석사 학위논문

Master's Thesis

신경 역학 모델을 이용한 로봇의 합성성과 맥락을 활용할 수 있는 소통 기술 개발

Development of Compositional and Contextual Communicative Skills of Robot by Using a Neuro-Dynamic Model

박 기 범 (朴 杞 范 Park, Gibeom)

전기 및 전자공학과

Department of Electrical Engineering

KAIST

2015

신경 역학 모델을 이용한 로봇의 합성성과 맥락을 활용할 수 있는 소통 기술 개발

Development of Compositional and Contextual Communicative Skills of Robot by Using a Neuro-Dynamic Model

Development of Compositional and Contextual Communicative Skills of Robot by Using a Neuro-Dynamic Model

Advisor : Professor Jun Tani

by

Park, Gibeom

Department of Electrical Engineering

KAIST

A thesis submitted to the faculty of KAIST in partial fulfillment of the requirements for the degree of Master of Science in Engineering in the Department of Electrical Engineering. The study was conducted in accordance with Code of Research Ethics[1].

2015. 1. 7.

Approved by

Professor Jun Tani

[Advisor]

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of KAIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

신경 역학 모델을 이용한 로봇의 합성성과 맥락을 활용할 수 있는 소통 기술 개발

박기범

위 논문은 한국과학기술원 석사(박사)학위논문으로
학위논문심사위원회에서 심사 통과하였음.

2015년 12월 17일

심사위원장 준 타 니 (인)

심사위원 김 대 식 (인)

심사위원 김 준 모 (인)

MEE		박 기 범. Park, Gibeom.
20133259		Development of Compositional and Contextual Communicative Skills of Robot by Using a Neuro-Dynamic Model. 신경 역학 모델을 이용한 로봇의 합성성과 맥락을 활용할 수 있는 소통 기술 개발. Department of Electrical Engineering. 2015. 52p. Advisor Prof. Jun Tani. Text in English

Abstract

The current study presents robotics experiments on acquisition of complex communicative skills with human through learning. A dynamic neural network model which is characterized by its multiple timescale dynamics property was utilized as a neuromorphic model for controlling a humanoid robot. In the experimental task, the humanoid robot was trained to generate specific sequential movement patterns as responding to various sequences of imperative gesture patterns demonstrated by the human subjects which follow particular communicative rules. The experimental results showed that (1) development of the compositional structures enable the MTRNN to recognize and generate sequentially combined movements, including for movements that are combined in an unexperienced way, through generalization in learning of continuous sensory-motor flow, (2) the MTRNN acquired the underlying rules in the imperative communicative gestures by means of extracting type structure among a set of movement patterns demonstrated, (3) the MTRNN developed meta-level cognition capability for turn-taking skills as well as executive control of working memory for *adaptive* remembering and forgetting of task related contexts. The analysis on the dynamic property of the MTRNN trained in the experiments indicated that the aforementioned higher cognitive competency of supporting the compositionality and the meta-level cognition was developed by self-organizing adequate functional hierarchy by utilizing the constraint of the multiple timescale property imposed on the network model. These results of the current research could contribute to developments of social robots with the higher cognitive communicative competency.

Keywords: Neural Network, Meta-cognition, Higher-order cognition, Self-organization, Compositionality, Temporal hierarchy, Neuro-robotics

Contents

List of Tables

List of Figures

Chapter 1. Introduction

Recently, studies on socially intelligent robots [1, 2] have attracted much attention in the research field of intelligent/cognitive robotics. The main motivation of these studies is to investigate theories and methods for building robots that can perform human-like interactions with other agents, including human as well as other robots, autonomously [1]. Studies on socially intelligent robots inherit some of their design philosophy, as discussed in behavior-based robotics, sourced from Rodney Brooks [3] in the late 80s. Although conventional studies on intelligent robots attempted to add behavioral components at a later time, having the basic components on "intelligence" for thinking and cognition built first, the researchers in behavior-based robotics fields did otherwise. By following the thoughts of embodied mind (Varela & Thompson; 1992), they considered that these two processes of thinking and acting should be organized inseparably. Similarly, the researchers in socially intelligent robotics consider the processes of thinking, acting, and communicating as one inseparable process [1, 2, 4].

Recently, not only academia, but also commercial industries, have made great efforts in the development of socially intelligent robots for possible use as home-robots or pet-robots. Such examples can be easily found, including the dog-like robot, AIBO, developed by Sony [5] and a human-interacting humanoid robot, Pepper, by Aldebaran [6]. They proposed new types of home entertainment for family members through interaction with these robots.

Some other researchers, especially in the research field referred to as developmental robotics [7-9], have tried to apply various psychological aspects evidenced in human infant development, in building cognitive models of socially intelligent robots. At the same time, they attempted to contribute to the understanding of the underlying mechanism or principles for the development of social cognitive functions via their reconstruction in robotics experiments, by utilizing psychologically and neurobiologically plausible models. These social cognitive functions include learning to imitate action demonstrated by others [10-14], emergence of turn-taking skills such as switching between following and followed among simulated agents [15] as well as between human and robots [16], or joint attention for achieving coordinated behaviors between human and robots [17, 18]. However, such reconstructions of social cognitive functions have been limited to relatively simple ones as compared to those in even three years old human infants in most cases because such studies in developmental robotics are still in an early stage.

Under the aforementioned research context, the current research aims for reconstruction of higher cognitive mechanisms, which enables robots to perform a complex communicative task with human subjects by employing a particular neural network model. In the adopted task, a humanoid robot is tutored to generate specific sequential behavior patterns as corresponding to imperative gesture patterns demonstrated by the human subject. An importance here is that the imperative gesture demonstrated by the human subject consists of various combinatorial sequences of movement patterns following a predefined compositional rule. For example, the robot is requested to respond by generating particular sequences of movement primitives either in order or in reverse order as well as either slowly, normally or quickly as specified by the imperative gesture. Because the imperative gestures expressed by combinatorial sequences of movement patterns contain linguistic complexity as like sign language, learning of this communicative task is not trivial, even for an adult human.

A technical challenge is that the robot should become able to extract such compositional rules for recognizing demonstrated movement sequences and for generating corresponding own movement sequences through iterative learning of continuous perceptual flows experienced during tutoring sessions while achieving a certain level of generalization. The essential difficulty in the task is that there are no cues that explicitly indicate the structures of the adopted tasks to the robot. First, there are no explicit cues that indicate types of movement patterns performed by the human subject. The demonstrated movement patterns could be either movement primitives to be memorized for regeneration or commands in specifying order of regular or reverse, or with a speed of slow, normal or fast. The underlying type structures should be learned from scratch out of iterative experience of continuous perceptual signals. Also, there are no explicit cues to segment on-going task flow between the human demonstration phase and the robot response phase in the course of continuous alternation of these two phases. The structure of turn-taking between these two phases should be developed spontaneously in the course of training. Furthermore, the robot cognitive process should be able to keep the context of the task flow during each session -- observing the imperative gesture first, and then generating corresponding behaviors, and meanwhile, the accumulated context in the previous session should be erased in the beginning of new session. Such control of the contextual flow, as well as the control of turn-taking, are considered to belong to meta-level cognition, which should be developed gradually via iterative experience on task. The technical challenges focused in the current research can be summarized as: (1) development of compositional structures with generalization to recognize, as well as generate, sequentially combined movement patterns via iterative learning of continuous perceptual flow, (2) extracting type structure among a set of demonstrated movement patterns, (3) development of meta-level cognition for controlling the turn-taking process as well as controlling the contextual flow in the continuously on-going task processes. It is considered that the aforementioned technical challenges of targeting development of higher-order cognitive competency out of lower level perceptual experiences can contribute significantly to the development of truly human-like socially intelligent robots.

For the purpose of accomplishing the aforementioned challenges, the current study takes on an approach based on the paradigm of dynamical systems and self-organization in modeling the development of target cognitive-behavioral processes, because the research outcomes accumulated for two decades have shown that this approach is one of the best to account for the essence of the embodied cognition [19-23].

Especially, the current research follows the results from a series of studies conducted by Tani and colleagues, which conjectured on how the higher-order cognitive mechanisms may be developed by self-organization of particular dynamic neural network models via iterative learning of sensory-motor experiences. Tani showed that compositional navigational knowledge in a maze environment can be learned as embedded in forward model implemented using a recurrent neural network (RNN) model [24]. Sugita and Tani [25] showed that compositional actional concepts can be developed via associative learning of linguistic word sequences and the corresponding sensory-motor sequence patterns, by utilizing the recurrent neural network with parametric biases (RNNPB). Yamashita and Tani [26] showed that functional hierarchy for generating complex behaviors can be developed through iterative learning of sensory-motor experiences by utilizing the reported multiple timescales recurrent neural network (MTRNN) model. In this study, it was shown that a set of behavior primitives can be learned in the fast timescale dynamics of the lower level subnetwork, whereas sequential combinations of these behavior primitives are learned in the slow timescale dynamics of the higher level subnetwork. Maniadakis and Tani [27] showed that meta-level cognition of the contextual flow control and rule switching can be developed by self-organizing adequate attractor dynamics structures via evolutional adaptation of a continuous time recurrent neural network (CTRNN) model in an adopted robot task analogous to the Wisconsin Card Sorting Test. Tani [28] challenges the conventional symbolist framework adopted in artificial intelligence research which has been known to encounter a notorious problem of symbol grounding [29] by claiming that compositional mechanisms self-organized in neurodynamic processes can be naturally situated within the sensory-motor reality because these two can interact densely with each other in the shared metric space of analog systems.

The current research attempts to apply this framework to realization of human-like socially intelligent robots. Detailed analysis on the results of the robotics experiment will clarify how the higher-order cognitive competency necessary for achievement of a human-like communicative response can be developed in the course of self-organizing adequate dynamic structures.

Chapter 2. Background: related models and robotics experiments

This chapter introduces basic mechanisms of different types of recurrent neural network models and their utilization in cognitive neuro-robotics experiments.

2.1 Recurrent neural network (RNN)

RNN models have been investigated because of their advantage in learning of temporal patterns. The most general models are Jordan type [30] and Elman-type [31] which are operated in discrete time. Because the current study will utilize an extended version of Elman-type RNN, its basic architecture and mechanism are explained in the following. Elman-type RNN consists of three layers of the input layer, the hidden layer and the output layer (see Figure 1).

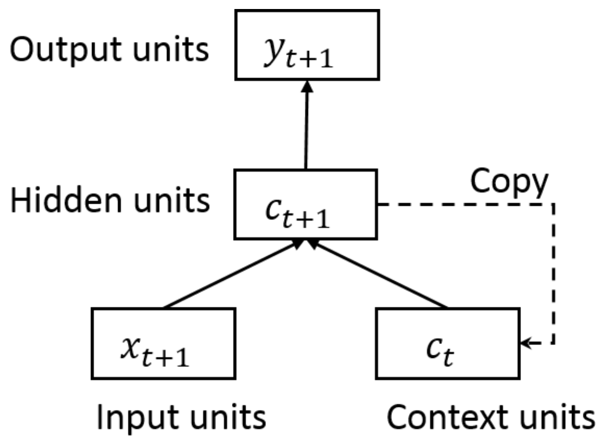

Figure 1. Elman-type RNN architecture

The input layer consists of a number of input units $x_{t,i}$ and so-called the context units $c_{t,l}$ where the suffix t represents the current time step. The hidden layer consists of a number of context units $c_{t+1,j}$ representing the context state at time step $t+1$. The output layer consists of the output units $y_{t+1,k}$ at time step of $t+1$. By receiving sequence of the inputs, the corresponding output sequence is computed by following the forward dynamics shown in the below. First, $c_{t+1,j}$ in the hidden layer is computed by following Eq. 2.1.

$$c_{t+1,j} = f(\sum_i w_{ji} x_{t+1,i} + \sum_l w_{jl}\, c_{t,l} + b_i) \quad (2.1)$$

where w_{ji} and w_{jl} are connectivity weights between jth context unit and ith input unit, and jth context unit and lth context unit. $f()$ is activation function like sigmoid, hyperbolic tangent and etc. Next, $y_{t+1,k}$ is computed by following Eq. 2.2.

$$y_{t+1,k} = f(\sum_j w_{kj} c_{t+1,k}) \quad (2.2)$$

where w_{kj} is connectivity weights between kth output unit and jth context unit.

After the forward computation from time step t to $t+1$, the context activation state $c_{t+1,j}$ is copied to next context inputs $c_{t,j}$ through the context loop. The forward computation is iterated from step *0* to step n. It is noted that context activation state should be developed to represent the internal state of dynamical system through supervised learning of sequences of observable inputs/outputs. This is required for regeneration of particular type of sequence patterns which are characterized by the hidden state problem where the observable outputs or inputs cannot represent the actual state. Although the observable output have their direct targets, the context units have no direct targets. The context activation sequences are developed by utilizing the output error signal indirectly in the course of learning by utilizing the context loop. This will be explained in the following.

A learning scheme referred to as back-propagation through time (BPTT) algorithm [32] is utilized for training of RNN. This scheme is an extension of the conventional error back-propagation algorithm [33] employed for multiple layered perceptron-type network models. The essential idea of BPTT is that the error signal propagates from the current step to the previous step through the context loop. We first consider E as the mean square error between the target and the prediction for all output units across all steps of target sequences. Each connectivity weight w_{ij} is updated in a direction opposite to that of the gradient $\partial E / \partial w_{ij}$ as:

$$w_{ij}(n+1) = w_{ij}(n) - \alpha \frac{\partial E}{\partial w_{ij}} \quad (2.3)$$

where α is the learning rate and *n* is an index representing the iteration step in the learning process. Then, $\frac{\partial E}{\partial w_{ij}}$ is given by:

$$\frac{\partial E}{\partial w_{ij}} = \sum_{t} \frac{\partial E}{\partial u_{i,t}} \, a_{j,t-1} \tag{2.4}$$

and the delta error at the ith unit $\frac{\partial E}{\partial u_{i,t}}$ is recursively calculated from the following formula:

$$\frac{\partial E}{\partial u_{i,t}} = \begin{cases} y_{i,t} - y^{*}_{i,t} + \dfrac{\partial E}{\partial u_{i,t+1}} & i \in Out \\ \displaystyle\sum_{k \in N} \frac{\partial E}{\partial u_{k,t+1}} \left[\delta_{ik} + w_{ki} f'(u_{i,t})\right] & i \notin Out \end{cases} \tag{2.5}$$

where $f'()$ is the derivative of the sigmoid output function and δ_{ik} is Kronecker delta function. The recursive computation of the delta error is iterated through time steps from the outputs in the last step in the sequence to the context units in the input layer in the first step. The context units in the input layer is also updated in the direction of minimizing the error by utilizing the delta error information obtained in these units. Once, optima values for all connectivity weights and the initial context state for each target sequence are obtained, all target sequences can be predicted. In the case that the outputs are just one step forward shift of the inputs with the same dimensionality, the target sequences can be regenerated by copying the output values at the current step to the input ones in the next step without receiving the inputs from the outside. This is called the closed-loop operation.

