

# 潜在空間上での人の動作識別に基づく ロボットの動作生成への取り組み

An Approach to Humanoid Robot Motion Generation  
based on the Identification of Human Motion on Latent Space.

大山まりほ<sup>\*1</sup>  
Mariho Ohyama

小林一郎<sup>\*2</sup>  
Ichiro Kobayashi

<sup>\*1</sup>お茶の水女子大学理学部情報科学科  
Dept. of Information Science, Ochanomizu University

<sup>\*2</sup>お茶の水女子大学基幹研究院自然科学系  
Graduate School of Humanities and Sciences, Ochanomizu University

In the near future, it is expected that robots are used at residential environments and are required to learn the residents' experience in their everyday lives by observing and mimicking residents' behavior. Considering this, as the first step for achieving the requirement, we aim to make a robot mimic human motion obtained through Kinect camera as visual information. To achieve this, we have introduced Gaussian Process Latent Variable Models (GPLVM) to reduce the high dimension of human kinematic motion to lower latent dimension of the motion. We have further introduced Multilayer perceptron to the framework of GPLVM for better convergence of the algorithm, and then obtained better latent variables rather than the original algorithm of GPLVM. As robot motion generation, we have employed Rviz and MoveIt! simulator.

## 1. はじめに

将来、家庭にロボットが導入されると予想される。ロボットが人と共存するためには、ロボットは人から与えられる言語や視覚などの複数の情報に基づいてタスクを学習する必要がある。このことから本研究では、その前段階としてロボットに人の行動を正しく認識させることを目指し、ロボットに視覚情報から得た人の行動を真似させ同じ動作をさせることを目的とする。ロボットに人の動作を真似させる手法として、高次元のデータから低次元のデータへの次元圧縮を行う Gaussian Process Latent Variable Models (GPLVM) を用いる。GPLVM のオリジナルのアルゴリズムでは、収束判定や学習における最適化の方法において改善の余地があり、本研究では、Multilayer perceptron (MLP) を採用することにより、次元圧縮の質を向上させる。また、ロボットの動作生成は、Robot Operating System (ROS) を用いて、MoveIt! というシミュレータ上で実装する。

## 2. 潜在空間を用いた人の動作識別

Kinect を用いて人の関節の動きを追跡し、人の動作を捉える。高次元となる人の関節のデータをそのままロボットに識別させるのは計算量が多くなるため、本研究では高次元の人の動作データを低次元な潜在空間への圧縮を行い、その潜在空間上で人の動作を識別する。人の動きは滑らかなものなので複数の観測点による高次元なガウス過程とみなすことができる。本研究では、GPLVM (Gaussian Process Latent Variable Models) [1] を用いてガウス過程で表現される人の動作の時系列データを潜在空間に圧縮する。GPLVM とは、確率的主成分分析にガウス過程を導入したものである。GPLVM を使用した次元圧縮の概要図を図 1 に、アルゴリズムを Algorithm 1 に示す。

観測データ全体を  $\mathbf{Y}$ 、圧縮先の潜在空間上のデータ全体を  $\mathbf{X}$  とする。まず PCA (Principal component analysis) を用いて  $\mathbf{Y}$  を圧縮し、 $\mathbf{X}$  の初期値を設定する。そして  $\mathbf{X}$ 、 $\mathbf{Y}$  それぞれをガウス過程に従っていると仮定した正規化を行う。その後、



図 1: GPLVM の概要図

### Algorithm 1 GPLVM

Require:  $\mathbf{X}, \mathbf{Y}, \mathbf{K}, T$

Initialize  $\mathbf{X}$  through PCA.

for  $T$  iterations do

Optimise (1) with respect to the parameters of  $\mathbf{K}$  using scaled conjugate gradients.

