

Acquisition and Extension of
the Robot Body Model Using
Motor Babbling through Deep Learning
ディープラーニングを利用した
モーターバブリングによる
身体モデルの獲得と拡張

February 2017

Kuniyuki Takahashi
高橋 城志

**Acquisition and Extension of
the Robot Body Model Using
Motor Babbling through Deep Learning**
ディープラーニングを利用した
モーターバブリングによる
身体モデルの獲得と拡張

February 2017

Waseda University

Graduate School of Creative Science and Engineering,
Department of Modern Mechanical Engineering,
Research on Intelligent Machines

Kuniyuki Takahashi

高橋 城志

Abstract

Today, intelligent robots are expected to play an important role in supporting humans in their daily activities. Conventional robots such as industrial robots are primarily designed to perform designated tasks with high speed and precision and would require extra modifications to perform other tasks. In addition, these robots are typically applicable only in highly controlled environments that have no or negligible uncertainty and are free of human movement. Recently, there has been a growing interest in the development of robots working closely with humans. In real environments, where humans exist, it is essential that robots would be required to adapt to dynamic and uncertain environments and cope with new tasks and situations in a short duration. Robot with tool use skills would be particularly useful in human society, as this would enable the robot to expand its capabilities in performing tasks. To perform various tasks, robots should have complex structure such as large numbers of applicable sensors and degrees of freedom (DOFs). Designing robot models manually has become difficult because of the above reasons. To address this challenge, I developed a model in which the robot body and environment is based on experiences the robot acquires through physical (embodied) interactions as opposed to models in which the robot body and environment is pre-designed; that is, the robot learns sensorimotor relationships from experience using machine learning, particularly deep learning. The focus of this approach is *embodiment*. In this thesis, I focus specifically on the *body model* acquired from interaction with the environment through embodiment.

Enabling robots to act intelligently is usually referred to as *artificial intelligence*. *Symbolic AI* is based on the manipulation of symbols, an assumption defined as the physical symbol systems hypothesis. These approaches are only valid if all possible

situations are described. However, it is difficult to describe all possible situations in the real world. In contrast, Brooks proposed a subsumption architecture in which the system and environment are connected using a connection between the sensor and motor without a description of the symbols. This approach is referred to as *embodied AI*. It is hypothesized that for complex intelligence, in particular for human-like intelligence, a human-like body is necessary for the robot to interact with the environment. To consider the complex relationship between the sensor and motor, machine-learning, particularly by using deep learning, has been proven effective. Deep learning has desirable characteristics, such as the ability to obtain individual features from data. The term *deep learning* refers to the use of deep neural networks (DNNs) and recurrent neural networks (RNNs). DNNs are feed-forward neural networks, whereas RNNs are recurrent connections.

When considering a robot body, body schema is an important concept used to explain the body model. Body schema involves mapping between the motor and the sensors located on the body and includes information about the size and shape of the body. In addition, body schema is not fixed after acquisition but is, instead, a plastic model. In particular, when humans are proficient in using a tool, they behave as if their bodies extend to the edge of the tool that they use. The phenomenon is referred to as *tool-body assimilation*. Owing to body schema, humans have the capability of recognizing the tool tip posture, position, and motion because their body schema is plastic with respect to their body and the tool. Hence, body schema can connect the body and tool easily; that is, it is possible to describe the tool using the model to extend the body schema. When a body schema is considered from an engineering perspective, it is a multi-modal association of motion and sensors. In my research, I added the concept of the forward/inverse model (In this study, the forward model predicts the next state of the arm-tip position from the current state of the joint angles, and the inverse model predicts the next state of the joint angles from the current state of the arm-tip position). I refer to it as the *body model*, and in particular, I refer to it when considering the dynamics as *body dynamics*.

The objective of my thesis is to propose a machine learning framework that the robot acquires its body model from the experiences it gains through embodiment

without pre-designed robot-environment model to adapt to dynamic and uncertain environments and cope with new tasks and situations in a short duration. Specifically, I focus on the acquisition and extension of the robot body model using motor babbling through deep learning to realize dynamic motion learning for flexible-joint robots and tool use with a tool-body assimilation model.

This thesis is organized into four chapters. Chapter 1 provides the background, research objective, related work, and overview of the proposed approach as an introduction of the current study about acquisition and extension of the robot body model using motor babbling through deep learning.

In chapter 2, I propose a learning strategy for robots with flexible joints having multi-DOFs in order to achieve dynamic motion tasks. In spite of there being several potential benefits of flexible-joint robots such as exploitation of intrinsic dynamics and passive adaptation to environmental changes with mechanical compliance, controlling such robots is challenging because of their increasingly complex dynamics. To achieve dynamic movements exploiting such benefits of flexible-joint dynamics, I introduce a two-phase learning framework of the *body dynamics* of the robot using a RNN motivated by a recent deep learning strategy consisting of *pre-training* and *fine-tuning*. This two-phase learning methodology comprises a *pre-training* phase with motor babbling and a *fine-tuning* phase with additional learning of dynamic motion tasks. In the pre-training phase with motor babbling, I consider *active* and *passive* exploratory motions in order to efficiently learn body dynamics. In the fine-tuning phase, the learned body dynamics are adjusted for specific tasks. I demonstrate the effectiveness of the proposed methodology in achieving dynamic tasks involving constrained movement requiring interactions with the environment on a simulated flexible joint robot model as well as in hardware experiments using a PR2 robot both of which have a seven DOF redundant arm. The results illustrate a reduction in the required number of training iterations for task learning as well as generalization capabilities for untrained situations with the learned body dynamics through motor babbling. In addition, I discuss the issues regarding the trade-off between task training iterations and the success rate of task execution. Furthermore, I discuss the small number of exploratory motor-babbling motions for body dynamics.

In chapter 3, I propose a tool-body assimilation model that considers grasping during motor babbling for using tools. A robot with tool-use skills could be useful in a human–robot symbiosis situation because this allows the robot to expand its task performing abilities. In the existing tool-use model, the body and tool models are separated. Therefore, the motions during tool use are expressed by each tool. To address these issues, I adopted *tool-body assimilation*. In a tool-body assimilation model, the tool model is expressed as a body with a tool. Therefore, the motions during tool use are expressed by the body model, and the robot effectively learns the tool functions. Almost all existing studies for robots to use tools require predetermined motions and tool features; the motion patterns are limited and the robots cannot use novel tools. Some of the other past studies fully search for all available parameters for novel tools, but this leads to massive amounts of calculations. Other past studies approaches were mainly focused on obtaining the functions of the tools, and showed the robot starting its motions with a tool pre-attached to the robot. This implies that the robot would not be able to decide whether and where to grasp the tool. In real life environments, robots would need to consider the possibilities of tool-grasping positions, then grasp the tool. To address these issues, the robot performs motor babbling by *grasping* and *nongrasping* the tools to learn both tool functions and the robot’s body model. In addition, the robot grasps various parts of the tools to learn the different tool functions from the different grasping positions. These rich motion experiences are learned using deep learning. Tool features were self-organized in parametric bias, modulating the body model according to the tool in use. Finally, I designed a neural network for the robot to generate motion only from the target image. To evaluate the model, I have the robot manipulate an object task without any tools or with several tools of different shapes. I have the robot generate motions after showing the initial and target states by deciding whether and where to grasp the tool. Therefore, the robot is capable of generating the correct motion and grasping decision when the initial and target states are provided to the robot.

In Chapter 4, the achievements of a series of both numerical and robot experiments of acquisition and extension of the robot body model. Finally, reviews on the remaining research topics and future directions conclude this thesis.

Acknowledgment

This work was carried out at the Graduate School of Creative Science and Engineering, Waseda University in 2013-2017. Here, I would like to express my sincere thanks and appreciation to those who were involved in my research and life past four years.

Firstly, I would like to gratefully and sincerely thank my supervisor, Prof. Shigeki Sugano, for providing me this precious research opportunity. Over the last seven years including bachelor's and master's degrees, his significantly important comments and suggestions on both my research and life as a researcher gave me extremely splendid and abundant experiences in this laboratory. I would also like to thank my examiners, Prof. Atsuo Takanishi, Prof. Tetsunori Kobayashi, Prof. Mitsuo Umezu, Prof. Jun Ohya, Prof. Tetsuya Ogata, and Prof. Hiroyasu Iwata, for providing me accurate and helpful comments on my presentation and thesis. Especially, I am very grateful to Prof. Tetsuya Ogata for allowing me to be a member of his laboratory from second years of the master course and I have always deeply impressed by for brilliant ideas, valuable advice, and extensive knowledge.

I would like to thank Prof. Gordon Cheng of Institute for Cognitive Systems (ICS), Technical University of Munich (TUM) for allowing me to be a member of his laboratory for a year during PhD course and for providing me insightful comments. I would also like to thank Dr. Jun Nakanishi for his valuable comments and suggestions on my research.

I would like to thank Prof. Hiroshi G. Okuno for his significantly important comments and suggestions in Graduate Program for Embodiment Informatics, Waseda University Program for Leading Graduate Schools. I am thankful to the other leading program member for their comments and enjoyable day.

I would like to thank Dr. Hiroaki Arie, Dr. Kuniaki Noda, Dr. Yuki Suga, and Dr. Shingo Murata for their valuable comments on my research. I would also like to thank the graduate and undergraduate students belonging to “tool-body assimilation” group, a joint research group between Sugano Laboratory and Ogata Laboratory, especially, Mr. Hadi Tjandra, Mr. Yang Pin Chu, Mr. Hiroki Yamada, and Mr. Kitae Kim from Sugano Laboratory; and Mr. Kanata Suzuki, Mr. Itsumi Shimokuri, Mr. Masumi Shinko from Ogata Laboratory with whom I have conducted really exciting research in Waseda University. The everyday discussion on their research contents and chats with them have made my life as a doctoral student all the more enjoyable. I am thankful to the other Sugano Laboratory and Ogata Laboratory members for their comments their encouragement, and their interest in my research.

I would like to thank the following secretaries in Waseda University: Ms. Kyoko Arai and Ms. Yoko Ono of Sugano Laboratory, Ms. Naomi Nakata and Ms. Junko Inaniwa of Ogata Laboratory, Ms. Ilona Nar-Witte, Ms. Wibke Borngesser, Ms. Anja Delanoff, Ms. Brigitte Rosenlehner of ICS. I would also like to thanks Ms. Katharina Stadler who supported me to conduct research in ICS. I would also like to thank Ms. Atsuko Tsuruya, Ms. Madoka Kuniyasu, Ms. Utako Yamada, and Ms. Masumi Toyofuku who supported me to spend time in Graduate Program for Embodiment Informatics, Waseda University Program for Leading Graduate Schools.

Special thanks to my friends and family for their ongoing support and encouragement, especially, to my parents who have allowed me to do whatever I want to do from childhood up to now.

Tokyo, 8th, February, 2017

Kuniyuki Takahashi

Contents

1	Introduction	1
1.1	Background	1
1.2	Artificial Intelligence for Robotics	2
1.2.1	Symbolic Artificial Intelligence	2
1.2.2	Embodied Artificial Intelligence	3
1.3	Overview of My Approaches: Acquisition and Extension of the Robot Body Model Using Motor Babbling through Deep Learning	5
1.3.1	Acquisition of the Robot Body Model Using Motor Babbling	8
1.3.2	Extension of the Robot Body with Tool-body Assimilation	13
1.4	Research Objective	17
1.5	Thesis Organization	18
2	Acquisition of the Robot Body Model Using Motor Babbling	19
2.1	Introduction	19
2.2	Dynamics Motion Learning for a Flexible-joint Robot with Pre-training and Fine-tuning	20
2.2.1	Pre-training of Body Dynamics with Active-passive Motor Bab- bling	22
2.2.2	Fine-tuning of Body Dynamics for Target Tasks with Addi- tional Learning	25
2.3	Numerical Simulation Results	27
2.3.1	Simulated Robot Model	27
2.3.2	Procedure for Motor Babbling with Simulated Robot Model	29

2.3.3	Target Tasks for Simulated Robot	30
2.3.4	Simulation Results	32
2.4	Hardware Experiments with PR2 Robot	38
2.4.1	Experimental Platform: PR2	38
2.4.2	Procedure for Motor Babbling with PR2 Robot	38
2.4.3	Experimental Results with PR2 Robot	39
2.5	Discussion about Motor Babbling	43
2.5.1	Exploratory Motor Babbling Using Variance Prediction from RNN	43
2.5.2	Numerical Simulation Results	46
2.5.3	Experimental Results and Discussion	49
2.6	Summary	51
3	Extension of the Robot Body Model with Tool-body Assimilation	53
3.1	Introduction	53
3.2	Tool-Body Assimilation Model Considering Grasping	54
3.2.1	Acquisition of Body Model through Motor Babbling	57
3.2.2	Body Model Modulation Using Extra-context Nodes	59
3.2.3	Motion Generation from the Initial State and Target Image	61
3.3	Experimental Setup	62
3.3.1	Simulated Robot Model	62
3.3.2	Evaluation Task	62
3.3.3	Procedure of Motor Babbling	62
3.4	Numerical Simulation Results	65
3.4.1	Grasping Positions of Tools	69
3.4.2	Representation of Grasping State on Initial Values of Extra- Context (C_e) Nodes of Body Modulation	73
3.4.3	Representation of Body Modulation on Slow-Context (C_s) Nodes of Body Model	75
3.4.4	Representation of Tool Function on Initial Values of Extra- Context (C_e) Nodes of Body Modulation	75

3.4.5	Motion Generation	81
3.5	Discussion	81
3.5.1	Comparison with Self-organizing Map	81
3.5.2	Comparison with Tool-body Assimilation Model Using PB nodes	87
3.6	Summary	87
4	Conclusion	91
4.1	Overall Summary of the Current Research	91
4.2	Future Works	93
	Bibliography	95
	Relevant Publications	109
	Other Publications	115
	What is <i>Embodiment Informatics</i>?	119

List of Figures

1.1	History of AI and my target	2
1.2	Body model	6
1.3	Overview of approaches	7
1.4	Past study about embodied interactions with the environment	9
1.5	Comparison of tool-use and tool-body assimilation models	14
1.6	Thesis organization	18
2.1	The proposed learning framework for acquiring body dynamics in two phases	21
2.2	Sequence of motor babbling consisting of active and passive motions	25
2.3	Dynamics representation with MTRNN	26
2.4	Simulated robot model	28
2.5	Tasks performed by the simulated robot model on the OpenHRP3 platform	32
2.6	Simulation result of crank-turning and door-opening/closing tasks	35
2.7	Using a small-size crank diameter to evaluate the generalization ability of the learned movement	36
2.8	Simulation result for the evaluation of the generalization ability of the learned movement in the crank-turning task	37
2.9	Experimental result with PR2 of the drawer-opening/closing task	41
2.10	Snapshots showing task execution of drawer-opening/closing with PR2	42
2.11	Exploratory motor babbling	47

2.12	Learning cycles for completing the task of crank-turning and door-opening/closing	50
2.13	Generated motions of crank-turning and door-opening/closing	51
3.1	Tool-body assimilation model	55
3.2	Learning process of tool-body assimilation	56
3.3	Tools used in experiment	63
3.4	Initial position of the arm and object, and first, second, and third sets of key poses	66
3.5	Representation of grasping position in PCA	70
3.6	Relationship between the object position and grasping position of tool	72
3.7	Representation of grasping state in PCA	74
3.8	Representation of body modulation by body modulation module in PCA of time steps	76
3.9	Representation of tool function for pulling and swinging in PCA	78
3.10	Success rate of each tool for pulling and swinging	79
3.11	Representation of all tool functions in PCA of the center of gravity	80
3.12	Success rates of manipulating objects	82
3.13	Original images and recovered images using the trained DNN	83
3.14	Reference vector of SOM	85
3.15	Reference vector of SOM (bare hand and T-shaped tool only)	86
3.16	Original images and recovered images using the trained DNN (with PB)	88
3.17	Training error for acquisition of body model	89

List of Tables

1.1	Past study about tool-body assimilation	16
2.1	Denavit-Hartenberg parameters and joint range of motion of the simulated robot model.	28
2.2	Parameter of the simulated robot model	29
2.3	Design of MTRNN	30
2.4	Design of S-MTRNN	48
3.1	Design of DNN	65
3.2	Design of MTRNN	67
3.3	Representation of main components of PCA	68

Mathematical Symbol list

IO	Input-output nodes
C_f	Fast-context nodes
C_s	Slow-context nodes
$u_i(t)$	The internal value of the i th neuron at step t
τ_i	The time constant of the i th neuron
I_{IO}	The set of indices for the input-output nodes
I_{C_f}	The set of indices for the fast-context nodes
I_{C_s}	The set of indices for the slow-context nodes
$x_j(t)$	The input value of the i th neuron from the j th neuron at step t
$C_j(t)$	The activate state of the i th context neuron at step t
ω_{ij}	The weight value of the i th neuron representing the connection from the j th neuron
$y_i(t)$	The output obtained by substituting the internal value at step t
E	The squared error
$T_i(t)$	The teaching signal for the i th neuron at step t
$Cs_i(0)$	The initial value of Cs_i of the i th neuron
k	The joint stiffness
d	The viscous damping
c	Coulomb friction coefficients
θ	The joint angle
$\dot{\theta}$	The joint velocity
θ_d	The desired joint angle
$\dot{\theta}_d$	The desired joint velocity

P_{gain}	P gain
D_{gain}	D gain
p	The probability density function
$v_i(t)$	The variance state of the i th neuron at step t
L_{out}	The likelihood function
N_r	The number of random motor babbling motions
N_e	The number of exploratory motor babbling motions with added noise
D_n	The number of n -th layer neurons
h_n	The output of the n -th layer
f	The activating function
W	The weight matrix
b	The bias
\hat{p}	The search vectors
H	The Hessian matrix
λ	The damping parameter
I	The unit matrix
C_e	Extra-context nodes
$C_e(0)$	The initial value of C_e

Chapter 1

Introduction

1.1 Background

Today, intelligent robots are expected to play an important role in supporting humans in their daily activities. Conventional robots such as industrial robots are primarily designed to perform designated tasks with high speed and precision and would require extra modifications to perform other tasks. In addition, these robots are typically applicable only in highly controlled environments that have no or negligible uncertainty and are free of human movement. Recently, there has been a growing interest in the development of robots working closely with humans, e.g., for the purpose of supporting daily living [1–3], cooperatively working with humans [4], assisting with tasks on the International Space Station [5] and industry [6], and performing disaster relief [7] activities. In real environments, where humans exist, it is essential that robots would be required to adapt to dynamic and uncertain environments and cope with new tasks and situations in a short duration. Robot with tool use skills would be particularly useful in human society, as this would enable the robot to expand its capabilities in performing tasks. To perform various tasks, robots should have complex structure such as large numbers of applicable sensors and degrees of freedom (DOFs). Designing robot models manually has become difficult because of the above reasons. To address this challenge, I developed a model in which the robot body and environment is based on experiences the robot acquires through physical (embodied)

interactions as opposed to models in which the robot body and environment is pre-designed; that is, the robot learns sensorimotor relationships from experience using machine learning, particularly deep learning.

The focus of this approach is *embodiment*. In this thesis, I focus specifically on the *body model* acquired from interaction with the environment through embodiment.

1.2 Artificial Intelligence for Robotics

Enabling robots to act intelligently is usually referred to as *artificial intelligence*. The term *Artificial Intelligence (AI)* was first used by John McCarthy at the 1956 Dartmouth Artificial Intelligence conference [8]. Up until today, many researchers have been fascinated with AI (Figure 1.1).

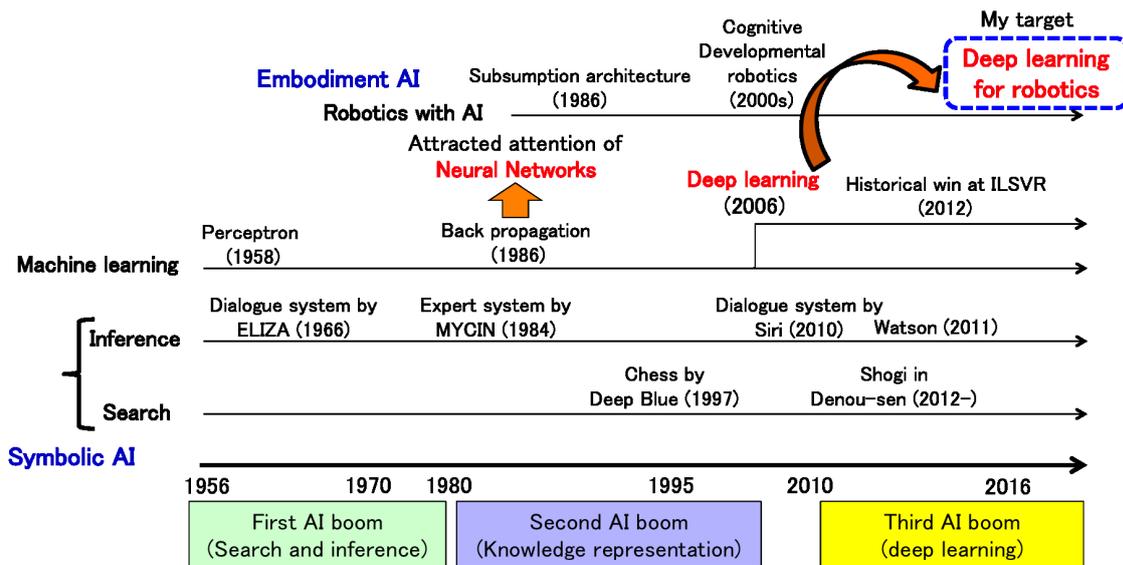


Figure 1.1: History of AI and my target

1.2.1 Symbolic Artificial Intelligence

Symbolic AI, also known as the *classical AI approach* or *good old-fashioned AI (GOFAI)* [9], is based on the manipulation of symbols, an assumption defined as the

physical symbol systems hypothesis [10]. Symbolic AI was the dominant focus of AI research from the 1950s until the 1980s. During this period, the concept of search and inference was developed in applications such as chess by Deep Blue [11]. Furthermore, a diagnosis system called ELIZA [12] was developed, as was an expert system called MYCIN [13]. These approaches are only valid if all possible situations are described. However, it is difficult to describe all possible situations in the real world. One of the challenges of symbolic AI is the frame problem [14], which implies that AI has a finite ability to process information and cannot handle all of the infinite possibilities of the real world. Another challenge is the symbol grounding problem [15], that is, the problem of how symbols in the symbol system can be linked with the meaning of the real world. These problems are occurred because humans try to describe the symbol.

1.2.2 Embodied Artificial Intelligence

Brooks proposed a subsumption architecture [16] in which the system and environment are connected using a connection between the sensor and motor without a description of the symbols. He insisted that the system should not have a designed complex internal representation and external (environment) model but, instead, should use the interaction system with the environment through the body [17, 18]. This approach is referred to as *embodied AI*, also known as the *behavior-based approach* [19]. At the time when subsumption architecture was originally proposed, relatively simple insect-type robots performed actions without complicated processing between sensors and motors [16]. A more complex system, a humanoid robot, Cog, has also been verified [20, 21]. Considering the relationship between the body and intelligence, researchers have frequently discussed how complex the body must be [22–26]. It is hypothesized that for complex intelligence, in particular for human-like intelligence, a human-like body is necessary for the robot to interact with the environment [22]. In cognitive developmental robotics (CDR), robots with human-like bodies, also known as humanoids, have been popular since the 2000s [27]. CDRs require a more constructive approach to understand the cognitive developmental processes and how to

achieve them through embodiment.

To consider the complex relationship between the sensor and motor, machine-learning, particularly by using neural networks, has been proven effective [28,29]. A simple perceptron was developed in 1958 as a linear separator [30]. In 1969, Marvin Minsky and Seymour Papert showed that it was impossible for a simple perceptron to learn an XOR function [31]. In 1986, nonlinear separation became possible through the development of backpropagation [32]. Training multi-layer neural networks for multi-dimensional data is difficult because of the vanishing gradient problem, which is the phenomenon where, as the number of layers is increased, the gradient of the loss function becomes dramatically closer to zero before the input layer. In 2006, Hinton et al. [33] proposed a pre-training and fine-tuning approach for deep learning to solve the problem. Deep learning has desirable characteristics, such as the ability to obtain individual features from data. The term *deep learning* refers to the use of deep neural networks (DNNs) and recurrent neural networks (RNNs). DNNs are feed-forward neural networks, whereas RNNs are recurrent connections. In past AI systems, humans designed features via observation results, such as scale invariant feature transform (SIFT) [34]. They designed models to simulate various phenomena, thereby causing the results to depend heavily on human efforts. As applications of deep learning, the performances of image, speech, and natural language recognition have been dramatically improving [35,36]. In ILSVR2012, the results of image recognition by deep learning were overwhelmingly superior to other methods [35]. Recently, image recognition in robots has exceeded the human recognition rate [37].

As an application of deep learning, I have focused on robotics since 2013. Using deep learning, the design of robots within the context of their environment is developed using sensorimotor relationships from experiences the robots acquire through physical interaction, as opposed to being pre-designed by humans. This approach enables the application of this architecture to other robots and tasks. Even though the current machine-learning approach requires numerous training samples and is inferior with respect to the accuracy of motion, as compared to a model-based approach and a combination of model-based and machine-learning approaches, I strongly believe it will be increasingly necessary for future developments in robotics. In the

robot-environment model, using deep learning, the robots can discern the relationship between the raw sensor information and motor and use that relationship to generate motion [28, 29].

1.3 Overview of My Approaches: Acquisition and Extension of the Robot Body Model Using Motor Babbling through Deep Learning

If there are redundant DOFs, there are various possibilities for the position and posture of the end-effector. By considering embodiment, even though there are redundant multi-DOFs, it is possible to generate motion because of the restricted redundant DOFs by embodiment restriction such as self-body collision, posture of the initial state, and range limitation of the joint angles, angular velocities, and torque [38]. Furthermore, during search activities of the body, it was confirmed that more adaptive and effective motions are searched for, discovered, and chosen [39].

In related research studies investigating embodiment with robots, robots with a relatively small number of DOFs were used, and the robots were only trained to execute specific motions for corresponding tasks; that is, the robots had sensorimotor only of the task motions. This means that the robots did not know how to move their bodies or how to move in unknown situations. To overcome this challenge, it is necessary to consider the robot body model as a whole body.

When considering a robot body, body schema is an important concept used to explain the body model. Body schema involves mapping between the motor and the sensors located on the body and includes information about the size and shape of the body [40]. In addition, body schema is not fixed after acquisition but is, instead, a plastic model. In particular, when humans are proficient in using a tool, they behave as if their bodies extend to the edge of the tool that they use. The phenomenon is referred to as *tool-body assimilation* [41, 42]. Owing to body schema, humans have the capability of recognizing the tool tip posture, position, and motion because their body schema is plastic with respect to their body and the tool. Hence, body schema

can connect the body and tool easily; that is, it is possible to describe the tool using the model to extend the body schema. When a body schema is considered from an engineering perspective, it is a multi-modal association of motion and sensors. In my research, I added the concept of the forward/inverse model (In this study, the forward model predicts the next state of the arm-tip position from the current state of the joint angles, and the inverse model predicts the next state of the joint angles from the current state of the arm-tip position) [43] (Figure 1.2). I refer to it as the *body model*, and in particular, I refer to it when considering the dynamics as *body dynamics*.

In my proposed system, first, the robot acquires the body model through deep learning. Then, (1) the robot with flexible joints learns dynamic motions using the acquired body model, and (2) the robot learns tool use via the tool-body assimilation model with the extension of the acquired body model (Figure 1.3).

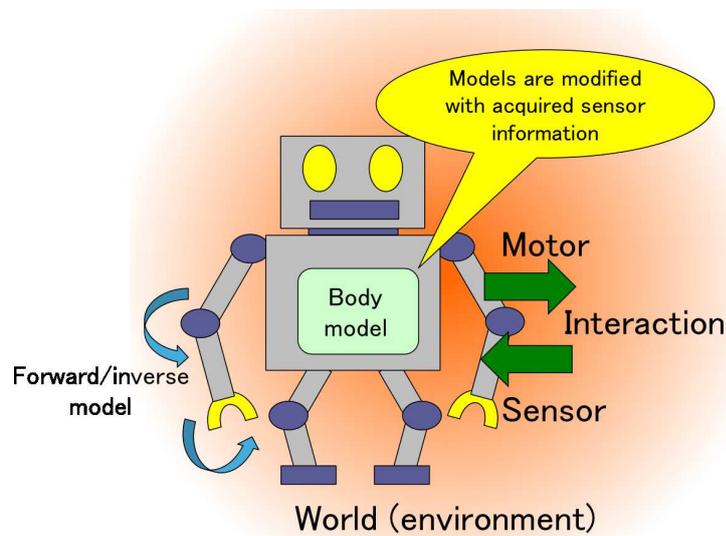


Figure 1.2: Body model based on sensorimotor relationship and forward/inverse model.

