Embodied Symbol Emergence based on Mimesis Theory

Tetsunari Inamura[*1] Yoshihiko Nakamura[*1*2] Iwaki Toshima[*1] Hiroaki Tanie[*1]

[*1] *Dept. of Mechano-Informatics, Graduate School of Information Science and Technology, The University of Tokyo, JAPAN*

[*2] *CREST program, Japan Science and Technology Corporation, JAPAN*

Abstract

"Mimesis" theory focused in the cognitive science field and "mirror neurons" found in biology field shows that the behavior generation process isn't independent of the behavior cognition process. The generation and cognition processes have a close relationship each other. During the behavioral imitation period, a human being doesn't practice simple joint coordinate transformation, but will acknowledge the parents behavior. It understands the behavior after abstraction as symbols, and will generate one's self behavior. Focusing on these facts, we propose a new method which carries out the behavior cognition and behavior generation processes at the same time. We also propose a mathematical model based on Hidden Markov Models in order to integrate four abilities; 1) symbol emergence, 2) behavior recognition, 3) self behavior generation, and 4) acquiring the motion primitives. Finally, the feasibility of this method is shown through several experiments on a humanoid robot.

1 Introduction

The research of humanoid robots has a long history and has accumulated a substantial amount of literature. The focus of early efforts was mostly on the dynamics and control of bipedial walking motion. Although it has not yet reached the level of a complete solution, with liability and adaptability, the hardware technology has been established for building autonomous humanoids [1][2][3].

Recently, the human behavioral science and intelligence has become conspicuous as a real research issue of robotics. Although the motivation of the artificial intelligence originated there, the physical limitations have forced or justified the researchers to carry on their research in a limited scope of complexity. It would be a major challenge of contemporary robotics to study robotic behaviors and intelligence in the full scale of complexity. This could then mutually share research outcomes and hypotheses with the human behavioral science and human intelligence.

The discovery of mirror neurons[4] has been a notable topic of brain science concerning such issues. Mirror neurons have been found in primates' and humans' brains, which fire when the subject observes a specific behavior and also fire when the subject starts to act in the same manner. It also locates on Broka's area, which has a close relationship with language management. This fact suggests that the behavior recognition process and behavior generation process are combined as the same information processing scheme. This scheme is nothing but a core engine of the symbol manipulation ability. Indeed, in Donald's "Mimesis Theory"[5], he said that symbol manipulation and communicative ability are founded upon behavior imitation, which is integration of behavior recognition and generation. We believe that a paradigm can be proposed taking advantage of the mirror neurons, with considerations of Deacon's contention[6] that language and the brain had caused each other to evolve.

So far, many researchers have tackled with the issues between the imitation learning for humanoids and human intelligence[7][8]. There are some suggestions that module structure of basic motions is needed for the symbolization and representation of complex behavior, such as Schaal's work[7]. In Kuniyoshi's approach[9], robots can reproduce complex behaviors from observation of human demonstration with the abstraction and symbolization, however, it is difficult to be applied to general recognition and reproduction process because of lack of dynamics point of view, that means the robots have to memorize the whole flow of basic behavior. Moreover, the basic behavior modules needed to be designed by developer. Samejima *et al* have proposed an imitation learning framework with symbolization modules[10]. In this case, a premise have been set that sequence of symbol is given from others by communication, thus a certain representation model

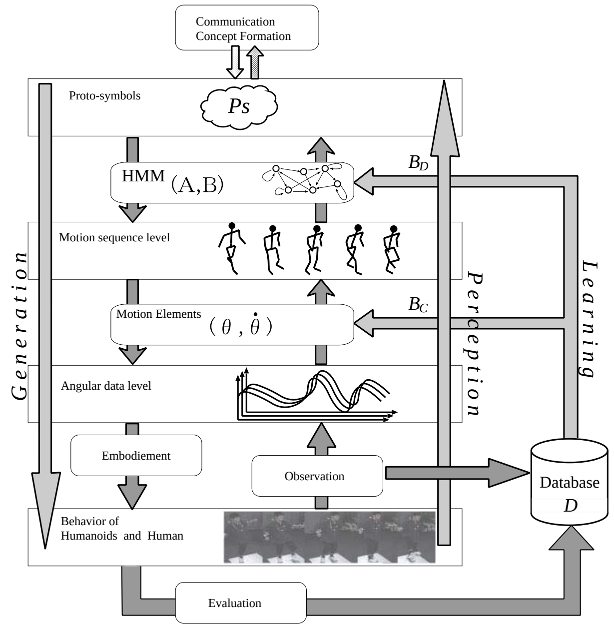

Figure 1: *An outline of proposed mimesis model*

for dynamics of the whole body motion are needed.

In this paper, we propose a mathematical model that abstracts the whole body motions as symbols, generates motion patterns from the symbols, and distinguishes motion patterns based on the symbols. In other words, it is a functional realization of embodied symbol emergence framework which is an inspiration from the mirror neurons and the mimesis theory. Therefore we call the framework as "mimesis model". The purpose of the research is to propose a methodology of mathematical design for the mimesis mode.

One as observer would view a motion pattern of the other as the performer; the observer acquires a symbol of the motion pattern. He recognizes similar motion patterns and even generates it by himself. The observer would then need to modify it from the performer's motion to observer's one according as his own body condition. The model is developed using Hidden Markov Models (HMMs). One issue is to identify appropriate motion primitives that enable both motion recognition and generation. This problem is to be solved using continuous Hidden Markov Models. The second issue is how to generate the motion patterns as time-series of the motion primitives, which is to be solved adopting discrete Hidden Markov Models. The acquired models are to be modified according as the observer's body. This is the third issue and to be discussed as a problem of database managements for HMM.

First, we introduce the mathematical model of mimesis in section 2. In section 3, computational methods for symbol emergence, motion recognition and generation are explained. In section 4, we discuss how to develop and design the motion primitive representation. The conclusion follows experimental results in section 5.

2 Mimesis model which recognize others' motion and generate self-motion

In this section, we explain the outline of mimesis models with showing the difference between usual imitation models.

In an imitation learning framework MOSAIC which has been proposed by Samejima *et al*[10][11], plural dynamics and inverse-dynamics modules for the prediction and control of motion are implemented in order to imitate others' motion. This framework is based on bi-directional theory suggested by Miyamoto and Kawato[12]. Both of them aim to imitate human's behavior and symbolize the motion patterns as motion primitives. One of disadvantages of these methods is that an others' motion is always needed as a reference pattern, because it has no ability of description for dynamics of time-series motion primitives. On the contrary, we aim not only to imitate others' motion but also abstract the time-series motion patterns as symbol representation. It causes a situation in which no reference motion pattern is needed, that is, more flexible for symbol emergence from behavior imitation.

We, here, propose an imitation framework which abstracts the dynamics of the motion as symbol representations, recognizes others' motions, and generates self-motions from the symbol representations. The realization of the framework leads to the implementation of the mirror neuron from engineering point of view.

2.1 Mimesis model based on Hidden Markov Models

The *mimesis model* consists of three parts; perception part, generation part and learning part, as shown in Fig.1. In the perception part, observed motion patterns are analyzed into basic motion primitives, and the dynamics in the sequence of the elements is abstracted as symbol representations.

