LEARNING FROM OBSERVATION PARADIGM: LEG TASK MODELS FOR REPRODUCING HUMAN DANCE MOTIONS ON BIPED HUMANOID ROBOTS

観察学習パラダイム：脚タスクモデルを用いた二足歩行ヒューマノイドロボットによる人の舞踊動作の再現

by

Shinichiro NAKAOKA
中岡 慎一郎

A Doctral Thesis
博士論文

Submitted to
the Department of Computer Science
the Graduate School of Information Science and Technology
the University of Tokyo
on December 16, 2005
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Information Science and Technology
in Computer Science

Thesis Supervisor: Katsushi IKEUCHI　池内 克史

ABSTRACT

This thesis proposes a framework that enables a biped humanoid robot to learn and reproduce whole body motions from human motions. This ability can increase the usefulness of biped humanoid robots, which have a potential to perform human-like motions of the whole body. We achieve this goal on the basis of the *Learning from Observation (LFO)* paradigm, and prove that the paradigm is valid for whole body motions. In our framework, the target motions are dances, and a robot uses its own legs to support its body during a dance performance. Reproducing such motions from human motions is a novel attempt. This thesis especially focuses on leg motions, which present a serious challenge to the achievement of our goal.

LFO is a paradigm of teaching a task to a robot. It consists of three processes: observation, recognition, and execution. First, a human instructor demonstrates a task and a robot observes the demonstration. Then the robot recognizes what is done in the observed motion. After that, the robot can execute the task according to the recognition results. In this paradigm, all an instructor has to do is to demonstrate a task with his or her body. This makes teaching simple and efficient because complex operations or programming by a robotic specialist is not required. This simultaneously means that a robot imitates an instructor's way of doing a task. These features are great advantages in teaching human-like motions to a robot that has a complex human-like body.

In the LFO paradigm, the recognition and execution processes are based on *task models* designed for a target task domain. Task models provide essential representations of tasks, which are necessary for executing tasks in various situations. One of the fundamental factors in different situations is the differences among bodies. In this study, to solve the problem caused by body difference is the most important role of task models.

In general, human or humanoid bodies have their own physical characteristics different from each other. The differences include geometry of body parts, weight distribution, and physical capabilities. In addition, humanoid robots can have their own body mechanisms quite different from those of another body. Due to these differences, observed motions of an instructor cannot be performed directly by another body of a robot. It is hence inevitable that the motions are partly modified so that they are adapted to the body that executes the motions. This adaptation must be processed so that the original purpose of a task is achieved. For motions of the whole body including legs, body differences significantly affect the execution of motions because the motions are performed under severe constraints caused by the interaction with the floor. For motions such as dances, important characteristics of the original motions must be preserved simultaneously. These factors make reproducing whole body motions performed by another body a difficult problem. Since legs play a major role in the

interaction with the floor, leg motion is especially important for solving the problem.

This thesis proposes *leg task models*, which are a set of task models that deal with leg motions of whole body performances. Leg task models are core elements for solving the problem of body difference according to the LFO paradigm. Basic unit of leg motions are modeled as *tasks*, and characteristics of each task are described by its *skill parameters*. Leg motions in a performance are represented as a sequence of a number of tasks that have their own values of skill parameters. This representation is quite different from low-level motion data such as joint angle trajectories.

The purpose of leg task models is to provide the essential information that is necessary for executing the same motion of a whole body in various situations including various bodies. In order to provide the essence of leg motions in dance performances, leg task models focus on the interaction between soles and the floor and important motion characteristics to preserve in dances. The models are designed so that they do not depend on a particular body or a particular sensing system for recognizing tasks. We define four leg task models that satisfy these factors.

After the task models are defined, problems to be dealt with are how to recognize a task sequence from demonstrations by an instructor and how a robot executes tasks according to the task sequence. The essential representation of task models has a potential to flexibly support various recognition methods and various robots. In this thesis, we employ motion captures as a sensing system for the recognition process, and propose a method that recognizes a sequence of leg tasks from captured human motion trajectories. We also propose a method for the execution process, which assumes current biped humanoid robots that have standard body mechanisms. Especially for leg motions, the current mechanisms cause severe constraints when the robots perform human motions. However, our method based on the leg task models can generate motions that the robots can perform stably with their own body mechanisms. The generated motions also express important characteristics of dances as much as possible with their bodies.

We tested the proposed framework on an actual dance and a robot called *HRP-2*. The dance is a Japanese folk dance, which includes various motions performed by a whole body. HRP-2 is a biped humanoid robot that has human-like size and weight. Motion data of two dancers were obtained by using an optical motion capture. Our framework recognized correct sequences of leg tasks, and generated joint angle trajectories that were executable on HRP-2 from the recognized task sequences. Finally, HRP-2 successfully performed the dance motions stably according to the generated motion data. This has been a novel achievement of biped humanoid robots.

This result proved the validity of our framework based on the leg task models. This simultaneously proved that the LFO paradigm is valid for whole body motions. Our achievement provides the basis of the LFO paradigm for whole body motions, which has expanded the potential use of biped humanoid robots.

論文要旨

本論文では，二足歩行ヒューマノイドロボットが全身の動作を人の動作から習得・再現することを可能にするフレームワークを提案する．この能力によって，人らしい全身の動作を行う能力を秘めた二足歩行ヒューマノイドロボットの有用性を高めることができる．我々はこの目標を”Learning from Observation (LFO)” のパラダイムに基づいて実現し，パラダイムが全身の動作に対して有効であることを示す．我々のフレームワークでは，対象とする動作は舞踊とし，ロボットは舞踊の演技を行う際に自身の脚を用いて身体の支持を行う．このような動作を人の動作から再現することは，今まで成されなかった試みである．本論文では，この目標を実現するにあたって大きな困難を引き起こす脚の動作を重点的に扱う．

LFO はあるタスクをロボットに教示するためのパラダイムである．これは観察，認識，実行の 3 つのプロセスからなる．まず人間の教示者がタスクの実演を行い，ロボットはこれを観察する．次にロボットは観察した動作において何が行われていたかを認識する．その後，ロボットは認識した結果に基づいてそのタスクを実行することが可能となる．このパラダイムでは，教示者は自身の身体でタスクを実演するだけでよい．これによって，ロボットの専門家による複雑な操作やプログラミングが必要でなくなるため，教示はシンプルで効率的なものとなる．これは同時に，教示者のタスク実行の仕方をロボットが摸倣することを意味する．これらの特徴は，人らしい動作を人に似た複雑な身体を持つロボットに教示する際に大きな利点となる．

LFO のパラダイムでは，認識と実行のプロセスは，対象とするタスクの種類に対して設計された「タスクモデル」に基づいて行われる．タスクモデルはタスクの本質的な表現を与えるものであり，これは様々な状況でタスクを実行する際に必要不可欠なものである．異なる状況の根本的な要因のひとつは身体の違いである．本研究では，身体差異に起因する問題を解決することがタスクモデルの最も重要な役割である．

一般的に，人間やヒューマノイドの身体は，それぞれ異なる身体的特徴を持つ．この違いとして，身体部位の幾何配置や重量配分，身体能力などが挙げられる．また，ヒューマノイドロボットは他の身体とは相当異なる身体のメカニズムを持ち得る．これらの差異により，観察した教示者の動作を他の身体であるロボットで直接実行することは不可能となる．従って，教示者の動作はその動作を実行する身体に適応するよう部分的に変更することが避けれらない．この適応はタスクの元の目的が達成されるように処理されなければならない．脚を含む全身の動作においては，動作は床とのインタラクションに起因する厳しい制約の下で実行されるため，身体差異が動作の実行に重大な影響を及ぼす．また舞踊のような動作では，元の動作の重要な特徴も保存する必要がある．これらの要素によって，他の身体で実行された全身の動作を再現することは困難な課題となる．脚は床とのインタラクションにおいて大きな役割を果たすため，脚の動作はこの課題を解決するにあたって特に重要となる．

本論文では，全身の演技における脚の動作を扱うタスクモデルの集合である「脚タス

クモデル」を提案する．脚タスクモデルは身体差異の問題を LFO のパラダイムに従って解決するにあたって中心となる要素である．脚動作の基本単位が「タスク」としてモデル化され，各タスクの特徴はそれぞれの「スキルパラメータ」によって記述される．演技における脚の動作は，それぞれ固有のスキルパラメータ値をもつ多数のタスクの列として表現される．この表現は関節角軌道のような低レベルな動作データとは非常に異なるものである．

脚タスクモデルの目的は，様々な身体を含む様々な状況で同じ全身動作を実行する際に必要となる本質的な情報を与えることにある．舞踊演技における脚動作の本質を与えるため，脚タスクモデルは足裏と床とのインタラクション及び舞踊において保存すべき重要な動作の特徴に着目する．モデルは特定の身体やタスク認識に用いる特定のセンシングシステムに依存しないように設計される．我々はこれらの要素を満たす 4 つの脚タスクモデルを定義する．

タスクモデルが定義されると，次に扱うべき問題は，いかにしてタスク列を教示者による実演から認識するかと，いかにしてロボットがタスク列に従ってタスクを実行するかである．タスクモデルによる本質的な表現は様々な認識手法や様々なロボットに柔軟に対応する可能性を秘めている．本論文では，認識プロセスのためのセンシングシステムとしてモーションキャプチャを用いることとし，キャプチャした人の動作軌道から脚タスク列を認識する手法を提案する．我々はまた，現在の二足歩行ヒューマノイドロボットで標準的な身体メカニズムを持つものを想定した実行プロセスの手法も提案する．特に脚動作に関しては，現在のメカニズムはロボットが人の動作を実行する際に厳しい制約を引き起こす．しかし，脚タスクモデルに基づく我々の手法は，ロボットが自身の身体メカニズムで安定に実行できる動作を生成することが可能である．またこの動作は舞踊の重要な特徴をロボットの身体において可能な限り表現する．

我々は，提案したフレームワークを実際の舞踊と HRP-2 型ロボットで検証した．舞踊は全身で演じられる様々な動作を含む日本の郷土舞踊を用いた．HRP-2 は人と同程度のサイズと重量をもつ二足歩行ロボットである．二人の演者の動作データを光学式モーションキャプチャを用いて取得した．我々のフレームワークは正しい脚タスク列を認識し，認識されたタスク列から HRP-2 で実行可能な関節角軌道を生成した．最終的に，HRP-2 は生成された動作データに従って安定して舞踊を演じることに成功した．これは二足歩行ヒューマノイドロボットの画期的な成果である．

この結果により，脚タスクモデルに基づく我々のフレームワークの有効性が示された．これは同時に，LFO のパラダイムが全身の動作に対して有効であることも示している．この成果は全身動作に関する LFO パラダイムの基礎を与えており，二足歩行ヒューマノイドロボットの可能な用途を広げるものである．

Acknowledgements

First and foremost, I would like to thank my adviser, Professor Katsushi Ikeuchi, for taking me on as his student and for his strong support and encouragement. He gave me the opportunity to conduct research on humanoid robots, which was the field I had desired to study for a long time. The research theme is so exciting and interesting for me that I have plunged enthusiastically into the research with great pleasure. In addition, the research environment provided by him enabled me to concentrate on testing my ideas at the cutting edge of the robotics field. He gave me a great deal of useful advice when needed, and encouraged me again and again. I would like to reiterate my thanks to him for giving me this opportunity and his much-needed support.

I am also very grateful to Dr. Atsushi Nakazawa, for giving me great support in my research. He has supported me as a senior researcher of the research project associated with this study since the project started. He proposed many useful ideas that encouraged the project, which helped my study in various ways. He has always been a kind, excellent senior researcher who gave good leadership to all the members of the project.

I also would like to thank Dr. Hirohisa Hirukawa, the leader of Humanoid Robotics Group, in the National Institute of Advanced Industrial Science and Technology. He generously allowed me to use humanoid robots from his laboratory for my study. In addition, he also gave me much valuable and encouraging advice. I am also very grateful to the members of the Humanoid Robotics Group. Their kind support enabled me to have a pleasant, exciting time in their laboratory. Dr. Shuuji Kajita, especially, introduced me to many useful methods for theoretical problems, and Dr.Kazuhito Yokoi, Dr. Fumio Kanehiro, Dr. Kenji Kaneko and Dr. Mitsuharu Morisawa spent much of their time with me and gave me many useful comments when I carried out experiments on the humanoid robots in their laboratory.

