Neural Computing and Applications (2023) 35:23343–23356
https://doi.org/10.1007/s00521-021-06192-3

SPECIAL ISSUE ON HUMAN-IN-THE-LOOP MACHINE LEARNING AND ITS APPLICATIONS

A deep learning framework for realistic robot motion generation

Ran Dong[1] · Qiong Chang[2] · Soichiro Ikuno[1]

Received: 4 January 2021 / Accepted: 2 June 2021 / Published online: 15 June 2021

Abstract
Humanoid robots are being developed to play the role of personal assistants. With the development of artificial intelligence technology, humanoid robots are expected to perform many human tasks, such as housework, human care, and even medical treatment. However, robots cannot currently move flexibly like humans, which affects their fine motor skill performance. This is primarily because traditional robot control methods use manipulators that are difficult to articulate well. To solve this problem, we propose a nonlinear realistic robot motion generation method based on deep learning. Our method benefits from decomposing human motions into basic motions and realistic motions using the multivariate empirical mode decomposition and learning the biomechanical relationships between them by using an autoencoder generation network. The experimental results show that realistic motion features can be learned by the generation network and motion realism can be increased by adding the learned motions to the robots.

Keywords Realistic motion generation · Convolutional autoencoder · Multivariate empirical mode decomposition · Human in the loop

1 Introduction

Recently, humanoid robots, using developments in artificial intelligence technology in the areas of the Internet of Things (IoT), computer vision, and big data, are becoming smarter and are expected to perform many human tasks, such as housework, human care, and even medical treatment. However, these tasks require a humanoid robot not only to have an intelligent brain but also to perform a series of flexible motions. These motions must be articulated well by the manipulators used on the humanoid robots, but this remains challenging. To improve the quality of robot motion, many researchers are focusing on designing humanoid robot motion for various tasks based on human motion. Ding et al. [1] developed a humanoid robot for nursing care. Their robot can carry patients from a wheelchair to a bed using a structure based on the human arm. Borovac et al. [2] developed a robot to provide physical rehabilitation for children with cerebral palsy. Their humanoid robot, MARKO, was designed in a cartoonish style. Nishiguchi et al. [3] suggested a behavior design method for humanoid robots. They experimented with several ordinary motions and examined whether it was important for these motions to be human-like in the interaction between humans and robots. All of these studies are based on methods that create continuous motion by linearly interpolating between discrete keyframes. However, because of the complexity of the human biomechanism, this type of motion is usually too rigid to have the realism for high-level applications, such as physical rehabilitation and entertainment. Fortunately, Sanzari et al. [4] advanced a theory that human motion is composed of basic units called "motion primitives," which can help us understand motion mechanics more deeply. According to [4], motion primitives are complicated and nonlinear. Thus, traditional methods, which determine human motion discretely based

✉ Qiong Chang
q.chang@c.titech.ac.jp

Ran Dong
randong@stf.teu.ac.jp

Soichiro Ikuno
ikuno@stf.teu.ac.jp

[1] School of Computer Science, Tokyo University of Technology, Tokyo 192-0982, Japan

[2] School of Computing, Tokyo Institute of Technology, Tokyo 152-8550, Japan

on keyframes, lose a significant amount of the motion primitives, resulting in a lack of realism in the motion. Therefore, to increase the realism of robot motion, it is crucial to add nonlinear motion primitives to the functionality of the robot.

In this study, we propose a nonlinear realistic robot motion generation method based on deep learning. To examine the mechanism of realistic robot motion, we first demonstrate the realistic motion features unique to human motion by decomposing the motion of robots and humans using multivariate empirical mode decomposition (MEMD). Then, we train an autoencoder generation network to generate realistic motion features from the basic motion, which is also extracted from humans but can be performed by robots. The contributions of our research are as follows:

- Understanding the role of realistic motion features provides a research foundation to improve the quality of robot actions in the future.
- Our method based on an unsupervised generation network offers a novel way to efficiently transfer the realism of human motion to robots.

The remainder of this paper is structured as follows. Section 2 briefly reviews related Works. Sections 3 and 4 introduce the theories of empirical mode decomposition and demonstrate realistic motion features, respectively. In addition, we introduce our proposed framework for the realistic robot motion generation in Sect. 5 and evaluate it in Sect. 6. Section 7 is the conclusion of this research.

2 Related works

Many studies have been conducted to generate realistic robot motion based on human activities. Okajima et al. [5] proposed a controller using a mechanical resonance mode to generate human-like movement in robots. Tomic et al. [6] focused on dual-arm manipulation based on human arms to accomplish interactive tasks within the environment. All these studies aimed to generate human-like motion to bring humanoid robots into our ordinary life, and they verified that using human features is the key to enacting human-like motion in robots.

Additionally, some studies have focused on feature extraction and synthesis of human motion. Beaudoin et al. [7] extracted a motion motif representing a cluster of similar motions from the motion capture database. This method is useful for motion compression and motion detection. Min et al. [8] introduced a generative statistical model that allows users to analyze and edit human motion semantically and kinematically. All of these methods extract motion features in the time domain. On the contrary, Dong et al. [9, 10] decomposed human motion into several motion primitives using MEMD in the instantaneous frequency domain. This decomposition not only helps to extract motion features but also makes it possible to learn these features.

Recently, deep learning method has received increasing attention as state-of-the-art technology. Wang et al. [11] introduced a deep network to extract features by creating a natural motion manifold. Alemi et al. [12] proposed GrooveNet, which can generate dance motions for the given music. Holden et al. [13] presented a neural network model of CNN autoencoder, which can extract a motion manifold to fix corrupted human motions. Using the extracted motion manifold, Holden et al. [14, 15] also presented different neural networks to generate complicated human motions. Although the above studies improved the quality of robot motion, their motion designs are aimed directly at humans without biomechanics, which results in a lack of human-like realism regardless of the level of detail in their motion.

3 Empirical mode decomposition

Empirical mode decomposition (EMD) is a method to decompose real-world signals into multiple intrinsic mode functions (IMFs) and a residual so-called trend. It was originally proposed by Huang et al. [16] and it was expanded from a single variable to multiple variables, or MEMD, by [17–20]. Because the IMFs are pseudo-monochromatic waves, their instantaneous frequency and amplitude can be calculated using the Hilbert transform (HT) [21]. For nonlinear multi-channel signals like human motion, MEMD is a powerful tool for extracting motion features. It uses Hammersley sequences of prime numbers to create an N-dimensional sphere and obtains the multivariate IMFs by projecting the multivariate signals onto the sphere. According to [16, 21], motion capture data can also be treated as a signal and processed by MEMD. Its function can be defined as follows:

$$\sum_{m=1}^{M} C(m,n,t) + R(n,t) = X(n,t), \quad (X \text{ and } R \in \mathbb{R}^{N\times T}, C \in \mathbb{R}^{M\times N\times T}), \tag{1}$$

where X represents the multivariate motion capture data to be decomposed. C represents the decomposed IMFs corresponding to motion primitives, and R represents the residual corresponding to the posture of the entire motion. $X(n, t)$ and $R(n,t) \in \mathbb{R}^{N\times T}$, where N and T represent the degrees of freedom (DOFs) and frames, respectively. $C(m,n,t) \in \mathbb{R}^{M\times N\times T}$, where M represents the number of IMFs. In addition, to obtain the instantaneous frequency

and amplitude of each decomposed motion primitive in each DOF, an analytical signal $Z(t) \in \mathbb{C}^T$ of each decomposed IMF and each DOF is defined as follows:

$$\begin{aligned} &Z(t) = C_{\mathrm{Re}}(t) + iC_{\mathrm{Im}}(t), \ (Z \in \mathbb{C}^{N\times T}) \\ &C_{\mathrm{Re}}(t) \in \{C(m,n,t)|m = 1,\ldots,M, n = 1,\ldots,N\}, \end{aligned} \tag{2}$$

where $C_{\mathrm{Re}}(t)$ is the real part, which is the decomposed IMFs of each DOF, and $C_{\mathrm{Im}}(t)$ is the imaginary part that can be obtained using the HT as follows [22]:

$$C_{\mathrm{Im}}(t) = \frac{1}{\pi} PV \int_{-\infty}^{\infty} \frac{C_{\mathrm{Re}}(\tau)}{t-\tau}\, d\tau = \frac{1}{\pi t} * C_{\mathrm{Re}}(t), \tag{3}$$

where PV represents the Cauchy principal value. Then, by considering (2) in the complex plane, the instantaneous frequency ω and instantaneous amplitude A are calculated as follows:

$$A(t) = \sqrt{{C_{\mathrm{Re}}}^2(t) + {C_{\mathrm{Im}}}^2(t)} \tag{4}$$

$$\omega(t) = \frac{d}{dt} \arctan \frac{C_{\mathrm{Im}}(t)}{C_{\mathrm{Re}}(t)} \tag{5}$$

Hilbert spectral analysis (HSA) uses the instantaneous frequency ω and the instantaneous amplitude A from (4, 5) to obtain the spectrum, which is widely used in feature analysis in the instantaneous frequency domain. Next, we can perform HSA to analyze realistic robot and human motion features using the instantaneous amplitudes and frequencies of all IMFs for each DOF.