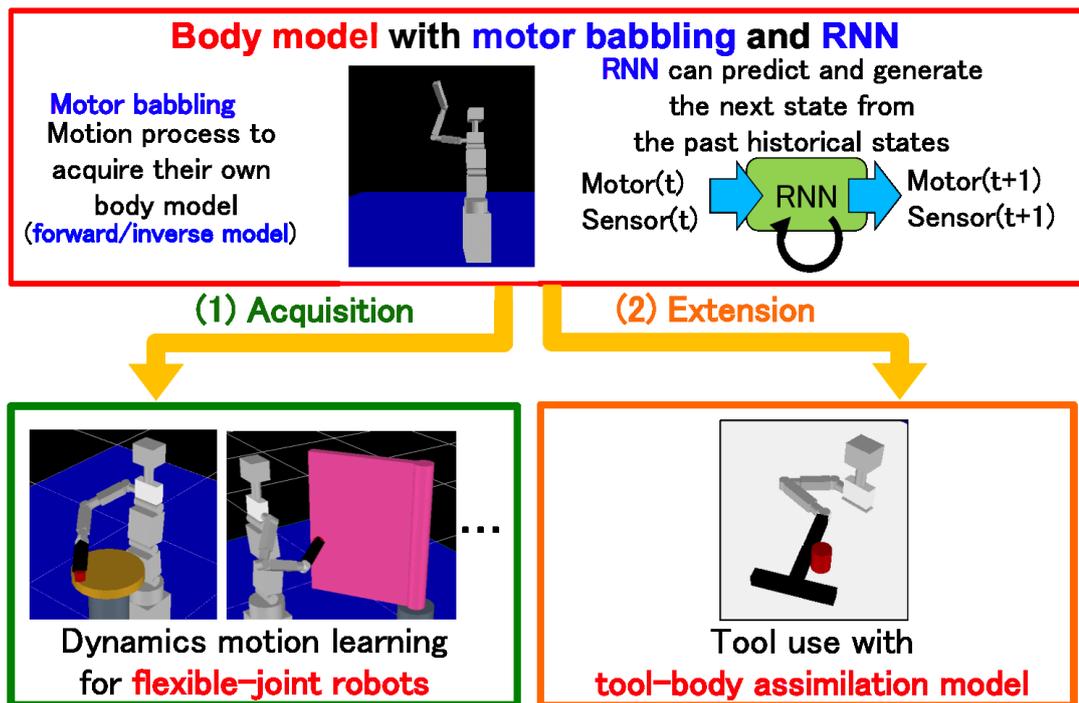


Figure 1.3: Overview of approaches: *acquisition* and *extension* of the robot body model using motor babbling through deep learning. After acquisition of the robot body model, (1) the robot with flexible joints learns dynamic motion using the acquired body model, and (2) the robot learns tool use via the tool-body assimilation model with the extension of the acquired body model.

1.3.1 Acquisition of the Robot Body Model Using Motor Babbling

Various humanoid robots have been developed to support humans' tasks. A human symbiotic robot, TWENDY-ONE, which was developed at Waseda University, demonstrated a number of assistance tasks of human daily activities such as support for standing up from a bed, moving to a wheelchair from a bed, cooking with kitchen utensils, and navigating in a congested environment [3]. NASA's humanoid robot, Robotanaut 2, is designed to assist astronauts in performing tasks, and has the capability of using various tools in space [5]. In a disaster-response situation, the Carnegie Mellon University's (CMU) humanoid robot CHIMP [7] performed various tasks such as removing obstacles, opening doors, turning cranks, and using tools in the DARPA Robotics Challenge [44]. Challenges in these tasks include efficient motion generation and control, and safe interaction with humans and the workspace environment. From this perspective, flexible-joint robots have recently attracted increasing attention with the following potential benefits:

- *Adaptability*: passive adaptation to environmental changes and safe human-robot interaction with mechanical compliance [3, 45–47]
- *Dynamic motion*: movement generation and control exploiting intrinsic dynamics [48–52]

Adaptability can be useful for the safe interaction between a robot and the environment. For example, adaptability with mechanical compliance can be beneficial when there are unmodeled environmental changes for tasks having constrained movement requiring interactions with the environment, such as a door-opening/closing task. Unlike conventional high-gain controlled stiff-joint (non-flexible) robots, it is possible to reduce the risk of damage to the robot or the environment by avoiding excessive interactive forces in the presence of uncertainties. An additional benefit of joint flexibility is that it can absorb the impact forces in the case of collisions between the robot and the environment or a human. In particular, the humanoid robot, TWENDY-ONE, is equipped with passive joints considering safe human-robot

interaction [3]. In terms of the *dynamic motion*, mechanical joint flexibility can also be useful to achieve dynamic tasks, e.g., periodic movements or repetitive movements by exploiting intrinsic dynamics with reduced control effort [21]. In spite of these benefits, it is difficult to control flexible-joint robots with a large number of degrees of freedom (DOFs) because of the increasingly complex dynamics [45]. To address this problem, several approaches have been previously proposed such as model-based control [45, 46, 53], model-free machine learning [47, 54–57], and dynamical systems approach (oscillators [21] and attractors [58]) (Figure 1.4).

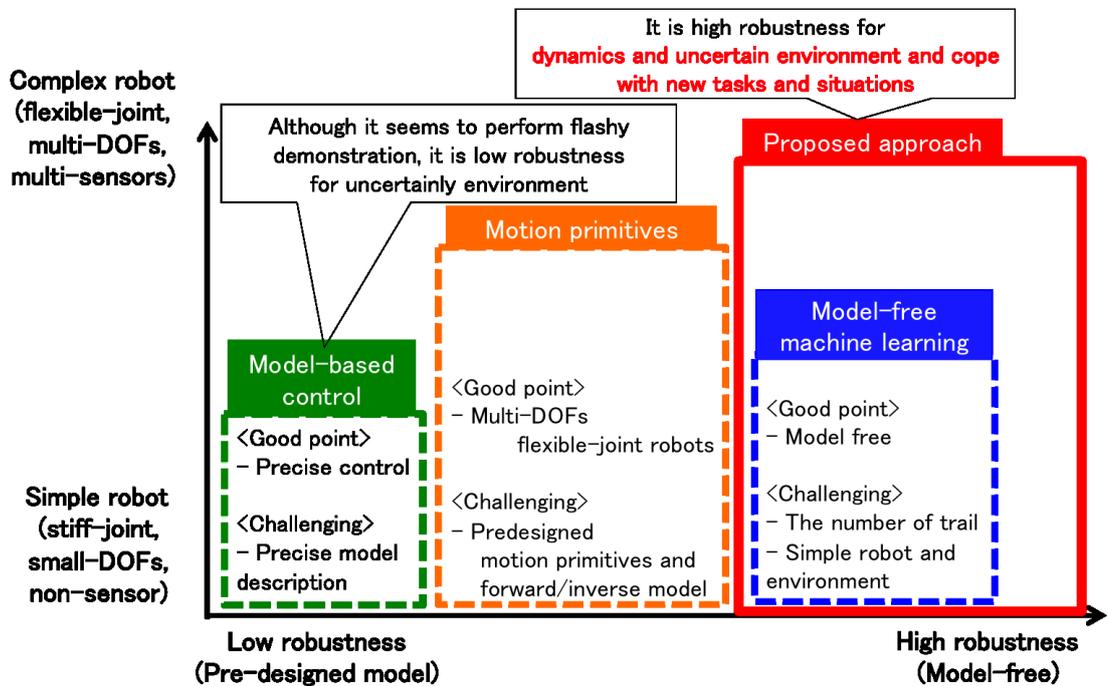


Figure 1.4: Past study about embodied interactions with the environment. Several approaches have been proposed such as model-based control, motion-primitives, and model-free machine learning. My proposed approach would be robustness for dynamics and uncertain environment thanks to acquire body model from experiences by robots itself through embodied interaction instead of humans’ pre-designed robot and environment model.

With a model-based control approach, it was demonstrated that a flexible joint

robot was able to achieve dynamic motion with passive adaptation to the environment, and was able to generate precise motion when drawing a line with a pen having contacts between its arm and the environment [45,46]. The focus of these studies was to avoid complex control by reducing the number of effective DOFs using contacts with the environment, and to exploit joint flexibility and human-like soft skin. However, when tasks with increased numbers of DOFs were considered, direct application of the simplified approaches presented in [45,46] may not be suitable. In order to fully exploit the benefits of joint compliance, a model-based optimal control technique was employed for a ball-throwing task on a real robot with variable stiffness actuation [53]. While this study used a highly complex variable-stiffness robot, the task was limited within a planar two-DOF movement of the arm. Generally speaking, application of model-based approaches would be very challenging on tasks with an increased number of DOFs and considering interactions between the robot and the environment because of the requirement for a precise model description of the complex dynamics of the entire system. Therefore, application of such model-based approaches on a complex robot may become significantly difficult in a highly dynamic and uncertain environment particularly including interactions with humans.

When the modeling of the robot and the environment is difficult, model-free machine-learning can be an attractive approach for motion generation with adaptation to dynamic and uncertain environments. In [47], reinforcement learning was used to achieve the task of sliding a switch on a highly complex tendon-driven robot hand by finding a control policy through interaction between the environment and the robot. However, the task was confined to the planar four-DOF movement of a single finger within a small range. One of the difficulties of reinforcement learning is that the number of trials would grow with the increase of the dimensionality of the state space [59].

As I have seen above, model-based control and model-free machine-learning approaches have been shown to be effective for systems with a relatively small number of DOFs in a simplified environment. For motion generation considering more complex dynamics and environments, a dynamical systems approaches have been explored.

In [21, 60], neural oscillators were used to achieve a variety of dynamic and complex tasks such as drumming, hitting, throwing and crank turning exploiting natural dynamics of the robot arm with joint compliance. In [58], as motion primitives, attractors and oscillators were designed to have desirable convergence properties to the specific point or limit cycle when the trajectories are perturbed. Typically, these motion primitives have a single-point attractor or a single-limit cycle attractor corresponding to a given task, e.g., the motion primitives proposed in [58] are designed to have a unique attractor with global convergence. Thus, it is difficult to consider trajectories having branch structures¹ (e.g., multiple goals or multiple objectives). Furthermore, in previous studies, the structure or representation of the dynamical systems or motion primitives was typically pre-designed by a user. For example, it is necessary to specify whether the output of the motion primitives represent joint-space trajectories or task-space trajectories with the knowledge of the properties of the system (e.g., forward/inverse model [43]) and the task. Furthermore, in the presence of joint flexibility, it is not straightforward to design a controller to achieve the desired end-effector movement, especially, in complicated situations considering interactions with the environment. The effectiveness of the neural oscillator based approach in [21, 60] was indeed demonstrated in achieving constrained dynamic tasks such as a crank turning with a flexible joint robot arm. However, it was necessary to select the network parameters and oscillator arrangement by the user.

In this paper, I propose a machine learning framework to acquire motion primitives and the forward/inverse model of the system based on a deep learning approach in training RNNs. Specifically, I call the forward/inverse model *body model*, and in particular, I refer to the body model considering dynamics as *body dynamics*. In order to acquire body dynamics, I employ motor babbling, which is the self-exploring movements of the early days of infants to acquire their own body dynamics. Motor babbling has recently attracted much attention for motion generation in the field of robotics. The effectiveness of motor babbling has been demonstrated in robotic applications such as posture prediction [62], hand-eye coordination [63, 64], reaching to a static target [65, 66], and drawing tasks [67–69]. In order to acquire body dynamics,

¹An example of a sequence of actions with branching trajectories is discussed in [61].

RNNs have been employed [61, 70] by exploiting their desirable characteristics being able to predict the next state from the past history of neurons' states by preserving the internal state with a recurrent connection to itself [71]. Effectiveness of the use of an RNN in learning the sensorimotor relationship has been demonstrated, i.e, the forward/inverse model, considering spatio-temporal aspects of the motions from the input and output signals [70]. Furthermore, in [61], it has been demonstrated that RNNs can learn movements with multiple objectives composed of trajectories having branch structures exploiting such characteristics of an RNN. However, the focus of these previous studies was to generate simple movements on a stiff-joint robot with a relatively small number of DOFs.

In order to train an RNN with complex tasks on a flexible joint robot with a large number of DOFs, I am motivated by the successful and effective training strategy for DNNs consisting of *pre-training* and *fine-tuning* [33, 72]. In the pre-training phase, parameters of each layer is initialized in order to obtain appropriate values. Then, in the fine-tuning phase, the whole network is trained with all layers connected to efficiently learn the target data using the result of pre-training. This method is shown to be effective in solving issues associated with neural networks such as overfitting, slowness of convergence, and difficulty in learning and generalizing high dimensional data. As a result, this two-phase strategy has been demonstrated to yield much improved learning performance in DNNs [33, 72]. In this paper, I apply this deep learning based two-phase approach to learning movements on a robot with complex dynamics with an RNN consisting of pre-training of body dynamics with motor babbling and fine-tuning with additional learning of body dynamics for a task movement.

In this paper, I propose a two-phase leaning strategy for an RNN to acquire motion primitives or body dynamics by performing motor babbling with pre-training and fine-tuning of body dynamics for a given complex task motion on a flexible-joint robot. I see an analogy between my proposed two-phase strategy and the learning strategy for DNNs in that in the pre-training phase, appropriate initial parameters of the network is learned, and then in the fine-tuning phase, efficient learning is achieved with adjustment and reuse of the learned results in the pre-training phase. I demonstrate the effectiveness of the proposed strategy in more complex settings, presenting

evaluations on a simulated flexible-joint robot model and in hardware implementation on an actual PR2 robot both of which a seven-DOF redundant arm. I consider tasks that involve dynamic interactions and constraints of the movement with the environment such as crank-turning, door-opening/closing, and drawer-opening/closing in a three-dimensional (3D) space. I show the reduction in the number of training iterations to perform the target task with motor babbling and generalization capabilities for untrained cases with the online generation of the movement. In addition, I discuss the relationship between the degree of motor babbling and the success rate of the target tasks.

1.3.2 Extension of the Robot Body with Tool-body Assimilation

By using tools, humans are capable of expanding their field of work and actions that otherwise would be impossible with bare hands. Current robots, such as industrial robots, are mainly designed to perform specialized tasks, and would require extra modifications when the need to conduct other tasks arises. However, if robots are capable of using tools as humans do, this could enable robots to adapt to various activities.

To realize tool use, humanoid robots are useful for handling humans' tools. One of the approaches through which robots use tools is model-based control. Humans provide preset knowledge to the robot about the setup of the scenario and usage of tools, and using modeling approaches, it is possible for robots to achieve high performance for tasks such as making pancakes [73] and sandwiches [74] with tools, drawing with a multifingered hand [46], putting a file into a folder, and serving a fried egg with a spatula [75]. However, a majority of these methods require humans to code most of the information about the situation and environment. Thus, they are applicable only in a highly controlled environments that have no or negligible uncertainty.

When the modeling of the robot and the environment is difficult, model-free

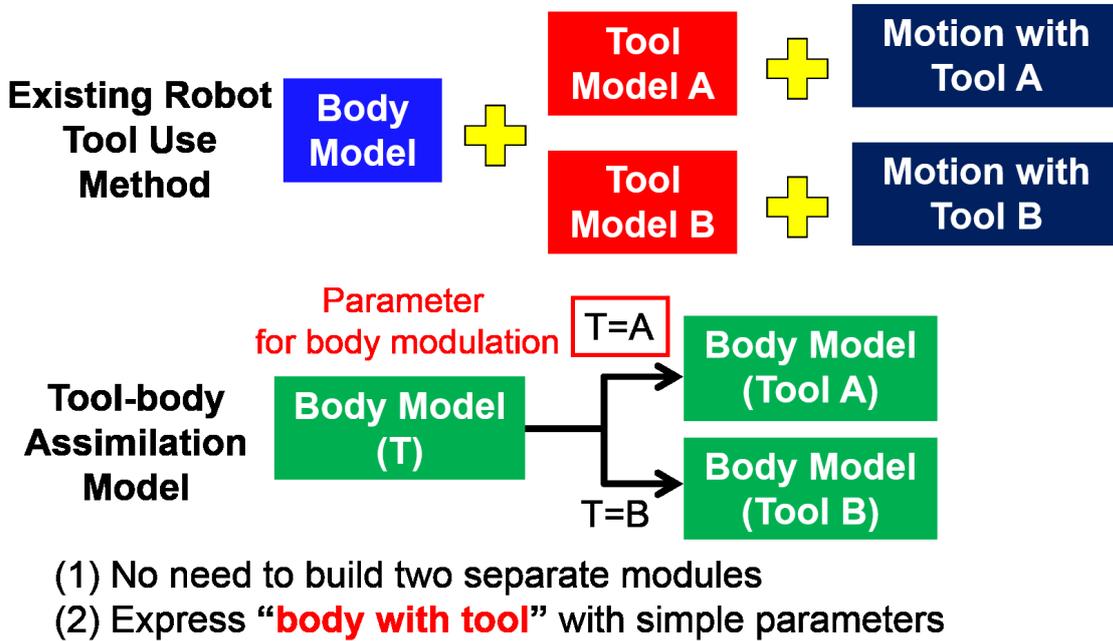


Figure 1.5: Comparison of tool-use and tool-body assimilation models. In the existing tool-use model, the body and tool models are separated. Therefore, the motions during tool use are expressed by each tool. Contrastively, in a tool-body assimilation model, the tool model is expressed as a body with a tool. Therefore, the motions during tool use are expressed by the body model, and the robot effectively learns the tool functions.

machine-learning can be an attractive approach for motion generation. Previous research about machine learning typically involves combining it with a model-based approach. In [76], a robot autonomously learns tool functions through machine learning. A robot is capable of acquiring a tool function using the corresponding tool. However, this research is still only able to use a trained tool because of the necessity of pre-designed models. In addition, the robot used in [76] can only move with pre-designed motions. This is because the body and tool models are separated. Therefore, the motions during tool use are expressed by each tool (existing robot tool-use method shown in Figure 1.5).

To address these issues, I adopted *tool-body assimilation*, which is a recent finding in the field of cognitive science, with the machine-learning (learning based) approach.

Tool-body assimilation has recently gained attention in the field of robotics [29, 76–79]. Tool-body assimilation is being applied in learning-based approaches in which a robot’s body model is defined as a plastic model, which can be deformed based on the tool being used (Figure 1.5). This implies that the robot would only need to gain its body model once to be modulated and reused again during tool use. In other words, through tool-body assimilation it is possible to use tools with only one model, i.e., a body model, to generate motion. In addition, the body modulation parameter does not represent the complicated methods of using a tool; it only represents the method of modulating the original body model. Because the tool is treated as a part of the body, I expect the robot to use tools based on the experience of moving the body. This implies that the robot would learn to use the tool easily.

In the early days of infants, they tend to predict tool functions using dynamic touch [80]. Dynamic touch refers to the movement of a body to acquire the characteristics of an object by moving the object [81]. In [77], a method was developed to have a robot determine tool inertia parameters using dynamic touch as the body modulation parameter. Then, the robot used this information to correctly use the tool to pull an object toward itself. However, pre-designed tool features, such as inertial parameters, make it difficult to use this method for typical tools. In [78], object manipulation was realized using tools through a neural network without pre-designed tool features. The model was capable of using unknown tools and performing object manipulation tasks with dynamic touch. The experimental results of Michaels et al. imply that humans estimates the tool function based on the shape of tool from vision after they has experiences of tool use [82]. In [79], which is a study inspired by [78], robot tool-use was realized through tool-body assimilation, with image data captured using the robot’s camera as the body modulation parameter. This allows the robot to adapt and perform pulling tasks using tools without dynamic touch in advanced motion. However, the motion sets used to train the model were limited, and the self-organizing map used for extracting image features did not have strong generalization abilities for unknown tools (Table 1.1).

To address the limitation of the motions, motor babbling has been employed.

Table 1.1: Past study about tool-body assimilation

Previous study	Motion type	Recognition of tool	Grasping decision
A. Stoychev [76]	Pre-designed motion	Tool's color	Fixed grasping position
K. Nabeshima et al. [77]	Pre-designed motion	Dynamic touch with pre-designed model	Pre-attached to robot hand
S. Nishide et al. [78]	Pre-designed motion	Dynamic touch with neuron dynamical system	Pre-attached to robot hand
Y. Yamaguchi et al. [79]	Pre-designed motion	Raw image data with SOM (low generalization ability)	Pre-attached to robot hand

Using motor babbling, infants learn about the relationship between their body movements and changes in sensorimotor information as body schema [40]. It is believed that the body has action-limiting influence on human cognition and tool use [83]. Without bodily constraints, there will be an infinite number of body postures during tool use. The problem of the brain selecting a particular movement is commonly referred to as the motor equivalence problem [84]. However, in most cases, humans tend to use tools in a similar manner. When using tools, several poses place burden on the body to achieve the motion; thus, humans try to avoid these poses. These poses are learned during human development. Thus, I consider that acquiring the system's forward/inverse model [43] through learning is a promising approach. Therefore, in this research, the body model is used with the assumption that learned constraints affect the motion decision.

To address the limitation of the unknown situation, deep learning such as DNN and RNN has been employed for the generalization performance. DNN allows for extraction of image features from image data without humans' pre-designed features. These image features can be reused for unknown situations. Lastly, the model enables a robot to compute motion using only the initial state and a target image of the generation sequence. The tool-body assimilation model would be improved occasionally.

Several researchers have modeled motor babbling as an attempt to realize robot tool use. Furthermore, tool-body assimilation approach primarily focus on obtaining tool functions, and a robot starts its motions using a tool pre-attached to the robot

or pre-designed grasping position of a tool. This implies that even if the robot was able to understand the tool functions and use the tools accordingly, it would not be able to decide whether and where to grasp the tool. In real-life environments, robots would need to consider the possibilities of tool-grasping positions, and grasp the tool required to achieve the target task.

In this research, I let the robot generate motion by decision whether and where to grasp the tool based on the initial state and a target image. To realize this, motor babbling that initiates motions with the tools not attached to the robot's hand is used. The robot then performs motor babbling by *grasping* and *nongrasping* the tool to learn the robot's body model and tool functions. In this research, *grasping* means that the robot holds the tool to attach the tool to the body and once the robot grasps the tool, whereas *nongrasping* means that the robot treats the tool as object such as push and pull without attach the object to the body even though the robot contacts it. For this purpose, I use deep learning such as DNN and RNN. In this research, the DNN extracts the image features as tool and movement features from the images obtained using the robot's camera. Next, the RNN learns the relationship between the image features, joint angles, and grasping signal data gained during the motor babbling of the body model. Lastly, a body modulation module learns the grasping decision, grasping positions, and tool functions from the grasping actions by comparing the differences between the image features when *grasping* or *nongrasping* the tools.

1.4 Research Objective

The objective of my thesis is to propose a machine learning framework that the robot acquires its body model from the experiences it gains through embodiment without pre-designed robot-environment model to adapt to dynamic and uncertain environments and cope with new tasks and situations in a short duration. Specifically, I focus on the acquisition and extension of the robot body model using motor babbling through deep learning such as RNN and DNN to realize dynamic motion learning for flexible-joint robots and tool use with a tool-body assimilation model.

1.5 Thesis Organization

The rest of the thesis is organized as shown in Figure 1.6. Chapter 2 describes the *acquisition* of the robot body model through motor babbling using deep learning. This chapter proposes a two-phase strategy for robots with flexible joints having multi-DOFs to perform dynamic motion tasks. Chapter 3 describes the *extension* of the robot body model with the tool-body assimilation model considering tool-grasping decision-making. This chapter proposes body model modulation using extra-context nodes. Chapter 4 summarizes the thesis.

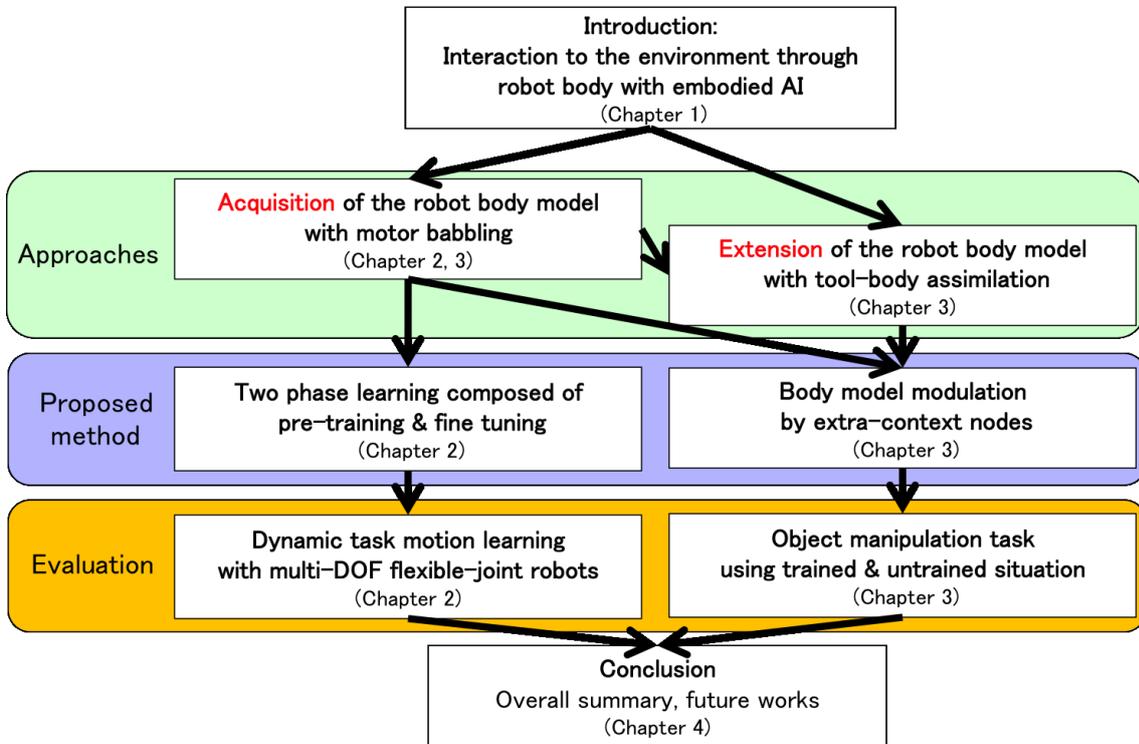


Figure 1.6: Thesis organization

Chapter 2

Acquisition of the Robot Body Model Using Motor Babbling

2.1 Introduction

In this chapter, I propose a learning strategy for robots with flexible joints having multi-degrees of freedom (DOFs) in order to achieve dynamic motion tasks. In spite of there being several potential benefits of flexible-joint robots such as exploitation of intrinsic dynamics and passive adaptation to environmental changes with mechanical compliance, controlling such robots is challenging because of their increasingly complex dynamics. To achieve dynamic movements exploiting such benefits of flexible-joint dynamics, I introduce a two-phase learning framework of the *body dynamics* of the robot using a recurrent neural network (RNN) motivated by a recent deep learning strategy consisting of *pre-training* and *fine-tuning* [33, 72]. This two-phase learning methodology comprises a *pre-training* phase with motor babbling and a *fine-tuning* phase with additional learning of dynamic motion tasks. In the pre-training phase with motor babbling, I consider *active* and *passive* exploratory motions in order to efficiently learn body dynamics. In the fine-tuning phase, the learned body dynamics are adjusted for specific tasks. I demonstrate the effectiveness of the proposed methodology in achieving dynamic tasks involving constrained movement requiring interactions with the environment on a simulated flexible joint robot model as well

as in hardware experiments using a PR2 robot both of which have a seven DOF redundant arm¹. The results illustrate a reduction in the required number of training iterations for task learning as well as generalization capabilities for untrained situations with the learned body dynamics through motor babbling. In addition, I discuss the issues regarding the trade-off between task training iterations and the success rate of task execution. I also discuss exploratory motor babbling used to acquire body dynamic efficiency.

2.2 Dynamics Motion Learning for a Flexible-joint Robot with Pre-training and Fine-tuning

In this section, I present my proposed strategy for dynamic motion learning with flexible-joint robots. Our learning strategy consists of two phases, i.e., the pre-training phase and the fine-tuning phase, as depicted in Figure 2.1. As mentioned above, this procedure is analogous to the successful learning strategy for DNNs.

- In the pre-training phase, the robot acquires its body dynamics with an RNN through motor babbling. The robot performs motor babbling with two types of motions (active and passive motions) to improve the efficiency in the learning process of the body dynamics.
- In the fine-tuning phase, the robot performs additional learning of the movement for specific tasks by adjusting the acquired body dynamics to modify the acquired network in the pre-training phase.

The objective of this strategy is to efficiently learn the desired movements to perform the given tasks with the reduction of training iterations and generalization to untrained situations with the learned body dynamics using motor babbling through an RNN.

¹An accompanied video is available at the following link:
<https://youtu.be/nu54pLZCWgA>

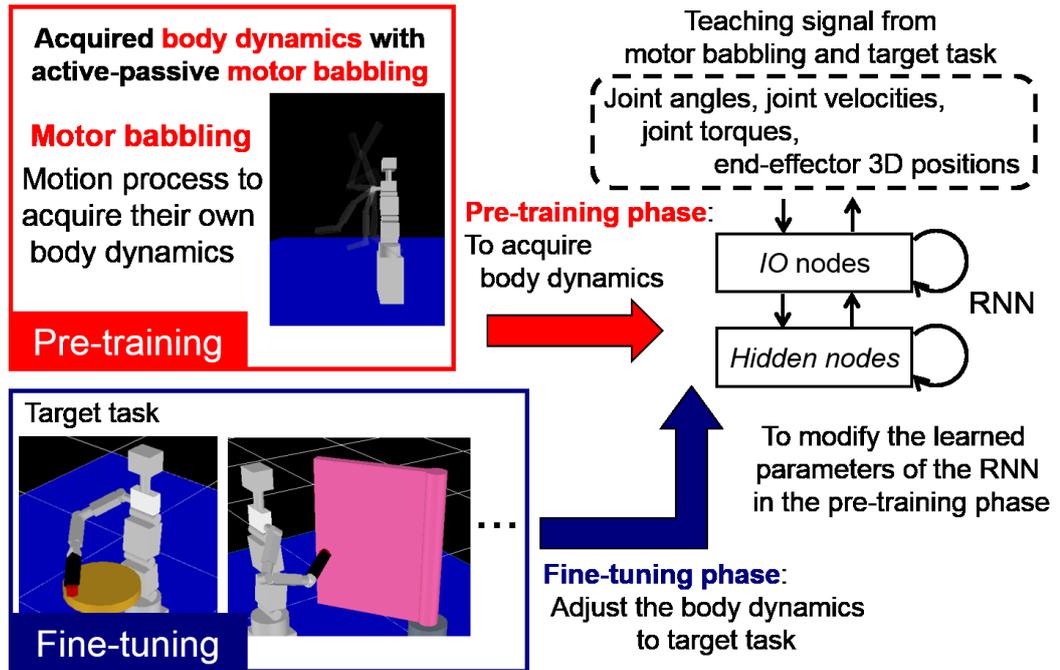


Figure 2.1: The proposed learning framework for acquiring body dynamics in two phases (pre-training and fine-tuning). In the pre-training phase, the robot acquires body dynamics with an RNN through motor babbling. I consider a sequence of active and passive motions to improve the efficiency in the learning process of the body dynamics. Then, in the fine-tuning phase, the robot performs additional learning to adjust acquired body dynamics to the target task. The method is analogous to the successful learning strategy for DNNs consisting of *pre-training* and *fine-tuning*.