I also would like to thank Ms. Hisako Yamada, who is an excellent grand master of traditional Japanese folk dances. She generously allowed me to use her dance performances for experiments in my study. In addition, she helped many presentations that introduced my research achievement. I am also very grateful to the members of her dance group, Aizu-Gyokusui-Kai.

I would also like to thank Dr. Joan Knapp for spending her time to proofread and edit my publications including this thesis. She also gave me much useful advice on technical writing in English.

I would like to thank the members of the Odori group and the Robotics group in the Ikeuchi Laboratory. I have discussed many interesting problems with them in our research, which encouraged me in my study. In particular, Dr. Shunsuke Kudoh gave me much useful advice for this thesis. I would like to thank my fellows, Tomohito Masuda and Shintaro Ono. They have been good friends in the laboratory, and I have been encouraged by them in graduate school life. I also appreciate the pleasant, exciting environment provided by all the members of the Ikeuchi laboratory.

I gratefully appreciate Taeko. My happy and pleasant time with her relaxed me in my daily work. Finally, I would like to thank my family for their support and encouragement throughout my long-term student life.

Contents

List of Figures

List of Tables

Chapter1

Introduction

In this thesis, we propose a framework that enables a humanoid robot to learn and reproduce human dance motions. As shown in Fig.1.1, the robot imitates a human dance performance. The purpose of this study is to achieve the *Learning from Observation (LFO)* paradigm [9] for this kind of whole body motions.

We employ biped-type humanoid robots as robot dancers, and the robots uses its own legs to support its body during a dance performance. To reproduce such a performance including legs from human motions, is both a novel attempt and a challenging problem. This thesis especially focuses on leg motions, which involve a serious difficulty in achieving our goal.

In this chapter, we first explain the background and the purpose of this study. Then we describe the LFO paradigm, which provides the basis for our framework, while considering the problems in reproducing human motions on a humanoid robot. After that, we introduce existing studies associated with this study, and clarify the difference between the existing studies and our study. Finally, we summarize this chapter, and present the organization of this thesis.

1.1 Background

Development of humanoid robots has recently been increasing in both academic and industrial fields, and the technology surrounding them has progressed simultaneously. Humanoid robots are one of the major research fields in robotics, and much research has been conducted in this area for various purposes. Applications for humanoid robots are expanding, and hardware development is consequently expanding also.

Why do human beings develop humanoid robots ? Why are humanoid robots required ? Probably the most fundamental motivation to develop humanoid robots is our interest in human beings themselves and our desire to create human-like beings

Figure 1.1: The goal of this study. A performance of the robot is reproduced from a human performance. The robot is a biped humanoid robot, and it uses its own legs to support the body during a dance performance.

artificially. Besides this fundamental motivation, humanoid robots have a potential to be used for various purposes. Two of these major purposes are to increase our knowledge of human beings, and to use robots practically in human environments.

To know human beings is one of the fundamental activities of human beings. It is important to know the mechanisms of the human body and human intelligence. In order to know those issues, an analytical approach can be employed; by decomposing a human body and observing human behavior, knowledge of human beings can be obtained. On the other hand, a constructional approach is also valid; an attempt to make human-like mechanisms has potential to bring us deep knowledge of human beings that cannot be acquired by the analytical approach. For example, much research on the human brain is taking place, and an attempt to reveal the brain mechanism by constructing an analogous artificial mechanism on humanoid robots has also been proposed [4].

The other major purpose for developing humanoid robots is to apply them to practical uses. Humanoid robots are expected to do various tasks in our environment as general-purpose machines. There is an opinion that robots should not necessarily have human-like bodies to work; robots should have their own body structure and mechanism rather than human-like ones to exploit advantages of machines. This is certainly a reasonable opinion in a sense. However, human-like bodies and looks provide great advantages if the robots work in the human environments.

First, advantages are caused by the fact that almost everything in the human environment is designed for human bodies. Locomotion infrastructures including sidewalk, stairs, and ladders are designed for the movement style of humans; humans can easily move in such infrastructures. However, machines with other kinds of moving mechanisms such as wheels are not usually reasonable for such environments. Machines and appliances operated by humans are also designed so that humans can easily operate them with their hands, feet, and other body parts. From this viewpoint, if we try to enable robots to move and work in a human environment, the most reasonable approach is to create robots that have human-like bodies rather than other kinds of bodies. Studies on practical applications based on this consideration have recently been increasing. In particular, *Humanoid Robotics Project (HRP)* [13] has studied applications such as construction tasks in which robots cooperate with humans [88], operating heavy equipment [87].

Human-like bodies have another advantage in their looks. First, humans can easily interact with a robot if the robot has human-like looks. With this feature, robots can help humans or work with humans smoothly. Second, there are tasks in which a human-like body itself is fundamental. To perform human-like motions as a substitute of a human performer is one of those tasks, and this use for entertainment has been increasing. Humanoid robots are expected to do various kinds of stage performance such as dances. In fact, a number of humanoid actors have already worked in amusement parks. Although most of the current robots for this use are not autonomous robots with intelligence but just automated puppets fixed with a stand, this use indicates there is a great demand for this kind of application. A robot must perform human-like motions with its human-like body when it plays in substitution for a human performer. In this way, a human-like body is useful in itself.

As described above, humanoid robots are expected to work in the human environment exploiting their human-like bodies. For this use, robots are required to perform various tasks as a general-purpose machine. However, it is almost impossible for developers of robots to make programs for all the imaginable tasks in advance. Hence it is desirable that robots can master various tasks from human teaching in daily life. This ability expands the use of robots, and it also enables a robot to acquire natural, efficient ways of performing tasks using its human-like body as humans do. In addition, from the viewpoint of knowing human beings, the development of this ability will provide hints on the mechanism of human learning ability.

For this purpose, the paradigm of Learning from Observation has been proposed. In this paradigm, all humans have to do in teaching a robot is to demonstrate a task that the robot is expected to master. This is the most efficient way of teaching robots. However, this paradigm has not been achieved for whole body performance of biped humanoid robots, which have recently been developed actively. The main

purpose of this study is to achieve this paradigm for whole body motions of biped humanoid robots through reproducing human dance motions. This will provide a basis for applying the LFO paradigm to biped humanoid robots in order to expand their use. To reproduce dance motions is a challenging problem because dances are a kind of extreme motion that makes full use of the whole body parts and physical capabilities. This characteristic is also effective for our purpose because such exterme motions can reveal many possible problems in reproducing human body motions.

A resulting technology of this achievement is also valuable from the viewpoint of practical use. This technology enables a humanoid robot to be "medium" for reproducing various kinds of human body performances. This medium is far more impressive for users than typical media based on video or computer graphics because users can appreciate "real" performances by real objects, and this medium has a potential to provide real interaction. This is a technology that makes full use of human-like bodies of humanoid robots, which are the most important advantage of the humanoid robots. As one valuable application, the technology can produce digital dance archives in which a wide audience can appreciate actual performances of valuable dances. These archives can also provide an effective way of preserving the traditional dances that are regarded as important intangible cultural heritages.

1.2 Learning from Observation Paradigm

1.2.1 Problem in Teaching a Robot

How to teach a robot a certain operation is one of the key issues in robotics. Robots are expected to be general-purpose machines that can perform various operations. In this study, a humanoid robot is required to master various kinds of dance performances. In order to make robots useful for human beings, the robots must have the ability to master various operations without requiring much effort for humans in teaching operations.

Making a program of robots for a particular operation is the most reliable, straightforward way of teaching the robot in a sense. Since any robot is controlled by a built-in program, creating that program can directly teach the robot what the programmer expects from it. However, programming a robot requires advanced skills; only the specialist in this robot can use this method of "teaching" the robot. Furthermore, such a method requires too much effort and time in many cases even for the specialist. Regarding dance performance, creating a control program for another dance performance is not reasonable.

Teaching by guiding (or *teaching by showing*) is a traditional method for reducing the programming effort of robot manipulators. In this method, an instructor directly

moves a manipulator so that the manipulator follows the sequence of an operation. The manipulator memorizes the trajectory of the operation, and executes the operation by playback of the trajectory. This method can eliminate the complex work required by programming alone. However, this method is not flexible because the manipulator just follows the trajectory in its memory. It cannot sufficiently support operations including force control and direct interaction with other objects. Although the programming work is eliminated, teaching can still require much effort for many operations.

If we use the method of teaching by guiding for dance performances, moving joints of the whole body is very difficult for an instructor. One solution is to replace the body for instruction with another virtual object, for example, a small substitute model of the body [61] or a software tool using 3D-CG [37]. However, composing whole body motions such as dances by setting the joints manually is still complicated work. In addition, expression of motion is fundamental in dance performances. The expression in produced performances, the dancers' skill and the entire performance, reflects the skill of a human operator under this kind of teaching system. This approach does not sufficiently achieve flexibility in dance performances. For example, a performance once taught to one robot cannot be applied to another robot that has a different body structure or a different hardware ability.

1.2.2 Learning from Observation

For the problem of teaching a robot, Ikeuchi *et al.* have proposed a paradigm of *Learning from Observation (LFO)* [9], which provides an advanced framework for teaching a robot. Kuniyoshi *et al.* have also proposed a similar paradigm, which they call *Learning by Watching* [34]. The basis of these paradigms is that a robot learns an operation by observing human demonstrations of the operation. Although internal processes of learning are different, many studies on a robot that learns from humans on the basis of a similar paradigm have recently been active [70].

The framework of LFO consists of three processes shown in Fig.1.2. First, a human demonstrator demonstrates an operation that the robot is expected to master, and the robot observes the demonstration by using a vision technique. Second, the robot recognizes what was done in the operation. In other words, the robot extracts information that represents the essence of the operation from the human motion observed. After the robot recognizes the essence of the operation from the human demonstration, the robot can execute the operation according to the recognition result.

This method enables those who are not specialists on the robot to teach operations without much effort. All a human instructor has to do is to demonstrate an operation in front of the robot as the instructor would do when teaching an operations to another person. This method does not require the instructor to have professional skills in

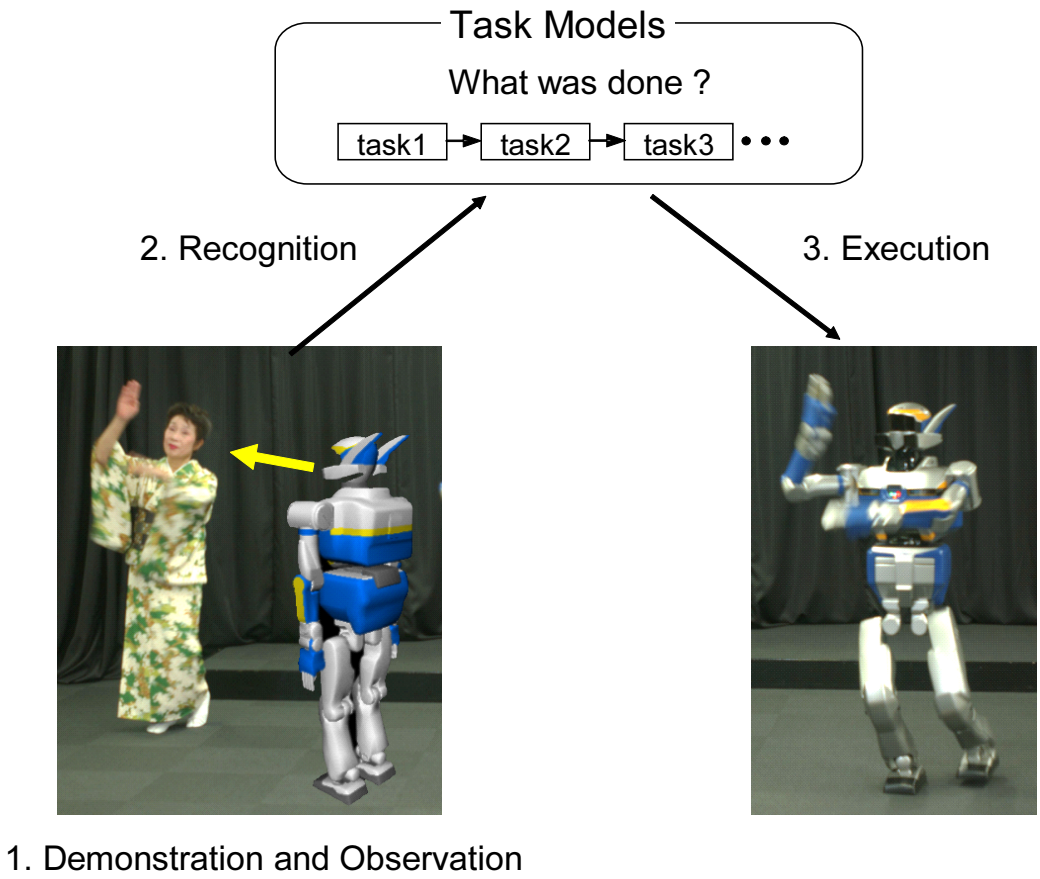


Figure 1.2: Concept image of the LFO paradigm. A performance of a robot is generated from a human performance through the three steps shown. Recognition and execution are processed based on task models defined from the top down.

manipulating the robot. These features are necessary if the robot performs various tasks as a partner of human beings in daily life.