4 Realistic motion feature analysis and synthesis

4.1 PremaidAI structure

To understand the mechanism of realistic motion features, we decompose the same motions of robots and humans using MEMD and compare them with each other.

In our research, the humanoid robot "PremaidAI," developed by DMM [23], was chosen because it has sufficient DOFs to perform realistic motions. Figure 1 shows the definition of a motor rotational angle of the PremaidAI head joint. The motor rotational angle can be considered as an Euler angle in robot control [24]. There are three DOFs in the PremaidAI head joint, represented by θx, θy and θz. Table 1 shows the comparison of the DOFs between the most important joints in humans and the PremaidAI. In this study, we discuss only joint angles of both humans and robots because we focus on the realistic features. As shown in the table, the PremaidAI has only 25 DOFs, while humans have many more. Because the physical structure of robots is simpler than that of humans, the ranges of the DOF are different. This means that to make the motion of a robot more like that of a human, a more sophisticated motion design is required.

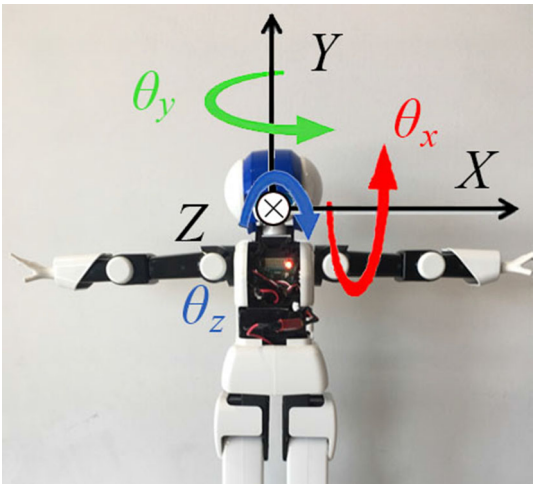


Fig. 1 Definitions of the robot motor DOFs

Table 1 Degrees of Freedom (DOFs) Comparison

Joints	Human body	PremaidAI
Head	3	0
Neck	3	3
Left shoulder	3	2
Left elbow	3	2
Left hand	3	1
Right shoulder	3	2
Right elbow	3	2
Right hand	3	1
Hip	3	0
Left hip	3	3
Left knee	3	1
Left feet	3	2
Right hip	3	3
Right knee	3	1
Right feet	3	2
Total	45	25

4.2 Motion analysis

Figure 2 shows some of the motions of the PremaidAI in an artistic performance [25]. These motions use only the upper body to finish simple hand and head motions. To ensure the motions are synchronized, we capture the same robot motion (Fig. 2a) from a human (Fig. 2b) using the Perception Neuron motion capture system [26].

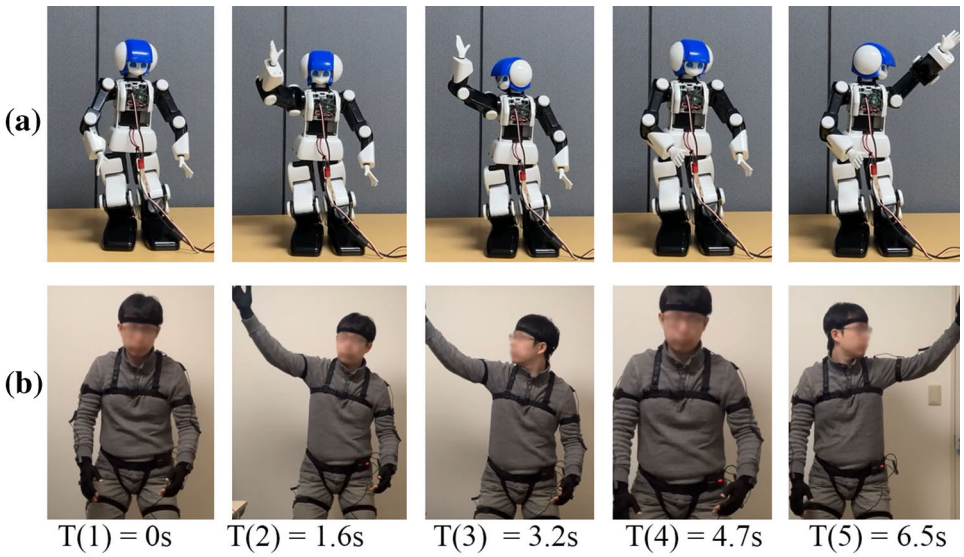


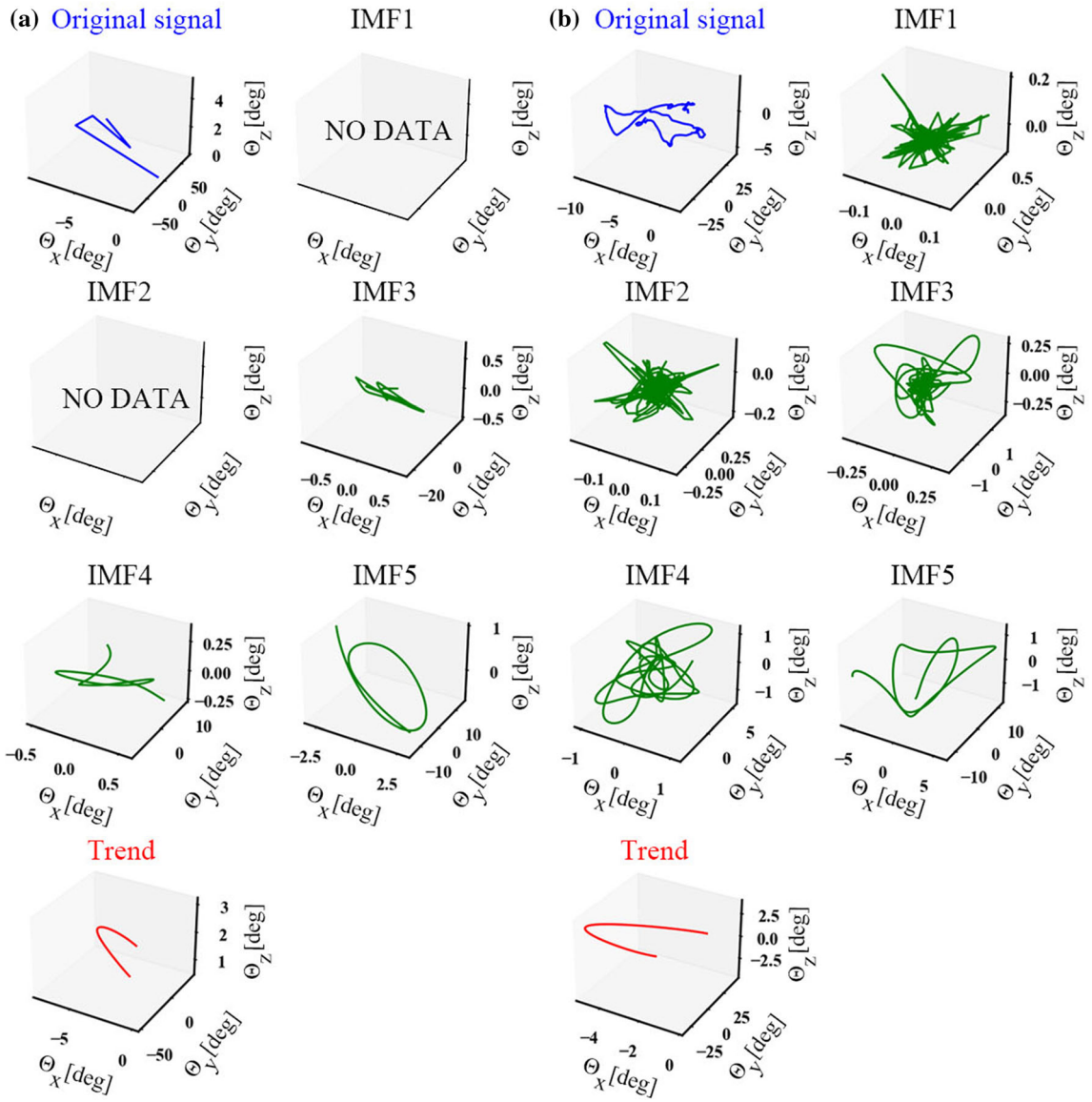
Fig. 2 Comparison of robot and human motions **a** robot motions **b** human motions

