NAIST-IS-DD0761001

Doctoral Dissertation

An Imitation Learning Framework that Explicitly Considers Robot Configurations and Robot-Environment Interactions

Yuka Ariki

November 4, 2010

Department of Bioinformatics and Genomics
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of Engineering

Yuka Ariki

Thesis Committee:

Professor Kazushi Ikeda	Supervisor
Professor Mitsuo Kawato	Co-supervisor
Professor Tukasa Ogasawara	Co-supervisor
Doctor Jun Morimoto	Co-supervisor

An Imitation Learning Framework that Explicitly Considers Robot Configurations and Robot-Environment Interactions[1]

Yuka Ariki

Abstract

Behavioral science has focused on imitation learning, a skill acquisition strategy in which learners try to learn skills by observing instructor's behaviors. The imitation learning is known as a learning strategy that can be executed only by animals with higher intelligence and considered as an efficient way to control high-dimensional system, e.g., human's body, that has many sensors and actuators.

Thus, in recent years, attention has been directed to imitation learning in humanoid robotics. For humanoid robots with many degrees of freedom, a considerable amount of time is required to prepare multiple motions in advance since the number of combinations of joint angle trajectories are quite large. Imitation learning is considered as a suitable approach to initialize parameters in the vast search space.

However, direct use of the instructor's motion trajectories often fails because of the difference of physical properties between the instructor and the robot. For example, a humanoid robot can falls over or hits own body by own hand if the robot directly copy the corresponding joint trajectories of an instructor's behavior.

Humans can imitate a motion without falling over by simply watching an instructor's movement even if their physical properties differ from the instructor's.

In this thesis, we show how to imitate like human, which overcome the difference of physical properties between the instructor and robot. We propose learning methods to deal with three major case of the difference of physical properties involved in the imitation learning paradigm. 1) The kinematics

[1] Doctoral Dissertation, Department of Bioinformatics and Genomics, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0761001,September 15, 2010

(e.g., size, the number of joints) of a demonstrator and an imitator are different, 2) The dynamics (e.g., weight, inertia) of a demonstrator and an imitator are different, and 3) The skill transfer problem rather than only imitating behaviors.

For the first case, we find a shared low-dimensional latent space between demonstrator's and imitator's postures. Then, we derive a corresponding imitator's movement to a demonstrator's behavior. For the second case, we estimate the ground reaction force from captured demonstrator's movements so that an imitator can generate physically consistent imitated behaviors. For the third case, we extract task-related features from a demonstrator's movements. Then, an imitator tries to improve task performance by using the extracted features.

Keywords: Imitation Learning, Skill Transfer, Switching State-Space Models, Shared-GPLVM

Acknowledgement

まずはじめに、研究活動全般に渡って熱心にご指導下さったATR脳情報通信総合研究所, 脳情報研究所, ブレインロボットインタフェース研究室, 森本淳室長に深く感謝したいと思います。文系出身者でロボットの研究がやりたいというだけの何の知識もない自分に興味深い研究テーマを与えてくださり, 工学, 数学, プログラムの基礎や論理的な考え方, 研究発表, 研究の仕方など研究生活に関わる全てを教えて頂き, 至れり尽くせりで5年以上もの長い間熱心にご指導頂きました。何も出来なかった自分がここまでやってこれたのはひとえに森本室長が理解力のない自分を最後まで見捨てず, 根気強く見守って下さったからです。改めて深く感謝致します. 本当に有難うございました.

川人光男先生, 神谷之康先生, 石井信先生, 池田和司先生, 柴田智広先生, ATR脳情報通信総合研究所, 脳情報解析研究所, 計算脳イメージング研究室, 佐藤雅昭室長には, 毎週の定例研究会などを通じて研究をより良い方向へと導いて下さいました. 先生方からは豊富な知識に基づく研究へのアプローチの仕方などを多くのことを教えて頂きました. 川人先生には脳情報通信総合研究所所長として、素晴らしい研究環境を与えて頂きました. ATRの連携研究室に所属して, ATRで博士課程の間この様な素晴らしい環境で研究できたことは自身の誇りであり, 一生の財産になりました. 改めて深く感謝致します. 銅谷賢治先生には, 修士課程の間, 定例研究会でご指導頂きました. 何度か研究発表も見ていただき, 研究への姿勢を教えて頂きました. ここに改めて感謝致します.

ATR脳情報通信総合研究所, 脳情報研究所, ブレインロボットインタフェース研究室, 玄相昊客員研究員にはヒューマノイドロボットのバランス制御に関しての知識を教えて頂き, 研究について直接ご指導頂きました. 自分の拙い質問にも熱心に答えて下さり、この学位論文を書き上げるにあたって, 論文の書き方や表現方法など多くのことを教えて頂き本当に感謝しています.

ATR脳情報研究所情報科学ヒューマノイドロボット研究室のGordon Cheng室長には, 研究活動全般に渡って多大なるご指導, ご助力を頂きました. 研究室や個別のミーティングを通じての研究指導, 英語表現の指導をして頂きま

した．見聞を広めるために外部の研究会などへの積極的な参加を勧めて下さり, 研究について多くの見識を与えてくださりました．ここに改めて感謝致します．

ATR 連携計算神経科学講座の先輩であり,奈良先端科学技術大学院大学, 情報科学研究科, 知能情報処理学講座, 松原崇充助教には学生時代を含め５年以上もの長い間, 公私共に非常にお世話になりました．研究発表の発表練習を見て頂き, その度にするどい質問から多くのことを教えていただきました．身近な存在として基礎的な知識や論理的な考え方など研究に関する全てのことから, 人生観に至るまでとても勉強になりました．松原さんの存在なしには自分の学位取得は有り得ませんでした．ここに改めて深く感謝致します．

博士後期課程３年次に Disney Research Pittsburgh でインターンとして研究する機会を頂きました．そこで Professor Jessica Hodgines, Senior Research Scientist 山根克氏にご指導頂く機会を受け,研究を出来たことは自身にとって何よりも変えがたい経験でした．Jessica と山根氏には研究の指導だけでなくアメリカでの生活の多くのサポートをしていただき, どれだけ多く感謝してもしたりません．二人には多くの事を教えて頂き研究の進め方, 豊富な知識と発想それを実現する堅実な実行力に強い憧れを頂きました．研究や人生について多くの事教えて頂き,貴重な経験をさせてもらった Disney Research Pittsburgh, Carnegie Mellon University Graphics Lab のメンバー全てに改めて深く感謝致します．

ATR 連携計算神経科学講座の先輩、同級生、後輩にも大変お世話になりました．藤原祐介さん, 南部功夫さん, 内田肇さん, 鬼塚美帆さんには公私共に仲良くさせて頂きました．研究生活が充実したものだったのは良き友人達に恵まれたからだと思います．ここに改めて深く感謝致します．

ATR 脳情報通信総合研究所, 脳情報研究所, ブレインロボットインタフェース研究室の皆様にも研究,　日々の生活面で大変お世話になりました。改めて感謝致します．基幹講座として修士課程, 博士課程で所属した生命システム学講座, 論理生命学講座の皆様, 石井信教授, 池田和司教授には大変お世話になりました．石井先生には博士前期課程, 後期課程の途中まで基幹講座教授として研究会や輪講を通じてご指導を頂き, いつも温かい言葉で励まして頂きました．池田先生にはその後の研究指導やこの学位論文を書き上げるにあたっても丁寧に指導して頂き, よく気にかけて励まして頂きました．ここに改めて深く感謝致します

お忙しい中, 論文審査を引き受けて下さいました池田和司教授,　川人光男教授,　小笠原司教授,　森本淳博士には改めて深く感謝致します．小笠原先生には修士課程の論文審査も引き受けて頂き、研究発表について多くのことを指導して頂き, ロボティクスの研究に対しての多くの知識を教えていただき

ました．最後に博士課程への進学を認め，その生活を支えてくれた家族，友人たちに深く感謝致します．本学位論文の提出に至るまで本当に多くの方々に支えられてきました．今一度改めて感謝を申し上げるとともに，今後もご指導くださいますように厚くお願い申し上げます。

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Imitation learning has been one of the major research topics in neuroscience, especially after remarkable neural activities related to the imitation learning are observed in a monkey's brain. Some neurons in monkey's premotor cortex (area F5), called mirror neurons, activate not only when a monkey observes a specific behavior but also when the monkey generates the same behavior (DiPellegrino, Fadiga, Fogassi, Gallese, & Rizzolatti, 1992).

Also, behavioral science has focused on imitation learning, a skill acquisition strategy in which learners try to learn skills by observing an instructor's behaviors. The imitation learning is known as a learning strategy that can be executed only by animals with higher intelligence and considered as an efficient way to control high-dimensional systems, such as the human body that has many sensors and actuators (Schaal, 1999).

Thus, in recent years, attention has been directed to imitation learning in humanoid robotics. For humanoid robots with many degrees of freedom, a considerable amount of time is required to prepare multiple motions in advance since the number of combinations of joint angle trajectories are quite large. Imitation learning is considered as a suitable approach to initialize parameters in the vast search space (Calinon, Guenter, & Billard, 2007; Nakaoka, Nakazawa, Yokoi, Hirukawa, & Ikeuchi, 2003).

However, direct use of the instructor's motion trajectories often fails because of the difference of physical properties between the instructor and the robot.

In this thesis, we focus on three representatives of differences of physical properties and describe influence of these differences.

1) The kinematics (e.g., size, the number of joints) of a demonstrator and an imitator are different.

Because robots have different kinematics, it is difficult to successfully apply joint trajectories of an instructor's behavior. For example, if a robot directly follow the joint angles of a human, the hand endpoint of the robot may be different from human's. However, if the robot achieves the hand endpoint of human, it is not necessarily the case that the robot successfully imitates. For an extreme example, how can we generate AIBO motion from human motion capturing data? This problem is focused on Computer Graphics (CG) as "retargetting" (Gleicher, 1998): the problem of adapting an animated motion from one character to another. A technique for retargetting motion treat character's configuration like human on CG, however almost all CG characters have big kinematics differences from human have to behave like human. The technique for retargetting of "non-human character" may be useful to generate motion of robot which has different configurations. Keyframing method is the simplest method of animating based on key poses of these characters. We assume shared latent spaces between human and non-human character. We describe how to interpolate the key poses on this shared latent spaces to overcome the difference of kinematics.

2) The dynamics (e.g., weight, inertia) of a demonstrator and an imitator are different.

The difference of dynamics between the instructor and the robot can complicate simply mimicking of demonstrated motion. For example, a humanoid robot can falls over or hits own body by own hand if the robot directly copy the corresponding joint trajectories of a instructor's behavior. By simply watching an instructor's movement, humans can imitate a motion without falling over even if physical properties differ from those of the instructor. This issue is related to the mind-reading problem, that is, Mind-reading is explained as the ability to estimate another individual's mental state by adopting own perspective (Gallese & Goldman, 1998). It is suggested that the mirror neurons are possibly used for the mind-reading (Gallese & Goldman, 1998). Similarly, it is supposed that humans take their own dynamics into account when humans are imitating an instructor's behaviors. Imitated movements are not identical to the instructor's behaviors because of the difference in dynamics. Then, we tackle the question is how the observed behaviors are converted to imitated actions. The concept of movement primitives is one of the key ideas to explain this conversion (Wolpert & Kawato, 1998). Movement primitives are components that can generate motor commands to accomplish a goal directed behavior (Schaal, Ijspeert, & Billard, 2003). Movements are essentially modular, in that we put through multiple qualitatively different tasks. Many behaviors are derived from combinations of previously acquired movement primitives. One of the main problem is how to define movement primitives. We represent how to define movement primitives which are able to estimate human mental state to overcome the difference of dynamics.

3) The skill transfer problem rather than only imitating behaviors.

Robot that has the difference of dynamics and kinematics from human cannot accomplish complicate task by following only joint angles or marker positions of human. This problem is related to skill transfer. It is important for skill transfer to find key features about complicate task. (Bentivegna, Atkeson, Ude, & Cheng, 2004) show imitation learning framework as skill transfer on air-hockey and marble maze, however experimenter's defined key features to accomplish goal of task in this study. Key features of task should be learned to apply another task. We try to acquire key feature of task from human learning data. It is supposed that robot should imitate how to accomplish goal of task based on key features.

1.2 Research purpose

Overcoming of these differences of physical properties is indispensable for utilization of imitation learning. We solve three cases of the difference of physical properties by considering feature spaces between human and robot, CG character. We propose imitation learning framework and its application:

1) Feature space to represent of kinematics

We have data set of a few key poses of human and non-human characters (e.g. lamp, penguin, squirrel) which have big differences from human in kinematics and topology. The key poses should be selected so that they are representative of the space of the poses in the captured human motion. Human's joint space may have very different of freedom than non-human character's joint space, however they may be made to assume similar pose. The latent variable space then characterizes the common pose space. The key poses of non-human character can be interpolated corresponding with human poses by using this shared pose space.

2) Primitives to represent of dynamics

We use primitives to recognize observed behaviours in addition to generate the imitated behaviors. Thus, we simply use the term *primitives* instead of movement primitives (Schaal et al., 2003). We proposed the learning system extracts primitives from observed behaviors, which are represented by a linear dynamical model. Then, based on the recognized primitives, the system generates imitated behaviours that are feasible for its own body. Parameters of the linear dynamical models are determined from the captured human behaviors and ground reaction force (GRF) measured simultaneously. Therefore, this method has only a small dependency on the classification criteria defined by an experimenter. We also consider balance control by estimating GRF from the captured human's motions. GRF is rescaled by weight ratio between an instructor and a robot. This procedure is consistent with the idea of mind-reading, the ability to estimate another individual's mental state by adopting own perspective. Estimation of dynamics is similar to estimation of mental state because human has inner dynamics with environment. We use GRF to generate proper interaction between robots and environments to imitate the instructor's behaviors.

3) Feature space to represent of task

We use key features on low dimensional space of task are extracted from human learning. The key features are defined as critical point of accomplishment skill. Robot can easily learn task by using these key features because it is easy to find direction which improve task performance on low dimensional space. The key features are extracted as correlation space from between human training via point data and performance measurement of task. Robot optimize robot's via point by using robot's performance based on the key features.

1.3 Related work

In this section, we describe previous studies related to our three imitation learning framework.

1)

First, we show related studies which convert human motion capture data to non-human characters on CG or robots. In imitation learning for robot, they solve this kinematics problem by defining the correspondence of robot's joints and human's, however it is dependent on experimenter view. This has been studied on CG as retargetting problem. While a number of algorithmic techniques have been developed for animating human characters, most of them are not applicable to non-human characters because they assume that the target character has human-like proportions and topology (Lee & Shin, 1999; Popovic & Witkin, 1999; Choi & Ko, 2000; Shin, Lee, Shin, & Gleicher, 2001), An exception is the work by Gleicher (Gleicher, 1998), where he extended his motion retargetting technique to non-human characters by explicitly specifying the correspondence of body parts in the original and new characters. In theory, Simulation- and physics-based techniques can handle any skeleton model and common tasks such as locomotion and balancing (Witkin & Kass, 1988; Liu & Popovic, 2002; Macchietto, Zordan, & Shelton, 2009). However, they are typically not suitable for synthesizing complex behaviors with specific styles due to the difficulty in developing a wide variety of controllers for characters of different topology.
(Ikemoto, Arikan, & Forsyth, 2009) use artist's input to learn a mapping function based on Gaussian processes from a captured motion to a different character's motion. Their method requires that another animation sequence, edited from the original motion capture data, is provided to learn the map-

ping function. Such input gives much richer information about the correspondence than the isolated key poses used in our work. Bregler et al. (Bregler, Loeb, Chuang, & Deshpande, 2002) also developed a method that transfers a 2D cartoon style to different characters. Our method employs a statistical model called shared Gaussian process latent variable models (shared GPLVM) (Ek, Torr, & Lawrence, 2007) to map a human pose to a character pose. Shon et al. (Shon, Grochow, Hertzmann, & Rao, 2005) used a shared GPLVM to map human motion to a humanoid robot with many fewer degrees of freedom, however this robot configuration is not different from human. In our work, robot model has more different configuration from human. Feng et al. (Feng, Kim, & Yu, 2008) used Kernel CCA to map control point to facial mesh. Kernel CCA is similar to shared GPLVM. Their method can find suitable shared deformation spaces to control high dimensional facial mesh movement. Urtasun et al. (Urtasun, Fleet, Geiger, Popovic, Darrell, & Lawrence, 2008) developed a method to incorporate explicit prior knowledge into GPLVM, allowing synthesis of transitions between different behaviors and with spacetime constraints. Grochow et al. (Grochow, Martin, Hertzmann, & Popovic, 2004) used another extension of GPLVM (scaled GPLVM) to bias the inverse kinematics computation to a specific style. In our work, we use a small set of key poses, rather than sequences, to learn a mapping function that covers a wide range of behaviors. We believe that it is much easier for actors and animators to create accurate character poses than to create appealing motion sequences, and that the dynamics, or velocity information, can best come from the actor's captured motion.

2)

Next, we describe work about imitation learning uses movement primitives and consider dynamics. The movement primitives can be considered as a method to convert recognized behaviors to imitated movements that can be executed on imitator's body dynamics. One of the popular approaches to design an imitation learning framework that uses primitive representations is using a hidden Markov model (HMM). However, in most of the studies using HMMs (Inamura, Tanie, & Nakamura, 2003; Takano, Yamane, Sugihara, Yamamoto, & Nakamura, 2006), the primitive representations are manually defined while our method extracts primitives by using the Gaussian mixture of linear dynamical model. Thus, performance of imitation learning can be highly depends on the experimenter's intuition for the primitive extraction. In addition, these studies only focus on imitating demonstrated joint trajectories and do not consider balance control.