For teaching dance performances, these features of LFO are also great advantages. The work of operating a number of joints of the whole body is eliminated. Instructors can teach complex joint movements just by performing a dance with their bodies. Professional skills for manipulating the robot are not required. Since additional operations are not required in the teaching process, expression in dance performances directly reflects that of the original performer in this method. This is an important advantage over the conventional methods in which motion expression depends not only on a human dance performance but also on skill in operating the robot.

The goal of this study is to achieve this LFO paradigm for the reproduction of human dance motions on biped humanoid robots. When we try to achieve this goal, we must solve the problem caused by the following facts: (1) the body of the robot is different from that of the original human dancer, and (2) the robot must interact with the floor to keep the body in balance under its own body property and mechanism. Since the interaction with the floor is mainly performed by the legs, how to reproduce leg motions is the key issue to solve. We propose a method for solving this problem on the basis of the LFO paradigm.

1.2.3 Observation Process in LFO

In the LFO paradigm, first a robot must observe human demonstrations. This requires a device and a technique appropriate for obtaining movements that are used in the subsequent processes.

In the studies on hand operations based on LFO [60], stereo cameras have been used for observing movement of parts operated by a demonstrator, and a 3D template matching technique has been used for tracking the assembly parts. In this technique, geometric models of the parts are given to the robot in advance, and the robot tracks the parts by matching the models to the depth image acquired by the stereo cameras. For grasping operations, Ogawara *et al.* [59] have used infra-red stereo cameras to track the shape of hands; infra-red cameras can clearly distinguish human hands from a grasped object. Also, a device for the observation process is not necessarily limited to a vision device. Bernardin *et al.* [2] used sensor gloves for obtaining a shape of grasping hands. As shown in these studies, a device and a technique appropriate for target tasks are required for the observation process.

In this study, a robot must obtain whole body motions in the observation process. Currently a technology for this purpose has been established as *motion captures*. We employ a motion capture for the process of observing human dance performances.

Several types of capturing mechanisms have been developed for motion captures.

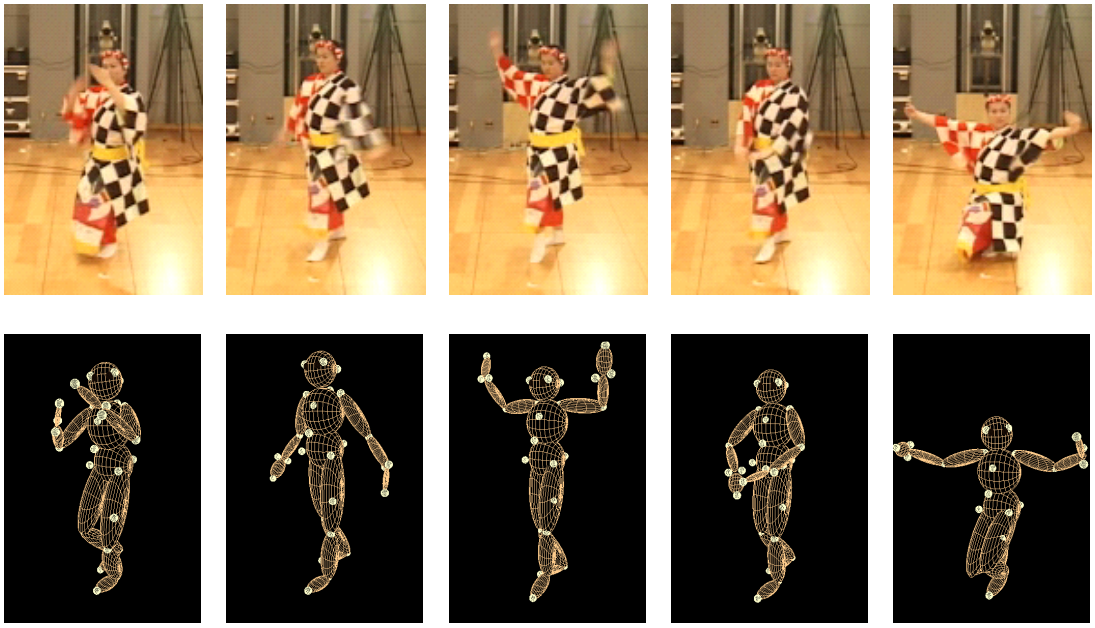


Figure 1.3: Obtaining human body motions by an optical-type motion capture. The upper images are a human motion sequence and the lower images are the captured motion sequence. Position sequences of a number of markers attached to the body are obtained. By fitting those marker positions to a human body model, the original human body motion is restored.

We use an optical-type motion capture, which is currently the most popular mechanism. In an optical system, a number of markers are attached to a human body, and body performances are captured by several video cameras from the different viewpoints. By processing the image sequences captured by the cameras, the system detects the markers and calculates their positions in 3D space. The system finally outputs sequences of marker positions for all the markers. If sufficient markers are attached with appropriate positions of the body, whole body motions can be restored from captured marker sequences. Fig.1.3 shows a capturing example. More detailed description of motion captures are in Appendix A.

In this thesis, we focus on the recognition process and execution process in the LFO paradigm. We use an existing motion capture for the observation process.

1.2.4 Problems in Direct Motion Mapping

Observation of human performances are processed by a motion capture, and trajectories of the markers in 3D space are obtained as the result of the observation. The problem after the observation is how a biped humanoid robot performs a motion based

on the captured motion trajectories.

All a robot seems to do is to move its body so that each body part follows the trajectories of the corresponding marker. However, the trajectories cannot be directly executed by the robot due to differences between the human body and the body of the robot. Although humanoid robots are designed so that their bodies are similar to those of humans, their bodies still have differences in many factors that include joint structure, shape, and composition. Also, the physical capabilities of current robots such as the maximum velocities of joints are less than that of humans in most cases. The differences cause constraints in following the original human motion trajectories.

The constraints become especially obvious in performing dance motions, which generally include dynamic, complex motions that make full use of a whole body. Furthermore, performances reproduced by a robot include leg motions in this study. This means that the robot is not fixed on a stand, but it must support its body with its own legs; the robot does not perform in air, but it must perform on the floor while keeping the body in balance. This condition makes the problem challenging.

Leg motion plays an important role in maintaining the body in balance so that the body does not fall down. This role must be fulfilled on a mechanical structure and mass distribution different from those of the human performer. Also, human soles can contact the floor in a flexible manner using any part from the toe to the heel and absorb impact well. On the other hand, soles of current robots are not flexible; they are just solid plates with few Degrees of Freedom (DOF), so that motions of soles are severely limited for performing stable support. In addition to these leg constraints, general constraints caused by body differences must be considered. Furthermore, the unique characteristics of the dancer's performance must be reflected in the robot's performance. Because of these constraints, if a robot is controlled so that it just follows the human motion trajectories, the robot cannot perform the motion as intended, and it falls down on the floor at once as shown in Fig.1.4.

One possibility for a solution is to improve robot hardware to eliminate all the body differences and constraints so that the robot can perform exactly the same motion trajectory as a human. Certainly hardware improvement is important; the advanced ability of the latest robots is the product of much effort in hardware improvement, and hardware ability will in the future be further improved. However, depending only on this approach is not a reasonable solution to the problem because of the following reasons.

First, to develop the robot body that has exactly the same physical mechanism and capacity as the human body is unrealistic. The differences will decrease as the hardware is improved, but it will be impossible to eliminate all the differences.

Second, this solution is not useful because human instructors are limited to a particular person for a particular robot. Even if all the differences in physical mechanism

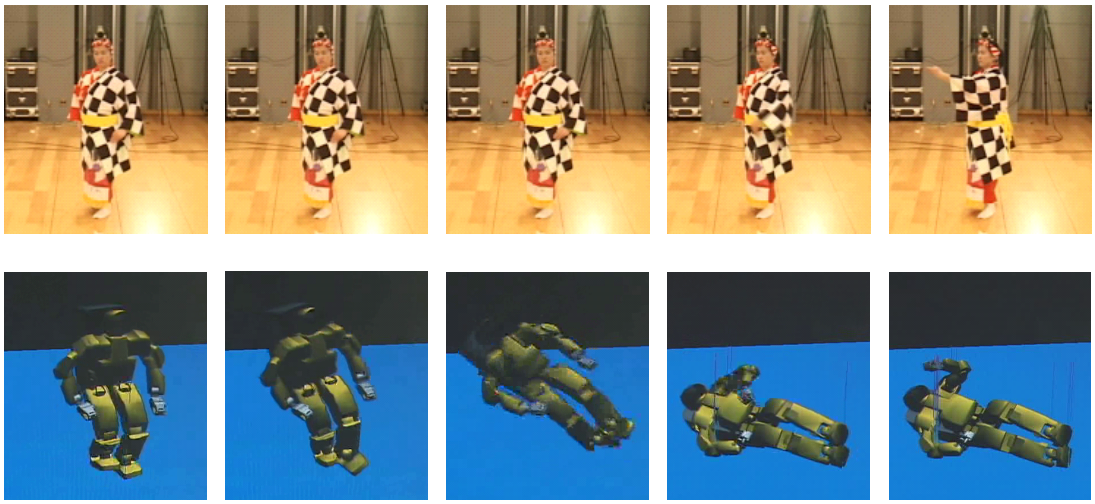


Figure 1.4: An example of falling down. The upper images are the original human motion and the lower images are dynamics simulation of a robot. In this simulation, the robot tried to follow the same motion trajectories as the human. Such an attempt results in the robot immediately falling down like this simulation.

and capacity are eliminated, this is not sufficient for the robot to perform the original motion trajectories. The robot must fall down if weight distribution of the whole body is different from that of the original dancer, because different weight distribution causes a different result of dynamics in the whole body. However, to make the weight distribution of the robot exactly the same as that of the original dancer is unreasonable. To achieve this seems almost impossible, and this solution only enables a particular robot to master the performances of a particular human body. It should be noted that human performers can learn dances from an instructor who has different physical properties and capacities such as body type, mass distribution, and muscle force. These differences present the same problem as those between robots and humans, to greater or lesser degrees.

Finally, the approach in which a robot follows directly the original motion trajectory does not achieve flexibility in performances. This just makes a replica of a particular dance performances under a a particular environment. It is desirable that the robot can perform under various stage conditions. Many dance performances also requires interaction with other dancers or music. This kind of flexibility in performances cannot be achieved by depending on following the original motion trajectories.

The above consideration indicates that differences in bodies and environments are fundamental factors in learning from others. From this viewpoint, this study focus on developing a software technique to achieve our goal under the current hardware mechanisms without depending on hardware improvement. The technology concentrates

on making full use of current hardware capabilities, but It can be used generally for various kinds of hardware mechanisms that include more improved ones in the future.

1.2.5 Recognition Process based on Task Models

In the LFO paradigm, the recognition process between observation and execution is fundamental as well as the observation process. As discussed above, a robot cannot necessarily achieve operations just by following the motion trajectories that are observed from human demonstrations. Like dance performances, most operations are performed under constraints in factors that include physical capabilities and interaction with other objects. The actual motion trajectories in operations depends on those constraints and operations may be also controlled by other factors such as force. This problem is similar to that in the teaching by guiding approach. If a robot performs an operation under various situations, the robot must recognize "what was done" in the operation. In other words, the robot must extract the essence of the operation from the observation result; essence means the information that is necessary for executing the operation correctly in various situations. Then the robot performs the operation not by following the original trajectory but by achieving the recognized essence.