In the generation part, a sequence of motion elements is decoded from a proto-symbol. However, the generated motion patterns would be inappropriate for real humanoids. For the issue, we introduce the learning part where motion elements are modified based

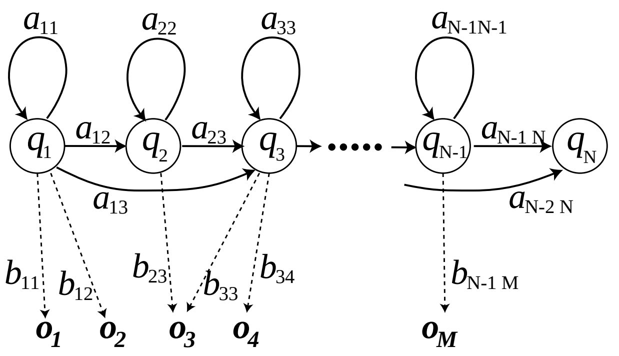

Figure 2: A simple left-to-right type HMMs

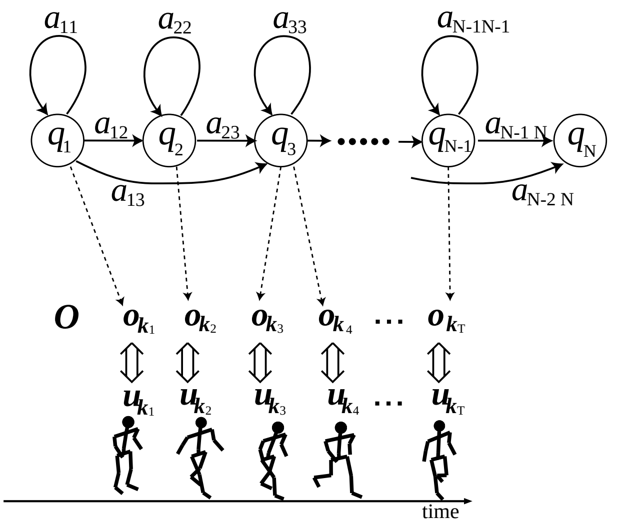

Figure 3: Motion Elements and Hidden Markov Models

on a database consist of performer's motions and observer's motions.

Characteristics needed by the mimesis model is to integrate three functions; motion recognition, motion generation and symbol emergence of motions. We focused on Hidden Markov Models (HMMs) as mathematical backbone for such integration. The HMM is one of stochastic processes which takes time series data as an input, then outputs a probability that the data is generated by the model. The HMM is one of most famous tools as a recognition method for time series data, especially in speech recognition. The HMM divided into two types; discrete HMM and continuous HMM. The former treats sequences of discrete labels, the latter treats sequences of continuous multi-dimensional vectors. In this subsection, we introduce the discrete HMM for the first step. The HMM consists of a finite set of states $\boldsymbol{Q} = \{q_1, \ldots, q_N\}$, a finite set of output label $\boldsymbol{S} = \{o_1, \ldots, o_M\}$, a state transition probability matrix $\boldsymbol{A} = \{a_{ij}\}$, an output probability matrix $\boldsymbol{B} = \{b_{ij}\}$, and an initial distribution vector $\boldsymbol{\pi} = \{\pi_i\}$, that is $\{\boldsymbol{Q}, \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$. In this framework, state transition is performed probabilistically and some labels $\boldsymbol{o}_i$ are output during the transition as shown in Fig.2.

2.2 Motion elements

In order to connect discrete symbol representations and time-series motion data, motion element is introduced. A motion element corresponds to a point in a phase space which consists of joint angle of humanoids, velocity, acceleration, and so on as follows:

$$\boldsymbol{u} \stackrel{\text{def}}{=} \boldsymbol{\mu}. \tag{1}$$

To say more generally, adopting displacement and velocity of base link and hands, and various sensory information, is effective for recognizing and generation of more complex behavior. In this paper, we do not determine the type of physical quantity or concrete target behavior. We stand on the stance that a developer determines the task and physical quantity according to need.

A time-series motion pattern

$$\boldsymbol{O} = [\boldsymbol{o}_{k_1} \boldsymbol{o}_{k_2} \ldots \boldsymbol{o}_{k_T}] \tag{2}$$

$$k_i \in \{1, 2, \ldots, M\} \tag{3}$$

indicates an other's motion and self-motion at the same time in the HMM, by correspondence of the M pieces of motion element $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M)$ to output label o as follows:

$$\boldsymbol{o}_i = \boldsymbol{u}_i, \tag{4}$$

where $\boldsymbol{O}$ indicates a row vector which consists of a sequence of motion elements $\boldsymbol{o}$. The i-th element from the left indicates the motion element at the i-th discrete time. The matter what kind of physical quantity is effective for the model, is affected by the characteristic of target behavior. In this paper, we adopted simple joint angle space as the motion element for the first step, and propose motion recognition, generation and abstract method independent of the type of physical quantity.

2.3 HMMs as a proto-symbol

Definition by Eq.(4) is nothing but a connection between a label (or an index) and feature vector for a certain moment. To represent the dynamics of feature vector sequence, certain representation method are needed. Symbols may be defined in a narrow sense as ones with embodied meaning and their mutual distances. We propose to consider the acquired HMMs as symbols. Although, in the scope of this paper, they have embodied meanings, their mutual distances have not yet been introduced. The authors' plans to discuss it in the future work. In this sense, it would be appropriate to call the HMMs proto-symbols.

We shall concentrate on the HMM parameters. As left-to-right type HMMs as shown in Fig.2 are used in the mimesis model, initial distribution vector $\boldsymbol{\pi}$ have fixed value as $(1, 0, \ldots, 0)$. A set of state $\boldsymbol{Q}$ and a set of output label $\boldsymbol{S}$ have no direct relationship between output time series data. State transitions probability matrix $\boldsymbol{A}$ and output probability matrix $\boldsymbol{B}$ can be regarded as a abstraction parameter of probabilistic dynamics of the HMM.

Thus, we define the proto-symbols as follow

$$\mathcal{P}_{\mathcal{S}} \stackrel{\text{def}}{=} \{\boldsymbol{A}, \boldsymbol{B}\}. \tag{5}$$

The Hidden Markov Model is stochastic mathematical framework for sequential data. It is furnished with well established algorithms of computation. The acquisition, recognition and generation of motion patterns are to be efficiently computed using the algorithms. It is also known that HMMs are successfully used in speech recognition.

An alternative of HMMs for such computation is the use of recurrent neural networks (RNNs). RNNs also memorize dynamics of patterns [13][14][15] [16]. The authors tested the use of RNNs for motion recognition and generation[14]. According to the result of[14], more than 500 nodes and more than 200,000 weight parameters between each node are needed in order to integrate the memorization and generation process on the same RNN. The RNN consists of motion element neurons, symbol representation neurons and buffer neurons for treating time-series data. The required number of weights increases in proportion to the square of the number of all nodes. On the contrary, the number of parameters used in HMMs is proportional to the product of the number of nodes and motion elements. To give a concretely example, a HMM consists of 25 nodes and 80 motion elements requires about 2,500 parameters, in order to recognize and generate the motion. Therefore, the drawback of RNNs is in the low efficiency of computation RNNs would use a large set of parameters to memorize a few motion patterns. The parameters would require a large computation to be adjusted.

3 Motion abstract, recognition and generation using HMMs

3.1 Creating proto-symbols through observation

Motion abstraction, that is proto-symbol generation, consists of two phases. In the first phase, observed motions are transferred into sequence of the motion elements by a segmentation process. In the second phase, dynamics exists in the motion elements sequence is abstracted, and represented as proto-symbols.

In order to transform the observed motion pattern $\boldsymbol{\Theta}(t)$ into a sequence of motion elements $\boldsymbol{O} = [\boldsymbol{o}_{k_1}\boldsymbol{o}_{k_2}\ldots\boldsymbol{o}_{k_T}]$, $\boldsymbol{\theta}$ for each short time period is sampled, then

$$j = \arg\max_i \frac{\exp\{-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu}_i)\}}{\sqrt{(2\pi)^D \det \boldsymbol{\Sigma}_i}} \tag{6}$$

is calculated. The meaning of above equation is letting j be i which causes the maximum value of the right side. Where D is the number of dimension of the motion elements, det indicates the determinant of a matrix, ^{T} indicates a transpose of matrix. The right side represents a Gaussian distribution function with a covariance matrix $\boldsymbol{\Sigma}$ and a mean vector $\boldsymbol{\mu}$. The calculation contributes to selecting a suitable motion element $\boldsymbol{u}_j$ which locates near by the sampled motion $\boldsymbol{x}$ in the phase space. Let the $\boldsymbol{u}_j$ be a motion element for each short time period. $[\boldsymbol{u}_{k_1}\boldsymbol{u}_{k_2}\ldots\boldsymbol{u}_{k_T}]$, namely a sequence of motion elements, is output by a repetition of above calculation for all short time period.

