

## ヒューマノイドロボットの動作生成に向けた基礎的検討

## Basic Study towards Motion Generation of Humanoid Robot

濱園侑美<sup>\*1</sup> 小林一郎<sup>\*1</sup> 麻生英樹<sup>\*2</sup> 持橋大地<sup>\*3</sup> ムハンマド アッタミミ<sup>\*4</sup>  
 Yumi Hamazono Ichiro Kobayashi Hideki Aso Daichi Mochihashi Muhammad Attamimi  
 中村友昭<sup>\*4</sup> 長井隆行<sup>\*4</sup>  
 Tomoaki Nakamura Tagai Takayuki

\*1お茶の水女子大学

Ochanomizu University

\*2産業技術総合研究所

National Institute of Advanced Industrial Science and Technology

\*3統計数理研究所

The Institute of Statistical Mathematics

\*4電気通信大学

The University of Electro-Communications

Towards ageing society, in the near future, it is convinced that robots will have more opportunities for being active at home. As a requirement for the robots being able to live together with the residents, it is necessary for them to acquire knowledge through the words and gestures by people, and to share the same experience with them. Considering these things, in this paper, we aim to propose a method to associate words with robot's motions so that a robot can properly behave by the instructions given by people, so, we first build a framework for generating robot's motions with its multiple primitive actions, and then make a robot learn the relation between its behaviour and the given words. In particular, we focus on making a robot learn how to cook by giving the instructions containing ambiguous words, 'slightly', 'promptly', etc.

## 1. はじめに

高齢化社会の到来によって、ロボットが家庭内で活躍する場面が増えてくると考えられる。近年では、人の感情を推定しコミュニケーションを行う Pepper などロボットが安価に入手できるようになり、家庭におけるロボットと人のコミュニケーションに大きな進展を見せる可能性がでてきている。このように家庭内にロボットが入ってきた際にロボットと居住者が協調して暮らせる条件として、言葉や身振りなどを使って居住者の経験をロボットに伝え、同じ動作を学習させることが必要になってくると考える。このことを踏まえ、本研究では、ロボットが人と同じ動作を真似て行うことができるように自らが行える基本動作を組み合わせてひとつの行為を生成する枠組みを構築し、複数の基本動作から複雑な行為を表現する対応関係を明確にすること、及び言葉と動作の対応関係を学習することによって初めて行う動作であっても言葉の意味から推測することを目的とする。とくに、対象とする行為を調理動作とし、基本動作からなる行為の具体的な関係について考察を行う。

## 2. 提案手法

いくつかの言葉の意味と動作表現の対応関係が既知であるとする。この時、動作と対応関係が分からない未知の言葉が与えられた際に、他の言葉との意味的な関係から対応する動作を推定する手法を提案する。図1に提案手法の概要を示す。

動作を表現空間へ配置する方法は Activity-Attribute Matrix を動作生成に適した形に改良したものを新たに提案し、言葉を意味空間へ配置する方法は word2vec [2] を用いる。また、言葉と動作の対応関係学習にはニューラルネットワークを用いる。

連絡先: 濱園侑美, お茶の水女子大学理学部情報科学科, 東京都文京区大塚 2-1-1 理学部 3 号館 506, 0359785708, gl020528@is.ocha.ac.jp

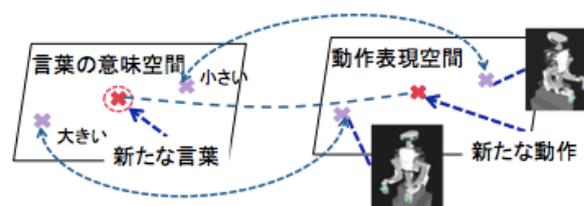


図1: 提案手法の概要

## 3. ロボットの動作

## 3.1 ヒューマノイドロボットの概観

使用するロボットは(株)川田工業社製ヒューマノイドロボット HIRO を用いる。(図2) \*1.

HIRO は首の左右上下運動に2つ, 体全体を動かす1つ, 両腕に各6つ, 両手に各4つの関節を持つ。それぞれの関節角と時間  $t$  を指定することで,  $t$  秒かけて指定された角度へと関節を動かすことが可能である。表1にそれぞれの関節が動く範囲を示す。

## 3.2 動作構成

ロボットの調理動作を関節軸の基本動作から構成するために, Cheng ら [1] による Activity-Attribute Matrix(AAM) を参考にする。AAM は動作と動作に関連している意味属性を符号化したものであり,  $M$  を Activity (活動),  $N$  を Attribute (属性) とし, 各要素  $a_{ij}(i \in M, j \in N)$  において Attribute の Activity への含有関係について Activity  $i$  を構成するのに Attribute  $j$  が用いられている場合は1, 用いられていない場合は0とする,  $M \times N$  行列により動作を表現する。本研究では, Activity として調理動作を例に捉え, それに伴い Attribute と

