する。本提案を適用したあとのクライアントからパルス変換サーバー間の命令ステップ数を提案の適用前の命令ステップ数と比較する。基礎実験基礎実験では白色の箱に入っている 9V 乾電池を透明の箱に移す際に必要な命令ステップ数を測定した。測定内容は PIN 番号と角度である。図 4 に基礎実験で行う動作の図を示す。箱 (白) には 9V 乾電池が入っている。箱 (透

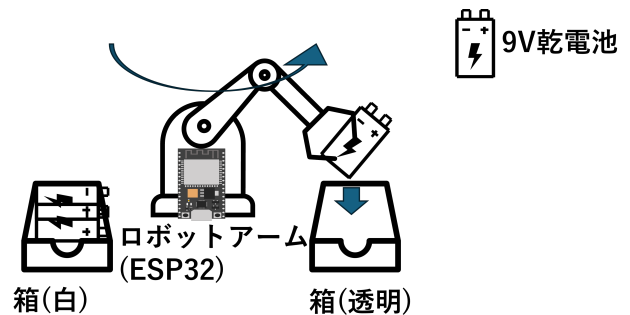


図 4: 基礎実験で行う動作

明) にはなにも入っておらず、ESP32 とつながっているロ

ボットアームが9V乾電池を移す動作である。

実験環境

場所は東京工科大学の研究棟 A910 で行った。ロボットアームは MG996R サーボモーターを 5 つ、DS3225 サーボモーターを 1 つ計 6 つのサーボモーターを搭載したものをを使用した。サーボモーターを PWM で制御するため PCA9685 をもちいる。PCA9685 を制御するため I²C 制御で ESP32 をもちいる。PCA9685 と ESP32 はジャンパー線で接続している。ロボットアームを動かしているシステムはクライアントからパルス変換サーバー、ESP32 の順番で命令を送信する。クライアントはユーザが入力した命令をパルス変換サーバーに送信する。ESP32 はパルス変換サーバーからの命令を PCA9685 に送信する IoT 機器としてパルス変換サーバーに接続する。PCA9685 は各サーボモーターとジャンパー線で接続し、ESP32 からの命令を伝達する。パルス変換サーバーはクライアントから送られてきた角度をパルスに変換する。

実験結果と分析

図 4 に示した動作を行う場合に必要となった 3 パターンのステップ数と命令を表 5 と 6 と 7 に示す。パターン 1 では

表 5: パターン 1 のステップ数と命令

ステップ数	1	2	3
命令	PIN:0_angle:0	PIN:1_angle:80	PIN:12_angle:60
ステップ数	4	5	6
命令	PIN:2_angle:100	PIN:1_angle:70	PIN:2_angle:95
ステップ数	7	8	9
命令	PIN:14_angle:180	PIN:1_angle:90	PIN:0_angle:180
ステップ数	10	11	
命令	PIN:1_angle:70	PIN:14_angle:0	

図 4 の動作を行うための命令が 11 ステップ必要となった。次にパターン 2 のステップ数と命令の結果を表 6 に示す。

パターン 2 が図 4 の動作を行うための命令が 15 ステップであった。次にパターン 3 のステップ数と命令の結果を表 7 に示す。

パターン 3 では図 4 の動作を行うための命令が 14 ステップであった。パターン 1 を基準としてパターン 2 とパターン 3 の命令の一致率を分析した。パターン 1 とパターン 2 ではパターン 1 のステップ数が 11、パターン 2 のステップ数が 15 であり 7 個が共通で一致率が約 63.64%、パターン 1 とパターン 3 ではパターン 1 のステップ数が 11 でパターン 3 のステップ数が 14 で 7 個が共通で一致率が約 63.64% であった。また、最初と最後に使用する PIN 番号と角度が一致することが分かった。

表 6: パターン 2 のステップ数と命令

ステップ数	1	2	3
命令	PIN:0_angle:0	PIN:1_angle:80	PIN:12_angle:90
ステップ数	4	5	6
命令	PIN:1_angle:70	PIN:12_angle:70	PIN:14_angle:180
ステップ数	7	8	9
命令	PIN:12_angle:80	PIN:1_angle:80	PIN:12_angle:90
ステップ数	10	11	12
命令	PIN:1_angle:90	PIN:0_angle:180	PIN:12_angle:80
ステップ数	13	14	15
命令	PIN:1_angle:70	PIN:12_angle:70	PIN:14_angle:0

表 7: パターン 3 のステップ数と命令

ステップ数	1	2	3
命令	PIN:0_angle:0	PIN:1_angle:100	PIN:2_angle:80
ステップ数	4	5	6
命令	PIN:14_angle:180	PIN:0_angle:60	PIN:1_angle:80
ステップ数	7	8	9
命令	PIN:2_angle:90	PIN:0_angle:120	PIN:1_angle:60
ステップ数	10	11	12
命令	PIN:2_angle:100	PIN:0_angle:180	PIN:1_angle:70
ステップ数	13	14	
命令	PIN:2_angle:90	PIN:14_angle:0	

6. 議論

本稿では、連続した命令をラベル付けすることでユーザが再利用できるようにしたが、ユーザが間違えて入力した命令の連続したパターンに対してもラベル付けをしてしまい、間違っているラベルが再利用される。これに対し、蓄積した命令とラベルをもとにユーザが入力している命令に対し補完機能を付けることを提案する。補完機能では、ユーザが命令を入力する途中で、過去に正常に実行された命令パターンと部分的に一致する場合に、残りの命令列を提示する。これにより、ユーザは正しい命令パターンに誘導され、間違った命令の入力を事前に防ぐことができる。