After that, a parameter of a HMM ($\{\boldsymbol{A}, \boldsymbol{B}\}$) which output the sequential elements plausibly is calculated and registered as a proto-symbol $\mathcal{P}_{\mathcal{S}}$. Humanoids gather several motion patterns as a stock of observed data for the learning. In the case of an unknown motion is input, the robot creates a new HMM. $\boldsymbol{A}$ and $\boldsymbol{B}$ can be calculated by Baum-Welch algorithm which is one of EM-algorithms[17].

3.2 Motion recognition using proto-symbols

To recognize others' motion, observed motions are transformed into sequence of motion elements $\boldsymbol{O} = [\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t]$, and a parameter $P(\boldsymbol{O}|\mathcal{P}_{\mathcal{S}})$ is calculated. This parameter indicates a probability that a motion pattern $\boldsymbol{O}$ is generated by a proto-symbol $\mathcal{P}_{\mathcal{S}}$. This value is called as likelihood, calculated by forward algorithm[17].

Each proto-symbol is corresponded to each motion, thus, likelihood values of the input motion against each proto-symbol are calculated. A proto-symbol which corresponds to input pattern should indicate high likelihood, and other proto-symbols ought to indicate low likelihood. In order to distinguish these likelihoods, following criterion is introduced

$$R(\boldsymbol{O}) = \log \frac{\max\{P(\boldsymbol{O}|\mathcal{P}_{\mathcal{S}i})\}}{\mathrm{second}\{P(\boldsymbol{O}|\mathcal{P}_{\mathcal{S}i})\}}, \tag{7}$$

where $\mathrm{second}(\boldsymbol{x})$ means that the second highest value in the components of $\boldsymbol{x}$. Mimesis model recognizes the input motion without any confusion when the R indicates high value. In this case, the recognition result becomes $\mathcal{P}_{\mathcal{S}j}$, where

$$j = \arg\max_i \{P(\boldsymbol{O}|\mathcal{P}_{\mathcal{S}i})\}. \tag{8}$$

When the R indicated low value, the recognition was failed and mimesis model tries to shift to proto-symbol creation phase.

3.3 Motion generation using proto-symbols

Basically, original patterns are decoded using expectation operator in stochastic model, however, applying the expectation operator is difficult in the HMM. The HMM has a two-stage stochastic process; state transition and label output. Applying expectation operator is simple for the latter process, however, difficult for the former process. The results of recurrent state transition would not fit on the same dimensional phase space. For example, the length of a state sequence changes every trial. It means that integration of the probability values could not executed holomorphically. Therefore, we adopted the averaging method over repetition of motion generation. The detailed order of the generation is as follows.

1. Initialization : Let the starting node be q_1, the node token be $i = 1$, motion elements sequence be $\boldsymbol{O} = \phi$.
2. Deciding the transition destination node q_j using transition matrix $\boldsymbol{A}$ stochastically.
3. Deciding the output label $\boldsymbol{o}_{k_t}$ during the transition from node q_i to q_j stochastically using output matrix $\boldsymbol{B}$.
4. Adding the output label $\boldsymbol{o}_{k_t}$ to the motion elements sequence $\boldsymbol{O}$. $\boldsymbol{O} := [\boldsymbol{O}\ \ \boldsymbol{o}_{k_t}]$.
5. Let the generation process be stopped when the token reach the end node q_N. Or, returns to the step (2) with letting $i := j, t := t + 1$.
6. Finally, the sequential motion elements are transformed into continuous joint angle representations.

The output motions using above operations are not the same, but have different time length and order of motion elements, because the output operations are stochastic. However, it is possible to generate an approximate motion pattern because the parameters $\boldsymbol{A}$ and $\boldsymbol{B}$ represent the abstraction of dynamics in the motion pattern. Therefore, above operations are repeated, and plural generated motions are averaged. As the time length of each generated motions are different, makes the time length uniform using

$$\theta'(t) = \theta\Big(T\frac{t}{T_u}\Big) \tag{9}$$

where T is the time length of each motion, T_u is the time length of the uniformed motion. After that, each joint angle are averaged.

Several researches already proposed motion recognition methods based on the HMM[18][19] [20][21][22], however, no research has been existed in which motion is generated from the HMM. Masuko *et al*[23][24] have been proposed a speech parameter generation method using the HMM, however, the generation process is not opposite direction of the speech recognition process. The most important characteristic of our method is that the motion recognition and motion generation process are integrated by only single HMM.

4 Development of motion elements through repetition of motion observation and generation

The performance of motion recognition and generation is influenced by the characteristic of motion elements. If the number of elements were too little, the generation would be filed. If the motion elements had no relationship between the observed motions, the recognition process would be failed. Therefore, we adopted an approach that the system searches the best motion elements with an evaluation criterion whether the generated motion would be fit for the body and the recognition would be succeeded against familiar motion. Using the method, the humanoid can acquire adequate motion elements through repetition of motion perception and generation.

4.1 Introduction of Continuous HMMs and applying to mimesis model

In this phase, we introduce continuous HMMs[17], which can treat continuous multi-dimensional data. The difference between normal discrete HMMs (DHMMs) and continuous HMMs (CHMMs) is that the transition process outputs continuous multi-dimensional vectors, different from the DHMMs in which the discrete labels are output ,shown in Fig.4.

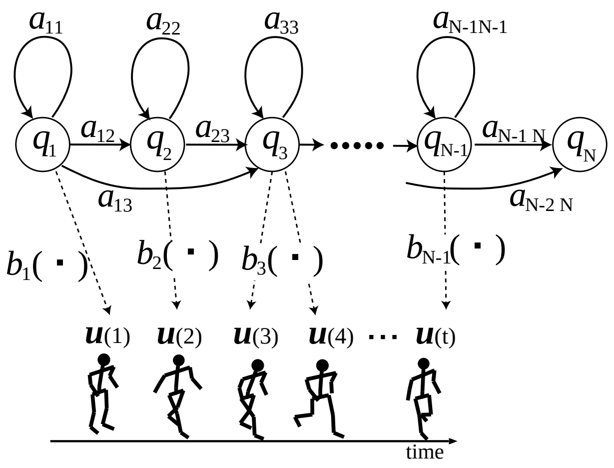

Figure 4: *A continuous Hidden Markov Models*

On the CHMMs, output probability matrix $\boldsymbol{B}$ becomes probability density functions in CHMMs. Here, the density function is approximated with linear combination of Gaussian functions as follows:

$$P_i(\boldsymbol{o}) = \sum_{j=1}^{m} c_{ij}\mathcal{N}_{ij}(\boldsymbol{o};\boldsymbol{\Sigma},\boldsymbol{\mu}), \tag{10}$$

where $P_i(\boldsymbol{o})$ is probability density function for output of continuous vector $\boldsymbol{o}$ at i-th state node, m is the number of mixture Gaussian functions, c_{ij} is mixture coefficient, $\mathcal{N}(\boldsymbol{o};\boldsymbol{\Sigma},\boldsymbol{\mu})$ is the Gaussian function:

$$\mathcal{N}_{ij}(\boldsymbol{o};\boldsymbol{\Sigma},\boldsymbol{\mu}) = \frac{\exp\left\{-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\mu}_{ij})^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu}_{ij})\right\}}{\sqrt{(2\pi)^D \det \boldsymbol{\Sigma}_{ij}}} \tag{11}$$

where $\boldsymbol{\Sigma}$ is covariance matrix, $\boldsymbol{\mu}$ is mean vector, and D is the number of dimension of continuous vector $\boldsymbol{o}$.

The characteristic of the CHMMs are decided by parameter $\{\boldsymbol{\pi},\boldsymbol{A},\boldsymbol{c},\boldsymbol{\Sigma},\boldsymbol{\mu}\}$. These parameters are calculated using Baum-Welch algorithm .