As mentioned above, even for the same motion, robots and humans still use many different dynamically linked joints due to their different physical structures. Here, the neck is chosen for analysis because both the PremaidAI and humans have the same number of DOFs. Figure 3 shows the comparison of their decomposition results using MEMD based on a stopping criterion discussed in [27]. As introduced in Sect. 3, motion can be decomposed multivariately using MEMD. In the algorithm, the motions are decomposed from high frequency to low frequency corresponding to IMF1-N. The non-periodic residual has lowest frequency and is known as the trend. Here, θx, θy, and θz represent the relation angles of each decomposed head joint motion. As can be seen in the Fig. 3, the head motions are decomposed multivariately, with IMF1 having the highest frequency and the trend having the lowest. In Fig. 3a, a PremaidAI linear original motion signal is decomposed into three IMFs and one trend, as discussed in Sect. 3. Because the original signal is simple owing to the keyframe design method, the decomposed IMFs are correspondingly less complicated. On the other hand, the original human signal shown in Fig. 3b is more complicated than the PremaidAI signal. It is no longer linear because it is obtained by sampling human motions. Hence, it is more complicated, and more IMFs are decomposed, representing more motion primitive details than the PremaidAI movement.

Furthermore, by applying HT to each IMF, the corresponding instantaneous amplitudes and frequencies of the PremaidAI and human can be obtained, as shown in Fig. 4. To obtain the figure, we applied HT to θx, θy and θz in Fig. 3 and obtained averaged frequencies and Euclidean distances of amplitudes for each IMF. In this figure, the horizontal axis represents the time, and the vertical axis represents the frequency. The color bar represents the amplitude in the range of [0,1], from the lowest in blue to the highest in red. Because the high-frequency motions have much lower amplitude than the low-frequency motions, we take log of the amplitudes to show the difference clearly. In addition, because the neck joint has three DOFs, we take the average of each frequency and the Euclidean distance of each amplitude.

A high frequency indicates a fast motion, while a large amplitude indicates heavy motion. Here, because of the decomposition errors of EMD at high frequencies, a weighted average frequency algorithm is used to denoise the decomposed motion and improve the accuracy [28]. Moreover, because HT uses the two-order differential method, the obtained maximum frequency is four times lower than the sampling frequency [22]. Because the PremaidAI sampling frequency is 60 Hz [29], the maximum frequency of motions in our analysis is 15 Hz, no motion lasting less than 0.07 s can be detected. However, this rate is sufficient for motion analysis and synthesis because few important motion primitives occur in less than 0.07 s. Figure 4a shows the keyframes of the PremaidAI motion, in which few decomposed motion primitives are under 2 Hz. In particular, the highest frequencies (IMF3) are very regular at approximately 0.8 Hz (between 0.125 and 1.8 Hz). Compared with the PremaidAI motion, human motion shown in Fig. 4b has nearly the same low-frequency motion primitives (approximately 1.0 Hz). The difference is that it also consists of complicated motion primitives belonging to the high-frequency domain, such as IMF1, IMF2, and IMF3.

To obtain a quantitative indicator for the decomposed motion primitives of the robot and the human, we use the absolute value of the correlation coefficients of the IMFs



	IMF1	IMF2	IMF3	IMF4	IMF5	Trend
Robot	×	×	○	○	○	○
Human	○	○	○	○	○	○

Fig. 3 Comparison of motion decomposition **a** robot **b** human

and trend in our research. We calculate the Euclidean distance of θx, θy, and θz, as shown in Fig. 3, to obtain correlation coefficients of each joint. As shown in Table 2, the upper body joints are listed in the instantaneous frequency domain. Because MEMD extracts IMFs from high frequency to low frequency, human motion has more complicated motion primitives (IMF1, IMF2), as shown in Fig. 4b. Thus, we calculate the correlation coefficients between the IMFs in the same frequency domain. As Table 2 shows, the low-frequency motion primitives (IMF5) and the trend have higher correlation coefficient values than the high-frequency motion primitives (IMF3-IMF4). In addition, the correlation coefficient values are different among the joints, because the desynchronized motion typically occurs in the tip of the body structure. Here, the IMF with the highest amplitude can be considered the threshold for splitting human motion into two distinct groups: (1) basic motion that both the robot and the

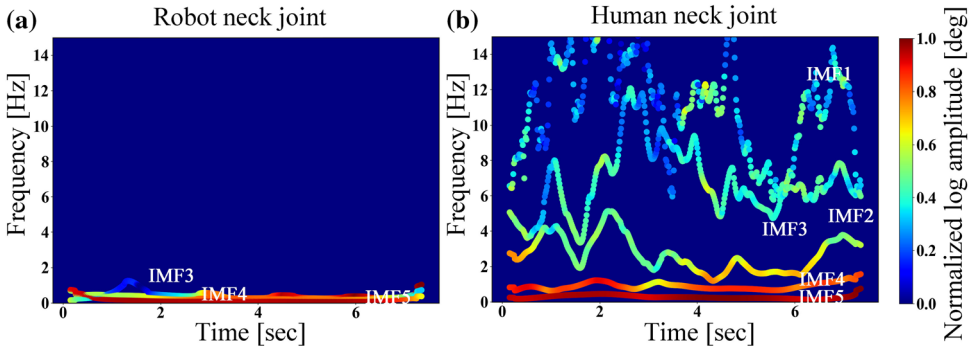


Fig. 4 Hilbert spectral of the neck joint **a** robot motion **b** human motion

Table 2 Comparison of IMF correlation coefficients

	Right			Left				
Robot & Human	Sr	Ew	Hd	Sr	Ew	Hd	Nk	Avg
IMF 3	0.050	0.130	0.211	0.073	0.208	0.089	0.275	0.148
IMF 4	0.057	0.055	0.062	0.352	0.453	0.083	0.105	0.167
IMF 5	0.264	0.524	0.777	0.450	0.075	0.679	0.679	0.493
Trend	0.878	0.685	0.310	0.987	0.966	0.910	0.819	0.794

Sr Shoulder, *Ew* Elbow, *Hd* Hand, *Nk* Neck, *Avg* Average

human have, and (2) high frequency complicated motion primitives the robot does not have.

The analysis above shows that the difference between the robot and the human is primarily in the high-frequency domain. Thus, we can speculate that the lack of high-frequency motion primitives causes the unrealistic nature of robot motion. Moreover, we can expect that realism can be improved by adding decomposed high-frequency motion primitives of the human movement.