In order to recognize operations, the robot needs to know what is the essence in the operations. The essence will be in different factors depending on a *task domain* of the operations, which classifies kinds of operations. The problem is how a robot obtains this knowledge about a particular task domain: what are the essential factors in the task domain, and how they are represented.

In the LFO paradigm, the knowledge is provided from the top down by human developers. Providing the knowledge is the central issue of the studies based on the LFO paradigm. The approach in which humans provide knowledge is based on the hypothesis that learning ability of the animals depends on whether the species inherits the fundamental framework of the subject to learn. If we apply this to the robots, it means a robot cannot learn and perform any operations in a particular task domain unless the robot has the knowledge of the task domain inherently. In the LFO paradigm, the knowledge is provided in the form of models that represent the essence of operations. These models are called *task models*. A set of task models is defined for a particular task domain. Based on the defined models, the robot recognizes and executes operations in the task domain.

The LFO paradigm has been employed by studies on learning hand operations; those studies are good examples of how task models are defined. Ikeuchi *et al.* proposed *Assembly Plan from Observation (APO)* [9], which enables a robot to learn assembly tasks of blocks. In APO, a state of a block in the assembly process is classified by the state of contact between the block and other objects, and changes of the state

are regarded as the essence of assembly operations [41, 73, 75]. APO defined the task models of assembly operations based on these states. The robot detects changes of the state through observation, and obtains a transition sequence of the state. Then the robot executes the sequence so that the transition is achieved. In a similar way, Morita *et al.* proposed *Knotting Plan from Observation (KPO)* [44, 74], which enables a robot to learn how to knot a string. KPO defined task models based on the knot theory. A state of a string is represented by a state of self-intersection, and essential operations that changes the state are represented by Reidemeister movements. Also, Sing Bing and Ikeuchi [29, 30] studied learning grasping operations based on LFO. In their method, the state of a hand is classified according to contact points between the hand and the object. Change of state is detected by checking contact with the grasped objects at several points along the fingers.

As shown by these studies, task models can be defined by focusing on the essence for representing a state of an operation subject. A robot can recognize the essence of an operation by detecting changes of state from observation, and it executes the operation so that state transition is achieved. This is the basic framework on the LFO paradigm.

The approach based on task models is also valid for leg motions in dance performances. Adjusting human motion trajectories to the legs of a robot is a significantly complex problem because of the severe constraints in the leg motion. Due to the constraints, human motion trajectories necessarily vary when they are performed by the robot. However, the modification itself is not a problem as long as the essence of the original motions is preserved. At this time, factors of motion are classified into two parts: the essential factors that must be preserved, and other factors that can be adjusted to body differences. In the LFO paradigm, these factors are distinguished by task models. A robot generates motions for its body by concentrating on achieving essential factors. Meanwhile, the body-related factors are generated depending on the body of the robot. In this way, the framework works effectively under the severe constraints of leg motions. This approach is more reasonable than the approach in which the robot modifies directly the original motion trajectories for its body.

This thesis proposes *leg task models*, which are the task models for reproducing leg motions in whole body performances. The models represent essential factors in leg performances by focusing on changes in contact between the soles and the floor, and turning points in the movement of the waist. This thesis also proposes a framework of reproduction based on the leg task models. The framework consists of a recognition method and a generation method. The recognition method automatically extracts a sequence of model instances from observed human motion trajectories. The generation method automatically generates motion data that are executable by a robot from a sequence of model instances. This framework enables a biped humanoid robot to

reproduce stable performances that preserve essential expressions of the original dance performances.

1.3 Related Work

In this section, we introduce related studies and compare them with this study. Related studies are classified according to the following viewpoints. First, the history of the development of biped humanoid robots is summarized. This study depends on the latest hardware technology of biped humanoid robots. Then we present an overview of studies on basic control methods of biped humanoid robots. A part of our method is based on theories proposed by those basic studies. Next, we introduce studies on processing human body motions. Those studies have been actively pursued, especially in the field of computer graphics, in order to produce human-like motions. Those studies are related to our study, but they are not sufficient for our purpose. Then we introduce various attempts to construct models of human body motions. Those models are developed as a basis for classifying, synthesizing, and learning human body motions. The policy of our model design is clarified by comparing it with those studies. Finally, we introduce studies on dancing humanoid robots. Many researchers have developed dancing robots that have their own purposes and characteristics. We clarify the novelty of our attempt by comparing it with those dancing robots.

1.3.1 Development of Biped Humanoid Robots

Humanoid robots are those robots who have human-like looks and abilities. Various types of humanoid robots have been developed for various purposes. In this study, we use a humanoid robot HRP-2. It is one of the latest biped humanoid robots that have human-like whole bodies including arms and legs; its legs can support its body and it has a potential to perform human-like whole body motions on the floor. Here we introduce humanoid robots that are a similar type to HRP-2, reviewing the development history of those humanoid robots. HRP-2 was developed on the basis of those robots.

The first humanoid robot that has a whole body including legs was Wabot-1 [31], which was developed by Kato *et al.* of Waseda University in 1973. Wabot-1 had entire body components that include eyes, arms, hands, and legs, and also had the ability to walk using its own legs. It also had the ability to operate objects using its hands. However, its performance was not as advanced as that of the latest robots. For example, the walking speed was very slow because its walking style was so- called *static walk*, in which the projection of center of mass (CM) on the floor is always inside the *support area*, which is a convex area full of *support soles*. However, this study was

a major milestone in the field of humanoid robots because it actualized the basic form and abilities of a humanoid robot with the entire body for the first time.

After Wabot-1 was developed, a more advanced way of walking had been aggressively studied in the field of biped robots. Development of practical walking ability was required first because walking is the fundamental locomotive method of biped humanoid robots. In the early stage of those studies, the goal was to achieve so-called a *dynamic walk*. In a dynamic walk, the CM projection is not always inside the support area. This style of walk is far more natural and efficient than static walk, but control is more difficult. Many researchers tried to solve this challenging problem, and many biped robots that did a dynamic walk were produced. However, a dynamic walk by those robots was not necessarily stable. Those robots could not prove sufficiently whether biped walk by human-size robots was practical or not.

In this situation, Honda released a biped humanoid robot P2 in 1996 [6]. P2 had an entire body component from head to feet; its height was 1.80[m], which is a similar size to a man, and its weight is 210 [kg]. The most advanced ability of P2 is its walk. P2 achieved smooth, fast, and stable walking that seemed far more practical than that of the existing robots. The walking motion looked like a human walk. P2 had a battery on its back so that it could move without cables connected to the external battery and the control system. This was also an advanced feature of P2. Honda then developed improved robots that were more compact and lighter. First P3 was released; its height was 1.60 [m] and its weight was 130 [kg]. Then Asimo [69] was released; its height was 1.20 [m] and its weight was 43 [kg].

P2 was epoch-making work in the field of biped humanoid robots. The robots proved that practical walking ability can be achieved on human-size biped robots. Most parts of the body structure and mechanism of their robots were standard technologies in biped humanoid robots. For example, P2 employed harmonic gear drives for joint actuators; their features enabled stable control for producing a stable walk. Nowadays harmonic gear drives are the standard mechanism of actuators of biped humanoid robots. The release of P2 might have discouraged some researchers who were working on biped robots. However, P2 eventually encouraged researchers in developing more advanced robots. P2 also brought public attention to humanoid robots.

The development of biped humanoid robots has been actively pursued in many research institutes and companies since P2 was released. At the University of Tokyo, Kagami *et al.* have developed biped humanoid robots H5[48], H6 [15] and H7 [14] since 1998. These robots achieved a stable walk like Honda's robots. H7 had advanced features including toe joints in its soles. The toe joints could extend the range of walking methods. For example, they enhanced the maximum height of a stair the robot could climb. These robots were used as a platform for studying various abilities required for humanoid robots.

At Waseda University, Ogura *et al.* [62] have developed Wabian-2, which achieved advanced walking ability. It could walk in a natural style in which the knee joints were not always bending. Previous robots could not walk in such a style because the leg fell into a singular point when the knee joint was fully extended. They solved the problem by using additional Degrees of Freedom (DOFs) in the coxa joints.

The Ministry of Economy, Trade, and Industry of the government of Japan has been working on the *Humanoid Robotics Project (HRP)* [13] since 1998 for the purpose of developing a basic technology and investigating the potential of applications. In the early years of the project, P3 was used as a robot platform. The built-in software system of P3 was replaced with that newly developed by Yokoi *et al.* [86]. This version of the robot was called *HRP-1S*. Then the project developed new robots. A prototype was called *HRP-2P* [28], and the completed product was called *HRP-2* [26], which we use in our study.

HRP-2 has a more compact, lighter body than HRP-1S. Its height and weight are 1.54[m] and 54[kg], respectively. HRP-1S has a big backpack of a battery, but HRP-2 does not have it, so that the trunk of its body is slimmer than that of HRP-1S; a battery is installed in the waist in HRP-2. The characteristics of its hardware mechanism are in its joint structure. In the robots described above, a coxa joint links to the center of a thigh root. On the other hand, a coxa joint of HRP-2 links to the side of a thigh root. This structure enables HRP-2 to cross its legs widely. Also, HRP-2 has a two-DOF joint (roll and pitch) between the chest and waist. This joint enables the robot to pose in various ways.

Now Kawada Industries has manufactured HRP-2 products, and information about its system architecture is available. These developments have enabled researchers to concentrate on studies of the latest hardware, and several research institutes have carried out various studies on HRP-2.

Sony has developed a small biped humanoid robot called QRIO [38], which can perform dynamic, high-speed motions by exploiting its size merit.

In the latest robots described above, the basic mechanisms are based on that established on P2. In this mechanism, body frames are basically a form of high-rigid external skeletons that include electric devices and joints. The joints of the body consist of a combination of one rotational axis. Each joint axis is actuated by an electric motor with a harmonic gear drive. Basic orders of joint axes in arms and legs are also established. A battery is installed in the body trunk. The controllability of robots has been improved on this mechanism, and the mechanism has been standard for robots that achieve stable biped walk. The method of this thesis is basically designed so that it is applicable for any robots that have this kind of standard mechanism.

On the other hand, different mechanisms have also been developed. Mizuuchi *et al.* have developed humanoid robots whose body is driven by a novel mechanism

[42, 43]. Their robots have a flexible spine in the trunk like that of the human body. In the standard mechanism, a trunk of the body consists of few rotational joints. This new spine mechanism enables dynamic twisting motions of the body trunk that cannot be performed on the standard mechanism. Okada *et al.* have developed a humanoid robot $UT-\theta$ [64]. This robot has novel mechanisms in the shoulder joints, the coxa joints, and the knee joints. The mechanism of the shoulder joints is called a *cybernetic shoulder* [63]. This joint does not consist of a combination of rotational axes, but it is a closed loop mechanism of two parallel axes and one rotational axis. This mechanism can perform human-like, flexible movements of shoulders. In the coxa joints, all the rotational axes intersect at the same position. This mechanism enables the body trunk to freely rotate with simple joints. In the knee joints, *backlash clutches* are installed. This mechanism allows the release of torque from the knee joints. The object of these developments is to achieve more human-like leg motions that include a passive actuated phase.

However, these robots have not achieved stable walk yet. These kinds of novel mechanisms will require novel control methods. This thesis does not deal with these robots, but it will expand the possibilities of performances on humanoid robots generally.

1.3.2 Control of Biped Humanoid Robots

A biped humanoid robot must maintain its body balance when it performs body motions on the floor. Otherwise, the robot falls down to the floor. A control method are also important in maintaining body balance in addition to a controllable hardware mechanism. Many control methods for this purpose have been studied in combination with hardware development.

At the least, motions performed by the robot must be consistent in the dynamics between the body and the floor. If the robot tries to execute a motion that lacks this condition, the body moves in the different way from the planned motion, so that the robot will fall down. Motions of biped robots are basically determined by methods that focus on this condition.

Kajita *et al.* [17] have proposed a method for generating walking motions based on a model called *Three-Dimension Linear Inverted Pendulum Mode (3D-LIPM).* In their method, a walking robot is modeled as the behavior of an inverted-pendulum (IP). In this model, the weight of a pendulum represents the center of mass (CM) of a robot, and the rod of the pendulum represents a leg (a line from a support sole to CM). The motion of CM is given as falling motions of the IP under the condition in which the CM is always at a constant height from the floor. By changing a pivot of IP with each step, the method generates a CM trajectory for a given stepping pattern. The

approximate model of this method seems too simple to represent complex bodies of robots. This simple model aims to provide a robot with walking ability. The outputs of this method are practical enough to be used for this purpose. The method has been used for the robots of HRP.