Optimise  $p(\mathbf{Y}|\mathbf{X}, \alpha, \beta, \gamma)$  with respect to  $\mathbf{X}$

end for

以下の式 (1) で表される  $p(\mathbf{Y}|\mathbf{X}, \beta)$  の対数尤度が最大となるよう式 (2) で表されるカーネル関数のパラメータ  $\alpha, \beta, \gamma$  を更新する。更新したパラメータを当てはめ、 $\mathbf{X}$  を再設定する。

$$p(\mathbf{Y}|\mathbf{X}, \beta) = \frac{1}{(2\pi)^{DN/2} |\mathbf{K}|^{D/2}} \exp\left(-\frac{1}{2} \text{tr}(\mathbf{K}^{-1} \mathbf{Y} \mathbf{Y}^T)\right) \quad (1)$$

$$k(x_n, x_m) = \alpha \exp\left(-\frac{\gamma}{2} (x_n - x_m)^T (x_n - x_m)\right) + \delta_{nm} \beta^{-1} \quad (2)$$

カーネルのパラメータの更新を  $T$  回繰り返すことで  $\mathbf{X}$  が更新され、高次元の  $\mathbf{Y}$  に対して低次元の潜在空間  $\mathbf{X}$  が求まる。

しかし、このアルゴリズムでは予め定めた回数  $T$  回  $\mathbf{X}$  の更新を繰り返しており、収束判定を行っていない。そのため最適な  $\mathbf{X}$  が求められた保証がされていない。そのため収束判定を行うために MLP (Multilayer perceptron) を組み込み、収束判定をするアルゴリズムを作成した。MLP を組み込んだプログラムの概要図を図 2 を Algorithm 2 に示す。

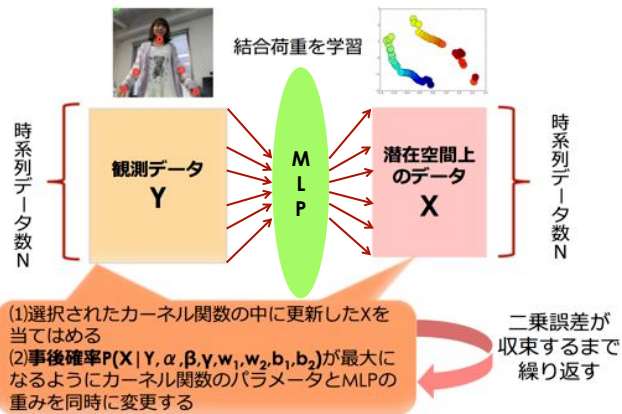


図 2: GPLVM with MLP の概要図

### Algorithm 2 GPLVM with MLP

Require:  $\mathbf{X}, \mathbf{Y}, \mathbf{K}$

Initialize  $\mathbf{X}$  through PCA.

while Mean squared error  $\geq \epsilon$  do

Select a new  $\mathbf{X}$  using the MLP algorithm.

Optimise (1) with respect to the parameters of  $\mathbf{K}$  using scaled conjugate gradients.

Optimise  $p(\mathbf{Y}|\mathbf{X}, \alpha, \beta, \gamma)$  with respect to  $\mathbf{X}$  using MLP.

end while

まず GPLVM のみの場合と同様に PCA を用いて  $\mathbf{Y}$  を圧縮し、 $\mathbf{X}$  の初期値を設定する。そして  $\mathbf{X}, \mathbf{Y}$  それぞれをガウス過程に従っていると仮定した正規化を行う。次に MLP を用いる。  $\mathbf{Y}$  を入力、  $\mathbf{X}$  を出力として結合荷重を求め、  $\mathbf{X}$  と  $\mathbf{Y}$  の対応関係を学習させておく。その後、以下の式 (1) で表される  $p(\mathbf{Y}|\mathbf{X}, \beta)$  の対数尤度が最大となるよう、式 (2) で表されるカーネル関数のパラメータ  $\alpha, \beta, \gamma$  を更新する。次に、勾配法を用いて MLP の結合荷重の更新を行い、  $\mathbf{X}$  を再設定する。結合荷重の更新とカーネルのパラメータの更新を最小二乗誤差が収束するまで繰り返すことで  $\mathbf{X}$  が更新され、高次元の  $\mathbf{Y}$  に対して低次元の潜在空間  $\mathbf{X}$  が求まる。

## 3. ロボットの動作生成

### 3.1 ヒューマノイドロボット HIRONXC

人の動作を真似させるロボットには、川田工業社製ヒューマノイドロボット HIRONXC を用いる。HIRONXC は首元に 2 つ、両腕にそれぞれ 6 つ、両手にそれぞれ 4 つ、体全体を動かす関節が 1 つと、総計 23 個の関節を持っている (図 3)。動かす部位と、各関節の  $x, y, z$  軸の座標値を設定することで動作生成を行う。今回は実機には接続せず、ヒューマノイドロボット HIRONXC のシミュレータ MoveIt! を動作させた。