\*1 <http://robot-support.kawada.jp/support/hiro/> より引用

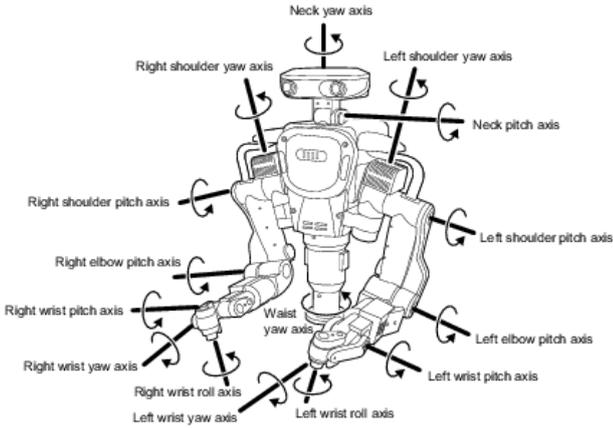


図 2: ロボットの関節

表 1: 関節の動作可能範囲

関節軸名称	記号	可動範囲 (deg)	最大速度 (deg/s)
腰ヨー軸	CY	-163 to +163	130
首ヨー軸	NY	-70 to +70	150
首ピッチ軸	NP	-20 to +70	300
右肩ヨー軸	RSY	-88 to +88	172
右肩ピッチ軸	RSP	-140 to +60	133
右肘ピッチ軸	REP	-158 to +0	229
右手首ヨー軸	RWY	-165 to +105	300
右手首ピッチ軸	RWP	-100 to +100	223
右手首ロール軸	RWR	-163 to +163	300
左肩ヨー軸	LSY	-88 to +88	172
左肩ピッチ軸	LSP	-140 to +60	133
左肘ピッチ軸	LEP	-158 to +0	229
左手首ヨー軸	LWY	-105 to +165	300
左手首ピッチ軸	LWP	-100 to +100	223
左手首ロール軸	LWR	-163 to +163	300

して右手の指先を前後に動かす ( $x$ ), 左右に動かす ( $y$ ), 上下に動かす ( $z$ ) 基本ベクトルを設定した. また, ロボットを実際に動かすにはそれぞれの関節角を指定する必要がある. そこで, Attribute の  $x, y, z$  それぞれに対して, 係数となるベクトルとの内積をとることにより, ロボットの動作生成を可能にする. さらに本研究では Activity を生成する過程において, それぞれの Attribute の度合い, 時系列性, 速度等が重要となるため,  $x, y, z$  の変動割合と時間  $t$  を組み合わせた  $[p_x, p_y, p_z, t]$  を時系列に  $n$  個並べた  $[[p_{x_1}, p_{y_1}, p_{z_1}, t_1], [p_{x_2}, p_{y_2}, p_{z_2}, t_2], \dots, [p_{x_n}, p_{y_n}, p_{z_n}, t_n]]$  を与えることにより動作の生成を可能にする.

表 2 に AAM を改良した, 本研究で提案する「時系列対応 AAM」を示す. 次章にて表 2 がどのように使用されるかについて具体例と共に示す.

#### 4. 動作生成実験

ロボットの動作生成の手順を以下に示す.

1. 基本ベクトルの準備: 要素動作となる  $x, y, z$  方向の基本ベクトルを準備する
2. 係数ベクトルの準備: 1. で準備したベクトル方向にどの程度移動させるかを示すベクトル (係数ベクトル) を生成する

表 2: 時系列対応 AAM の概要

	0	0	-5	1	0	2	5	1	0	2	5	1	0	2	5	1
速く切る	0	0	-5	5	0	2	5	5	0	2	5	5	0	2	5	5
ゆっくり切る	0	0	-5	2	0	4	5	2	0	4	5	2	0	4	5	2
粗く切る	0	0	-5	2	0	1	5	2	0	1	5	2	0	1	5	2
細かく切る	0	0	-5	5	0	4	5	5	0	4	5	5	0	4	5	5
ざっくり切る	0	0	-5	5	0	4	5	5	0	4	5	5	0	4	5	5
	$p_{x_1}$	$p_{y_1}$	$p_{z_1}$	$t_1$	$p_{x_2}$	$p_{y_2}$	$p_{z_2}$	$t_2$	...	$p_{x_n}$	$p_{y_n}$	$p_{z_n}$	$t_n$			
	$x$	$y$	$z$		$x$	$y$	$z$			$x$	$y$	$z$				
CY	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
NY	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
NP	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
RSY	0.1	1.8	0.7		0.1	1.8	0.7			0.1	1.8	0.7				
RSP	-2.3	0.7	0.1		-2.3	0.7	0.1			-2.3	0.7	0.1				
REP	2.1	-0.5	-2.7		2.1	-0.5	-2.7			2.1	-0.5	-2.7				
RWY	0.0	0.0	0.1		0.0	0.0	0.1			0.0	0.0	0.1				
RWP	0.2	0.2	2.7		0.2	0.2	2.7			0.2	0.2	2.7				
RWR	0.0	-1.8	0.0		0.0	-1.8	0.0			0.0	-1.8	0.0				
LSY	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
LSP	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
LEP	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
LWY	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
LWP	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
LWR	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
RH1	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
RH2	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
RH3	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
RH4	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
LH1	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
LH2	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
LH3	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				
LH4	0.0	0.0	0.0		0.0	0.0	0.0			0.0	0.0	0.0				