4.3 Motion synthesis

To verify our conclusion described above, we attempt to add the human motion primitives (IMF1-IMF4) into the robot motion. Here, a simple hierarchical model and Euler angles are used to record the human motion primitives. Then, they can be simply matched to the corresponding DOFs of PremaidAI to synchronize their motion. Figure 5 shows an example to verify our supposition. Figure 5a shows the original robot motion, which is the same as given in Fig. 2a. On the other hand, Fig. 5b shows a series of edited motions based on the motions shown in Fig. 5a. They are edited by manually adding some high-frequency motion primitives extracted from human motion, as described above. As shown in Fig. 5, the motions generated not only maintained the original posture but also further added some detailed motion features, represented by the red circles in Fig. 5.

Our verification demonstrates that the realism of robot motion can be improved by adding high-frequency motion primitives extracted from human motion. However, it is unrealistic in practice for people to repeat each motion required of robots to obtain the high-frequency motion primitives. Therefore, an automatic high-frequency generation method is necessary.

5 Realistic robot motion generation

To generate realistic motion features that the robot does not currently have, it is necessary to determine the biomechanical relationship between basic motions and realistic motion features in humans. Fortunately, according to [30], this relationship exists. Therefore, in this section, we introduce a framework to learn the relationship and use it to generate realistic motion for the robots.

5.1 Network structure

Our framework uses the autoencoder neural network proposed by Holden et al. [13, 14]. Plenty of studies have proved that the autoencoder is helpful for learning many useful features of motion data and is used to solve many applied problems, from motion detection and recognition to generation [31, 32]. Moreover, because it is difficult to label all motion data to be learned, the unsupervised learning feature of the autoencoder network is suitable for our task.

Figure 6 shows our autoencoder network, which consists of two parts: the encoder and decoder. As 6 shows, the

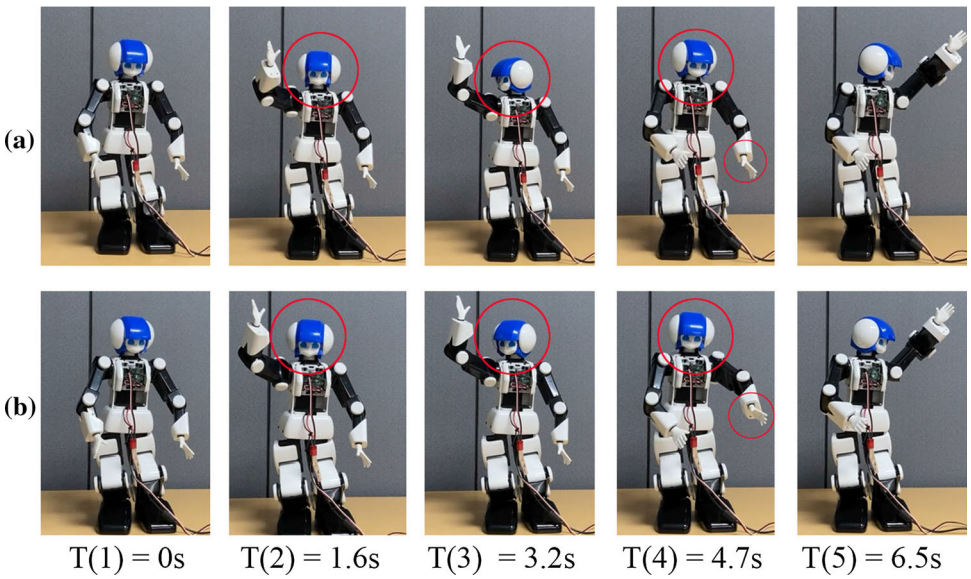


Fig. 5 Comparison of robot motions **a** original robot motions, **b** edited realistic robot motions

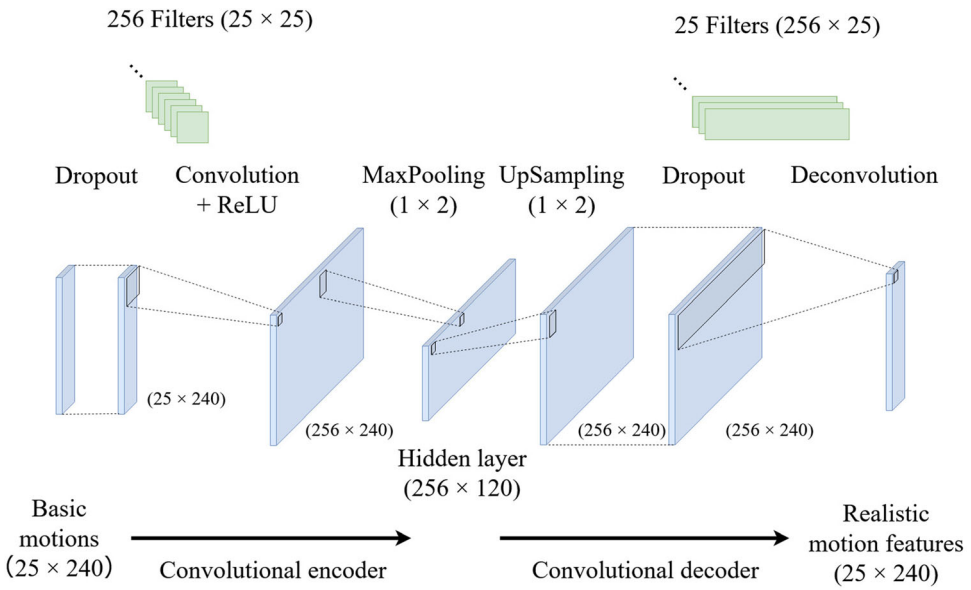


Fig. 6 Framework of realistic motion features generation

encoder first receives motion data B as the input and maps them to the hidden layer H through the convolution with weights matrix W, the nonlinear operation $ReLU$ [33] and the max pooling operation Ψ. Then, the decoder receives H as input and further maps it to reconstruct the motion data $\Phi^{\dagger}(H)$ through the up-sampling operation $\Psi^{\dagger}$ and the deconvolution with weights matrix $\tilde{W}$. Here, to avoid overfitting caused by the limited data set, two dropout layers are added before the convolution and deconvolution layers.

$$\begin{cases} H = \Phi(B) = \Psi(\mathrm{ReLU}(B * W + b)) \\ \Phi^{\dagger}(H) = \Psi^{\dagger}(H) * \tilde{W} + \tilde{b} \end{cases} \tag{6}$$

Furthermore, adaptive moment estimation (ADAM) is used as the optimizer, and the mean squared error (MSE) is used as the cost function, as follows:

$$\mathrm{Cost}(B, R) = \|R - \Phi^{\dagger}(\Phi(B))\|_2^2, \tag{7}$$

where R represents the label motion data. By minimizing the cost function Eq. 7, the key properties of the input which can be used to generate the output can be learned.

5.2 Training the autoencoder network

To find the biomechanical relationship between basic motions and realistic motion features using the above autoencoder network in our implementation, we decompose human motion into two groups: basic motions and realistic motion features, using the following algorithm 1:

Algorithm 1 Motion Assignment

Require: X // input motion
$B(n,t) \Leftarrow \emptyset$ // basic motions, $B \in \mathbb{R}^{N\times T}$
$R(n,t) \Leftarrow \emptyset$ // realistic motion features, $R \in \mathbb{R}^{N\times T}$
//**Decomposition**
$C(m,n,t) \Leftarrow \mathrm{MEMD}(X)$ // $C \in \mathbb{R}^{M\times N\times T}$
$A(m,n) \Leftarrow \frac{1}{T}\sum_{t=0}^{T} \mathrm{Hilbert}(C(m,n,t))$ //normalized average amplitude, $A \in \mathbb{R}^{M\times N}$
//**Reconstruction**
for $n = 1$ to N **do**
  for $m = 1$ to M **do**
    if $A(m,n) == 1$ **then**
      $B(n,t) \Leftarrow \sum_{m}^{M} C(m,n,t) + \mathrm{Trend}(n,t)$
      $R(n,t) \Leftarrow \sum_{1}^{m-1} C(m,n,t)$
      Break
    end if
  end for
end for

The input motion is first decomposed by MEMD into several IMFs C. Next, the HT is performed on the IMFs C, and a normalized average amplitude A over a time segment T is calculated. $A \in \mathbb{R}^{M\times N}$, which can be used as a reference threshold to split the decomposed C. The value of the amplitude A is traversed in the IMF space M and the DOF space N. When the corresponding A in any number m of the IMF is equal to 1 (the highest amplitude), the motion primitives with higher frequencies are categorized in the basic motions group B, and the remainder is categorized in R as realistic motion features. Then, the basic motions are used as input in the autoencoder network, and the realistic motion features are used as the labels, which can help us learn the manifold between them. Since the PremaidAI joints have 25 DOFs, each motion is represented using a vector of length 25. Our autoencoder network performs a one-dimensional convolution to slide one frame over in temporal domain and learns 240 frames of motion in each batch. Simultaneously, 256 independent filters with kernel size 25×25 are used for the encoder, and 25 filters with size 256×25 are used for the decoder. Therefore, B shown in Eq. 7 represents the basic motions and R represents the realistic motion features, where B and $R \in \mathbb{R}^{25\times 240}$. The encoder maps 240 frames of basic motions into a 256×120 hidden layer H, where $H \in \mathbb{R}^{256\times 120}$, through a 1×2 max pooling operation Ψ. On the other hand, the decoder reconstructs the motions through a 1×2 up-sampling operation Φ.