However, this method is not appropriate for generating more complex motions like dances. If an upper body performs dynamic motions, errors between this simple model and actual motions increase and the robot will be uncontrollable. Also, the vertical waist position is fixed at a constant height. These characteristics constrain possible expressions. In addition, steps are not strictly determined in detail because timings of steps are fixed at a constant cycle, and given positions of steps may be changed slightly. These characteristics are not appropriate for dance performances in which irregular steps must be performed in a strict form.

In order to determine a strict form of steps, methods based on *Zero Moment Point (ZMP)* are practical. The concept of ZMP was proposed by Vukobratović *et al.* [81, 80]. The concept is useful in considering whether the robot falls down or not. When a robot is in a static state, the robot does not fall down as long as the CM projection on the floor is inside the support area consisting of soles. ZMP can be regarded as a dynamic version of this point. ZMP is always inside the support area in motion. We provide a more detailed description of ZMP in Section 5.5.1. Several methods that determine motions based on a given ZMP trajectory have been proposed. If a given ZMP trajectory is always inside the support area, the determined motion can be performed without falling down in theory. Even irregular step patterns can be achieved by giving the detailed ZMP patterns for the steps.

Takanishi *et al.* [77, 82] proposed a method that compensates ZMP with movement of a body trunk so that a given ZMP is satisfied. This method uses techniques of the fast Fourier transform (FFT) and the inverse FFT for calculating a solution. The method basically deals with cyclic motion patterns. Long irregular patterns can be solved by giving a long period of motion as one cycle. In this case, the method requires long calculation time, which is a disadvantage of this method. The model of this method that represents a correlation between a body and ZMP is as simple as 3D-LIPM. However, the process of this method can be iterated so that the error of the output converges. This feature enables a robot to perform motions using the whole body.

Other methods for the same purpose have also been proposed by Nagasaka *et al.* [47], Nishiwaki *et al.* [57] and Kajita *et al.* [16]. In Nishiwaki's method, correlation between ZMP and a body position are expressed by simple discrete equations, and the equations are solved as tridiagonal simultaneous linear equations. Since this kind of problem is solved by an efficient algorithm of matrix computation, the method can output the results more efficiently than Takanishi's method. They also proposed a

system that uses this method for on-line motion generation [58, 56]. That system has been used on the humanoid robots of the H series. Kajita's method uses a technique of *preview control* for determining motions. Their method is appropriate for online use. The method has been used on the HRP series. Both methods can be iterated so that model errors converge as in the case of Takanishi's method.

This study employs the ZMP-based approach to satisfy the dynamic body balance because this approach can achieve both accurate step patterns and dynamic upper body motions that are required for dances. We use Nishiwaki's method to determine the horizontal waist movement in a process of generating robot motions because that method has advantages in efficiency and simplicity.

Although these ZMP-based methods are useful, they can solve only a part of our problem. All these methods determine is a global body trajectory that achieves a given ZMP trajectory. The ZMP trajectory and motions of detailed body parts must be determined independently, and this process must consider many factors that affect stability. Dance performances push a robot to the limitation of the hardware capability. Such a performance reveals factors that make a robot unstable; those factors are not a problem when the motions leave sufficient margins for the limitation of the machine. In fact, sufficient margins have been allowed in generating motions of biped humanoid robots in most existing cases. Otherwise, detailed parts of motions would be determined manually so that unstable factors are eliminated. In this study, motions are automatically reproduced from human dance performances that do not consider those factors of the robot at all. We propose a method that considers various factors related to stability that were not sufficiently considered in the existing methods.

As described above, motions of biped robots are determined so that they satisfy the dynamic body balance. However, even if a robot that has a controllable mechanism follows those motions, the robot cannot necessarily perform the motion without falling down. This is due to the uncertain factors in actual performances: errors in theoretical models, errors in control, and disturbances produced by environments. Those factors will produce the errors between a planned motion and an actual motion, and the errors will diverge to cause falling down. Accordingly, a control method for correcting the errors during execution is required for a control system. Those stabilization methods have also been studied in step with the improvement of walking ability.

When a motion is planned by a ZMP-based method, it is reasonable to correct errors by focusing on ZMP. In this approach, a robot obtains an actual ZMP from force sensors in the feet during a performance, and the robot adjusts the planned motion so that the actual ZMP follows the planned ZMP. Nagasaka *et al.* [46] have proposed a method that corrects difference in ZMP by adjusting the horizontal movement of a body trunk. This method has been used on the H series. A similar method has been built into the control system of the HRP series [86]. Hirose *et al.* [7, 6] have proposed

another method called *model ZMP control.* This method corrects the error in a body inclination. When the body inclines toward the front, the body is accelerated ahead so that the inclination is resolved. This method has been used on the robots produced by Honda. These methods compensate errors with local modification of a planned motion.

Consequently, they do not necessarily solve significant errors that are caused by unexpected factors such as a bump of the floor. Kajita *et al.* [18] have proposed a method for solving this problem. Their method globally modifies planned motions when unexpected errors occur. The method enables a robot to perform adaptive walk on a bumpy floor. Slip is also an unexpected factor in walking. Kaneko *et al.* [25] have proposed a method for stabilizing walking on a slippy floor.

After simple walking became standard technology, running by a biped robot was studied. Kajita *et al.* [20] has proposed a method for generating running motions. This method calculates a CM trajectory of cyclic running motions. They used this method for a biped robot HRP-2L [21], which has the same body parts as HRP-2 except for the upper body, and achieved a running motion by HRP-2L [19]. Nagasaka *et al.* [49] has proposed a general theory of dynamics in motion that include a period when a whole body floats in air. They enabled QRIO to run on the basis of this method. Honda also achieved a running motion by Asimo. These studies expanded the physical capability of biped humanoid robots.

Currently these running motions have not been the natural motions that humans do. Running in these studies means that the motion has a period when both soles float in the air. In the current achievement, the floating period is an extremely brief moment, with a floating height close to the floor. Although Raibert [67] developed a running robot that achieved sufficient floating period and height, his robot had only one leg and it must always been jumping. The existing hardware mechanisms and control methods of biped robots cannot absorb the impact of landing well. To achieve stable, natural running motions on biped robots is still a challenging problem.

The above studies deal with only the interaction between the soles and the floor. On the other hand, controlling whole body motions that include interaction using other body parts have also been studied. For example, a rising motion from a lying position has been achieved. Kanehiro *et al.* [22] have proposed a method that controls a rising motion by making a transition between statically stable poses. Although this method takes a certain time for a rising process, it achieves reliable rising motions.

Kuniyoshi *et al.* [35, 79] have achieved roll-and-rise motions on a humanoid robot RDS1 that was developed by Nagakubo *et al.* [45]. A roll-and-rise motion is a dynamic, high-speed motion that enables a robot to rise from a lying position; the robot lifts its legs high from a lying position, and lowers the legs swiftly to role and rise from that position. They have proposed a paradigm of *global dynamics structure* through

this attempt. This paradigm aims to enable a robot to perform more dynamic, high-speed motions that make full use of the whole body than the existing control methods. They proposed a key to this problem through experiments on roll-and-rise motions. Parts of motion trajectories that determine the success or failure are limited in time and space. This suggests that motions can be controlled by focusing on those local parts. Although their roll-and-rise is an impressive motion that had never been seen in existing performances, the validity of their paradigm has not been sufficiently proved for various kinds of motions.

1.3.3 Processing of Human Body Motion

If a biped humanoid robot can make full use of its whole body, the robot can perform more various motions than when it only uses its legs and moves on the floor. The ability of whole body motions expands the possible uses of the robot. Dance performances are one of those uses. As the total technology of biped humanoid robots has been improved, studies for this purpose have increased.

One approach is to adapt motion data to a robot body so that the motion can be stably performed by the robot. Tamiya *et al.* [78] have proposed a method called *auto balancer*, which is based on this approach. Motion trajectories of all the joints are given, and the method modifies the given joint trajectories so that they do not cause falling down and spins. This means that the method can be used as a filter of low-level motion data of joint trajectories. This filter directly provides motion data with consistency in dynamics between the robot and the floor. This method enabled robots to reproduce human motions, including leg motions. However, this method can deal only with the situation in which the robot keeps standing on one foot. It is useful for such a situation, but insufficient for total balance maintenance in dance performances including steps.

A more general-purpose method using this approach has been proposed by Yamane *et al.* The method is called a *dynamics filter* [84]. The filter modifies motion trajectories so that inconsistent dynamics on a body model are converted to consistent dynamics. This filter is useful because it can adapt the motion of one body to another body. However, satisfying dynamic consistency by this method does not necessarily mean producing stable motions. If body difference is not trivial, resulting motions can be unstable in spite of satisfying dynamics. The body may fall down while all the body motions are following the law of dynamics. This method is practical enough for CG animation, in which body motion is not constrained by any extra factors. However, this feature is not appropriate for our purpose because body differences between our robot and dancers are not trivial and an actual robot has many extra factors that affect stability. Thereby the filtered motions are not necessarily performed stably by

the robot without falling down during extreme motions.

These studies attempted to develop a general-purpose method for adapting arbitrary motion data to a particular body. In order to achieve generality, the methods are designed as a uniform, bottom-up process; the methods integrate the modifications of local joint motions to obtain a desired motion. Although a general-purpose method using a uniform process is useful, such a method does not necessarily provide the best solution to a specific area, especially for a robot to perform dance motions on the floor.

Manual edits are still a well-used approach in generating whole body motions. There are methods that can reduce human effort in manual editing. Kuroki *et al.* [37] have developed a software tool that makes manual edit easy. This tool automatically generates balanced body trajectory for edited motion patterns. Yamane *et al.* [85] have proposed a novel method of inverse kinematics (IK). In their method, arbitrary body parts can be fixed in space during IK. This feature makes editing whole body poses by IK significantly efficient. Those methods make manual edit one of the valid approaches in generating whole body motions. However, this is not our approach.

In the field of CG animation, methods of processing whole body motions have been aggressively studied in order to generate natural, human-like motions. Like the auto balancer and the dynamics filter, adapting motions of one body to another body has actively been studied in this field [39, 5, 72].

Studies on reusing captured human motions have also been made. Kovar *et al.* [33] have proposed *motion graph*, which is a method for synthesizing new motions from a database of a number of captured motion units. The method basically synthesizes motions by connecting motion units in the database. A sequence of motion units is optimized so that connected motions appear natural. A similar method was also proposed by Lee *et al.* [40].

On the basis of a motion graph, Nakazawa *et al.* [55] have proposed a method for synthesizing dance motions that are synchronized with given music. Edges of motion units are synchronized with beats of the music, and motion units that include more dynamic motions are used according to the intensity of the music. These criteria result in motions well matched to the music. This method could also be useful for dancing robots if the method were available for robots.

These studies in the CG field are suggestive for generating motions of humanoid robots. However, these methods cannot be used directly for humanoid robots because they basically concentrate on how natural motions look, but they do not necessarily satisfy the dynamics and constraints of the body in a strict sense. These factors are not required for CG animation, but they are required for performances by an actual humanoid robot.

1.3.4 Models of Human Body Motions

Processing raw motion data that just represents entire trajectories of body parts is not necessarily sufficient for high-level, abstract operations of human body motions. Models for representing and understanding human body motions are useful for such operations, as our goal is achieved by leg task models that represent essential leg motions. Many studies have also proposed their own model of human body motions for various purposes.

Yukawa *et al.* [89] have developed a *Buyo-fu* system that enables advanced use of captured dance motions. Buyo-fu is a Japanese word that means a score of a dance like a musical score. Their system modeled captured motion data as structured form. The whole body is separated into four parts: head, body, arms, and legs. For each body part, basic units of motion that have a certain period are provided. Whole body motions are represented by sequences of those units for all body parts. The system shows labels of motion units, so that entire motions are symbolically represented. This mechanism provides a similar function as a musical score. A user can check each basic motion independently and understand whole dance motions as structured forms. The system can connect motion units smoothly so that a user can also compose new dance patterns by putting units on the editor. These functions are useful for presentation and training of dancers. In fact, the system has been used by a theater company Warabi-za [1].