Here, each mean vector of the Gaussian function is regarded as important representation of the observed motion. Therefore, we divide the parameters of CHMMs, and redefine the *motion elements* $\boldsymbol{e}$ as follows.

$$\boldsymbol{u}_i \stackrel{\text{def}}{=} \{\boldsymbol{\Sigma}_i,\boldsymbol{\mu}_i\} \tag{12}$$

In other words, the number of motion elements is as many as the number of mixture Gaussian components. An important issue is that the motion elements are automatically calculated by the Baum-Welch algorithm as well as mentioned in Section 3.1.

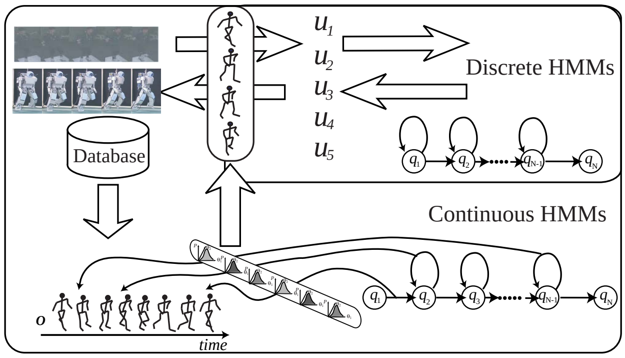

Figure 5: *A model of Hybrid Hidden Markov Model*

Motion elements could be regarded as filter between continuous motion representation and discrete motion representation. When continuous motion would be transferred into discrete motion, $\boldsymbol{e}_i$ where $i = \arg\max_j \mathcal{N}_j(\boldsymbol{o})$, is adopted as a typical motion element for each time period. When discrete motion would be transfered into continuous motion, sequence of $\boldsymbol{\mu}_i$ is used directly.

To sum up, mimesis system can take following advantages with CHMMs.

- Motion elements are able to express the whole body motion, therefore, various motion patterns are available easily.
- Parameters of motion elements are automatically calculated.

4.2 Hybrid Hidden Markov Model

Although many advantages are available, CHMMs have a disadvantage that huge computational quantity is needed. It should take much time for motion generation and recognition. Therefore we propose a hybrid Hidden Markov Model which consists of CHMMs and DHMMs as shown in Fig.5.

In motion recognition and generation phase, DHMMs are used which computational quantity is little. In motion elements acquisition phase, CHMMs are used which computational quantity is large.

4.3 Closing the mimesis loop for embodiment

Parameters which decide the characteristic of HMMs and motion elements are acquired using Baum-Welch algorithm [25] which is a kind of EM algorithm.

This algorithm can be expressed by following equations:

$$\mathcal{D} = \{\boldsymbol{O}^1, \boldsymbol{O}^2, \ldots, \boldsymbol{O}^l\} \tag{13}$$

$$\{\boldsymbol{A}, \boldsymbol{B}\} := \mathcal{B}_D(\mathcal{D}) \tag{14}$$

$$\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\} := \mathcal{B}_C(\mathcal{D}), \tag{15}$$

where $\mathcal{B}_D, \mathcal{B}_C$ are operations using the Baum-Welch algorithm, $\mathcal{D}$ is a database consists of l observations as Initial database $\mathcal{D}^0$ consists of only observed others' motions, that is, motion elements and proto-symbols which have no relationship between learner's physical characteristic are acquired by above operations. Therefore, let the proto-symbols and motion elements be acquired with database manipulation during repetitions of motion recognition and generation as follows.

1. Generating a motion $\boldsymbol{O}$ from a proto-symbol $\mathcal{P}_S$ and motion elements.
2. Judging whether the generated motion $\boldsymbol{O}$ is suitable or not.
3. Adding the motion to the database when the result of judge is good. $\mathcal{D}^{t+1} := \mathcal{D}^t \cup \boldsymbol{O}$
4. Acquiring the proto-symbols and motion elements using above Eq. (14)(15), and returns back to step (1).

For the evaluation at the step (2), two evaluation criteria were introduced; an inner evaluation for checking the characteristic of proto-symbol, and an outside evaluation for checking the aim and meaning of the motion from point of teacher's view. For the outside evaluation, we prepared following criterion

$$E_\theta = \frac{1}{T}\int_0^T |\boldsymbol{\theta}_{in}(t) - \boldsymbol{\theta}_{out}(t)| dt, \tag{16}$$

where $\boldsymbol{\theta}_{in}(t)$ and $\boldsymbol{\theta}_{out}(t)$ indicate the joint angle of an observed ideal motion and a generated motion. For the inner evaluation, recognition rate $R(\boldsymbol{O})$ explained in Section 3.2 is used.

Considering above two criterion, following integrated criterion is used for the experiment;

$$V = \alpha E_\theta + \beta R^{-1}(\boldsymbol{O}), \tag{17}$$

where α and β is a certain constant. When the value V is larger than a certain threshold, the mimesis model judges that i-th motion data is suitable for recognition process, adds the motion data into database, and calculates the motion elements again. These constant and threshold is adjusted according to each experiment case.

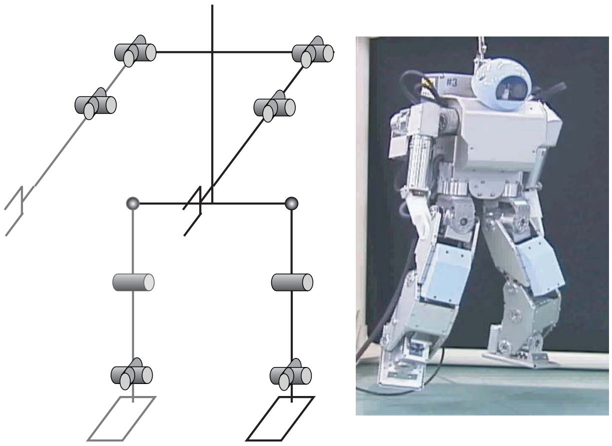

Figure 6: *Humanoid HOAP-1*

At the step (3), it is desirable that the generated self-motion and observed others' motions are distinguished. As the motion elements are used for both motion recognition and generation, simple distinction leads to a deterioration of the process. Thus, a distinction strategy has been introduced that generated self-motions are stored into the database vividly, observed others' motions are stored dimly. Due to the distinction method, the influence of initial others' motion would be decrease, and the database would be gradually under the control of generated self-motions. Actually, vivid motions are stored with little variance and dim motions are stored with large variance. In the learning phase, the number of motion samples in the database is controlled using the variance value.

5 Experiments of motion elements acquisition

A humanoid used in experiments is shown in Fig.6. The humanoid has four degrees of freedom at each arm, six degrees of freedom at each leg, namely 20 degrees of freedom for the whole body. We have confirmed the performance of our method by experiments where the mimesis model observes humans' motion and generate motions for a real humanoid. Using by the *Behavior Capturing System* [26], joint angle data for 20 degrees of freedom are directly observed because the degrees of freedom of the humanoid is 20. The time period of each motion is about 2[sec] with sampling time 20[msec].

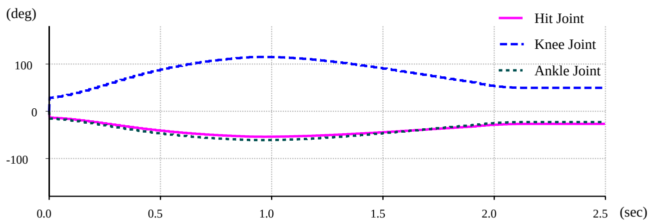

Figure 7: *Original motion pattern*

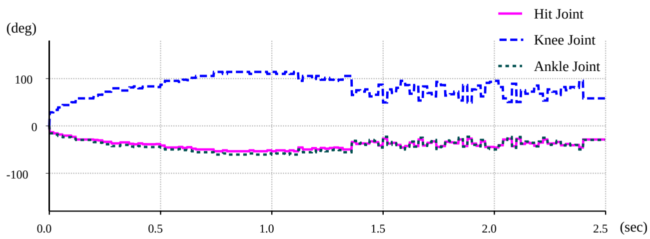

Figure 8: *A Motion pattern using only one time generation*

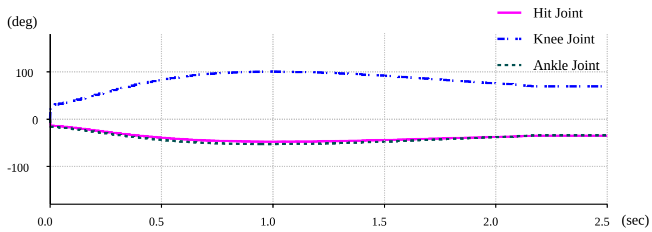

Figure 9: *A Motion pattern using 1000 times generation*

5.1 Experiments of motion generation

A humanoid used in this experiment has 20 degrees of freedom. We investigated basic performance for squat behaviors. In the squat behavior, characteristic motion collected around the lower body. Therefore, we adopted simple motion element which consists of three joint angles, that is, hip (pitch axis), knee and ankle (pitch axis). In this subsection, several experiments are performed by the simple motion element.