3. 基本ベクトルと係数ベクトルの内積: 1. の基本ベクトルと 2. で生成された係数ベクトルの内積を計算し, 関節角座標 (以下, 「座標」と呼ぶ) を求める
4. 移動先座標の取得: 3. で計算された座標に現在の座標を加えて, 移動先の座標を取得する. ロボットに与える座標のフォーマットは以下の様になる.

[[CY, NY, NP], [LSY, LSP, LEP, LWY, LWP, LWR], [RSY, RSP, REP, RWY, RWP, RWR], [RH1, RH2, RH3, RH4], [LH1, LH2, LH3, LH4]]

具体例を上記の順に従って以下に示す.

1. 基本ベクトルの準備  
準備される基本ベクトルは, 表 2 の  $x, y, z$  に示される値となる.
2. 係数ベクトルの生成  
係数行列の生成を調理動作「切る」を例に説明する, 切る動作は常に  $x$  が 0 で (前後の動きはなく), まず  $z$  が負 (下へ) の動きをし, 次に  $z$  が正 (上へ) の動きをしながら  $y$  が正 (左へ) 動く, という特徴から生成される. このとき, 動作の違いは特に  $y$  (左右の動き) と時間  $t$  によって種類を生成できる. 表 2 を見ると, 複数の種類の「切る」という動作が,  $y$  と  $t$  の値によって差別化されていることがわかる. 次に, 「速く切る」という動作を見てみると,  $[[p_{x_1}, p_{y_1}, p_{z_1}, t_1], [p_{x_2}, p_{y_2}, p_{z_2}, t_2], \dots, [p_{x_n}, p_{y_n}, p_{z_n}, t_n]]$  の値は  $[[0, 0, -5, 1], [0, 2, 5, 1], \dots, [0, 2, 5, 1]]$  となっており, 時間の経過と共にその動作を示す係数ベクトルが表示されていることがわかる.

3. 基本ベクトルと係数ベクトルの内積, 移動先座標の取得および速度の追加  
表 2 中, 「速く切る」における最初の係数行列は,  $[p_{x_1}, p_{y_1}, p_{z_1}, t_1] = [0, 0, -5, 1]$  であり, それと基本ベクトルの内積の結果に現在の座標を加え, さらに動作の速度  $t_1$  を加えた結果, 以下のようになる.

[[[0.0, 0.0, 0.0], [-4.1, -0.4, -86.4, 14.7, -3.9, 3.2], [0.6, 0.1, -99.9, -15.2, 9.6, -3.2], [0.0, 0.0, 0.0, 0.0], [0.0, 0.0, 0.0, 0.0]], 1]

この動作生成を引き続き  $[p_{x_2}, p_{y_2}, p_{z_2}, t_2] = [0, 2, 5, 1]$  にも適用した結果のロボットの動作を図3に示す。

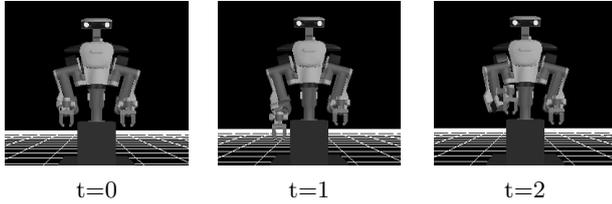


図3: 「速く切る」の動作例

## 5. 言語の分散意味表現

Mikolov ら [2] によって提案された word2vec は、単語をベクトルで表現し、同じ文脈の中にある単語はお互いに近い意味を持つように単語をベクトル化して表現する定量化手法である。これにより、単語の意味はベクトル計算によって表現可能となり、例えば ‘king’-‘man’+‘woman’ (「王様から男を引いて女を足したもの」) を求める計算をすると、単語 ‘queen’ (「女王」) のベクトルに近い値が得られる、といった単語の演算が可能になる。本研究ではこれを利用し、単語の意味関係から、未知の単語に対する動作の推定を可能にする。