Dynamic movement primitive (DMP) (Ijspeert, Nakanishi, & Schaal, 2003) is becoming a popular imitation learning framework. DMP is a method to design dynamics that can generate an imitated movement. DMP focuses on generating one observed trajectory and modulating the trajectory by manipulating parameters.

When we apply an imitation learning method to a robot that is not fixed to the ground such as a humanoid robot, we need to consider balance control. In (Nakaoka et al., 2003) and (Grimes, Chalodhorn, & Rao, 2006), balance control is taken into account while humanoid robots are imitating a dancing behavior or one-leg balancing behavior. However, these methods design specific imitated movements to the given tasks. Thus, it is not easy to generalize the acquired movements to other behaviors or robots have different dynamical properties.

In computer graphics and vision research fields, segmentation and recognition of observed behaviors is popular research topic (Barbic, Safonova, Pan, Faloutsos, Hodgins, & Pollard, 2004; Pavlovic, Rehg, & MacCormick, 2000; Li, Wang, & Shum, 2002).

In (Yamane & Nakamura, 2003; Macchietto et al., 2009; Muico, Lee, Popovic, & Popovic, 2009), optimization methods are used to convert observed behaviors to physically consistent behaviors.

Using modular architecture to represent brain functions have been proposed by, e.g., (Gomi & Kawato, 1993; Haruno, Wolpert, & Kawato, 2001; Wolpert, Doya, & Kawato, 2003; Doya, Samejima, ichi Katagiri, & Kawato, 2002; Samejima, Katagiri, Doya, & Kawato, 2002). The modular selection and identification for the control (MOSAIC) is one of the popular models (Haruno et al., 2001; Wolpert et al., 2003). MOSAIC is originally proposed to explain functions of cerebellum that has modular architecture. MOSAIC uses linear forward and inverse dynamics to select the module and use the module to control a target object. On the other hand, applications of MOSAIC model have been limited to simple mechanical systems such as a mass-spring-dumper system. One of the reasons is that MOSAIC uses inverse dynamics that are not easy to be estimated in the real environments. Our proposed imitation learning framework dose not use Inverse dynamic and only uses forward kinematics that is easily estimated.

3)

Finally, we describe work related to skill transfer. One imitation learning framework of the skill transfer is Dynamic movement primitive (DMP) (Ijspeert et al., 2003). DMP is a method to design dynamics model of task. DMP can be changed goal point or trajectory depend on kinematics of robot or task (Pastor, Hoffmann, Asfour, & Schaal, 2009). Similarly, (Atkeson & Schaal, 1997) show that learning from demonstration of balancing pole based on learning a task model and a reward function. In that way, the method of skill transfer combining imitation learning and reinforcement learning have been focused in recent years (Guenter, Hersch, Calinon, & Billard, 2007; Bitzer, Howard, & Vijayakumar, 2010). However, these framework were not applied to complex task. On the other hand, (Miyamoto, Schaal, Gandolfo, Gomi, Koike, Osu, Nakano, Wada, & Kawato, 1996; Miyamoto & Kawato, 1998) show representation of via point to work Kendama and tennis, which is extracted human movement trajectory using a forward-inverse relaxation model and is treated as a control variable. As other usage of representation of via point, (Ude, Riley, Nemec, Kos, Asfour, & Cheng, 2007) show that proposed framework can imitate throwing ball task from human demonstration. Our proposed framework is based on (Morimoto, Hyon, Atkeson, & Cheng, 2008; Bitzer et al., 2010), we use human motion capture data and find a low dimensional space representation that preserve human task performance.

1.4 Outline of the thesis

This thesis is organized as follows. Chapter 2 describes how to find shared low-dimensional latent space between demonstrator's and imitator's. In this chapter, we apply this framework to generate motion of non-human character on CG. We apply the method to three characters and six emotional movements. In chapter 3, we present our imitation learning framework, which estimate ground reaction force from captured demonstrator's movement so that an imitator can generate physically consistent imitated behaviors. We show how to generate an imitated behaviors based on the recognized primitives, and how we apply the proposed method to a four and seven -link simulated robot model. We also present the generalization performance of our framework on the four link models with different physical parameters. Chapter 4 describes how to find critical point of task accomplishment by using via point and task performance of human. In this chapter, we apply this method to Kendama task. Finally, In chapter 5, we describe the conclusions and future work.

Chapter 2

Common Low-Dimensional Space between Humans and Non-Human Characters

In this chapter, we introduce a method for generating whole-body, motion of non-human character from human motion capture data without relying on big different configuration. Examples of such non-human characters and snap-shots of their motions are shown Figure 2.1 along with the human motions from which the motions are synthesized. Such characters are often inspired by animals or artificial objects, and their limb lengths, proportion's and even topology may be significantly different form humans. We use these non-human characters as an extreme example of robot which has big different configuration. The characters are expected to be anthropomorphic, i.e., convey expressions through body language understandable to human viewers, rather than moving as real animals.

Keyframing has been almost the only technique available to animate such characters. Although data driven techniques using human motion capture data are popular for human animation, most of them do not work for non-human characters because of the large differences between the skeletons and motions styles of the actor and the character. Capturing motion of the animal does not help solve the problem because animals can not behave like human. Another possible approach is physical simulation, but it is is very difficult to build controllers that generate plausible and stylistic motions.

To create the motion of a non-human character, we first capture motions for a human subject acting in the style of the target character. The subject then selects a few key poses from the captured motion sequence and creates corresponding character poses on a 3D graphics software system. The remaining steps can be completed automatically with little user interaction. The key poses are used to build a statistical model for mapping an human pose to a character pose. We can generate a sequence of poses by mapping every frame of the motion capture sequence using the mapping function. Finally an optimization process adjust the fine details of the motion such as contact constraints and physical realism. We evaluate our approach by comparing to principal component analysis(PCA), nearest neighbors(NN), and Gaussian processes(GP), and verify that our method produces more plausible results.

Compared to keyframe animation, our method significantly reduces the time and cost required to create animations of non-human characters. In our experiment, our method uses two hours for a motion capture session, 18 hours for selecting and creating key poses, and 70 minutes of computation time to generate 18 animations (7 minutes in total) of three characters, while an animator can spend weeks to crate the same amount of animation by keyframing.

2.1 Overview

Figure 2.2 shows an overview of our proposed motion synthesis process. The rectangular blocks indicate manual operations, while the rounded rectangles are automatic operations.

We first capture motions of a trained actor or actress performing in the style of the target character. We provide instructions about the capability and characteristics of the character and then rely on the actor's talent to portray how the character would act in a particular situation.

The actor then selects a few keyposes among the capture motion sequences. the poses should be selected so that they cover and are represen-

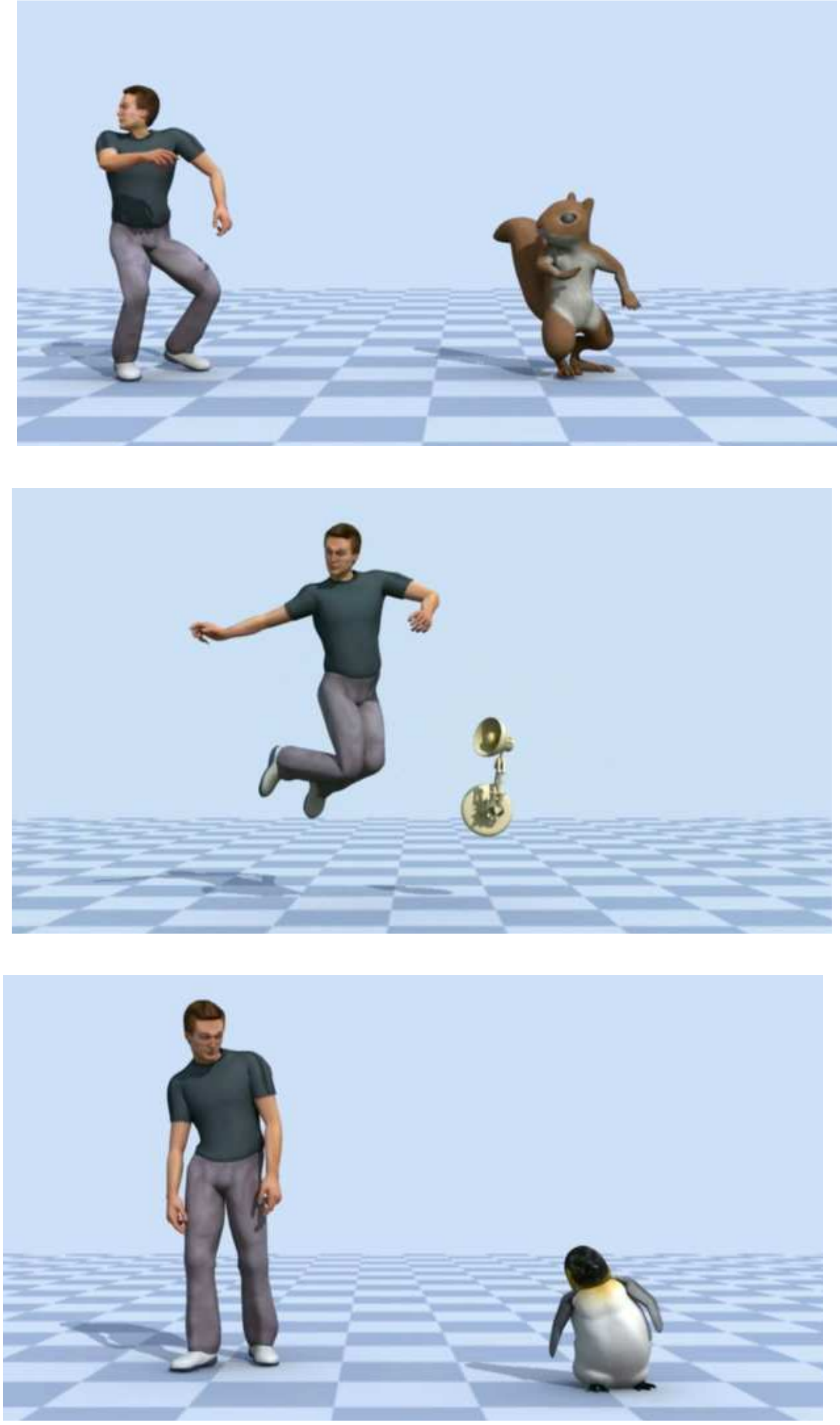

Figure 2.1: Non-human characters animated using human motion capture data and human motion capture data.

tative of the space of the poses that appear in the captured motions. The last task for the actor is to create a character pose corresponding to each of the selected key poses. If necessary, an animator can operate a 3D graphics software system to manipulate the character's skeleton. This process is difficult to automate because the actor often has to make intelligent decisions to , for example, realize the same contact states on characters with completely different limb length. The actor may also want to add poses that are not possible for the human body, such as an extreme back bend for a character that is much more flexible that humans.

The key poses implicitly define the correspondence between the body parts of the humanoid and character models, event if the character's body has different topology. The remaining two steps can be completed automatically without any user interaction. First we build a statistical model to map the human poses in each frame of the captured motion data to a character pose using the given key poses in Section 2.2. In this thesis, we focus on this static mapping. We then obtain the global transformation of the poses by matching the linear and angular momenta of the character motion to that of the human motion. In many cases, there are still a number of visual artifacts in the motion such as contact points penetrating the floor or floating in the air. We therefore fine tune the motion by correcting the contact point positions and improving the physical realism through an optimization process taking into account the dynamics of the character. We describes these ways of dynamics optimization in Section 2.3.

2.2 Static mapping

We employ a statistical method called shared Gaussian latent variable model (shared GPLVM) (Ek et al., 2007; Shon et al., 2005) to learn a static mapping function from a human pose to a character pose which has big different configuration. Shared GPLVM is suitable for our problem because human poses and corresponding character poses will likely have some underlying nonlinear relationship. Moreover,shared GPLVM gives a probability distribution over the character poses, which can potentially be used for adjusting the character pose to satisfy other constraints.

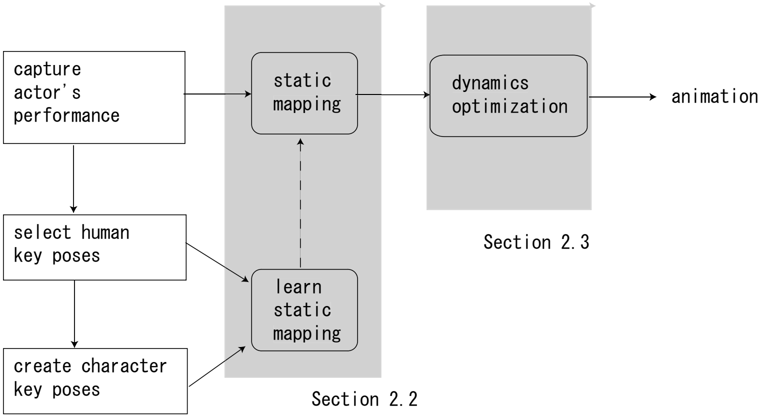

Figure 2.2: Overview of the system. Rectangular blocks indicate manual operations and rounded rectangles are processed automatically.

Shared GPLVM is an extension of GPLVM (Lawrence, 2003), which models the nonlinear mapping from a low dimensional space (latent space) to an observation space. Shared GPLVM extends GPLVM by allowing multiple observation spaces sharing a common latent space. The main objective of using shared GPLVM in previous work is to limit the output space with ambiguity due to, for example, monocular video (Ek, Rihan, Torr, Rogez, & Lawrence, 2008). Although our problem does not involve ambiguity, we adopt shared GPLVM because we only have a sparse set of corresponding key poses. We expect that there is a common causal structure between human and character motions. In addition, it is known that a wide variety of human motions are confined to a relatively low-dimensional space (Safonova, Hodgins, & Pollard, 2004). A model with a shared latent space would be an effective way to discover and model the space that represents that underlying structure.

Our mapping problem involves two observation spaces: the D_Y -dimensional human pose space and the D_Z dimensional character pose space. These spaces are associated with a D_X -dimensional latent space. In contrast to the existing techniques that use time-series data for learning a model, the main challenge in our problem is that the given samples are very sparse compared to the complexity of the human and character models.

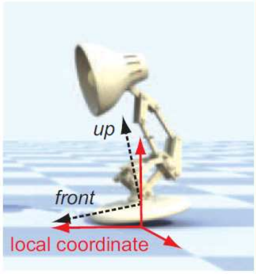

Figure 2.3: The local coordinate frame (shown in solid, red line) for representing the feature point positions.

2.2.1 Motion retargetting

There are several options to represent poses of human and character models. In our implementation, we use the Cartesian positions of multiple feature points on the human and character bodies, as done in some previous work (Arikan, 2006). For the human model, we use motion capture markers because marker sets are usually designed so that they can well represent human poses. Similarly, we define a set of virtual markers for the character model by placing three markers on each link of the skeleton, and use their positions to represent character poses.

The Cartesian positions must be converted to a local coordinate frame to make them invariant to global transformations. In this paper, we assume that the height and roll/pitch angles are important features of a pose, and therefore only cancel out the horizontal position and yaw angle. For this purpose, we determine a local coordinate frame to represent the feature point positions.

The local coordinate is determined based on the root position and orientation as follows Figure 2.3. We assume that two local vectors are defined for the root joint: the front and up vectors that point in the front and up directions of the model. The position of the local coordinate is simply the projection of the root location to a horizontal plane with a constant height. The z axis of the local coordinate points in the vertical direction. The x axis faces the heading direction of the root joint, which is found by first obtaining the single axis rotation to make the up vector vertical, and then applying the same rotation to the front vector. The y axis is chosen to form a right-hand system.

For each key pose i, we form the observation vectors y_i and z_i by concatenating the local-coordinate Cartesian position vectors of the feature points of the human and character models, respectively. We then collect the vectors for all key poses to form observation matrices Y and Z. We denote the latent coordinates associated with the observations by X.

2.2.2 Learning and mapping

The learning and mapping processes are outlined in Figure 2.4, where the inputs are drawn with a black background. In the learning process, the parameters of the GPLVMs and the latent coordinates for each key pose are obtained by maximizing the likelihood of generating the given pair of key poses. In the mapping process, we obtain the latent coordinates for each motion capture frame that maximize the likelihood of generating the given human pose. The latent coordinates are then used to calculate the character pose using GPLVM.

An issue in shared GPLVM is how to determine the dimension of the latent space. We employ several criteria as detailed in 2.4.3 for this purpose.

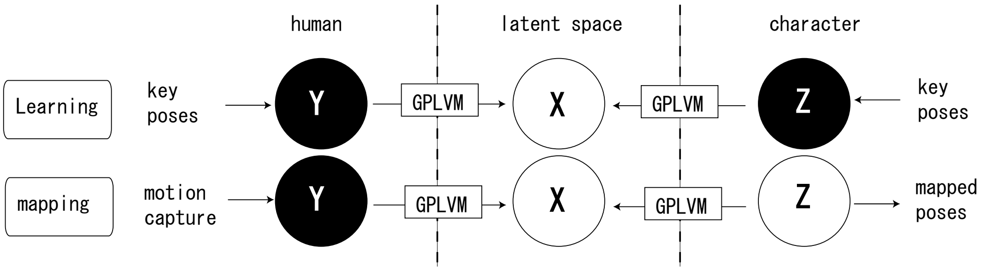

Figure 2.4: Outline of the learning and mapping processes. The inputs are drawn with black background.