本稿の提案ではラベルは事前に定義 PIN 番号と角度のルール的一致にもとづいて付与されているが、事前に定義したルールでは”つかむ”, ”置く”, ”戻す”の動作の意味を完璧に反映していない。これに対し、未知の動作パターンが検出された際に管理者に通知してルール追加を促すシステムを導入することを提案する。システムの成長とともにより包括的で実用的なラベル付けが可能になる。このシステムでは、既存のラベルルールに該当しない新しい動作パターンが実行された場合に自動的に管理者にアラートを

送信し、その動作パターンに対して適切な動作の意味を反映したラベルとルールを定義できるようにする。これにより、システムの運用を通じて段階的にルールを拡張し、より実際の動作に即したラベル付けが可能になる。

7. おわりに

課題では命令列が1つのまとまりとして扱われておらず、ラベル付けもされていないため、ユーザは同じ連続した命令を毎回個別に入力する必要がある。その結果、ユーザがロボットアームに与える指示の回数が不必要に増加する。提案では、Client.log から頻出する命令の連続パターンを抽出し定義されたルールをもとにラベル付けを行う。これにより、クライアントからパルス変換サーバー間の命令ステップを削減できる。評価実験ではクライアントからパルス変換サーバー間の命令のステップ数の削減率を評価する。本提案を適用したあとのクライアントからパルス変換サーバー間の命令ステップ数を提案の適用前の命令ステップ数と比較する。基礎実験では白色の箱に入っている9V乾電池を透明の箱に移すことに必要な命令ステップ数を3パターン測定した。パターン1を基準としてパターン2とパターン3の命令の一致率を分析した。パターン1とパターン2ではパターン1のステップ数が11でパターン2のステップ数が15であり7個が共通で一致率が約63.64%、パターン1とパターン3ではパターン1のステップ数が11でパターン3のステップ数が14で7個が共通で一致率が約63.64%であった。また、最初と最後に使用するPIN番号と角度が一致することが分かった。

謝辞

本稿の執筆にあたり、丁寧に指導してくださった東京工科大学大学院バイオ・情報メディア研究科コンピュータサイエンス専攻の平尾 真斗さんに御礼申し上げます。

参考文献

- [1] Du, Z., He, L., Chen, Y., Xiao, Y., Gao, P. and Wang, T.: Robot Cloud: Bridging the power of robotics and cloud computing, *Future Generation Computer Systems*, Vol. 74, pp. 337–348 (online), DOI: <https://doi.org/10.1016/j.future.2016.01.002> (2017).
- [2] Guiochet, J. and Vilchis González, A.: Safety analysis of a medical robot for tele-echography (2002).
- [3] Kim, G., Park, J., Choi, T., Do, H., Park, D. and Kyung, J.: Case studies of a industrial dual-arm robot application, *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 301–302 (online), DOI: 10.1109/URAI.2017.7992735 (2017).
- [4] Do, H. M., Park, C. and Kyung, J. H.: Dual arm robot for packaging and assembling of IT products, *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1067–1070 (online), DOI: 10.1109/CoASE.2012.6386417 (2012).
- [5] Çoban, M. and Gelen, G.: Wireless teleoperation of an industrial robot by using myo arm band, *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, pp. 1–6 (online), DOI: 10.1109/IDAP.2018.8620789 (2018).
- [6] House, B., Malkin, J. and Bilmes, J.: The Voice-Bot: a voice controlled robot arm, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, New York, NY, USA, Association for Computing Machinery, p. 183–192 (online), DOI: 10.1145/1518701.1518731 (2009).
- [7] Madakam, S., Ramaswamy, R. and Tripathi, S.: Internet of Things (IoT): A Literature Review, *Journal of Computer and Communications*, Vol. 3, pp. 164–173 (online), DOI: 10.4236/jcc.2015.35021 (2015).
- [8] Shahparan, M., Klykov, A., Ananth, C., Odilov, A. and Ulug'bekovich, S.: Qualitative Analysis on Smart Warehouse: From Logistics 4.0 Perspective, *2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, pp. 1–6 (online), DOI: 10.1109/ICONSTEM60960.2024.10568677 (2024).
- [9] Autso, S., Kudelina, K., Vaimann, T., Rassõlkin, A. and Kallaste, A.: Principles and Methods of Servomotor Control: Comparative Analysis and Applications, *Applied Sciences*, Vol. 14, No. 6 (online), DOI: 10.3390/app14062579 (2024).
- [10] Yu, Z., Mohammed, A. and Panahi, I.: A review of three PWM techniques, *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, Vol. 1, pp. 257–261 vol.1 (online), DOI: 10.1109/ACC.1997.611797 (1997).
- [11] De Gregorio, D., Tonioni, A., Palli, G. and Di Stefano, L.: Semiautomatic Labeling for Deep Learning in Robotics, *IEEE Transactions on Automation Science and Engineering*, Vol. 17, No. 2, pp. 611–620 (online), DOI: 10.1109/TASE.2019.2938316 (2020).
- [12] Allevalo, A., Short, E., Pryor, M. and Thomaz, A.: Learning Labeled Robot Affordance Models Using Simulations and Crowdsourcing, (online), DOI: 10.15607/RSS.2020.XVI.037 (2020).
- [13] Scherzinger, S., Becker, P., Roennau, A. and Dillmann, R.: Motion Macro Programming on Assistive Robotic Manipulators: Three Skill Types for Everyday Tasks (2023).