Their study proved the validity of a symbolic, structured model of human body motions. However, their system cannot automatically extract the elements of the model from captured motion data. Basic motion units are manually detected and classified by a human specialist on target dances. This means that the system depends on specialists and requires a burden of work for them. It is desirable that the system can automatically extract the structured motion units from the captured data.

Nakazawa *et al.* [52] have proposed a method for extracting a structure of dance performances from captured motion data. This is an attempt to automatically extract basic motion unit similar to that of the Buyo-fu system. The same features as the Buyo-fu system can be available without manual operations by specialists. The method first segments a whole motion sequence by focusing on "pausing motions" of hands and feet. Then it clusters the detected segments into a number of groups. Each group represents one basic motion unit. Most boundaries in the segmentation by human specialists are detected by this method. However, the system tends to output additional boundaries, so that results do not necessarily correspond to those of a specialist.

Shiratori *et al.* [71] have proposed a method that increases the accuracy of Nakazawa's method. Their method uses music that is being played during a dance performance. The method segments a motion sequence by analyzing wave data of

music as well as captured motion data. The beat timings of the music are detected, and only boundaries that are close to those timings are recognized as valid ones. Since the dancer basically performs to music, this approach works well. This method can provide segmentation results that are very similar to the result of human specialists.

On the basis of the structural motion models, more advanced ways of reusing motion data have been proposed. Nakazawa *et al.* have proposed a concept of *base motions* and *styles* [53]. They represented the trajectory of a basic motion unit as the synthesis from two motion trajectories. One is a base motion, which is a direct path in joint space from the initial pose to the final pose. The other is a style in which the base motion is subtracted from the original trajectory. In general, base motions of one dancer are similar to that of another dancer, and styles differ according to dancers. Their method can extract individual styles of motions, and it can control the style factor in motion synthesis. For example, styles of different dancers can be blended in an arbitrary ratio.

Inamura et al. [12] have proposed a general method for recognizing and generating body motions. They integrated generation models with recognition models based on Hidden Markov Model. The main feature of their method is in this integration. Once recognition models are constructed from samples of human body motions, the acquired models can also be used for generating and synthesizing motions from memorized motions. This method enables a robot to recognize human motions and imitate them. It should be noted that a unit of recognition and generation is basically a certain length of sequential motions rather than primitive units because the method does not segment the given motions. The method is not aimed at representing a whole dance motion by a sequence of basic motion units.

In the method described above, motion models are basically constructed from motion data in a bottom-up way, and contents of structured models are parts of raw motion data. These features are reasonable for general-purpose use. However, if a robot tries to perform motions according to these models, it is difficult to complete the motions that include interaction with other objects or environments. Leg performances on the floor are among those motions.

In contrast to these bottom-up approaches, there is a top-down approach in which motion models are designed for a particular domain of motions. If factors corresponding to the possible interactions in the domain are well represented by a model, the model will deal with those interactions. However, this approach loses generality because target motions are limited to a particular domain. After all, the top-down approach and the bottom-up approach are trade-offs between generality and feasibility. Leg task models in this thesis are constructed by employing the top-down approach so that actual dance performances by robots are achieved.

1.3.5 Dances by Humanoid Robots

To make a humanoid robot perform motions that fundamentally require a human-like body is a valuable application of technology because only humanoid robots are the machines that can perform those motions as a substitute for humans. Dances are one of those applications, and many studies on dancing humanoid robots have been conducted [10].

Ikeura *et al.* [11] attempted to make a robot dance according to human dance motions. They investigated minimum Degrees of Freedom (DOF) required for expressing human dance motions, and proposed a method for optimizing motions for a robot with reduced DOF so that the robot can express similar motions to humans. This study is useful because many robots do not have sufficient DOF for expressing human motions. However, their method can deal only with the problem of DOF. Bodies of robots cause many other problems when they perform human motions.

Pollard *et al.* [65, 68] were first to enable a robot to perform human-like, smooth dance motions. They used human dance motions for generating joint trajectories of the robot. Human motions were obtained by a motion capture. They developed a method to adjust captured motion data to the joints of the robot. First the captured marker trajectories were converted into joint trajectories. Then their method constrained the joint trajectories so that the trajectories were under the limit of the joints. The generated motions were successfully performed by a robot that had whole body joints that are driven by hydraulic actuators, and a waist fixed on a stand.

They achieved reproduction of human dance performances on the robot, which may seem similar to our goal. However, it should be noted that their robot was fixed firmly on a stand and it performed motions in air. On the other hand, our attempt includes leg motions; a robot performs dances on the floor. This factor makes our goal quite different from their achievement. Their method is useful for generating upper body motions, so we use a part of their method in the process of generating upper body motions. Besides, we reproduce motions of the lower body through quite a different process, which is a main part of our study.

Yamane *et al.* [83] have proposed a method for controlling the strings of a marionette according to captured human motions. In a marionette, motions of the strings and motions of the joints are not in a one-to-one relationship. In addition, unintended swings of body parts are apt to arise in control, because body parts are suspended by strings of a certain length. Their method solves these problems. This study is useful because it also attempted to reproduce human performances on a machine that has a different body mechanism from humans. However, the concrete problem to solve is different from that of our study.

Sony has produced dances performances including leg motions; a robot performed

dances by using its legs on the floor. They first used SDR-3X, which was a small biped humanoid robot, and recently they have used an improved robot called QRIO [38]. This is an achievement of a biped humanoid robot in the sense that they have expanded the potential of robot performances. However, the goal of our study is different from that of this achievement. In their dance performances, the motion was manually edited by a human designer from scratch so that the motions satisfied constraints and dynamics of the robot [37]. On the other hand, our intent is that a robot reproduces dance motions that are actually performed by human dancers. In addition, they used a small robot, but we use a robot that has similar size and weight to the human body. QRIO is 0.58[m] in height and 7[kg] in weight, while HRP-2 is 1.54[m] and 54[kg]. These differences cause additional problems when the robot performs dynamic whole body motions. For example, the magnitude of the impact when a sole lands on the floor tends to be considerable in a bigger robot. In these respects, our study is a novel attempt.

Kosuge *et al.* [32] have studied a humanoid robot that can perform a ballroom dance with a human dancer. Their robot adjusts its movement in response to the movement of a human partner during a dance performance. Their robot is not a biped robot; all the joints are in its upper body and it moves by wheels. This study is hence not a novel attempt in a sense of expanding the physical ability of whole body performances. The purpose of their study is to develop an ability to interact with humans.

As described above, many studies on producing dances by robots have been conducted with various purposes and approaches. This indicates that there has been much demand for robots that can dance. With this background, our goal is to achieve the following features simultaneously from the viewpoint of dance robots:

- Motion of the robot is not created by a human operator, but automatically reproduced from motions of existing dances performed by a human dancer.
- The robot performs dances maintaining its body balance with its legs on the floor.

This is novel attempt as compared with the existing studies on dancing robots.

1.4 Summary

In this study, we enable a biped humanoid robots to learn and reproduce human dance motions on the basis of the Learning from Observation paradigm. The purpose, issue and solution of this study are summarized as follows.

Purpose To achieve the *Learning from Observation (LFO)* paradigm for whole body motions including legs on biped humanoid robots.

Biped humanoid robots have recently been developed actively, Those robots are expected to perform tasks that make full use of their human-like whole body. If the LFO paradigm is applicable for those tasks, the use of those robots will be significantly expanded, especially for use as partners of human beings.

Dance performances are a good target for establishing the basis of the LFO paradigm for biped humanoid robots. Dances involve extreme motions that make full use of the whole body. This property reveals the problems and requirements in the learning process. Furthermore, dance performances themselves are a valuable application of the robots.

The robot learns and perform motions includes legs, which is a novel, challenging factor. Although there have been studies that reproduced human motions on robots, those performances did not include leg motions that actually support a body. Also, existing whole body performances including leg motions were not reproduced from human motions.

It should be mentioned that we do not develop or improve the hardware of robots. This study concentrates on developing a software method. We use the latest robot hardware with the standard mechanism that has achieved a stable walk.

Issue How to adapt human leg motions to the legs of a robot, which are under severe constraints caused by the interaction with the floor and the difference between the human body and the robot body

In the observation process, motion trajectories of human body parts are obtained by the existing technology of motion captures. However, those data cannot be executed directly on robots. When a robot tries to follow the human motion trajectories, leg motions are severely constrained as follows.

- The weight distribution of the body is different from that of the human body, so that human motions cannot keep the dynamic body balance.
- Soles of robots are not as flexible as human soles, so that human motions cannot keep stable contact between the soles and the floor.
- Because of the limits of joints and actuators, the robot cannot follow all the parts of human motion trajectories.

Under these constraints, a robot is required to express essential characteristics of the original dance motions as far as possible.

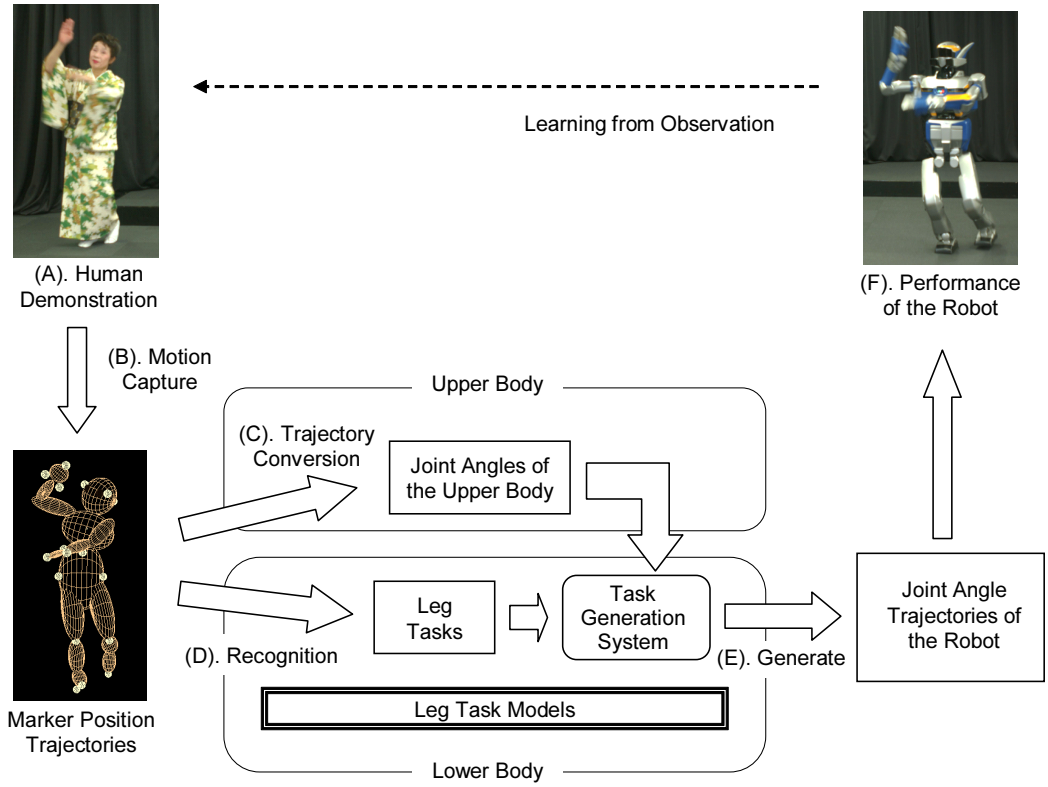


Figure 1.5: Overview process from a human dance performance to a performance by a robot

Solution Model-based motion recognition and generation based on the LFO paradigm

We define leg task models, which are top-down models for representing the essence of leg motions. Adapting human motions to a robot follows two processes based on the leg task models. First, instances of the models are recognized from a human performance. Next, a motion of a robot is generated from the recognized instances. We develop methods of recognition and generation required for these processes. The leg task models and these methods solve the problem in reproducing human motions on robots.

1.5 Thesis Overview

Fig.1.5 shows the overview process from a human dance performance to a performance by a robot.

First, human dancer performs dances. The motion data of the performance are obtained with a motion capture. Details of motion captures are described in Appendix A.

The obtained motion data are used for generating motion data of a robot. The generation process is separated into the process of upper body and the lower body.