The RNN models have been utilized for learning of word sequences [31] by extracting the underlying grammar structures as well as sensory-motor sequences [34] by extracting the underlying regularity in the perceptual experiences accumulated during interactions with the outer environment. In the field of neurorobotics study, Tani [24] showed that RNN can be utilized for learning forward prediction model based on robot experiences of sensory-motor sequences during navigation. It was shown that RNN model can learn to predict next step perceptual input of range sensor reading caused by the current step action to be taken via iterative learning of the sensory-motor experiences during exploratory travel of the robot in a maze environment. It was further shown that the robot can generate goal-directed plans by using the acquired forward prediction model.

2.2 RNNPB

The RNN with parametric bias (RNNPB) is an improved version of the conventional RNN. The RNNPB can extract intentions from intentional actions by utilizing the parametric biases (PB) units. The RNNPB can be utilized for both the generation and recognition of dynamic patterns, as Tani and colleagues have shown [17]-[19]. The architecture of the RNNPB is shown in Figure 2.

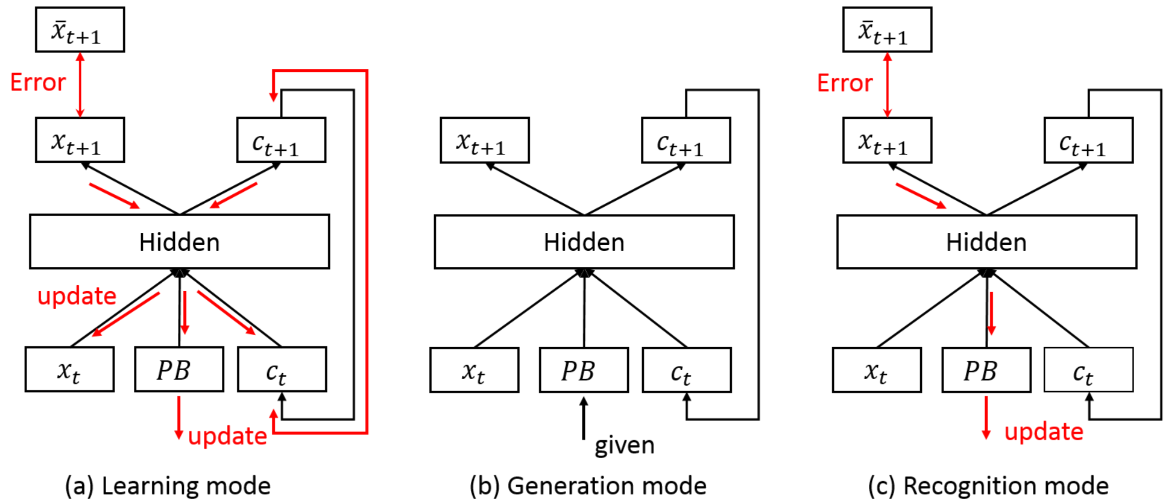

Figure 2. RNNPB model. (a) Learning mode. Connectivity weights and PB units for encoding intentions of each target sequence are optimized through the prediction error BPTT. (b) Generation mode. Perceptual sequences can be regenerated by utilizing the PB units. (c) Recognition mode. Recognition of the perceptual sequences can be done through modulation of PB units in the direction of minimizing prediction error.

The RNNPB is characterized with the PB units allocated in the input layer. The PB functions as a key parameter in shifting dynamic functions developed in the network in terms of parameter bifurcation. In application of the RNNPB to cognitive modeling of action generation, the PB represents the intention of actions. The model network predicts perceptual sequences resulting from the intentional state encoded in the PB units. An actual movement is made by generating the one-step prediction of the next perceptual state through the use of the current one. The perceptual state should include proprioception (joint angles in cases of robots), of which prediction leads to movements by sending it to the motor controller as a target to reach at each time step.

The training process of the RNNPB to learn a group of intentional actions is explained below. The learnable parameters, *i.e.* connectivities and biases, which are common to all actions are optimized by minimizing prediction error through the error BPTT algorithm [32] described in Chapter 2.1. The RNNPB, however, also extracts the intentions of each action in the PB units by backpropagting prediction errors to the PB units [35]. The PB units affect networks output, in a top-down manner, and the PB units are updated to generate correct top-down predictions through iterative learning.

After training, the RNNPB can generate trained perceptual sequences via closed-loop operation by utilizing the PB units corresponding to the perceptual sequences [35]. These results confirm that PB units encode information of perceptual sequences. Thus, the PB units can also be utilized for the recognition of a perceptual sequence by minimizing the errors between the perceived sequence and predicted sequence by adjusting PB units while other parameters are fixed. This is referred to as recognition by error regression.

Ito and Tani [14] utilized the RNNPB model for a robotics task of imitative learning of human behavior patterns. In this study, a humanoid robot with RNNPB learns to imitate a set of human-demonstrated behavior primitive patterns. More specifically, the RNNPB learns to predict the visual perceptual sequences and the corresponding own proprioception sequences (sequences of multiple joint angles) as synchronized for a set of human-demonstrated movement patterns. The corresponding proprioceptive sequences were obtained via direct tutoring by the human trainer, with their physical guidance for movements as synchronized with the demonstration of the human movements. After the prediction learning of a set of behavior primitive patterns converged with minimized error, a test was conducted for determining synchronization of imitatively generated behaviors by the robot with movements of the human demonstrator. The result showed that the robot can imitate the demonstrated patterns by dynamic recognition of the on-going human demonstrated patterns by means of the error regression with the PB values. The shift of the PB values represents the shift of action intended by the human demonstrator, and thus behavior generation by the robot can be shifted accordingly. One technical limitation in this study was that a fixed-length of time window was required for performing the error regression of the perceptual flow in recognition of the human behavior patterns.

Sugita & Tani [25] also utilized the RNNPB model in a robotics task of proto-language and behavior association learning, where the model’s aim was development of compositional actional concepts via interactive learning between the linguistic modality and the behavior modality. They introduced a novel scheme for the associative learning of the linguistic sequences and behaviors. Their model was composed of two RNNPBs. One was the linguistic module for recognizing the linguistic inputs, and another was the behavioral module for generating the corresponding behaviors. Figure 3 illustrates the associative learning of two modules.

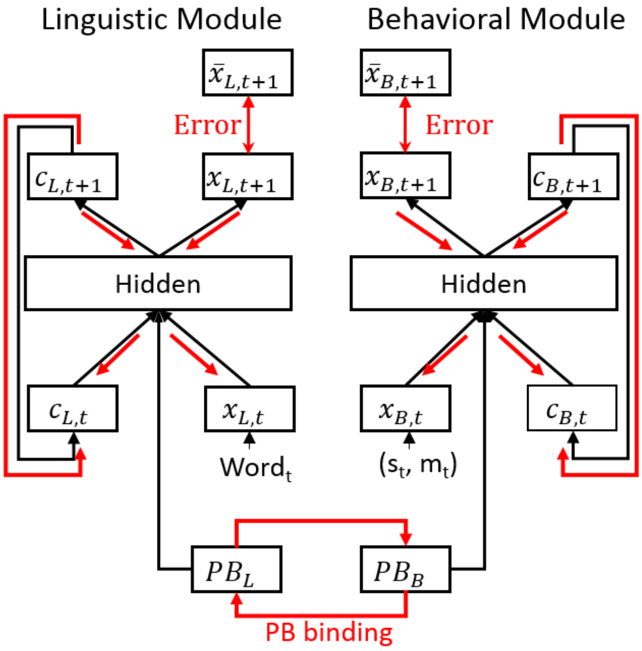

Figure 3. The RNNPB model for the associative learning of the sentences and behaviors. The linguistic module and behavior module learn the sentences and the behaviors respectively by means of prediction error minimization. word_t, s_t, and m_t are word, sensor, proprioception inputs, respectively. The RNNPB learn the associations between the sentence and behaviors via interaction between each of their PB.

They used eighteen different word sequences, corresponding to nine different behaviors, as a linguistic input: a verb came first, followed by a noun. Three verbs and six nouns were used in the 18 sentences. Three verbs specified three different actions: point, push, and hit. Six nouns indicated three different objects: three nouns, left, center, and right, specified a location of the objects, and the other three nouns, red, green, and blue, specified the color of the objects. Both modules were trained to predict the next input through error BPTT: the linguistic module was trained to predict the next word input, and the behavioral module was trained to predict the next sensory-motor input. The linguistic module clearly self-organized the syntactic structure of the word sequences because of the same combinatorial rule between word sequences: a verb comes and a noun comes. However, the linguistic module itself cannot understand the meaning of each word from training of the next word prediction, but the meaning of each word can be understood in terms of the sensory-motor flow from the associative learning of the word sequences and behaviors. The novel associative training method, called PB binding, was used to unify the representations between the two modalities. The PB binding made PB units for sequences with same meanings converged to a similar value. By doing this, the linguistic module enabled the behavior module to organize compositional structure, and the behavior module empowered the linguistic module to categorize the words in terms of the similarities of the sensory-motor flow. The experimental result showed that the underlying compositional structure stemming from the combination of a set of verbs by a set of objective nouns, could be developed through the associative learning, by self-organizing the corresponding two-dimensional grid-like activation states of the PB units among a set of acquired actional concepts. They also demonstrated that the RNNPB model can generate behaviors corresponding to unlearned sentences by utilizing the acquired semantic compositionality.

However, a drawback is that the RNNPB model will only function under the assumption that the task process is explicitly segmented between the language recognition phase and the behavior generation phase. In addition, generalization for the unlearned behavior was not tested.

2.3 MTRNN

Multiple timescale RNN (MTRNN) [26] is an advanced model of the continuous time RNN (CTRNN). MTRNN has hierarchical structure and each layer has different time scales. The CTRNN, the main component of the MTRNN, can be characterized by the following differential equation (Eq. 2.6) for computing contexts units in CTRNN.

$$\tau_i \dot{u}_{t,i} = -u_{t,i} + \sum_j w_{ij} x_{t,j} + \sum_l w_{jl} c_{t-1,l} \quad (i \in I_C, I_o \wedge l \in I_C \wedge j \in I_I) \tag{2.6}$$

where $x_{t,j}$, $u_{t,i}$, $c_{t,l}$ represent the jth input, ith internal state, and lth context activation value at time t, respectively, τ_i is the time constant of the ith input-output and context units, I_C, I_I and I_o are the neuron indices of the context, input, and output layers, and w_{ji} is the ith unit to jth unit connectivity. As described in Eq. 2.6, the time constant of the context unit has a significant effect on how long the context unit retains its previous history and the extent of utilization of accumulated history and current synaptic inputs. The CTRNN has a lot of advantages over the discrete time recurrent neural network (DTRNN) in terms of modeling continuous time series, which usually uses a first-order difference equation (Eq. 2.1), and is more biologically plausible than the DTRNN [26, 36, 37]. The Conventional

CTRNN still has potential for improvement, however, in modeling functional hierarchy, which will be explained below.

Functional hierarchy here means to model primitives and sequences of the primitives separately, which results in multiple subnets, each with their own individual time scales. There are psychology theory and biological evidences [38-40] that emphasize the importance of the functional hierarchy and prove its existence in our brain. According to the schema theory of psychology, the perceptual flow that humans experience in our lives can be decomposed into the multiple primitives, which are reusable components, and human can acquire these primitives from experiences. Furthermore, humans can produce various imagined and actual actions by sequencing primitives in varying orders [41, 42]. Therefore, functional hierarchy can be treated as a key factor to explain how humans acquire, process, and organize the knowledge and how humans generate complex behavior [42, 43].

In spite of the importance of the functional hierarchy, the CTRNN has limited capacity to self-organize functional hierarchy. The innate limitation of the CTRNN is that all of its contexts units have the same time constant. This results in difficulty in self-organization of the functional hierarchy between primitives and their sequences, without training primitives and sequences of primitives separately, using different modules with different goals, because primitives and sequences of primitives have different timescales.

In this context, Yamashita & Tani proposed an MTRNN [26], which is hierarchically organized into two temporally distinct subnets allowing the network to self-organize the functional hierarchy. Figure 4 illustrates the MTRNN model.

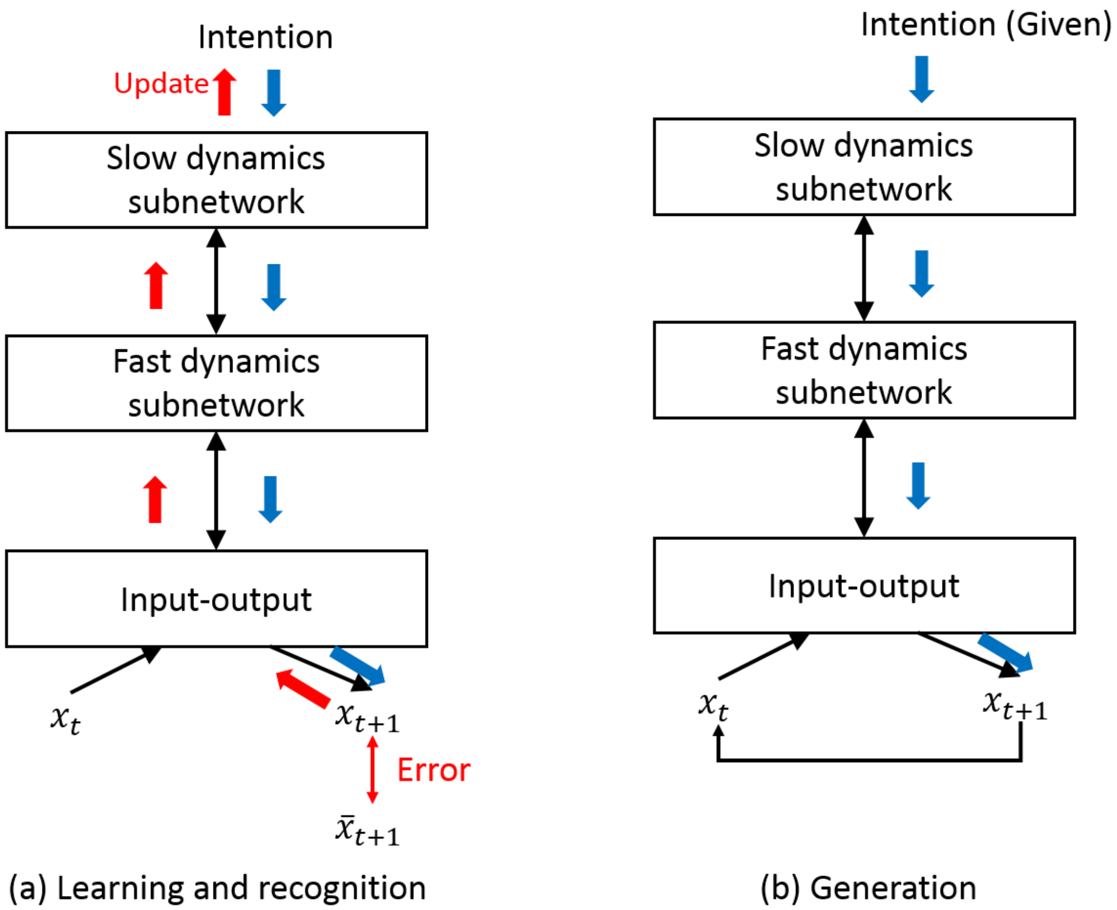

Figure 4. The MTRNN model. The MTRNN has the hierarchical structure of the slow dynamics subnetwork and fast dynamics subnetwork. (a) The MTRNN learns target sequences by optimizing the connectivity weights, which are common to all target sequences, and the intentional states for each target sequences via prediction error BPTT. The intentional states resulting from learning, can be utilized for the recognition of the target sequence with error regression. Figure 4b illustrates the closed loop generation of the perceptual flow by utilizing the intentional states.