Learning

A GPLVM (Lawrence, 2003) parametrizes the nonlinear mapping function from the latent space to observation space by a kernel matrix. The (i, j) element of the kernel matrix K represents the similarity between two data points x_i and x_j , and is calculated by

$$K_{ij} = k(x_i, x_j) = \theta_1 \exp\left(-\frac{\theta_2}{2}\|x_i - x_j\|^2\right) + \theta_3 + \beta^{-1}\delta_{i,j}, \tag{2.1}$$

where $\Phi = \{\theta_1, \theta_2, \theta_3, \beta\}$ are the model parameters and δ represents the delta function. We denote the parameters of the mapping functions from latent space to human pose by Φ_Y and from latent space to character pose by Φ_Z

Assuming a zero-mean Gaussian process prior on the functions that generates the observations from a point in the latent space, the likelihoods of generating the given observations are formulated as

$$P(Y|X, \Phi_Y) = \frac{1}{\sqrt{(2\pi)^{ND_Y}|K_Y|^{D_Y}}} \exp\{-\frac{1}{2}\sum_{k=1}^{D_Y} y_k^T K_Y^{-1} y_k\}, \tag{2.2}$$

$$P(Z|X, \Phi_Z) = \frac{1}{\sqrt{(2\pi)^{ND_Z}|K_Z|^{D_Z}}} \exp\{-\frac{1}{2}\sum_{k=1}^{D_Z} z_k^T K_Z^{-1} z_k\}, \tag{2.3}$$

where K_Y and K_Z are the kernel matrices calculated using (2.1) with Φ_Y and Φ_Z respectively, and y_k and z_k denote the k-th dimension of the observation matrices Y and Z respectively. Using these likelihoods and priors for Φ_Y, Φ_Z and X, we can calculate the joint likelihood as

$$P_{GP}(Y, Z|X, \Phi_Y, \Phi_Z) = P(Y|X, \Phi_Y)P(Z|X, \Phi_Z)P(\Phi_Y)P(\Phi_Z)P(X). \tag{2.4}$$

Learning shared GPLVM is essentially an optimization process to obtain the model parameters Φ_Y , Φ_Z and latent coordinates X that maximize the joint likelihood. The Latent coordinates X and parameters Φ_Y, Φ_Z are optimized by conjugate gradient method. It is important to find appropriate initial value on gradient method. The latent coordinates are initialized using Kernel Canonical Correlation Analysis (KCCA) (Akaho, 2006)(see Appendix B)

After the model parameters Φ_Z are learned, we can obtain the probability distribution of the character pose for given latent coordinates x by

$$\bar{z}(x) = \mu_Z + Z^T K_Z^{-1} k(x) \tag{2.5}$$

$$\sigma_Z^2(x) = k(x,x) - k(x)^T K_Z^{-1} k(x) \tag{2.6}$$

where $\bar{z}$ and σ_Z^2 are the mean and variance of the distribution respectively, μ_Z is the mean of the observations, and $k(x)$ is a vector whose i-th element is $k_i(x) = k(x, x_i)$.

Mapping

The mapping process starts by obtaining the latent coordinates that correspond to a new human pose using a method combining nearest neighbor search and optimization (Ek et al., 2007). For a new human pose y_{new}, we search for the key pose y_i with the smallest Euclidean distance to y_{new}. We then use the latent coordinates associated with y_i as the initial value for the gradient-based optimization process to obtain the latent coordinates $\hat{x}$ that maximize the likelihood of generating y_{new}, i.e.,

$$\hat{x} = argmax_x P(y_{new}|x, Y, X, \Phi_Y) \tag{2.7}$$

The optimization process converged in all examples we have tested. We use the latent coordinates $\hat{x}$ to obtain the distribution of the character pose using (2.5),(2.6).

2.3 Dynamics optimization

The sequence of poses obtained so far does not include the global horizontal movement. It also does not preserve the contact constraints in the original human motion because they are not considered in the static mapping function. Dynamics optimization is performed in three steps to solve these issues. We first determine the global transformation of the character based on the linear and angular momenta of the original human motion. We then correct the contact point positions based on the contact information. Finally, we improve the physical plausibility by solving an optimization problem based on cost function that have the equations of motion of the character, a penalty-based contact force model, and the probability distribution given by the

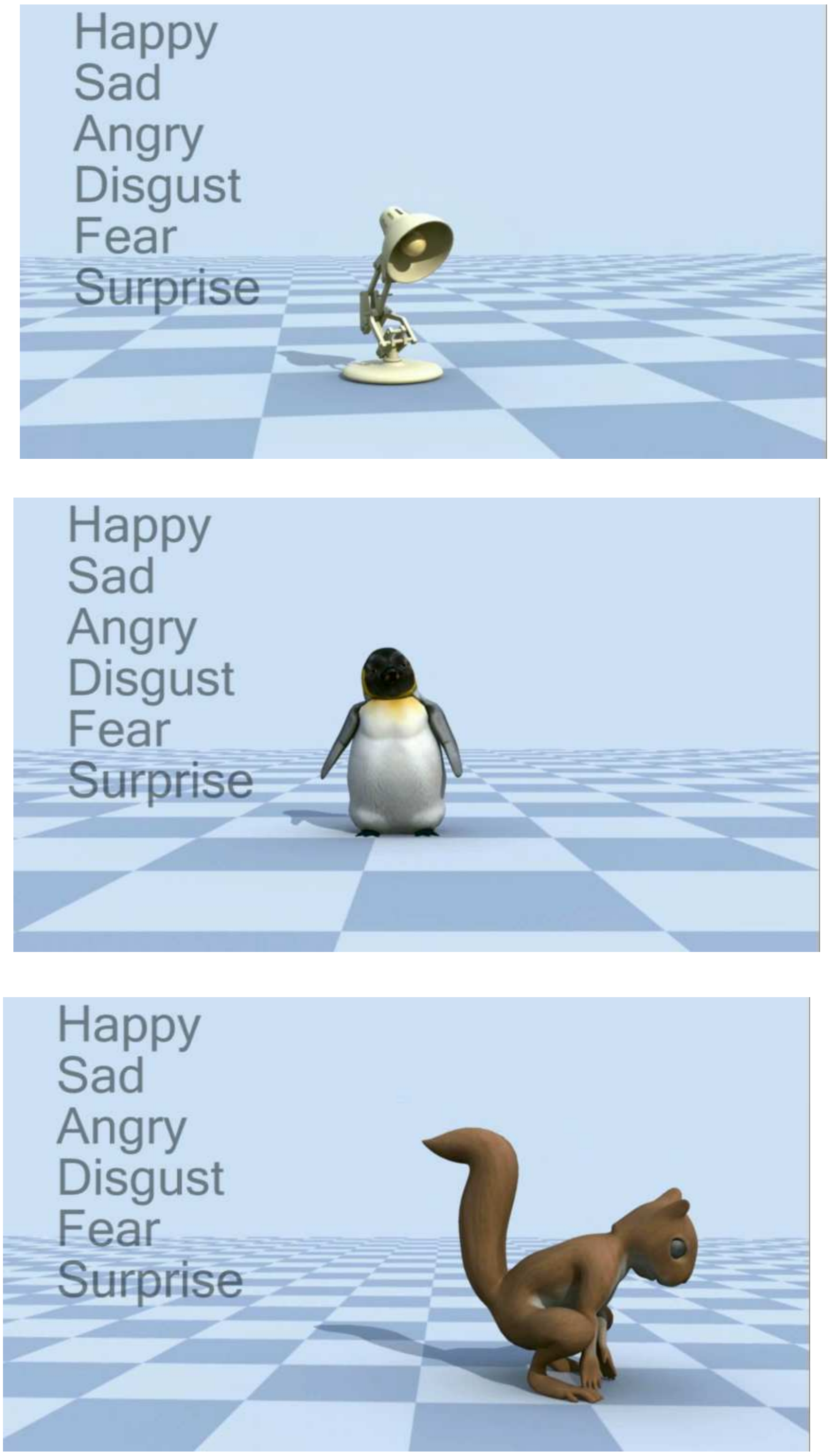

Figure 2.5: Three characters used for the experiment: lamp, penguin and squirrel.

static mapping function. The modulated trajectory Δz is defined as:

$$\Delta z(t) = \sum_{i=1}^{N} w_i \varphi_i(t) \tag{2.8}$$

$\varphi_i(t)$ is RBF function. Mean and variance of RBF function is fixed. w_i is optimized as parameter. A penalty based contact force model is simply described as :

$$F = -k_p y - k_d \dot{y} \tag{2.9}$$

F is contact force which calculated by character's foot position y from floor. Parameter w_i is optimized so that trajectories are satisfied with this contact force. Dynamics optimization is described more detail in Appendix C C.2 2. After the optimization, we can generate imitated character's motion including physical constraint.

2.4 Simulation results

We prepared three characters for the tests: lamp, penguin, and squirrel (see Fig. 2.5). The lamp character is an example of character inspired by an artificial object but yet able to perform human-like expressions using the arm and lamp shade as body and face. The completely different topology and locomotion style from humans make it difficult to animate the character. The penguin character has human-like topology but the limbs are extremely short with limited mobility. Although it still does biped walking, its locomotion style is also very different from humans because of its extremely short legs. The squirrel character has human-like topology but may also walk on four legs. The tail is occasionally animated during the key pose creation process, but we do not extensively animate the tail in the present work.

The software system consists of three components.

- An in-house C++ code library for reading motion capture data and key poses, converting them to feature point data, computing the inverse kinematics, and evaluating the cost function.
- A publicly available MATLAB implementation of the learning and mapping functions of shared GPLVM (Lawrence, 2003).
- MATLAB code for evaluating the cost function and performing the optimization using the MATLAB function lsqnonlin.

emotion	lamp		penguin		squirrel	
anger	18.6	16	14.3	3	22.9	14
disgust	34.4	1	23.3	7	20.0	3
fear	29.7	2	25.2	4	28.5	18
happiness	20.1	7	23.7	11	25.8	9
sadness	19.5	3	29.6	4	26.7	3
surprise	10.7	1	26.1	4	19.2	5
total	10.7	30	152.2	33	143.1	52

Table 2.1: Statistics of the measured and created data. Each column show the duration of the sequence in seconds(left) and the number of key poses selected from each sequence(right).

2.4.1 Manual tasks

We recorded the motions of a professional actor expressing six emotions (anger, disgust, fear, happiness, sadness and surprise) for each of the three characters. Before the motion capture session, we showed a picture of each character and verbally explained the kinematic properties (e.g., no or extremely short legs, may walk on four legs). The capture session lasted about two hours.

The actor and an animator worked together with a 3D graphics software system (Maya) to select key poses from the motion capture data and create corresponding poses for the characters. It took 18 hours in total to select and create 115 key poses from 18 motion sequences, which averaged approximately 9 minutes per pose. The average interval between key poses in the motions is 3.6 seconds, which is obviously much longer than the interval in standard keyframe animations. Table 2.1 summarizes the statistics of the data obtained through this process. Table 2.2 show the number of marker on each character. Thus, D_Y -dimension of human pose space is 156, and Each D_Z dimension of character pose space are 60 (lamp), 121 (penguin), and 792 (squirrel).

2.4.2 Static mapping

We trained a shared GPLVM for each character using the key poses created by the actor and animator. An issue in using GPLVM is how to determine the

	lamp	penguin	squirrel	human
the number of marker	20	107	264	52

Table 2.2: The number of marker on each character. Marker has three coordinate values.

dimension of the latent coordinate, D_X . We use Kernel CCA to determine D_X.
We evaluate generalization error by 1 leave out cross-validation. We use Kernel CCA to initialize the latent coordinates, this initial value is important to determine the dimension of the latent coordinate in shared GPLVM. We define generalization error as mean squared error of test data's marker. We use Gauss kernel ($\theta_1 = 1, \theta_3 = 0, \beta = 0$) in (2.1), Gaussian process (GP) can reconstruct from latent coordinates. $\sqrt{\theta_2^{-1}}$ is standard deviation, and modulate generalization error. About squirrel data, we get low dimensional data $D_Z = 20$ to decrease computational requirements by using PCA because the dimension of squirrel data is very high.

Figure 2.6 shows result of cross-validation on each character. The error improved by increasing D_X up to 6 (generalization error = 1.5983) in the case of lamp. Similarly, minimum error of penguin is at $D_X = 8$ (generalization error = 6.2062), optimal dimension for the latent space on squirrel data is $D_X = 5$ (generalization error = 40.727).

Desired property is that the character motion becomes continuous when the human motion is continuous. Figure 2.7 shows the first two dimensions of the trajectories in the latent space when a human motion capture sequence (happy lamp) is input to the models when we used 6 dimensional spaces and 10 standard deviation. We therefore can represent generated trajectories are continuous based on 6 dimensional shared spaces.

After static mapping, the character stays above a fixed point on the floor because the horizontal movement has been removed from the key poses before learning the mapping function. The dynamics optimization tries to match the two forces by modifying the trajectory, which changes both the required and actual contact forces. Figure 2.8 show snapshots of the original human poses and the results of final animation after dynamics optimization.

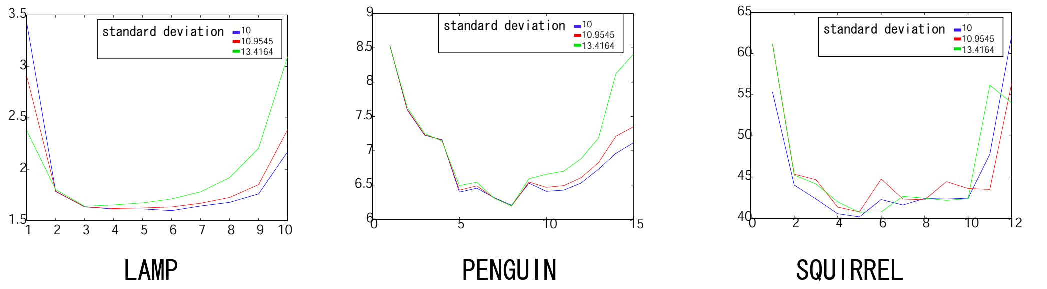

Figure 2.6: Generalization error by using 1 leave out cross-validation. Standard deviation of Gauss kernel make a small change error.

2.4.3 Comparative discussion

We show additional examples of the synthesized animations in the supplemental movie.

We compare shared GPLVM with the following mapping techniques:

- Principal component analysis and linear mapping (PCA): We obtain the 30-dimensional spaces that contain human and character key poses using principal component analysis, and then obtain a linear mapping between the two low-dimensional spaces.
- Linear interpolation of nearest neighbors (NN): We find the N nearest key poses using the Cartesian distance of the vector composed of human marker data, and obtain the weighted sum of the character feature point data where the weight is inversely proportional to the distance. We have tested with N = 3 (NN3) and N = 10 (NN10).
- Gaussian Process (GP): We obtain the Gaussian process model that maximizes the likelihood of generating the character key poses from the human key poses. We then use the mean of the distribution obtained by each motion capture data frame as the output of the model.

We use the happy lamp example and the mapped motions are processed by the dynamics optimization algorithm to obtain the final results.

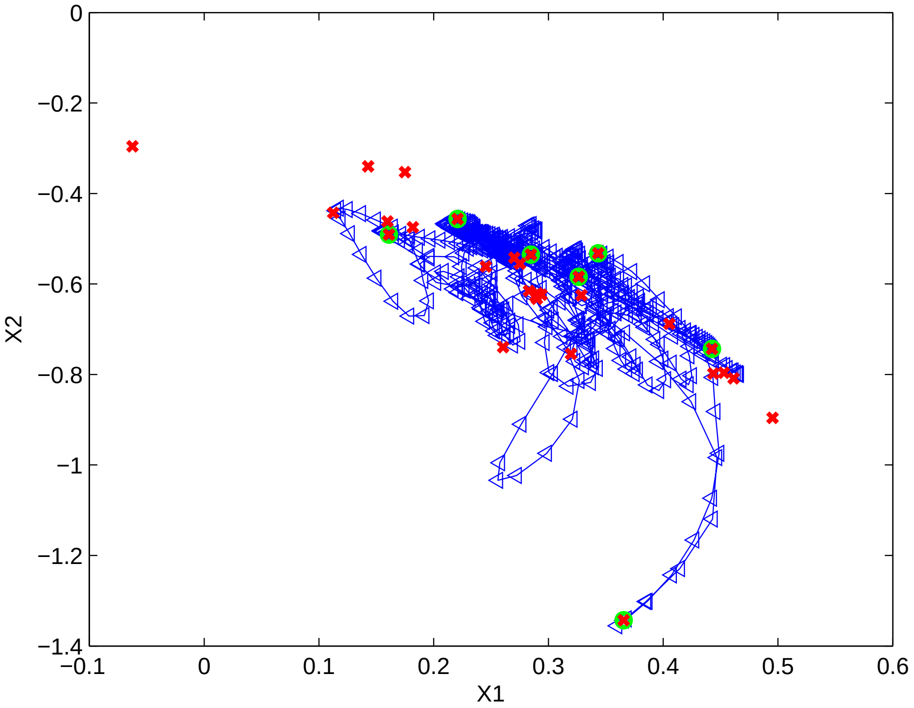

Figure 2.7: Trajectories in the latent space when a human motion capture sequence is input to the model, projected to the first two dimensions of the latent space. The trajectory is represented by a line with triangles that denote the latent coordinates of individual frames. The (green) circles represent the key poses from the same motion sequence as the trajectory, and (red) crosses are the key poses from other motions.