By training the autoencoder network with the decomposed motion primitives from human motion, it is easy to obtain realistic motion features from basic motion, which is also applicable to robots. Here, some high-frequency motion features generated by the autoencoder network cannot be performed owing to the limitation of the motor speed. We eliminate these and add the rest to the robot motions to increase their realism based on the max speed of motor using MEMD [34]. Moreover, since people's evaluation of motion realism is subjective, it is important to allow the robot motion designers to edit the amplitude of the motion features to obtain the desired effect, for example, exaggerating certain features. In this way, our framework can be a powerful tool to generate realistic robot motion.

6 Results and discussions

6.1 Training data preparation

To train our autoencoder network, we prepared 15 h of human motion capture data from [35]. The data set includes various types of human motion collected by previous studies [36–39]. Although the original sample rate of the data set was 120 Hz, we resampled all motion to 60 Hz to fit the motor speed. Furthermore, we split the motion into 240 frames per window with an overlap of 120 frames, making the training data into 30 h. Here, because the PremaidAI has only 25 DOFs, as mentioned in Sect. 4.1, to make the data set suitable for the autoencoder, all DOFs of each motion in the data set were reduced and processed using MEMD.

We directly reduced the DOFs from the human body structure (45) to the robot motor structure (25), as shown in Table 1. Figure 7 shows two examples, walking and jumping, after reducting the DOFs. As seen in these figures, even though there are some changes in the robot motions due to the DOF reduction, the main motion features, including biomechanics, are preserved. Thus, we can use the motion data with 25 DOFs to train our proposed framework and generate realistic features for the robot.

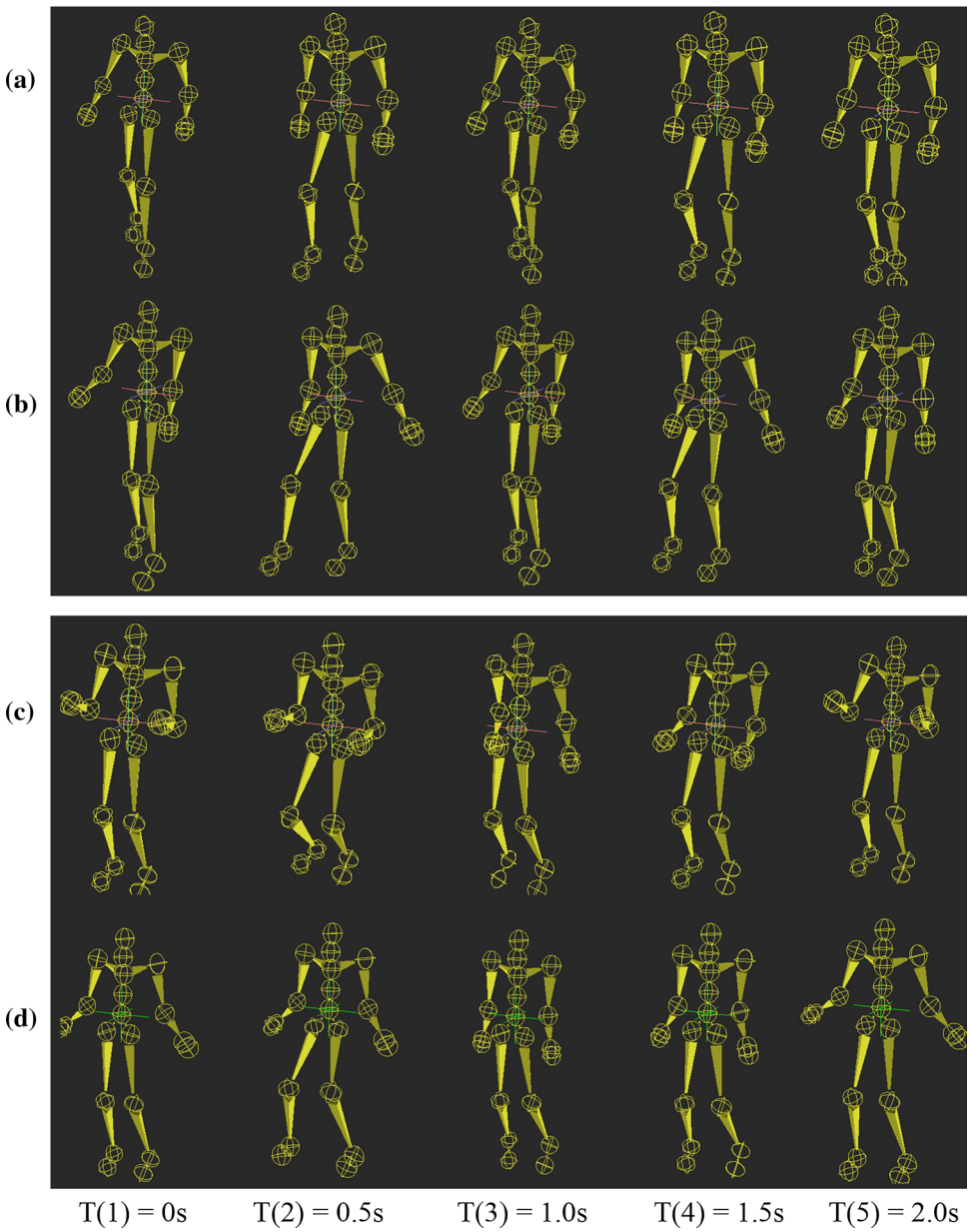


Fig. 7 Comparison of human and robot DOFs **a** human walking motion (DOFs = 45), **b** robot walking motion (DOFs = 25) **c** human jumping motion (DOFs = 45), **d** Robot jumping motion (DOFs = 25)

6.2 Neural network performance

After preparing the training data, the autoencoder network was trained to obtain the manifold between the basic motions and realistic motion features. In our training step, the learning rate was set to 0.00001, and the batch size was set to 1. We randomly selected 90% of the motions (13.5 h) as the training data and 10% of the motions (1.5 h) as the testing data. Figure 8 shows the result of our framework. The blue curve represents the training result, and the orange curve represents the testing result. During the first 20 epochs, both the training and testing losses significantly decreased. After 60 epochs, the losses decreased more slowly and became stable at approximately 100 epochs.

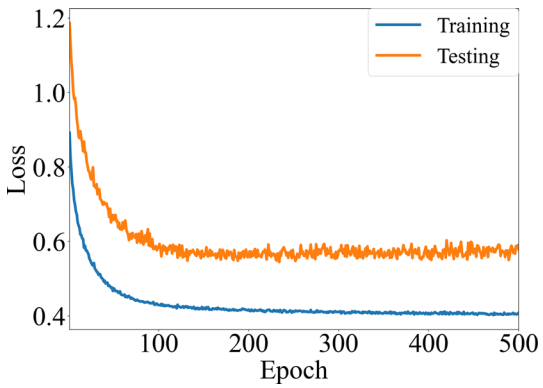


Fig. 8 Performance of the proposed network after 500 epochs with a learning rate of 0.00001

Through 500 epochs of training, the mean training loss decreased from 0.9 to 0.41, while the testing loss decreased from 1.18 to 0.59. However, after 100 epochs, the testing loss was nearly stable, while the training loss was still declining. This indicates that overfitting occurred after 100 epochs. Thus, our results show that the motion manifold can be learned successfully and realistic motion features can be generated by our proposed framework after 100 epochs using a learning rate of 0.00001.