The temporal properties of the each subnet are determined by the time constant of the context units in each subnet as shown in Eq. 2.6. One subnet is the fast dynamics subnet that encodes and controls primitives with a short-time window. The fast dynamics subnet is located in the lower layer and directly communicates with the input-output layer. It consists of the fast context units whose state changes quickly subsequent to changes in synaptic input, with easily removable memory. This characteristic of the fast dynamics subnet is appropriate to decompose primitives and adaptively model primitives with a certain level of the generalization. Another subnet is the slow dynamics subnet that encodes and controls sequences of primitives with a long-time window. The slow dynamics subnet is located in a higher layer than the fast dynamics subnet. It consists of the slow context units whose states retain and temporally abstracted representations due to its location, the temporally filtered synaptic inputs and large time constant. Therefore, the slow dynamics subnet is quite important and suitable to generate complex behaviors consisting of multiple primitives, robustly recognize the category of the primitives, and remember sequences of the events. The following equations (Eqs. 2.7~9) describe forward dynamics.

$$u_{t,i} = \left(1 - \frac{1}{\tau_i}\right) u_{t-1,i} + \frac{1}{\tau_i}\left(\sum_j w_{ij} x_{t,j} + \sum_l w_{il} c_{t-1,l} + b_i\right) \quad (i \in I_C, I_o \wedge l \in I_C \wedge j \in I_I) \tag{2.7}$$

$$c_{t,i} = f(u_{t,i}) \tag{2.8}$$

$$y_{t,i} = \frac{\exp(u_{t,i})}{\sum_i \exp(u_{t,i})} \tag{2.9}$$

where $y_{t,i}$ is the ith output activation value at time t, b_i is the bias of the ith unit, and $f()$ is the activation function for context units. The softmax function was used as an activation function of the output units. As described in Eqs. 2.6 and 2.7, the time constant of the context τ determines how long it retains its memory and how much it utilizes synaptic inputs to update memory.

The MTRNN is trained to predict next step input through iterative learning with gradient-descent-based error BPTT in an offline manner as written in the following formula (Eqs. 2.11~14). The following function is the error function E used to train the MTRNN, Kullbak-Leibler divergence.

$$E = \sum_t \sum_{i \in O} y_{t,i} \log\left(\frac{\bar{y}_{t,i}}{y_{t,i}}\right) \tag{2.10}$$

where $\bar{y}_{t,i}$ is next step input $x_{t+1,i}$.

$$\theta(n+1) = \theta(n) - \alpha \frac{\partial E}{\partial \theta} \tag{2.11}$$

$$\frac{\partial E}{\partial w_{ij}} = \begin{cases} \frac{1}{\tau_i} \sum_s \sum_t \frac{\partial E}{\partial u_{t,i}} x_{t,j} & (i \in I_C, I_O \wedge j \in I_I) \\ \frac{1}{\tau_i} \sum_s \sum_t \frac{\partial E}{\partial u_{t,j}} c_{t-1,j} & (i \in I_C, I_O \wedge j \in I_C) \end{cases} \tag{2.12}$$

$$\frac{\partial E}{\partial b_i} = \frac{1}{\tau_i} \sum_s \sum_t \frac{\partial E^s}{\partial u^s_{t,i}} \tag{2.13}$$

$$\frac{\partial E}{\partial u_{t,i}} = \begin{cases} y_{t,i} - y^*_{t,i} + \left(1 - \frac{1}{\tau_i}\right)\frac{\partial E}{\partial u_{t+1,i}} & (\ i \ \in I_O\ ) \\ \sum_j \frac{\partial E}{\partial u_{t+1,j}}\left[\delta_{ij}\left(1 - \frac{1}{\tau_i}\right) + \frac{1}{\tau_j} w_{ji} f'(u_{t,i})\right] & (\ i \ \in I_C\ ) \end{cases} \qquad (2.14)$$

where θ indicates learnable network parameters consist of connectivity, bias and initial context state, and s is the index of training sequences, α is learning rate, and $f'()$ is the derivative of the activation function. Eq. 2.14 has important meaning. Slow dynamics subnet, due to the large time constant in its context units, more focuses on model long-term dependencies than short-term dependencies with a long-time window. It encodes and modulate sequences of the primitives. On the contrary, the fast dynamics subnet more focuses on modeling short-term dependencies. It segments and encodes primitives. This means, the network naturally self-organizes functional hierarchy from the learning of the perceptual flow, thanks to its hierarchical structure and different time scales in different subnets. A large time constant in the slow dynamics subnet has another significant meaning in overcoming the long-standing problem in modeling long-term dependencies with gradient based error BPTT, called the vanishing gradient problem stated in [44, 45].

As shown in Figure 4, initial context unit states of the slow contexts $u_{0,i}$ can be utilized as the intentional states of certain primitives or a series of primitives in recognizing and generating actions like the PB units in Chapter 2.2. Recognition through error regression of the Initial context unit states has the same difficulty as PB: ambiguity in determining time window size and learning rate for error regression. Updating initial states can be done by using Eqs. 2.11 and 2.14.

Yamashita & Tani [26] demonstrated that the MTRNN self-organized functional hierarchy through repetitive learning of the continuous sensory-motor sequences in their robotic experiment. They trained a network to generate 5 different actions. Four among 5 actions were object-related actions, which require both recognition of the object and generation of the action. The MTRNN successfully generated different actions specified by initial states of the context unit.

After training 5 different actions, they trained novel combinations of the actions by training only connectivities between the slow context units to verify their hypothesis about functional hierarchy in the MTRNN, which states that the slow dynamics subnet encodes sequences of the primitives modeled in the fast dynamics subnet. The network generated novel sequences of the primitives with the aid of its configuration while emphasizing significance of the using different time scales for different subnets; the network can self-organize functional hierarchy only when the time constants ratio between fast context units and slow contexts units is sufficiently large.

The MTRNN also showed a certain level of generalization related to another important issue. Despite of the object location changes, slow context activities were almost the same for generating the same actions, while fast context activities were adjusted to the object location changes without categorizing same actions as different primitives. This means, the network has the ability to resolve the dilemma of whether generalize or separate the similar patterns.

Through the series of their experiments they verified the ability of the MTRNN to perform as a computational cognitive model, however, the MTRNN couldn't generate a novel combination of primitives without additional training, and all the combinations of the primitives have to be trained. It will be meaningful to improve both the model itself and the training method to overcome this limitation.

Chapter 3. General description of the current task.

Tasks in this paper were designed to investigate how robots implemented by the MTRNN can generate adequate behavioral responses by recognizing communicative gestures of having certain complexity demonstrated by human subjects. The human-robot communicative tasks designed in the current study are designed to investigate the following technical problems: (1) acquisition of the compositional structures enable the MTRNN to recognize and generate sequentially combined movements, including movements that are combined in an unexperienced way, through repetitive training of continuous perceptual flow, (2) categorization of demonstrated movement patterns according to not only shape of movement patterns but also the syntactic and semantic rules hidden in relationships between movement patterns, and movement patterns and behavioral responses, and (3) acquiring the meta-cognitive turn-taking skills and working memory manipulation skills for *adaptive* remembering and forgetting. We designed the following communicative tasks while considering these aforementioned problems.

Communicative tasks in this research consist of sequentially-dependent imperative gestures demonstrated by a human and corresponding responses generated by a robot. An imperative gesture is a sequential combination of human movement patterns: 3 different movement primitives, 2 different order commands and 3 different speed commands. One to three human movement primitives came first, followed by an order command and then a speed command. Thus, the number of possible combinations of imperative gestures is 234. Human movement primitives determine the corresponding robot motor primitives that have to be generated in the response phase. Order commands are verbal-like commands indicating either forward or reverse order in generation of motor primitive sequences. Speed commands are adverbial-like commands indicating speed of robot behaviors, *i.e.* either fast, normal or slow. A corresponding robot response is sequential combination of motor primitives resulting from movement primitives, and its order and speed were determined by order command and speed command. After demonstrating certain human movement patterns, a human subject goes back to the home position and stays for 5-10 time steps. After demonstrating certain robot movement patterns, a robot returns to home position and stops for 12 time steps. Figure 5 shows the examples of continuous perceptual flow used in the communicative tasks.

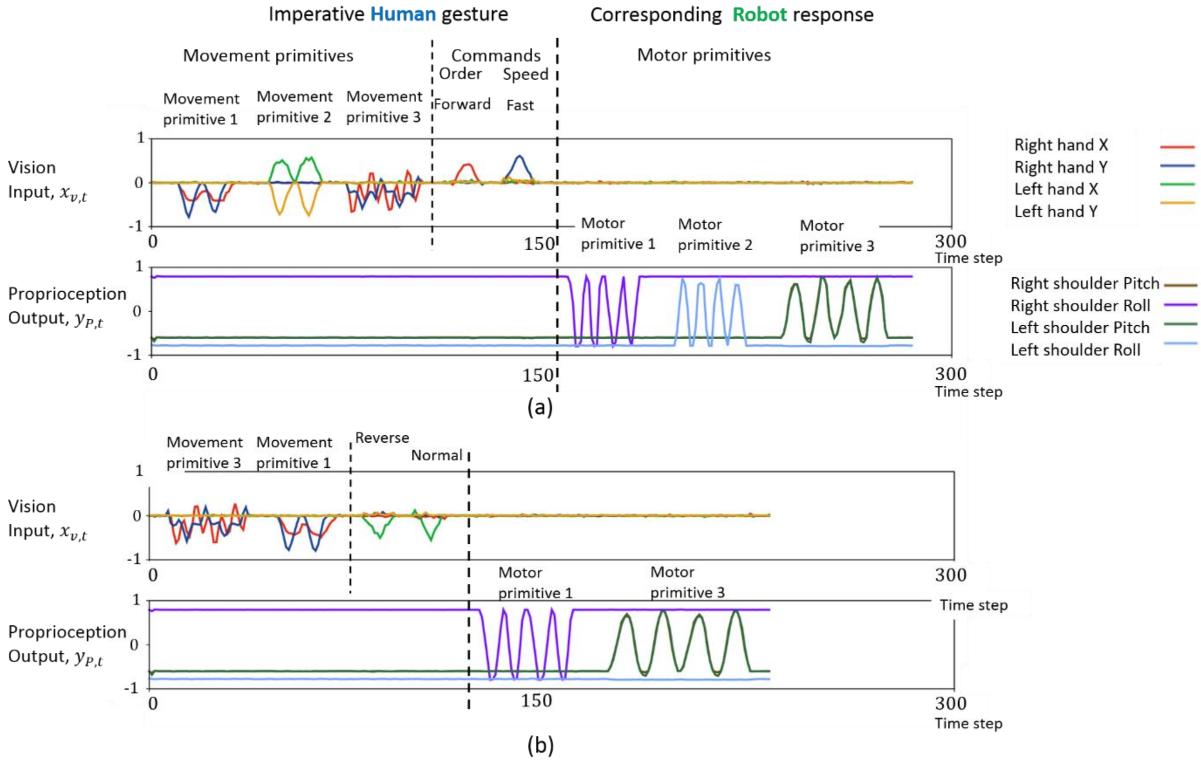

Figure 5. Examples of the communicative task. A human demonstrates an imperative gesture pattern, and after some delay, a robot generates the corresponding response. An imperative gestures consists of the movement primitives, order and speed commands. First, sequentially combined movement primitives comes. The movement primitives determine the motor primitives that generated in the corresponding response generation phase. The number of the movement primitives in each imperative gesture can be varied: one to three movement primitive can be demonstrated in each task session. Next, an order command, which determines the order of the sequences of the motor primitives, comes. Finally, a speed command that specifying the robots moving speed comes.

These communicative tasks are complex in the following ways. First, both imperative human gestures and the corresponding robot responses are a combination of movement patterns. Second, the imperative gestures have a complex compositional rule; a human subject can demonstrate one to three movement primitives before demonstrating commands. Third, the same movement patterns have different meanings depending on their previous pattern's type. For example, movement patterns for order commands followed by another order command should be interpreted as normal speed command, if previous movement patterns was order commands. Note that there is no explicit cue, *e.g.* symbol or label, in sensory-motor flows that network can utilize to segment the patterns and decode the meaning of each of the segmented human movement patterns and extract the task rules. The proposed network has to develop higher-order cognition to adequately perform the tasks, utilizing only the iterative learning of the lower-level sensory-motor flows.

Following are the details of the human movement patterns and robot motor primitives. The human movement patterns are left and right arm movements. Trajectory data of the arm movement patterns were recorded by tracking xy position of the palm positions through the KINECT skeletal tracking system [46] while a human subject performed the movement patterns. For all dimensions, the position value for initial position was 0, and the range of the position values was [-1:1] for experiment 1 and [-0.8:-0.8] for experiment 2~4. We employed the humanoid robot NAO as an intelligent robot agent for the communicative task. To get trajectory data of the motor primitives of the robot action, the human experimenter guided both arms of the NAO by holding and moving NAO's arms. While guiding, we encoded 4 joint angles comprising shoulder rolls and pitch angles of both arms. Ranges of the angles were [-0.8:0.8]. The sampling rate of both human movements and robot motor primitives were 10Hz. Human movement patterns and robot motor primitives adopted in this study are described in the figure 6.