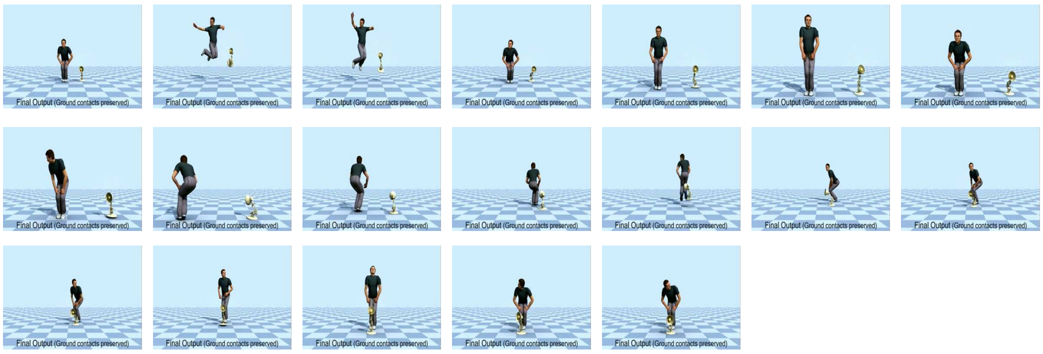

Figure 2.8: Original human motion and after static mapping, dynamics optimization character motion.

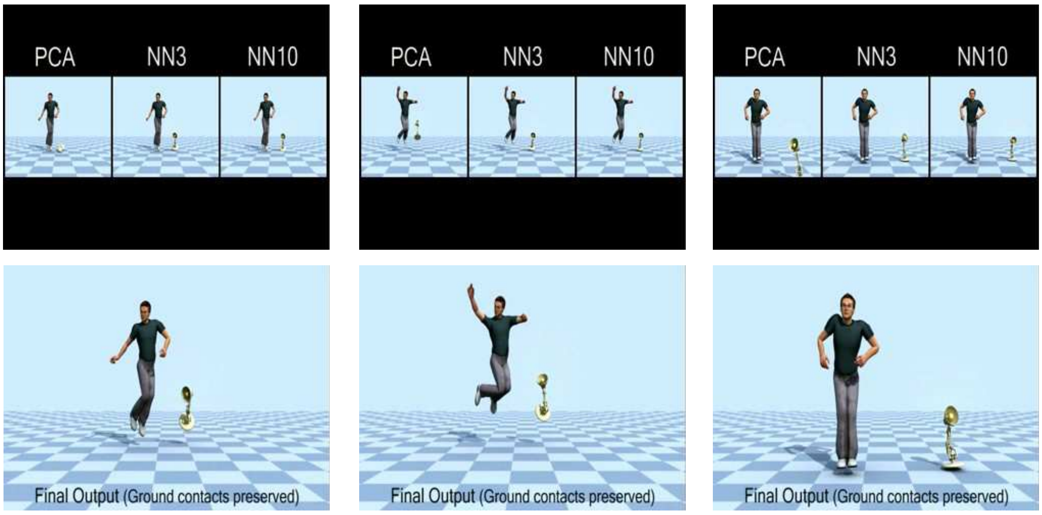
PCA
NN3
NN10
PCA
NN3
NN10
PCA
NN3
NN10
Final Output (Ground contacts preserved)
Final Output (Ground contacts preserved)
Final Output (Ground contacts preserved)

Figure 2.9: Comparison static mapping between shared-GPLVM and PCA,NN3, and NN10.

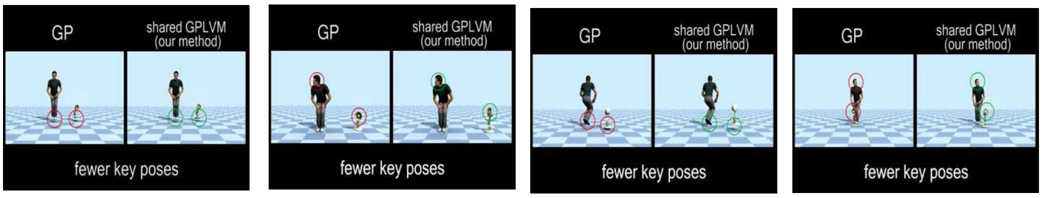
GP
shared GPLVM (our method)
fewer key poses
GP
shared GPLVM (our method)
fewer key poses
GP
shared GPLVM (our method)
fewer key poses
GP
shared GPLVM (our method)
fewer key poses

Figure 2.10: Comparison static mapping from fewer key poses between shared-GPLVM and GP.

Simple models (PCA and NN) cannot effectively model the nonlinear relationship between the human and character pose spaces with sparse examples (see Fig. 2.9). NN works better especially with 10 neighbors but does not generalize well enough. The GP model produces results similar to the shared GPLVM, but slightly less dynamic. This issue could be fixed by applying the dynamics optimization to many DOFs but it may significantly modify the actor's original intention.

We also test the effect of the number of key poses by using GP and shared GPLVM learned from fewer key poses (see Fig. 2.10). We use the lamp model and remove the seven key poses from the happy motion for learning, and synthesize the happy lamp motion. Neither models can predict the extreme poses during the first high jump, which is expected because we do not have any example close to that pose. However, the shared GPLVM seem to have better generalization capability in terms of jump heights and head directions as demonstrated in the movie.

We test generalization error about the some example. We calculate generalization error by using one leave out cross-validation (see Table 2.3). This result show shared-GPLVM have minimal error in Lamp and Penguin and Squirrel by using low dimensional initial value in section . Generalization error is too depend on the type and number of training data and initial value of kernel's parameter. In the case of few training data, NN method show better result than shared-GPLVM. We should decide threshold of number of training data by using generalization error with each character. Also, we need to explore kernel parameter to acquire good model.

Finally, we acquire several examples of the synthesized emotional and

	PCA	NN10(N=10)	NN3(N=3)	GP	S-GPLVM
Lamp	0.33	0.23	0.21	0.23	0.173
Penguin	0.85	0.62	0.57	0.84	0.51
Squirrel	2.93	2.33	2.11	1.78	1.69

Table 2.3: Generalization error with each method.

dance motions. The lamp and penguin dances are synthesized using only the key poses shown in Table 2.1, while we provided six more key poses to synthesize the squirrel dance.

2.5 Conclusion

The basic idea of this approach is to leverage the creativity of actors in imagining how non-human characters should move, in contrast to the current standard animation procedure where animators have creative control via 3D graphics software. The use of motion capture also opens up a second pool of talent as actors are skilled at using their bodies to tell a story or convey an emotion. The problem with non-human characters is that we have to map human motion to characters with significantly different proportions or topologies.

We tested our algorithm on three characters with different levels of similarity to humans. Although the squirrel model is the closest to humans in terms of the proportions and mobility, motions of the lamp and penguin models look more plausible and expressive. We hypothesize that this result occurs because the set of key poses for the squirrel model is not sufficient to cover its wide range of motion, and because viewers do not expect as much expression from the simple bodies of the lamp and penguin.

On the other hand, we focus on what low dimensional spaces represent. We measured penguin's and squirrel's one of low dimensional space which has maximum eigen value by adding small value change, and compare reconstructed character motion with test data. In the result, we show one of low dimensional spaces of squirrel represents pitch angle, penguin's represents up and down. To extend this method, we need to find more meaningful low dimensional spaces, for example low dimensional spaces which represent emotion. Meaningful low dimensional spaces will be reused for other character's.

We show only result on CG animation, however it is easy to apply this method to robot model because these CG character's have link structure.

Chapter 3

Movement Primitives for Imitation Learning

In this chapter, we introduce our imitation learning framework that the learning system first extracts and recognizes primitives from observed behaviors, where each primitive is represented by a linear dynamical model. Then, based on the recognized primitives, the system generates imitated behaviours that are feasible with its own body. Parameters of the linear dynamical models are determined from the captured human behaviors and simultaneously measured ground reaction force (GRF). We use the GRF profiles generated by the instructor rather than calculating GRF profiles that are consistent with joint angle trajectories after capturing behaviors. This calculation of consistent GRF profiles requires extra computation (Yamane & Nakamura, 2003), while we can easily and simultaneously measure the GRF profiles while capturing joint angle trajectories. However, GRF is not directly observable from visual images, and it is difficult to always use a device to measure tri-axial GRF for the overall range of human movements. Also, measurement of an accurate acceleration that needs to estimate GRF is rather difficult due to the observation noise.

Therefore, we represent primitives containing the GRF by using linear dynamical models. We then use switching state-space models (SSSMs) (Ghahramani & Hinton, 2000) to recognize the appropriate primitives that corresponds to the observed human behavior. Simultaneously, SSSMs are used to estimate joint angle, joint angle velocity, and GRF from the noisy observation data.

To evaluate our proposed method, we apply our imitation learning framework to recognize human squatting motions and generate the squatting motions using a four and seven-link simulated robot model.

3.1 Imitation learning framework using movement primitives

In our learning system, we first construct a motion database that collects joint angles $\boldsymbol{\theta}$, joint angle velocities $\dot{\boldsymbol{\theta}}$, and GRF $\mathbf{f}_{grf}$ from various humans behaviors by using a motion capture system and a set of force plates. Then, apply our imitation learning system to recognize newly observed human behaviors and generate the imitated movement using robots:

1. We extract primitives by learning the parameter of the linear dynamical models with using the constructed motion database (see Appendix A).

2. The extracted primitives are used in switching state-space models (SSSMs) to recognize newly observed motion sequences. Simultaneously, SSSMs estimate joint angles $\boldsymbol{\theta}^d$, joint angle velocities $\dot{\boldsymbol{\theta}}^d$, and the ground reaction force $\mathbf{f}_{grf}$ (GRF) from the noisy observation of joint angles $\boldsymbol{\theta}^o$ and joint angle velocities $\dot{\boldsymbol{\theta}}^o$.

3. To generate imitated movements, first, the estimated GRF is scaled by the weight ratio between an instructor and a robot. Then, feed forward joint torques $\boldsymbol{\tau}_F$ to generate the scaled GRF $\mathbf{f}^d_{grf}$ are derived. To track the estimated joint angles $\boldsymbol{\theta}^d$ and joint angle velocities $\dot{\boldsymbol{\theta}}^d$, we also use a proportional-derivative (PD) controller where the output of the controller is denoted as $\boldsymbol{\tau}_J$. The estimated joint angles $\boldsymbol{\theta}^d$ and joint angle velocities $\dot{\boldsymbol{\theta}}^d$ are used as the desired values for the PD controller.

Figure 3.1 shows a schematic diagram of the proposed method.

In the following sections, we explain how we extract and recognize primitives and how we generate the imitated behaviors.

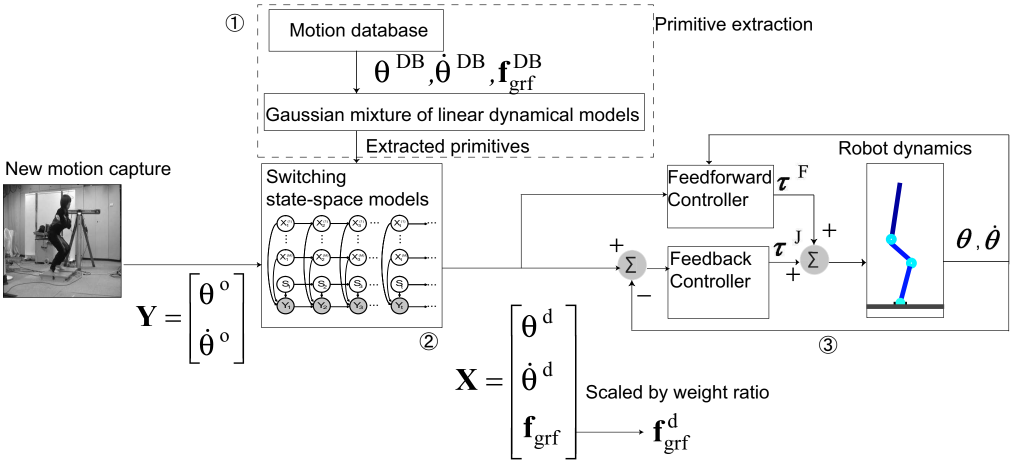
①
Motion database
$\theta^{DB}, \dot{\theta}^{DB}, \mathbf{f}_{grf}^{DB}$
Primitive extraction
Gaussian mixture of linear dynamical models
Extracted primitives
New motion capture
Switching state-space models
$\mathbf{Y} = \begin{bmatrix} \theta^o \\ \dot{\theta}^o \end{bmatrix}$
②
Feedforward Controller
$\boldsymbol{\tau}^F$
Robot dynamics
Feedback Controller
$\boldsymbol{\tau}^J$
Σ
+
−
$\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}$
③
$\mathbf{X} = \begin{bmatrix} \theta^d \\ \dot{\theta}^d \\ \mathbf{f}_{grf} \end{bmatrix}$
Scaled by weight ratio
$\mathbf{f}_{grf}^{d}$

Figure 3.1: Proposed imitation learning framework: ① Primitives are extracted from motion database and switching state-space models are constructed by using the primitives. ② State variables (joint angles $\boldsymbol{\theta}^d$, joint angle velocities $\dot{\boldsymbol{\theta}}^d$, and GRF $\mathbf{f}_{grf}$ are estimated using switching state-space models for new observation. The desired GRF $\mathbf{f}_{grf}^d$ is derived from the estimated GRF $\mathbf{f}_{grf}$ by considering weight ration between an instructor and a robot. ③ The robot model is controlled based on the estimated variables by using feedforward controller and feedback controller.

3.1.1 Extraction and recognition of primitives

We first extract primitives from captured human behaviors by using the Gaussian mixture of linear dynamical models (see Appendix A). We define the one linear dynamical model as one primitive. Then, a newly observed human behavior is recognized as a combination of these primitives by using switching state-space models (SSSMs) (Ghahramani & Hinton, 2000).

3.1.2 Switching state space models

The schematic diagram of SSSMs is depicted in Fig. 3.2. This model can be viewed as a combination of Hidden Markov Model (HMM) with a set of linear dynamical systems. The model has multiple Markov chains of continuous linear-Gaussian latent variables $\mathbf{X}$, together with a Markov chain of discrete variables $\mathbf{S}$ of the form used in a HMM.

We try to estimate the M continuous latent vectors $\mathbf{X}_t^{(m)}(m = 1, ..., M)$, and the discrete latent state $\mathbf{S}_t$ from the sequence observation $\mathbf{Y}_t(t = 1, ..., T)$.

The joint probability of observations and hidden state can be factored as

$$P(\{\mathbf{S}_t, \mathbf{X}_t^{(1)}, ..., \mathbf{X}_t^{(M)}, \mathbf{Y}_t\}) = \\ P(\mathbf{S}_1) \prod_{t=2}^{T} P(\mathbf{S}_t|\mathbf{S}_{t-1}) \prod_{m=1}^{M} P(\mathbf{X}_1^{(m)}) \prod_{t=2}^{T} P(\mathbf{X}_t^{(m)}|\mathbf{X}_{t-1}^{(m)}) \prod_{t=1}^{T} P(\mathbf{Y}_t|\mathbf{X}_t^{(1)}, ..., \mathbf{X}_t^{(M)}, \mathbf{S}_t), \tag{3.1}$$

where $\pi = P(\mathbf{S}_1)$ is the initial state probability of the discrete latent state, and $\Phi = P(\mathbf{S}_t|\mathbf{S}_{t-1})$ is the transition matrix. The linear Gaussian dynamical model of each continuous latent vector can be represented as

$$P(\mathbf{X}_{t+1}^{(m)}|\mathbf{X}_t^{(m)}) \quad = \quad \mathcal{N}(\mathbf{X}_{t+1}^{(m)}|\mathbf{A}^{(m)}\mathbf{X}_t^{(m)}, \mathbf{Q}^{(m)}), \tag{3.2}$$

where $\mathcal{N}(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes Gaussian distribution with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$.