Figure 8 shows the output motion pattern by one time output operation. Comparing with an original motion pattern used in the learning (Fig.7), approximately pattern is generated but noise was awfully arisen. The cause of the noise is that the discrete motion elements are selected at each moment stochastically, thus the coarse and the discontinuity was stood out.

Fig.9 shows the output motion pattern after 1000 times operation explained in Section 3.3. A CG animation using the pattern is shown in Fig.14. Comparing with the motion pattern by one time operation (Fig.8), the joint angle became to be smoothed. There were some joint angle errors between the original patterns. We think that the cause of the error is the influence of coarse discrete motion elements.

The computational time for the generation process was about 1[sec] using Pentium-III 1[GHz] processor. The time is enough fast as the off-line pattern generator for humanoids. For this sort of problems, Okada *et al* have been proposed a compression method[27] in which a motion pattern of humanoids which have over 20 degrees of freedom is transferred into a three-dimensional vector We think that reduction of the computational cost can be performed with adopting the method.

5.2 Experiments of motion recognition

For the motion recognition experiments, seven behaviors; (a) tennis swing (swing), (b) walking (walk), (c) Cossack dancing (dance), (d) kicking (kick), (e) backward walking (back), (f) crawling (crawl), are prepared as shown in Fig.10. The behavior from (a) to (e) were treated as already-known motions, the behavior (g) was treated as a unknown motion. Table 1 is the recognition result.

The value in the table indicate the logarithm of likelihood $P(\boldsymbol{O}|\boldsymbol{A},\boldsymbol{B})$. Proto-symbols arranged lengthways indicate the target of the recognition, and behavior names arranged sideways indicate the proto-symbols already learnt. The value indicates larger, the target motion matches better with the proto-symbol. The value of a certain target motion against a proto-symbol which is corresponds to the motion indicates high, that is located on a diagonal line. The values of unfamiliar motion (unknown) against each proto-symbol are almost the same. Thus we see that the recognition process would be succeeded without mistake, when recognition rate R is set to about 1000 empirically.

5.3 Experiments of motion elements acquisition

For this experiment, four kinds of motions; walking, squat, picking up, and Cossack dancing, were recorded. The dimension of motion elements is three; hip joint (pitch), knee and ankle joint (pitch), as well as Section 5.1

After the 50 observations, motion generation process is executed 50 times, and appropriate motions are

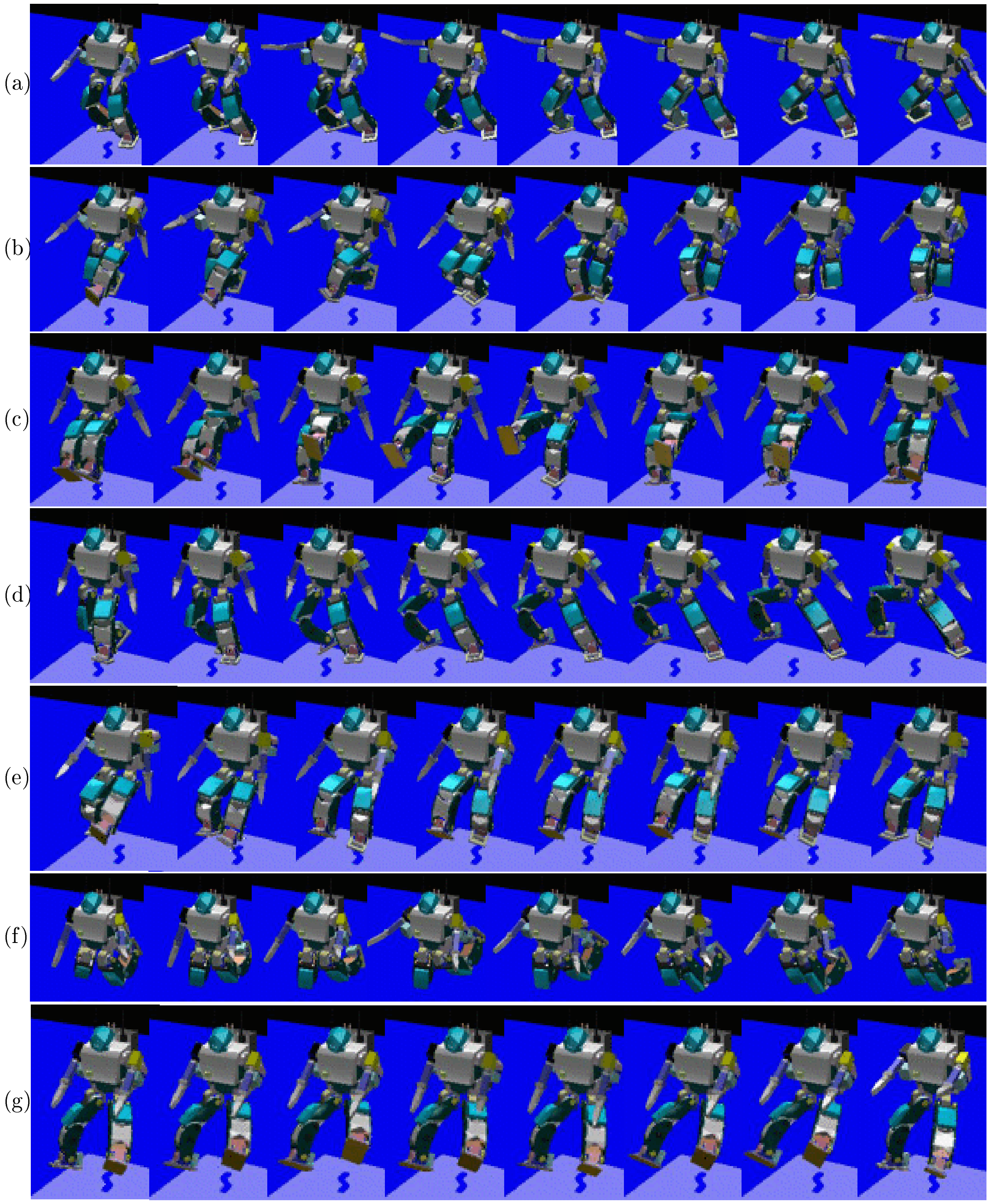

Figure 10: *Target behaviors (a) tennis swing, (b) walking, (c) Cossack dance, (d) kicking, (e) backward walking, (f) crawling, (g) unknown behavior.*