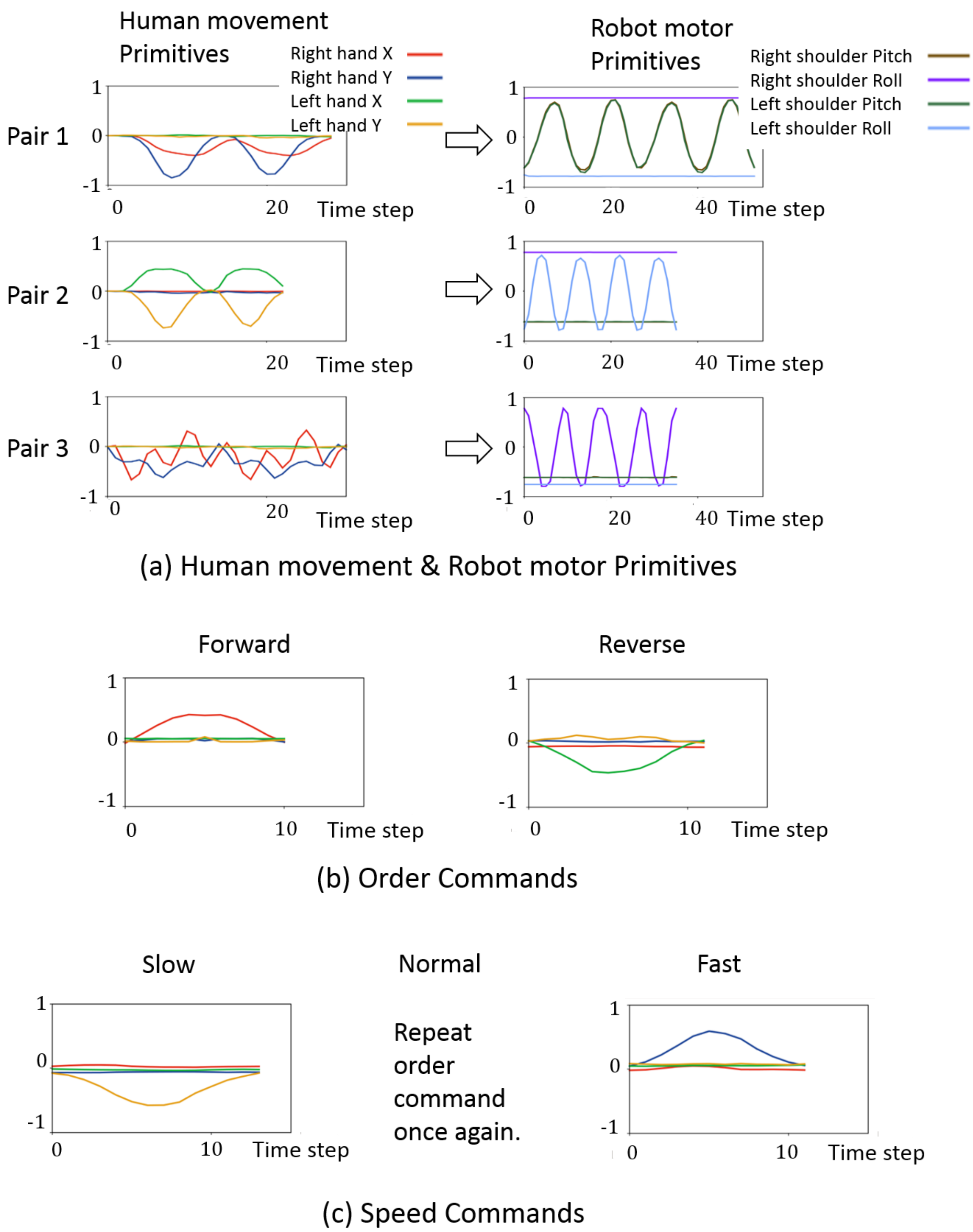

Figure 6. Human movement patterns and robot motor primitives of the communicative tasks. Three different human movement primitive and corresponding robot motor primitive pairs, two different order commands and two different speed commands compose perceptual flow of the communicative tasks.

To help the learning of the network by increasing the distance between both movement patterns and motor primitives, we applied a softmax transformation to the recorded data of movement patterns. These transformed data were used as the input and target-output of the network. Softmax transformation was independently applied to each dimension of the movement patterns and motor primitives. The following equation (Eq. 3.1) describes the softmax transformation [47]:

$$p_{ij,t} = \frac{exp\frac{-\|k_{ij} - k_{i,t}^{sample}\|^2}{\sigma}}{\sum_{j\in Z} exp\frac{-\|k_{ij} - k_{i,t}^{sample}\|^2}{\sigma}} \tag{3.1}$$

where $k_{i,t}^{sample}$ indicates position value of the ith dimension at time t, k_{ij} is the value of the jth element of the ith dimension's reference vector, σ is a constant that determines sharpness of the distribution, and p are transformed vectors. Values of the elements of the reference vectors are calculated by the following Eq. 3.2:

$$k_{ij} = -B_i + 2\frac{B_i}{l(i) - 1}(j - 1) \tag{3.2}$$

where $l(i)$ is the length of the ith dimension's reference vector and B_i is the ith dimension's upper bound. We used 0.01 as a σ, and 9 for $l(i)$.

We tested whether and how the MTRNN can develop higher-order cognition that satisfies the three requirements, addressed in technical problems of the current task, through a series of experiments. Each experiment focuses on the separate problems. To focus on the different problems individually, the form of the tasks varies with experiment. Experiment 1 focuses on how to efficiently train the network to become robustly recognize movement patterns regardless of its variant features, speed and amplitude. Experiment 2 and 3 focuses on the condition (1) and (2). Experiment 4 focuses on the condition (3) with the data containing concatenated sessions in one data. Details of the experimental setting are will be described in Chapter 5 Experiment.

4. MTRNN used in the current experiment

We utilize an MTRNN for performing the current task. For achieving the task of intelligent communicative response by robot, the adopted MTRNN model should satisfy the following technical conditions as mentioned in the previous chapter. First, the network should be equipped with a compositional structure that can be utilized to recognize and generate not only experienced movement combinations but also unexperienced movement combinations that are connected in unexperienced way, through iterative learning of the continuous perceptual flow. Secondly, the network should understand task rules hidden in a set of movement patterns. The final condition for the MTRNN model is acquisition of metacognitive knowledge, *i.e.* knowing when to remember and when to forget, to successfully manage on-going successive independent tasks. Among these technical issues, controlling of the context flow in the unarticulated task processes in (3) cannot be treated well by utilizing the conventional error regression scheme used in RNNPBs and MTRNNs. In the prior study on imitation learning [14] or in language-behavior associative learning [25], an explicitly segmented behavior pattern or word sentence was presented to the network model as a chunk. In such case, the error regression scheme for recognizing these input sequences can work by applying the BPTT scheme only to the windows of explicitly segmented sections. However the scheme of the error regression for recognition cannot be applied to the current task because a part of human demonstration is not explicitly segmented in the task process. Thus, the network model has to recognize the intention of the human subjects contextually in the on-going flow of the perceptual signal.

On considering these requirements, we propose to utilize the MTRNN model in a novel manner as described in the following. First, the MTRNN model is built as consisting of three subnetworks with different timescales. It is expected that the allocation of the slow dynamics subnetwork assigned with a large time constant can deal with the technical conditions of (1) and (2) by assuming that higher cognitive mechanism related to development of compositionality and type structures could be well achieved by integrating experiences of longer periods into abstract structures by utilizing the inherent slow dynamics of the network. Next, a novel way of using the MTRNN is proposed. In the current study, the network is exposed to continuous flow of visual perception from Kinect while it is expected to generate adequate behavior response in a timely manner without allocating specific neural units such as PB or the initial context units to the encoding of intention states. Instead, our expectation is that the intention of the human subjects can be recognized in the activation patterns of the context units as integrated though time with various timescales while receiving continuous input streams. The activation dynamics of the context units responding to the contextual recognition of the human subject would generate adequate behaviors at right time. A hypothesis that must be examined in the current experiment is that the task processes can be naturally segmented when a mechanism for adequately controlling the contextual flow in the model network is developed.

4.1 Model overview.

We extended Yamashita's MTRNN model, which was explained in Chapter 2.3. The following figure, Figure 4.1, shows a general form of the MTRNN.

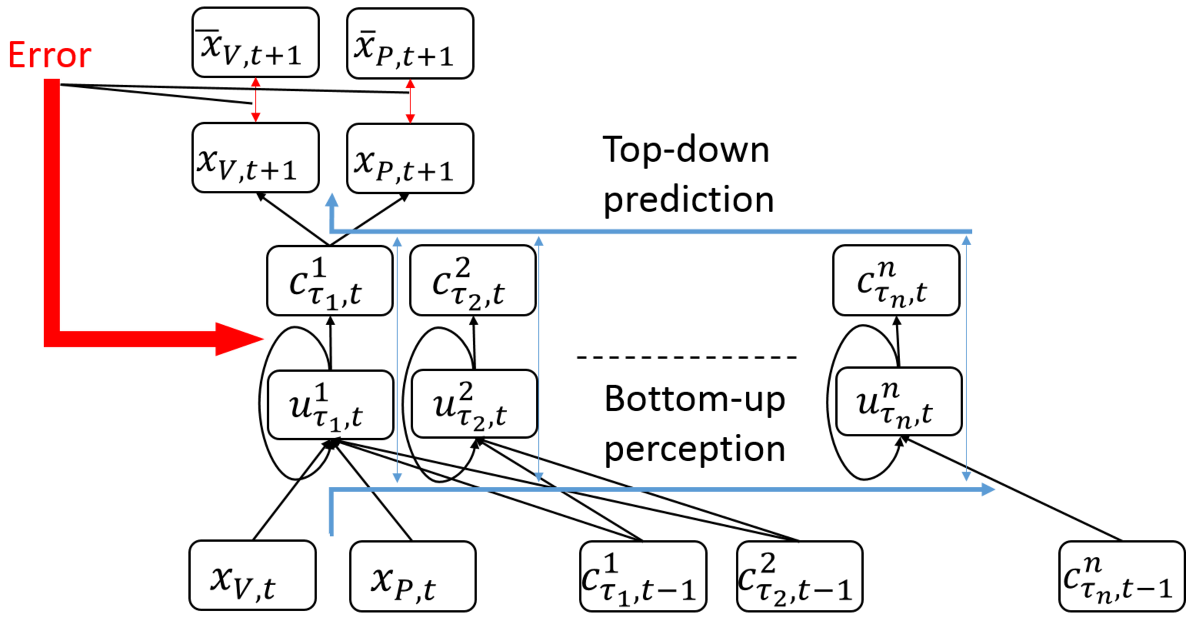

Figure 7. The system flow of the MTRNN in the learning phase. The MTRNN optimizes its learnable parameters through prediction error BPTT. $x_{V,t}$ and $x_{P,t}$ are the vision input and proprioception input, respectively. $x_{V,t+1}$ and $x_{P,t+1}$, are vision and proprioception prediction output, and $\bar{x}_{V,t+1}$ and $\bar{x}_{P,t+1}$ are vision and proprioception targets, respectively. n is the index of the layer, τ_n is the time constant of the context unit in the nth hidden layer, and c_t and u_t are the activation value and internal state of the context unit at time step t, respectively.

In this study, the time constant of the context units in the higher layers are larger than the time constants of context units in the lower layers, and the context units in the same layer use the same time constant, with exception of the experiment designed to verify the effect of multiple time constants. Both input and output nodes are connected to the lowest layer, which has the smallest timescale, because output of the tasks in this research is prediction of the perception, not categorical output for classifying tasks. The number of hidden layers can vary depending on the task environment. The functional hierarchy for modeling, such as, characters, words, and sentences, can be self-organized by utilizing 3 different subnetworks with different timescales. It is Important to note that introducing an intermediate subnetwork between the lower and higher subnetworks to increase the distance between them can significantly improve the performance of the MTRNN [48].

The goal of the communicative tasks in this research, which is recognition of the sequences of the primitives and generation of the sequences of the primitives, has been described in Chapter 3. By considering the goal and the importance of the intermediate subnetwork, we employed the 3 different subnetworks with different timescales for the adopted MTRNN: a fast dynamics subnetwork to model primitives, a slow dynamics subnetwork to model sequences of the primitives, and an intermediate dynamics subnetworks for separation of the fast and slow dynamics subnetworks. The hidden layer between the input and output layers was also introduced to increase context dependency of the outputs, and increase non-linearity between inputs and outputs.

4.2 Forward dynamics

Internal states, and activation values of the context units and output units are determined using by Eqs 2.7~9 described in Chapter 2.3.

$$u_{t,i} = \left(1 - \frac{1}{\tau_i}\right) u_{t-1,i} + \frac{1}{\tau_i}\left(\sum_j w_{ij} x_{t,j} \ + \sum_l w_{il} c_{t-1} \ + b_i\right) \ \ (\, i \in \ I_C, I_o \ \wedge \ l \ \in \ I_C \ \ \wedge \ j \ \in I_I) \quad (2.7)$$

$$c_{t,i} = f\left(u_{t,i}\right) \quad (2.8)$$

$$y_{t,i} = \frac{\exp(u_{t,i})}{\sum_i \exp(u_{t,i})} \quad (2.9)$$

The time constants of the context units have a significant effect on the temporal properties of the dynamics of the context units and the subnetworks consisting of the respective contexts units. The important difference between Yamashita's MTRNN and the adopted MTRNN in forward dynamics is that the adopted MTRNN does not utilize PB values or initial context states encoding the corresponding time series in the generation of prediction output. To generate correct prediction output, the adopted MTRNN utilizes patterns of neural activation driven by sensory inputs. This means that the adopted MTRNN dynamically adapts its higher level representations of a perceptual sequence with weak prediction rather than strong prediction of the future input.

4.3 Training

The training method for the adopted network is similar to the MTRNN introduced in Chapter 2.3. The objective function is the Kullbak-Leibler divergence between the target and next step input and output, which works to predict the next step input, as described in the following Eq. 2.10.

$$E = \sum_t \sum_{i \in O} y_{t,i} \log(\frac{\bar{y}_{t,i}}{y_{t,i}}) \tag{2.10}$$

The network's learnable parameters are optimized to minimize the above objective function. The learnable parameters are updated via the following equations, Eqs. 2.11~14, through iterative learning of the continuous perceptual flows.

$$\theta(n+1) = \theta(n) - \alpha \frac{\partial E}{\partial \theta} \tag{2.11}$$

$$\frac{\partial E}{\partial w_{ij}} = \begin{cases} \frac{1}{\tau_i} \sum_s \sum_t \frac{\partial E}{\partial u_{t,i}} x_{t,j} & (\, i \in I_C, I_O \ \wedge \ j \in I_I \,) \\ \frac{1}{\tau_i} \sum_s \sum_t \frac{\partial E}{\partial u_{t,j}} c_{t-1,j} & (\, i \in I_C, I_O \ \wedge \ j \in I_C \,) \end{cases} \tag{2.12}$$

$$\frac{\partial E}{\partial b_i} = \frac{1}{\tau_i} \sum_s \sum_t \frac{\partial E}{\partial u_{t,i}} \tag{2.13}$$

$$\frac{\partial E}{\partial u_{t,i}} = \begin{cases} y_{t,i} - \bar{y}_{t,i} + \left(1 - \frac{1}{\tau_i}\right) \frac{\partial E}{\partial u_{t+1,i}} & (\, i \in I_O \,) \\ \sum_j \frac{\partial E}{\partial u_{t+1,j}} \left[\delta_{ij} \left(1 - \frac{1}{\tau_i}\right) + \frac{1}{\tau_j} w_{ji} f'(u_{t,i}) \right] & (\, i \in I_C \,) \end{cases} \tag{2.14}$$

Eq. 2.12 describes how to calculate delta weights, where the condition $(\, i \in I_C, I_O \ \wedge \ j \in I_I \,)$ had to be revised to $(\, i \in I_C \ \wedge \ j \in I_I \,)$, because there were no connections between the input and output. Note that the adopted MTRNN did not utilize PB units nor initial context states. There were no parameters allocated to encode specific sequences. All parameters were common to all sequences.