The output at each time step is determined by stochastically selecting the output from one of the linear Gaussian dynamics according to the state of the discrete latent variable. Here we assume that the observation noise is Gaussian. Therefore, once the m-th linear dynamical model is selected, the observation can be represented as

$$P(\mathbf{Y}_t|\mathbf{X}_t^{(1)}, ..., \mathbf{X}_t^{(M)}, \mathbf{S}_t = i) = \mathcal{N}(\mathbf{Y}_t|\mathbf{C}^{(m)}\mathbf{X}_t^{(m)}, \mathbf{R}). \tag{3.3}$$

3.1.3 Recognition of primitives

The primitives are recognized by finding the parameters of SSSMs. These parameters can be derived by generalizing the EM algorithm. The EM algorithm alternates between optimizing a distribution over hidden states in E-step, and optimizing the parameter given the distribution over hidden states in M-steps. In E-step, however, it is intractable to exact distribution inference in SSSMs. Therefore, variational approximation is used to find the posterior distribution of the continuous and discrete variables with estimated parameters. For this purpose, tractable distribution $\mathcal{Q}$ is introduced, where $\mathcal{Q}$ is defined by the parameter ξ (Ghahramani & Hinton, 2000):

$$\mathcal{Q}(\mathbf{S}_t, \mathbf{X}_t|\xi) = \frac{1}{Z_Q}\left[\psi(\mathbf{S}_1|\xi) \prod_{t=2}^{T} \psi(\mathbf{S}_{t-1}, \mathbf{S}_t|\xi)\right] \prod_{i=1}^{M} \psi(\mathbf{X}_1^{(i)}|\xi) \prod_{t=2}^{T} \psi(\mathbf{X}_t^{(i)}, \mathbf{X}_{t-1}^{(i)}|\xi), \tag{3.4}$$

ψ are unnormalized probabilities. The posterior can be approximated by varying the parameter ξ to obtain the tightest possible bound by minimizing Kullback-Leibler (KL) divergence between $\mathcal{Q}$ and the true posterior P:

$$KL(\mathcal{Q}||P) = \sum_{\mathbf{S}_t} \int \mathcal{Q}(\mathbf{S}_t, \mathbf{X}_t|\xi) \log \frac{\mathcal{Q}(\mathbf{S}_t, \mathbf{X}_t|\xi)}{P(\mathbf{S}_t, \mathbf{X}_t|\mathbf{Y}_t)} d\mathbf{X}_t. \tag{3.5}$$

This can be done by using the Kalman smoothing recursion and forward-backward algorithm. Since each state estimation mutually depends on the parameter ξ, the estimation processes are repeated until the KL divergence in (3.5) converges. The M-step optimizes the parameters of both the linear dynamical systems and the discrete hidden states to increase the expectation of the log likelihood. In this study, we focus on optimizing the parameters of discrete hidden states while we use the parameter of the linear dynamical systems that are extracted from the motion database by the method described in Appendix A.

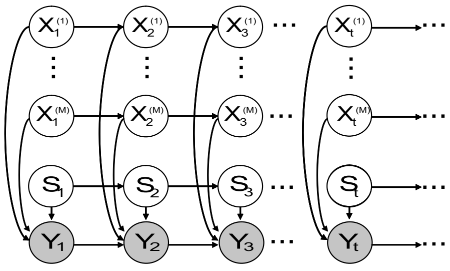

Figure 3.2: Switching state-space models. $\mathbf{X}_t^{(m)}$ denotes the state vector of a linear model m at time t. $\mathbf{S}_t$ is the discrete switching state. $\mathbf{Y}_t$ denotes the observation vector.

3.1.4 Generating joint torques from estimated data

By using SSSMs, we estimate joint angles $\boldsymbol{\theta}^d$, joint angle velocities $\dot{\boldsymbol{\theta}}^d$, and the GRF $\mathbf{f}_{grf}$. The estimated GRF $\mathbf{f}_{grf}$ is scaled by the weight ratio between the instructor and the robot to derive the desired GRF $\mathbf{f}_{grf}^d$. We then use the method proposed in (Hyon, Hale, & Cheng, 2007) to derive joint torque $\boldsymbol{\tau}_F$ at each joint to generate the desired GRF $\mathbf{f}_{grf}^d$.

	link1	link2	link3	link4
mass[kg]	39.4	13.2	6.12	1.32
length[m]	0.85	0.38	0.44	0.36
inertia moment($\times 10^{-4}$)[$kg.m^2$]	2.370	0.159	0.987	0.720

Table 3.1: Parameters of four-link robot model (total mass: 60kg).

	link1	link2	link3	link4	link5	link6	link7
mass[kg]	39.4	6.6	3.06	0.66	6.6	3.06	0.66
length[m]	0.85	0.38	0.44	0.4	0.38	0.44	0.4
inertia moment($\times 10^{-4}$)[$kg.m^2$]	2.370	0.079	0.049	0.008	0.079	0.049	0.008

Table 3.2: Parameters of seven-link robot model (total mass: 60kg).

The joint torque $\boldsymbol{\tau}_F$ to generate the estimated GRF $\mathbf{f}^d_{grf}$ can be calculated by the following equation (Hyon et al., 2007):

$$\boldsymbol{\tau}_F = \mathbf{J}_P^T(\boldsymbol{\theta})(-\mathbf{f}^d_{grf}) + \mathbf{D}\dot{\boldsymbol{\theta}}, \tag{3.6}$$

where $\mathbf{J}_P$ is the Jacobian from the Center of Mass (COM) to the contact point with the ground (see Fig. 3.3(b)) . $\mathbf{D}$ is dumping coefficient. We adopt a proportional-derivative (PD) controller to track the estimated joint angles. The output of the PD controller $\boldsymbol{\tau}_J$ is given according to the estimated desired joint angles $\boldsymbol{\theta}^d$ and joint angle velocities $\dot{\boldsymbol{\theta}}^d$:

$$\boldsymbol{\tau}_J = \mathbf{K}_p(\boldsymbol{\theta}^d - \boldsymbol{\theta}) + \mathbf{K}_d(\dot{\boldsymbol{\theta}}^d - \dot{\boldsymbol{\theta}}), \tag{3.7}$$

with positive gain matrices $\mathbf{K}_p, \mathbf{K}_d > 0$. Therefore, the total joint torque input is given by:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_F + \boldsymbol{\tau}_J. \tag{3.8}$$

3.2 Simulation

3.2.1 Simulation setup

We apply our proposed method to a simulated four and seven-link robot(see Fig. 3.3(a) and Fig. 3.4(a)). Physical parameters of each robot model are

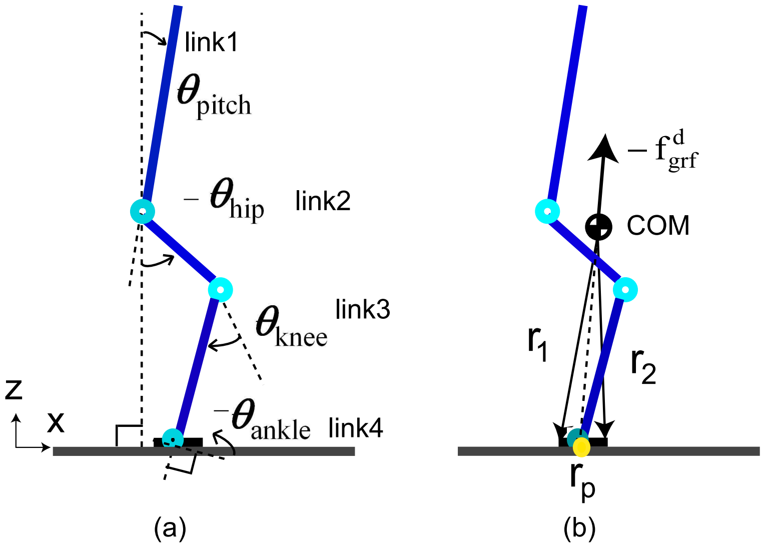

Figure 3.3: (a): Four-link robot model (b): $\mathbf{r}_1$ and $\mathbf{r}_2$ represent contact position from the Center of Mass(COM). $\mathbf{r}_p$ represents the center of pressure. $\mathbf{f}^{d}_{grf}$ is the desired ground reaction force.

described in Table 3.1 and Table 3.2. It has the same total mass to the instructor's weight. The weight ration of each link was extracted from anatomical data of human body (e.g., (Hukashiro, Sakurai, Hirano, & Ae, 2000)). The subject who is considered as the instructor in the proposed imitation learning framework is asked to demonstrate **two kinds of squat motions with knee amplitudes of 45 and 90 degrees**, and **movement back and forth**. The demonstrated motions were measured using a motion capture system and force plates. Using this measurement system, joint angles, joint angle velocities, and the GRF were measured. We measured five squat cycles for each squat motion and five cycles for movement back and forth. Average squat frequency was about 0.5 Hz. Three of the five cycles were used as training data to determine the parameters of the linear models. The other two cycles were used as test data for the proposed imitation learning method. Therefore, we had four different squat cycles as test data. We denoted the test data of the 45-degree knee amplitude by **45a** and **45b**, while the test data of the 90-degree knee amplitude are denoted by **90a** and **90b**.

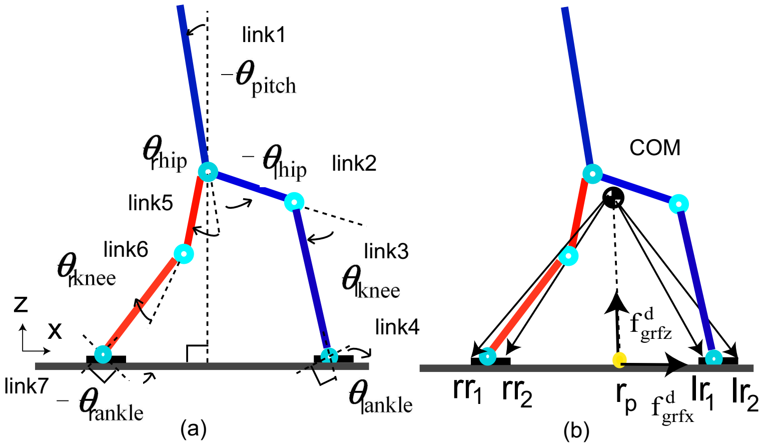

Figure 3.4: (a): Seven-link robot model (b): $\mathbf{rr}_1$, $\mathbf{rr}_2$, $\mathbf{lr}_1$ and $\mathbf{lr}_2$ represent contact position from the Center of Mass(COM). $\mathbf{r}_p$ represents the center of pressure. $\mathbf{f}^d_{grfx}, \mathbf{f}^d_{grfz}$ is the desired ground reaction force on horizontal and vertical axes.

We selected two of these four test squatting cycles and used these combined behaviors as observed motions to evaluate the proposed imitation learning system. Thus, we get $16 = {}_4\Pi_2$ sets of observed trajectories. In the case of movement back and forth, we get one test data.

We only consider the generation of squat motions in a two-dimensional sagittal plane. Latent continuous state variable $\mathbf{X}$ in (3.2) is defined as joint angles, joint angle velocities, and the GRF (see Fig. 3.3,Fig. 3.4):

$$\mathbf{X} = (\theta_{ankle}, \theta_{knee}, \theta_{hip}, \dot{\theta}_{ankle}, \dot{\theta}_{knee}, \dot{\theta}_{hip}, f_x, f_z)^T, \tag{3.9}$$

where f_x, f_z is the horizontal and vertical component of GRF $\mathbf{f}_{grf}$. Each joint angle has left left and right legs. In the squat motion, we used average joint angles and joint angle velocities of the left and right legs for simplicity. We consider the case only the joint angles and the joint angle velocities can be observed. Therefore, output matrix $\mathbf{C}$ is a 6×8 matrix in the squat motion. In the movement back and forth, output matrix $\mathbf{C}$ is a 12×14 matrix The element of this matrix is $\mathbf{C}_{ij} = \delta_{ij}$, where δ_{ij} is the Kronecker delta.

We use three linear models ($M = 3$) to represent all squatting behaviors. To make the learning procedure stable, we adopt the parameters of the linear models **A** and **Q** derived in the extraction process of primitives (see Appendix1 A). Covariance of the observation noise **R** is derived from training data. In SSSMs, we estimate **X** and **S** while parameters (π, Φ) are optimized.

Then, we derive joint torques using (3.8) based on the estimated joint angles, joint angle velocities, and GRF of the four and seven -link robot model. The PD gains in (3.7) are set as $(\mathbf{K}_p, \mathbf{K}_d) = (100\mathbf{I}, 90\mathbf{I})$, where **I** is identity matrix in the squat motion. In the movement back and forth, $(\mathbf{K}_p, \mathbf{K}_d) = (30\mathbf{I}, 15\mathbf{I})$

3.2.2 Result

Recognition of squat movements

The continuous latent state variables including GRF and the discrete latent state are successfully estimated for all 16 observed trajectories. Figure 3.5 shows the estimated trajectories when new motion (**45a→90a** and **90a→45a**) are observed. This result indicates that model 1 corresponds to 45 degree squat movement, model 2 corresponds to 90 degree squat movement, and model 3 corresponds to motionless sequences.

Recognition of movement back and forth

The continuous latent state variables including GRF(f_x, f_z) and the discrete latent state are successfully estimated for observed trajectories. In the figure 3.6, we shows the estimated trajectories when new movement back and forth are observed. This result indicates that model 1 corresponds to movement forth, model 2 corresponds to movement back. There is no sequences corresponds to model 3 in this test data.

Generation of squat movements

The four-link robot model could generate successful squatting motions without falling over for all $16 = ({}_4\Pi_2)$ combined squatting motions. Table 3.3 shows the mean squared error of joint angle trajectories:

$$\frac{1}{N}\sum_{i}^{N} \|\boldsymbol{\theta}_i^o - \boldsymbol{\theta}_i\|^2 , \tag{3.10}$$

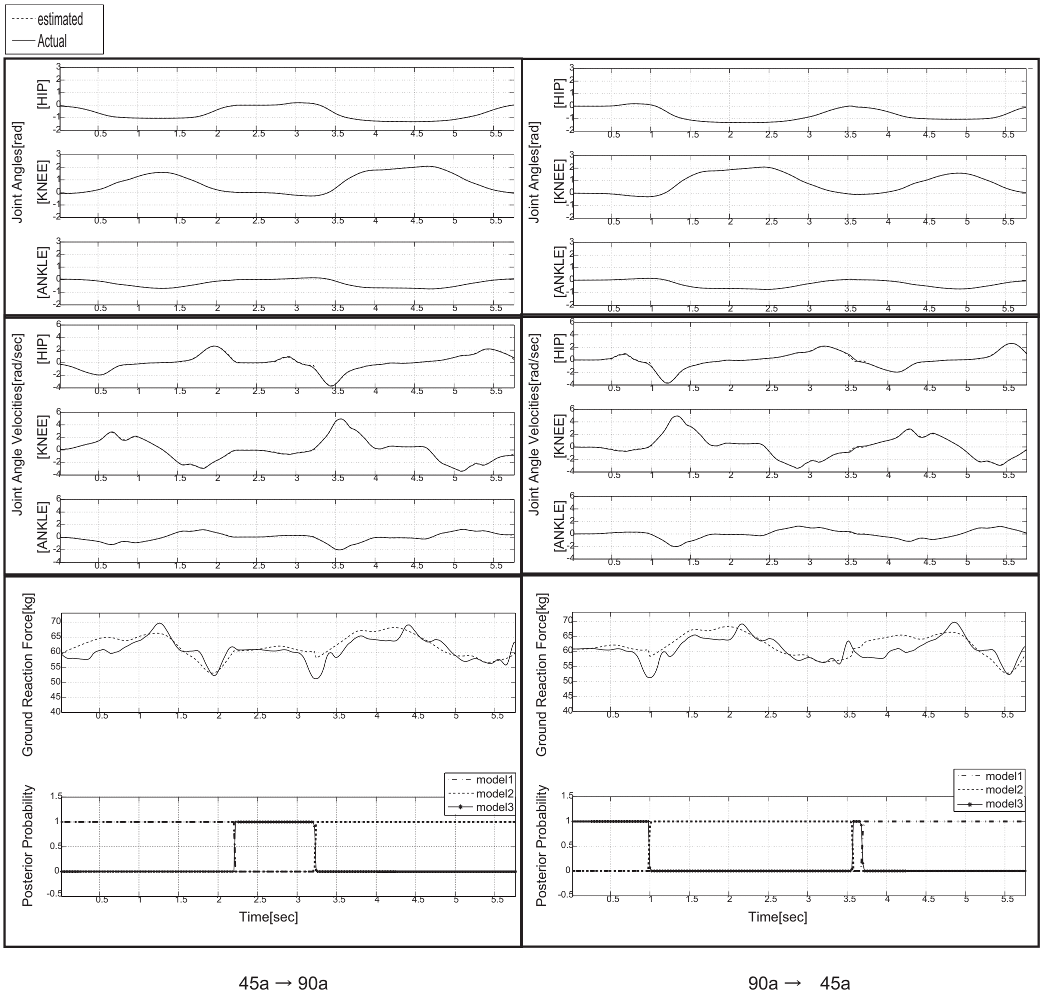

Figure 3.5: Estimated trajectories, switching state for data sets **45a→90a** and **90a→45a**: The solid line show the measured trajectories, and the dashed lines show the estimated trajectories. (Top) Joint angles (hip, knee, ankle). (Middle) Joint angle velocities (hip, knee, ankle). (Bottom) Ground reaction force (f_z), and approximated probability of the discrete switching state $\mathbf{S}_t$.