To give an overall evaluation of the learning performance of our proposed framework, we set the learning rate to 0.000001 and 0.0001. Figure 9a shows the results with the learning rate set to 0.000001. As the figure shows, if the learning rate is too low, 100 epochs are not sufficient to obtain the best result, even though the learning process is smooth. On the other hand, Fig. 9b shows the result with the learning rate set to 0.0001. If the learning rate is too high, the best result shown in Fig. 8 cannot be obtained, even though its learning process is fast. Furthermore, the learning process of the testing data is unstable, with significant errors. Thus, the appropriate learning rate for our proposed framework is 0.00001.

Next, we use 0.00001 learning rate to evaluate the performance using different data sets. Figure 10 shows the learning results using two different data sets. Figure 10a shows the learning results using only the CMU data set. Figure 10b shows the learning results using only the punching motion data set. As shown in the figures, although our proposed framework produces different results using two different data sets, both the training loss and the testing loss decline during the learning process. Even if better parameters for the learning rate and epoch exist, the robustness of our framework is verified. Hence, our learning method can be used to synthesize various motions for different purposes.

6.3 Basic realistic robot motion synthesis

In addition, we used two different groups of motions to further demonstrate the performance of our framework. The first group of motions consists of squatting and standing activities were chosen as samples because they have fewer motion primitives. The motions are concentrated in the leg joints, where the effect is easily confirmed. Figure 11 shows a comparison of the results for the first group using linear interpolation and our framework. Figure 11a shows that the basic motions consist of first squatting and then standing. Because there are only three keyframes in the legs, the basic motion is filled linearly, resulting in a lack of realism. On the contrary, after blending the realistic motion features with the basic ones, the motion becomes more realistic, as shown in Figure 11b. To show the comparison clearly, we doubled the amplitude of the realistic motion primitives generated by our framework. It is worth noting that although all the basic motions were input into the autoencoder network, the generated realistic motion primitives occur only in the squat. This shows that our framework can learn the manifold correctly. Furthermore, it can be seen that when the robot squatted, the handshake motion was also added because of the biomechanics of human motion when the center of mass was lowered. Therefore, the motions edited by our framework are more realistic than the basic ones.

6.4 Advanced realistic robot motion synthesis

Next, we adapt the second group of motions to our framework. In this group, we choose a set of greeting motions designed by a professional robot motion designer [40]. These motions are much more complicated than the previous ones because all the upper joints are used to perform the full motions. As shown in Fig. 12, the original greeting motions use only the arm and head joints. On the other hand, the motions edited by our framework also use the leg joints to change the angle at the hip, which makes the motion more realistic. The edited motion spreads the legs just before the bow motion, which reproduces human biomechanical motions, similarly to squatting and standing motions.

To analyze the learning results more deeply, we also calculated the Hilbert spectrum of both 12(a) and 12(b). Figure 13 shows the left hip joint spectra that demonstrate

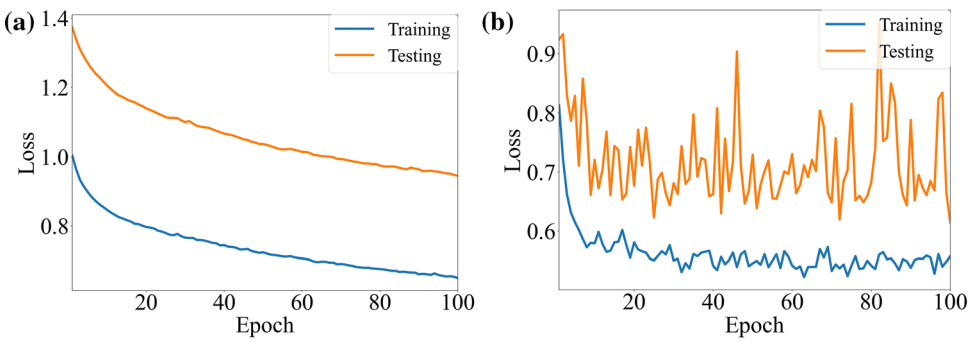


Fig. 9 Performance of the proposed network after 100 epochs **a** Learning rate of 0.000001, **b** learning rate of 0.0001

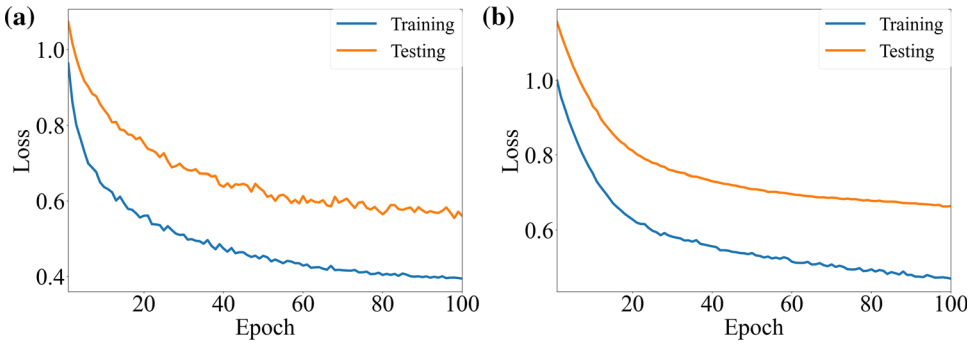


Fig. 10 Performance of the proposed network after 100 epochs with a learning rate of 0.00001. **a** Training results using CMU data only. **b** Training results using punching motion data only

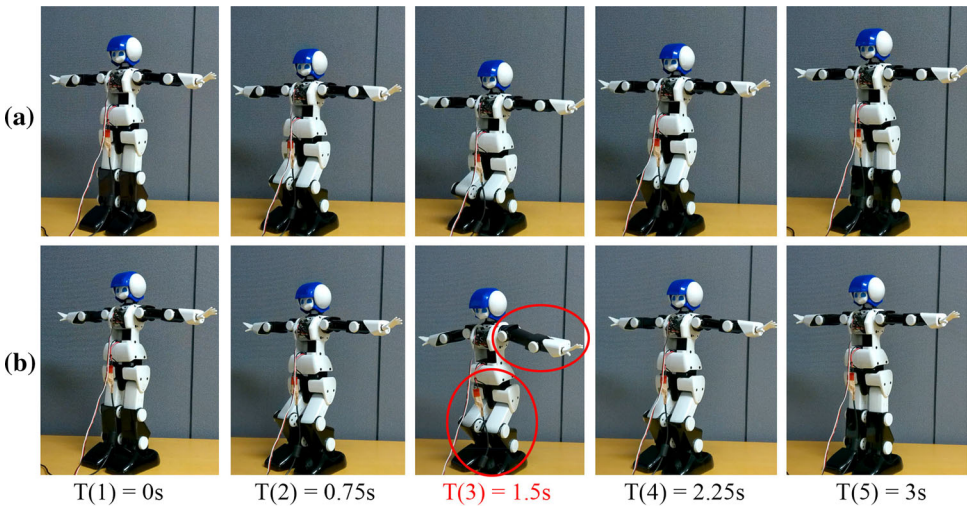


Fig. 11 Comparison of squatting and standing motions. **a** Original robot motions. **b** Edited realistic robot motions

