Unsupervised phase learning and extraction from repetitive movements

Jatesiktat, Prayook; Anopas, Dollaporn; Ang, Wei Tech

2018

Jatesiktat, P., Anopas, D., & Ang, W. T. (2018). Unsupervised phase learning and extraction from repetitive movements. Proceedings of 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 227-230.

doi:10.1109/embc.2018.8512196

<https://hdl.handle.net/10356/137348>

<https://doi.org/10.1109/EMBC.2018.8512196>

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at:
<https://doi.org/10.1109/EMBC.2018.8512196>

Downloaded on 26 Apr 2023 14:05:33 SGT

Unsupervised Phase Learning and Extraction from Repetitive Movements*

Prayook Jatesiktat¹, Dollaporn Anopas¹, and Wei Tech Ang¹

Abstract—Phase extraction from repetitive movements is one crucial part in various applications such as interactive robotics, physical rehabilitation, or gait analysis. However, pre-existing automatic phase extraction techniques are specific to a target movement due to some handcrafted-features. To make it more universal, a novel unsupervised-learning-based phase extraction technique is proposed. A neural network architecture and a cost function are designed to learn the concept of phase from records of a repetitive movement without any given phase label. The method is tested on a rat's gait cycle and a human's upper limb movement. The phases are successfully extracted at the sample level despite the variations in movement speed, trajectory, or subject's anthropometric features.

I. INTRODUCTION

Movements of living things are challenging to model due to their large variations in anatomy, subject conditions, or even the state of the environment. The most studied movements are in the group of repetitive tasks such as bipedal gait, quadrupedal gait, athletic or rehabilitative exercises. Modeling a particular piece of movement usually needs an expert to observe the movements and to engineer a heuristic way to extract key characteristics from motion capture data to make it useful. This study will demonstrate that some parts of such time-consuming process can be done automatically with machine learning.

One fundamental feature of repetitive movements is phase. By knowing phase at all sample points in a movement record, a lot more information such as repetition count, frequency, and average movement trajectory can also be extracted. However, modeling a movement for phase extraction is not a trivial task because a movement can have variations in speed and trajectory.

There are a few common techniques to assign phase value to every sample in the record. The simplest way is to manually assign phases to at least one sample per data period and fill in the gaps with linear or non-linear interpolation. However, this method is not scalable with large data size. To make it more automatic, specific feature detection algorithms can be designed to assign some pre-known phases associated with the features. For example, in a rat gait analysis, the direction of horizontal velocity is used to indicate the beginning of the stance phase [1], and in a human gait analysis, the signature of a heel strike can be

used to identify the transition point from a swing phase to a stance phase [2]. Nonetheless, those features are usually designed specifically for each movement, and they need to be redesigned for every new type of movement.

To automate the gap filling process, one common technique is to use linear interpolation. However, it is inaccurate in the situation where phase progressions in the movement are inconsistent. Therefore, dynamic time warping (DTW) is introduced to deal with the inconsistency in the phase progression. DTW has become a common technique in human movement research to align multiple trials of a movement [3]. Still, the need of manually-segmented template prevents the method from being fully automated.

The cross-correlation between the template signal and a window of an input signal can also be used as a tool to determine the phase lag [4]. Besides the need of template, this technique will not work well if the frequency of the input signal is different from those of the template. Such scenario is common among movements of living things.

In this study, an unsupervised learning method is designed to automatically model a repetitive movement and use that model to extract phase at the level of each individual sample. As the method is unsupervised, the training data of the target repetitive movement do not need to be labeled with phase. The learning algorithm only needs one or more continuous records of the target repetitive movement without the need of time-consuming pre-segmentation.

II. METHOD

A. Method Overview

The method contains 2 steps, a learning step and an inferencing step. First, a neural network will be trained with high-dimensional time-series data recorded from one or more records of the target movement. As the data do not come with phase, a special cost function is designed to force the network to organize itself and learn the concept of phase. Each pair of 2 consecutive samples will constrain the network not to go backward in time for a tiny region in the network input space. Collectively, with a large enough number of pairs in the training data, the network will eventually be able to assign a reasonable phase to the training data. After the learning, the network will be able to infer a phase by feed-forwarding the movement data through the trained network.