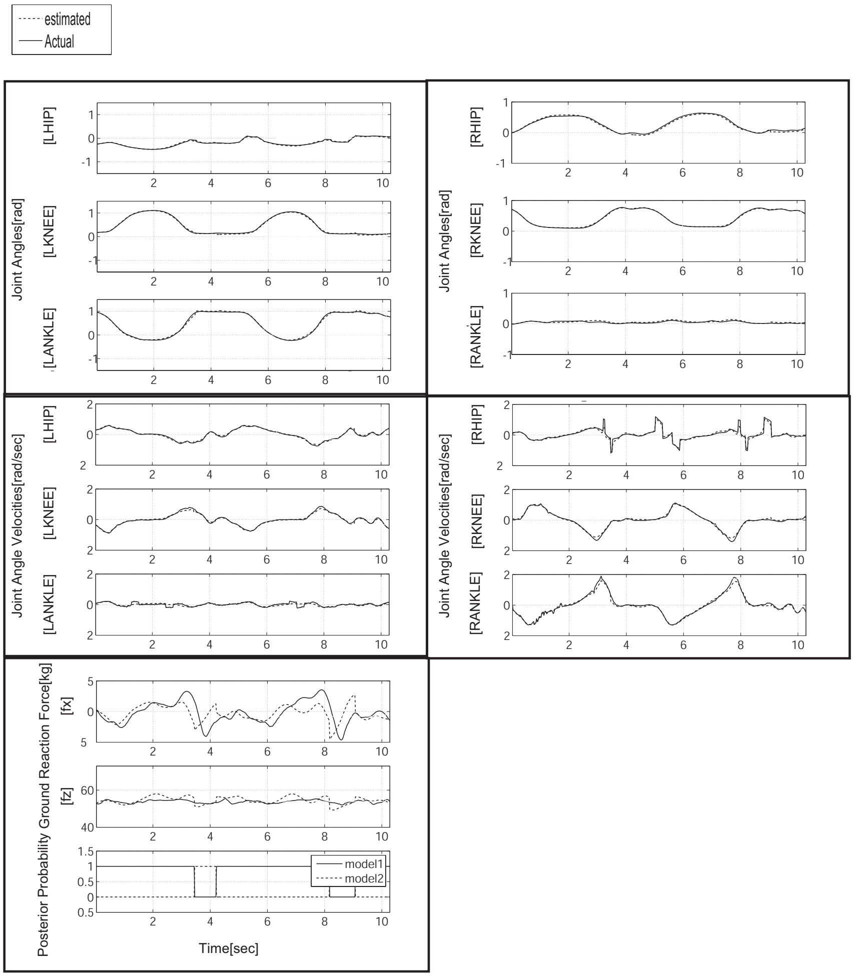

Figure 3.6: Estimated trajectories, switching state for data set movement back and forth: The solid line show the measured trajectories, and the dashed lines show the estimated trajectories. (Top) Joint angles of left and right leg (lhip, lknee, lankle, rhip, rknee, rankle). (Middle) Joint angle velocities of left and right leg (lhip, lknee, lankle, rhip, rknee, rankle). (Bottom) Ground reaction force (f_x, f_z), and approximated probability of the discrete switching state $\mathbf{S}_t$.

Observed squat motion	MSE(deg^2)	
45a→90a	0.73	
45a→90b	1.13	
45a→45a	0.60	
45a→45b	0.51	Minimum
45b→90a	0.75	
45b→90b	1.05	
45b→45a	0.52	
45b→45b	0.52	
90a→45a	0.72	
90a→45b	0.80	
90a→90a	1.90	Maximum
90a→90b	0.86	Average
90b→45a	0.90	
90b→45b	1.00	
90b→90a	0.84	
90b→90b	0.97	
Average	0.86	

Table 3.3: Tracking errors for squat movements derived from (3.10). The maximum error was observed with the squat motion **90a** → **90a**. The minimum error was observed with the squat motion **45a** → **45b**. For the squat motion **90a** → **90b**, the tracking error was closest to the average.

for the 16 successful squatting motions, where N is the number of samples. The average number of samples was 282, and this corresponds to 5.6 s. $\boldsymbol{\theta}_i^o$ are the observed joint angles, and $\boldsymbol{\theta}_i$ are the generated joint angles of the robot model using (3.8) (see Fig. 3.1). The maximum error was observed when the observed squat motion was **90a** → **90a**. The minimum error was observed when the observed squat motion was **45a** → **45b**. In both case, the four-link robot model imitated the observed instructors movements comparatively well without falling over. We depict deviations of generated squat motions from the observed motions in Fig. 3.7.

Comparison with a simple controller

We compare control performance of the proposed method with that of a simple PD controller. Figure 3.8 and Figure 3.9 show results of this comparison with two observed data sets: **45a→90a**, and **90a→45a**. By using the pro-

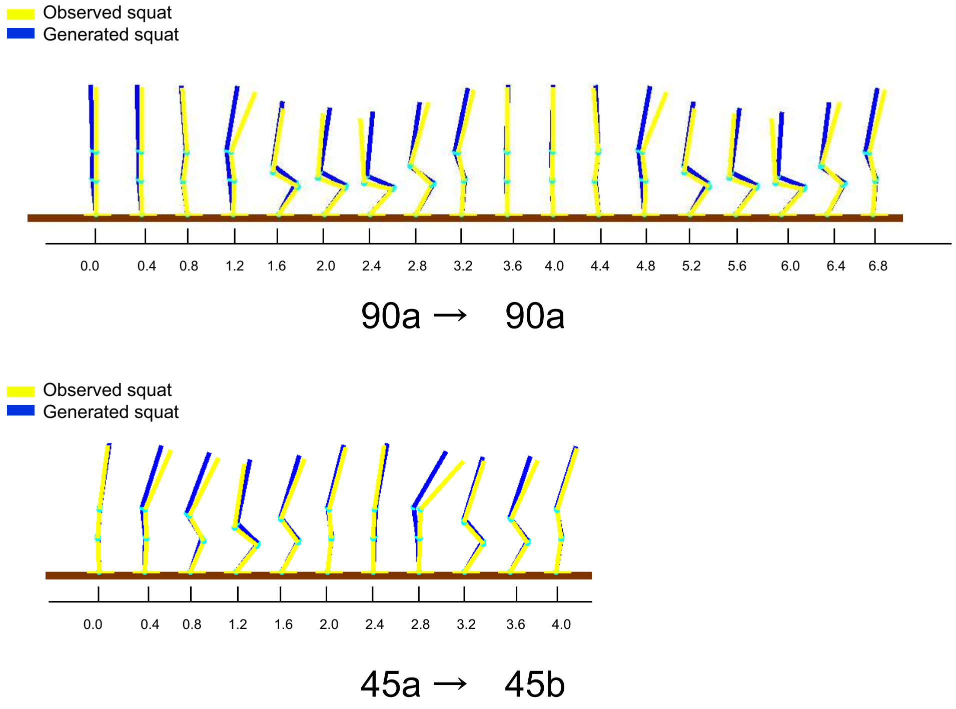

Figure 3.7: Comparison between human squat movements and generated squat movements. Gray colored model shows observed squat motions. Black colored model shows generated squat motions. (Top) Squat motion generated with the observed trajectory **90a** → **90a**. With this data set, the trajectory error was maximum among the 16 test data. (Bottom) Squat motion generated with the observed trajectory **45a** → **45b**. With this data set, the trajectory error was minimum among the 16 test data.

posed method given in (3.8), successful squat motions were generated for the both estimated trajectories. On the other hand, by only using simple PD controller given in (3.7), the robot fell over in both cases. These results indicate an advantage of estimating and using GRF in an imitation learning framework.

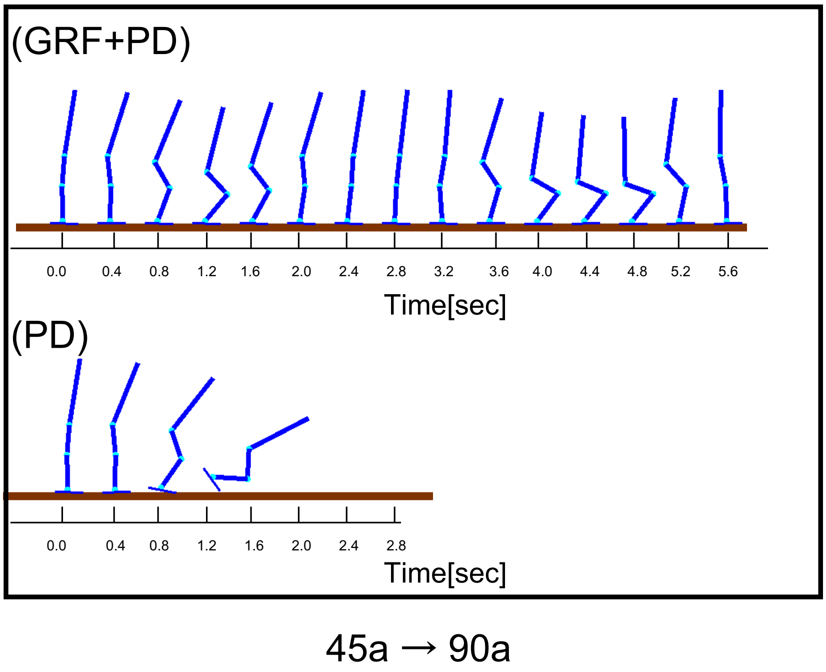

Figure 3.8: Comparison between control performance of the proposed method and that of the simple PD controller. Generated squat movements for the observed trajectory **45a→90a**. By only tracking the desired joint trajectories with the PD controller, the robot fell over. The results show effectiveness of using the estimated GRF.

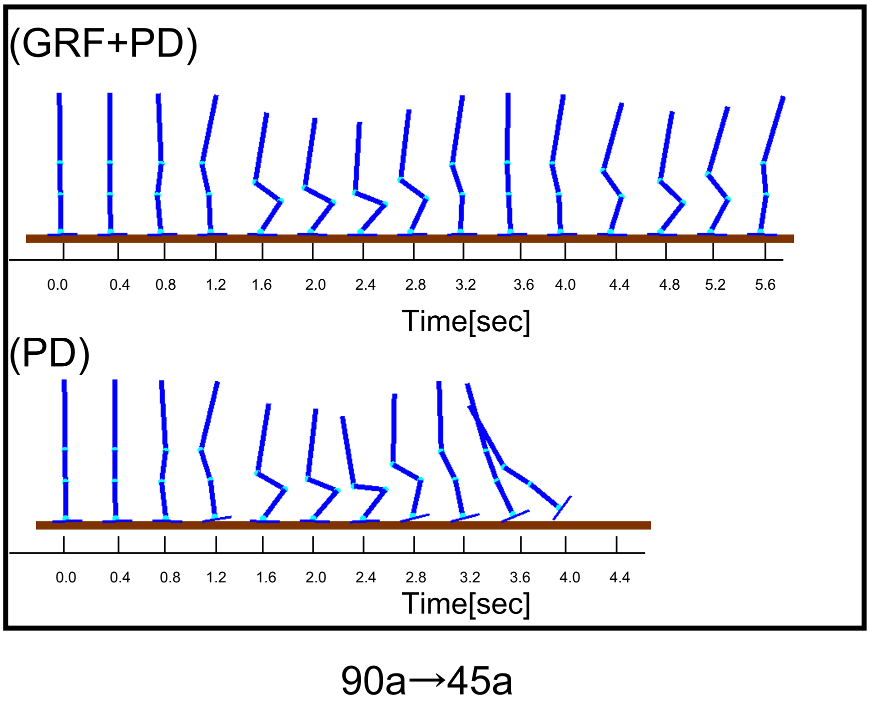

Figure 3.9: Comparison between control performance of the proposed method and that of the simple PD controller. Generated squat movements for the observed trajectory **90a→45a**. By only tracking the desired joint trajectories with the PD controller, the robot fell over. The results show effectiveness of using the estimated GRF.

Availability of ground reaction force in horizontal

We verify value of horizontal ground reaction force f_x. We compare control performance of the proposed method for considering f_x. Figure 3.10 show results of this comparison with observed data sets: new movement back and forth. By considering f_x and f_z in (3.8), successful back and forth motion was generated for the estimated trajectory. On the other hand, by only considering f_z given in (3.8), the robot fell over. These results indicate estimating and using horizontal GRF f_x work well in some cases including movement back and forth.

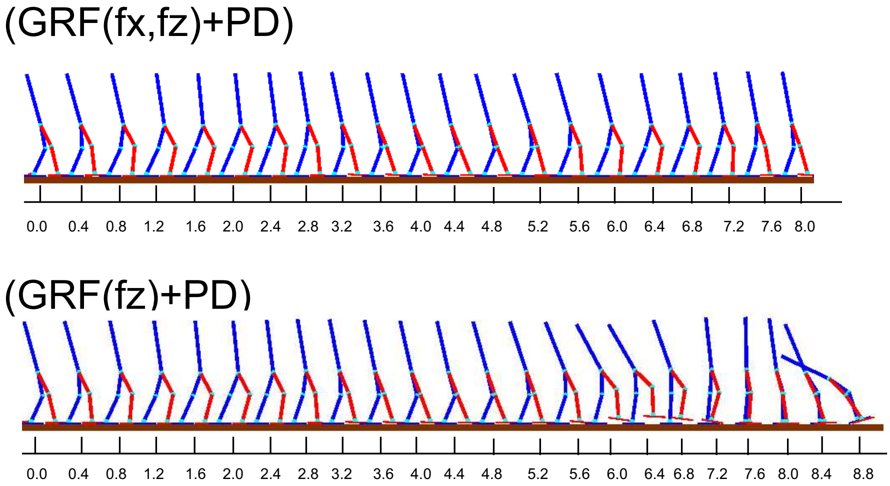

Figure 3.10: Comparison between control performance of controller using f_x and f_z and that of controller only using f_z. Generated squat movements for the observed trajectory movement back and forth. By tracking the desired joint trajectories with controller using only the estimated GRF f_z, the robot fell over. The results show availability of using the estimated GRFf_x in movement back and forth.

Evaluation of generalization performance

Different weight

Kinematic and dynamic properties of an imitator usually differ from that of an instructor. Here, we show the generalization performance of our framework. We apply our proposed method to robot models that have the same kinematics as the model shown in Table 3.1 but have different weights.

We apply the proposed method to the different robot models with five different weights: 40, 50, 60, 70, and 80 kg, while the weight of the robot model shown in Table 3.1 is 60 kg. As mentioned in Section 3.1, we scaled GRF by the ratio of the robot total mass to the instructor's weight. Note that we can easily measure the total mass by a force plate.

We verified that each of the five different robot models could successfully generate the squatting motions. In Table 3.4, we show tracking error with the different weights when system observe the squat motion **90a** $\rightarrow$ **90b**. For the robot that has the same weight to instructor's, the tracking error was closest to the average with the observed squat motion **90a** $\rightarrow$ **90b** as shown in Table 3.3. These results show the generalization property of our imitation learning framework.

Different velocity

Moreover, we apply the proposed method to squat motion which have different velocity from human. The proposed method observe trajectory **with knee amplitudes of 45 degrees** with twice the velocity of training data trajectory **with knee amplitudes of 45 degrees**. We show squat with different velocity squat movement **with knee amplitudes of 45 degrees** are generated by using different velocity training data.(see Fig. 3.11).

3.3 Conclusion

We proposed an imitation learning framework that can extracts primitives from captured movements and measured ground reaction force to generate physically consistent imitated behaviors. We applied the proposed method to a four and seven-link simulated robot model that has different kinematic and dynamic properties from an instructor. The extracted primitives are

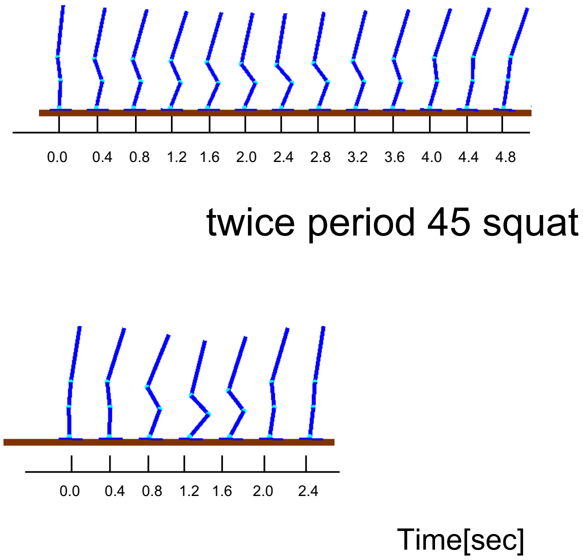

Figure 3.11: Generated squat movements for the observed trajectory **with knee amplitudes of 45 degrees** with twice the velocity of training data trajectory **with knee amplitudes of 45 degrees**. The results show generalization ability of velocity.

Weight of robot model	MSE (deg^2)
40kg	0.76
50kg	0.85
60kg	0.86
70kg	0.91
80kg	0.96

Table 3.4: Tracking error with the five different robot models. We selected the data set **90a→90b** as the observed squat motion. The instructor's weight is 60kg.

used to recognize newly observed squat movements. The four and seven - link robot model generated sixteen different combined squatting behaviors by using estimated GRF. We also showed the generalization performance of our proposed method by using five different robot models.

So far, we have only applied the proposed method to simple human behaviors. One of our future studies will apply our imitation learning framework to recognize larger numbers of behaviors and generate more complex motions by using robots with higher DOF.

In theory, we can apply the proposed method to recognize and generate a larger number of input motions by increasing the number of the linear dynamics in SSSMs. However, in practice, the computational requirements increase according to the increase in linear dynamics, and the state estimation problems can be computationally intractable. To cope with this issue, we will consider using a feature extraction method to find low-dimensional state spaces for human behaviors.

Chapter 4

Via Point Extraction to Transfer Skill

We proposed solution to overcome difference dynamics and kinematics between instructor's and imitator's. These methods are applicable to movement of whole body (ex. squat, dance, locomotion). However, these methods are not applicable complex task because these methods only don't take into account performance of task. We need to consider additional constraint to accomplish task. It is important to treat not only behavior but also skill transfer in imitation learning framework. In this chapter, we propose imitation learning framework which is applicable to skill transfer by using feature of human training data. Feature is extracted from correlation between motion capture data and criteria of task accomplishment. Human can imitate complex task after a process of trial and error. We suppose that human can pick up tips during training term. (Kuniyoshi, Ohmura, Terada, & Nagakubo, 2004) was to show sparse critical points is derived from human motion data to accomplish Roll-and-Rise.