As for the upper body, motion data is generated in a conventional way. First, captured motion trajectories are converted into joint angle trajectories of the robot. Then the trajectories are constrained so that they are under the limit of joints.

As for the lower body, motion data is generated on the basis of the leg task models. First, a sequence of leg tasks is recognized from the captured motion trajectories. Then the leg tasks are input into the task generation system, and the system generates the motion data of the robot from the input tasks. The system also uses the motion data of the upper body in this process.

Finally, the generated motion data are input into the robot. The robot reproduces the human dance performances according to these data.

The thesis is organized as follows. In Chapter 2, we discuss the central issue of the LFO paradigm, and describe the basis of designing task models. Through the discussion of this Chapter, the concrete problems to solve in this thesis are clarified. In Chapter 3, leg task models are defined. Then a method of recognizing leg tasks from captured motion trajectories is described in Chapter 4. The method is used in the process (D) in Fig.1.5. After that, a method of generating leg motions of a robot from leg tasks is described in Chapter 5. The method is used in the process (E). Then experimental results are presented in Chapter 6. Finally, we conclude this thesis in Chapter 7. In Appendix D, a method of converting upper body motions is described. This method is used in the process (C).

Chapter2

Design Issue of Task Models

The purpose of this study is to prove that the paradigm of Learning from Observation can be achieved for whole body motions of biped humanoid robots. For this purpose, we propose a framework based on the LFO paradigm that enables a robot to learn and reproduce dances from human demonstrations, and prove the validity of this framework using an actual robot. The central issue of the LFO paradigm is how to design task models, which are core elements in the framework. In this chapter, we discuss this issue.

2.1 Central Issue of the LFO Paradigm

The most fundamental elements of a framework based on the LFO paradigm are task models. Appropriate task models make the advantages of LFO available. Thereby the central issue of achieving LFO is how to define task models. So we describe why task models are necessary for the framework of teaching robots by considering the advantages of LFO.

The framework of the LFO paradigm consists of three processes: observation, recognition, and execution. First, a demonstration of a task performed by an instructor is observed. Then, the result of observation is recognized as a representation of task models. Once the representation of task models is obtained, the representation can be mapped onto robots, and the robots can execute the task. Fig.1.2 shows the overview of these processes.

The most important feature in this framework is that the execution process and the recognition process are separated from each other. Task models that make this separation possible; task models connect these two processes. In this structure, the task models can concentrate on the representation of tasks without considering the recognition process and the execution process. In other words, the representation does not depend on a specific recognition technique or a specific robot. This can make

the representation essential. At the same time, this structure enables the framework to replace a recognition technique and the hardware of a robot independently. The framework can employ various recognition techniques according to its purposes. The framework also enables various robots to execute tasks while expending minimum effort. If a replaced robot has different physical characteristics and capabilities so that the existing generation process cannot support it, all the system must do is change the way of mapping tasks onto the robot in the generation process. In these replacements, the system keeps the validity of existing tasks that have been taught. In addition, each recognition technique and each robot can make full use of their own ability because they depend only on the task models, which do not depend on any recognition techniques or any robots.

These features produce great advantages in reducing the effort in teaching robots, which is the original purpose of LFO. First, the way of teaching in which an instructor can teach a robot just by demonstrating tasks can eliminate much effort from the instructor. However, if the system restricts robots that can execute tasks within a particular kind of robot, the system cannot provide a sufficient usefulness. It is especially important that the results of teaching can be shared between different kinds of robots. A system that consists of a monolithic framework unlike LFO is not flexible enough to do this. On the other hand, the framework of LFO is flexible enough. It can change a robot without losing availability of the tasks that have already been taught. This is also valid for changing a device and a method for task recognition. This makes the system more useful from the viewpoint of reducing human effort.

Fig.2.1 shows the essence of the LFO paradigm by an ideal situation in which different dancers perform the same dance. A performance of a dance is based on the intentions that determine the essential characteristics of the performance. The motion is the result of expressing the intentions by the body of the dancer. In different dancers who perform the same dance, the intentions of the dance are common although the bodies are different. Since the physical characteristics of a body are different from each other, the detailed motions of their performance necessarily vary depending on the characteristics of their own bodies. However, their performances can be regarded as the same dance as long as the common intentions of the dance are faithfully reflected in the motions. This concept is valid for humanoid robots that have various kinds of bodies.

This consideration guides what is important in order for learners to acquire the skill of a dance when they learn the dance from a master. If a learner adjusts the motion of the master to their own body without knowing the intention of the master, the adjustment can lose the intention. In this case, the learner cannot achieve the skill of the master. On the other hand, if a learner expresses the intention of the master although their motion is slightly different from that of the master, the learner

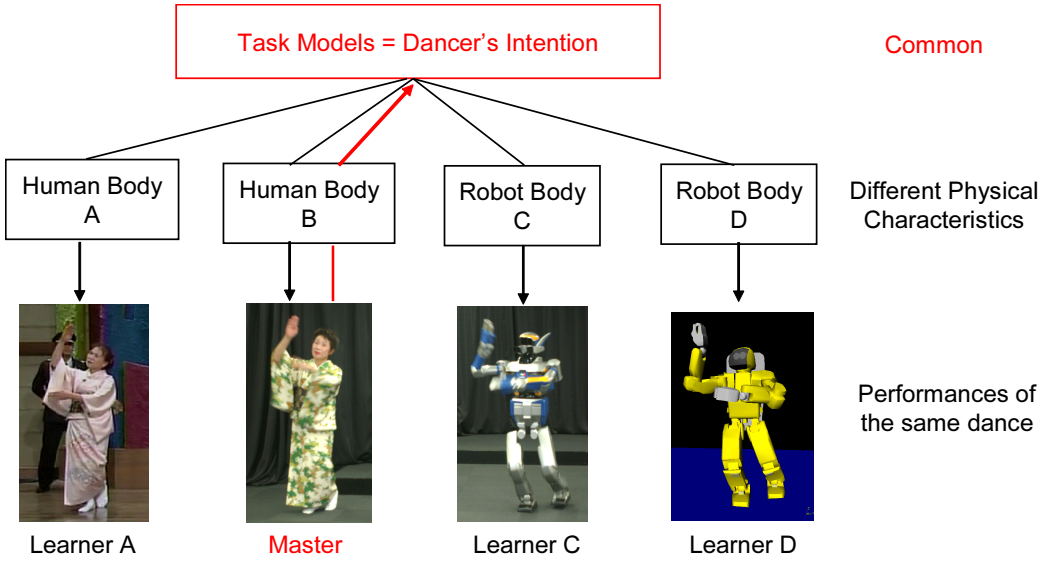


Figure 2.1: An ideal situation in which dancers learn and perform a dance. This situation is associated with the essence of the LFO paradigm.

successfully acquires the skill of the master. In fact, the learner cannot necessarily express all the intentions due to the body difference. In this case, it is important for the learner to express the intentions as much as possible. In summary, learning dances from the master means the following activities:

1. Acquiring the intentions of the master
2. Expressing the intentions with their own body

In the framework of LFO, the former corresponds to the recognition process, and the latter corresponds to the execution process. The intention corresponds to the representation of task models. In this way, the framework is based on the essence of the learning process. It enables a robot to express the skill of the master as much as possible under its own body.

It should be noted that Fig.2.1 shows an ideal situation. In fact, human dancers do not necessarily have the common intention in the strict sense. This is because a human learner cannot necessarily acquire the accurate intentions of the master, or a dancer may aim to have their own intentions. In such a situation, even if dancers perform the same dance, their performances can result in different representations in task models. Conversely, when human dancers have their own intentions, task models must represent the difference between them, and the difference should be reflected in performances by robots. On the other hand, if the different recognition results of dancers can be integrated into the common representation, the representation may

be regarded as the essence of the dance in itself. Task models are valid for the both purposes.

As described above, a framework based on task models brings great advantage to the problem of teaching robots. However, the advantage is not achieved unless task models are appropriately designed so that they represent the essence of tasks. Also, the recognition process and the execution process can be built only after task models are defined. Consequently, to define task models is the first important problem to solve. In the following sections, we discuss factors that are required for task models that deal with whole body motions of the human body.

2.2 Objectives of Task Models

Tasks are motions in which a defined procedure is performed so that the purpose of a task is achieved. Task models must provide representation of tasks so that robots can achieve the tasks according to the representation. In order to define task models, factors required for the representation must be clarified.

First, tasks generally include interactions between a performer and other things. A robot must manage the interactions performed in a task so that the robot achieves the procedure of the task. For example, the goal of an assembly operation is achieved if the interactions between the body and assembly parts are successfully performed so that each purpose of the interactions is achieved. If task models only provide an example of motions that perform the interactions, the robot cannot achieve purposes of the interactions because the essence of interactions is not necessarily their motions in themselves. Interactions generally depends on other factors that include the characteristics of the body and states of the objects. Interactions can also include uncertain factors caused by variable things. In order to achieve purposes of interactions, the robot must thereby know the essence of the interactions: what kind of interaction it must perform in the next place, and what is the purpose of the interaction. Task models must represent these factors without depending on a particular example or a particular body.

Next, there are tasks in which their motions themselves have meaning to the purpose of tasks. This thesis deals with this kind of tasks, dances. It is sure that this purpose is achieved if all the motion trajectories of the existing performance are completely achieved. However, this is impossible because of the body differences and constraints caused by interactions. This impossibility is a natural situation as described in the previous section. Hence the purpose of these tasks is not achieving the existing motion trajectories. The problem is defining the important characteristics to preserve in motion. If the important characteristics are represented by task models, the robot

can keep the important characteristics by its own body as much as possible. If the robot express the important characteristics, the purpose of tasks is achieved.

In summary, task models must represent the following factors in order to achieve purposes of tasks.

1. Essence of the interactions that are performed in the tasks
2. Factors that determine important characteristics of the motions

We call the former factors *interactive factors* and the latter factors *characteristic factors*.

Task models are defined so that they provide the both kinds of factors. In order to define task models that provide required factors, a category of motions must be determined first. The definition of task models depends on categories of motions because kinds of interactions and important characteristics in motions vary according to the categories. Categories of motions that are classified by this point of view are called *task domains*. In the process of the definition, first a task domain that the models deal with must be determined. Then the factors required for the motions in the task domain are clarified, and a set of task models based on these factors is defined. This approach of the LFO restricts the target tasks within a particular task domain. However, a task domain can include various possible tasks, and the framework enables various robots to robustly learn and reproduce those various tasks in the task domain. This satisfies the purpose of the LFO paradigm.

In this thesis, whole body motions of the human body are the target motions. We propose task models that deal with interactive factors of whole body motions in order to prove that the LFO can be achieved for whole body motions. However, possible whole body motions include numerous kinds of motions from the viewpoint of interactive factors. We first classify motions based on kinds of interactions they include, and achieve the paradigm for a particular task domain in the classification.

At the same time, our task models also deal with characteristic factors of whole body motions in order to prove the validity of these factors for whole body motions. For this purpose, the dance is a good target because the purpose of most dances is to express characteristic motions. For the dance, there are also many categories of dances that are based on different interactive factors and different characteristic factors. In this thesis, we achieve the paradigm for a particular category of dances that are based on the same characteristic factors.

In the following, first we classify the generality of whole body motions by focusing on interactive factors. We define the target task domain in terms of this classification. After that, we determine the more concrete task domain that deals with dances from the viewpoint of characteristic factors.

2.3 Motion Classes based on Interactions with the Floor

In order to consider the factors required for task models, kinds of interactions that are performed in a task domain must be clarified. In this section, we classify the generality of whole body motions of the human body from this viewpoint.

We focus on physical interactions between the body and other objects. Physical interactions in whole body motions are roughly classified into the followings:

1. interactions with the floor
2. interactions with objects other than the floor

The subject of the former interactions is force: the force that the body exerts on the floor and the reaction force that the floor exerts on the body. Motions that include the latter interactions are not the target of this thesis.

For the motions that only include the interaction between the body and the floor, their degree of generality can be classified by two evaluation factors:

- Body parts that contact the floor
- The way of contact of the body parts

Fig.2.2 shows these two axes to indicate the generality of whole body motions. This figure also shows the examples of motions that are classified into each position. Since soles are basic body parts in interaction with the floor, the axis of body parts especially focuses attention on soles. In the horizontal axis, the level of generality increases as the number of body parts that contact the floor increases. The vertical axis represents the level of generality with regard to dynamic factors in the contact. It should be noted that the vertical axis does not necessarily represent the strict order of the generality; it represents a rough order so that the generality can be classified by a simple form. Each position in the graph represents one motion class.