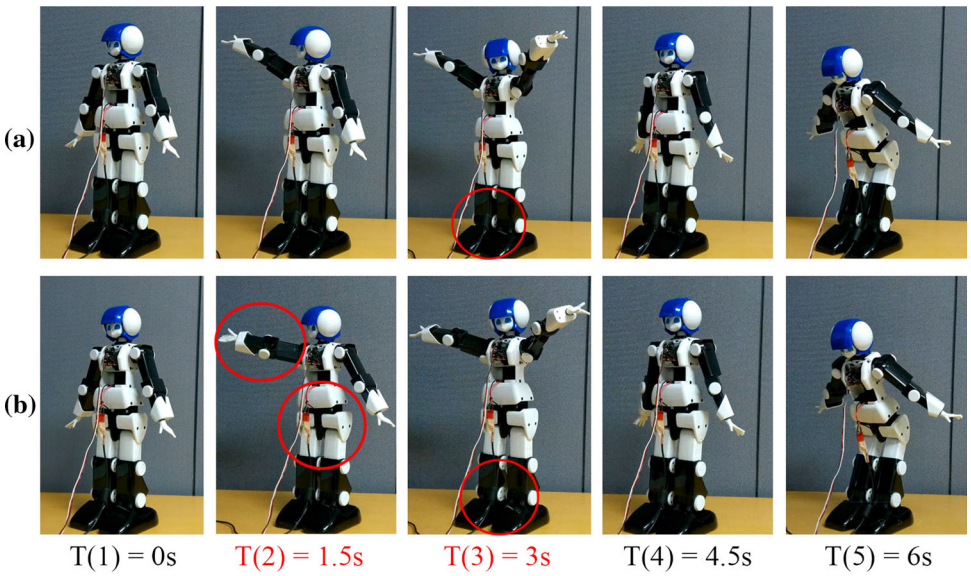


Fig. 12 Comparison of greeting motion. **a** Original robot motions. **b** Edited realistic robot motion

the leg joint movement in the instantaneous frequency domain. By comparing 12a and b, the frequency of the basic motions is almost entirely under 4 Hz. On the other hand, the edited realistic robot motions have 8 Hz motions at approximately 2 s, and 6 Hz motions at approximately 2 s and 5 s. These motions are realistic leg motions corresponding to T(2) and T(4) shown in Fig. 12. The valley at approximately 4 s, shown in the 12b, is caused by the pause in movement corresponding to Fig. 12 T(4). This also demonstrates that our proposed method generated realistic features based on temporal correlations correctly.

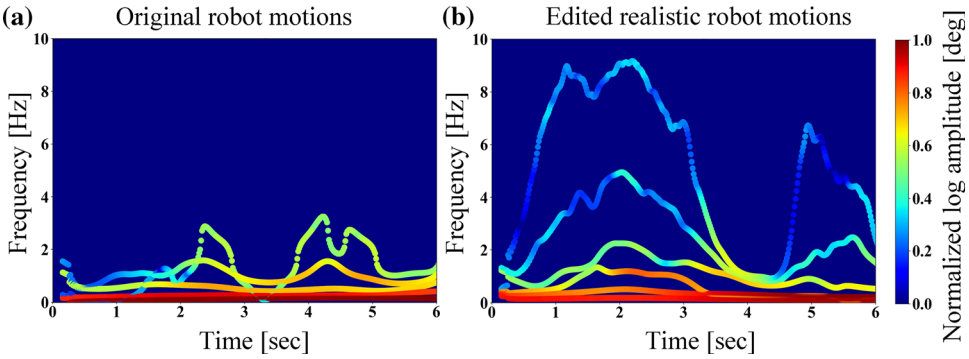


Fig. 13 Hilbert spectrum analysis of the greeting motion (left hip joint). **a** Original robot motion. **b** Edited realistic robot motion

6.5 Performance evaluation compared with other methods

To evaluate the performance of our proposed framework, we compared our results with the previous motion editing results obtained proposed by Holden et al. [14]. The previous research showed that styled motions could be obtained by editing the hidden units using both content (original) and style (target) motion based on a Gram matrix [14]. We used robot keyframe motion as the content motion to be edited and human motion as the style motion for analyzing the realistic features generated by the neural network.

Figure 14 shows the results of both neural networks. We use the same squatting and standing motions created by the keyframe method that we used in 6.4 to show the differences. For the style (target) motions, we captured human squatting and standing motions following the robot motions. As can be seen in Fig. 14a, b, both methods generate certain realistic features extracted from the human motions. However, because the method of the previous research edits the entire motion, basic motions are also changed by the neural network, such as arm motions from T(1) to T(5). In addition, because of the potential for the robot to become unbalanced, we must synchronize both robot legs to prevent falls. On the other hand, our proposed method only generates realistic features where required by the original motion, which is in the squatting motion occurring in T(3). Furthermore, it is unnecessary to counter the imbalance problem because the basic motions are not changed. Hence, our previous research can edit realistic features biomechanically without changing the basic motion.

The spectra of both generated motions also demonstrated the same results. Figure 15a shows the spectrum of the motion generated by the previous research. As the figure shows, the edited motion is decomposed into four motion primitives (IMFs). All are colored, which means the edited motion primitives exist from 0.2 to 5.8 Hz. On the contrary, Fig. 15b shows the spectrum of the motion generated by our proposed framework. Our edited motion has the same number of IMFs as the motion from the previous research. However, the amplitudes and frequencies are only altered at T(3), where the squatting motion occurs. This indicates that our proposed framework can identify where realistic features need to be added, which is

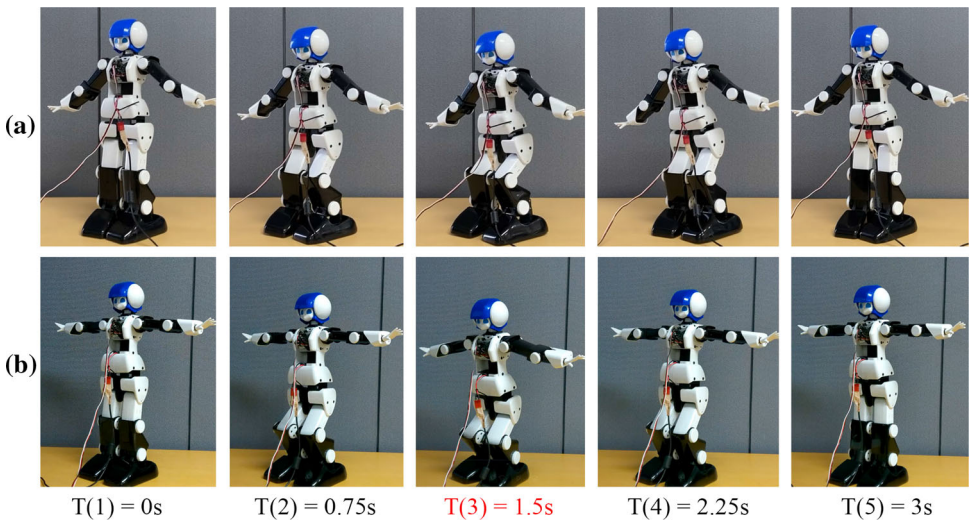


Fig. 14 Comparison of the squatting and standing motion. **a** Based on previous research. **b** Edited realistic robot motion using our proposed framework

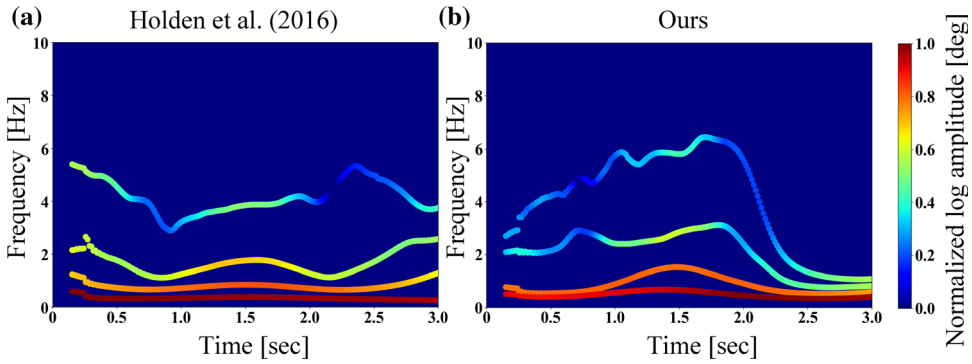


Fig. 15 Hilbert spectrum analysis of the squatting and standing motion (left hip joint). **a** Based on previous research. **b** Edited realistic robot motion using our proposed framework

different from the previous research. For example, the highest frequency (6 Hz) is generated only in T(3), rather than the previous research generated in the beginning. In addition, our proposed motion does not require any referred motions, as were required in the previous research. Thus, although our framework needs realistic feature amplitudes to be decided interactively by designers, our proposed method can efficiently generate high-quality realistic features based on the biomechanical connections between basic motions and realistic features.