2.2.1 Pre-training of Body Dynamics with Active-passive Motor Babbling

In the pre-training phase, the robot performs motor babbling with an RNN to acquire its body dynamics. In motor babbling, I consider a sequence of active and passive motions. As depicted in Figure 2.2, during the course of motor babbling, the type of movement is switched from an active one into a passive one. In the active motion phase, the robot generates joint motions with an active torque input to learn dynamic association between the control input and the robot states. In the passive-motion phase, the torque input is turned off, and the joint moves passively under the effect of the inertia, gravity, and joint friction and stiffness. The purpose of this process is to learn the intrinsic characteristics of such natural dynamics of the flexible-joint robot.

Note that my methodology in this paper is different from explicit parameter-identification approaches in the kinematics and dynamics models as in [85, 86]. In the case of controlling flexible-joint robots having interactions with an unknown environment, learning dynamic parameters of the robot and the environment model is non-trivial because it depends on the pre-designed model structure. In contrast, an RNN learns the sensorimotor relationship through training in a data-driven manner and predicts the next state having the previous states with the learned network parameters.

To acquire body dynamics, I use the multiple timescales RNN (MTRNN), which is a variant of an RNN [61], where MTRNN can predict the next state from the past history of neurons' states, as well as learn multiple sequences of data. The MTRNN is composed of three types of neurons that have different time constants: input-output (IO) nodes, fast-context (C_f) nodes, and slow-context (C_s) nodes (Fig. 2.3). The input-output (IO) nodes are associated with the sensorimotor states and control of the robot. The fast-context (C_f) nodes learn the small segments in the data as primitive movements, whereas the slow-context nodes learn the sequence of the primitives of the data. The MTRNN learns the dynamics of the sequence of the data by combining these nodes. In addition, a specific learned motion sequence can

be generated among the multiple learned patterns by searching the corresponding acquired initial values of the slow-context (C_s) nodes. To train the MTRNN, I use the back-propagation through time (BPTT) algorithm [32]. This algorithm is composed of forward calculations and backward calculations as follows:

In the forward calculation, the output of the neurons is computed. First, the internal value of the i th neuron at step t , $u_i(t)$, is calculated as

$$u_i(t) = \left(1 - \frac{1}{\tau_i}\right)u_i(t-1) + \frac{1}{\tau_i} \left[\sum_{j \in I_{IO}} \omega_{ij}x_j(t-1) + \sum_{j \in I_{C_f}, I_{C_s}} \omega_{ij}C_j(t-1) \right] \quad (2.1)$$

where τ_i is the time constant of the i th neuron, I_{IO} , I_{C_f} , and I_{C_s} are the set of indices for the input-output (IO), fast-context (C_f), and slow-context (C_s) nodes, respectively, $x_j(t)$ is the input value of the i th neuron from the j th neuron, $C_j(t)$ is the activate state of the i th context neuron² from the j th context neuron, ω_{ij} is the weight value of the i th neuron representing the connection from the j th neuron. Then, the output $y_i(t)$ is obtained by substituting the internal value (2.1) into the sigmoid function:

$$y_i(t) = \text{sigmoid}(u_i(t)) = \frac{1}{1 + \exp(-u_i(t))}. \quad (2.2)$$

The next input value to the i th neuron in (2.1) is recursively calculated using the output value in (2.2) of the previous step as

$$x_i(t+1) = y_i(t). \quad (2.3)$$

In this paper, the output $y_i(t)$ corresponding to the joint-angle node is used as a target joint angle of the robot. When the robot generates the motion *offline* to replay the movement of the training data set, the predicted output $y_i(t)$ of the MTRNN is

²Context neurons are referred to as the neurons in the fast-context (C_f) nodes and the slow-context (C_s) nodes.

used as the input $x_i(t + 1)$ in (2.3), i.e., no actual robot states are fed back to the MTRNN as an input. In contrast, when the robot generates motion *online* for the purpose of generalization, the actual robot states (joint angles and the end-effector positions) are partially used as an input $x_i(t + 1)$ to the MTRNN to predict the next state $y_i(t + 1)$.

In the backward calculation, the BPTT algorithm is used to minimize the squared error E as an objective function given by

$$E = \sum_t \sum_{i \in IO} (y_i(t - 1) - T_i(t))^2 \quad (2.4)$$

where $T_i(t)$ is the teaching signal for the i th neuron at step t . In this study, the teaching signal is given as a sequence of robot motions in motor babbling and target tasks. Details of the specific motions for teaching signals will be described in Sections 2.4 and 3.4.

In the BPTT algorithm, the weight from the i th neuron to the j th neuron is updated with the gradient of the training error E as:

$$\omega_{ij} = \omega_{ij} - \alpha \frac{\partial E}{\partial \omega_{ij}}, \quad (2.5)$$

where α is the learning rate. The initial value of C_{s_i} of the i th neuron in (2.1), $C_{s_i}(0)$, is also updated by the BPTT algorithm:

$$C_{s_i}(0) = C_{s_i}(0) - \alpha \frac{\partial E}{\partial C_{s_i}(0)}. \quad (2.6)$$

After the network training, $C_{s_i}(0)$ represents the association to each learned motion sequence such that the learned motion sequence can be recovered by substituting the obtained $C_{s_i}(0)$ values into the MTRNN.

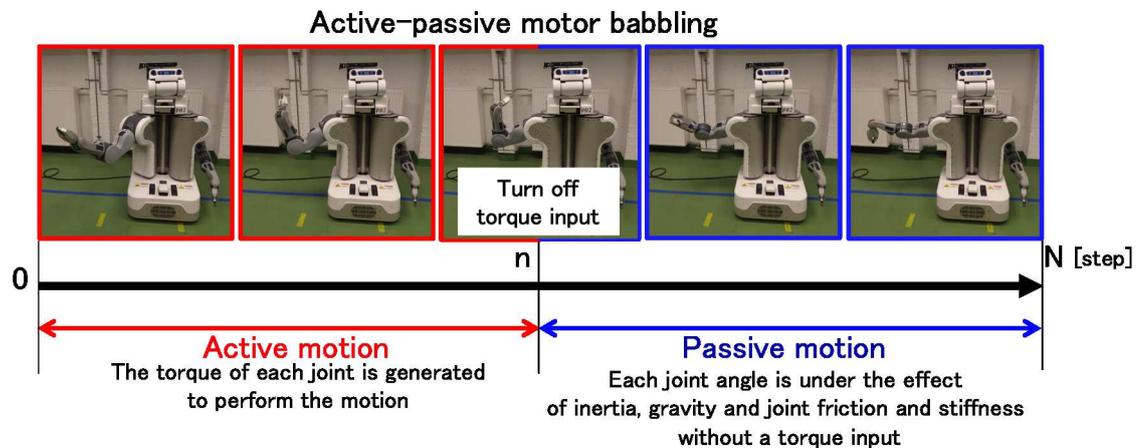


Figure 2.2: Sequence of motor babbling consisting of active and passive motions. In the active motion phase, the robot generates the torque of each joint to perform motor babbling. In the passive motion phase, the torque input is turned off and the resultant passive movement of the robot is exploited. The phase of motor babbling is switched from the active mode to the passive mode during the movement at the time step n .

2.2.2 Fine-tuning of Body Dynamics for Target Tasks with Additional Learning

In the fine-tuning phase, the robot performs additional learning of the specific task with the adjustment of the acquired body dynamics. In additional learning, the learned parameters of the MTRNN (i.e., the weight values ω_{ij} and the initial values of the context C_{s_i}) in the pre-training process will be modified for specific tasks. Once the body dynamics have been acquired in the pre-training phase through motor babbling, they can be commonly used for different target tasks, i.e., there is no need to perform motor babbling before learning each task. In the case of direct-task learning without pre-training, the robot only learns the sensorimotor relationship for the specific task without motor babbling. Thus, the robot may not be able to generate appropriate motions for situations which has not been previously experienced. In contrast, having pre-trained body dynamics with motor babbling would be beneficial in efficient learning with different task requirements because of the generalization

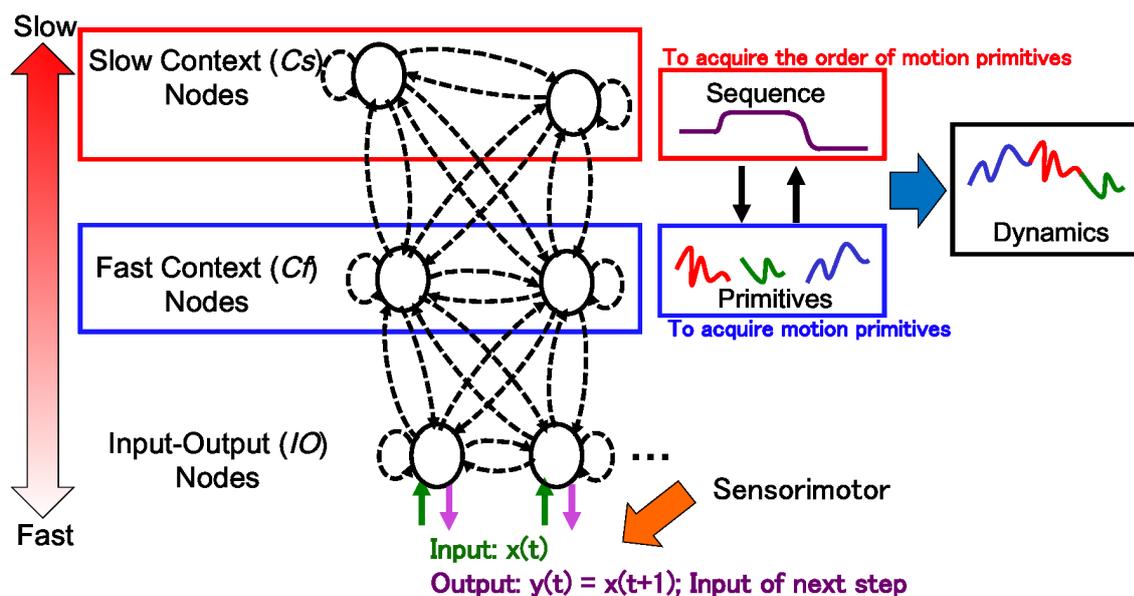


Figure 2.3: Dynamics representation with MTRNN. MTRNN can predict and generate the next state from the past history of neurons' states. The input value is calculated using the output values of the previous step. In this paper, joint angles, joint velocities, joint torques, and end-effector 3D positions are used as input-output values. MTRNN is composed of three types of neurons that have different time constant: input-output (IO) nodes, fast-context (C_f) nodes, and slow-context (C_s) nodes. In fast-context (C_f) nodes, small movement segments are acquired as motion primitives. In slow-context (C_s) nodes, the sequence of motion primitives is acquired. By combining the contexts, the entire motion dynamics are learned.

ability of the learned body dynamics with MTRNN as various motion primitives. In addition, the required learning iterations for the tasks could be reduced because the acquired body dynamics can be adjusted and reused for different target tasks. Although pre-training itself requires time to learn the body dynamics, the overall training required to learn various tasks reusing the acquired body dynamics could be more efficient than learning each task separately without pre-trained body dynamics.

2.3 Numerical Simulation Results

2.3.1 Simulated Robot Model

In numerical evaluations, I use a simulated robot model based on the structure of the humanoid robot ACTOROID [87]. This simulated robot has a seven-DOF right arm with flexible-joint actuation, which is implemented on the OpenHRP3 simulator [88] (see Figure 2.4). The total length of the arm is 0.684m and the range of motion (joint angles) is designed to be comparable to that of a human [89] (Table 2.1 and Table 2.2). In the flexible-joint model, the joint stiffness k , viscous damping d , and Coulomb friction coefficients c are given as $k = 0.2$, $d = 10.0$, and $c = 10.0$ for joints 1–4, and $k = 0.2$, $d = 50.0$, and $c = 50.0$ for joints 5–7, respectively (Fig. 2.4). These parameters are chosen so that the simulated robot model has higher joint compliance than the actual ACTOROID robot. For active motions, the motor torque command to each joint actuator is given as

$$\tau = P_{gain}(\theta_d - \theta) + D_{gain}(\dot{\theta}_d - \dot{\theta}), \quad (2.7)$$

where θ is the joint angle, $\dot{\theta}$ is the joint velocity, θ_d is the desired joint angle, $\dot{\theta}_d$ is the desired joint velocity, and P_{gain} and D_{gain} are the PD gains. For passive motions, the motor torque command is set to be zero.

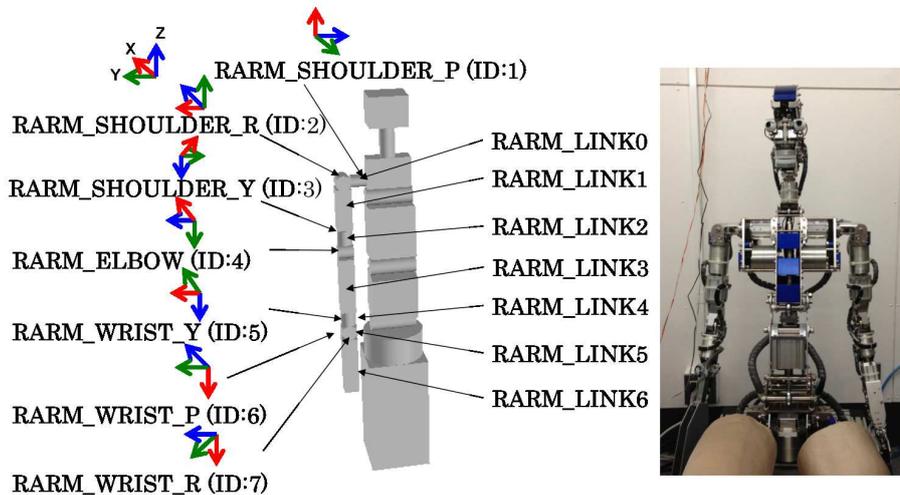


Figure 2.4: Simulated robot model. Left: joint configuration. Right: actual AC-TROID robot on which the simulated robot is based.

Table 2.1: Denavit-Hartenberg parameters and joint range of motion of the simulated robot model.

Joint Name (Arm)	a [mm]	α [deg]	d [mm]	θ [deg]	q^{max*} [deg]	q^{min*} [deg]
RARM_SHOULDER_P	0	90	171	90	180	-50
RARM_SHOULDER_R	0	90	0	90	140	-30
RARM_SHOULDER_Y	0	90	273	90	110	-80
RARM_ELEBOW	-9	90	0	0	145	0
RARM_WRIST_Y	0	-90	240	90	90	-85
RARM_WRIST_P	0	-90	0	-90	85	-85
RARM_WRIST_R	0	-90	0	0	15	-55

* range of motion of the joint angle

Table 2.2: Parameter of the simulated robot model

Link Name (Arm)	Mass [kg]	Center of mass	Moment of inertia
RARM_LINK0	1.0	(0.0 0.0 0.0)	[1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0]
RARM_LINK1	1.0	(0.0 0.0 0.0)	[1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0]
RARM_LINK2	1.0	(0.0 0.0 0.0)	[1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0]
RARM_LINK3	1.0	(0.0 0.0 0.0)	[1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0]
RARM_LINK4	1.0	(0.0 0.0 0.0)	[1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0]
RARM_LINK5	1.0	(0.0 0.0 0.0)	[1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0]
RARM_LINK6	1.0	(0.0 0.0 0.0)	[1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0]

2.3.2 Procedure for Motor Babbling with Simulated Robot Model

In performing motor babbling, first, the initial position of the arm is randomly set within its range of motion. Then, a sequence of three random desired angles for each joint of the arm (seven DOFs) are given at 3.0 s intervals. Then, I record the joint angles, joint velocities, torques, and the end-effector 3D positions of the resultant movement of the robot according to the joint torque given by (2.7) for a 9.0 s duration at a sampling interval of 0.1 s. The data from 5.5 s to 8.5 s are used as the training data for MTRNN. As depicted in Figure 2.2, during motor babbling, the motion is switched from the active motion in (2.7) to the passive motion at the time step of $n = 5$ (0.5 s). This process is repeated 1, 2, and 150 times to generate exploratory motions. For comparison in the evaluations reported in Section 2.3.4, I also performed motor babbling only with active motions, i.e., no switching to passive motions.

Once I have recorded the movement data above, the MTRNN is trained to acquire the body dynamics. The recorded joint angles, joint velocities, torques, and the end-effector 3D positions are used as the teaching signals in (2.4) to learn the weights ω_{ij} and initial values of context Cs_i according to (2.5) and (2.6), respectively. Table 2.3 presents the specifications of the MTRNN used in this paper. Note that the Input-Output (*IO*) nodes are only connected to the Input-Output (*IO*) nodes and

the fast-context (C_f) nodes, and the fast-context (C_f) nodes are fully connected to all the nodes, while the slow-context (C_s) nodes are only connected to the slow-context (C_s) nodes and the fast-context (C_f) nodes (Fig. 2.3). The size of the network was determined by empirically finding the minimum number of nodes that can successfully learn the motions in motor babbling and the given tasks. The network with a smaller size was preferred in order to reduce the required learning time and improve the generalization performance. The time constants were also empirically determined so that the error E in (2.4) could be successfully minimized.

Table 2.3: Design of MTRNN. Each type of node has different time constants. The structure of the MTRNN is depicted in Figure 2.3

Type of nodes	Node name	No. of nodes	Time constant τ
Input-Output (IO)	Joint angle input	7	2
	Joint velocity input	7	2
	Joint torque input	7	2
	end-effector 3D positions input	3	2
Fast Context (C_f)	Fast context	50	5
Slow Context (C_s)	Slow context	20	70

2.3.3 Target Tasks for Simulated Robot

To evaluate the effectiveness of the proposed approach to learning body dynamics, I consider dynamic tasks having constrained movement requiring interactions with the environment such as crank-turning and door-opening/closing tasks. Figure 2.5 depicts these two tasks in simulations. The crank-turning task (Fig. 2.5(a)) involves a constrained periodic rotational motion. The door-opening/closing task (Fig. 2.5(b)) consists of a repetitive motion with changes in the direction of the movement.

The teaching signals for these tasks are generated in the following three steps: 1) The end-effector of the robot arm is manually guided to passively follow the movement of crank rotation and door-opening/closing. In crank rotation, the movement of the crank is given at a constant angular velocity. In door-opening/closing, the angle of

the door is given as a fifth-order polynomial interpolation between the start and the end positions. During these manually guided movements, I obtain a sequence of joint angles of the arm that results from the passive movement. 2) I generate the robot movement using the recorded joint angles in step 1) as the desired joint angles, and then I record the resultant actual joint angles and the Cartesian positions of the end-effector during these generated motions. 3) For each task, I divide the path of the movement into the actively controlled part and the passive movement part depending on the nature of the task (see the specific descriptions for each task below). In the active control part, I apply the joint torque to the robot, and in the passive control part, I turn off the torque input and the robot follows the natural movement. The movement data in this final step such as the joint angles, joint velocities, joint torques, and the end-effector 3D positions are used as the teaching signals for each task.

In the crank-turning task, the robot repeatedly rotated the crank 10 times for a duration of 48.95 s (979 steps at a sampling interval of 0.05 s). Note that for the execution of the task, a different sampling interval was used from that of motor babbling since my MTRNN formulation uses discrete time representation which is independent from the actual time. Thus, the output of the MTRNN would depend on the choice of the sampling interval even for the same parameter settings. I have adjusted it so that the resultant velocity of the output of the MTRNN becomes similar to that of motor babbling. The diameter of the crank was 0.3m. To turn the crank, the robot only applied a force at the nearest and most distant positions from the body. Between these positions, the robot followed the passive movement of the arm and the crank without a torque input to the joints.

In the door-opening/closing task, the robot repeatedly opened and closed the door (start: 0° , end: 145°) 30 times for 66.95 s (1339 steps with a sampling interval of 0.05 s). The radius of the door-opening/closing trajectory was 0.55m. To open and close the door, the robot only applied forces at the initial point of the door-opening/closing. Similar to the crank-turning task, the robot did not apply forces between these two positions following the passive movement of the arm and the door without a joint torque input.

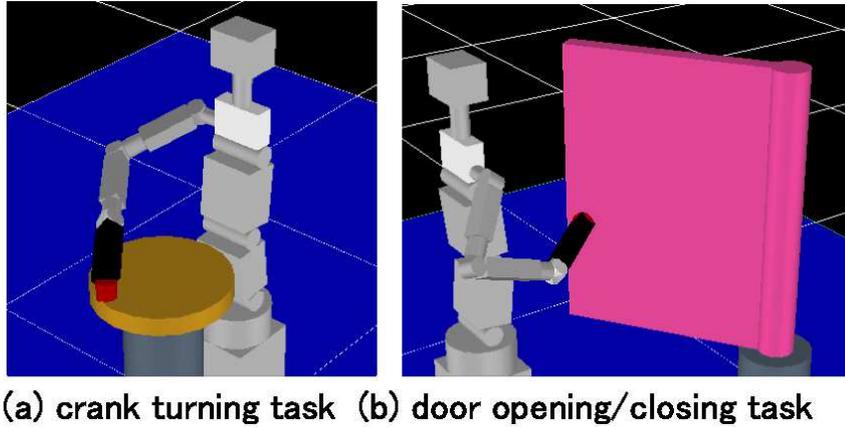


Figure 2.5: Tasks performed by the simulated robot model on the OpenHRP3 platform. These tasks involve constrained repetitive motion having interactions with the environments.

2.3.4 Simulation Results

2.3.4.1 Effectiveness of Motor Babbling with Active-Passive motions

To evaluate the effectiveness of using motor babbling with active and passive motions to acquire the body dynamics with pre-training, I compared the following three cases:

1. Learn the task directly *without* motor babbling
2. Learn the task after 150 motor babbling motions *with only active* movements
3. Learn the task after 1, 2, and 150 motor babbling motions *with active and passive* movements

Fig. 2.6 shows the minimum training iterations needed to perform the tasks successfully. Training iterations are defined as the number of iterations of BPTT required to reduce the mean-squared error given in (2.4) between the output from the trained RNN and the teaching signals. During the training of an RNN, the forward calculation in (2.1) and (2.2), and backward calculation in (2.4), (2.5), and (2.6) are performed alternately. The criterion of minimum training iterations is as follows: (a) in the case of crank-turning, the robot successfully rotates the crank 10 times repeatedly, (b) in

the case of door-opening/closing, the robot successfully opens and closes the door 30 times repeatedly covering the specified range of the door angles (start: 8° , end: 138°). When the robot was not able to perform the tasks within 160,000 training iterations, I defined it as failure in learning the tasks. Each experiment was conducted 10 times using the randomly set initial values for the weights in the MTRNN. Note that the training iterations shown in Figure 2.6 do not include the training iterations required in motor babbling. The numbers in the bar graph in Figure 2.6 show the success rate for learning the task. The mean and standard deviation of the training iterations are calculated from the number of motions with successful learning. Note that the success rate in Figure 2.6 is based on offline motion generation for the trained cases. The generalization ability for untrained situations with online generation of the movement will be discussed in Section 2.3.4.2.

As shown in Figure 2.6, the number of training iterations after 150 active-passive motor babbling was reduced by 60.7% (crank-turning task) and 66.0% (door-opening/closing task) in comparison to the case without motor babbling. I confirmed that this difference was significant using t-test ($p < 0.01$). The total number of training iterations including active-passive babbling (case (c), (d) and (e) in Figure 2.6) was still smaller than that of without babbling (case (a)) and including only active babbling (case (b)). In the case of only active motor babbling, although a reduction in the training iterations can be observed, the success rate of task learning was notably low. I suspect that with only active motor babbling, the robot was not able to learn the passive component of the robot dynamics that was to be exploited during execution of the task.

Here, I discuss the relationship between task-training iterations and the success rate of the tasks depending on the number of exploratory motions in motor babbling with active-passive motions. As shown in Figure 2.6, a small number of exploratory motor-babbling motions yielded higher success rates (both 100%) compared with the case having a large number of exploratory active-passive motions in motor babbling (60 and 80%). However, a large number of exploratory motions in motor babbling resulted in much smaller training iterations. This implies that there is a trade-off

between task-training iterations and the success rate depending on the number of exploratory motions in motor babbling. Increasing exploratory motions in babbling would be beneficial in acquiring a variety of motion primitives which could be reused for a given task. However, failure in task learning after motor babbling with a large number of exploratory motions may be due to overfitting in pre-training. I have observed excessive changes only in a small number of weights in the network in additional task learning in such a case. Further analyses of this issue and possible improvement would be of my future interest from a viewpoint of the acquired network properties.

2.3.4.2 Generalization Ability in Simulation

In this section, I evaluate the generalization performance of the learned motions for tasks with novel situations. I consider changing the diameter of the crank from the original size in the crank-turning task (Fig. 2.7 shows a crank with the diameter of 0.14m as an example). I compared the cases between the learned movement without motor babbling and that with motor babbling with different numbers of active and active-passive exploratory motions (one exploratory motion and 150 exploratory motions).

In order to generate movements in the novel situation, the actual joint angles and the end-effector 3D positions are used as an input $x(t)$ to the MTRNN in (2.3). As a result, the robot generates the movement with online state feedback according to the output of the predicted next state from the actual current state of the robot by the MTRNN, which is different from the offline feedforward-motion generation in Section 2.3.4.1 (where no actual robot state is fed back to the MTRNN).

Fig. 2.8 shows the success rate for the crank-turning task execution depending on the change in the diameter of the crank (0.18m and 0.14m) from the original size (0.30m). The criterion of the success is that the robot is able to rotate the crank 10 times repeatedly. Without motor babbling, the learned controller could not be generalized for the cases of untrained situations (success rate of 0%) with online motion generation. On the contrary, with offline motion generation, the success rate of the learned movement without motor babbling was very high. I consider that the trained network was likely to be over-fitted to specific training data. In contrast, with

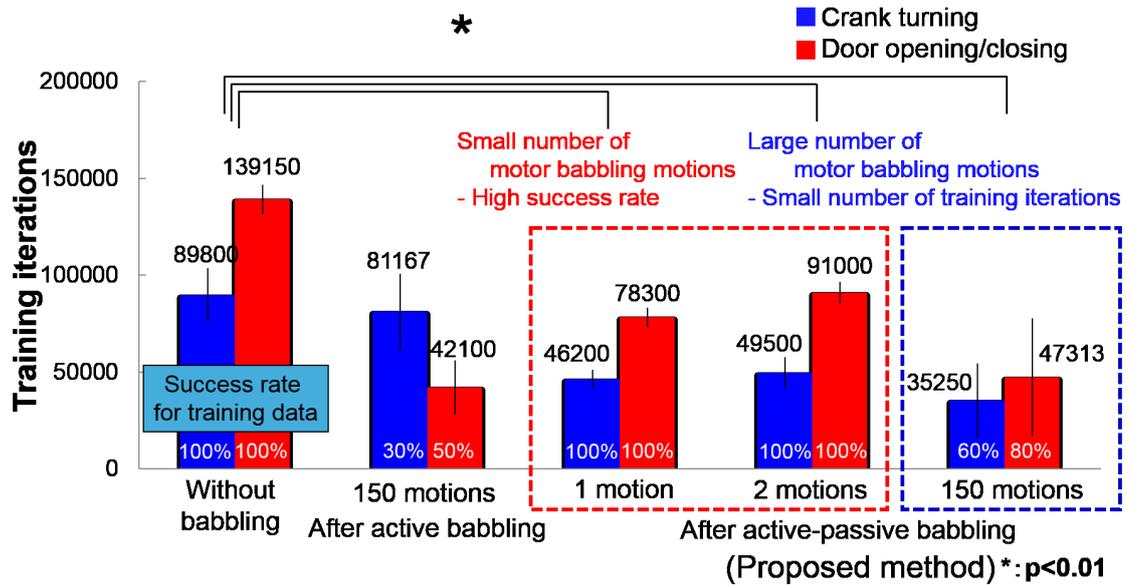


Figure 2.6: Simulation result of crank-turning and door-opening/closing tasks. Comparison of the training iterations for learning tasks (a) direct-task learning without motor babbling, (b) task learning with 150 motor-babbling motions consisting of only active movements, and (c) task learning with 1, 2, and 150 motor-babbling motions consisting of active-passive movements(c, d, e), respectively. The bar graph shows the minimum training iterations needed to perform the tasks successfully. Note that the training iterations do not include those required for motor babbling. The average iterations for motor babbling for each case (b-e) were (b) 27000, (c) 5800, (d) 10000 and (e) 20000. The error bar shows standard deviation of the training iterations. The numbers in the bar graph show the success rate for learning tasks with the training data. As the number of exploratory motions in motor babbling increased, the number of training iterations was reduced to achieve the task. However, I observed a trade-off between the training iterations and the success rate. As discussed in Section 2.3.4.2, even though the success rate with the training data is high in direct task learning without motor babbling, the learned movement could not be generalized for untrained cases.

motor babbling, the robot was able to successfully perform the crank-turning task for untrained situations in online-motion generation. Although the success rate was not remarkably high, the robot was able to perform the crank-turning task with as small as a 0.14m crank diameter (Fig. 2.7).