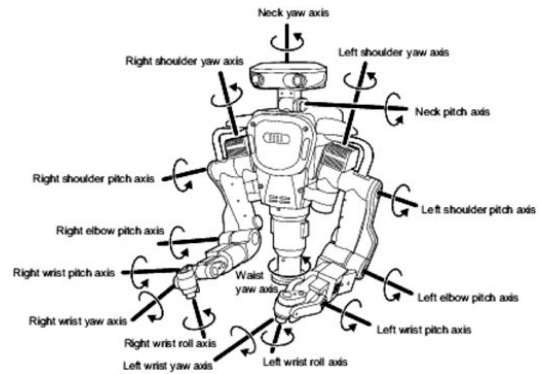


図 3: HIRONXC の関節情報

### 3.2 Rviz を用いた動作設定

Rviz と呼ばれるロボットの動作を視認しながら設定することが出来るツールを用いてロボットに目標の動きの設定を行った (図 4)。Rviz 上で、ロボットに目標のポーズを設定しプランニングと実行を行う。図 5 に示すように目標のポーズの各関節の角度や、  $x, y, z$  軸の座標値を取得することが出来る。Rviz を用いた目標のポーズの座標値の取得を図 5 に示す。

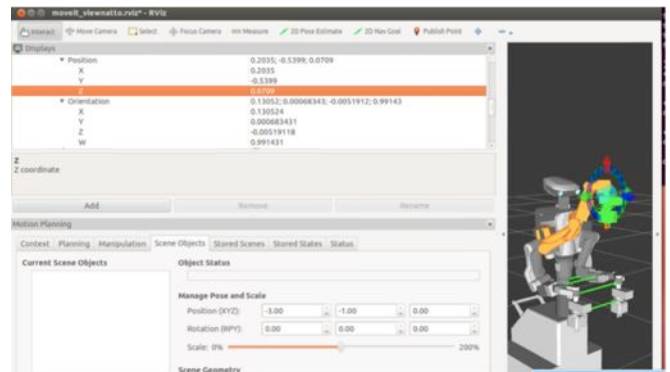


図 4: Rviz 上で動きをプランニングしている様子



図 5: Rviz 上での座標値の取得

### 3.3 MoveIt! を用いた動作生成

ロボット開発のためのソフトウェアの集合である ROS (Robot Operating System) を用いシミュレータ MoveIt! 上で動作生成を行った。図 5 で取得した動作パラメータ (関節の座標値) を入力することでロボットはシミュレータ上で目標の動きをすることが出来る (図 6)。

## 4. 実験

2 節で述べたアルゴリズムに従い人の動作データの潜在空間への圧縮を行った。

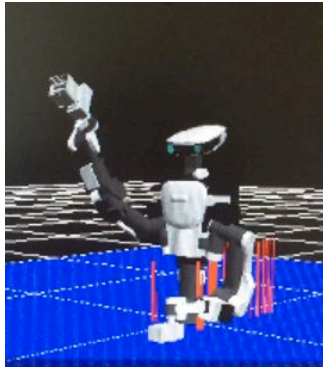


図 6: ロボットが右手を上げている様子

#### 4.1 実験設定

人の動作のデータとして「右手を上げる」「右手を下げる」「左手を上げる」「左手を下げる」の4つの動作を Kinect を用いて撮影した動画の時系列データを使用する。関節は右手首・左手首・右肘・左肘・肩の中心の5つ、それぞれの  $x, y, z$  軸の座標値を取得する。今回の実験では計 15 次元のデータを二次元の潜在空間に圧縮した。圧縮の際に用いるカーネルには RBF カーネル (式 (2)) を用いた。

#### 4.2 実験結果

図 7~10 は上記 4 つの動作に対する潜在空間上での表現である。紺色を始めとして、時間に沿って赤色までグラデーションで変化するようにした。また各図において左は GPLVM に MLP を用いて次元圧縮したもの、右は用いないものを示す。

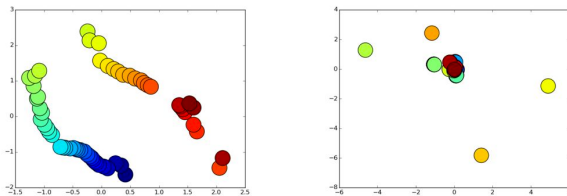


図 7: 右手を上げる (左: MLP 有 右: MLP 無)