B. Neural Network Architecture

The network architecture is shown in Fig. 1. The input layer is a pre-processed $2K$ -dimensional vector concatenated from K -dimensional state $\mathbf{p}$ and K -dimensional heading

¹Prayook Jatesiktat, Dollaporn Anopas, and Wei Tech Ang are with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, 639798, Singapore. prayook001@e.ntu.edu.sg; anop0001@e.ntu.edu.sg; wtang@ntu.edu.sg

*This work was supported by the first and the second Rehabilitation Research Grants (RRG1/16004 and RRG2/16001) from Rehabilitation Research Institute of Singapore (RRIS).

Fig. 1: Neural network architecture. The output from the feed-forwarding process will always land on the unit circle which represent phase of the input sample.

direction $\mathbf{h}$ (Section II-C). There are 3 fully connected hidden layers with hyperbolic tangent activation function. Each hidden layer contains H hidden nodes. All nodes in the last hidden layer are fully-connected to 2 nodes in the pre-phase layer with linear activation function. The result in the pre-phase layer will be normalized to a 2D unit vector at the output layer.

The feed-forwarding function $\mathbf{U}(\Theta, s) = [x \ y]^T$ can be seen as a transformation from the $2K$ -dimensional input layer to a 2D point on a unit circle. This phase representation is chosen to avoid the wrapping point (0 and 2π) in the 1D phase representation that usually confuses the neural network. However, to simplify plots in this paper, the conversion $\phi = \arctan 2(y, x) + \pi$ is done only for the visualization purpose.

C. Training Data Preparation

The training data will be standardized so that all K dimensions have the mean of zero and the standard deviation of $\frac{1}{3}$. At each transformed sample $\mathbf{p}_t$, the surrounding samples $\mathbf{p}_{t-r}, \dots, \mathbf{p}_{t+r}$ are used to perform a linear regression independently in each dimension to extract slopes. Those slopes are combined as a vector and normalized as a heading direction $\mathbf{h}_t$. This heading direction is a good clue for a neural network to recognize phases in a movement [5].

All pairs of two-consecutive samples will be packed together as $\langle \bar{s}_i, \hat{s}_i \rangle$ where $\bar{s}_i = [\mathbf{p}_i^T \ \mathbf{h}_i^T]^T$ is from the earlier sample and $\hat{s}_i = [\mathbf{p}_{i+1}^T \ \mathbf{h}_{i+1}^T]^T$ is from the later sample.

D. Cost Function

In the training, all the weights and biases parameters Θ are searched to minimize a cost function defined as

$$F(\Theta) = \frac{1}{N} \sum_{i=1}^N \max(0, \hat{x}_i \bar{y}_i - \bar{x}_i \hat{y}_i + \epsilon) + \left(\frac{1}{N} \sum_{i=1}^N \bar{x}_i \right)^2 + \left(\frac{1}{N} \sum_{i=1}^N \bar{y}_i \right)^2 + \frac{1}{N} \sum_{i=1}^N \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{|\mathbf{V}(\Theta, \bar{s}_i)|^2}{2\sigma^2}} \quad (1)$$

where $\mathbf{U}(\Theta, \bar{s}_i) = [\bar{x}_i \ \bar{y}_i]^T$ and $\mathbf{U}(\Theta, \hat{s}_i) = [\hat{x}_i \ \hat{y}_i]^T$ are the feed-forwarding calculations from data pair i . The function $\mathbf{V}(\Theta, \bar{s}_i)$ is a 2D activation vector from the pre-phase layer.

The first *clockwise penalty* term will activate when the direction of movement at the output layer from the earlier sample $\mathbf{U}(\Theta, \bar{s}_i)$ to the later sample $\mathbf{U}(\Theta, \hat{s}_i)$ is moving in the clockwise direction. The term $\hat{x}_i \bar{y}_i - \bar{x}_i \hat{y}_i$ is from the Z-component of a cross product $\mathbf{U}(\Theta, \hat{s}_i) \times \mathbf{U}(\Theta, \bar{s}_i)$. The positive small margin ϵ is added to force 2 consecutive samples to have some gap on the unit circle space.

The next two terms calculate the centroid from all samples in the training data and measure the square distance from the origin (center of the unit circle). These terms reflect how well the outputs are distributed on the unit circle. In the case that the outputs are not evenly distributed, the centroid will shift away from the center, and the network will be penalized.