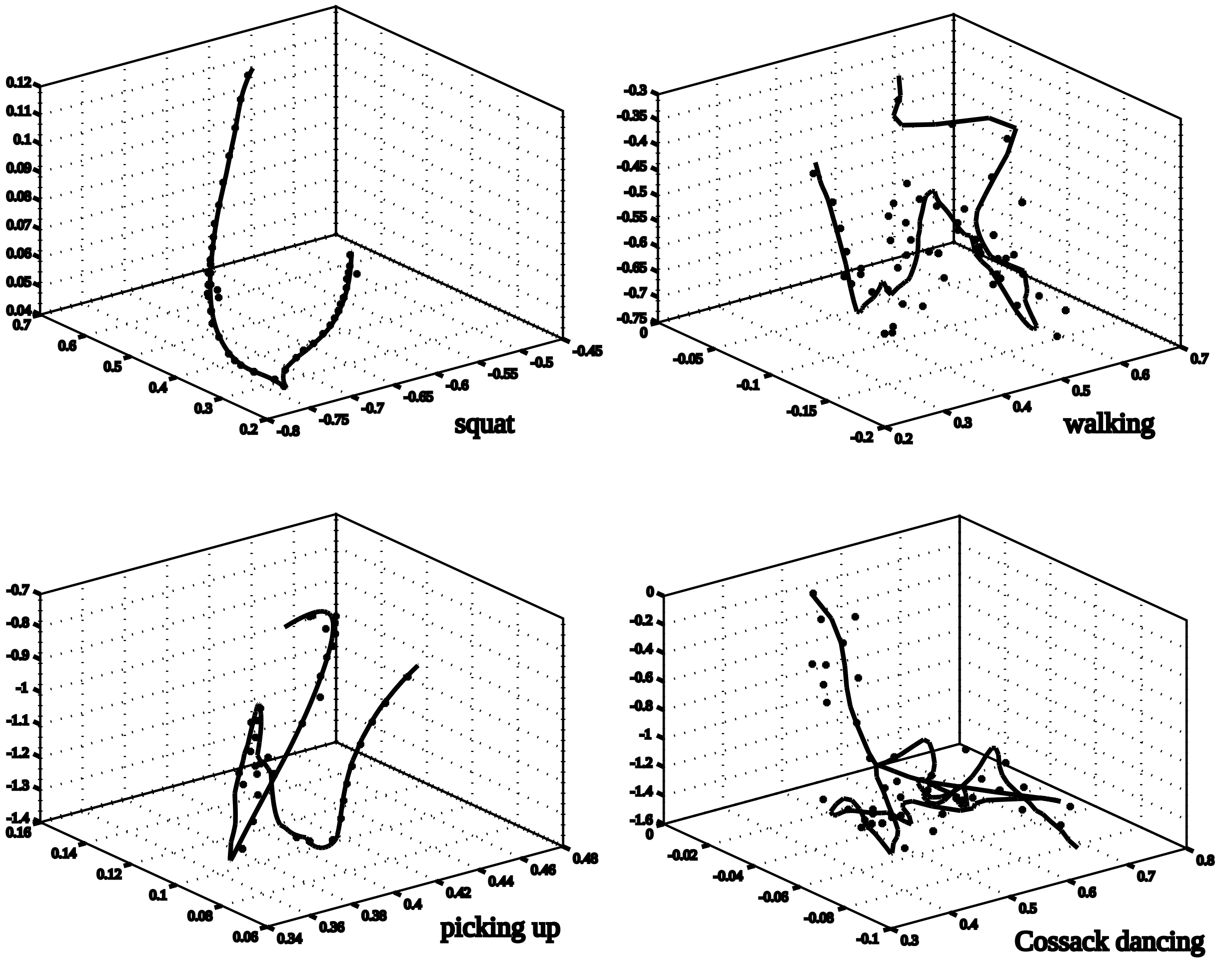

Figure 11: *A result of motion elements acquisition against four kinds of motion*

Table 1: *Recognition result of others' motion using HMMs.*

Input Behavior	Proto-symbols						
	swinging	walking	dancing	kicking	backward walking	crawling	R
swing	-430	-3915	-4077	-3940	-4114	-4007	3485
walking	-3048	-225	-3071	-1646	-3099	-3019	1420
dance	-1656	-1603	-144	-1613	-1683	-1577	1433
kicking	-2543	-1574	-2562	-199	-2585	-2519	1374
backward walking	-2395	-2318	-2413	-2332	-202	-2372	2117
crawling	-4083	-3950	-3815	-3976	-4151	-488	3327
unknown behavior	-1915	-1853	-1928	-1865	-1946	-1896	11

Figure 12: *Motion Capturing System: step motion for learning data*

added into database. Figure 11 shows the acquired motion elements. Each dot mark indicate the motion element, solid line indicate the original motion's trajectory. As the figure shows, the motion elements are located near the original motion, that is, our method shows good performance.

5.4 Experiments of motion elements development based on embodiment

Here, we set up a situation where the joint angle limitation of the humanoid's knee is about 40[deg], less than humans' one. We investigated whether motion elements for the humanoid are acquired by observations of humans' motions under such a condition. In the experiment, 80 times loop are repeated as explained in Section 4.3.

Figure 12 shows the original motion which is performed by a human. Figure 15 shows the acquired motion elements from the performance when the joint angle limitation is not existed. A result with the limitation condition is shown in Fig.16. In these figures, three axes indicate hip joint (pitch), knee joint and ankle joint (pitch), as well as Section 5.1. The curved line in the figures corresponds to the motion trajectory. The dots indicate acquired motion elements. As Fig.15 indicates, the motion elements are located near by the original motion trajectory. Comparing with the Fig.15, motion elements are gathered not only on the A area, but also on the B area in Fig.16. These motion elements located on the B area is acquired by the generated self-motions in the database, which fits for the humanoid embodiment. This result shows that the both motion elements are acquired; elements for the recognition of others' motions (A area) and ones for the generation of self-motion (B area).

5.5 Designing of HMMs

Here, We shall concentrate on the rest parameter, namely structure of HMMs. As the HMMs adopted in this paper are left-to-right type, the rest parameter is the number of nodes. It is possible to use the evaluation criterion explained in Section 4.3 for investigation of the number of nodes, during the repetition of motion recognition and generation.

Swing of tennis is selected for the experiment, the error value E_θ is measured with changing the number of nodes from 10 to 40. The result if shown in Fig.18. As the diagram indicates, the error value decreases hardly where the number of nodes shifts from 24 to 25. Figure 17 shows the generated motion pattern for four conditions; the number of nodes is 20, 24, 25 and 40, respectively. The diagram focused on the right