## 6. 言語と動作の関係学習

言語と動作の関係を学習するために、3層の階層型ニューラルネットワーク (NN) を使い、学習方法としてはバックプロパゲーションを用いた。また活性化関数はシグモイド関数を用いた。言語は word2vec によって 50 次元の分散意味表現で表した。動作は「切る」の様々なパターンを 3.2 節に示したで時系列対応 AAM 作成し、 $t_1$  秒と  $t_2$  秒の2つの動作を合わせて 8 次元のベクトルで表した。これにより入力次元を 50、出力次元数を 8 とし、中間層の数をいくつか設定し、教師信号との平均二乗誤差を調べることで収束の良かった中間層を 13 とする NN を構築した。訓練データとして、4 つの言語表現 (大きい, 小さい, 速く, ゆっくり) に対して各 10 個の計 40 個を与えた。

学習をした後、評価データとして 4 つの言語表現を入力したところ、表 3 に示す結果になった。

表 3: 評価結果と訓練データとの比較

言語表現	訓練データ	評価結果	誤差
大きく	[0,0,-5,3],[0,5~8,5,3]	[0,0,-5,3],[0,7,5,3]	0.02
小さく	[0,0,-5,3],[0,0.4~2,5,3]	[0,0,-5,3],[0,2,5,3]	0.04
速く	[0,0,-5,1~2],[0,3,5,1~2]	[0,0,-1,3],[0,7,8,3]	0.14
ゆっくり	[0,0,-5,5~8],[0,3,5,5~8]	[-4,6,2,4],[2,7,9,2]	0.29

なお、ここでの「誤差」は訓練データとして与えた動作の平均と、評価結果として出てきた数値の平均二乗誤差である。表 3 の結果から「大きく」と「小さく」の識別は上手く行っている

ことが確認された。「速く」において、「切る」の特徴である「常に  $x$  が 0 で、まず  $z$  が負、次に  $z$  が正と  $y$  が正」、という動作はきちんと反映されていることが確かめられるが、「速く」の特徴である  $t$  の値が小さくなっておらず、識別がされていないと言える。「ゆっくり」については、「切る」動作の特徴すら現れていない。

また、未知の言語表現として「厚い」と「細い」を入力とした時、その予測される動作との比較において表 4 に示す結果を得た。

表 4: 結果と予測データとの比較

言語表現	予測データ	結果	誤差
厚い	[0,0,-5,3],[0,4~6,5,3]	[0,0,-5,3],[0,7,5,3]	0.03
細い	[0,0,-5,3],[0,0~1,5,3]	[0,-1,-5,3],[0,7,5,3]	0.12

表 4 の結果より、「厚い」において「切る」の動きの特徴も「厚い」の特徴である  $y$  の値が大きくなっており、対応関係の学習が上手く行った。しかし「細い」では、「切る」動作の特徴は捉えられているものの、「細い」の特徴である  $y$  の値が小さくなっておらずは見られなかった。実際にシミュレータで結果を入力したところ、「厚く」切る動作は出来ていたものの、「細く」切る動きは予想よりも大きく  $y$  の正方向に動いた。これらより、「切る」動作の特徴はある程度捉えられているものの、各言語表現の特徴は捉えきれていないことが分かる。理由として、基本的にはデータ数が少ないこと、また、分散意味表現を入力として与えているので、動作としては同じ方向だが、言葉の意味としては極性が異なるものを与えたことによりモデルをきちんと学習できなかった可能性が挙げられる。

## 7. まとめ

本研究では、ロボットの動作生成を行うため調理動作生成を例に動作生成に対応するため、Cheng らによる Activity-Attribute Matrix を参考にして動作要素の時系列変化を捉える関係行列を作成した。作成した時系列対応 AAM を使い、シミュレータを利用してロボットが意図する動作を行えていることを確認した。また、言語と動作の対応関係を取るため、言語は word2vec を使った分散意味表現をとり、それと動作のベクトルによるニューラルネットワークを用いた学習を行った。今後の課題としては、言語と動作の対応関係をより明確にし、さらに多種多様な言語から動作生成を可能にすることである。

## 謝辞

本研究は、JSPS 科研費 26280096 の助成を受けて実施した。

## 参考文献

- [1] Heng-Tze Cheng, Feng-Tso Sun, Martin Griss, Paul Davis, Jianguo Li, Di You, “NuActiv: Recognizing Unseen New Activities Using Semantic Attribute-Based Learning”, MobiSys’13, Taipei, 2013.
- [2] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, “Efficient Estimation of Word Representations in Vector Space”, CoRR, abs/1301.3781, 2013