The last *singularity penalty* term is designed to prevent the pre-phase layer from staying too close to the origin. This is the area where the phase output can be unstable especially with noisy inputs.

E. Initialization & Optimization

All the weights are initialized randomly from a Gaussian distribution with zero mean and standard deviation of $1/\sqrt{n}$ where n is the number of nodes in the previous layer of that weight. All the biases are initialized as zeros. Adam optimization algorithm [6] with default parameters on TensorFlow is used to minimize the cost function (Eq.1) for 50,000 iterations.

III. EVALUATION

The phase learning and extraction method are tested in 2 different applications. The first one is about robotic rehabilitation in rats where the robot needs to know the phase of the rat's forelimb gait in order to actuate/assist the rat's affected hind-limb in a synchronous manner. The second one is about an analysis of human upper limb exercise movement.

A. Rat's Forelimb Gait Cycle

In a record, a rat with color markers will walk on a treadmill. An Intel RealSense R200 camera is used to record multiple RGB images at 60 frames per second. Positions in 2D of two color markers on its scapula (light blue) and elbow (red) are tracked using a mean-shift algorithm. The relative position of the elbow with respect to the scapula is the state of every frame ($K = 2$).

In this experiment, the training data are taken from one healthy rat, but the phase extraction test is done on another

Fig. 2: A healthy rat on a running treadmill. The arrows point at the two tracked markers. The 2D position of the elbow (red marker) relative to the scapula (light blue marker) is the data used in the phase extraction.

rat with some variations in the walking conditions. The first rat is set to walk freely in one direction on a treadmill running with a constant speed of 12 cm/s (Fig. 2). The second rat is spinalized and has no ability to move its hind-limb. Therefore, its posterior parts are constrained by a weight support system and an ankle driving system [7]. Its forelimbs are still moved by the rat on a slower treadmill at 3 cm/s.

The parameters used in this setup are $K = 2$, $r = 7$, $H = 8$, $\epsilon = 5\pi/180$, and $\sigma = 0.1$.

B. Human's Upper Limb Movement

A compact motion capture system is developed based on one Microsoft Kinect v2 recording at 30 frames per second and two inertial measurement units (IMU) on the subject wrists [8]. General features of upper-body movement in each frame are extracted as a 12-dimensional vector ($K = 12$) which includes upper trunk orientation (3D), right upper arm pointing direction (3D), right forearm pointing direction (3D), right forearm pronation/supination (1D), right scapula elevation (1D), and right scapula protraction (1D).

Five healthy subjects are recruited to perform a movement. Each subject slides their right hand on a table to draw a rectangle for 100 repetitions. All subjects are asked to mix slow, medium, and fast paces to their movements for data variations. Data from 4 subjects are used as the training set. Data from the other subject are excluded for the phase extraction test. The parameters used in this setup are $K = 12$, $r = 3$, $H = 48$, $\epsilon = \pi/180$, and $\sigma = 0.1$.

IV. RESULTS & DISCUSSIONS

A. Rat's Forelimb Gait Cycle

The movement of the elbow with respect to the scapula collected from the healthy rat contains inconsistent gait (Fig. 4) with some additional noise from its unpredicted behaviors. For example, it sometimes just stops walking for a few seconds to scratch its head and then does a quick leap to catch up with the treadmill. Despite such contamination in the training data, the neural network has successfully learned the phase extraction as expected. When applying this network to the movement from the spinalized rat, the network is able to extract the phase (Fig. 5) even though the treadmill speed is 4 times slower than the speed used in the training

Fig. 3: With phase information assigned automatically to every sample in the training data, the data can be aligned directly from the record without any pre-segmentation or time-warping algorithm. **Grey points**: samples of relative elbow positions from the training data against the extracted phase. **Black lines**: average trajectory in each dimension of the training data as a by-product calculated from extracted phases.

Fig. 4: **Grey lines**: training data recorded from the healthy rat. They contain about 110 footsteps from about 3 minutes of record. The data contain not only the walking motion but also other random movements such as face scratching or body shaking. Zoom in to see the details in the digital version. **Colored dotted loop**: the average trajectory automatically extracted from the training data.

data. The capability of phase extraction from an unseen rat at the untrained speed could provide useful information for the hind-limb gait driving system to stay synchronized with forelimbs.