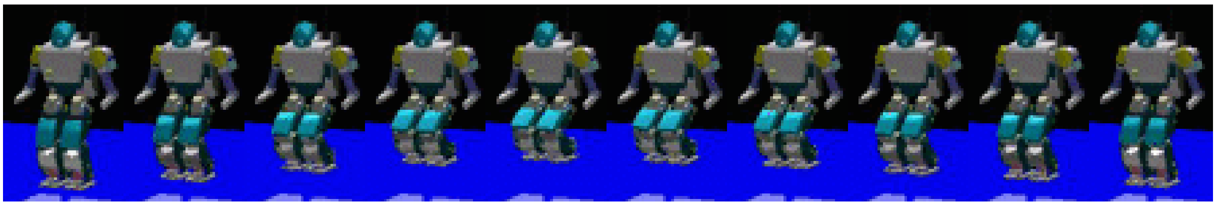

Figure 13: *Original motion for proto-symbol creation*

Figure 14: *Generated motion from proto-symbol*

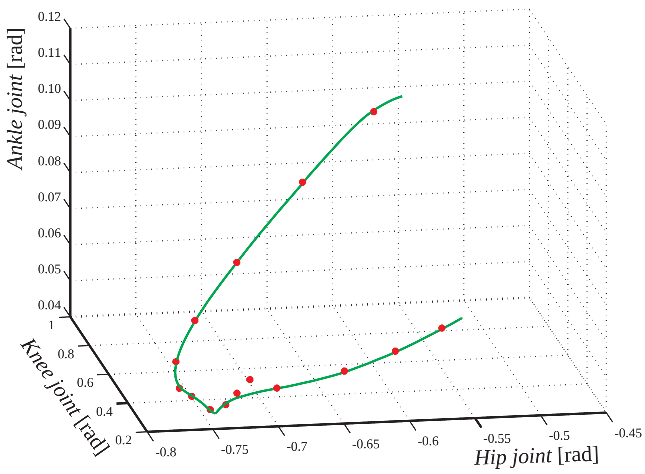

Figure 15: *Acquired motion elements without loop structure*

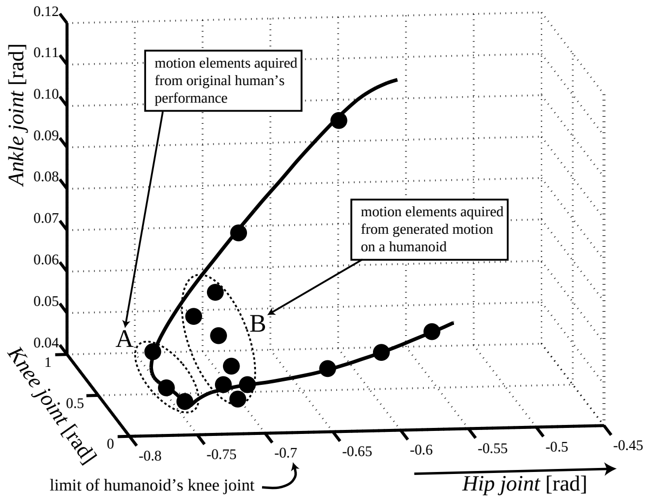

Figure 16: *Acquired embodied self-motion elements using loop structure*

shoulder's yaw joint. Solid lines indicate generated motion pattern, broken lines indicate original motion pattern. The diagram supports the result that the desirable number of nodes is above 25.

6 Conclusions

In this paper, we proposed a framework named "mimesis model" which integrates motion recognition/generation and symbolization of motion patterns based on mimesis theory. In our mimesis model, proto-symbols and motion elements are introduced with Hidden Markov Models in order to integrate following three ability using only one mathematical model; (1) abstraction of motion patterns and symbol representation , (2) generation of self-motions from the symbol representation, and (3) recognition of others' motions using the symbol representation. Through experiences, the feasibility of the mimesis model is cleared. Furthermore, we proposed an approach in

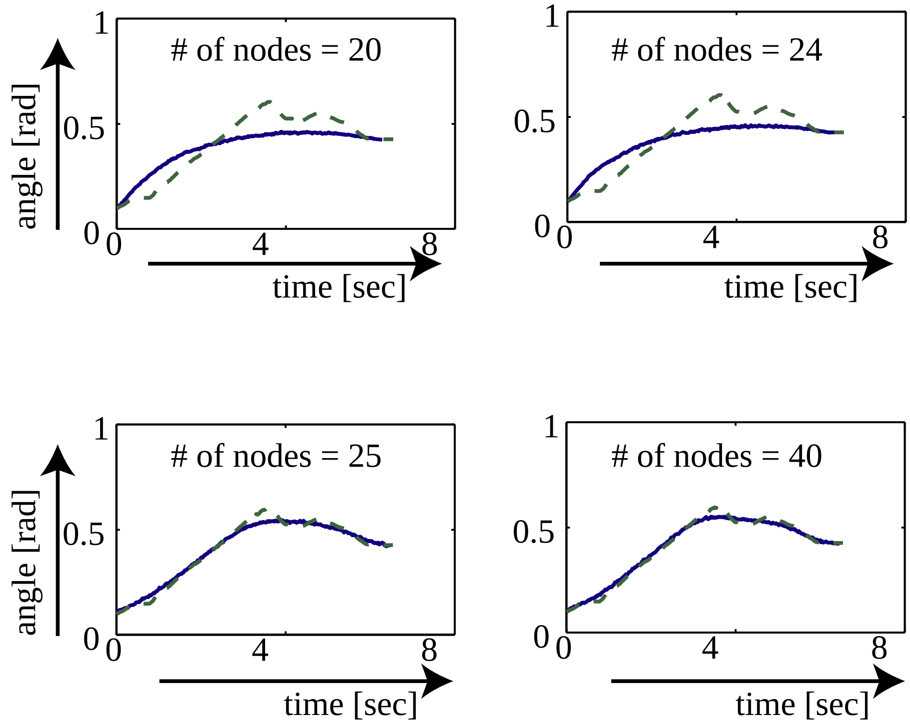

Figure 17: *Generated motion (shoulder's yaw joint) for each number of nodes*

which the development of motion elements is resulted as the management of motion database, and investigated the effectiveness through an experiment in which the learner's physical body condition is different from the teacher's one.

The mimesis model is not simple method for motion recognition, generation, abstraction. The recognition process which transfers an observed others' motion into proto-symbol representation, and the generation process which transfers a proto-symbol representation into self-motions is implemented as opposite direction function by only one mathematical model. The most important characteristic is integration between imitation learning and symbol emergence is established with defining the bidirectional computation model as proto-symbols.

In the current stage, the proposed model can be applied to simple motion patterns; however, the application of the method to complex behavior is difficult, because consideration of external environment is needed such as tracking an object by eyes, throwing a ball, and so on. It is desirable that some abstracted behavior units are designed, and HMMs are applied to such behavior units. For the issue, we plan to construct hierarchical HMMs in order to be applied to from simple motion level to complex behavior level.

We think this result is the first step to connect language development process to the motion acquisition process using the mimesis model, for instance, humanoids try to make communications with others, and build a relationship representation between proto-symbols and linguistic symbols. For such direction, we try to define the distance between each HMM and establish computational method in order for the proto-symbols to evolve into general symbols. We believe that this approach leads to build an intelligent system which connects humanoids intelligence and behavior science.

Acknowledgement

This research was supported by the "Robot Brain Project" under the Core Research for Evolutional Science and Technology (CREST program) of the Japan Science and Technology Corporation. The authors would like to thank to Mr. K.Kurihara for the use of *Behavior Capturing System*.

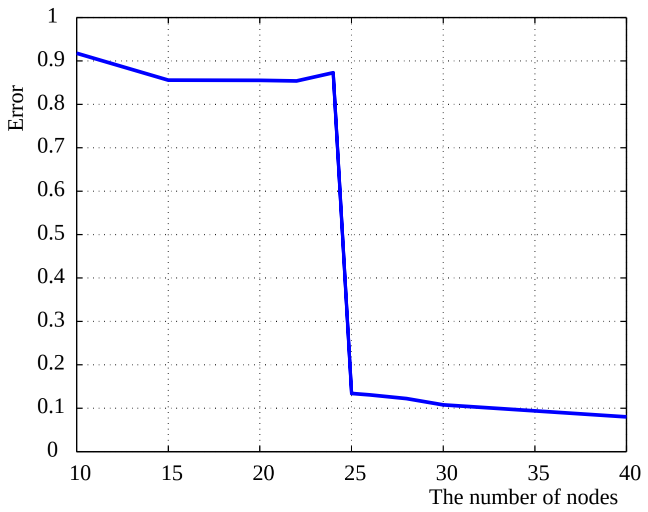

Figure 18: *Error value E_θ and the number of nodes*

References

[1] K. Nishiwaki, T. Sugihara, S. Kagami, F. Kanehiro, M. Inaba, and H. Inoue. Design and development of research platform for perception-action integration in humanoid robot: H6. In *Proc*, pp. 1559–1564, 2000.

[2] Y. Kuroki, T. Ishida, J. Yamaguchi, M. Fujita, and T. Doi. A small biped entertainment robot. In *Proc. of IEEE-RAS Int'l Conference on Humanoid Robots*, pp. 181–186, 2001.

[3] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of honda humanoid robot. In *Proceedings of IEEE International Conference on Robotics & Automation*, pp. 1321–1326, 1998.

[4] V. Gallese and A. Goldman. Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, Vol. 2, No. 12, pp. 493–501, 1998.

[5] Merlin Donald. *Origins of the Modern Mind*. Harvard University Press, Cambridge, 1991.

[6] Terrence W. Deacon. *The symbolic species*. W.W. Norton & Company. Inc., 1997.

[7] Stefan Schaal. Is imitation learning the way to humanoid robots? *Trends in Cognitive Sciences*, Vol. 3, No. 6, pp. 233–242, 1999.

[8] Maja J Mataric. Getting humanoids to move and imitate. *IEEE Intelligent Systems*, pp. 18–24, 2000.