Note that the evaluation results of generalization performance in the crank-turning task is primarily reported here. In the door opening/closing task, I did not observe significant difference in the performance even with different door sizes. This is because the door opening/closing task is easier than the crank turning task, where the door opening/closing task only requires repetitive movement of about quarter of an arc, whereas the crank turning task requires full revolution of the crank for successful task execution.

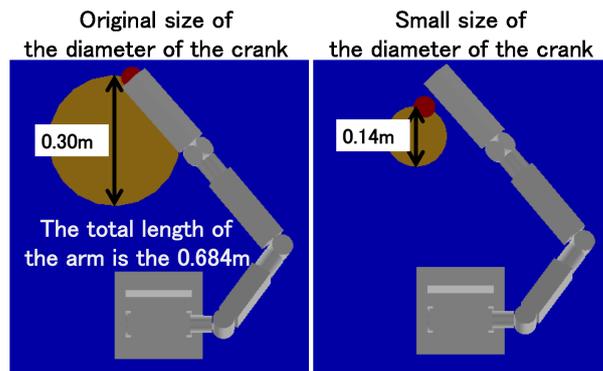


Figure 2.7: Using a crank with a smaller diameter (0.14m) to evaluate the generalization ability of the learned movement in the crank turning task. Left: crank with the original diameter of 0.30m. Right: crank with a small diameter of 0.14m.

2.3.4.3 Comparison of proposed method and PD control method

In this section, I compare the learned RNN with one exploratory motion in motor babbling with 90,000 training iterations performed in Figure 2.7 and PD control using the motion performed as a teaching signal for the crank-turning task (see item 3) in Section 2.3.3 regarding the generalization performance of the crank-turning tasks (the crank with a nominal diameter of 0.3m) as follows: 1) random noise between 0 to 5° is

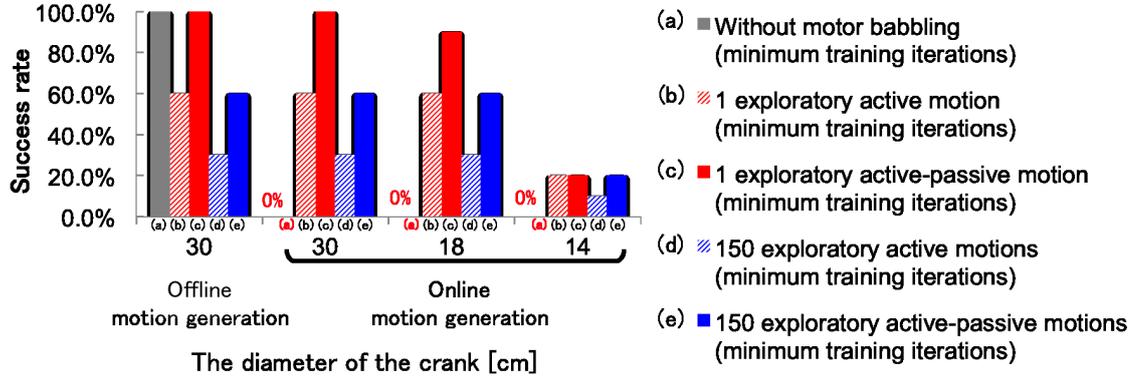


Figure 2.8: Simulation result for the evaluation of the generalization ability of the learned movement in the crank-turning task. The success rate of the generalization ability depends on the variation of the crank diameter from the original position. I compared the five cases: (a) without motor babbling, (b) one exploratory active motion in motor babbling with minimum training iterations, (c) one exploratory active-passive motion in motor babbling with minimum training iterations, (d) 150 exploratory active motions in motor babbling with minimum training iterations, and (e) 150 exploratory active-passive motor babbling with minimum training iterations.

added to the measured robot joint angles of the arm during the motion, 2) Coulomb friction ($c = 2.0$) is added to the crank, and 3) a constant external force (Torque $2.0Nm$ to the direction opposite to the direction of rotation which the robot rotates) is added as a disturbance in the direction opposite to the direction of rotation of the crank. The experiment with the proposed method was conducted 10 times using the randomly set initial values for the weights in the MTRNN.

As a result, the robot could successfully achieve the given tasks in these novel situations after motor babbling using the proposed method (the success rate of 100% to all situations). In contrast, the PD control method could not achieve the tasks in the novel situations. The results demonstrate the generalization capabilities of the proposed method compared to the PD control method.

2.4 Hardware Experiments with PR2 Robot

2.4.1 Experimental Platform: PR2

We use a PR2 [90] as a hardware platform. The right arm of the robot (seven DOFs) was used in my experiments (see Figure 2.10). The three joints of the PR2 arm (shoulder lift, upper arm roll, and elbow flex joints) exhibit flexibility because of its passive spring counterbalance mechanism, whose dynamical characteristics are difficult to model. For active motion, the actuator torque input is given by the following PD controller for each joint as:

$$\tau = P_{gain}(\theta_d - \theta) + D_{gain}(\dot{\theta}_d - \dot{\theta}). \quad (2.8)$$

To perform passive motion, the joint torque input is turned off so that the resultant arm movement follows its intrinsic passive dynamics.

2.4.2 Procedure for Motor Babbling with PR2 Robot

The robot performs motor babbling in the 3D space with the seven DOFs of the right arm. During the babbling motions, I measured the joint angles, angle velocities, joint torques, and end-effector 3D positions. When performing motor babbling, first, I randomly set the initial position of the arm within its range of motion. Then, a sequence of three random desired angles for each joint of the arm (seven DOFs) is given at 3.0 s intervals, and I record the joint angles, joint velocities, torques, and the end-effector 3D positions of the resultant movement of the robot at a sampling time of 0.1 s using the output of an RNN. The data from 4.0 s to 7.0 s are used for the teaching signals for the MTRNN. As in Figure 2.2, during motor babbling, the motion is switched from active motion to passive motion at a time step of $n = 20$ (2.0 s). In the experiment, I choose a single exploratory movement in motor babbling to achieve a high success rate for task learning. With the obtained motion data, the MTRNN was trained. In the hardware experiments, I used an MTRNN with the same structure as in the numerical simulations (see Table 2.3) although the structure and properties

of the robots are not exactly the same. The network structure, e.g., the number of the nodes, needs to be appropriately adjusted for the given robot platform depending on its complexity. In this paper, I choose to use the same network structure to make the experimental conditions similar in both cases.

2.4.2.1 Target Task with Real Robot, PR2

In hardware experiments with PR2, I consider a drawer-opening/closing task. Specifically, in this task, I performed a repetitive movement of continuous drawer-opening/closing, which involves constrained movement requiring dynamic interactions with the environment (unknown mass and friction of the drawer) and changes in the movement directions. In order to generate teaching signals, the start and end positions of the drawer-opening/closing movement are first specified by direct (manual) teaching with the robot, and I obtained the corresponding joint angles of the arm at these positions. Then, the robot performs linearly interpolated movements between these joint angles at the start and end positions, and I record the joint angles, joint velocities, joint torques, and the Cartesian positions of the end-effector during such movements. Figure 2.10 shows the drawer-opening/closing task with a length of 0.52m. To collect the training data, I recorded two repetitions of the drawer-opening/closing movement performed by the robot, whose total duration was 20.0 s (400 steps with a sampling time of 0.05 s).

2.4.3 Experimental Results with PR2 Robot

2.4.3.1 Effectiveness of Motor Babbling with Active-Passive motions

The simulation results in Section 2.3.4.1 suggested that motor babbling with only active motions was not particularly effective at acquiring the body dynamics. Thus, to evaluate the effectiveness of motor babbling during task learning for a real robot, I compared the following two cases:

1. Learn the task directly *without* motor babbling
2. Learn the task after motor babbling *with active and passive* motions

Fig. 2.9 shows the minimum training iterations required to perform the drawer-opening/closing task successfully. The criterion of the minimum training iterations is defined such that the robot successfully opens and closes the drawer two times repeatedly covering the specified range of drawer length (0.36 m) (Fig. 2.10). I conducted the experiment five times using different initial parameters for the weights in the MTRNN. Note that the training iterations in Figure 2.9 do not include the training iterations in motor learning. The average number of training iterations required for motor babbling itself was 20000. In this case, the total number of training iterations including active-passive babbling is larger than that of without babbling. However, in terms of generalization ability, as reported below, the learned network without babbling was not able to achieve tasks with novel situations. The numbers in the bar graph in Figure 2.9 show the success rate when learning the task with the training data. The mean and standard deviation of the training iterations were calculated from the number of motions with successful learning. According to the result in Figure 2.9, the number of training iterations with active-passive motor babbling was reduced by 51.5% when compared with the case without motor babbling. I confirmed that this difference was significant using t-test ($p < 0.01$). The result demonstrates the effectiveness of active-passive motor babbling in learning body dynamics.

2.4.3.2 Generalization Ability with PR2 Robot

In this section, I evaluate the generalization performance of the learned motions for the drawer-opening/closing task with novel situations. I consider three novel situations: (a) changing the number of repetitions of the drawer-opening/closing movement increased to four from two, (b) motion generation using only end-effector 3D positions, (c) motion generation using only joint angles with added uniformly distributed random noise in the interval of 0 and 10°. In (a) and (b), I compared the cases between the learned movement *without* motor babbling and that *with active-passive* motor babbling. In (c), I compared the cases between PD control and the learned movement with active-passive motor babbling. For each novel situation, I generated movements as follows: In (a), I generated the longer duration of motion to achieve four repetitions of drawer-opening/closing movement by extending time

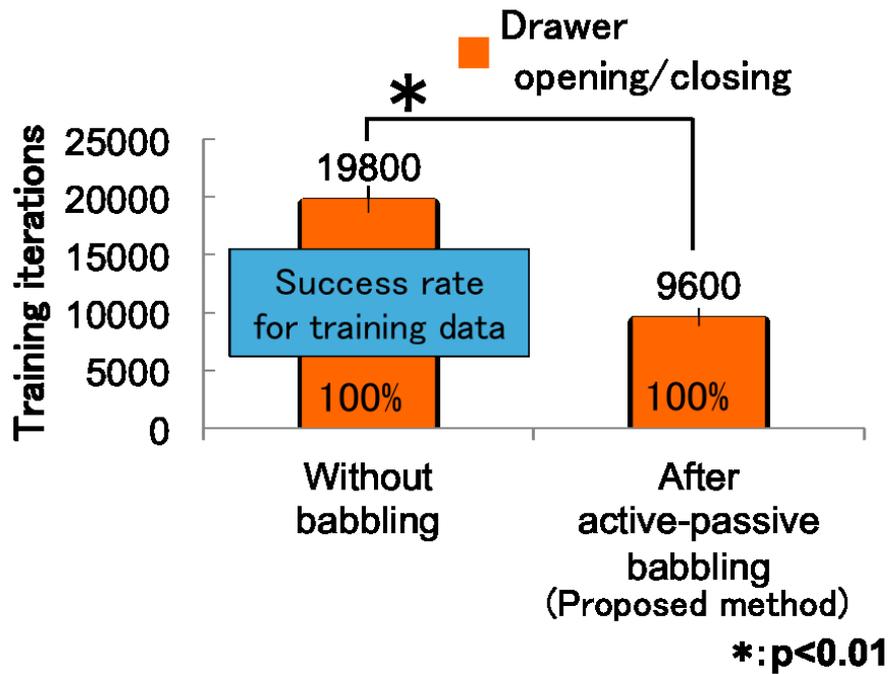


Figure 2.9: Experimental result with PR2 of the drawer-opening/closing task. Comparison of the training iterations for learning tasks 1) direct-task learning without motor babbling, 2) task learning after active-passive motor babbling with one motion consisting of active-passive motions. The figure shows the minimum training iterations required to perform the drawer-opening/closing task successfully. Note that training iterations does not include the training iterations during motor babbling. The average of number of training iterations for motion active-passive motor babbling itself is 20000. The error bar shows standard deviation of the training iterations. The numbers in the bar graph show the success rate in the learning tasks with offline motion generation using the training data.

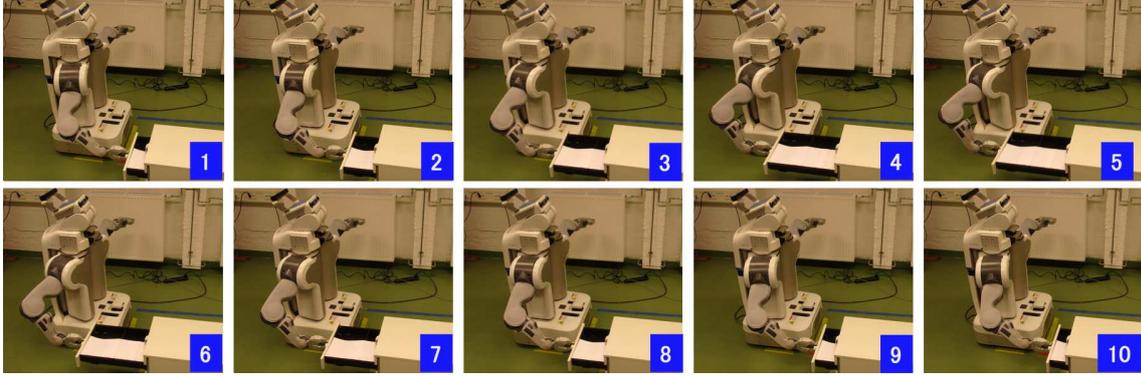


Figure 2.10: Snapshots showing task execution of drawer-opening/closing during 11.0 s to 20.0 s with PR2 after motor babbling of the active and passive motions.

evolution of the learned RNN. In (b), I generated motion by calculating $C_s(0)$ in (2.6) only using to the target end-effector 3D positions. In (c), I added random noise to the training data for the input neuron in (2.3) corresponding to the joint angles. The experiment with the proposed method was conducted five times using the randomly set initial values for the weights in the MTRNN.

In the result of (a) *without* motor babbling, the robot could not achieve the task of four repetitions of the movement, i.e., the robot could not generate the desired motion after two times drawer-opening/closing (the success rate of 0%). In contrast, in the result of (a) *with* motor babbling, the robot successfully generated the task motion with four times of drawer-opening/closing repetitions (the success rate of 100%). The result demonstrated that the RNN with motor babbling could learn the task movement as a cyclic motion even with only two repetitions in training.

In the result of (b) *without* motor babbling, the robot could not achieve the task (the success rate of 0%). In the result of (b) *with* motor babbling, the robot could successfully generate the task motion using only the partial information. Furthermore, it was demonstrated that the robot could achieve the task in the presence of added noise (Gaussian noise of the interval of -25 and 25mm) as well as the delay (50 time steps) in the 3D end-effector positions (the success rate of 100% in all these cases).

In the result of (c) *with* motor babbling, the robot could achieve the task in the

presence of added noise to the input neuron in the RNN (the success rate of 100%). In contrast, the robot could not perform the task with PD control because of the added noise. This result demonstrates the robustness of the proposed method in generating the desired movement against noise.

2.5 Discussion about Motor Babbling

In this chapter, random motion was used as motor babbling. However, it is difficult to apply random motion to an actual robot when the enormous number of random motion is required to acquire body model. Motor babbling is considered to be exploratory movement [91–94]. Some motions are easy to predict, whereas others are considerably more difficult. In this section, I discuss about exploratory form of motor babbling to explore and learn motions that are difficult to predict, enabling body dynamics to be acquired efficiently.

2.5.1 Exploratory Motor Babbling Using Variance Prediction from RNN

2.5.1.1 Stochastic Multiple Time-scales Recurrent Neural Network

In motor babbling, there are easy-to-predict motions and those that are more difficult to predict. If a robot has more experience of a motion, it will be easier to predict. This means that the variance will be small. If a robot has less experience of a motion, it will be more difficult to predict. This means that the variance will be large. The robot can evaluate the prediction accuracy of its motion from the variance. Therefore, to learn efficiently, the robot needs to explore and learn motions that are difficult to predict.

To realize this, I use a Stochastic MTRNN (S-MTRNN). The S-MTRNN can predict and generate the next state from the past history of neurons' states, and determine the prediction accuracy from the variance [95,96]. To train the S-MTRNN, I use the maximum likelihood method with the steepest descent method [97]. This algorithm is composed of forward calculations and backward calculations as follows:

In the forward calculation, the output of the neurons is computed. First, the internal value of the i th neuron at step t , $u_i(t)$, is calculated as

$$u_i(t) = \begin{cases} (1 - \frac{1}{\tau_i})u_i(t-1) + \frac{1}{\tau_i} \left\{ \sum_{j \in I_{IO}} \omega_{ij}x_j(t) + \sum_{j \in I_{C_f} \cup I_{C_s}} \omega_{ij}C_j(t-1) \right\} & (i \in I_{C_f} \cup I_{C_s}), \\ \sum_{j \in I_{IO}} \omega_{ij}C_j(t) & (i \in I_{IO} \cup I_V), \end{cases} \quad (2.9)$$

where τ_i is the time constant of the i th neuron, I_{IO} , I_{C_f} , and I_{C_s} are the set of indices for the input-output (IO), fast-context (C_f), and slow-context (C_s) nodes, respectively, $x_j(t)$ is the input value of the i th neuron from the j th neuron, $C_j(t)$ is the activate state of the i th context neuron from the j th context neuron, ω_{ij} is the weight value of the i th neuron representing the connection from the j th neuron. Then, the activation states of context neuron, output neuron, and variance neuron is obtained by substituting the internal value (2.9) into the following activation function:

$$C_i(t) = \tanh(u_i(t)) \quad (0 \leq t \wedge i \in I_{C_f}, I_{C_s}), \quad (2.10)$$

$$y_i(t) = \tanh(u_i(t)) \quad (1 \leq t \wedge i \in I_{C_f}, I_{IO}), \quad (2.11)$$

$$v_i(t) = \exp(u_i(t)) \quad (1 \leq t \wedge i \in I_{C_f}, I_V), \quad (2.12)$$

The next input value to the i th neuron in (2.9) is recursively calculated using the output value in (2.10), (2.11) and (2.12) of the previous step as

$$x_i(t+1) = y_i(t). \quad (2.13)$$

In this chapter, the output $y_i(t)$ corresponding to the joint-angle node is used as a target joint angle of the robot. When the robot generates the motion *offline* to replay the movement of the training data set, the predicted output $y_i(t)$ of the S-MTRNN is used as the input $x_i(t+1)$ in (2.13), i.e., no actual robot states are fed back to the S-MTRNN as an input.

In backward calculation, firstly, the learnable parameters of the network are denoted as q . The probability density function p of the teaching signal $T_i(t)$ is defined

as

$$p(T_i(t)|\{x(t')\}_{t'=1}^t, q) = \frac{1}{\sqrt{2\pi v_i(t)}} \exp\left(-\frac{(T_i(t) - y_i(t))^2}{2v_i(t)}\right) \quad (2.14)$$

where $v_i(t)$ is the variance state of the i th neuron at step t generated by networks.

The likelihood function L_{out} is defined as

$$L_{out} = \prod_{t=1}^T \prod_{i \in I_{IO}} p(T_i(t)|\{x(t')\}_{t'=1}^t, q) \quad (2.15)$$

The parameters q are optimized through the learning process in the direction to maximize the likelihood L_{out} . More precisely, I use the gradient ascent method with a momentum term as the procedure for the parameter optimization. Here, the logarithm of the expression in (2.15) is used to facilitate the calculation.

$$\ln L_{out} = \prod_{t=1}^T \prod_{i \in I_{IO}} \left(-\frac{\ln(2\pi v_i(t))}{2} - \frac{(T_i(t) - y_i(t))^2}{2v_i(t)} \right) \quad (2.16)$$

The network generates v as an estimate of the prediction error. The squared error is divided by the variance; therefore, it is possible to avoid unstable learning. Namely, if the variance is large, the influence of the prediction error will decrease. If the variance is small, the influence of the prediction error will increase. Learning is performed according to the maximum likelihood using the gradient descent method. Therefore, the variance is calculated from the input signal without a teaching signal.

2.5.1.2 Learning Process of Exploratory Motor Babbling

This section describes the learning process of exploratory motor babbling. This method is composed of two steps (Figure 2.11). First, a robot learns the random motor babbling and its variance. The number of motor babbling motions is N_r . Then, the basic body dynamics are acquired.

In the next step, the robot performs exploratory motor babbling by applying noise to its joint angles based on the predicted variance acquired in the previous step. The number of exploratory motor babbling motions with this added noise is N_e . The noise

added to the joint angle is expressed by the following equation based on a Gaussian distribution:

$$\theta_{new} = \theta_{old} + \frac{1 - (\bar{v} - v_i)}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right), \quad (2.17)$$

where θ_{new} is new joint angle, θ_{old} is old joint angle, $\sigma^2(= 15.0)$ is the variance, x is random value, \bar{v} is the average variance of joint angles of all sequences, and v_i is the average variance of joint angles of each sequence.

Since the variance is large in the vicinity of unpredictable motion, the noise added to the joint angle is increased. Therefore, the robot will gain more experience of motions that are similar to those that are difficult to predict. By doing so, the robot acquires body dynamics with higher prediction accuracy.

2.5.2 Numerical Simulation Results

2.5.2.1 Simulated Robot Model

In numerical evaluations of exploratory motor babbling, I use a simulated robot model based on the structure of the humanoid robot ACTOROID [87] as with section 3.3.1. In the experiment, all joints were stiff instead of flexible joint, and only three DOFs (joint ID 2, 4, 6 in Figure. 2.4) of the seven DOFs of the robot's arm were used, and the remaining four DOFs are maintained constant. For motions, the motor torque command to each joint actuator is given by (2.7). The simulations were performed on the Ubuntu 12.04 LTS operating system with an Intel Core i7-4790 CPU.

2.5.2.2 Procedure for Exploratory Motor Babbling

To evaluate the effectiveness of the proposed approach, the robot's movement was confined to two-dimensional movements on the plane of a desk with only three of the seven DOFs in the robot's arm. To perform motor babbling, first, the initial position of the arm is randomly set within its range of motion. The robot performed motor babbling for 30 steps over a period of 2.07 s. Then, I recorded the joint angles, joint velocities, torques, and the end-effector 2D positions of the resultant movement of the robot. The number of random motor babbling motions N_r was 15.

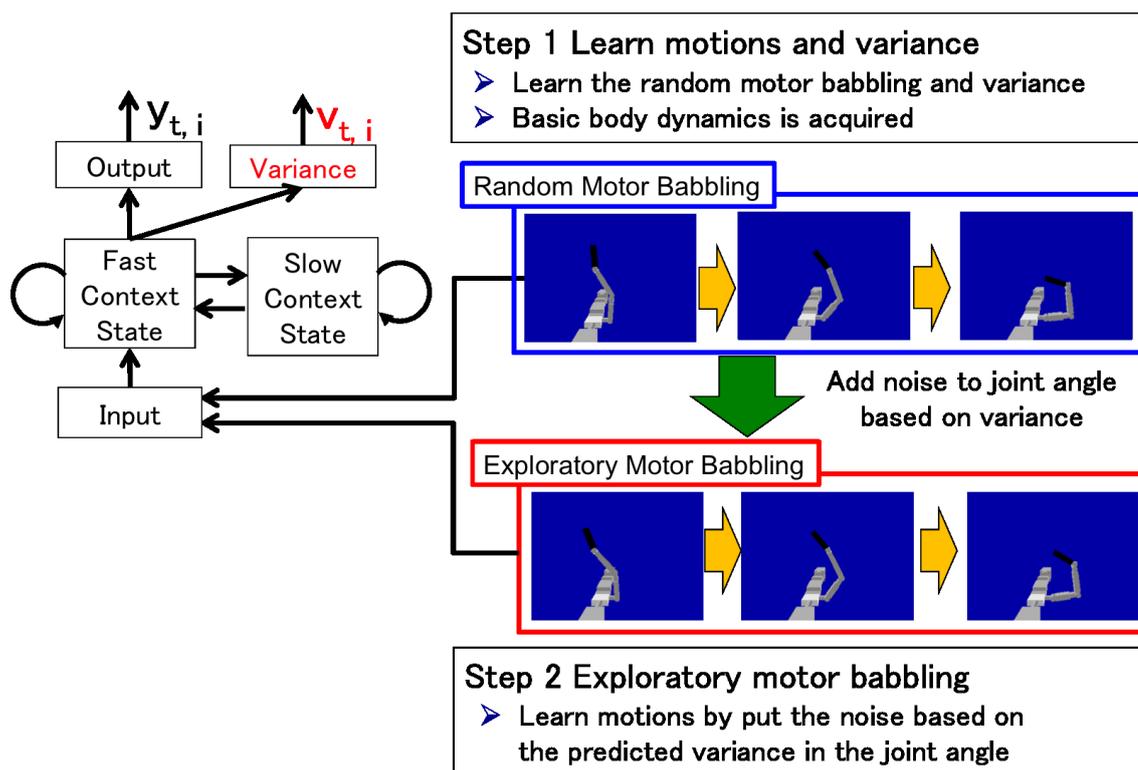


Figure 2.11: Exploratory motor babbling. The proposed method is composed of two steps. First, the robot performs random motor babbling and learns the motion and its variance. Then, the robot performs exploratory motor babbling motion to add noise to joint angles based on acquired variance at the first step.

Once I have recorded the movement data above, the S-MTRNN is trained to acquire the body dynamics and variance which was produced by adding a noise term based on the predicted variance in the joint angle. In this research, noise was added to all random motor babbling motions. Therefore, the number of exploratory motor babbling motions with added noise N_e was 15. Figure 2.11 illustrates the motor babbling, where the upper figure shows random motor babbling and the lower part shows exploratory motor babbling.

To train motor babbling and the task described in the next section, I used the S-MTRNN. Table 2.4 describes the construction of the S-MTRNN. Note that the Input-Output (IO) nodes are only connected to the fast-context (C_f) nodes, and the fast-context (C_f) nodes are fully connected to all the nodes including variance (V) nodes, while the slow-context (C_s) nodes are only connected to the slow-context (C_s) nodes and the fast-context (C_f) nodes. The probability distribution p of the teaching data was assumed to be 0.01.

Table 2.4: Design of S-MTRNN. Each type of node has different time constants.

Type of nodes	Node name	No. of nodes	Time constant τ
Input-Output (IO)	Joint angle input	3	2
	Joint velocity input	3	2
	Joint torque input	3	2
	End-effector 2D positions input	2	2
Variance (V)	Variance	11	-
Fast Context (C_f)	Fast context	20	5
Slow Context (C_s)	Slow context	2	70

2.5.2.3 Target Tasks for Simulated Robot

To evaluate the effectiveness of the proposed approach to learning body dynamics, I consider dynamic tasks having constrained movement requiring interactions with the environment such as crank-turning and door-opening/closing tasks as with section 2.3.3.

The crank had a diameter of 0.3m. The robot turned the crank five times. Each rotation consisted of 240 steps over a period of 12.0 s. Thus, the total motion consisted of 1200 steps, taking 60.0 s. The radius of the arc of the door’s trajectory was 0.55m. The robot opened and closed the door 10 times. Each opening/closing cycle had 120 steps, and took 6.0 s. Thus, there were a total of 1200 steps over 60.0 s.

2.5.3 Experimental Results and Discussion

To evaluate the effectiveness of exploratory motor babbling, I conducted the following experiments:

- Learn the task directly *without* motor babbling
- Learn the task after *random babbling* motions
- Learn the task after *exploratory motor babbling* motions

Figure 2.12 shows the number of iterations that were required to learn the tasks. This number of iterations was needed by the robot to generate the motion of crank turning and door opening/closing. Each experiment was performed five times with different initial parameters for the neural network, and the figures show the mean and variance over these repetitions. For the crank turning task, my method produced a reduction of 87.2% in terms of learning cycles compared with random babbling. I confirmed that this difference was significant using a t-test ($p < 0.05$). The mean computation time of the learning cycle with exploratory motor babbling was 273.5 s, whereas random motor babbling with the same number of motion patterns required 2453.0 s. Without any motor babbling, the mean computation time was 2650.2 s. For the door opening/closing task, my method produced a reduction of almost 75.0% when using the proposed method compared with random babbling. I confirmed that this difference was again significant using a t-test ($p < 0.05$). The mean computation time of the learning cycle with exploratory motor babbling was 535.0 s; random motor babbling and the same number of motion patterns took 1916.3 s, and without any motor babbling the learning cycle required 2130.7 s. In summary, Figure 2.12 indicates that exploratory motor babbling drastically decreased the learning time.

Fig. 2.13 shows the angle of crank turning and door opening/closing after 2350 and 3540 iterations, respectively. These iteration numbers correspond to one of the evaluated times in Figure 2.13 at which the target motions had been generated correctly with exploratory motor babbling. The joint angles of the teaching signal and the motion learned with exploratory motor babbling are similar. In the case of crank turning with random motor babbling and without motor babbling, the robot could not turn the crank, and performed forward and reverse rotation repeatedly. In the case of crank turning with random motor babbling and without motor babbling, the angle of door opening and closing became gradually smaller. It is clear that the robot can perform these motions correctly after a relatively small number of iterations using exploratory motor babbling.