Critical points are extracted as via point many human motion capture data including ”know-how” of task in our proposed framework. We apply our proposed framework to Kendama task. In recent years, Kendama task have already done various research from the view point mechanical system (Sato, Sakaguchi, Masutani, & Miyazaki, 1993) or reinforcement learning (Kober & Peters, 2008) and probabilistic model (Chiappa, Kober, & Peters, 2008). However, there is no related work which imitate ”know-how” on human learning process. One of the famous research extract via point from motion capture data, and modify via point by using performance of robot (Miyamoto et al., 1996). In our proposed method, via point are extracted from many human training data and performance through low dimensional space. We show robot model can accomplish Kendama task on simulator by using our imitation learning framework.

4.1 Imitation learning framework and representation of via point

In our learning system (see Fig. 4.1), we first construct a motion database that collects ball position and hand position from various humans behaviors of kendama by using a motion capture system (see Fig. 4.2). We record positions of the hand, ball and cup on 45 trial. This database have success and failure of task. We extract via point based on randomness time, and trajectories are represented by 5-th spline function from via point. Control input is represented as between via point and task performance:

1. We extract features of task from database by using Canonical Correlation Analysis (CCA) (see Appendix B).
   Performance measurement to accomplish task is defined as "relative velocity" when ball position contact with hand position. Feature is extracted as canonical correlation between via point and performance measurement.
2. The extracted features are used to optimize robot motion. We optimize robot model motion by moving these features on simulation.
3. We can know critical point of that task by observing extracted features.

In the following sections, we show our simulation and result.

4.2 Simulation

4.2.1 Simulation set up

We show our simulation setup. To put it simply, we consider only vertical two dimensional (see Fig. 4.3). We define hand position as y_{hand} and ball position as y_{ball} on vertical. Database have $Y_{hand} = \{y^1_{hand}, ..y^m_{hand}\}$ and

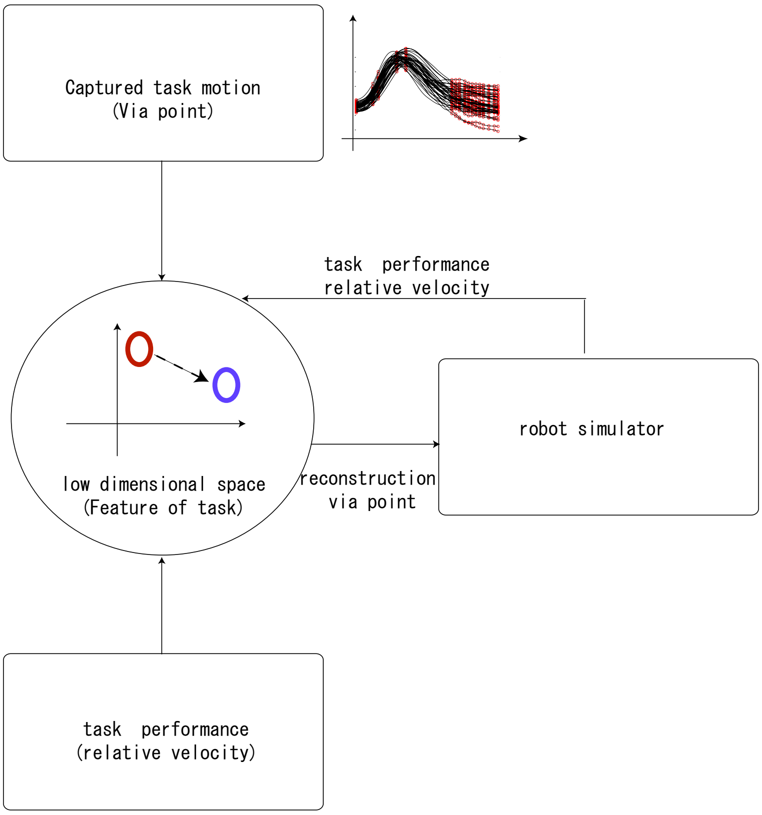

Figure 4.1: Proposed imitation learning framework: Features of task in low dimensional space are extracted from via point and human task performance. Via points are reconstructed by these features, which is optimized by robot performance.

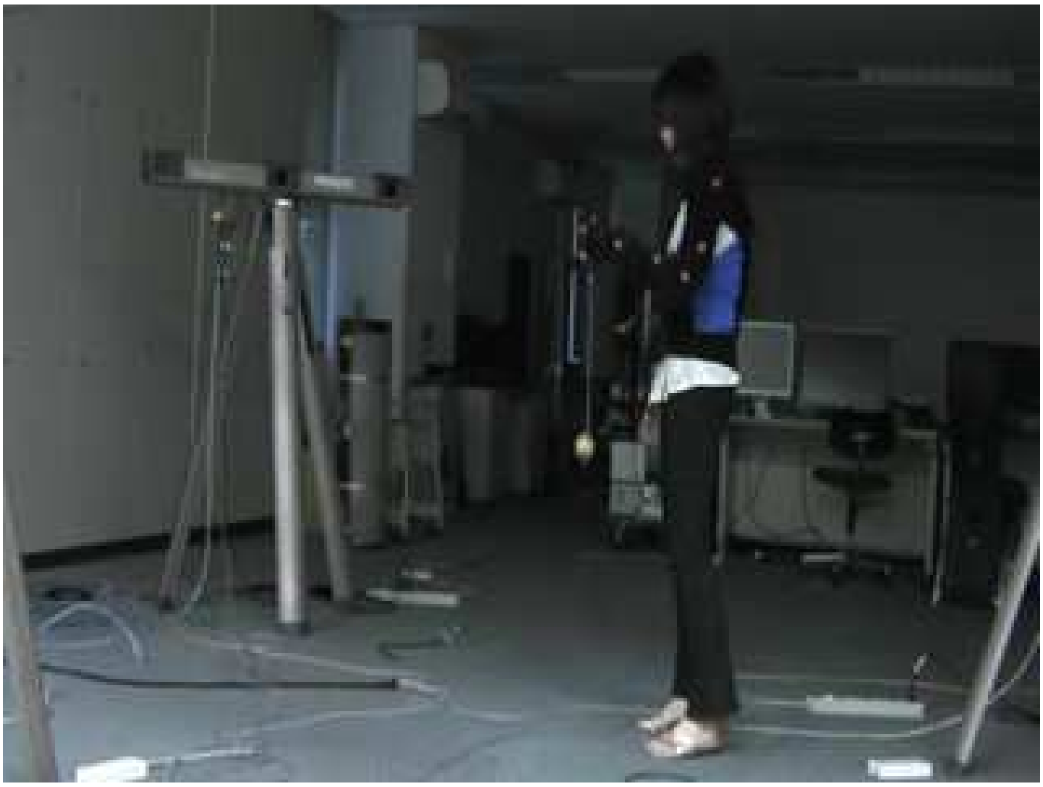

Figure 4.2: Experiment setup. The upper arm of human and kendama cup and ball motions are measured by using optical motion capture.

$Y_{ball} = \{y^1_{ball}, ..y^m_{ball}\}$. m is the number of sample. These training data are aligned in time series. We select k the via point $Y_{handviapoint}$ and relative velocity $Y_{handvelocity}$ on each sample. Thus, the dimension of data set $Y_{handviapoint}$ is $k \times m$, the dimension of data set $Y_{handvelocity}$ is $1 \times m$. We extract feature in reduced coordinates U between $Y_{handviapoint}$ and $Y_{handvelocity}$ by using CCA. The matrix of bases can be represented as a linear mapping D, which transforms the reduced coordinates U to optimally approximate the example deformations $Y_{handviapoint}$. Thus, it is the solution of the following linear least-squares problem:

$$min_D \sum_m \|DU - Y_{handviapoint}\|^2 \tag{4.1}$$

Using this parameter D, our framework modify U so that $Y^{robot}_{handvelocity}$ converge to 0. Therefore cost function is that:

$$E = \frac{1}{2}\|0 - Y^{robot}_{handvelocity}\|^2 \tag{4.2}$$

Minimizing this cost function, we modify U by following term.

4.2.2 Simulation result

After robot simulation, via points of hand position are successfully optimized, $E = 0.0382$. Initial via point is $E = 0.5497$. We show Figure 4.4 and figure 4.5 shows the trajectories of hand position based on initial and optimized via points. This result indicates that gradient of optimized hand position trajectory is almost equivalent to gradient of ball position trajectory when ball contact to hand. We show that our proposed imitation learning framework can optimize via point through reduced coordinates.

To verify validity of their low dimensional space U, we optimize via points by using 1) random a(B.1) and 2) select via points to change by hand. Both case 1) and case 2) can not generate via points which have error lower than $E = 0.0382$. Therefore, we suppose that it is importance to use canonical correlation space between via point and relative velocity as performance measure.

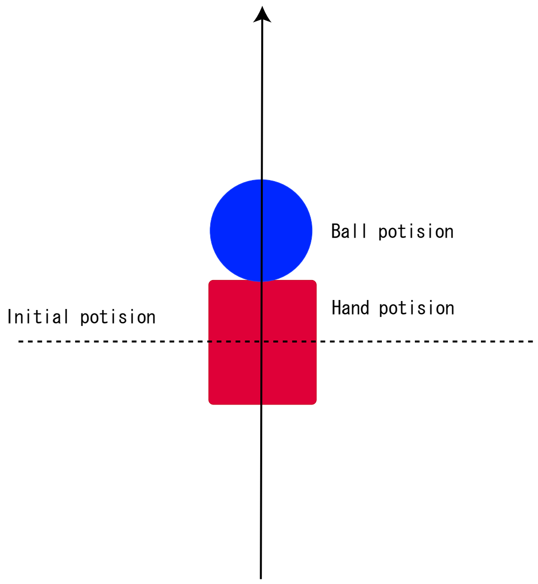

Figure 4.3: Kendama model for simulation. Blue circle show ball position and red box show hand position. We calculate relative velocity when this time after ball is jumped over hand position.

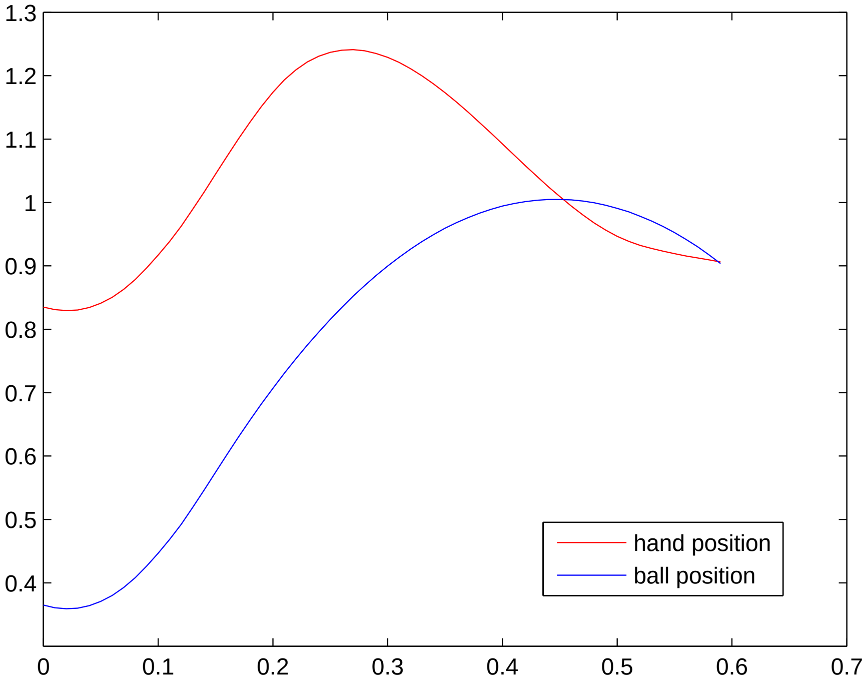

Figure 4.4: Trajectory of initial hand and ball position.

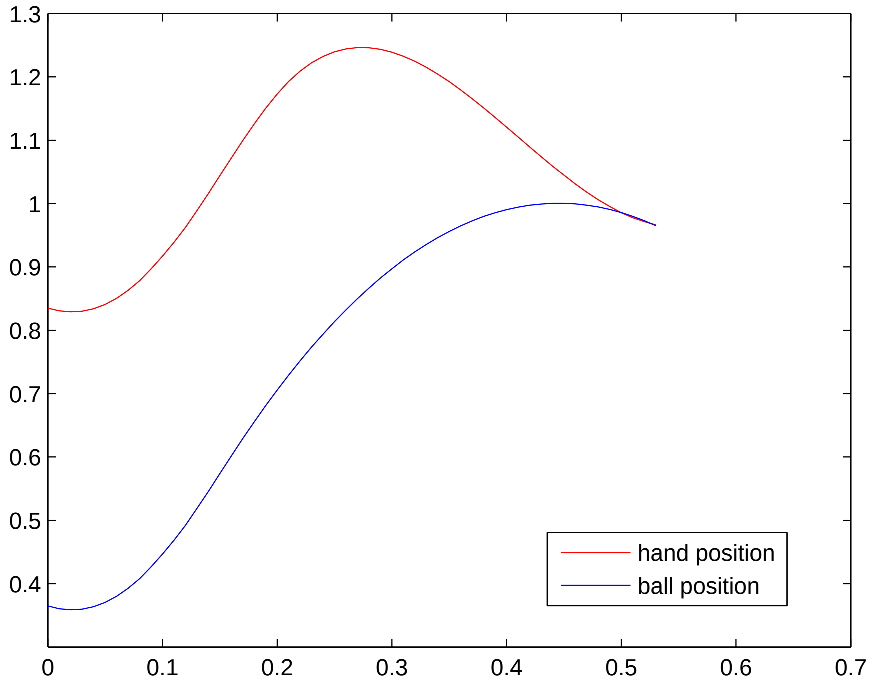

Figure 4.5: Trajectory of hand and ball position after optimization.

4.3 Conclusion

In this chapter, we show new imitation learning framework to apply complex task using human trial and error data. Moreover, we focused on relative velocity when ball contact to hand, which is introduced in cost function as performance measure. Finally, we can optimize via points through low dimensional space which is extracted from canonical correlation between via points and relative velocity.

Chapter 5

Conclusion

5.1 Summary

This thesis describes a statistic imitation learning framework for humanoid robots and CG character. These frameworks accommodate instructor's motion to robot and CG character which has different kinematics and dynamics. These frameworks are based on feature extraction from instructor's motion. The main contributions of this thesis are generating motion of CG character or robot consistent with physical condition.

In chapter 1, the significance of imitation learning for intelligence behavior was described. We focused on three intractable problem under prior imitation learning framework and proposed these solution. Related works were introduced in a various fields such as neuroscience, behavior science, robotics, computer graphics and so on.

Chapter 2 described a method generating motion of non-human characters from human motion capture data. Characters considered in this work have proportion and/or topology significantly different from humans but are expected to convey expression and emotions through body language that are understandable to human viewers. Our proposed method provides motion of non-human characters that leverages motion data from human subject performing in the style of the target character. The method consists of a statistical mapping function learned from a small set of corresponding key poses, and physics-based optimization process to improve the physical realism. We demonstrate our approach on three characters and a variety throughout shared low dimensional coordinates.

Chapter 3 explained our proposed imitation learning frameworks which select movement primitives from training data and can generate physically consistent imitated behaviors. Switching state-space models is applied to abstraction of instructor's motion based on movement primitives are automatically extracted from training data. Our proposed framework estimate GRF by switching these movement primitives.
In chapter 4, we proposed imitation learning framework to apply complex task. Features of task as correlation low dimensional space is extracted between motion capture data and task performance. Robot can easily learn task by using these features of task.

5.2 Future work

Human motion is useful to generate humanoid or non-human robot motion, however cases of the failure are occurred by the differences of dynamics and kinematics between human and robot. Our proposed method will be overcome this difference by extracting shared features between human and robot. As another advantage for using human motion, it is possible that robot can recognize and estimate human action. By extending our proposed framework, robot will be move in alien environment by observing human motion in daily life. Before our proposed frameworks can be put to practical use, we construct a wealth of database of human motion. In chapter 4, we show database may be designed based on what to imitate. The development of new framework focusing on low dimensional database based on proposed framework will be considered.

Appendix A

Mixture of Experts

We extract primitives from the motion database by using the Gaussian mixture of linear dynamical models. As in Fig. 3.1, the state $\mathbf{X}$ in motion database is defined as $\mathbf{X} = \left(\left(\boldsymbol{\theta}^{DB}\right)^T, \left(\dot{\boldsymbol{\theta}}^{DB}\right)^T, \left(\mathbf{f}_{grf}^{DB}\right)^T \right)^T$.