As a result, the adopted MTRNN was trained to adaptively change its internal representation of human gestures and prediction of the subsequent patterns in coincidence with the dynamically changing input patterns occurring during perception of imperative human gestures, as opposed to maintaining its prediction. It was impossible to predict specific human movement patterns in the starting phase and part of the imperative human gesture following perception onset. On the other hand, in the generating robot action phase, maintaining prediction was desirable because robot action sequences were deterministic in given imperative human gestures. The network was developed to generate robot action sequences by utilizing its prediction resulting from the accumulated context values that were determined by the recognition of imperative gestures. This means that representations resulting from recognizing a human gesture, and the representation that initiated an action sequence, which corresponded to a recognized human gesture, were spontaneously coupled through the learning without employing special methods, such as PB binding or cell assembly [25, 49]. Furthermore, the network can segment sensory-motor sequences into primitives [26]. The network could also understand the meaning of each movement pattern from determining the relationship between them.

Initial weights and biases were randomly chosen from a Gaussian distribution. The range of the Gaussian distributions were [-0.1, 0.1] for the weights and [-1, 1] for the biases. To accelerate the learning speed, achieve better generalization capability, and mitigate the learning rate sensitivity of the results, we used the following adaptive learning rate method. The initial learning rate was set as $^{0.1}/_{Ttotal\ *\ d}$, where $Ttotal$ means summation of the time step over all training data, and d is dimensionality of the output. In each epoch, the learning rate was updated by the following algorithm [50].

(1) Calculate the delta errors using part of the training data.

(2) Calculate the rate (r) of the KL-divergence before updating parameters, and the KL-divergence after updating the parameters using whole training data.

(3) If $r > r_{th}$, then update α to $\alpha\alpha_{dec}$ and go back to (2).

(4) If $r < 1$, then update α to $\alpha\alpha_{inc}$ and go to the next epoch.

In this study, we used 1.1 for the r_{th}, 0.7 for the α_{dec} and 1.2 for the α_{inc}.

By calculating the gradient using only part of the data and determining the learning rate based on the r of the whole training data, we could force the network to consider the generalization capability, and not just attempt to minimize prediction error of the selected data when updating its parameters. Adjusting the learning rate by only utilizing the training error, however, has a limitation in improving the generalization capability of the network because the network only encounters perceptual flow in the training data. For the network, the perceptual flow used in training data essentially becomes the “world” in which the network is aware. Through iterative learning with error backpropagation through time (BPTT), the network acquires skills and extracts rules hidden in perceptual flows to minimize prediction error that occurs in its “world”. At this point in time, it does not seem likely that acquired skills and extracted rules will be able to be applied to situations outside of its “world” as well.

In this context, the training data were divided into two groups. One group consisted of the data that was used for both calculating the delta error and adjusting the learning rate. Another group consisted of the data that was concerned with only adjusting the learning rate and was used as validation data. By also considering validation data in updating learning rate, the network would be able to indirectly sense the existence of the outside of its “world.” That is because it would encourage the network to acquire skills and extract rules that can be generally applied outside of its “world” rather than acquiring specific skills and extracting particular rules only be applicable to the its “world”. The concept of considering the validation error in adjusting the learning rate is similar to Duffner & Garcia’s [51] method. However, the validation data in the current study were unexperienced human gesture and robot responses, whereas the validation data in their work were only different from training patterns in noise. Consequently, considering validation errors when adjusting the learning rate had more important meaning in this study, because this can widen the network’s awareness of the world.

4.4 Recognition and generation

Due to the error regression scheme’s limitations regarding its application to the current tasks, a novel way of recognition and generation in place of adopting the error regression scheme, was applied. In each task session, the network perceived the imperative human gesture first. Then, the network generated the corresponding action sequences by utilizing its context states. These context states were consequences of the extraction of human intention from the imperative human gesture. This was all achieved without utilizing the PB units or initial context states. It was expected that the network would utilize a slow dynamics subnetwork to extract intentions from the input streams while perceiving the imperative human gestures. It was further expected that the representations of slow dynamics would be robust even when dealing with variant features such as amplitude and speed, as well as noise, which are a natural part of human movement that needed to be addressed. The input stream would be temporally filtered while passing through each layer. Figure 8 shows the system architecture applied in the communicative tasks requiring both recognition and generation.

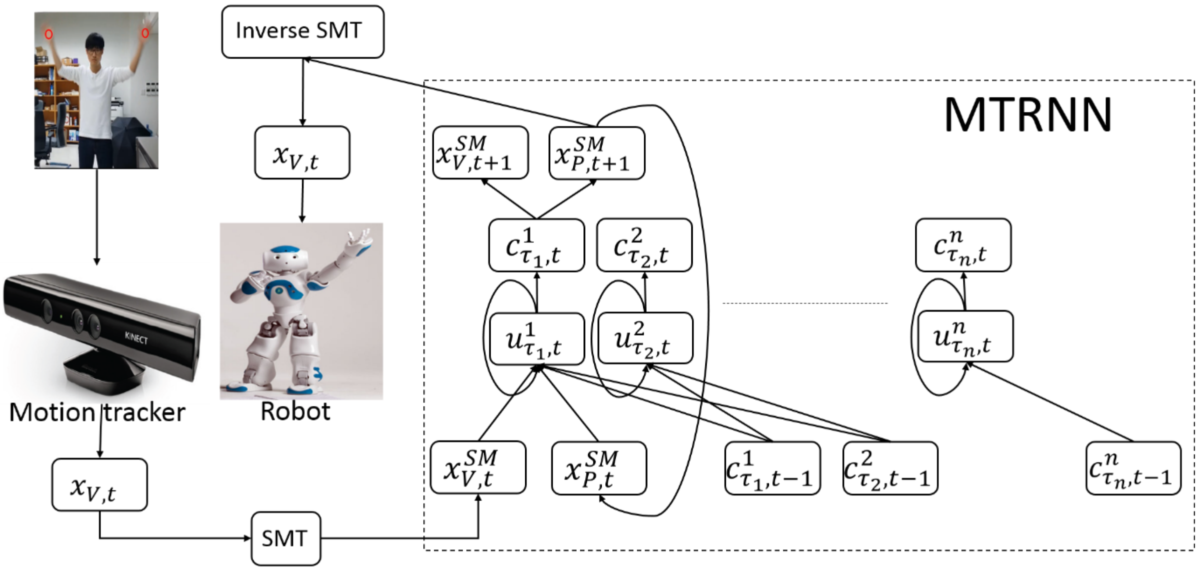

Figure 8. The system architecture for the performing of communicative tasks. The motion tracker tracks palm positions of the human ($x_{V,t}$). The tracked position values are transformed using softmax transformation (SMT) and then used as vision inputs ($x^{SM}_{V,t}$) of the MTRNN. The proprioception input ($x^{SM}_{P,t}$) are the network's previous proprioception output ($x^{SM}_{P,(t+1)-1}$). The proprioception outputs ($x^{SM}_{P,t+1}$) are generated, based on the abstracted intentions of an imperative gesture that the network perceived, and the network's previous output. These output are inversely transformed and transmitted to the robots to control joint angles.

The applied method had several advantages over error regression. First, the applied method does not require a properly set time window for calculating the prediction errors or the learning rate of the intentional states, whereas error regression cannot successfully work without properly setting a time window and learning rate. Through the iterative learning by means of minimizing prediction errors, the network became able to extract task rules hidden in continuous perceptual flows (e.g., what type of movement would come next, when it needed to generate the corresponding actions). By utilizing these extracted task rules, network was able to autonomously set its time window. Furthermore, the network could also acquire a working memory-related meta-cognitive skill that decided what and when it needed to forget and remember. The details regarding the acquisition of a meta-cognitive skill will be discussed in Chapter 5.4. Secondly, the proposed method is freer from the local minima as compare with the error regression scheme. The network was developed to utilize the neural representations of the imperative human gestures to generate corresponding actions. These neural representations were the outcome of continuous abstraction and integration of the continuous perceptual flows during imperative human gestures perception. This means that the resultant neural representations following human gesture perception and the neural representations that triggered the generation robot actions were coupled through the learning. Thus, recognition of the human gestures naturally led to the generation of robot actions. In case of error regression, the adaption of the intentional states by utilizing prediction error could lead to the utilization of the local minima. These local minima may only be able to minimize prediction errors of the gesture patterns, and not generate corresponding action sequences, because, in recognition and generation phase network could not utilize errors came from proprioception.

However, although it has long been known that integration of the bottom-up and top-down recognition is important, the recognition strategy of the applied method exhibited limitations in reflecting top-down processing. In addition, the context states could be captured in a strong attractor when an exception occurs, such as the introduction of a human-demonstrated pattern that was not shown in training or a gesture that violates the task rules. This could disturb the performance of the network. In the applied method, it was difficult to release the captured context states from an attractor, whereas error regression can do this easily. Despite these disadvantages, the proposed method showed satisfactory results for the communicative tasks used in this study.

Chapter 5. Experiments and results.

The aim of this study was to develop a socially intelligent robot with higher level cognitive competency and to understand the underlying mechanisms. In this chapter, technical challenges will be dealt with through a series of experiments using communicative tasks. As summarized in the introduction, the technical challenges were (1) development of compositional structures with generalization to recognize, as well as generate, sequentially combined movement patterns via iterative learning of continuous perceptual flow, (2) extracting type structure among a set of demonstrated movement patterns, (3) development of meta-level cognition for controlling the turn-taking process as well as controlling the contextual flow in the continuously on-going task processes. In Experiment 1, a novel training method that can efficiently improve lower feature level generalization ability of the network will be examined. In Experiment 2-1, technical challenges (1) and (2) will be addressed. Importance of the temporal hierarchy in acquisition will be examined in Experiment 2-2. Technical challenge (3) will be addressed in Experiment 3.

5.1 Experiment 1: Lower level generalization

Human movement patterns have many variant features such as amplitude or speed. Even when the human demonstrates the same movement patterns, the speed and amplitude of the movement patterns can differ between every trial. To understand that every movement pattern can be demonstrated with different variant features, if it is necessary to show every movement pattern with every possible combination of variant features, *e.g.* large and fast, large and slow, *etc*., the number of training data become too large. On the other hand, if the network becomes able to recognize the intention of many different movement patterns and robust to variant features by understanding the existence of the variant features that needed to be ignored, via observing a small number of movement patterns demonstrated with different variant features, the amount of training data can be dramatically reduced. The network can be trained much more efficiently. Sugita [25] demonstrated that different words that indicated the same object could be represented with quite similar PB values via PB binding, and Yamashita & Tani [26] demonstrated that the activities of the slow contexts were almost independent to the position difference of the object while the robot was performing the same movement with the objects located in different positions. As explained in Chapter 4.2, the neural representations of an imperative human gesture and its corresponding robot response would be coupled after training, and would show a result similar to that of PB binding. In addition, the baseline model of the network was an MTRNN. Thus, it was expected that the network would be able to develop robust representations for the variant features by training the network to generate the same robot response for the same human movement primitives demonstrated with some possible combinations of AMPs and PERs. In this experiment, the network was tested on whether it could be efficiently trained to make robust representations for the variant features using a small amount of training data as described earlier.

To focus on variant features generalization, this experiment was conducted using a simplified task setting. An imperative gesture in this experiment consists of one movement primitive among 3 different primitives, followed by the forward order and normal speed commands. The movement primitives in this experiment were comprised of 7 different AMPs and 7 different Pers. The generalization performance were tested with 4 different data sets. Table 1 describes the training data set and test data set.

Table 1. The configuration of each data set.

AMP \ Per	50	70	80	100	110	130	150
70							
80							
90							
100				P1 P2 P3			
110							
130							
150							

(a) Training data set 1

AMP \ Per	50	70	80	100	110	130	150
70							
80				P2			
90							
100		P1		P1 P2 P3		P1	
110							
130				P2			
150							

(a) Training data set 2

AMP \ Per	50	70	80	100	110	130	150
70							
80		P1		P1 P2		P1	
90							
100		P1 P3		P1 P2 P3		P1 P3	
110							
130		P1		P1 P2		P1	
150							

(a) Training data set 3

AMP \ Per	50	70	80	100	110	130	150
70							
80		P1 P2 P3		P1 P2 P3		P1 P2 P3	
90							
100		P1 P2 P3		P1 P2 P3		P1 P2 P3	
110							
130		P1 P2 P3		P1 P2 P3		P1 P2 P3	
150							

(a) Training data set 4

Each human movement was constituted with seven different AMPs: 70, 80, 90, 100, 110, 130, and 150%, and seven different Pers: 50, 70, 80, 100, 110, 130 and 150%. As a result, each human movement primitive can be expressed in 49 different ways using combinations of AMPs and Pers. Human movement primitives 1, 2, and 3 were denoted by P1, P2, and P3 in the tables, respectively. P1, P2 and P3 in Table 1(b), for example, show that training data set 2 contained movement primitive 2 which yields 80% AMP and 100% PER, 100% AMP and 100% PER, and 130% AMP and 100% PER. As shown in table 1(a), all three primitives, which all represent the combination of only a single AMP and Per. As demonstrated in table 1(b), primitives 1,2, and 3, which each represent the combinations of a single AMP with three different Pers, the combinations of three different AMPs and a single Per, the combination of only a single AMP and Per, respectively. Table 1(c) shows primitives 1, 2, and 3, which each represent the combinations of a single AMP with three different Pers, the combinations of three different AMPs with three different Pers, the combinations of a single AMP with three different

Pers, the combinations of three different AMPs and a single Per, repectively. As can be seen in table 1(d), all primitives, which all represent the combinations of three different AMPs with three different Pers. For the test data set, the primitives that were not utilized in the training data set were used. For example, for 100% AMP and 70% Per, the movement primitive 1 was used for the training data of data set 2 (see Table 1(b)), which means that movement primitives 2 and 3 were used as the testing data set of data set 2.

The number of context layers was 3, the 1st layer was the context layer between input and output, and the 2nd and 3rd layers were fast and slow dynamics layers, respectively. The numbers of the context units were 30, 20, and 6 with time constants of 2, 5, and 30 for the 1st, 2nd, and 3rd layers, respectively.