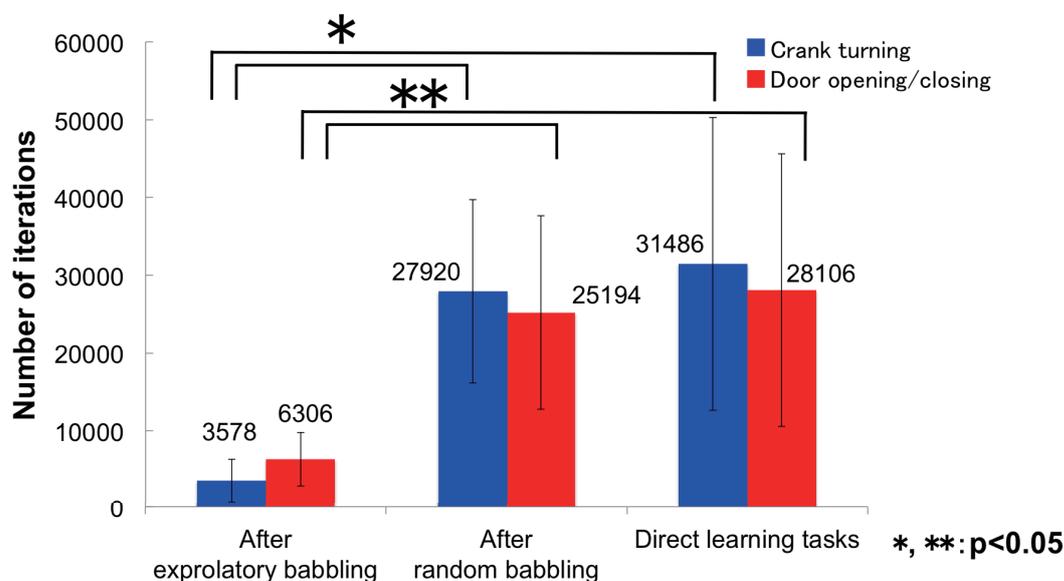


Figure 2.12: Learning cycles for completing the task of crank-turning and door-opening/closing

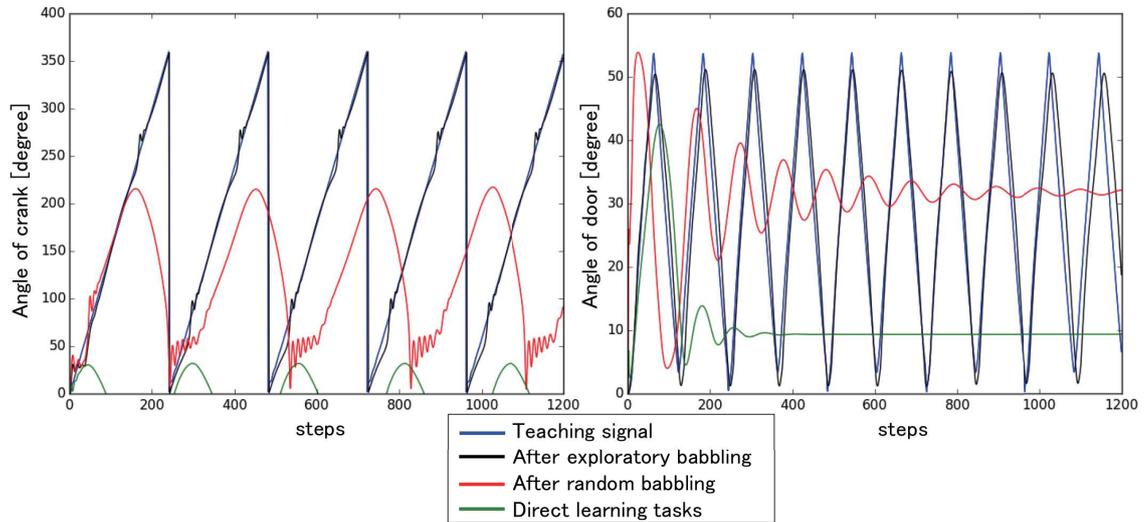


Figure 2.13: Generated motions of crank-turning and door-opening/closing after 2350 and 3540 iterations (Left: crank turning, Right: door opening/closing)

2.6 Summary

In this chapter, I presented an effective two-phase strategy for learning the body dynamics of multi-DOF flexible-joint robots in order to achieve dynamic-motion tasks motivated by a deep learning strategy. In the pre-training phase, the robots acquired the body dynamics via motor babbling with active and passive movements using an RNN. In the fine-tuning phase, the learned body dynamics were adjusted by performing additional learning of the target task. I presented the numerical results with a simulated flexible-joint robot model and hardware experimental results using a real PR2 robot both of which have a seven-DOF redundant arm to demonstrate the effectiveness of the proposed method while performing various tasks. The results showed that the number of training iterations for the target tasks was reduced after learning the body dynamics with active-passive motor babbling. Furthermore, with active-passive motor babbling, I demonstrated the generalization ability of the learned task for untrained situations. However, I observed that the motor-babbling performance depended on the initialization of the weights of the MTRNN. In addition, I found that there is a trade-off between the number of training iterations and the success

rate for task accomplishment depending on the number of exploratory motor babbling motions. Furthermore, I employed an explorative motor babbling scheme using variance predictions from an RNN by efficiently acquiring body dynamics with a small number of motor babbling movement.

Chapter 3

Extension of the Robot Body

Model with Tool-body Assimilation

3.1 Introduction

In this chapter, I propose a tool-body assimilation model that considers grasping during motor babbling for using tools. A robot with tool-use skills could be useful in a human-robot symbiosis situation because this allows the robot to expand its task performing abilities. Almost all existing studies for robots to use tools require predetermined motions and tool features; the motion patterns are limited and the robots cannot use novel tools. Some of the other past studies fully search for all available parameters for novel tools, but this leads to massive amounts of calculations. Other past studies approaches were mainly focused on obtaining the functions of the tools, and showed the robot starting its motions with a tool pre-attached to the robot. This implies that the robot would not be able to decide whether and where to grasp the tool. In real life environments, robots would need to consider the possibilities of tool-grasping positions, then grasp the tool. To address these issues, the robot performs motor babbling by *grasping* and *nongrasping* the tools to learn both tool functions and the robot's body model. In addition, the robot grasps various parts of the tools to learn the different tool functions from the different grasping positions. These rich motion experiences are learned using deep learning such as a deep neural network

(DNN) and multiple time-scale recurrent neural network (MTRNN). Tool features were self-organized in parametric bias, modulating the body model according to the tool in use. Finally, I designed a neural network for the robot to generate motion only from the target image. To evaluate the model, I have the robot manipulate an object task without any tools or with several tools of different shapes. I have the robot generate motions after showing the initial and target states by deciding whether and where to grasp the tool. Therefore, the robot is capable of generating the correct motion and grasping decision when the initial and target states are provided to the robot.

3.2 Tool-Body Assimilation Model Considering Grasping

In this section, I present the process of a tool-body assimilation model that considers grasping during motor babbling for using tools. Figure 3.1 shows an overview of the model, which consists of the following four modules: a motor babbling module, an image-feature-extraction module with the DNN, a body model module with the MTRNN, and a body modulation module with extra-context nodes. The process consists of three phases:

1. Acquisition of the body model using the DNN and MTRNN through motor babbling
2. Training the body modulation module with extra-context nodes
3. Motion generation using the initial state and a target image

Figure 3.2 shows the learning process of tool-body assimilation. The robot acquires training data from motor babbling, producing joint angles, grasping signal, and image data of *grasping* and *nongrasping* motions. During this process, the robot moves its arm with the target object and tool present. When the robot touches the tool during motion, the robot either *grasps* or *does not grasp* the tool, to gain experience of using and not using the tool. The image data dimension is then reduced using

the image-feature-extraction module. Next, relationship between the image features of nongrasping motions, joint angles, and the corresponding grasping signal is learned by the body model module as the robot's body model. The relationship between joint angles, image features, and corresponding grasping signal is then learned by the body modulation module for grasping decisions, resulting in tool functions. After training, the system is capable of generating motions when provided with the initial state and a target image.

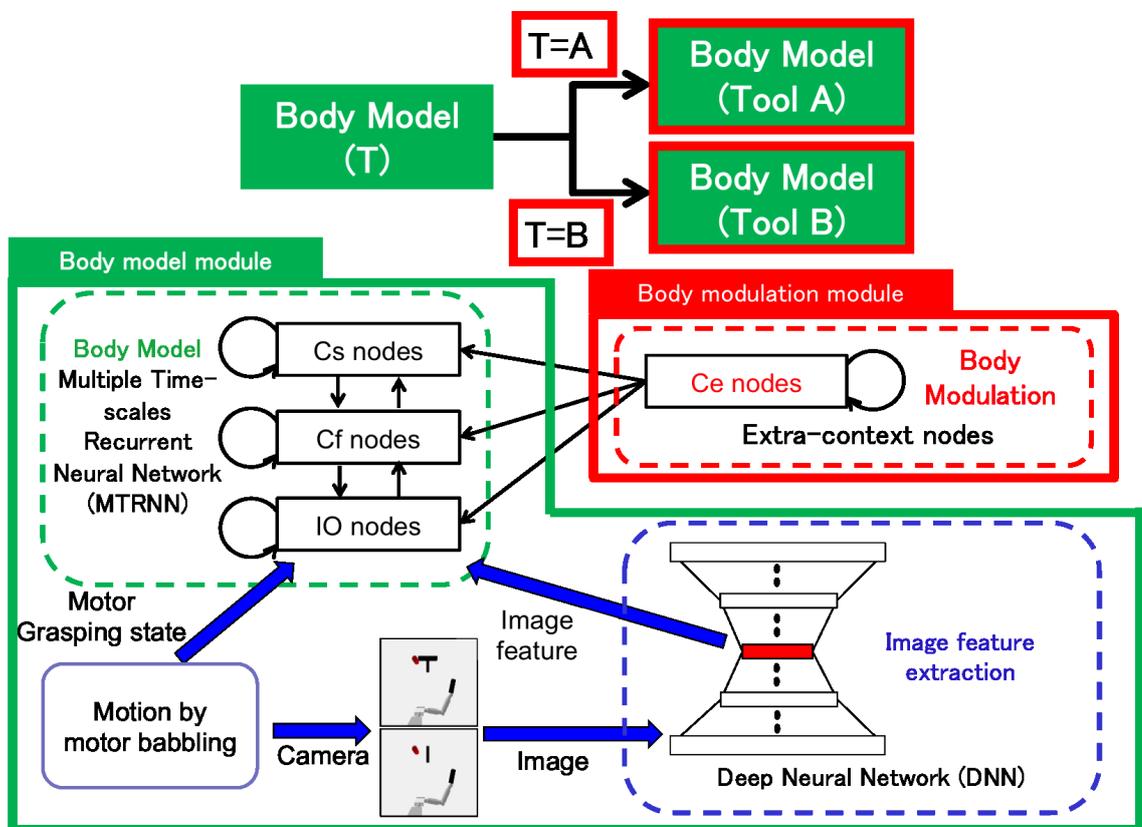
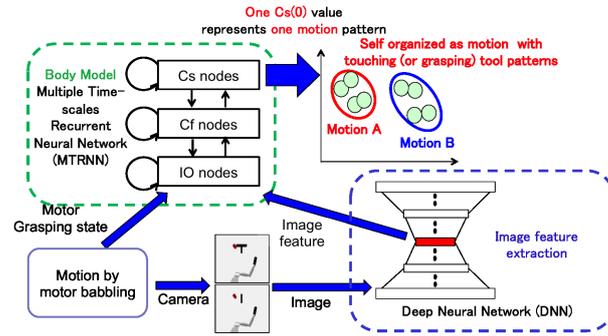
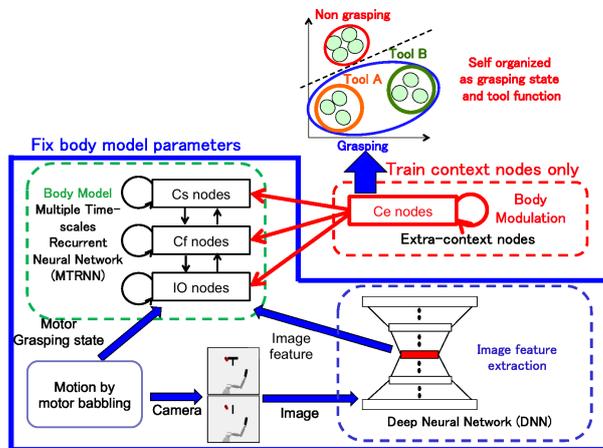


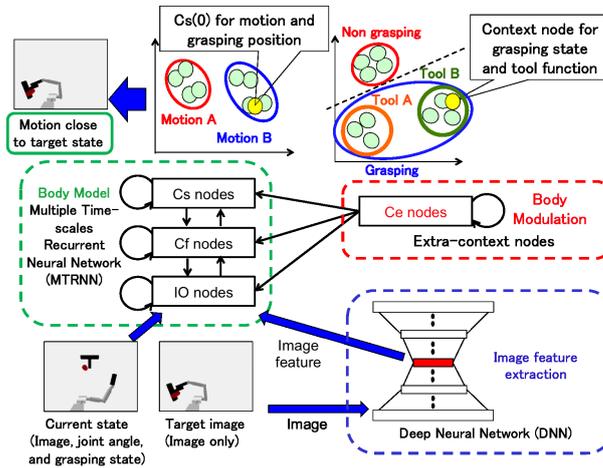
Figure 3.1: Tool-body assimilation model. The model consists of four modules: Motor babbling module, image-feature-extraction module obtained through the DNN, body model module obtained through the MTRNN, and body-model modulation module with extra-context nodes. The model is separated into two parts: the body model and body modulation.



(a) Learning process of body model



(b) Learning process of body modulation



(c) Process of motion generation

Figure 3.2: Learning process of tool-body assimilation. (a) The robot learns its body model through the DNN and MTRNN using motor babbling. The initial values of context nodes in the MTRNN are self-organized as motions with touching (or grasping) tool patterns. (b) The robot learns only the body modulation module with extra-context nodes by fixing the parameters of the body model. The grasping state and tool function are self-organized in the extra-context nodes. (c) The robot generates motion through current state and a target image.

3.2.1 Acquisition of Body Model through Motor Babbling

In this phase, the robot learns the relationship between the joint angles and sensors, i.e., the image features and grasping state, to acquire its body model through motor babbling using the DNN and MTRNN (Figure 3.2-(a)). The robot performs the motions of manipulating the tool without grasping (*nongrasping* motions) by considering various manipulating positions during motor babbling. Raw image data is used to train the image-feature-extraction module using the DNN. After the training, the image-feature-extraction module is capable of extracting image features for untrained data. Next, the sequence of image features, joint angles, and grasping state is used to train the body model using the MTRNN. The motion patterns with touching (or grasping when the robot performs tool grasping) tool patterns are self-organized in the MTRNN.

3.2.1.1 Image Feature-Extraction with DNN

DNNs are multilayered feed-forward neural networks proposed by Hinton et al. [33]. DNNs allow for highly precise speech and image recognition [35, 36]. In recent years, DNNs have gained attention in robotics studies [28, 29, 59, 68, 69, 98, 99]. In this study, a sandglass-type DNN (the number of n -th layer neurons satisfy $D_n > D_{n+1}$ in the encode layer and $D_n < D_{n+1}$ in the decode layer) is used as an auto-encoder, in which D_n is fully connected to D_{n+1} , and the central hidden layer is reconstructed as the output layer. This is achieved by training the auto-encoder to provide output values that are equal to the input values. The output of the n -th layer h_n is calculated as follows:

$$h_n = f(Wh_{n-1} + b), \quad (3.1)$$

where f is the activating function (the sigmoid function is used in this study), W is the weight matrix, and b is the bias. Thus, the DNN can reduce high-dimensional image data and extract image features without predetermined information.

In this study, I used a DNN training approach, which is based on the Hessian-free approach proposed by Martens [100], because this method is more time efficient than its pre-training and fine-tuning counterpart [33]. In Newton's method, the parameters

$\theta \in \mathbb{R}^N$ of an objective function f are iteratively calculated by updating θ as $\theta + \alpha \hat{p}$ for α with search vectors \hat{p} . The primary concept of Newton's method is that f , which can be locally approximated around each θ , is calculated as

$$f(\theta + \hat{p}) \approx q_\theta(\hat{p}) \equiv f(\theta) + \nabla f(\theta)^\top p + \frac{1}{2} p^\top B p \quad (3.2)$$

where matrix B is either the Hessian matrix $H = \nabla^2 f$, or an approximation of it. In the standard Newton's method, $q_\theta(\hat{p})$ is optimized by computing the $N \times N$ matrix B , and solving the system $B\hat{p} = -\nabla f(\theta)^\top$. However, calculation cost is high when N is large, even for modestly sized neural networks. Therefore, Martens used an N -dimensional vector d and the linear conjugate gradient algorithm (CG) [100] for optimizing quadratic objectives. This requires only matrix vector products with B . Details about the implementation are provided in several other studies [100–102]. The modified Newton's method is applied by reconditioning Hessian matrix H as

$$B = H + \lambda I \quad (3.3)$$

where B is the damped Hessian matrix of f at θ , $\lambda \geq 0$ is a damping parameter, and I is the unit matrix.

3.2.1.2 Body Model Acquisition with MTRNN

The MTRNN proposed by Yamashita and Tani is a variant of an RNN [61]. RNNs have desirable characteristics for predicting the next state from the history of a neuron state by preserving the internal state using recurrent connection [71]. The MTRNN is capable of learning multiple sequences of data to consider trajectories with branch structures¹, e.g., multiple goals or objectives. The MTRNN is composed of three types of neurons that have different time constants, which determine the node firing rates, and thus affect the manner in which the following nodes remember short- and long-term data: input-output (IO), fast-context (C_f), and slow-context (C_s) nodes (Figure 2.3). The input-output (IO) nodes are associated with sensorimotor states.

¹An example of a sequence of actions with branching trajectories is discussed in [61].

The fast-context (C_f) nodes learn small segments in the data as primitive movements, whereas the slow-context (C_s) nodes learn the sequence of the primitives in the data. By combining these nodes, the MTRNN learns the dynamics of the data sequence. In addition, a specific motion and sequence can be generated by considering the multiple learned patterns and computing the corresponding acquired initial values of the slow-context (C_s) nodes. In this study, the C_s space is self-organized by motion patterns with touching (or grasping) tool patterns. The back propagation through time (BPTT) algorithm [32] is applied during the learning phase of the MTRNN. The calculation of the algorithm is the same as with section 2.2.1.

3.2.2 Body Model Modulation Using Extra-context Nodes

In this phase, the robot acquires the grasping decision and tool function as body model modulation through motor babbling, using the *grasping* and *nongrasping* tools by considering various manipulating and grasping positions with transfer learning in terms of reusing the body model [103] (Figure 3.2-(b)). Transfer learning is the learning of a framework to apply the learned knowledge of a task to other tasks.

In addition to the input-output (IO), fast-context (C_f), and slow-context (C_s) nodes, the extra-context (C_e) nodes are connected to the MTRNN. The extra-context (C_e) nodes are connected unidirectionally to themselves and to the input-output (IO), fast-context (C_f), and slow-context (C_s) nodes, whereas other nodes are connected bidirectionally (Figure 3.1).

The *grasping* and *nongrasping* data is used to train the extra-context (C_e) nodes as body modulation modules. As modulation can be performed at any time, the robot is allowed to start its motion without the tool being pre-attached to its hand. During training, the weights and $C_s(0)$ of the acquired body model module in the previous phase are fixed, and only the weights and $C_e(0)$ of the body modulation module are trained. The robot performs the same motions with respect to joint angles for *grasping* and *nongrasping*. As the robot performs the *grasping* or *nongrasping* of the tool using the same motions, the grasping signal and image data, such as the movement of the object and tools, are different. As a result, the body modulation module learns the

differences caused by the grasping actions with the tool, that is, the extra context (C_e) nodes learn body modulation timing (or grasping timing) and tool function. Therefore, after training, the $C_e(0)$ space that represents the grasping state and tool function is formed. The number of the extra-context (C_e) nodes is much fewer than that required for the body model (in this study, the number of the extra-context (C_e) nodes is five and the body model requires 74 nodes.). The extra-context (C_e) nodes do not represent the complicated methods of tool use, but only represent the manner in which the original body model is modulated. Because the tool is treated as a part of the body, I expect the robot to use tools based on the experience of moving the body. Therefore, even if the number of tools is increased, the number of neurons of the extra-context (C_e) nodes will be fewer than that required for the body model. The calculation time required to learn the extra-context (C_e) nodes is shorter than that required for the body model.

I used context nodes with *finite* values of time constants instead of parametric bias (PB) nodes, which have *infinite* values of time constants. This method is important and differs from a previous tool-body assimilation model [29]. *Infinite* values of time constants imply that the values of PB nodes do not change during the sequence. Therefore, PB nodes cannot represent the change from the nongrasping to grasping state. With the tool not pre-attached to the hand, it is possible for the extra-context (C_e) nodes to consider grasping decisions and positions during tool use. A comparison of the proposed model with a previous tool-body assimilation model that uses PB nodes [29] instead of extra-context (C_e) nodes is discussed in section 3.5.2. The extra-context (C_e) nodes apply bias to the body model based on the learned grasping state and tool features, and change the dynamics of the body model accordingly. The value of the extra-context (C_e) nodes is calculated using the same method as that in (2.6), while the weights from the extra-context (C_e) nodes to other nodes are updated in the same manner as that for other weights of the MTRNN in (2.5).

3.2.3 Motion Generation from the Initial State and Target Image

The robot generates motion from the initial state (image, joint angles, and grasping state) and a target image provided to the robot (Figure 3.2-(c)). As a feature of the MTRNN, the $C_s(0)$ of the network after training represents each learned motion in the $C_s(0)$ space, that is, the sequence can be recovered using appropriate $C_s(0)$ and initial state (image, joint angles, and grasping state) to substitute into the MTRNN in (4) to (6). In this research, the robot determines appropriate $C_s(0)$ and $C_e(0)$ to be close to the initial state and target image to associate joint angles, image features, and grasping state. To calculate $C_s(0)$ and $C_e(0)$, the weights of the body model and body modulation module are fixed. Moreover, the following equations are applied to calculate the generation error:

$$E = \begin{cases} \sum_{i \in IO} (y_i(t-1) - T_i(t))^2 & t = 0, st \\ 0 & otherwise \end{cases} \quad (3.4)$$

where st represents the steps of the target image.

The $C_s(0)$ and $C_e(0)$ values are computed using (2.6) with the following conditions, in which part of the input-output (IO) nodes can be shown. As only the error of the first and last step is considered using (3.4), the robot associates a motion during the initial state (image, joint angles, and grasping state) and a target image to minimize the error. Note that the association is conducted in the internal model of the robot. Therefore, calculation precision depends on network conditions such as a full teaching signal and appropriate training. The calculation time is shorter than that required to train the network because the weights of the body model and body modulation module are fixed, and only appropriate $C_s(0)$ and $C_e(0)$ are computed.

3.3 Experimental Setup

3.3.1 Simulated Robot Model

In numerical evaluations, I used a simulated robot model based on the structure of the humanoid robot ACTOROID [87] as with section 3.3.1. In the experiment, all joints was stiff instead of flexible joint. To evaluate the proposed method, the movement of the robot is simplified such that it only moves on the plane of a desk (two-dimensional movement). As at least three DOFs are required for any movement on a two-dimensional plane, only three DOFs (joint ID 2, 4, 6 in Figure 2.4) out of the seven DOFs of the robot’s arm were used, and the remaining four DOFs were maintained constant. For motions, the motor torque command to each joint actuator was given by (2.7).

3.3.2 Evaluation Task

To evaluate the effectiveness of the proposed model, I use an object manipulation task, which is commonly used in tool-body assimilation studies [29, 42, 76–79]. The robot conducts motions in the presence of a target object, which is 0.08 m in diameter, and a tool object (the robot’s hand and I-, T-, L-, reverse L-, J-, “|–”, and C-shaped tools; Figure 3.3) on a table. The L- and reverse L-shaped tools are treated as unknown tools, which are considerably similar to the learned tools (I- and T-shaped tools), and the J-, “|–”, and C-shaped tools, which have several dissimilarities with the learned tools, are used only for evaluation, and not for training. There is friction between the target object, tool object, and table.

3.3.3 Procedure of Motor Babbling

To perform motor babbling for teaching data, the initial position of the arm and object, and several desired joint angles are provided for simplicity in result analysis. First, the initial positions of the arm and objects are set. The target object has four initial locations (Figure 3.4-(a)). Next, a sequence of three desired joint angles (Figure 3.4-(b) to (d)) are provided as key poses for each joint of the arm (three

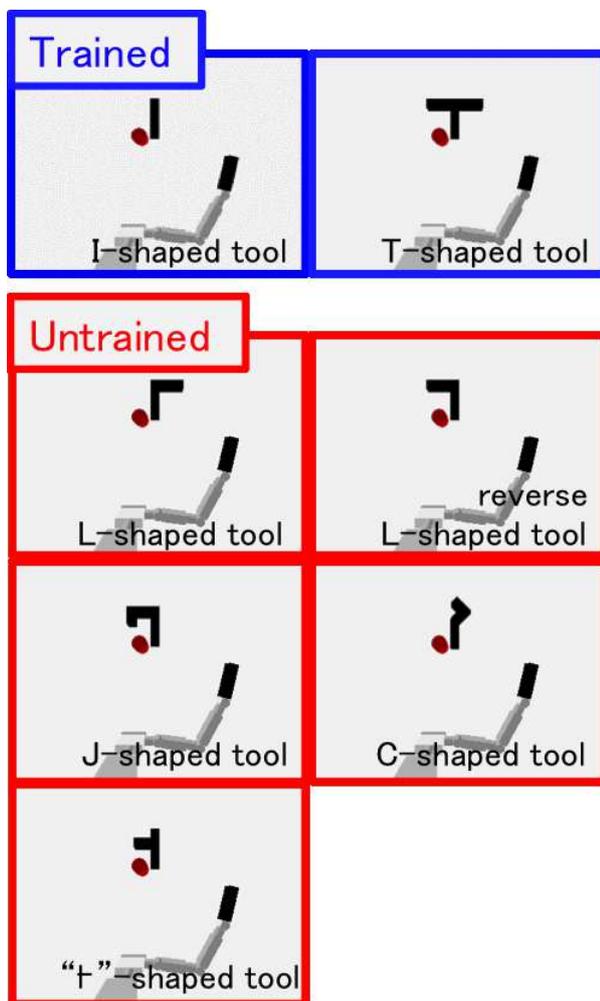


Figure 3.3: Tools used in experiment. I- and T-shaped tools are used as training data. The L- and reverse L-shaped tools are untrained tools considerably similar to the learned tools, and J-, “†”, and C-shaped tools have several dissimilarities with the learned tools.

DOFs) for a 6.0 s duration, that is, the robot performs motion slowly from the four initial locations, (a), to (b) \rightarrow (c) \rightarrow (d). The first desired joint angle has four arm poses for touching the tool, indicating that the robot is able to grasp the tool, and the grasping positions are determined by the first key pose (Figure 3.4-(b)). In the second key pose, the robot stretches the arm to maximize its reach while swinging its arm to the left or right (Figure 3.4-(c)). In the third set of key poses, the robot performs either the *swinging* or the *pulling* motion (Figure 3.4-(d)). Depending on the combination of the key poses, tools, and object positions, the robot either succeeds or fails to pull the object toward itself, and learns these different outcomes as tool functions. The robot performs 128 motion sets per tool ((a) four initial locations, (b) four first sets of key poses, (c) two second sets of key poses, (d) two third sets of key poses, and considering *grasping* and *nongrasping*, that is, $4 \times 4 \times 2 \times 2 \times 2$ motion sets). In a set of 64 motions, the robot grasps the tool upon touching it, that is, the tool is attached to the body and once the robot grasps the tool, the robot never releases it; whereas in the other set of 64 motions, the robot treats the tool as object such as push and pull without attach the object to the body even though the robot contacts it. Additionally, the robot performs motions for the I- and T-shaped tools. As a result, the total number of motion sets for the teaching data is 256.

During motor babbling, the robot obtains the teaching data used during training. The acquired data is composed of joint angle, image, and grasping signal sequential data as follows:

1. Joint angle data: The motor data of the three movable DOFs is recorded for 30 steps during the 6.0 s of motor babbling (sampling interval is 0.2 s). This data is then scaled to [0.1, 0.9] to be used as the teaching data in the MTRNN.
2. Image data: Image data is acquired using a camera set affixed above the robot. This camera has a field of vision that is sufficient for capturing the task area (the desk). Similar to the joint angle data, the image data is recorded for 30 steps. This image data (320×240 pixels) is then reduced to 32×24 pixels (minimum size to distinguish the object) to reduce the calculation cost in this experiment, and the image features are extracted through the DNN. This results

in a 15-dimension image feature data, scaled to $[0.1, 0.9]$.

3. Grasping signal: A grasping signal scaled to $[0.4, 0.6]$ is also recorded during motor babbling. The range of grasping signal is decided not to change too large to train stability. When the robot does not grasp the tool, the signal remains low (0.4), whereas it increases (0.6) after the tool is grasped. The signal is a command from the robot.

These data are then input into the DNN and MTRNN to train the body model and body modulation modules. Table 3.1 presents the specifications of the DNN used in this study. Note that the n -th and $n+1$ -th layer neurons are fully connected. Table 3.2 presents the specifications of the MTRNN used in this study. Note that the input-output (IO) nodes are connected only to themselves and the fast-context (C_f) nodes. Furthermore, the fast-context (C_f) nodes are fully connected to all nodes, whereas the slow-context (C_s) nodes are only connected to themselves and the fast-context (C_f) nodes. Moreover, the extra-context (C_e) nodes are connected unidirectionally to themselves and the input-output (IO), fast-context (C_f), and slow-context (C_s), nodes (Figure 2.3).