Here we consider M mixture of liner Gaussian dynamical models:

$$P(\mathbf{X}_{t+1}|\mathbf{X}_t, \mathbf{H}) = \sum_{m=1}^{M} \alpha_m P(\mathbf{X}_{t+1}|\mathbf{X}_t, \boldsymbol{\eta}_m), \tag{A.1}$$

where α_m is mixture coefficient for m-th model that is defined as:

$$\alpha_m = \left(\frac{P(m)\mathcal{N}(\mathbf{X}_t|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{j=1}^{M} P(j)\mathcal{N}(\mathbf{X}_t|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \right). \tag{A.2}$$

$P(m)$ is the prior of the model. Here we use $P(m) = \frac{1}{M}$. $\mathbf{H}$ consists of $\{\boldsymbol{\mu}_m\}_1^M$, $\{\boldsymbol{\Sigma}_m\}_1^M$, $\{\boldsymbol{\eta}_m\}_1^M$, where $\{\boldsymbol{\eta}_m\}_1^M$ consists of $\{\mathbf{A}^{(m)}\}_1^M$, $\{\mathbf{Q}^{(m)}\}_1^M$. The linear dynamical model is given by:

$$P(\mathbf{X}_{t+1}|\mathbf{X}_t, \boldsymbol{\eta}_m) = \mathcal{N}(\mathbf{X}_{t+1}|\mathbf{A}^{(m)}\mathbf{X}_t, \mathbf{Q}^{(m)}). \tag{A.3}$$

The EM algorithm can be used to optimize the parameters (Xu, Jordan, & Hinton, 1994), the posterior of the latent variable is calculated as

$$P(m|\mathbf{X}_{t+1}, \mathbf{X}_t, \mathbf{H}) = \frac{P(\mathbf{X}_t|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)P(\mathbf{X}_{t+1}|\mathbf{X}_t, \boldsymbol{\eta}_m)}{\sum_{j=1}^{M} P(\mathbf{X}_t|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)P(\mathbf{X}_{t+1}|\mathbf{X}_t, \boldsymbol{\eta}_j)}. \tag{A.4}$$

In M-step, the parameters are optimized as follows:

$$\boldsymbol{\mu}_m^{new} = \frac{\sum_t P(m|\mathbf{X}_{t+1}, \mathbf{X}_t, \mathbf{H}_{old})\mathbf{X}_t}{\sum_t P(m|\mathbf{X}_{t+1}, \mathbf{X}_t, \mathbf{H}_{old})} \tag{A.5}$$

$$\boldsymbol{\Sigma}_m^{new} = \frac{\sum_t P(m|\mathbf{X}_{t+1}, \mathbf{X}_t, \mathbf{H}_{old})}{\sum_t P(m|\mathbf{X}_{t+1}, \mathbf{X}_t, \mathbf{H}_{old})}(\mathbf{X}_t - \boldsymbol{\mu}_m^{new})(\mathbf{X}_t - \boldsymbol{\mu}_m^{new})^T \tag{A.6}$$

$$\begin{aligned}\mathbf{A}_{new}^{(m)} = & \sum_t P(m|\mathbf{X}_{t+1}, \mathbf{X}_t, \mathbf{H}_{old})\mathbf{X}_{t+1}\tilde{\mathbf{X}}_t \\ & \times \left[\sum_t P(m|\mathbf{X}_{t+1}, \mathbf{X}_t, \mathbf{H}_{old})\tilde{\mathbf{X}}_t\tilde{\mathbf{X}}_t^T\right]^{-1}\end{aligned} \tag{A.7}$$

$$\begin{aligned}\mathbf{Q}_{new}^{(m)} = & \frac{\sum_t P(m|\mathbf{X}_{t+1}, \mathbf{X}_t, \mathbf{H}_{old})}{\sum_t P(m|\mathbf{X}_{t+1}, \mathbf{X}_t, \mathbf{H}_{old})} \\ & \times (\mathbf{X}_{t+1} - \mathbf{A}_{new}^{(m)}\tilde{\mathbf{X}}_t)(\mathbf{X}_{t+1} - \mathbf{A}_{new}^{(m)}\tilde{\mathbf{X}}_t)^T,\end{aligned} \tag{A.8}$$

where $\tilde{\mathbf{X}}_t = [\mathbf{X}_t^T, 1]^T$.

Appendix B

Kernel Canonical Correlation Analysis

CCA(Canonical Correlation Analysis) find to identify and quantify the associations between corresponding sets of heterogeneous observations. Further pairs of maximally correlated linear combinations are chosen such, that they are orthogonal to those already identified. The pairs of linear combinations are called 'canonical variables' and their correlations 'the canonical correlations'. The canonical correlations measure the strength of association between two sets of variables. CCA is closely related to other linear subspace method like Principal Component Analysis (PCA), Multivariate Liner Regression (MLR). We define two variables are $X \in R^{nx}$ and $Y \in R^{ny}$. Linear combinations of the variables in these equation.

$$U_a =< a, X >= \sum_i^{nx} a_i X_i \tag{B.1}$$

$$V_b =< b, Y >= \sum_i^{ny} b_i Y_i \tag{B.2}$$

$< a, X >$ is inner product. Correlation is defined from equation(B.1),(B.2).

$$p = \frac{cov(U_a, V_b)}{\sqrt{var(U_a)var(V_b)}} \tag{B.3}$$

CCA is solution of the maximization equation B.3 under $var(U_a) = 1$ and $var(V_b) = 1$. However, in the case there are strong non-linear correlation

between X and Y, we should extend CCA by using Kernel method.
Given an nonlinear map ϕ into a high dimensional feature space, we can define a kernel method. learning samples are $(X_i, Y_i)_{i=1}^{N}$.

$$U_a =< a, \phi_x(X) > \tag{B.4}$$

$$V_b =< b, \phi_y(Y) > \tag{B.5}$$

From Lagrangean and derivation, a is by maximizing correlation,

$$a = \sum_{i}^{N} \alpha_i \phi_x(X_i) \tag{B.6}$$

As a result, we know

$$U = \sum_{i}^{N} \alpha_i < \phi_x(X_i), \phi_x(X) > \tag{B.7}$$

U can be defined as inner product in a high dimensional feature space. $< \phi_x(X_i), \phi_x(X) >$ is kernel function$k_X(X_1, X_2)$. In similar way V can be defined by β. Lagrangean of maximizing correlation between U and V can be defined from kernel function. From Lagrangean, α and β can be calculated by solving eigen value problem.

Appendix C

List of Publications

C.1 Journal Papers

1. 動作認識における床反力情報の推定と見まね学習への適用
   有木 由香, 森本 淳, 玄 相昊
   電子情報通信学会論文誌 Vol.J91-D,No.9,pp.2394-2403,Sep. 2008.

C.2 International Conference Proceedings

1. Behavior recognition with ground reaction force estimation and its application to imitation learning
   Yuka Ariki, Jun Morimoto, Sang-Ho Hyon
   International Conference on Intelligent RObots and Systems (IROS), pp.2029-2034, France, September 2008.

2. Animating Non-Humanoid Characters with Human Motion Data
   Katsu Yamane, Yuka Ariki, Jessica Hodgins
   Proc.The ACM SIGGRAPH/ Eurographics Symposium on Computer Animation 2010

C.3 Awards

1. 有木　由香　IEEE 関西支部, 学生研究奨励賞受賞, 2009 年 2 月

C.4 Domestic Conference Proceedings

1. Extraction of movement primitives without explicit labeling for imitation learning
   Yuka Ariki*, Jun Morimoto, Sang-Ho Hyon
   2010 Neuro

2. Behavior recognition with ground reaction force estimation and its application to imitation learning
   Yuka Ariki*, Jun Morimoto, Sang-Ho Hyon
   第 31 回日本神経科学大会

3. 動作認識における床反力情報の推定と見まね学習への適用
   有木 由香, 森本 淳, 玄 相昊
   第 25 回日本ロボット学会学術講演会 (RSJ 2007)，3G15(CD-ROM).

4. 床反力情報とモーションキャプチャデータを用いた人間の動作認識
   有木 由香, 森本 淳
   電子情報通信学会技術研究報告，NC2006-44, pp. 37-41.

5. ASCONE2007 田中宏和先生講義録 [ロコモーションの計算理論-生物とロボットの接点-]
   杉本徳和　有木由香
   日本神経回路学会誌　Vol.15,NO.4(2008)

Bibliography

Akaho, S. (2006). A kernel method for canonical correlation analysis. *CoRR*, *abs/cs/0609071*.

Arikan, O. (2006). Compression of motion capture databases. *ACM Trans. Graph.*, *25*(3), 890–897.

Atkeson, C. G., & Schaal, S. (1997). Robot learning from demonstration. In *ICML*, pp. 12–20.

Barbic, J., Safonova, A., Pan, J. Y., Faloutsos, C., Hodgins, J. K., & Pollard, N. S. (2004). Segmenting motion capture data into distinct behaviors. In *Proceedings of the Graphics Interface Conference*, pp. 185–194.

Bentivegna, D. C., Atkeson, C. G., Ude, A., & Cheng, G. (2004). Learning to act from observation and practice. *I. J. Humanoid Robotics*, *1*(4), 585–611.

Bitzer, S., Howard, M., & Vijayakumar, S. (2010). Using dimensionality reduction to exploit constraints in reinforcement learning. In *IROS*.

Bregler, C., Loeb, L., Chuang, E., & Deshpande, H. (2002). Turning to the masters: motion capturing cartoons. In *SIGGRAPH*, pp. 399–407.

Calinon, S., Guenter, F., & Billard, A. (2007). On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, *37*(2), 286–298.

Chiappa, S., Kober, J., & Peters, J. (2008). Using bayesian dynamical systems for motion template libraries. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 297–304.

Choi, K.-J., & Ko, H.-S. (2000). Online motion retargetting. *Journal of Visualization and Computer Animation*, *11*(5), 223–235.

DiPellegrino, G., Fadiga, L., Fogassi, L., Gallese, V., & Rizzolatti, G. (1992). Understanding motor events: A neurophysiological study.. *Experimental Brain Research*, *91*, 176–180.

Doya, K., Samejima, K., ichi Katagiri, K., & Kawato, M. (2002). Multiple model-based reinforcement learning. *Neural Computation*, *14*(6), 1347–1369.

Ek, C. H., Rihan, J., Torr, P. H. S., Rogez, G., & Lawrence, N. D. (2008). Ambiguity modeling in latent spaces. In *MLMI*, pp. 62–73.

Ek, C. H., Torr, P. H. S., & Lawrence, N. D. (2007). Gaussian process latent variable models for human pose estimation. In *MLMI*, pp. 132–143.

Feng, W.-W., Kim, B.-U., & Yu, Y. (2008). Real-time data driven deformation using kernel canonical correlation analysis. *ACM Trans. Graph.*, *27*(3).

Gallese, V., & Goldman, A. (1998). Mirror neurons and the simulation theory of mind-reading.. *Trends Cogn. Sci*, *2*, 493–501.

Ghahramani, Z., & Hinton, G. E. (2000). Variational learning for switching state-space models. *Neural Computation*, *12*(4), 831–864.

Gleicher, M. (1998). Retargeting motion to new characters. In *SIGGRAPH*, pp. 33–42.

Gomi, H., & Kawato, M. (1993). Recognition of manipulated objects by motor learning with modular architecture networks. *Neural Networks*, *6*(4), 485–497.

Grimes, D. B., Chalodhorn, R., & Rao, R. P. N. (2006). Dynamic imitation in a humanoid robot through nonparametric probabilistic inference. In *Proceedings of Robotics: Science and Systems*.

Grochow, K., Martin, S. L., Hertzmann, A., & Popovic, Z. (2004). Style-based inverse kinematics. *ACM Trans. Graph.*, *23*(3), 522–531.

Guenter, F., Hersch, M., Calinon, S., & Billard, A. (2007). Reinforcement Learning for Imitating Constrained Reaching Movements. *RSJ Advanced Robotics, Special Issue on Imitative Robots*, *21*(13), 1521–1544.

Haruno, M., Wolpert, D. M., & Kawato, M. (2001). Mosaic model for sensorimotor learning and control. *Neural Computation*, *13*(10), 2201–2220.

Hukashiro, S., Sakurai, S., Hirano, Y., & Ae, M. (2000). *Sports Biomechanics (in Japanese).*

Hyon, S.-H., Hale, J. G., & Cheng, G. (2007). Full-body compliant human-humanoid interaction: Balancing in the presence of unknown external forces. *IEEE Transactions on Robotics*, *23*(5), 884–898.

Ijspeert, A., Nakanishi, J., & Schaal, S. (2003). Learning attractor landscapes for learning motor primitives. In *Advances in neural information processing systems 15*, pp. 1547–1554.

Ikemoto, L., Arikan, O., & Forsyth, D. A. (2009). Generalizing motion edits with gaussian processes. *ACM Trans. Graph.*, *28*(1).

Inamura, T., Tanie, H., & Nakamura, Y. (2003). Keyframe compression and decompression for time series data based on the continuous hidden markov model. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1487–1492.

Kober, J., & Peters, J. (2008). Policy search for motor primitives in robotics. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 849–856.

Kuniyoshi, Y., Ohmura, Y., Terada, K., & Nagakubo, A. (2004). Dynamic roll-and-rise motion by an adult-size humanoid robot. *I. J. Humanoid Robotics*, *1*(3), 497–516.

Lawrence, N. D. (2003). Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in Neural Information Processing Systems (NIPS).*

Lee, J., & Shin, S. Y. (1999). A hierarchical approach to interactive motion editing for human-like figures. In *SIGGRAPH*, pp. 39–48.

Li, Y., Wang, T., & Shum, H.-Y. (2002). Motion texture: a two-level statistical model for character motion synthesis. *ACM Trans. Graph.*, *21*(3), 465–472.

Liu, C. K., & Popovic, Z. (2002). Synthesis of complex dynamic character motion from simple animations. In *SIGGRAPH*, pp. 408–416.

Macchietto, A., Zordan, V. B., & Shelton, C. R. (2009). Momentum control for balance. *ACM Trans. Graph.*, *28*(3).

Miyamoto, H., & Kawato, M. (1998). A tennis serve and upswing learning robot based on bi-directional theory. *Neural Networks*, *11*(7-8), 1331–1344.

Miyamoto, H., Schaal, S., Gandolfo, F., Gomi, H., Koike, Y., Osu, R., Nakano, E., Wada, Y., & Kawato, M. (1996). A kendama learning robot based on bi-directional theory. *Neural Networks*, *9*(8), 1281–1302.

Morimoto, J., Hyon, S.-H., Atkeson, C. G., & Cheng, G. (2008). Low-dimensional feature extraction for humanoid locomotion using kernel dimension reduction. In *ICRA*, pp. 2711–2716.

Muico, U., Lee, Y., Popovic, J., & Popovic, Z. (2009). Contact-aware non-linear control of dynamic characters. *ACM Trans. Graph.*, *28*(3).

Nakaoka, S., Nakazawa, A., Yokoi, K., Hirukawa, H., & Ikeuchi, K. (2003). Generating whole body motions for a biped humanoid robot from captured human dances. In *IEEE International Conference on Robotics and Automation*, pp. 3905–3910.

Pastor, P., Hoffmann, H., Asfour, T., & Schaal, S. (2009). learning and generalization of motor skills by learning from demonstration. In *international conference on robotics and automation (icra2009)*.

Pavlovic, V., Rehg, J. M., & MacCormick, J. (2000). Learning switching linear models of human motion. In *Advances in Neural Information Processing Systems 13*, pp. 981–987.

Popovic, Z., & Witkin, A. P. (1999). Physically based motion transformation. In *SIGGRAPH*, pp. 11–20.

Safonova, A., Hodgins, J. K., & Pollard, N. S. (2004). Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. Graph.*, *23*(3), 514–521.

Samejima, K., Katagiri, K., Doya, K., & Kawato, M. (2002). Symbolization and imitation learning of motion sequence using competitive modules. *The transactions of the Institute of Electronics, Information and Communication Engineers. D-II*, *85*(1), 90–100.

Sato, S., Sakaguchi, T., Masutani, Y., & Miyazaki, F. (1993). Mastering of a task with interaction between a robot and its environment : "kendama" task. *Transactions of the Japan Society of Mechanical Engineers. C*, *59*(558), 487–493.

Schaal, S. (1999). Is imitation learning the route to humanoid robots?. *Trends in cognitive sciences*, pp. 233–242.

Schaal, S., Ijspeert, A., & Billard, A. (2003). Computational approaches to motor learning by imitation. *Philosophical transaction of the royal society of london: series b, biological sciences*, pp. 537–547.

Shin, H. J., Lee, J., Shin, S. Y., & Gleicher, M. (2001). Computer puppetry: An importance-based approach. *ACM Trans. Graph.*, *20*(2), 67–94.

Shon, A. P., Grochow, K., Hertzmann, A., & Rao, R. P. N. (2005). Learning shared latent structure for image synthesis and robotic imitation. In *Advances in Neural Information Processing Systems (NIPS)*.

Takano, W., Yamane, K., Sugihara, T., Yamamoto, K., & Nakamura, Y. (2006). Primitive communication based on motion recognition and generation with hierarchical mimesis model. In *IEEE International Conference on Robotics and Automation*, pp. 3602–3609.

Ude, A., Riley, M., Nemec, B., Kos, A., Asfour, T., & Cheng, G. (2007). Synthesizing goal-directed actions from a library of example movements. In *IEEE-RAS International Conference on Humanoid Robots*.

Urtasun, R., Fleet, D. J., Geiger, A., Popovic, J., Darrell, T., & Lawrence, N. D. (2008). Topologically-constrained latent variable models. In *ICML*, pp. 1080–1087.

Witkin, A. P., & Kass, M. (1988). Spacetime constraints. In *SIGGRAPH*, pp. 159–168.

Wolpert, D. M., Doya, K., & Kawato, M. (2003). A unifying computational framework for motor control and social interaction.. *Phil. Trans. R. Soc. Lond.*, *B358*, 593–602.

Wolpert, D. M., & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, *11*(7-8), 1317–1329.

Xu, L., Jordan, M. I., & Hinton, G. E. (1994). An alternative model for mixtures of experts. In *Advances in Neural Information Processing Systems 7*, pp. 633–640.

Yamane, K., & Nakamura, Y. (2003). Dynamics filter - concept and implementation of online motion generator for human figures. *IEEE Transactions on Robotics*, *19*(3), 421–432.