The network was trained 4 times for different data sets - one time per data set, with the method explained in Chapter 4.3 for 15,000 epochs. Data from a single epoch of the training data set was randomly selected to calculate delta error.

As shown in Figure 9, the network could successfully achieve lower feature level generalization through training using even a small amount of data. Training with both data sets 3 and 4 showed almost 100% success rates for test data, whereas training using data sets 1 and 2 showed poor success rates for test data. If the mean KL-divergence for motor output of the data was 0.01, then it was considered successful.

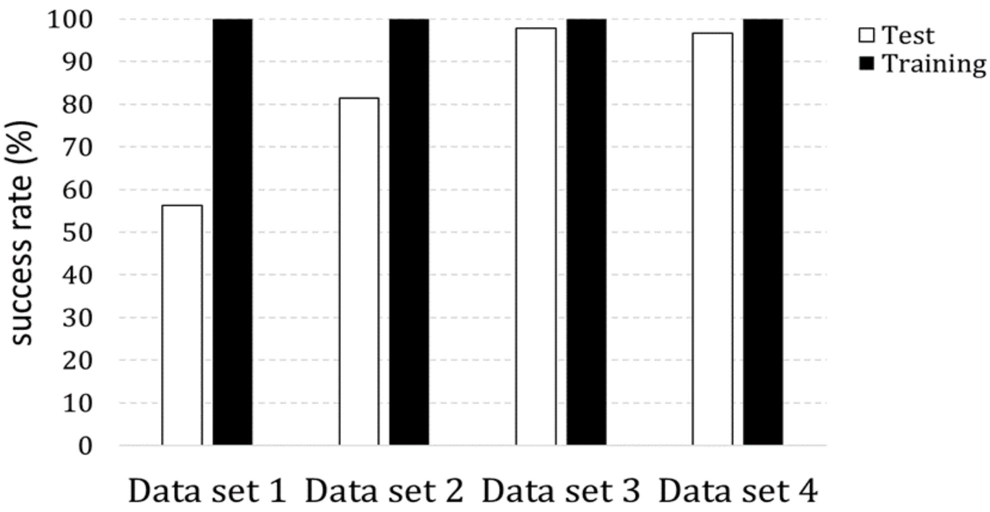

Figure 9. Comparison of the success rates of training data test data for 4 different data sets.

In the case of data set 3, only human movement 1 constituted combinations of 3 different AMPs and 3 different PERs, and movement 2 and movement 3 comprised with 3 different AMPs and 3 different PERs respectively. Although, this experiment was conducted with a small number of movement primitives and variant features, and human movements 2 and 3 also had to comprise with 3 different AMPs and 3 PERs respectively, the network could get excellent results while reducing almost half of the data as compared to data set 4. This result suggests that the network could be efficiently trained to achieve generalization capability for variant features. This could contribute to the break of the 'curse of dimensionality' by reducing the number of training data.

5.2 Experiment 2: Higher level generalization

Acquiring semantic compositionality from experiences is essential for communicating with humans as socially intelligent robots [9]. This is because humans can make numerous expressions even by combining a small number of components in different order. For example, in this experiment a human subject was able to demonstrate 234 different imperative human gestures, and all of them were created by arranging only 3 different human primitives, 2 order commands, and 3 speed commands in different order. Furthermore, same patterns can have different meanings, *e.g.* both forward and reverse order commands could also be used as normal speed command, and *vice versa*. Thus, it is impossible to learn every expression to understand humans' intentions. Socially intelligent robots have to be able to acquire semantic compositionality. As mentioned earlier, the aim of this experiment is to investigate whether the adopted MTRNN can overcome these technical challenges: (1) development of compositional structures with generalization to recognize, as well as generate, sequentially combined movement patterns via iterative learning of continuous perceptual flow, and (2) the ability to extract type structure among a set of demonstrated movement patterns. As Sugita [25] showed, the RNNPB could acquire semantic compositionality from associative learning of word sequences and behaviors with relatively simple tasks, and Yamashita & Tani [26] showed the MTRNN could self-organize functional hierarchy that enables the MTRNN to learn complex behaviors by utilizing its hierarchical structure with different time scales. From these results, it can be expected that the adopted MTRNN could be able to develop compositional structure as well as extract type structure via iterative learning of the lower level perceptual experiences. In experiment 2-1, we examined the capability of the MTRNN by training with part of the imperative human gesture and corresponding robot response pairs, and tested with unlearned pairs. In experiment 2-2, we verified the importance of the explicit temporal hierarchy of the network via comparison of the generalization capability of different types of networks through the employment of some networks with and without explicit temporal hierarchy.

All 234 different imperative human gesture and corresponding robot response pairs were used in this experiment. Among them, two of thirds, *i.e.* 156 pairs, were used as training data and the remaining 78 were used as test data. The training data included validation data which were composed of 12 pairs. The test data consisted of the following data: gesture and response pairs consisting of either unlearned imperative human gestures or robot responses, and both unlearned imperative human gestures and robot responses. For all human gesture and robot response pairs, there were 3 sample-pair combinations for each of the 234 pairs containing the same patterns but with different idle time between human movement patterns and different noise. The total amount of training data was 468, including 36 validation data, and the total amount of test data was 234.

5.2.1 Experiment 2-1

The number of context layers was 4, where the 1st layer was the context layer between input and output, and the 2nd , 3rd , and 4th layers were the fast, intermediated, slow dynamics layers, respectively. The number of contexts units were 30, 30, 20, and 20 with time constants of 2, 5, 10, and 60 for the 1st, 2nd, 3rd , and 4th layers, respectively.

The network was trained 3 times with different initial learnable parameters. The training method explained in Chapter 4.2 was used to train the network. Data from 21 of 432 sample-pair combinations of a single epoch that were not used as validation data were randomly selected and utilized to calculate delta error. As explained in Chapter 4.2, our adaptive learning method also used. The best epoch was selected after 300,000 epochs, while considering both training and validation errors, instead of stopping the training when the validation error starts to increase, because both training and validation errors fluctuated as training proceeded. The epoch that minimizes the following error function (Eq. 5.1) was treated as the best epoch. :

$$E = E_{tr} + \frac{N_{tr}}{N_v} E_v \tag{5.1}$$

where, E_{tr} and E_v are the training and validation errors, and N_{tr} and N_v are the numbers of training and validation data, respectively.

The average success rate of the 3 networks that were trained with different initial learnable parameters was 87.0% for the test data and 99.4% for the training data. As we expected, the network successfully performed the communicative tasks, despite the task having complications such as flexible syntactic rule, where the command could come after the first, second, and third human movement primitives, and the potential problem of the same movement pattern possibly containing different meanings depending on the previously demonstrated movement patterns. This result indicates that the network can acquire task rules hidden in continuous flow, and self-organize into a compositional structure with generalization for performing the communicative tasks via iterative learning of continuous perceptual flow by utilizing its temporal hierarchical structure. Figure 10 and Figure 11 display examples of the results.

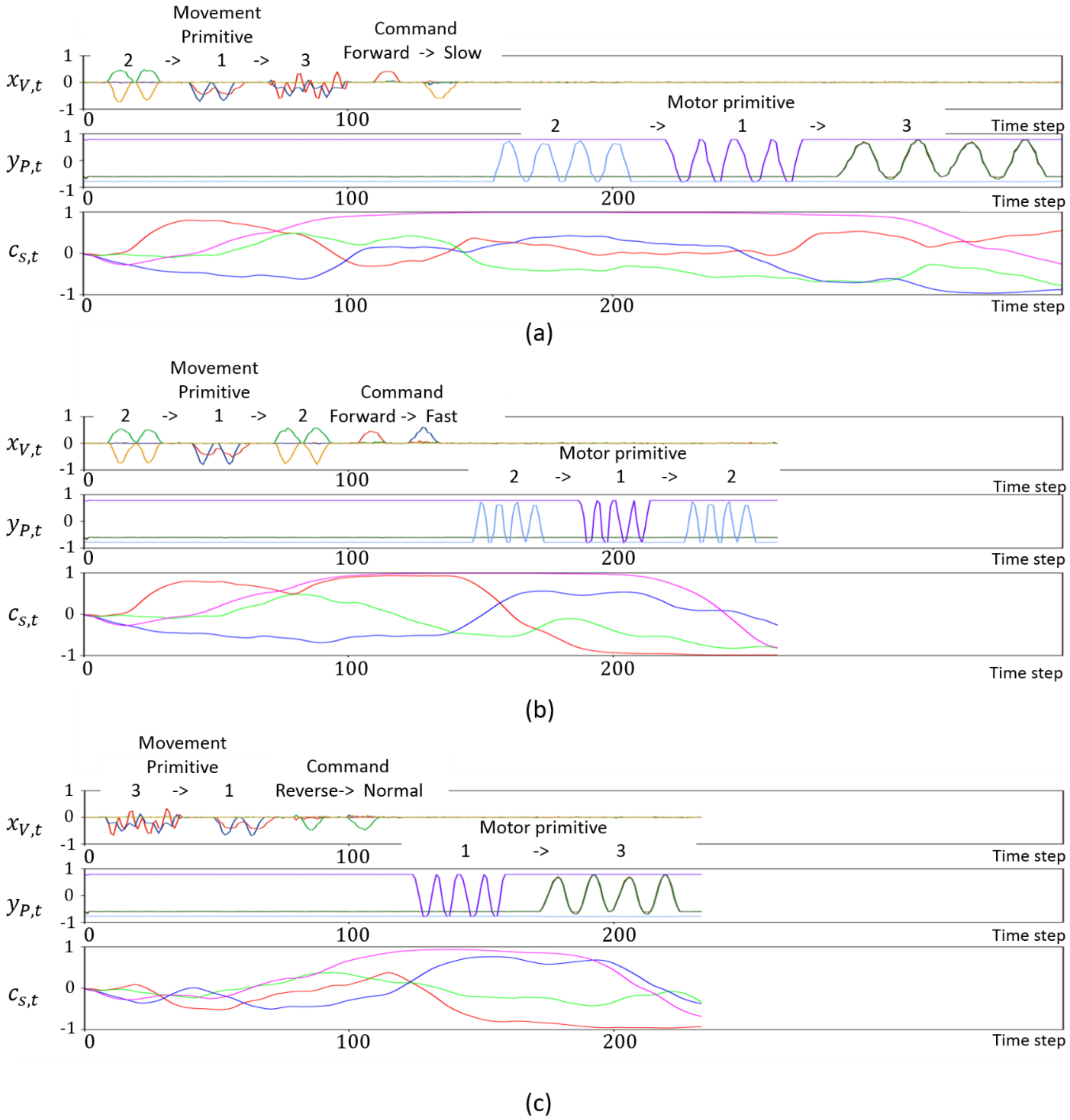

Figure 10. Examples of the test results, plotted with slow context activities. $x_{v,t}$, $y_{p,t}$, and $c_{s,t}$ indicates vision input, proprioception output and slow context. Four selected slow context units were plotted in the bottom window of Figure 10(a), (b), and (c).

As illustrated in Figure 10(a) and (b), slow context unit activities remained the same while perceiving the same sequence of human movement patterns, until the perception of the second movement primitive, and were differentiated after perceiving different movement patterns, from the third movement primitive. After perceiving the different imperative human gestures, the network correctly generated different robot responses that corresponding to the perceived imperative human gestures. This means that the slow dynamics encoded even unlearned sequences of the human movement patterns, and the network utilized the neural representations of the slow context units to generate adequate robot responses. Also, as we can see Figure 10(a), (b), and (c), the slow dynamics subnetwork was tolerant enough to encode the differentially combined movement patterns. Contrary to the slow dynamics subnet, the fast dynamics subnet encoded each human movement pattern and robot motor primitive, as can be seen Figure 11.

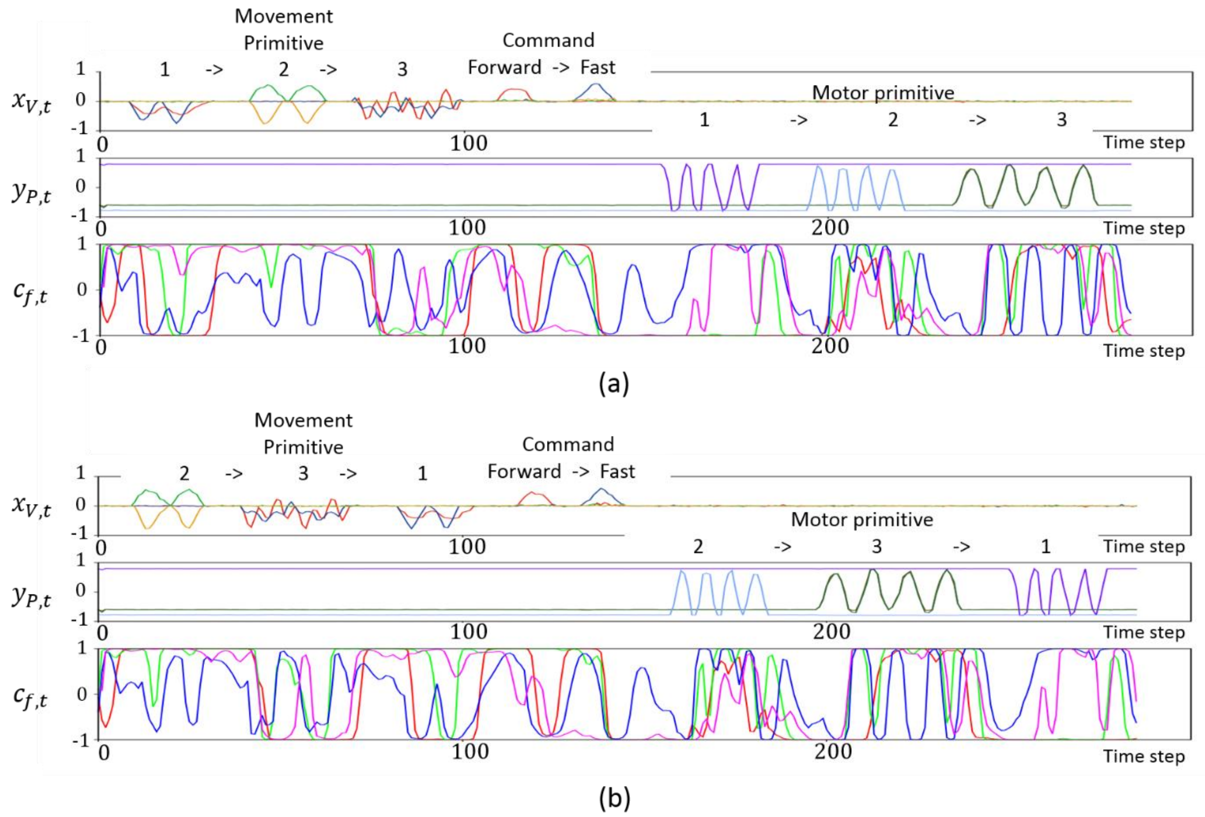

Figure 11. Examples of the test result, plotted with fast context activities. $x_{v,t}$, $y_{p,t}$, and $c_{f,t}$ indicates vision input, proprioception output and fast context. Four selected fast context units were plotted in the bottom window of Figure 11(a) and (b).