Table 3.1: Design of DNN

Dimensions of IO Nodes	768
Number of Hidden Layers	11
Dimensions of Hidden Nodes	500-250-100-50-25-15-25-50-100-250-500
Number of Training Data	7680

3.4 Numerical Simulation Results

In this section, the internal representations of the robot body model and body modulation modules are described. The observation results show the formation of clusters depending on different states of the data. For example, different clusters are formed depending on the grasping positions, grasping state, and tool functions. This information is not clearly defined in the data, that is, there are no labels; therefore, if

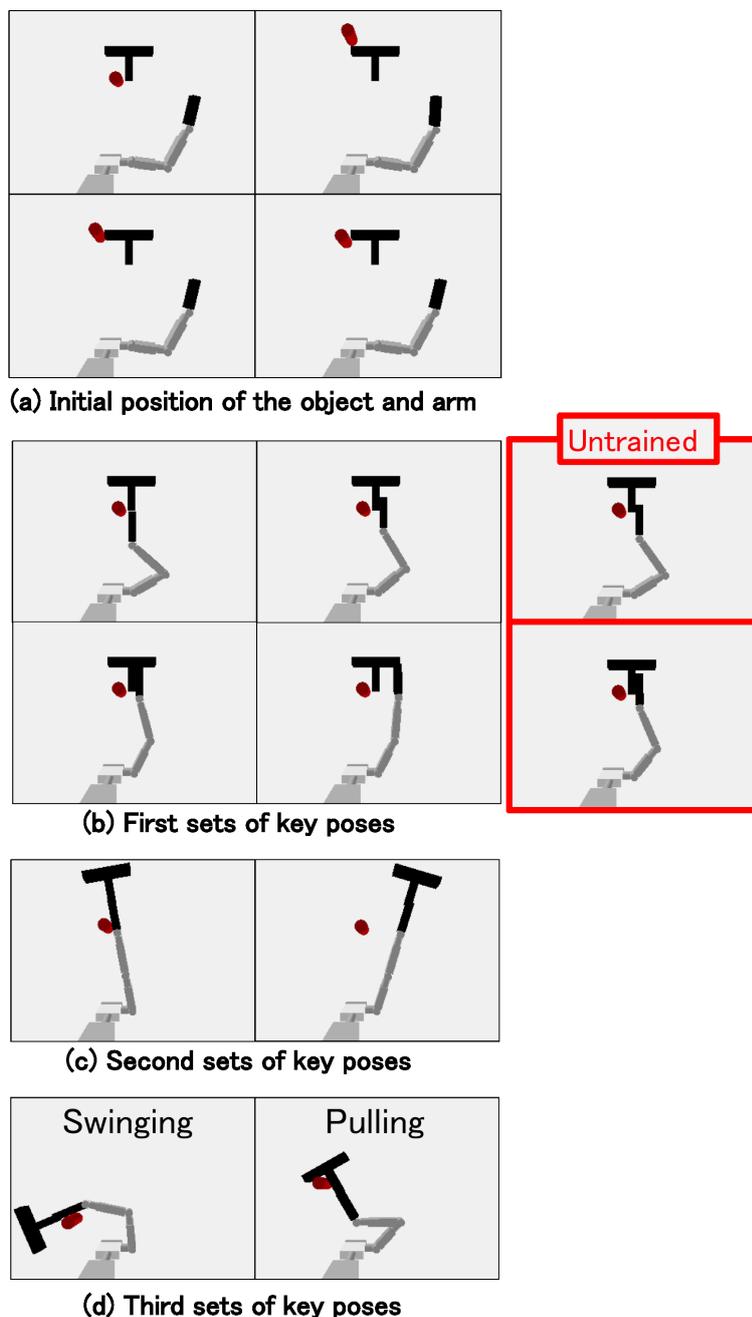


Figure 3.4: Initial position of the arm and object, and first, second, and third sets of key poses (example with T-shaped tool. The motion with I-shaped tool is the same as that with T-shaped tool.). The robot performs motion from four initial locations, (a), to (b) \rightarrow (c) \rightarrow (d), i.e., 128 motion sets per tool ((a)four initial locations, (b) four first sets of key poses, (c) two second sets of key poses, (d) two third sets of key poses, and considering *grasping* and *nongrasping*, that is, $4 \times 4 \times 2 \times 2 \times 2$ motion sets). In a set of 64 motions, the robot grasps the tool upon touching it, whereas in the other set of 64 motions, the robot pushes the tool away upon contact. Additionally, the robot performs motions for the I- and T-shaped tools. As a result, the total number of motion sets for teaching data is 256.

Table 3.2: Design of MTRNN. Each type of node has different time constants. The structure of the MTRNN is depicted in Figure 2.3

Type of nodes	Node name	No. of nodes	Time constant τ
Input-output (IO)	Joint angle input	3	2
	Image feature input	15	2
	Grasping state input	1	2
Fast Context (C_f)	Fast context	50	5
Slow Context (C_s)	Slow context	5	20
Extra Context (C_e)	Extra context	5	20

clusters are formed, it can be said that the robot is able to understand different data features. These data features can be reused later for unknown situations.

Table 3.3: Representation of main components of PCA

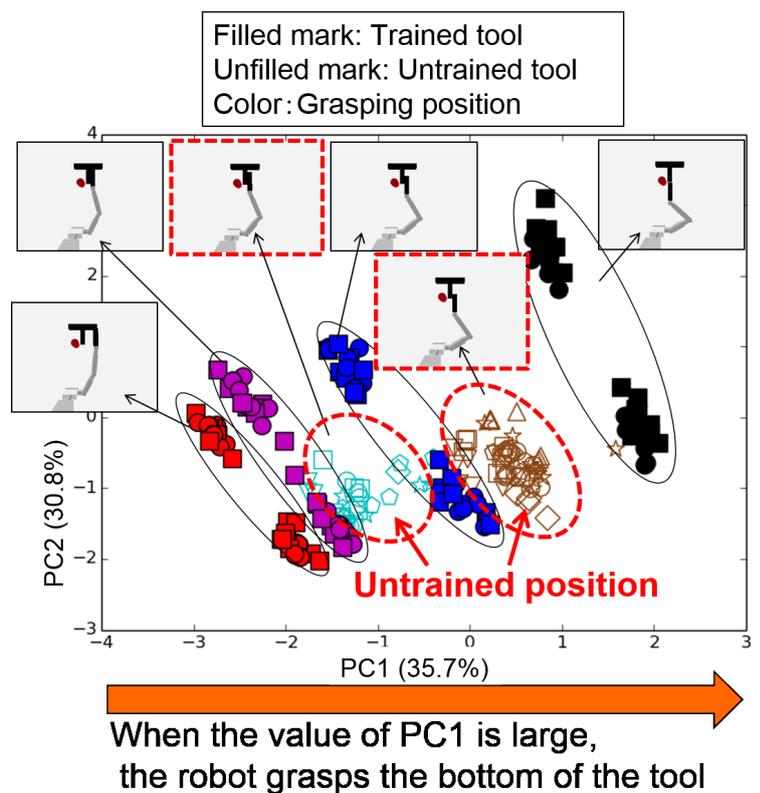
Target of analysis	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
Body model module ($C_s(0)$)	Grasping position	Motion	Motion	-	-	-	-	-	-
Body modulation module ($C_e(0)$)	Grasping state	Mix of information	Tool function	Grasping state and tool function	-	-	-	-	-
Image feature extraction (DNN)	Grasping position and motion	Motion	Grasping position and motion	Motion	Motion	Grasping state and grasping position	Grasping position	Motion and tool function	Motion and tool function

3.4.1 Grasping Positions of Tools

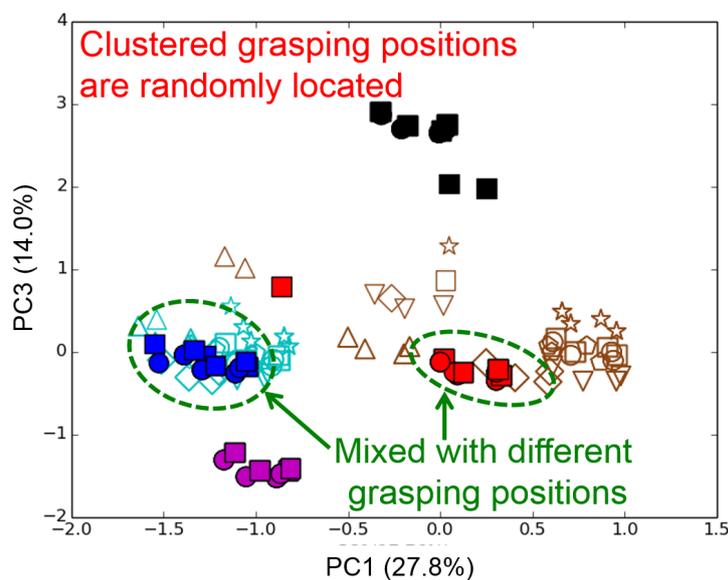
3.4.1.1 Representation of Grasping Positions on the Initial Values of Slow Context (C_s) Nodes of Body Model

To evaluate the representation of grasping positions on body model, I have the robot recognize the grasping position only from the provided initial state (image, joint angles, and grasping state) and target image in which the robot's arm contacts the tool to calculate (3.4) ($st = 8$) for trained and untrained tools and grasping positions. Figure 3.5-(a) shows the plots of the principal component analysis (PCA) of $C_s(0)$ of the body model with trained and untrained tools and grasping positions. In this experimental setting, the grasping position depends on the first set of key poses (Figure 3.4). Clusters are formed for the trained and untrained tools and grasping positions. Clustered grasping positions are relative to the grasping positions. This implies that when the value of PC1 is large, the robot grasps the bottom of the tool. Therefore, the robot is said to be able to recognize the grasping positions.

Figure 3.5-(b) shows the plots of the PCA of the DNN with image extraction. Table 3.3 shows the representation of the main components of the PCA. To focus on grasping positions, PC1 and PC3 are plotted in Figure 3.5-(b). The grasping positions for trained data appear to be clustered. When the grasping position is the same with the different tool, image data would be similar and it appears that image features are also similar; however, those for untrained data are mixed. In addition, the clustered grasping positions are irregularly located compared to those in Figure 3.5-(a). The DNN composed of feed-forward neural network is trained with only image data, whereas the body model is trained with time series images, joint angles, and grasping state using RNN with recurrent connection. Thus, I believe that time-continuous images and joint angles using neural networks with recurrent connection is necessary for well-clustered grasping positions.



(a) PCA of $C_s(0)$ of body model module



(b) PCA of DNN with image extraction

Figure 3.5: Representation of grasping position in PCA of (a) $C_s(0)$ of body model and (b) DNN with image extraction.

3.4.1.2 Relationship between the Object Position and Grasping Position of Tool

To evaluate the relationship between the object position and grasping position of the tool, I have the robot recognize the grasping position from the provided initial state (image, joint angles, and grasping state) and target image in the case of two different object positions (the object is *close to* and *away from* the robot) with trained and untrained tools (Figure 3.6) to calculate (3.4) ($st = 30$), and then generate motion using the calculated $C_s(0)$ and $C_e(0)$. The translucent image shows the initial state, and the clear image shows the target image in Figure 3.6. The image is used for describing the setup, and not for motion generation. To prevent determining the grasping position directly from the target image, it is shown without the tool. As a result, the robot grasps the top of the tool when the object position is close to the robot, whereas it grasps the bottom of the tool even though untrained tool.

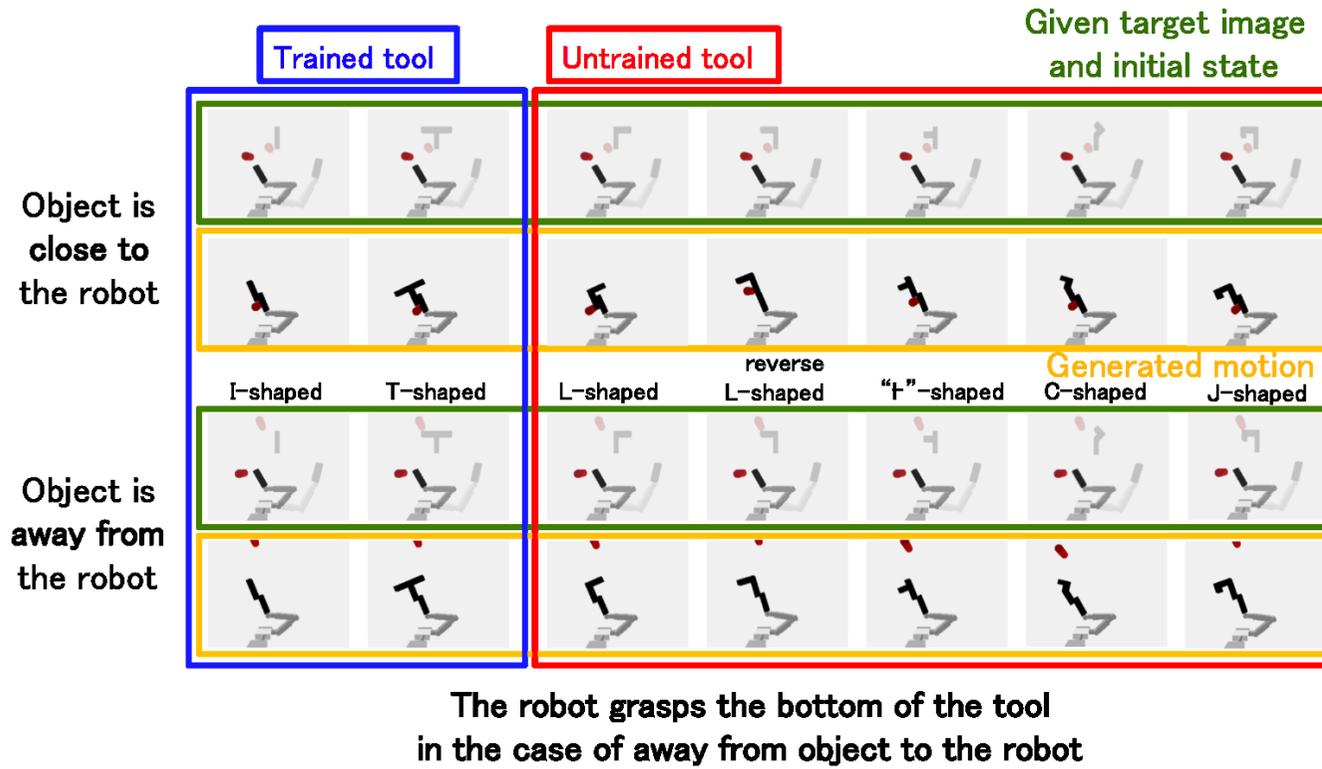
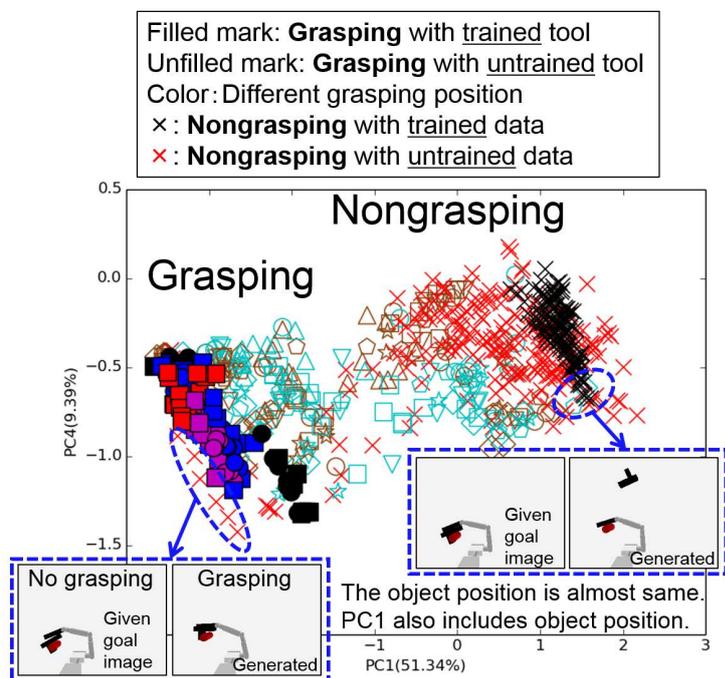


Figure 3.6: Relationship between the object position and grasping position of tool. Translucent image shows initial state and clear image shows the target image. The image is used for describing the setup, and not for motion generation.

3.4.2 Representation of Grasping State on Initial Values of Extra-Context (C_e) Nodes of Body Modulation

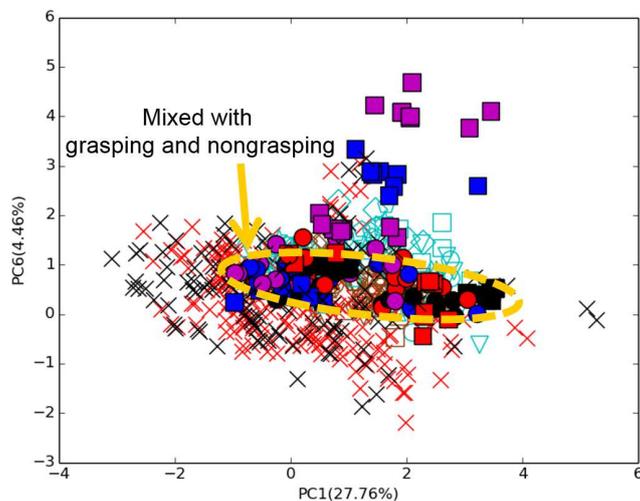
To evaluate the representation of grasping state on body modulation, I have the robot recognize the grasping state only from the provided initial state (image, joint angles, and grasping state) and target image to calculate (3.4) ($st = 30$). Figure 3.7-(a) shows the plots of the PCA of $C_e(0)$ of body modulation. To focus on the grasping state, PC1 and PC4 are plotted in Figure 3.7-(a) (Table 3.3). The clusters of *grasping* and *nongrasping* data are formed for the trained and untrained tools. This implies that the robot can recognize the grasping state from the target image. A few of the plots are incorrectly recognized in Figure 3.7-(a). In the cases, the nongrasping plot is in the grasping cluster. This is because the given target image appears to be a grasping image (the robot’s arm touches tool without grasping). Therefore, the robot recognizes this as a grasping tool, even though it does not actually grasp the tool. I did not provide any labels such as *grasping* and *nongrasping* during training. It is important how robots define grasping, and not how I define it. In other cases, the grasping plot is in the nongrasping cluster. This is because the object position after motion is almost the same for the given target image and generated motion. Therefore, it appears that PC1 includes the object position. The result shows that the robot can decide not to grasp the tool to move the object to the target position, even though the robot grasps the tool in the provided target image. Thus, the robot is said to be able to decide the grasping state.

Figure 3.7-(b) shows the plots of the PCA of the DNN with image extraction. To focus on the grasping state, PC1 and PC6 are plotted in Figure 3.7-(b) (Table 3.3). To compare with the PCA of $C_e(0)$ of body modulation, the grasping state is not clustered for trained and untrained data. The DNN composed of feed-forward neural network is trained with only image data, whereas the extra-context (C_e) nodes are trained with a continuous image, joint angles, and the grasping state using RNN with recurrent connection. Therefore, the robot is able to recognize the grasping state easily with MTRNN, to compare with only the DNN.



Given goal image looks like grasping.
Therefore, the robot recognizes as grasping.

(a) PCA of $C_e(0)$ of body modulation module



(b) PCA of DNN with image extraction

Figure 3.7: Representation of grasping state in PCA of (a) $C_e(0)$ of body modulation module and (b) DNN with image extraction.

3.4.3 Representation of Body Modulation on Slow-Context (C_s) Nodes of Body Model

To evaluate the body modulation in the body model by the body modulation module, I have the robot recognize grasping state only from the provided initial state (image, joint angles, and grasping state) and target image to calculate (3.4) ($st = 30$), and then generate motion using the calculated $C_s(0)$ and $C_e(0)$. Figure 3.8 shows the plots of the PCA of C_s of the body model with time steps. For better visibility, two motions are shown in Figure 3.8. To focus on body modulation, PC2 and PC3 are plotted in Figure 3.8 (Table 3.3). In Figure 3.8, the *grasping* and *nongrasping* motions branch off after contact between the robot hand and tool (approximately 8 steps) in each motion. Therefore, it can be said that the robot modulates the body after grasping the tool.

3.4.4 Representation of Tool Function on Initial Values of Extra-Context (C_e) Nodes of Body Modulation

To evaluate the representation of tool function on body modulation, I have the robot recognize tool function only from the provided initial state (image, joint angles, and grasping state) and target image to calculate (3.4) ($st = 30$), and then generate motion using the calculated $C_s(0)$ and $C_e(0)$. Figure 3.9-(a) shows the plots of the PCA of $C_e(0)$ of body modulation. To focus on tool function, PC3 and PC4 are plotted in Figure 3.9-(a) (Table 3.3). Figure 3.9-(a)-(left) includes the failed motion of pulling and swinging (see Figure 3.4) the object. This indicates that the robot performed pulling and swinging motions; however, it could not move the object to the correct position (the object is away from the tool). In this case, the tool function is not clustered. The failed motions are removed from Figure 3.9-(a)-(left), and Figure 3.9-(a)-(right) is composed of the successful motions of pulling and swinging the object. Thus, the tool functions are clustered. This result includes the untrained tool in addition to the trained tool. Tool function is not only recognized by the joint angles as pulling or swinging, but also by the object position. From the above, the robot can be said to recognize the tool functions.

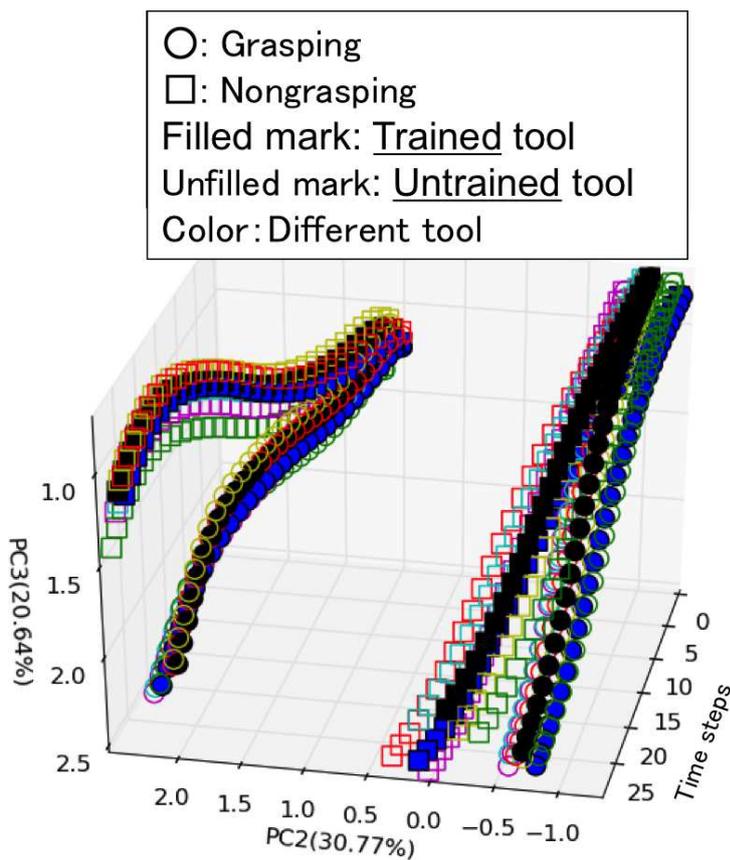


Figure 3.8: Representation of body modulation by body modulation module in PCA of time steps C_s of body model. Two motions are selected for better visibility.

Figure 3.9-(b) shows the plots of the PCA of the DNN with image extraction. To focus on tool function, PC8 and PC9 are plotted in Figure 3.9-(b) (Table 3.3). After removing the failed motions from Figure 3.9-(b)-(left), Figure 3.9-(b)-(right) is composed of the successful motions of pulling and swinging the object. Even though fail motion is included in Figure 3.9-(b)-(left), it seems that the clusters of pulling and swinging motion are almost formed. Therefore, PC8 and PC9 almost represent motion, that is, when the motion is the same with the different tool, image data would be similar and it appears that image features are also similar, that is, the object is ignored. When the contribution ratios in Figure 3.9-(b) are compared with those in Figure 3.9-(a), the contribution ratios of the DNN are comparatively smaller than those of the body model module. The body model modules are trained with continuous images, joint angles, and grasping state. Therefore, the robot can recognize the tool function easily with MTRNN, to compare with only the DNN.

To evaluate the ease in movement according to tool shape, the success rate of each motion (pulling or swinging) with each tool is calculated to count the number of motions in which the object was successfully moved in Figure 3.9-(a). The number of the each motion with each tool is $32 ((a)4 \times (b)4 \times (c)2 \times (d)1)$ (pulling or swinging) (Figure 3.10). The success rate of motion is higher for tools with protrusions on the left side than for those without it, such as the I- and L-shaped tools. These tools are more helpful for swinging an object close to the robot. The protrusion on the left side is required to perform the pulling function successfully. The C-shaped tool is longer than other tools, as a result, it is suitable for pulling an object close to the robot.

Figure 3.11 shows the plots of the PCA of the center of gravity of $C_e(0)$ of the body modulation of each tool. The larger values on the PC3 axis and smaller values on the PC4 axis represent the T-shaped tool, whereas the smaller values on the PC3 axis and larger values on the PC4 axis represent the I-shaped tool. Therefore, the C-shaped tool with a small protrusion is plotted at the smaller values on the PC3 axis and larger values on the PC4 axis. Based on this, the robot can be said to recognize tool functions.

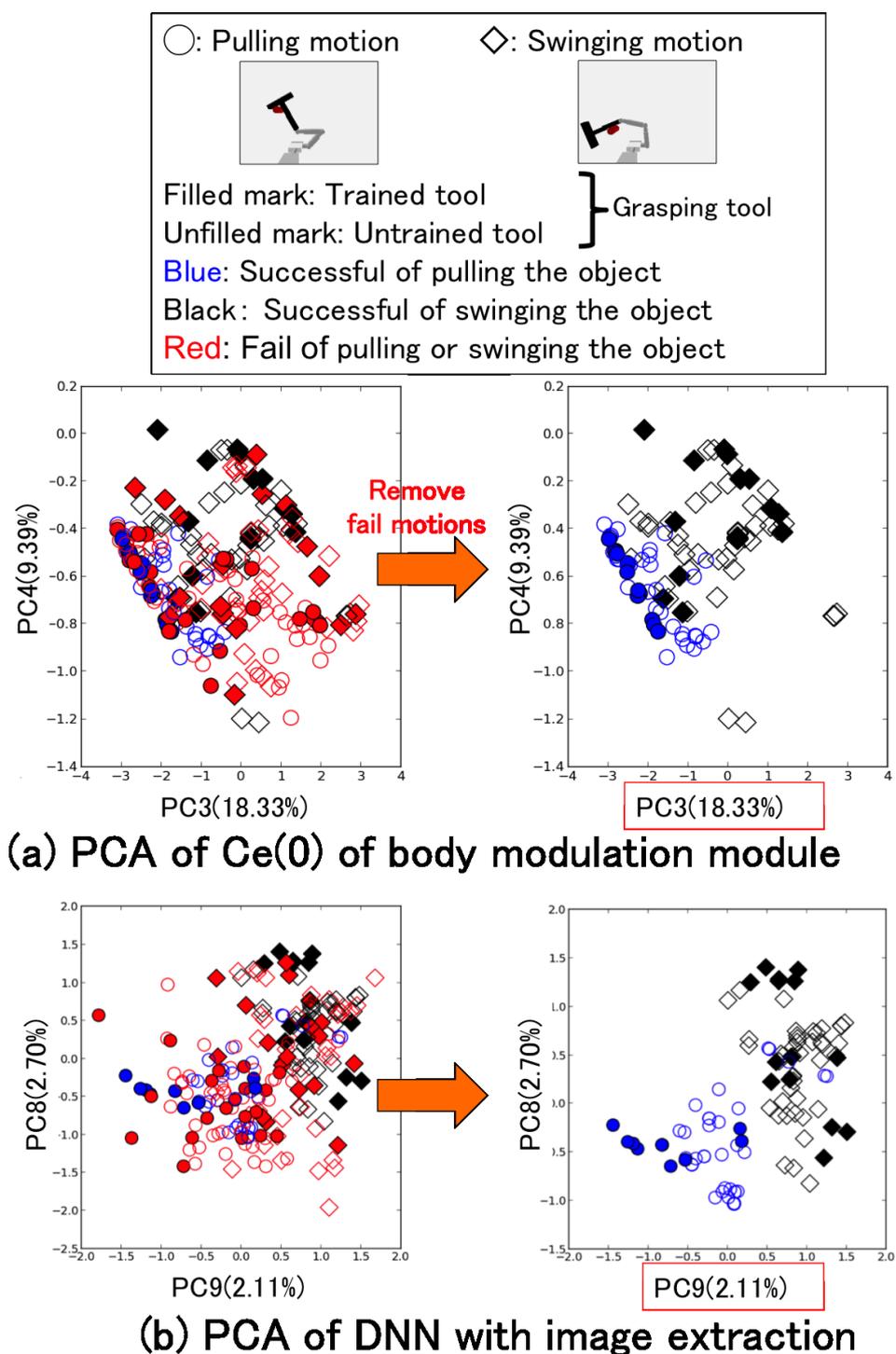


Figure 3.9: Representation of tool function for pulling and swinging in PCA of (a) $C_e(0)$ of body modulation model and (b) DNN with image extraction. The percentages of axes are contribution ratios to the total amount of information of an eigenvalue. The left figure includes the failed motions of pulling and swinging the object. This indicates that the robot performed pulling or swinging motions; however, it could not move the object correctly. The right figure comprises of the successful motions of pulling and swinging the object.