7 Conclusion

This study aimed to generate realistic motion features for robots based on human motion. Based on a comparison of keyframe-designed robot and human motion, we first discovered that human motion has larger and more complicated motion primitives than the robot motion. Next, we verified that adding realistic motion features to robots could improve the realism of the basic motion. In addition, we proposed an autoencoder network-based framework to explore the biomechanical relationship between the basic motions and realistic motion features of humans. Based on a series of experiments, we proved that our framework can effectively increase the realism of robot motion and can generate these realistic motions easily. In addition to motion generation, our framework can also contribute to other aspects of motion design, such as character animations in computer graphics. We believe that our research can make humanoid robots interact with humans automatically and self-consistently in the future.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00521-021-06192-3.

Acknowledgements This work was supported by JSPS KAKENHI Grant Number JP20K23352 and the Sasakawa Scientific Research Grant from The Japan Science Society.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ding M, Ikeura R, Mori Y, Mukai T, Hosoe S (2013) Measurement of human body stiffness for lifting-up motion generation using nursing-care assistant robot–RIBA. In Sensors, IEEE. 1–4
2. Borovac B, Gnjatović M, Savić S, Raković M, Nikolić M (2016) Human-like robot marko in the rehabilitation of children with cerebral palsy. New Trend Med Service Robots. 191–203. Springer, Cham
3. Nishiguchi S, Ogawa K, Yoshikawa Y, Chikaraishi T, Hirata O, Ishiguro H (2017) Theatrical approach: designing human-like behaviour in humanoid robots. Robot Autonom Syst 89:158–166
4. Sanzari M, Ntouskos V, Pirri F (2019) Discovery and recognition of motion primitives in human activities. PLoS ONE 14(4):e0214499
5. Okajima S, Tournier M, Alnajjar FS, Hayashibe M, Hasegawa Y, Shimoda S (2018) Generation of human-like movement from symbolized information. Frontiers in neurorobotics 12:43
6. Tomić M, Jovanović K, Chevallereau C, Potkonjak V, Rodić A (2018) Toward optimal mapping of human dual-arm motion to humanoid motion for tasks involving contact with the environment. Int J Adv Rob Syst 15(1):1729881418757377
7. Beaudoin P, Coros S, van de Panne M, Poulin P (2008) Motion-motif graphs. In: Proceedings of the 2008 ACM SIGGRAPH/ Eurographics symposium on computer animation. pp. 117-126
8. Min J, Chai J (2012) Motion graphs++ a compact generative model for semantic motion analysis and synthesis. ACM Trans Graph 31(6):1–12
9. Dong R, Cai D, Asai N (2017) Nonlinear dance motion analysis and motion editing using Hilbert-Huang transform. In: Proceedings of the computer graphics international conference (pp. 1-6)
10. Dong R, Cai D, Ikuno S (2020) Motion capture data analysis in the instantaneous frequency-domain using hilbert-huang transform. Sensors 20(22):6534
11. Wang H, Ho ES, Shum HP, Zhu Z (2019) Spatio-temporal manifold learning for human motions via long-horizon modeling. IEEE Trans Vis Comput Graph
12. Alemi O, Françoise J, Pasquier P (2017) GrooveNet: Real-time music-driven dance movement generation using artificial neural networks. Networks 8(17):26
13. Holden D, Saito J, Komura, T, Joyce T (2015) Learning motion manifolds with convolutional autoencoders. In SIGGRAPH Asia 2015 Technical Briefs, pp. 1-4

14. Holden D, Saito J, Komura T (2016) A deep learning framework for character motion synthesis and editing. ACM Trans Graph 35(4):1–11
15. Holden D, Komura T, Saito J (2017) Phase-functioned neural networks for character control. ACM Trans Graph 36(4):1–13
16. Huang NE, Shen Z, Long SR, Wu MC, Shih HH, Zheng Q, Yen NC, Tung CC, Liu HH (1998) The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proc R Soc London Ser A Math Phys Eng Sci 454(1971):903–995
17. Rilling G, Flandrin P, Gonçalves P, Lilly JM (2007) Bivariate empirical mode decomposition. IEEE Signal Process Lett 14(12):936–939
18. Rehman N, Mandic DP (2009) Empirical mode decomposition for trivariate signals. IEEE Trans Signal Process 58(3):1059–1068
19. Rehman N, Mandic DP (2009) Multivariate empirical mode decomposition. Proc R Soc A Math Phys Eng Sci 466(2117):1291–1302
20. Rehman N, Park C, Huang NE, Mandic DP (2013) EMD via MEMD: multivariate noise-aided computation of standard EMD. Adv Adapt Data Anal 5(02):1350007
21. Huang NE, Shen Z (2014) Hilbert-Huang transform and its applications, 400. World Scientific
22. Bracewell RN (1986) The Fourier transform and its applications. McGraw-Hill, New York
23. PremiadAI - World-class dance communication robot - [Internet], DMM.com. Japanese. Available from: http://robots.dmm.com/robot/premaidai/spec
24. Spong Mark W (2006) Seth Hutchinson, and Mathukumalli Vidyasagar, Robot modeling and control
25. Tokyo Shimbun web. A performance of AI Robot and Hachioji's Kuruma Ningyo Joruri. https://www.tokyo-np.co.jp/article/68132
26. Neuronmocap. Perception neuron 2.0. https://neuronmocap.com/products/
27. Rilling, G., Flandrin, P., and Goncalves, P. (2003, June). On empirical mode decomposition and its algorithms. In IEEE-EURASIP workshop on nonlinear signal and image processing. 3(3): 8–11. NSIP-03, Grado (I)
28. Niu J, Liu Y, Jiang W, Li X, Kuang G (2012) Weighted average frequency algorithm for Hilbert-Huang spectrum and its application to micro-Doppler estimation. IET Radar Sonar Navig 6(7):595–602
29. "KONDO Robot" KRS-2552RHV ICS, Available from: https://kondo-robot.com/product/03067e
30. Winter DA (2009) Biomechanics and motor control of human movement. Wiley, Hoboken
31. Xu, P., Ye, M., Li, X., Liu, Q., Yang, Y., and Ding, J. (2014, November). Dynamic background learning through deep auto-encoder networks. In: Proceedings of the 22nd ACM international conference on Multimedia, 107-116. (2014)
32. Zhang Y, Liang X, Zhang D, Tan M, Xing E (2020) Unsupervised object-level video summarization with online motion auto-encoder. Pattern Recogn Lett 130:376–385
33. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In ICML
34. Dong R, Chen Y, Cai D, Nakagawa S, Higaki T, Asai N (2020) Robot motion design using bunraku emotional expressions-focusing on Jo-Ha-Kyũ in sounds and movements. Adv Robot 34(5):299–312
35. Holden, A deep learning framework for character motion synthesis and editing. http://theorangeduck.com/page/deep-learning-framework-character-motion-synthesis-and-editing
36. CMU. Carnegie-mellon mocap database. http://mocap.cs.cmu.edu/
37. Xia S, Wang C, Chai J, Hodgins J (2015) Realtime style transfer for unlabeled heterogeneous human motion. ACM Trans Graph 34(4):119:1-119:10
38. Ofli F, Chaudhry R, Kurillo G, Vidal R, Bajcsy R (2013) Berkeley mhad: a comprehensive multimodal human action database. Appl Comput Vis. 2013 IEEE Workshop on, 53–60
39. Müller M, Röder T, Clausen, M, EberhardT B, Krüger B, Weber A (2007) Documentation mocap database hdm05. Tech. Rep. CG-2007-2, Universität Bonn, June
40. Robotyuenchi. PremaidAI RCB version dance song list and dance data. https://robotyuenchi.com/dans.html