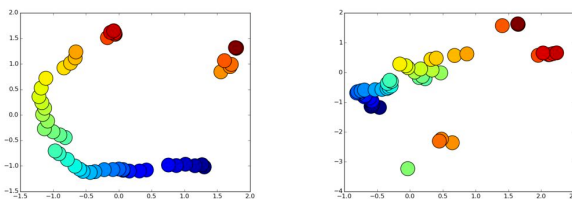


図 8: 右手を下げる (左: MLP 有 右: MLP 無)

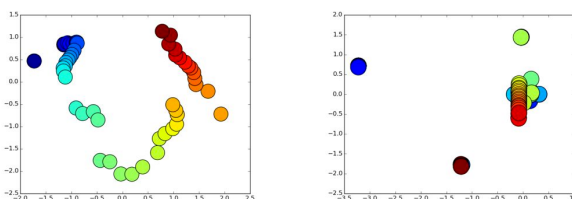


図 9: 左手を上げる (左: MLP 有 右: MLP 無)

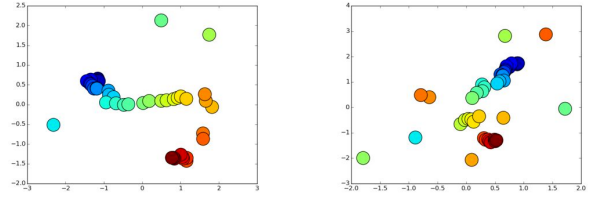


図 10: 左手を下げる (左: MLP 有 右: MLP 無)

#### 4.3 考察

Lawrence[1] によって提案されたオリジナルのアルゴリズムでは、カーネル関数  $K$  のパラメータを求める際の  $X$  の初期値およびイテレーションの回数によって得られる  $X$  がガウス過程に従っているか定かではないケースが多く観測された。そこで本研究では、GitHub 上で公開されているソースコード\*1を参考に、対数尤度の収束を観測できる MLP を採用し、高次元の動作データを低次元に圧縮し良い結果を得ることができた。図 7~10 において、4 つの動作全てが滑らかで繋がった動きをしているため二次元に圧縮した場合でも連続な曲線が描かれていることによりデータが潜在空間上でガウス過程に従っていることがわかる。一方で、同じ動作の動画を複数個撮影し実験を行ったところ、同じ動作の潜在空間上での表現において結果は初期値に依存するため再現性がないことも分かり、潜在空間上での人の動作識別は容易ではないことが分かった。

### 5. まとめと今後の課題

本研究においては、人の動作データに基づくヒューマノイドロボット HIRO の行動学習・行動生成のための検討を行った。Kinect を用いて取得した人の関節の情報を利用し、計 15 次元の人の動作データを 2 次元の潜在空間に圧縮、視覚化するプログラムおよびロボットの動作生成プログラムを作成した。GPLVM を用いた次元圧縮では、潜在データの初期値によって結果が大きく異なることを確認し、人の動作識別は困難であることが分かった。今後の課題として、shared GPLVM[2] や GPDM(Gaussian Process Dynamical Models)[3] を用いて動作識別が可能な次元圧縮の手法についての再検討を行うと共に、識別した動作を実際にロボットに結びつけ動作を生成させる。

#### 謝辞

本研究は、科学研究費補助金 (15K12104) の支援を受けた。

#### 参考文献

- [1] Neil D.Lawrence“Gaussian Process Latent Variable Models for Visualisation of High Dimensional Data”, 2004.
- [2] Carl Henrik Ek“Shared Gaussian Process Latent Variable Models”, (PhD thesis),Oxford Brookes University,2009.
- [3] Jack M.Wang, David J.Fleet, Senior Member, IEEE, and Aaron Hertzmann, Member, IEEE“Gaussian Process Dynamical Models for Human Motion”, IEEE Transactions on Pattern Analysis and Machine Intelligence,Vol.30,No.2,2008.
- [4] 持橋大地 “ガウス過程の基礎と教師なし学習”, 統計数理研究所公開講座資料,2015.

\*1 <http://github.com/jameshensman>