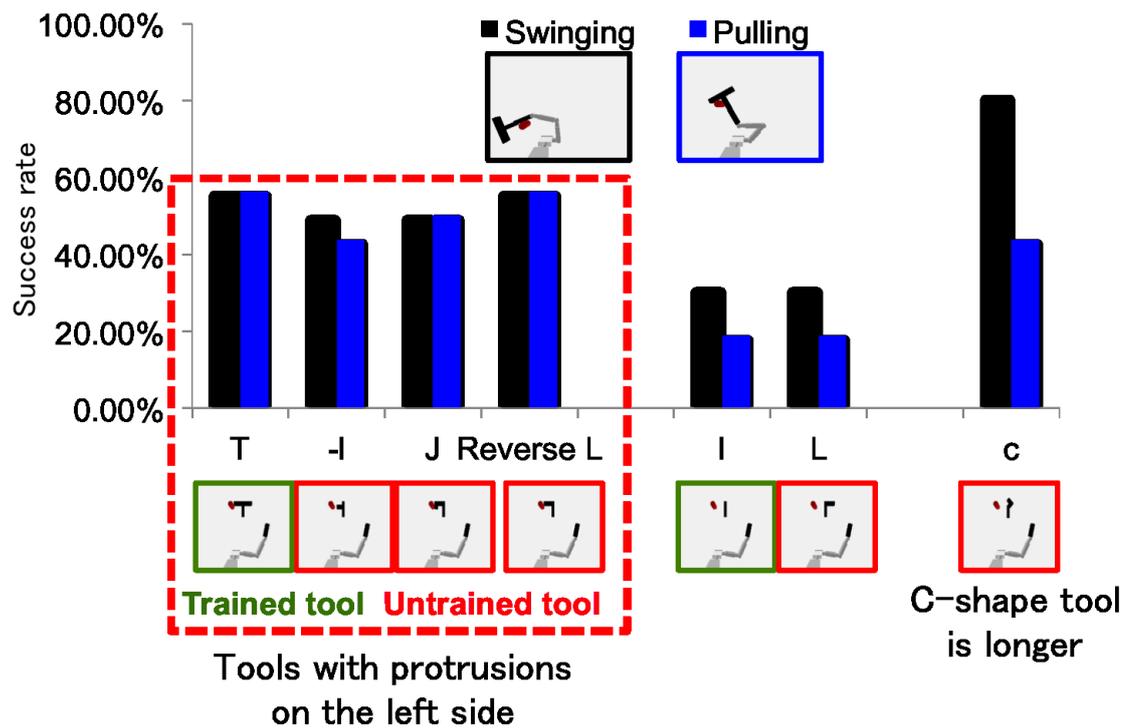


Figure 3.10: Success rate of each tool for pulling and swinging in Figure 3.9-(a). The success rate is calculated to count the number of motions in which the object was successfully moved.

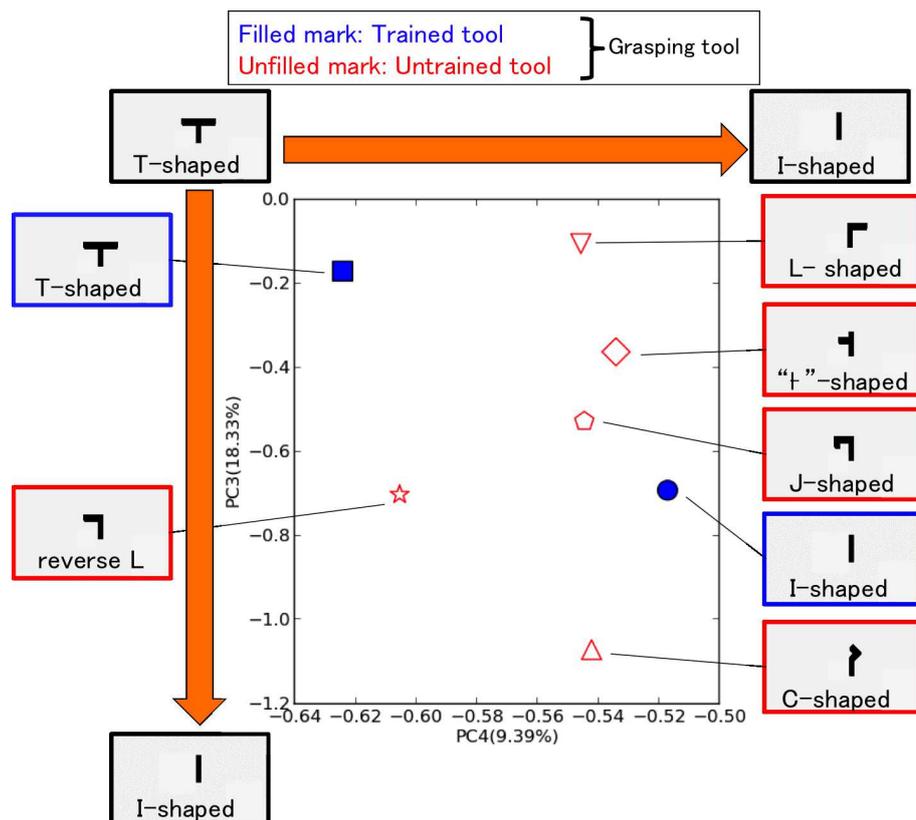


Figure 3.11: Representation of all tool functions in PCA of the center of gravity of $C_e(0)$ of body modulation of each tool. The larger value on the PC3 axis and smaller values on the PC4 axis represent the T-shaped tool, whereas the smaller values on the PC3 axis and larger values on the PC4 axis represent the I-shaped tool.

3.4.5 Motion Generation

To evaluate the performance, I have the robot generate motion only from the provided initial state (image, joint angles, and grasping state) and target image to calculate (3.4) ($st = 30$) and then generate motion using the calculated $C_s(0)$ and $C_e(0)$. For the evaluation, the robot recognizes the trained grasping motions with trained and untrained tools (Figures 3.3 and 3.4). This implies that 64 motions ($(a)4 \times (b)4 \times (c)2 \times (d)2$ in Figure 3.4) are tested for each tool. I evaluated the performance of the proposed system by counting the number of tasks in which the object was successfully moved to a position within n times of the object's diameter d ($d = 8\text{cm}$) from the target position in the visible area (32×24 pixels). Figure 3.12 shows the relationships between the success rate and d . The success rate within d was approximately 20–35%; however, the rate within $2d$ was more than 50% even when the robot used the untrained tools. The success rate appears to be low for precise control. This is because the object size is 1 to 2 pixels in the image, that is, the object size is considerably smaller than the robot size. In addition, in this experiment, the robot attempts to perform motion to close to the target image, considering the robot posture in addition to the position of target object. There is no difference between the trained and untrained tools with respect to success rate because the DNN could extract good image features, such as tool shape, as a generalization, and the RNN could acquire a good sensorimotor relationship.

Regarding the evaluation of the generalization ability of the DNN, Figure 3.13 shows the original images and recovered images to substitute the image using the trained DNN. The images of the tools and the target object were recovered. Moreover, even with unknown tools, it was possible to recover the shape of the tools.

3.5 Discussion

3.5.1 Comparison with Self-organizing Map

In previous studies, self-organizing map (SOM) has been used for the extraction of features [78,104]. Arie et al. proved that SOM is the high compatibility with MTRNN.

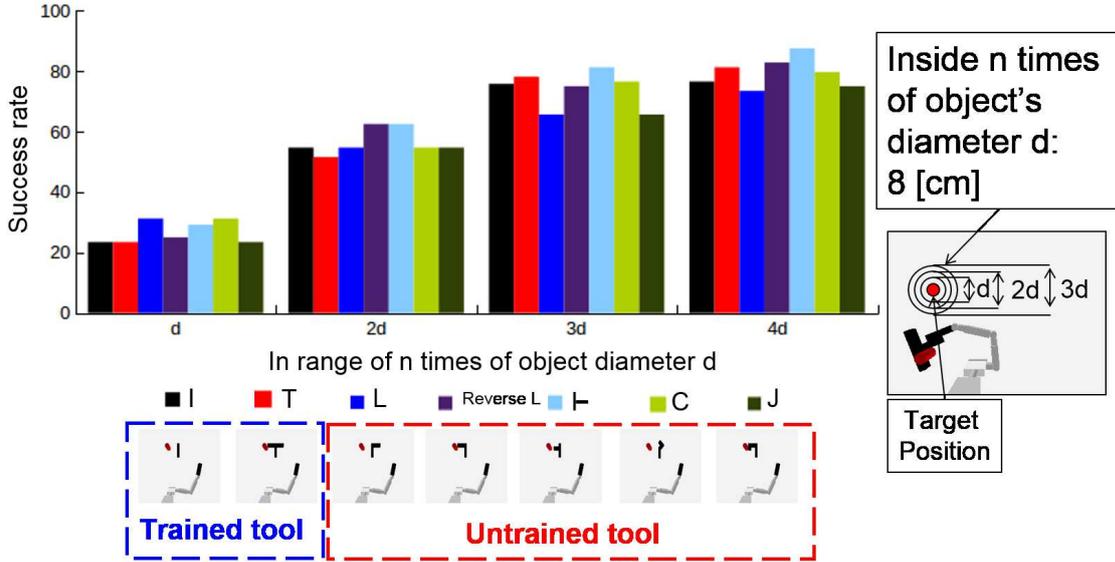


Figure 3.12: Success rates of manipulating objects (Test with unknown tool). The success rate is calculated such that the object position is within n times of object's diameter d ($d = 8\text{cm}$) from the target position in the visible area (32×24 pixels).

SOM is an unsupervised learning neural network proposed by Kohonen [105]. It is composed of input and output neurons. The neurons in the output layers are two-dimensionally arranged and possess weight vectors, w . The weight vectors are set to have the same dimensions (I) as the input vector, v . The image features are defined by the following formula:

$$p_i = \frac{\exp \left\{ -\frac{\|w_i - v\|^2}{\sigma} \right\}}{\sum_{j \in N} \exp \left\{ -\frac{\|w_j - v\|^2}{\sigma} \right\}}, \quad (3.5)$$

where N is the dimension of the SOM and $i \in I$.

It is possible to reduce the dimensionality of data by using an SOM. However, if there are many motion patterns for the robot, it is difficult to extract features using an SOM. Therefore, I used an auto-encoder with the DNN for the extraction of features.

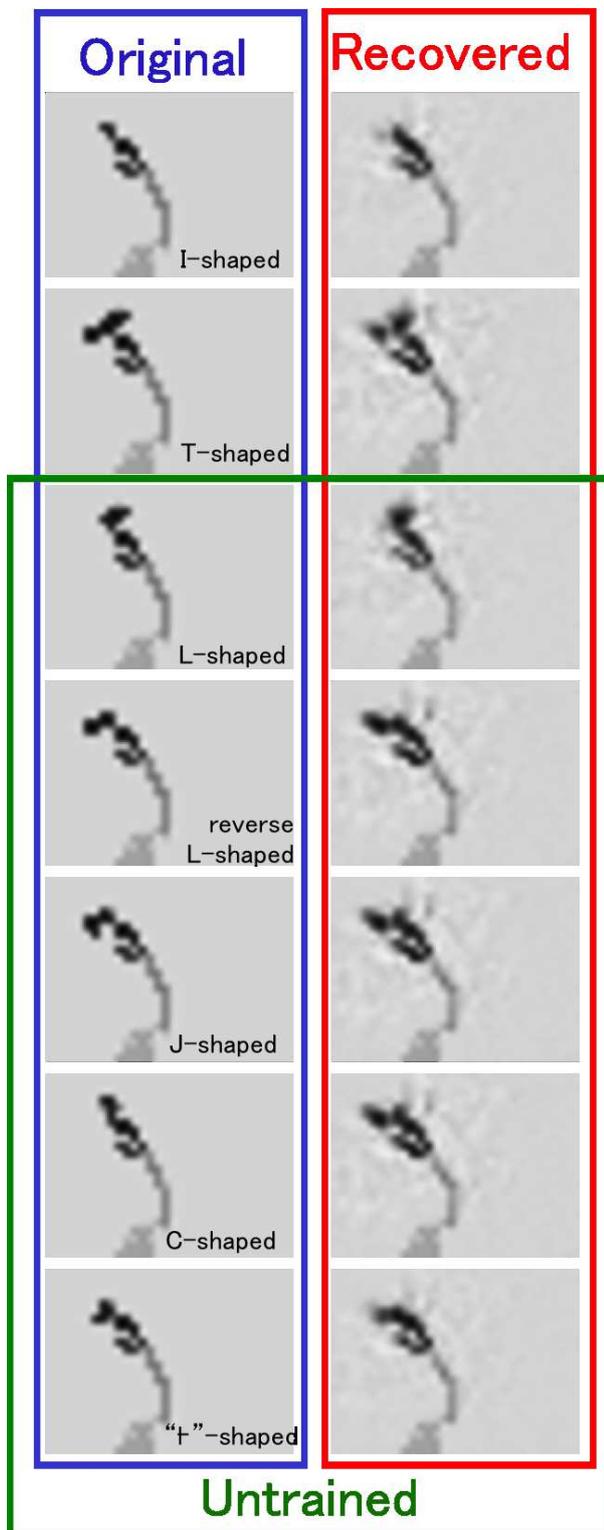


Figure 3.13: Original images and recovered images using the trained DNN

3.5.1.1 Extraction of Image Features by Self-organizing Map

In previous studies, SOM has commonly been used as an image feature extraction method [78, 104, 106, 107]. To compare image features extraction by DNN, image features were also extracted by SOM. Figure 3.14 shows a visualization of the reference vectors of the SOM. Reference vectors are the visualization of image features. Reference vectors represent the patterns that are extracted from the input data. The characteristics of reference vectors are that the units that are mapped close to each other will have close resemblances to each other. In addition, each input data is classified to the locations of the reference vectors that are similar to the data. The dimensions of this SOM were 5×5 . The results show that the difference between the bare hand and tools is not learned accurately and that the motion patterns are not learned accurately. I changed the dimensions of SOM to 10×10 . However, the motion patterns were not learned accurately after increasing the SOM dimension. Even if feature extraction is done well by increasing the dimension, it is difficult to learn by RNN because of the greater dimension. When more tools are introduced, the various tool conditions were included in each vector, causing the tool feature classification to fail. When only a few tools are used, it is possible to learn the image features accurately with SOM. This is shown in Figure 3.15 where image features of the bare hand and T-shaped tool were extracted by SOM.

3.5.1.2 Extraction of Image Features by DNN

Original images were recovered by substituting the image features extracted by DNN (Figure 3.16). The image of the bare hand and tools were accurately recovered. In addition, the position of the target object was recovered. Moreover, even with unknown tools it was possible to recover the shapes of the unknown tools.

In the case of the SOM, extracted features of the target object were unclear; therefore, it was difficult to recover the position of the object accurately. DNN does not use classifications, but instead makes use of auto-encoders that are trained to produce the same output as the input data. With this, it is possible to reduce the



Figure 3.14: Reference vector of SOM



Figure 3.15: Reference vector of SOM (bare hand and T-shaped tool only)

dimensionality of large numbers of high-dimensional data followed by high reproduction performance. In addition, there is no need to fine tune the parameter settings with DNN. Thus, if there are large amounts of training data, the DNN is superior to the SOM in the extraction of image features.

3.5.2 Comparison with Tool-body Assimilation Model Using PB nodes

The tool-body assimilation model with the MTRNN and PB [29] instead of extra-context (C_e) nodes was tested to determine whether the model is capable of learning different grasping decisions. Each experiment was conducted three times using the randomly set initial values of the weights and context in the MTRNN. Next, the average values were calculated. Here, the MTRNN acted as the body model, while PB learned the different grasping decisions. The training of the body modulation module failed because the grasping signal error remained high (Figure 3.17). This is because PB, with *infinite* values of time constants, can provide only a constant modulation effect to the body. It is difficult to express the transition between two grasping states (nongrasping to grasping).

3.6 Summary

In this chapter, I proposed a tool-body assimilation model that considers grasping using deep learning. Previous tool-body assimilation models [29] required the robot to have the tool pre-attached to its hand before the motion started. This leads to problems in which the robot cannot make decisions about grasping state (*grasping* or *nongrasping*) and positions. This is because *grasping* is not the only option for using tools, and tool functions change depending on the grasping positions. To overcome these issues, in this study, the robot started its motions without the tool being pre-attached. The robot then performed motor babbling by *grasping* and *nongrasping* the tool in the process to learn its body model and tool functions. Next, the robot grasped different parts of the tool, learning the different tool functions of the different grasping

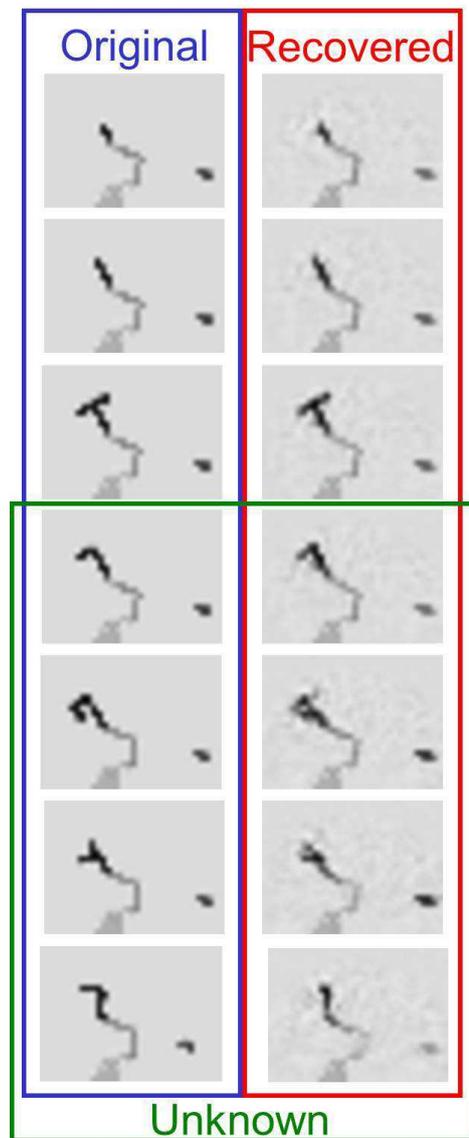


Figure 3.16: Original images and recovered images using the trained DNN (with PB)

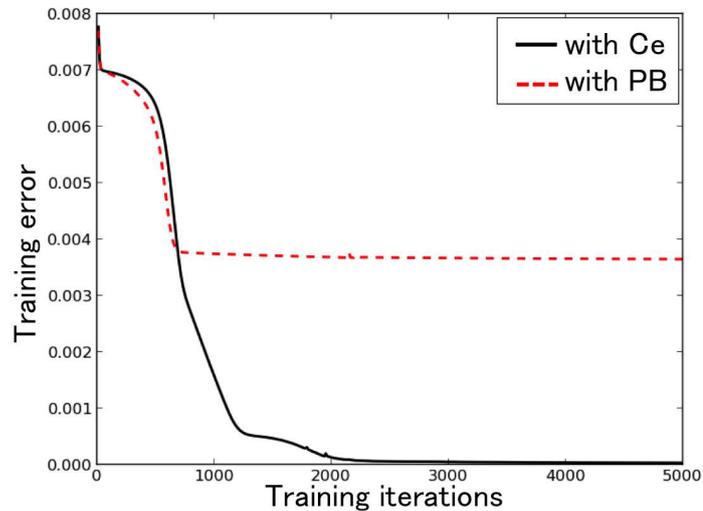


Figure 3.17: Training error for acquisition of body model with extra-context C_e nodes and PB nodes

positions. To encode the change in the body model after grasping, a body modulation module was attached to the body model. Instead of the PB nodes with *infinite* values of time constants (state does not change), which were used in previous models, context nodes with *finite* values of time constants (state is capable of changing) were used so that the nodes could provide different modulation effects to the body during motion. After training the tool-body assimilation model, the system was evaluated using a robotics simulator. Furthermore, the generalization performances of tool types and grasping positions were evaluated. During the evaluations, the robot was shown the initial states and target images, that is, the robot considered the required motions, grasping decisions, and grasping positions to reproduce the target images. Thus, the system exhibited generalization abilities with respect to tool types and grasping positions.

Chapter 4

Conclusion

4.1 Overall Summary of the Current Research

This paper proposes a machine learning framework for the acquisition and extension of a robot body model using motor babbling through deep learning such as RNN and DNN. Existing robots include industry-standard robots used for speed, precision, and cost-effectiveness, and simple robots having a small number of DOFs. The future wave of robots will have a large number of DOFs, various sensors, and flexible joints used for complex tasks considering dynamic motion and safety, and will require the means for accommodating dynamic environments. This research demonstrates the effectiveness of such robots.

For the acquisition of the robot body model, I propose a two-phase learning strategy for robots with flexible joints having multi-DOFs to accomplish dynamic motion tasks. The two-phase learning methodology comprises *body dynamics* of the robot, including a *pre-training* phase with motor babbling and a *fine-tuning* phase with incremental learning of task motions. In the pre-training phase, I employ an RNN considering *active* and *passive* exploratory motions to ensure that the robot efficiently learns body dynamics. In the fine-tuning phase, the learned body dynamics are adjusted for specific tasks. The proposed method is analogous to the successful learning strategy for DNNs consisting of *pre-training* and *fine-tuning*. In the pre-training

phase of DNNs, the parameters of each layer are initialized to obtain the appropriate values. Then, in the fine-tuning phase of DNNs, the whole network is trained with all layers connected to efficiently learn the target data using the results of the pre-training phase. I demonstrate the effectiveness of the proposed methodology on a simulated flexible joint robot model as well as in hardware experiments using a PR2 robot both of which have a seven DOF redundant arm for dynamic tasks such as crank-turning, door-opening/closing, and drawer-opening/closing involving constrained movement requiring interactions with the environment. The results show a reduction in the required number of training iterations for task and generalization capabilities for untrained situations with the learned body dynamics through motor babbling. In addition, I discuss issues regarding the trade-off between task training iterations and the success rate of task execution. Furthermore, I discuss the explorative motor babbling to acquire body dynamics with a small number of motor babbling movement.

For the extension of the robot body model, I propose a tool-body assimilation model with regard to grasping during motor babbling for using tools. The robot performs motor babbling using *nongrasping* and *grasping* tools required to learn both the body model and tool functions. In addition, the robot grasps various parts of the tools to learn the different tool functions associated with different grasping positions. These motion experiences are learned using a DNN and RNN. Tool features were self-organized in extra-context nodes, and the body model was modulated according to the tool in use. Finally, I designed a neural network for the robot to generate motion only from a target image. To evaluate the model, I had the robot manipulate an object task without any tools or with several tools of different shapes. Moreover, I had the robot generate motions after demonstrating the initial and target states by deciding whether and where to grasp the tool. Through this, the robot showed its ability to generate the correct motion and grasping decisions when the initial and target states were provided to the robot.

The robot demonstrated the following abilities:

- Performing dynamic motions and complex tasks such as contacting the environment using a large number of DOFs and flexible joints

- Adapting to dynamic and uncertain real environments with highly generalized abilities because of the approach without pre-designed models through deep learning
- Coping with new tasks and situations in a short duration using the robot body model through motor babbling

4.2 Future Works

This research demonstrated the effectiveness of motions with sensors using motor babbling. However, humans can acquire a body model with a multi-modal information involving tactile sensation, natural language, and audio through motor babbling. The machine learning framework proposed in this work eliminates the need for a pre-designed robot and environment models; that is, the robot acquires its model through its experiences. It is possible to apply the approach not only for motion but also other modes with integration.

In previous research regarding feature extraction, it was necessary to design the features of each mode by humans. Using deep learning, the internal representation of data is autonomously structured as features from the large amount of data, and accurate results are acquired independent of human-designed features. However, in research using machine learning, such as deep learning, the structure of the network is determined by trial and error. I believe it would be also possible to acquire the structure of the network in training. By doing so, even when new data is added, it would be possible to acquire a suitable network structure by online learning.

Furthermore, in previous studies, features of each modal were acquired, and then combined. However, the features of each modal were not always suitable features when combined with other modals. For example, let me consider the case of a reaching task using the arm of the robot and a camera. In this task, the important factor is the relationship between the desk and the object, and even if the original background is changed, there should be no problem. Thus, it is important to extract features considering multiple modals.

To combine all the above ideas, the robot adapts to the internal and external models through interaction with the environment at all times.

Bibliography

- [1] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, “The development of honda humanoid robot,” in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2, pp. 1321–1326, IEEE, 1998.
- [2] K. Kaneko, K. Harada, F. Kanehiro, G. Miyamori, and K. Akachi, “Humanoid robot hrp-3,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2471–2478, IEEE, 2008.
- [3] H. Iwata and S. Sugano, “Design of human symbiotic robot TWENDY-ONE,” in *IEEE International Conference on Robotics and Automation*, pp. 580–586, 2009.
- [4] C. Pérez-D’Arpino and J. A. Shah, “Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6175–6182, IEEE, 2015.
- [5] M. A. Diftler, J. Mehling, M. E. Abdallah, N. A. Radford, L. B. Bridgwater, A. M. Sanders, R. S. Askew, D. M. Linn, J. D. Yamokoski, F. Permenter, *et al.*, “Robonaut 2—the first humanoid robot in space,” in *IEEE International Conference on Robotics and Automation*, pp. 2178–2183, 2011.
- [6] E. Guizzo and E. Ackerman, “The rise of the robot worker,” *IEEE Spectrum*, vol. 49, no. 10, pp. 34–41, 2012.

- [7] A. Stentz, H. Herman, A. Kelly, E. Meyhofer, G. C. Haynes, D. Stager, B. Zajac, J. A. Bagnell, J. Brindza, C. Dellin, *et al.*, “CHIMP, the CMU highly intelligent mobile platform,” *Journal of Field Robotics*, vol. 32, no. 2, pp. 209–228, 2015.
- [8] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon, “A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955,” *AI magazine*, vol. 27, no. 4, p. 12, 2006.
- [9] J. Haugeland, *Artificial intelligence: The very idea*. MIT press, 1989.
- [10] A. Newell and H. A. Simon, “Computer science as empirical inquiry: Symbols and search,” *Communications of the ACM*, vol. 19, no. 3, pp. 113–126, 1976.
- [11] M. Campbell, A. J. Hoane, and F.-h. Hsu, “Deep blue,” *Artificial intelligence*, vol. 134, no. 1, pp. 57–83, 2002.
- [12] J. Weizenbaum, “Eliza—a computer program for the study of natural language communication between man and machine,” *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.
- [13] E. H. Shortliffe, R. Davis, S. G. Axline, B. G. Buchanan, C. C. Green, and S. N. Cohen, “Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the mycin system,” *Computers and biomedical research*, vol. 8, no. 4, pp. 303–320, 1975.
- [14] D. C. Dennett, “Cognitive wheels: The frame problem of ai,” *Language and Thought*, vol. 3, p. 217, 2005.
- [15] S. Harnad, “The symbol grounding problem,” *Physica D: Nonlinear Phenomena*, vol. 42, no. 1-3, pp. 335–346, 1990.
- [16] R. Brooks, “A robust layered control system for a mobile robot,” *IEEE journal on robotics and automation*, vol. 2, no. 1, pp. 14–23, 1986.
- [17] R. A. Brooks, “Intelligence without representation,” *Artificial intelligence*, vol. 47, no. 1, pp. 139–159, 1991.

- [18] R. A. Brooks, “Intelligence without reason,” *The artificial life route to artificial intelligence: Building embodied, situated agents*, pp. 25–81, 1995.
- [19] R. C. Arkin, *Behavior-based robotics*. MIT press, 1998.
- [20] M. M. Williamson, “Postural primitives: Interactive behavior for a humanoid robot arm,” in *Fourth International Conference on Simulation of Adaptive Behavior*, pp. 124–131, 1996.
- [21] M. M. Williamson, “Neural control of rhythmic arm movements,” *Neural Networks*, vol. 11, no. 7, pp. 1379–1394, 1998.
- [22] R. A. Brooks, “From earwigs to humans,” *Robotics and autonomous systems*, vol. 20, no. 2, pp. 291–304, 1997.
- [23] R. Pfeifer and C. Scheier, *Understanding intelligence*. MIT press, 2001.
- [24] R. Pfeifer, M. Lungarella, and F. Iida, “Self-organization, embodiment, and biologically inspired robotics,” *science*, vol. 318, no. 5853, pp. 1088–1093, 2007.
- [25] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, and C. Yoshida, “Cognitive developmental robotics: a survey,” *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 1, pp. 12–34, 2009.
- [26] A. Cangelosi, M. Schlesinger, and L. B. Smith, *Developmental robotics: From babies to robots*. MIT Press, 2015.
- [27] M. Asada, K. F. MacDorman, H. Ishiguro, and Y. Kuniyoshi, “Cognitive developmental robotics as a new paradigm for the design of humanoid robots,” *Robotics and Autonomous Systems*, vol. 37, no. 2, pp. 185–193, 2001.
- [28] K. Noda, H. Arie, Y. Suga, and T. Ogata, “Multimodal integration learning of robot behavior using deep neural networks,” *Robotics and Autonomous Systems*, vol. 62, no. 6, pp. 721–736, 2014.

- [29] K. Takahashi, T. Ogata, H. Tjandra, Y. Yamaguchi, and S. Sugano, “Tool-body assimilation model based on body babbling and neurodynamical system,” *Mathematical Problems in Engineering*, vol. 2015, 2015.
- [30] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain.,” *Psychological review*, vol. 65, no. 6, p. 386, 1958.
- [31] M. Minsky and S. Papert, *Perceptrons*. MIT press, 1969.
- [32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, 1986.
- [33] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [34] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of keypoints,” in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, pp. 1–2, Prague, 2004.
- [35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [36] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- [37] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.
- [38] N. A. Bernstein, *The co-ordination and regulation of movements*. Pergamon Press Ltd., 1967.