Comparing Figure 11(a) and (b) shows that fast context activities were quite similar for the same patterns regardless of the timing pattern introduction. However, the fast contexts activities for the same human movement patterns showed somewhat different activities, especially during the early phase of each movement pattern, whereas the fast context activities for the same robot motor primitives were almost the same. This could suggest that the differences in fast contexts activities for same movement patterns could be due to the unpredictability of the movement patterns. Although the network was trained to predict next step input, its prediction could be incorrect, while perceiving imperative human gestures, it may take a time to revise its prediction, however, on the other hand, the robot responses were deterministic for give imperative human gestures. Nevertheless, it seems that the network successfully self-organized functional hierarchy and efficiently represented both patterns and their sequences.

From this experiment, we demonstrated that the network successfully overcame the aforementioned technical challenges. Despite this, there is also potential for improving the results. One example can be seen in the initial parameter sensitivity of the result. As explained, the network was trained 3 times with different initial parameters and the success rates for the test data were 94.4, 74.4 and 92.3%. Another flaw the absence of the error correcting process. In the current recognition/generation method, the network could not correct prediction errors that could lead to failure of task performance. Future work therefore should adopt the methods for mitigating initial parameter sensitivity of the result, such as neural network committee [52], and unsupervised pre-training [53], and for correcting prediction error such as error regression [35]. Nonetheless, the result of this experiment can be significant to the further development of social intelligence and understanding of the underlying mechanism of social cognitive functions.

5.2.2 Experiment 2-2

In Experiment 2-1, the temporal hierarchical structure of the network was treated as one of the key factor in achieving the goal. Despite this, it may natural to be suspicious of the importance of the temporal hierarchy of the network. Thus, this experiment was conducted to evaluate the importance of the temporal hierarchy with different network structures.

To evaluate the importance of the multiple time constants and hierarchical structure, 4 different types of networks were tested: single-layer with single time constant, single-layer with multiple time constants, multi-layer with single time constant and multi-layer with multiple time constants. Each network had 100 contexts units. Details are shown in Table 2.

Table 2. Configuration of the networks.

		1st layer	2nd layer	3rd layer	4th layer
Network 1	τ	2	5	10	60
	N_c	30	30	20	20
Network 2	τ	2	5	10	100
	N_c	30	30	20	20

(a) Multi-layer & Multiple time constants

		1st layer	2nd layer	3rd layer	4th layer
Network 3	τ	2	2	2	2
	N_c	30	30	20	20
Network 4	τ	20	20	20	20
	N_c	30	30	20	20
Network 5	τ	60	60	60	60
	N_c	30	30	20	20

(b) Multi-layer & Single time constant

		1st layer			
Network 6	τ	2	5	10	60
	N_c	30	30	20	20
Network 7	τ	2	5	10	100
	N_c	30	30	20	20

(c) Single-layer & Multiple time constants

		1st layer
Network 8	τ	2
	N_c	100
Network 9	τ	20
	N_c	100
Network 10	τ	60
	N_c	100

(d) Single-layer & Single time constant

τ and N_c are the time constant of the context units and number of the context units, respectively. In the case of Network 1 (see Table 2(a)), for example, the number of context units were 30, 30, 20, and 20, each with a time constant of 2, 5, 10, and 60 for the 1st, 2nd, 3rd and 4th layers respectively. The input and output were connected to the 30 contexts with the smallest time constant among the networks context units, and context units in the 1st layer in multiple-layer network case. For the networks with a single time constant, either 2 or 20 or 60 was used for the time constant. For the network with multiple time constants, two different sets of the time constants (2, 5, 10, and 60, and 2, 5, 10, and 100) were used for the time constant to examine whether the results were seriously affected by the time constants. The networks were training using the same training method and procedure as in Experiment 2.

It was revealed that the explicit temporal hierarchy of the network was important for generalization. As we can see in Figure 12, the networks with temporal hierarchy showed best results among different types of the networks.

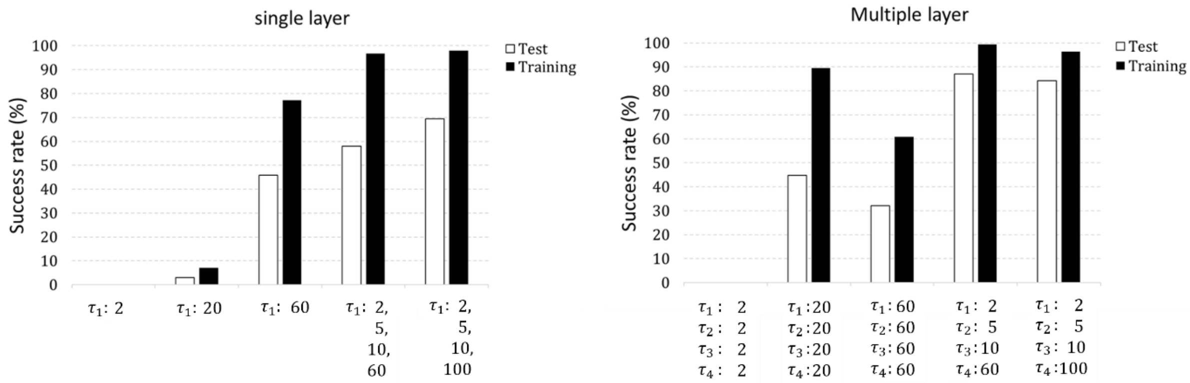

Figure 12. Comparison of the networks in the results.

Also, both networks with temporal hierarchy showed similar results. These results are consistent with previous study [26] that showed the performance of the network was not sensitive to the time constant if the ratio between the fast time constant and slow time constant is sufficiently large. Regardless of the number of layers, the network with a single time constant showed much worse results than the networks with multiple time constants. Also, the single-layer networks with multiple time constants could successfully perform experienced tasks but showed low generalization results. Furthermore, we tested with another type of the multiple-layer network that had multiple time constants in each layer. In that case, both training and test results were worse than the networks with explicit temporal hierarchy. Thus, it could be concluded that explicit temporal hierarchy of the network is important in developing higher-order competency.

5.3 Experiment 3: Development of metacognition

In Experiment 1, the network was able to be efficiently trained to achieve generalization ability by developing representation that is robust to variant features as a result of utilizing its slow dynamics subnet. From Experiment 2-1 it was concluded that the network could overcome previously mentioned technical challenges (1) and (2). From experiment 2-2, the importance of the temporal hierarchical structure of the network for overcoming these technical challenges was verified.

However, utilizing slow dynamics subnets has potential problems. The problems were not visible in Experiment 1 and 2 because all the data used in those experiments consisted of a single task session. For successively performing the tasks, it is necessary to train the network on data which each contain several task sessions. When the network is trained on those data, it is possible that the network may try to model dependencies between each task session although each task session is independent from each other. Also, the performance of the current task session could be affected by previous task sessions. To successively perform the tasks while avoiding these problems, the network had to regulate its memory capacity when it was necessary to successfully learn the tasks. The network should be able to control contextual flow, *i.e.* adequately remember as well as forget depending on the context. In this context, this experiment was designed to focus on technical challenge (3): the development of meta-level cognition for controlling the turn-taking process as well as controlling the contextual flow in continuously on-going task processes.

In this experiment, 36 different human gesture and robot action pairs were used. Each task session was composed of an imperative human gesture followed by a corresponding robot response. The amount of training data and test data were 18 and 24, respectively, with each data set containing 18 and 24 randomly concatenated task sessions, respectively. Every 36 gesture and response pairs appeared 9 times and 12 times in training and test data sets, respectively. The same network configuration and training method used in Experiment 2 was applied in this experiment again. The training proceeded for 60,000 epochs. Data from 4 of 18 concatenated task sessions of a single epoch were randomly selected to calculate delta error. After training with data that contained randomly concatenated task sessions, the network became able to successfully perform the tasks regardless of the previous task sessions. Also, the network was able to perform a turn-taking process, which proceeded as follows: first the network perceived a human subject demonstrating gestures until the human demonstrated a speed command while the robot remained still, and then the robot started to generate robot responses, and finally proceeded to the next task session. The mean KL-divergences of the proprioception output were 0.0032 for training data and 0.0034 for test data. Examples of the test results are shown in Figure 13.

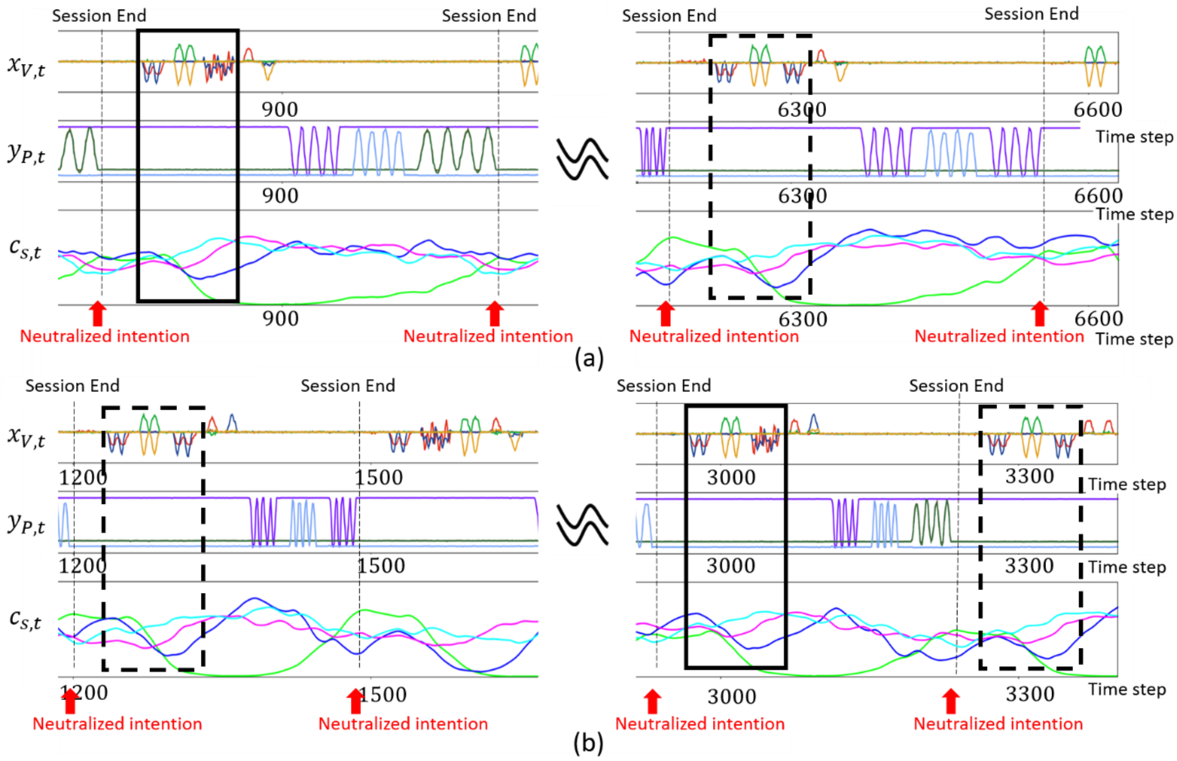

Figure 13. Examples of the test results, plotted with slow context activities. $x_{v,t}$, $y_{p,t}$, and $c_{s,t}$ indicates vision input, proprioception output and fast context. Four selected slow context units were plotted in the bottom window of Figure 13(a) and (b).

In Figure 13, the same imperative gestures are marked using dashed and solid line rectangles. As shown in Figure 13, slow context activities at the end of each task session were remarkably similar. It could be assumed that the network autonomously neutralized its contexts states to be unaffected by the previous task sessions. Although slow context unit activities for the same imperative gestures showed little differences in the beginning of each task session, they quickly became quite similar as the task proceeded. To verify whether the network developed contextual flow controlling skills, the differences between the slow context activities at the end step of each task session were measured in the following way, and compared with the result coming from the training with data which contained only a single task session in each data. For comparing, for both networks, training with concatenated data and non-concatenated data, slow context unit activities at the end step of the robot responses after performing every single sessions were extracted. After that, for each slow context unit, variances between activities after performing different task sessions were calculated, and then mean variance of those variances were calculated. For the network training with concatenated data, mean variance was only 0.063, whereas mean variance was 0.378 for the network trained on non-concatenated data. It is apparent that the network was able to control its contextual flow via training when using concatenated data.

It is evident from the results that through iterative learning the network was able to develop metacognitive skills [54]. The term metacognitive skills here is described in the following manner. First, metacognitive knowledge is knowing what kind of task the network was performing to enable the network to perform better. More specifically, each task session was independent, forgetting memory of the current task session after completing the task would be beneficial for next session performance. Second, adequate use of metacognitive knowledge to efficiently and successfully learn and perform tasks through controlling contextual flows, as well as utilizing turn-taking process. Despite the results, the amount of training data needed to be able to successfully perform the task was relatively large, and suggesting that training of the network using error minimization rule and the BPTT algorithm could be compared with the human developmental process is still controversial. However, it could be concluded that the result of this experiment could contribute to the development of a socially intelligent robot.

Chapter 6. Discussion

It should once again be noted that current human-robot communicative interaction realizes only one-way imperative communication from human to robot and is incapable of bi-directional interaction. There have been a few research studies that have explored bi-directional communicative interactions based on the learning of others. The aforementioned study on robot imitation learning [14] conducted an additional experiment on a mutual imitation game. In this experimental game, the robot had been trained on four movement patterns and the participating human subjects were unaware of what the robot had learned. In the imitation game, the subjects were instructed to identify as many movement patterns as possible and to synchronize their movements with those of the robot through interactions. It was observed that various emergent phenomena took place during the trials of mutual interactions. One interesting phenomena was the illustration of turn-taking-like behaviors, in which the roles of the imitator and imitatee switched spontaneously between the two.

In the study of the mutual imitation game by [14], learning took place only on the human subject side after learning on the robot side was terminated. Ikegami and Iizuka [55] explored the possible effects of mutual learning about behavioral scheme of each other on turn taking phenomena between follower and followee in a simulation experiment on adaptive mobile agents. They reported that the mutual learning induces intermittency in occurrence of turn-taking.

The current study can also be extended to achieve mutual communicative interactions accompanied by mutual learning between robot and human for future study. This, however, could bring on another difficult issue of dynamic or incremental learning of neural network models under dynamic changes of interrelation between the two sides. Especially, future studies should examine how compositionality can be developed in the course of dynamic learning.