In addition, the method could also be used to extract the average trajectory of the movement (Fig. 3 and 4) without the need to clean-up the unrelated movement or segmenting each movement period manually.

B. Human's Upper Limb Movement

With multiple subjects and intentional speed variations in the training data, the network has learned and performed

Fig. 5: Phase extraction from the spinalized rat's forelimb movement (testing dataset). Despite the variations of the input position in both axes, the learned network is able to assign reasonable phases to those samples.

Fig. 6: Phase extraction from the last subject's movement (testing dataset). The top plot shows selected 2 dimensions from all 12 dimensions (K). Even with more than 3 folds of variation in speed of the movement, the extracted phase can follow. The sudden speed transitions at the 14th and the 30th second are handled responsively.

phase extractions from an unseen subject with a large contrast in the movement frequency (Fig. 6).

This robustness against speed variation fits well with a need in rehabilitative movement analysis. The model can be trained using movements from healthy subjects. Then, the trained model could be used to analyze rehabilitative patients who usually have relatively slower movement.

C. Computational Cost

1) *Training Speed:* The speed depends largely on the size of the network and the size of training data. For the rat's gait case with the smaller network ($H = 8$) and 11,300 pairs of training samples, the training takes under 6 minutes on

Nvidia GTX 1080 GPU. For the human movement case with the larger network ($H = 48$) and 22,807 pairs of training samples, it takes under 25 minutes on the same GPU.

2) *Phase Extraction Speed:* The feed-forwarding process on both networks takes less than 1 ms on a core of Intel Core i7-3770 CPU.

V. LIMITATION & FUTURE WORK

With the proposed cost function (Eq.1) that is purely unsupervised, the learned function provides just one possible way of mapping among infinite possibilities. Some preferred characteristics of the mapping are still uncontrollable. For example, the phase zero could map to an arbitrary spot on the period. A section of the movement could also occupy a large or a small proportion in the phase without a proper way to control. To give the method more control over the mapping function without sacrificing much automation, a modification towards a semi-supervised approach should be explored.

VI. CONCLUSION

A new universal framework for automatic phase learning and extraction is introduced to tackle some problems in repetitive movement modeling. With a unique advantage of being an unsupervised method with robustness to noise and movement speed variation in the data, this method is suitable for various applications that have access to a fair amount of target movement recorded in a repetitive manner. Our experiments on 2 different types of movement show that this method could potentially be useful in many applications, especially in robotics and rehabilitation. Nonetheless, as the method works with any high-dimensional time-series data, it is not limited to just movements but other kinds of periodic signals as well.

REFERENCES

- [1] W. Timoszyk, R. De Leon, N. London, R. Roy, V. Edgerton, and D. Reinkensmeyer, "The rat lumbosacral spinal cord adapts to robotic loading applied during stance," *Journal of Neurophysiology*, vol. 88, no. 6, pp. 3108–3117, 2002.
- [2] D. Holden, T. Komura, and J. Saito, "Phase-functioned neural networks for character control," *SIGGRAPH 2017*, 2017.
- [3] S. B. Thies, P. A. Tresadern, L. P. Kenney, J. Smith, D. Howard, J. Y. Goulermas, C. Smith, and J. Rigby, "Movement variability in stroke patients and controls performing two upper limb functional tasks: a new assessment methodology," *Journal of NeuroEngineering and Rehabilitation*, vol. 6, no. 1, p. 2, Jan 2009.
- [4] L. Rabiner and B. Gold, *Theory and Applications of Digital Signal Processing*. Prentice Hall Press, 1975.
- [5] P. Jatesiktat and W. T. Ang, "Unsupervised phase extraction using dual autoencoder," in *International Conference on Advanced Computing and Applications (ACOMP)*, 2017, pp. 71–76.
- [6] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [7] D. Anopas, S. Y. Chew, J. Lin, S. K. Wee, T. P. Er, and W. T. Ang, "A developmental rehabilitation robotic system for a rat with complete thoracic spinal cord injury in quadruped posture," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2109–2115, July 2018.
- [8] P. Jatesiktat, A. Dollaporn, and W. T. Ang, "Personalized markerless upper-body tracking with a depth camera and wrist-worn inertial measurement units," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2018.