- [39] R. M. Angulo-Kinzler, “Exploration and selection of intralimb coordination patterns in 3-month-old infants,” *Journal of Motor Behavior*, vol. 33, no. 4, pp. 363–376, 2001.
- [40] H. Head and G. Holmes, “Sensory disturbances from cerebral lesions,” *Brain*, vol. 34, no. 2-3, pp. 102–254, 1911.
- [41] A. Iriki, M. Tanaka, and Y. Iwamura, “Coding of modified body schema during tool use by macaque postcentral neurones,” *Neuroreport*, vol. 7, no. 14, pp. 2325–2330, 1996.
- [42] A. Maravita and A. Iriki, “Tools for the body (schema),” *Trends in cognitive sciences*, vol. 8, no. 2, pp. 79–86, 2004.
- [43] D. M. Wolpert and M. Kawato, “Multiple paired forward and inverse models for motor control,” *Neural networks*, vol. 11, no. 7, pp. 1317–1329, 1998.
- [44] “DARPA Robotics Challenge.” <http://www.theroboticschallenge.org/>, 2016. accessed on February 2, 2016.
- [45] T. Sugaiwa, M. Nezumiya, H. Iwata, and S. Sugano, “Motion-planning method with active body-environment contact for a hand-arm system including passive joints,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 69–74, 2010.
- [46] T. Sugaiwa, K. Takahashi, H. Kano, H. Iwata, and S. Sugano, “Handling and grasp control with additional grasping point for dexterous manipulation of cylindrical tool,” in *IEEE International Conference on Robotics and Biomimetics*, pp. 733–738, 2011.
- [47] E. Rombokas, M. Malhotra, E. Theodorou, Y. Matsuoka, and E. Todorov, “Tendon-Driven Variable Impedance Control Using Reinforcement Learning,” in *Robotics: Science and Systems*, pp. 369–376, 2012.

- [48] Y. Asano, H. Mizoguchi, T. Kozuki, Y. Motegi, J. Urata, Y. Nakanishi, K. Okada, and M. Inaba, “Achievement of twist squat by musculoskeletal humanoid with screw-home mechanism,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4649–4654, 2013.
- [49] T. Shirai, J. Urata, Y. Nakanishi, K. Okada, and M. Inaba, “Whole body adapting behavior with muscle level stiffness control of tendon-driven multi-joint robot,” in *IEEE International Conference on Robotics and Biomimetics*, pp. 2229–2234, 2011.
- [50] R. Niiyama, S. Nishikawa, and Y. Kuniyoshi, “Biomechanical approach to open-loop bipedal running with a musculoskeletal athlete robot,” *Advanced Robotics*, vol. 26, no. 3-4, pp. 383–398, 2012.
- [51] S. Seok, A. Wang, M. Y. Chuah, D. Otten, J. Lang, and S. Kim, “Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 3307–3312, IEEE, 2013.
- [52] A. Spröwitz, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. J. Ijspeert, “Towards dynamic trot gait locomotion: Design, control, and experiments with cheetah-cub, a compliant quadruped robot,” *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 932–950, 2013.
- [53] D. J. Braun, F. Petit, F. Huber, S. Haddadin, P. Van Der Smagt, A. Albu-Schäffer, and S. Vijayakumar, “Robots driven by compliant actuators: Optimal control under actuation constraints,” *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1085–1101, 2013.
- [54] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, “Learning object affordances: From sensory–motor coordination to imitation,” *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 15–26, 2008.

- [55] J. Morimoto and K. Doya, “Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning,” *Robotics and Autonomous Systems*, vol. 36, no. 1, pp. 37–51, 2001.
- [56] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *arXiv preprint arXiv:1509.02971*, 2015.
- [57] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016.
- [58] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, “Dynamical movement primitives: learning attractor models for motor behaviors,” *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.
- [59] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection,” *arXiv preprint arXiv:1603.02199*, 2016.
- [60] M. M. Williamson, *Robot arm control exploiting natural dynamics*. PhD thesis, Massachusetts Institute of Technology, 1999.
- [61] 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, no. 11, p. e1000220, 2008.
- [62] J. Sturm, C. Plagemann, and W. Burgard, “Unsupervised body scheme learning through self-perception,” in *IEEE International Conference on Robotics and Automation*, pp. 3328–3333, 2008.
- [63] R. Saegusa, G. Metta, G. Sandini, and S. Sakka, “Active motor babbling for sensorimotor learning,” in *IEEE International Conference on Robotics and Biomimetics*, pp. 794–799, 2009.

- [64] R. Saegusa, G. Metta, G. Sandini, and L. Natale, “Developmental perception of the self and action,” *IEEE transactions on neural networks and learning systems*, vol. 25, no. 1, pp. 183–202, 2014.
- [65] M. Rolf, J. J. Steil, and M. Gienger, “Goal babbling permits direct learning of inverse kinematics,” *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 3, pp. 216–229, 2010.
- [66] M. Rolf and J. J. Steil, “Efficient exploratory learning of inverse kinematics on a bionic elephant trunk,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 6, pp. 1147–1160, 2014.
- [67] K. Mochizuki, S. Nishide, H. G. Okuno, and T. Ogata, “Developmental human-robot imitation learning of drawing with a neuro dynamical system,” in *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2336–2341, 2013.
- [68] K. Sasaki, H. Tjandra, K. Noda, K. Takahashi, and T. Ogata, “Neural network based model for visual-motor integration learning of robot’s drawing behavior: Association of a drawing motion from a drawn image,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2736–2741, IEEE, 2015.
- [69] K. Sasaki, K. Noda, and T. Ogata, “Visual motor integration of robot’s drawing behavior using recurrent neural network,” *Robotics and Autonomous Systems*, vol. 86, pp. 184–195, 2016.
- [70] K. Takahashi, K. Kim, T. Ogata, and S. Sugano, “Tool-body assimilation model considering grasping motion through deep learning,” *Robotics and Autonomous Systems*, vol. 91, pp. 115–127, 2017.
- [71] J. L. Elman, “Finding structure in time,” *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

- [72] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks.,” in *International Conference on Artificial Intelligence and Statistics*, vol. 9, pp. 249–256, 2010.
- [73] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mosenlechner, D. Pangercic, T. Ruhr, and M. Tenorth, “Robotic roommates making pancakes,” in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 529–536, IEEE, 2011.
- [74] K. Ramirez-Amaro, E. Dean-Leon, and G. Cheng, “Robust semantic representations for inferring human co-manipulation activities even with different demonstration styles,” in *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 1141–1146, IEEE, 2015.
- [75] K. Nagahama, K. Yamazaki, K. Okada, and M. Inaba, “Manipulation of multiple objects in close proximity based on visual hierarchical relationships,” in *IEEE International Conference on Robotics and Automation*, pp. 1303–1310, 2013.
- [76] A. Stoytchev, “Behavior-grounded representation of tool affordances,” in *IEEE International Conference on Robotics and Automation*, pp. 3060–3065, 2005.
- [77] C. Nabeshima, Y. Kuniyoshi, and M. Lungarella, “Towards a model for tool-body assimilation and adaptive tool-use,” in *IEEE International Conference on Development and Learning (ICDL)*, pp. 288–293, IEEE, 2007.
- [78] S. Nishide, J. Tani, T. Takahashi, H. G. Okuno, and T. Ogata, “Tool-body assimilation of humanoid robot using a neurodynamical system,” *Autonomous Mental Development, IEEE Transactions on*, vol. 4, no. 2, pp. 139–149, 2012.
- [79] Y. Yamaguchi, H. Nobuta, S. Nishide, H. G. Okuno, and T. Ogata, “Tool-body assimilation model using neuro-dynamical system for acquiring representation of tool function,” in *Proceedings of IROS-2012 Workshop on Cognitive Neuroscience Robotics*, pp. PM2–3, 6 pages, 2012.

- [80] J. J. Gibson, “The senses considered as perceptual systems.,” 1966.
- [81] A. Streri and J. Féron, “The development of haptic abilities in very young infants: From perception to cognition,” *Infant Behavior and Development*, vol. 28, no. 3, pp. 290–304, 2005.
- [82] C. F. Michaels, Z. Weier, and S. J. Harrison, “Using vision and dynamic touch to perceive the affordances of tools,” *Perception*, vol. 36, no. 5, pp. 750–772, 2007.
- [83] T. Sasaoka, H. Mizuhara, and T. Inui, “The interaction between the parietal and motor areas in dynamic imagery manipulation: an fmri study,” in *Advances in Cognitive Neurodynamics (II)*, pp. 345–349, Springer, 2011.
- [84] M. Carpenter, “The co-ordination and regulation of movements.,” *Journal of Neuropathology & Experimental Neurology*, vol. 27, no. 2, p. 348, 1968.
- [85] A. D’Souza, S. Vijayakumar, and S. Schaal, “Learning inverse kinematics,” in *IEEE International Conference on Intelligent Robots and Systems*, pp. 298–303, 2001.
- [86] T. Iwasaki, G. Venture, and E. Yoshida, “Identification of the inertial parameters of a humanoid robot using grounded sole link,” in *IEEE International Conference on Humanoid Robots*, pp. 449–454, 2012.
- [87] Kokoro Company Ltd, “Kokoro: Custom-made robot: ACTROID.” http://www.kokoro-dreams.co.jp/english/rt_tokutyu/actroid.html, 2016. accessed on February 2, 2016.
- [88] F. Kanehiro, H. Hirukawa, and S. Kajita, “OpenHRP: Open architecture humanoid robotics platform,” *The International Journal of Robotics Research*, vol. 23, no. 2, pp. 155–165, 2004.
- [89] I. A. Kapandji, *The Physiology of the Joints: Upper Limb*, vol. 1. Churchill Livingstone, 2007.

- [90] Willow Garage, “Willow Garage: PR2.” <http://www.willowgarage.com/pages/pr2/overview>, 2016. accessed on February 2, 2016.
- [91] B. Hopkins and H. PrechtI, “A qualitative approach to the development of movements during early infancy,” *Clinics in developmental medicine*, no. 94, pp. 179–197, 1984.
- [92] G. Taga, R. Takaya, and Y. Konishi, “Analysis of general movements of infants towards understanding of developmental principle for motor control,” in *Systems, Man, and Cybernetics, 1999. IEEE SMC’99 Conference Proceedings. 1999 IEEE International Conference on*, vol. 5, pp. 678–683, IEEE, 1999.
- [93] C. Von Hofsten, “An action perspective on motor development,” *Trends in cognitive sciences*, vol. 8, no. 6, pp. 266–272, 2004.
- [94] C. Einspieler and H. F. PrechtI, “PrechtI’s assessment of general movements: a diagnostic tool for the functional assessment of the young nervous system,” *Mental retardation and developmental disabilities research reviews*, vol. 11, no. 1, pp. 61–67, 2005.
- [95] S. Murata, J. Namikawa, H. Arie, J. Tani, and S. Sugano, “Learning to reproduce fluctuating behavioral sequences using a dynamic neural network model with time-varying variance estimation mechanism,” in *IEEE International Conference on Development and Learning and Epigenetic Robotics*, pp. 1–6, 2013.
- [96] S. Murata, Y. Yamashita, H. Arie, T. Ogata, S. Sugano, and J. Tani, “Learning to perceive the world as probabilistic or deterministic via interaction with others: A neuro-robotics experiment,” *IEEE Transactions on Neural Networks and Learning Systems*, 2015.
- [97] J. Namikawa and J. Tani, “A model for learning to segment temporal sequences, utilizing a mixture of rnn experts together with adaptive variance,” *Neural Networks*, vol. 21, no. 10, pp. 1466–1475, 2008.

- [98] S. Funabashi, A. Schmitz, T. Sato, S. Somlor, and S. Sugano, “Robust in-hand manipulation of variously sized and shaped objects,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 257–263, IEEE, 2015.
- [99] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” *arXiv preprint arXiv:1504.00702*, 2015.
- [100] J. Martens, “Deep learning via hessian-free optimization,” in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 735–742, 2010.
- [101] B. A. Pearlmutter, “Fast exact multiplication by the hessian,” *Neural computation*, vol. 6, no. 1, pp. 147–160, 1994.
- [102] N. N. Schraudolph, “Fast curvature matrix-vector products for second-order gradient descent,” *Neural computation*, vol. 14, no. 7, pp. 1723–1738, 2002.
- [103] S. J. Pan and Q. Yang, “A survey on transfer learning,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [104] H. Arie, T. Endo, T. Arakaki, S. Sugano, and J. Tani, “Creating novel goal-directed actions at criticality: A neuro-robotic experiment,” *New Mathematics and Natural Computation*, vol. 5, no. 01, pp. 307–334, 2009.
- [105] T. Kohonen, *Self-organization and associative memory*, vol. 8. Springer Science & Business Media, 2012.
- [106] K. Takahashi, T. Ogata, H. Tjandra, Y. Yamaguchi, Y. Suga, and S. Sugano, “Tool-body assimilation model using a neuro-dynamical system for acquiring representation of tool function and motion,” in *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1255–1260, IEEE, 2014.
- [107] K. Takahashi, T. Ogata, H. Tjandra, S. Murata, H. Arie, and S. Sugano, “Tool-body assimilation model based on body babbling and a neuro-dynamical system

for motion generation,” in *International Conference on Artificial Neural Networks*, pp. 363–370, Springer, 2014.

Relevant Publications

Journal Papers

1. Kuniyuki Takahashi, Kitae Kim, Tetsuya Ogata, Shigeki Sugano: “Tool-body Assimilation Model Considering Grasping Motion through Deep Learning,” *Robotics and Autonomous Systems*, Volume. 91, pp. 115-127, January, 2017
DOI: 10.1016/j.robot.2017.01.002
2. Kuniyuki Takahashi, Tetsuya Ogata, Hadi Tjandra, Yuki Yamaguchi, Shigeki Sugano: “Tool-body Assimilation Model Based on Body Babbling and Neuro-dynamical System,” *Mathematical Problems in Engineering*, Article ID 837540, vol. 2015, 15 pages, 2015
DOI: 10.1155/2015/837540

Lecture Notes

1. Kuniyuki Takahashi, Hadi Tjandra, Tetsuya Ogata, Shigeki Sugano: “Body Model Transition by Tool Grasping During Motor Babbling using Deep Learning and RNN,” *Lecture Notes in Computer Science (The 25th International Conference on Artificial Neural Networks (ICANN 2016))*, pp 166-174, Barcelona, Spain, September 6th-9th, 2016
DOI: 10.1007/978-3-319-44778-0_20
2. Kuniyuki Takahashi, Kanata Suzuki, Tetsuya Ogata, Hadi Tjandra, Shigeki

Sugano: “Efficient Motor Babbling Using Variance Predictions from a Recurrent Neural Network,” Lecture Notes in Computer Science (22nd International Conference on Neural Information Processing (ICONIP2015)), pp.26-33, Istanbul, Turkey, November 9th - 12th, 2015

DOI: 10.1007/978-3-319-26555-1_4

3. Kuniyuki Takahashi, Tetsuya Ogata, Hadi Tjandra, Shingo Murata, Hiroaki Arie, Shigeki Sugano: “Tool-body Assimilation Model based on Body Babbling and a Neuro-dynamical System for Motion Generation,” Lecture Notes in Computer Science (The 24th International Conference on Artificial Neural Networks (ICANN 2014)), pp. 363-370, Hamburg, Germany, September 15th-19th, 2014
DOI: 10.1007/978-3-319-11179-7_46

International Conferences (Full paper)

1. Kuniyuki Takahashi, Tetsuya Ogata, Hiroki Yamada, Hadi Tjandra, Shigeki Sugano: “Effective Motion Learning for a Flexible-Joint Robot using Motor Babbling,” 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS2015), pp. 2723-2728, Hamburg, Germany, September 28th - October 2nd, 2015
DOI: 10.1109/IROS.2015.7353750
2. Kuniyuki Takahashi, Tetsuya Ogata, Hadi Tjandra, Yuki Yamaguchi, Yuki Suga, and Shigeki Sugano: “Tool-body Assimilation Model using a Neuro-dynamical System for Acquiring Representation of Tool Function and Motion,” IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2014), Accepted for Oral Presentation, ThAT3.6, pp. 1255 - 1260, Besancon, France, July 8th-11th, 2014
DOI: 10.1109/AIM.2014.6878254

Domestic Conferences (in Japanese)

1. 高橋城志, 尾形哲也, 中西淳, Gordon Cheng, 菅野重樹: “多自由度柔軟関節ロボットのためのモーターバブリングを用いた効率的な動的動作の学習,” 第17回計測自動制御学会システム・インテグレーション部門講演会 (SI 2016), 1C4-1, 札幌, 2016年12月15日-17日
2. 金紀泰, 高橋城志, 尾形哲也, 菅野重樹: “Deep Learningによる把持動作を考慮した道具身体化モデルの学習と動作生成,” 第17回計測自動制御学会システム・インテグレーション部門講演会 (SI 2016), 2C1-1, 札幌, 2016年12月15日-17日
3. 金紀泰, 高橋城志, 尾形哲也, 菅野重樹: “Deep Learningによるロボット道具使用のための把持を考慮した道具身体化モデル,” 第34回日本ロボット学会 学術講演会 (RSJ2016), 3Y3-02, 山形, 2016年9月7日-9日
4. 金紀泰, 高橋城志, Hadi Tjandra, 尾形哲也, 菅野重樹: “ディープニューラルネットワークとリカレントニューラルネットワークによる把持位置を考慮した道具使用モデル,” 日本機械学会ロボティクス・メカトロニクス講演会 (ROBOMECH2016), 2P1-12b1, 横浜, 2016年6月8日-11日
5. 山田浩貴, 下栗逸爾, 高橋城志, Hadi Tjandra, 菅野重樹, 尾形哲也: “道具機能表現を分類するための道具身体化モデルの転移学習,” 日本機械学会ロボティクス・メカトロニクス講演会 (ROBOMECH2016), 2P1-12b4, 横浜, 2016年6月8日-11日
6. 金紀泰, 高橋城志, Hadi Tjandra, 尾形哲也, 菅野重樹: “モーターバブリングと神経回路モデルを用いた道具身体化～複数道具把持動作による身体遷移過程の学習と生成～,” 情報処理学会 第78回全国大会 (IPSJ2016), 7L-02, 東京, 2016年3月10日-12日
7. Hadi Tjandra, 高橋城志, 尾形哲也, 菅野重樹: “ロボットを用いた身体モデル変調による道具身体化モデル～道具把持動作による身体遷移過程の学習と動作生成～,” 第16回計測自動制御学会システム・インテグレーション部門講演会 (SI 2015), 2C3-1, 名古屋, 2015年12月14日-16日

8. Kuniyuki Takahashi, Tetsuya Ogata, Shigeki Sugano, Gordon Cheng : “Dynamic Motion Learning for a Flexible-Joint Robot using Active-Passive Motor Babbling,” The 33st Annual Conference of the Robotics Society of Japan (第33回日本ロボット学会 学術講演会 (RSJ2015)), 2G1-07, Tokyo, September 3rd-5th, 2015
9. 山田浩貴, 高橋城志, 尾形哲也, Hadi Tjandra, 菅野重樹 : “受動的な動作を含むモータバブリングによる柔軟関節を有するロボットの動作学習,” 第33回日本ロボット学会 学術講演会 (RSJ2015), 1B3-07, 東京, 2015年9月3日-5日
10. 陽品駒, 高橋城志, 尾形哲也, 菅祐樹, 菅野重樹 : “RTC による深層学習モデルと柔軟関節ロボットの統合 ～道具身体化モデルの学習データ収集と動作実現～,” 第33回日本ロボット学会 学術講演会 (RSJ2015), 3F3-06, 東京, 2015年9月3日-5日
11. 鈴木彼方, 高橋城志, Hadi Tjandra, 村田真悟, 菅野重樹, 尾形哲也 : “再帰神経回路モデルによる分散予測を用いた柔軟関節ロボットの身体ダイナミクスの探索,” 日本機械学会ロボティクス・メカトロニクス講演会 (ROBOMECH2015), 2P1-S06, 京都, 2015年5月17日-19日
12. 山田浩貴, 高橋城志, 尾形哲也, Hadi Tjandra, 菅野重樹 : “身体バブリングによる事前学習を用いた柔軟関節ロボットの効果的な動作学習,” 情報処理学会 第77回全国大会 (IPSJ2015), 5T-08, 京都, 2015年3月17日-19日
13. 高橋城志, Hadi Tjandra, 菅野重樹, 尾形哲也 : “ロボットを用いた Body Image の変調による道具身体化,” 日本発達神経科学学会 第3回大会, No.9, 東京, 2014年10月18日-19日
14. 高橋城志, Hadi Tjandra, 尾形哲也, 菅野重樹 : “神経回路モデルと身体バブリングによる道具身体化を用いた複数の未学習道具の使用,” 第32回日本ロボット学会 学術講演会 (RSJ2014), 3I1-06, 福岡, 2014年9月4日-6日
15. 高橋城志, 尾形哲也, Hadi Tjandra, 野田邦昭, 村田真悟, 有江浩明, 菅野重樹 : “神経回路モデルと身体バブリングによる道具身体化と道具機能の獲得,” 日本

機械学会ロボティクス・メカトロニクス講演会 (ROBOMECH2014), P3P2-P02, 富山, 2014年5月25日-29日

16. 高橋城志, 尾形哲也, Hadi Tjandra, 野田邦昭, 村田真悟, 有江浩明, 菅野重樹: “身体バブリングと再帰結合型神経回路モデルによる道具身体化～深層学習による画像特徴量抽出～,” 人工知能学会全国大会 (第28回) (JSAI2014), 1I4-OS-09a-4, 愛媛, 2014年5月12日-15日
17. Hadi Tjandra, 高橋城志, 村田真悟, 有江浩明, 山口雄紀, 尾形哲也, 菅野重樹: “神経力学モデルと身体バブリングに基づく道具身体化と動作生成,” 情報処理学会 第76回全国大会 (IPSJ2014), 1S-4, 東京, 2014年3月11日-13日
18. 高橋城志, Hadi Tjandra, 山口雄紀, 菅佑樹, 菅野重樹, 尾形哲也: “神経回路モデルを用いた道具身体化による道具機能と動作の獲得,” 第14回計測自動制御学会システム・インテグレーション部門講演会 (SI 2013), 2K1-5, 神戸, 2013年12月18日-20日
19. 高橋城志, 尾形哲也, 岩田浩康, 菅野重樹: “全探索と人間のアフォーダンスとの定量的差異の検証,” 日本赤ちゃん学会 第13回学術集会, P-43, 福岡, 2013年5月25日-26日

Other Publications

International Conferences

1. Kazuma Sasaki, Hadi Tjandra, Kuniaki Noda, Kuniyuki Takahashi, Tetsuya Ogata: “Neural Network based Model for Visual-motor Integration Learning of Robot’s Drawing Behavior: Association of a Drawing Motion from a Drawn Image,” 2015 IEEE/RSJ International Conference on Intelligent Robots and System(IROS2015), pp. 2736-2741, Hamburg, Germany, September 28th - October 2nd, 2015
DOI: 10.1109/IROS.2015.7353752
2. Taisuke Sugaiwa, Kuniyuki Takahashi, Hiroyuki Kano, Hiroyasu Iwata, and Shigeki Sugano: “Handling and Grasp Control with Additional Grasping Point for Dexterous Manipulation of Cylindrical Tool,” 2011 IEEE International Conference on Robotics and Biomimetics(ROBIO2011), Accepted for Poster Presentation, pp. 733 - 738, Phuket, Thailand, December 7th-11th, 2011
DOI: 10.1109/ROBIO.2011.6181373

Domestic Conferences

1. 陽品駒, 佐々木一磨, 鈴木彼方, 加瀬敬唯, 高橋城志, 菅野重樹, 尾形哲也 : “Wizard of Oz と深層学習によるロボットの柔軟物折り畳み作業,” 第34回日本ロボット学会 学術講演会 (RSJ2016), 2G2-05, 山形, 2016年9月7日-9日
2. 鈴木彼方, 新古真純, 陽品駒, 高橋城志, 菅野重樹, 尾形哲也 : “CNNによる

二次元物体画像から実ロボットでの把持動作生成，” 日本機械学会ロボティクス・メカトロニクス講演会 (ROBOMECH2016)，2P1-12b7，横浜，2016年6月8日-11日

3. 鈴木彼方，高橋城志，Gordon Cheng，尾形哲也：“深層学習を用いた多自由度ロボットによる柔軟物の折り畳み動作生成，” 情報処理学会 第78回全国大会 (IPSJ2016)，1P-04，東京，2016年3月10日-12日
4. 濱翔平，平井諒，高橋城志，山田浩貴，尾形哲也，菅野重樹，金天海：“力学系学習木による効率的な学習のための階層性を利用した入力ベクトル決定法，” 情報処理学会 第78回全国大会 (IPSJ2016)，4P-02，東京，2016年3月10日-12日
5. 藤倉理詠，高橋城志，Somlor, Sophon，佐藤高志，戸村摩美，菅野重樹：“滑りを利用した円柱形道具の外部環境に非接触な持ち直し手法，” 第15回計測自動制御学会システム・インテグレーション部門講演会 (SI2014)，東京，2014年12月15日-17日
6. 野久陽介，高橋城志，藤倉理詠，小島康平，Alexander Schumitz，岩田浩康，菅野重樹：“受動柔軟ハンドの操りにおける誤差許容性の高い姿勢経路による把持力安定化” 第31回日本ロボット学会 学術講演会 (RSJ2013)，東京，2013年9月4日-6日
7. 伊神克哉，市川健太郎，高橋城志，早川正一，袁国良，Moondeep Chandra Shrestha，岩田浩康，菅野重樹：“人間とロボットとの協調移動制御-人間の抵抗に応じて接触による働きかけと回避を行う移動調整手法-，” 第13回計測自動制御学会システム・インテグレーション部門講演会 (SI2012)，3N2-6，福岡，2012年12月18日-20日
8. 市川健太郎，菅岩泰亮，高橋城志，長嶺伸治，前代アト夢，加納弘之，野口博史，岩田浩康，金道敏樹，菅野重樹：“人間とロボットとの協調移動制御-優先関係に基づく接触移動計画手法-，” 第12回計測自動制御学会システム・インテグレーション部門講演会 (SI2011)，1C1-2，京都，2011年12月23日-25日
9. 市川健太郎，菅岩泰亮，高橋城志，長嶺伸治，前代アト夢，加納弘之，野口博史，岩田浩康，金道敏樹，菅野重樹：“人間とロボットとの協調移動制御-移動

中の接触による働きかけを実現する全身制御手法-, ” 第29回日本ロボット学会
学術講演会 (RSJ2011), 3H3-2, 東京, 2011年9月7日-9日

10. 高橋城志, 菅岩泰亮, 加納弘之, 岩田浩康, 菅野重樹: “円柱形道具の精密操作のための把持・操り制御-幾何学的拘束を考慮した接触点配置による姿勢安定化-, ” 第29回日本ロボット学会 学術講演会 (RSJ2011), 3E2-7, 東京, 2011年9月7日-9日

What is *Embodiment Informatics*?

Embodiment Informatics means the object with the body, composed of sensors and motors, permanently adapts to its internal and environmental models through interaction with the environment. In other words, the internal and environmental models are monitored from the acquired data through the body, and if there is a discrepancy between the prediction by the constructed model and the actual result, the internal and environmental models will be reconstructed. For example, as the situation changes, the acquired data also changes, so the internal and environmental model also changes. Furthermore, when the body changes, the influence on the environment changes, so the internal and environmental models are reconstructed and adapted. In the field of AI, so far, data pertaining to a specific target is collected first, and the design of the internal representation and features suitable for that data are established next. In the robotics field as well, there is target to be solved before, and the view point is how to describe those dynamic models. Therefore, it is necessary for humans to design feature quantities and models for each task. This is possible in an environment where modeling can be accomplished such as chess and Go, where all conditions are known, such that the industrial robot has a constructed environment that prevents inconsistencies with the model. However, as long as humans continue designing models, unexpected situations cannot be dealt with by the robot. Going forward, the ability of a robot to accommodate dynamic and uncertain environments, such as those involving human interaction, will be required. It is important not to set data or tasks targeted by humans in advance but to acquire and update models themselves from time to time with acquired data from the interaction of the body of the robot with the environment.

As an example, I demonstrated the results of applying this concept to humanoid robots in this thesis. The robots performed acquisition and extension of the body model using motor babbling through deep learning. Through this approach, the robots successfully produced internal and environmental models through interacting with the environment independent of pre-designed models. By performing motor babbling, the robot could learn the relationship between the motors and sensors as the internal model as well as learning how the environment changed as the environment model. By establishing the robot body model, that is, dynamic motion learning for a flexible-joint robot, after establishing the body model with motor bubbling, the body model is adjusted according to the task. Consequently, it was possible to perform dynamic tasks including contact with the environment using a multi-DOFs flexible-joint robot, which has been considered difficult in the field robot research so far. This indicates that robots are capable of dealing with dynamic situations. Furthermore, the extension of the established robot body model, that is, the tool-body assimilation model considering grasping, enabled the robot to use tools by modulating its body model. This indicates the ability of the robot to accommodate a modified body.

This thesis employed a humanoid having a complex body to demonstrate the capabilities of the proposed approach. This approach could be applied to simpler structures as well. In the case of a device comprising a microphone and a speaker, it would possible to acquire environmental information with a microphone and the corresponding output information with a speaker.