Now, it is interesting to compare the current scheme which does not utilize prediction error explicitly for reading others' intentions, and the prediction coding scheme which utilizes prediction error for inferring intention of others, as shown in [14, 56]. It can be said that intentions are read unconsciously in the former model and consciously in the latter model if we assume that consciousness originates from prediction error [57, 58]. It has been known that imitation can be performed for both meaningless movement without objects and meaningful motor acts with target objects. Rizzolatti and colleagues [59] observed in their monkey experiments that mirror neurons in the parietal lobe and F5 fire in the former and the latter case, respectively. They consider that conscious imitation of meaningful motor acts – *i.e.* motor actions performed to reach a target object, may be accompanied by the firing of mirror neurons in F5. Furthermore, unconscious imitation without meaning can be observed in human neonates, while conscious imitation with meaning can be observed in infants after 2 years old. Meltzoff [60] explained the development of imitation mechanism by proposing a hypothesis on a "like me" mechanism which connects the perceptions of others as "like me" and understanding of others' minds. In the first stage in newborn, innate sensory-motor mapping can generate the aforementioned imitative behaviors by means of automatic response. In the second stage, infants develop an interdependent relationship between their mental state and performed actions through repetition, which thereby affords learning. Finally in the third stage, infants come to understand that others who act "like me" have mental states "like me". From these accounts, it can be said that the current model explains an unconscious pathway of communicative interactions. The current model could be developed toward conscious goal-directed communicative interaction if incorporated with an error regression scheme for inferring the intention of others. Future studies will investigate this possibility.

In Experiment 2, the MTRNN showed relatively good generalization capability. However, some limitations of the results are worth noting. The results were somewhat sensitive to the initial learnable parameters. This problem could be solved by introducing the neural network committee [52]. It can be said that the network was able to perform the task for all human gesture and robot response pairs since the network failed only for different pairs inside the test data set. This possibility to solve initial parameter sensitivity of the results should be verified. With that stated, the more challenging problem for future studies is, the generalization capability of the network in the current research. The network was trained using 2/3 of all possible pairs and showed good generalization results for the unlearned 1/3 pairs. However, this generalization capability is still incomparable to humans' generalization capability. Future research is needed to enhance generalization capability.

References

[1] C. Breazeal, "Social interactions in HRI: the robot view," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 34, pp. 181-186, 2004.

[2] K. Dautenhahn, "Socially intelligent robots: dimensions of human–robot interaction," *Philosophical Transactions of the Royal Society B: Biological Sciences,* vol. 362, pp. 679-704, 2007.

[3] R. A. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of,* vol. 2, pp. 14-23, 1986.

[4] A. Billard and K. Dautenhahn, "Grounding communication in autonomous robots: an experimental study," *Robotics and Autonomous Systems,* vol. 24, pp. 71-79, 1998.

[5] M. Fujita and K. Kageyama, "An open architecture for robot entertainment," in *Proceedings of the first international conference on Autonomous agents*, 1997, pp. 435-442.

[6] Aldebaran., "*who-is-pepper*." Available: http://www.aldebaran.com/en/a-robots/who-is-pepper, 2014.

[7] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: a survey," *Connection Science,* vol. 15, pp. 151-190, 2003.

[8] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa*, et al.*, "Cognitive developmental robotics: a survey," *Autonomous Mental Development, IEEE Transactions on,* vol. 1, pp. 12-34, 2009.

[9] A. Cangelosi, G. Metta, G. Sagerer, S. Nolfi, C. Nehaniv, K. Fischer*, et al.*, "Integration of action and language knowledge: A roadmap for developmental robotics," *Autonomous Mental Development, IEEE Transactions on,* vol. 2, pp. 167-195, 2010.

[10] P. Gaussier, S. Moga, M. Quoy, and J.-P. Banquet, "From perception-action loops to imitation processes: A bottom-up approach of learning by imitation," *Applied Artificial Intelligence,* vol. 12, pp. 701-727, 1998.

[11] S. Schaal, "Is imitation learning the route to humanoid robots?," *Trends in cognitive sciences,* vol. 3, pp. 233-242, 1999.

[12] A. Billard, "ft Imitation: A Means to Enhance Learning of a Synthetic Protofanguage in Autonomous Robots," *Imitation in animals and artifacts,* p. 281, 2002.

[13] J. Demiris and G. Hayes, "f 3 Imitation as a Dual-Route Process Featuring Predictive and Learning Components; 4 Biologically Plausible Computational Model," *Imitation in animals and artifacts,* p. 327, 2002.

[14] M. Ito and J. Tani, "On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system," *Adaptive Behavior,* vol. 12, pp. 93-115, 2004.

[15] H. Iizuka and T. Ikegami, "Adaptability and diversity in simulated turn-taking behavior," *Artificial Life,* vol. 10, pp. 361-378, 2004.

[16] J. Nadel, A. Revel, P. Andry, and P. Gaussier, "Toward communication: first imitations in infants, low-functioning children with autism and robots," *Interaction Studies,* vol. 5, pp. 45-74, 2004.

[17] Y. Nagai, K. Hosoda, A. Morita, and M. Asada, "A constructive model for the development of joint attention," *Connection Science,* vol. 15, pp. 211-229, 2003.

[18] J. Triesch, C. Teuscher, G. O. Deák, and E. Carlson, "Gaze following: why (not) learn it?," *Developmental science,* vol. 9, pp. 125-147, 2006.

[19] D. J. Lewkowicz and R. Lickliter, "A dynamic systems approach to the development of cognition and action," *Journal of Cognitive Neuroscience,* vol. 7, pp. 512-514, 1995.

[20] J. S. Kelso, *Dynamic patterns: The self-organization of brain and behavior*, 1997.

[21] R. F. Port and T. Van Gelder, *Mind as motion: Explorations in the dynamics of cognition*: MIT press, 1995.

[22] A. Clark, "An embodied cognitive science?," *Trends in cognitive sciences,* vol. 3, pp. 345-351, 1999.

[23] R. D. Beer, "Dynamical approaches to cognitive science," *Trends in cognitive sciences,* vol. 4, pp. 91-99, 2000.

[24] J. Tani, "Self-organization of symbolic processes through interaction with the physical world," in *IJCAI*, 1995, pp. 112-118.

[25] Y. Sugita and J. Tani, "Learning semantic combinatoriality from the interaction between linguistic and behavioral processes," *Adaptive Behavior,* vol. 13, pp. 33-52, 2005.

[26] Y. Yamashita and J. Tani, "Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment," *PLoS computational biology,* vol. 4, p. e1000220, 2008.

[27] M. Maniadakis, P. Trahanias, and J. Tani, "Self-organizing high-order cognitive functions in artificial agents: Implications for possible prefrontal cortex mechanisms," *Neural Networks,* vol. 33, pp. 76-87, 2012.

[28] J. Tani, "Model-based learning for mobile robot navigation from the dynamical systems perspective," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 26, pp. 421-436, 1996.

[29] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena,* vol. 42, pp. 335-346, 1990.

[30] M. I. Jordan, "Serial order: A parallel distributed processing approach," *Advances in psychology,* vol. 121, pp. 471-495, 1997.

[31] J. L. Elman, "Finding structure in time," *Cognitive science,* vol. 14, pp. 179-211, 1990.

[32] D. RUMELHART, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error propagation," *Parallel Distributed Processing,* pp. 318-362, 1986.

[33] J. L. McClelland, D. E. Rumelhart, and P. R. Group, "Parallel distributed processing," *Explorations in the microstructure of cognition,* vol. 2, 1986.

[34] M. I. Jordan and D. E. Rumelhart, "Forward models: Supervised learning with a distal teacher," *Cognitive science,* vol. 16, pp. 307-354, 1992.

[35] J. Tani, M. Ito, and Y. Sugita, "Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using RNNPB," *Neural Networks,* vol. 17, pp. 1273-1289, 2004.

[36] R. D. Beer, "Parameter space structure of continuous-time recurrent neural networks," *Neural Computation,* vol. 18, pp. 3009-3051, 2006.

[37] M. L. Forcada and R. C. Carrasco, "Finite-state computation in analog neural networks: steps towards biologically plausible models?," in *Emergent neural computational architectures based on neuroscience*, ed: Springer, 2001, pp. 480-493.

[38] D. Badre, "Cognitive control, hierarchy, and the rostro–caudal organization of the frontal lobes," *Trends in cognitive sciences,* vol. 12, pp. 193-200, 2008.

[39] G. Rizzolatti and G. Luppino, "The cortical motor system," *Neuron,* vol. 31, pp. 889-901, 2001.

[40] E. Koechlin, C. Ody, and F. Kouneiher, "The architecture of cognitive control in the human prefrontal cortex," *Science,* vol. 302, pp. 1181-1185, 2003.

[41] M. A. Arbib, *Neural organization: Structure, function, and dynamics*, 1998.

[42] F. A. Mussa–Ivaldi and E. Bizzi, "Motor learning through the combination of primitives," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences,* vol. 355, pp. 1755-1769, 2000.

[43] M. Botvinick, "Hierarchical models of behavior and prefrontal function," *Trends in cognitive sciences,* vol. 12, pp. 201-208, 2008.

[44] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *Neural Networks, IEEE Transactions on,* vol. 5, pp. 157-166, 1994.

[45] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *arXiv preprint arXiv:1211.5063,* 2012.

[46] Z. Zhang, "Microsoft kinect sensor and its effect," *MultiMedia, IEEE,* vol. 19, pp. 4-10, 2012.

[47] C. M. Bishop, *Pattern recognition and machine learning* vol. 1: springer New York, 2006.

[48] F. Alnajjar, Y. Yamashita, and J. Tani, "The hierarchical and functional connectivity of higher-order cognitive mechanisms: neurorobotic model to investigate the stability and flexibility of working memory," *Frontiers in neurorobotics,* vol. 7, 2013.

[49] S. Heinrich and S. Wermter, "Interactive Language Understanding with Multiple Timescale Recurrent Neural Networks."

[50] J. Namikawa, R. Nishimoto, and J. Tani, "A neurodynamic account of spontaneous behaviour," *PLoS computational biology,* vol. 7, p. e1002221, 2011.

[51] S. Duffner and C. Garcia, "An online backpropagation algorithm with validation error-based adaptive learning rate," in *Artificial Neural Networks–ICANN 2007*, ed: Springer, 2007, pp. 249-258.

[52] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, 2011, pp. 1135-1139.

[53] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?," *The Journal of Machine Learning Research,* vol. 11, pp. 625-660, 2010.

[54] J. A. Livingston, "Metacognition: An Overview," 2003.

[55] T. Ikegami and H. Iizuka, "Turn-taking interaction as a cooperative and co-creative process," *Infant Behavior and Development,* vol. 30, pp. 278-288, 2007.

[56] S. Murata, Y. Yamashita, H. Arie, T. Ogata, J. Tani, and S. Sugano, "Generation of Sensory Reflex Behavior versus Intentional Proactive Behavior in Robot Learning of Cooperative Interactions with Others."

[57] J. Tani, "An interpretation of theself'from the dynamical systems perspective: a constructivist approach," *Journal of Consciousness Studies,* vol. 5, pp. 5-6, 1998.

[58] J. Tani, "Autonomy of Self at criticality: The perspective from synthetic neuro-robotics," *Adaptive Behavior,* vol. 17, pp. 421-443, 2009.

[59] G. Rizzolatti, L. Fogassi, and V. Gallese, "Neurophysiological mechanisms underlying the understanding and imitation of action," *Nature Reviews Neuroscience,* vol. 2, pp. 661-670, 2001.

[60] A. N. Meltzoff, "Imitation and other minds: The “like me” hypothesis," *Perspectives on imitation: From neuroscience to social science,* vol. 2, pp. 55-77, 2005.

Summary

Development of Compositional and Contextual Communicative Skills of Robot by Using a Neuro-Dynamic Model

이 연구에서는 사람과 의사 소통을 하기 위한 복잡한 의사 소통 기술들을 학습을 통해 획득하는 로봇 실험을 진행하였다. 로봇을 제어하기 위한 두뇌 모델로는 뇌신경계를 모방한 동적 신경망 모델을 사용하였다. 사용된 동적 신경망 모델은 심층 구조 (deep structure) 속 각 층의 동적 특성 (dynamics property)이 서로 다른 시간 척도 (timescale)를 가지는 multiple timescale dynamics property를 가지는 특징이 있다. 이 실험에서 로봇은 특정한 의사 전달 규칙에 의해 전달되는 사람의 명령 투의 제스처를 인식하고 그에 맞는 순차적인 행동을 생성해 내도록 훈련이 되었다. 이 실험의 결과는 사용된 동적 신경망 모델이 사람과 반복적으로 의사 소통을 하면서 일련의 과정 속의 연속적인 감각운동 흐름을 학습하여 다음과 같은 능력을 개발했음을 보여 준다. (1) compositional structure의 발달을 통해 학습하지 않았던 방식으로 연결된 순차적인 행동 패턴까지 인식하고 생성해 내는 능력, (2) 행동 패턴들의 종류를 구별해 냄으로써 명령적 제스처 속에 있는 근원적인 규칙을 이해하는 능력, (3) 순서 주고 받기를 할 뿐만 아니라 작업 기억 (working memory)을 수행 통제 (executive control)하여 작업 맥락을 상황에 맞게, 필요에 따라 기억하고 망각하는 메타 레벨 인지 능력. 훈련된 신경망 모델의 동적 특성을 분석한 결과는 신경망 모델이 앞서 언급한 고위 인지 (higher-order cognitive) 능력과 메타 레벨 인지 능력을 multiple timescale dynamics property를 활용하여 적절한 기능적 구조 (functional hierarchy)를 자기 구조화함으로써 개발했음을 보여 준다. 이 연구의 결과들은 고차원적인 의사 소통 능력을 가진 사회적 로봇을 발달시키는데 기여할 수 있을 것이다.

핵심어: 신경망, 메타 인지, 고위 인지, 자기구조화, 합성성, 시간적 계층 구조, 신경로봇

Acknowledgment

Foremost, I would like to express my special thanks to my advisor Professor Dr. Jun Tani. I would like to thank him for encouraging my research and for allowing me to grow as a research scientist. His advice on both research as well as on my career have been priceless. I would also like to thank my committee members, Professor Dr. Dae-Shik Kim, Professor Dr. Junmo Kim for serving as my committee members. I also want to thank them for letting my defense be a valuable moment, and for their superb comments and suggestions, thanks to you. I would especially like to thank colleagues at my laboratory, cognitive neuro-robotics lab. All of them have been there to support me.

This work was supported by the Industrial Strategic Technology Development Program(10044009, Development of a self-improving bidirectional sustainable HRI technology for 95% of successful responses with understanding user’s complex emotion and transactional intent through continuous interactions) funded by the Ministry of Knowledge Economy(MKE, Korea)