[9] Yasuo Kuniyoshi, Masayuki Inaba, and Hirochika Inoue. Learning by Watching: Extracting Reusable Task Knowledge from Visual Observation of Human Performance. *IEEE Transaction on Robotics and Automation*, Vol. 10, No. 6, pp. 799–822, 1994.

[10] Kazuyuki Samejima, Ken'ichi Katagiri, Kenji Doya, and Mitsuo Kawato. Symbolization and imitation learning of motion sequence using competitive modules. *The trans. of the institute of electronics, information and communication engineers*, Vol. J85-D-II, No. 1, pp. 90–100, 2002.

[11] K. Samejima, K. Doya, and M. Kawato. Intra-module credit assignment in multiple model-based reinforcement learning. *Neural Networks*, 2003.

[12] Hiroyuki Miyamoto and Mitsuo Kawato. A tennis serve and upswing learning robot based on bi-directional theory. *Neural Networks*, Vol. 11, pp. 1331–1344, 1998.

[13] Jun Tani. On the dynamics of robot exploration learning. *Cognitive Systems Research*, Vol. 3, No. 3, pp. 459–470, 2002.

[14] Tetsunari Inamura, Yoshihiko Nakamura, and Moriaki Simozaki. Associative computational model of mirror neurons that connects missing link between behaviors and symbols. In *Proc. of Int'l Conf. on Intelligent Robots and Systems*, pp. 1032–1037, 2002.

[15] Masahiko Morita. Memory and learning of sequential patterns by nonmonotone neural networks. *Neural Networks*, Vol. 9, No. 8, pp. 1477–1489, 1996.

[16] Masahiko Morita and Satoshi Murakami. Recognition of spatiotemporal patterns by nonmonotone neural networks. In *Proceedings of the 1997 International Conference on Neural Information Processing*, Vol. 1, pp. 6–9, 1997.

[17] Steve Young et al. *The HTK Book*. Microsoft Corporation, 2000.

[18] Koichi Ogawara, Jun Takamatsu, Hiroshi Kimura, and Katsushi Ikeuchi. Modeling manipulation interactions by hidden markov models. In *Proc. of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1096–1101, 2002.

[19] Takahide Yoshiike, Atsushi Konno, Koichi Nagashima, Masayuki Inaba, and Hirochika Inoue. On-line recognition and mimicking of human posture. In *Proceedings of the 3rd International Conference on Advanced Mechatronics*, pp. 430–435, Aug 1998.

[20] Toshikazu Wada and Takashi Matsuyama. Appearance based behavior recognition by event driven selective attention. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 759–764, 1998.

[21] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 379–385, 1992.

[22] P.K. Pook and D.H. Ballard. Recognizing teleoperated manipulations. In *IEEE International Conference on Robotics and Automation*, pp. 578–585, 1993.

[23] T. Kobayashi T. Masuko, K. Tokuda and S. Imai. Speech synthesis from hmms using dynamic features.

In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, pp. 389–392, 1996.

[24] Satoshi Imai Keiichi Tokuda, Takao Kobayashi. Speech parameter generation from hmm using dynamic features. In *Proc. of ICASSP95*, pp. 660–663, 1995.

[25] S.Young et al. *The HTK Book*. Microsoft Corporation, 2000.

[26] Kazutaka Kurihara, Shin'ichiro Hoshino, Katsu Yamane, and Yoshihiko Nakamura. Optical motion capture system with pan-tilt camera tracking and realtime data processing. In *the Proc. of IEEE Int'l Conf. on Robotics & Automation*, pp. 1241–1248, 2002.

[27] Masafumi Okada, Koji Tatani, and Yoshihiko Nakamura. Polynomial design of the nonlinear dynamics for the brain-like information processing of whole body motion. In *Proc. of IEEE International Conference on Robotics and Automation*, pp. 1410–1415, 2002.

A Appendix

A.1 Notation

N	The number of nodes
M	The number of motion elements
T	The length of observation sequence
R	The number of motions in database
$\boldsymbol{O}$	a sequence of observations
$\boldsymbol{o}_t$	the observation at time t
a_{ij}	the probability of a transition from state i to j.
c_{jm}	weight of mixture component m in state j.
$\boldsymbol{\mu}_{jm}$	vector of means for the mixture component m of state j.
$\boldsymbol{\Sigma}_{jm}$	covariance matrix for the mixture component m of state j.

A.2 Viterbi algorithm for motion recognition

$P(\boldsymbol{O}|\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$ is calculated using following equation which is called 'Viterbi algorithm'. Let the forward probability $\alpha_j(t)$ for some model $\mathcal{P}_\mathcal{S}$ be defined as

$$\alpha_j(t) = P(\boldsymbol{o}_1,\ldots,\boldsymbol{o}_t, x(t)=j|\mathcal{P}_\mathcal{S}). \tag{18}$$

That is, $\alpha_j(t)$ is the joint probability of observing the first t motion elements and being in state j at time t. This forward probability can be efficiently calculated by the following recursion:

$$\alpha_1(i) = 1 \tag{19}$$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)a_{ij}\right] b_j(\boldsymbol{o}_{t+1}) \tag{20}$$

$$\alpha_j(i) = a_1 b_j(\boldsymbol{o}_1) \tag{21}$$

$$P(\boldsymbol{O}|\boldsymbol{A},\boldsymbol{B}) = \sum_{i=1}^{N} \alpha_i(T) \tag{22}$$

A.3 Learning of discrete HMM parameters

To calculate the HMM parameters $\boldsymbol{A} = \{a_{ij}\}, \boldsymbol{B} = \{b_{ij}\}$ when observation sequence $\boldsymbol{O}$ is given,

$$\gamma_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\alpha_T(i)} \tag{23}$$

$$\gamma_t(i) = \sum_{j=1}^{N} \gamma_t(i,j) \tag{24}$$

are firstly defined. After that, new parameters are estimated using following EM algorithms.

$$\hat{\pi}_i = \gamma_1(i) \tag{25}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1}\gamma_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} \tag{26}$$

$$\hat{b}_{i(k)} = \frac{\sum_{t:o_t=k}\gamma_t(i)}{\sum_{t=1}^{T}\gamma_t(i)} \tag{27}$$

After that, parameter update is executed using following equations. The inference by Eq. (25),(26)and(27) are repeated till the value would be converged.

$$\pi = \hat{\pi} \tag{28}$$

$$a_{ij} = \hat{a}_{ij} \tag{29}$$

$$b_{i(k)} = \hat{b}_{i(k)} \tag{30}$$

Above processes are called as Baum-Welch algorithms.

A.4 Learning of continuous HMMs

In case of continuous HMMs, Baum-Welch algorithms are used as well as discrete HMMs.

$$\hat{\boldsymbol{\mu}}_{jm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T}L_{jm}(t)\boldsymbol{o}_t}{\sum_{r=1}^{R}\sum_{t=1}^{T}L_{jm}(t)} \tag{31}$$

$$\hat{\boldsymbol{\Sigma}}_{jm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T}L_{jm}(t)(\boldsymbol{o}_t-\boldsymbol{\mu}_j)(\boldsymbol{o}_t-\boldsymbol{\mu}_j)'}{\sum_{r=1}^{R}\sum_{t=1}^{T}L_{jm}(t)} \tag{32}$$

$$\hat{\boldsymbol{c}}_{jm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T}L_{jm}(t)}{\sum_{r=1}^{R}\sum_{t=1}^{T}L_{jm}(t)} \tag{33}$$

where

$$L_j(t) = \frac{1}{P(\boldsymbol{O}|\boldsymbol{A},\boldsymbol{B})}\alpha_j(t)\beta_j(t) \tag{34}$$

$$\alpha_j(t) = \left\{\sum_{i=2}^{N-1}\alpha_i(t-1)a_{ij}\right\} b_j(\boldsymbol{o}_t) \tag{35}$$

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_j(t+1) \tag{36}$$

with initial condition:

$$\alpha(1) = 1 \tag{37}$$

$$\beta(1) = \sum_{j=2}^{N-1} a_{1j} b_j(\boldsymbol{o}_1) \beta_j(1). \tag{38}$$