Basically to perform motions classified into a higher level is more difficult because increase in the generality requires additional factors to consider in performing the motions. Higher level motions also require higher physical capabilities. In our framework, additional factors are required for task models as the generality increases. That is to say, this generality determines a task domain with regard to physical interactions. In the following sections, we describe concrete factors required for performing motions according to this generality.

2.3.1 Motions with No Interaction

Motions that do not include any interaction with the floor are classified into the lowest level of the generality indicated by Fig.2.2-(a). This class of motions cannot

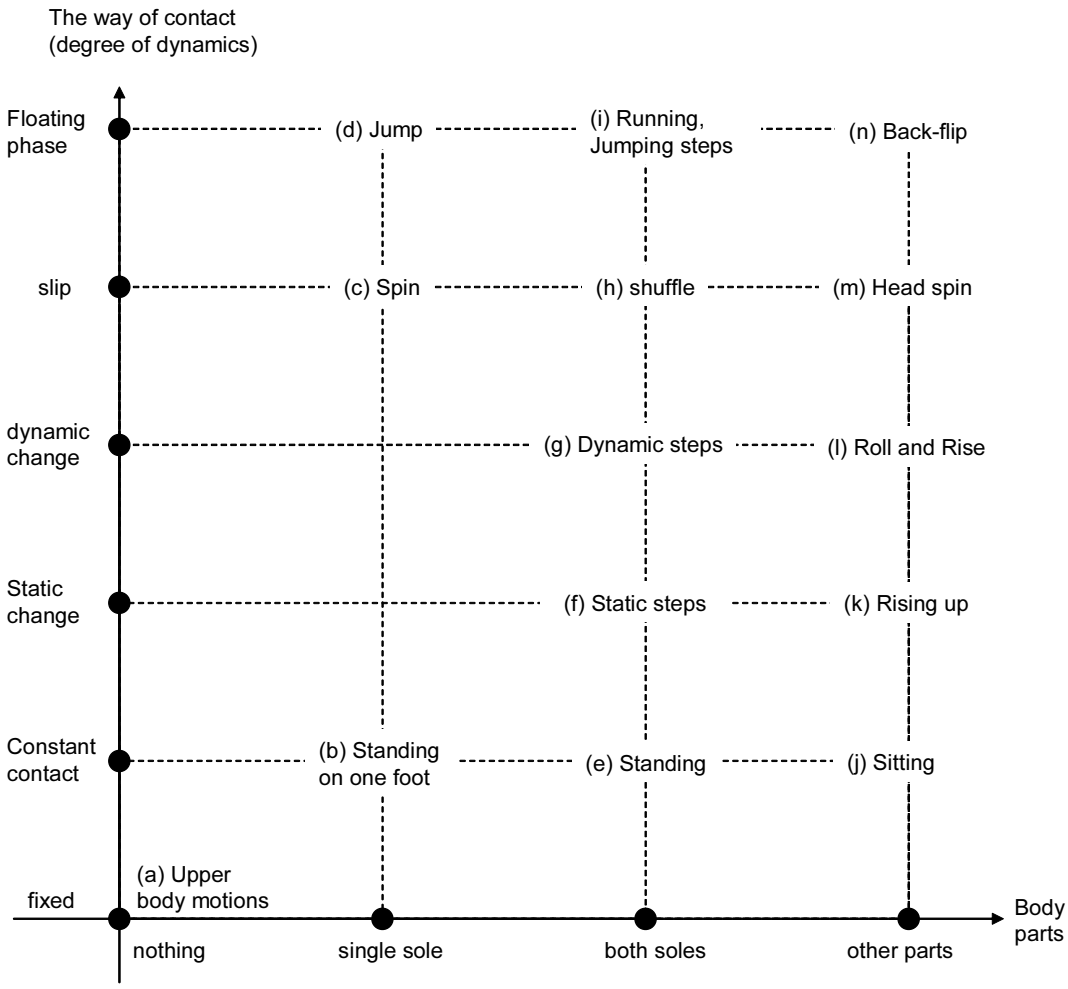


Figure 2.2: Generality of whole body motions that only include the interaction between the body and the floor. Such motions are classified by two axes of factors: body parts that contact the floor, and the way of contact of the body parts. Each position in the graph represents one motion class. Labels in the graph show example motions of each motion class.

be regarded as whole body motions in the strict sense; only a part of the whole body can perform this class of motions. Otherwise, the motions are performed in air in a situation such that the body is firmly fixed at some body part to a stable object.

Although this class does not include any factors to consider for interactions with the floor, the class includes a more fundamental factor to consider. This factor is a body in itself. This means that possible poses and possible motions of a particular body depend on its own characteristics. If an intended motion is possible for the body, the motion can be easily performed in this motion class. However, if an intended motion is impossible for the body, the motion must be adapted to the body so that it can be performed by the body. Although the adaptation is a role of the generation process in the framework, task models can also control the adaptation by indicating important points to preserve in motion. Since the important points are not a factor that depends on a particular body, clarifying the important points in motion is a role of task models in this motion class.

This factor caused by a body in itself is fundamental to all the motion classes. On the basis of this factor, higher motion classes require additional factors caused by interactions.

2.3.2 Constant Contact of a Single Sole

Keeping constant contact between the body and the floor can be regarded as the most simple interaction with the floor. Motions that keep the constant contact between a single sole and the floor are classified into the generality indicated by Fig.2.2-(b).

In this class, motions must be those that can maintain the whole body in balance. Since a sole is not firmly fixed on the floor, the body cannot keep the contact if it loses its balance. In this case, intended motions cannot be achieved and the body will fall down. Dynamic balance depends not only on motions but also physical properties of the body; especially the weight distribution of the body affects dynamic balance. Even if intended motions can be performed under the condition of motion class (a), those motions can fail in keeping the contact in this motion class. A robot must perform motions so that it can maintain dynamic balance on its own body properties. However, the robot cannot achieve this without knowing that it must keep the contact between its sole and the floor.

Hence a necessary factor of task models in this class is clarifying the information on this interaction: the intention to keep the contact, and which sole the target of this contact is. When these factors are clarified by task models, maintaining balance using a particular body is a role of the generation process.

2.3.3 Variable Contact of Both Soles

In motion class (b), motions are performed at the same position on the floor, in which whole body motions are obviously constrained. Allowing the change of contact by using both soles significantly extends possible whole body motions. Those motions are classified into motion class (f) or higher classes in Fig.2.2.

One of the important factors in these classes is stability of the contact. A sole must keep stable contact while it supports the body. When the body is supported by both soles, the correlation between two soles must be adjusted so that both of them can keep a stable contact. The change of the contact also affects stability. A sole must absorb the impact force caused by the landing of the sole so that the change is achieved as intended. For this purpose, forms and motions of soles when they contact the floor are common important factors in these motion classes.

A robot must consider these factors in order to perform a variable contact stably. It is necessary for a robot to know the change of the contact before it performs the contact. This requires task models to provide information on changes of the contact: which sole is the target of the next change, and what is changed in the state of the contact. These are fundamental factors of task models that deal with motion class (f) or higher motion classes.

The motions in these classes must also consider the dynamic balance between the body and the floor. In motion class (b), maintaining balance means keeping constant contact of one sole at all times. This is not sufficient for motion class (f) or higher motion classes because the change of contact requires additional factors for balance.

In motion class (f), the change of the contact is restricted in a static way. This means that the body moves so that it can always keep its balance whenever it stops the motion. This kind of motions can be achieved by satisfying the following conditions: (1) The center of mass (CM) is always above the *support area*, which is a convex hull consisting of contacting soles, and (2) Motions are performed slowly. Although this kind of static motions are restricted motions, they can be easily controlled by satisfying these conditions. If the fact that intended motions are static motions is indicated by task models, a robot can perform the motions by satisfying the above conditions on its own body.

The generality increases in motion class (g). This class allows a dynamic way of contact change, meaning that motions need not satisfy the conditions of the static way. The CM is not above the support area, and the speed of motions is not restricted. At the same time, the conditions for static motions are not valid for this motion class. This class requires more general conditions of dynamic balance. The details of the conditions are described in Chapter 5. Here, the important point is that the conditions change as the motion class changes.

Since a stable way of contact and dynamic body balance depend on physical characteristics of a body, the process that considers these factors is the execution process in our framework. The role of task models is to provide the information required for this purpose. The execution process can manage these factors only after the required information is clearly provided by the task models.

2.3.4 More Dynamic Way of Interaction

In class (c) or higher classes, motions include slips between a sole and the floor. In these classes, slips are performed as intentional motions such as a spin turn. On the other hand, unintentional slips must be prevented. Preventing unintentional slips is also valid for all the motions in the lower motion classes. For this purpose, task models for these motion classes must clarify whether a motion intends to make a slip or not. If the motion intends to slip, what kind of slip the motion intends must also be clarified. These are necessary factors of task models that deal with these classes.

In motion class (d) or higher classes, motions include a period in which the whole body floats in air. These motions include jumping and running. In the floating phase, the floor does not exert a reaction force on the body in contrast to the phase in which some body parts contact the floor. This means that controlling a body in the floating phase must consider conditions different from the contacting phase. Hence it is necessary for a robot to know about a change into the floating phase in advance. This requires task models to clearly represent the change between the floating phase and the contacting phase.

It should be noted that floating motions must generally absorb a big impact between a sole and the floor when the sole lands after the floating phase. This must be considered by the execution process.

2.3.5 Interactions Performed by Additional Body Parts

In motion class (j) or higher classes, motions include interactions in which additional body parts other than soles contact the floor. In the classification by Fig.2.2, those motions are roughly classified into the classes on the same horizontal position for the purpose of simplicity. In fact, motions are classified into more specific classes according to body parts that essentially perform the interaction with the floor. For example, motion class (k) includes a back-flip. A back-flip is classified into the motion class in which soles and hands contacts the floor in interactions. Additional factors for this motion class are interactions by hands. In this way, target body parts that interact with the floor must be clarified, and task models must provide information that is required for performing target interactions.

2.3.6 Related Studies based on the Motion Classes

Fig.2.3 shows related studies based on the motion classes. We compare this thesis with major studies that deal with each motion class.

First, we focus on the studies that are associated with the problem of adapting human motions into robot motions. In motion class (a), the adaptation on actual robots was achieved by Pollard *et al.* [65]. In motion class (b), the adaptation on actual robots was achieved by Tamiya *et al.* [78]. Yamane *et al.* [84] have proposed a method that deals with the highest, unlimited motion class, but their method has not achieved adaptation to actual robots. Although their method achieves the local-level consistency of dynamics completely, the global-level consistency of dynamics is not necessarily satisfied, and extra factors that affect stability of motions are not well considered. These are the major studies that deal with the problem of the adaptation. Consequently, the highest motion class in which the existing studies achieved the adaptation of human motions to robot motions has remained class (b). In contrast to a framework based on LFO, these methods basically consist of a process that directly modifies the original motion trajectories into motion trajectories of a robot.

In motion class (g), many methods that enable a biped robot to perform a dynamic walk have been proposed. However, these methods only deal with the global trajectory of the body. Kuroki *et al.* [37] have developed a framework that enables a human operator to edit detailed motions of the entire body parts manually.

In the higher motion classes, motions by actual robots have been achieved by many studies. In (d), Raibert [67] developed a single-leg robot that hops. In (h), Kaneko *et al.* [25] have proposed a method for detecting and stopping slips of soles. In (j), running by biped robots has been achieved by Kajita *et al.* [20] and Nagasaka *et al.* [49]. In (k), Kanehiro *et al.* [22] have achieved a rising motion from a lying position based on the transitions between statically stable poses. In (l), Kuniyoshi *et al.* [35, 79] have achieved a more dynamic rising motion called a roll-and-rise. However, these studies achieved only a particular motion. They cannot provide the sufficient generality of producing various patterns of motions of each motion class. In these motion classes, the limitation of the current hardware also makes problems difficult.

2.4 Target Task Domain

In this section, we define the concrete task domain that this thesis deals with. As described in Section 2.2, task models must represent the following interactive factors and characteristic factors to achieve a purpose of a task.

1. Essence of the interactions that are performed in the task

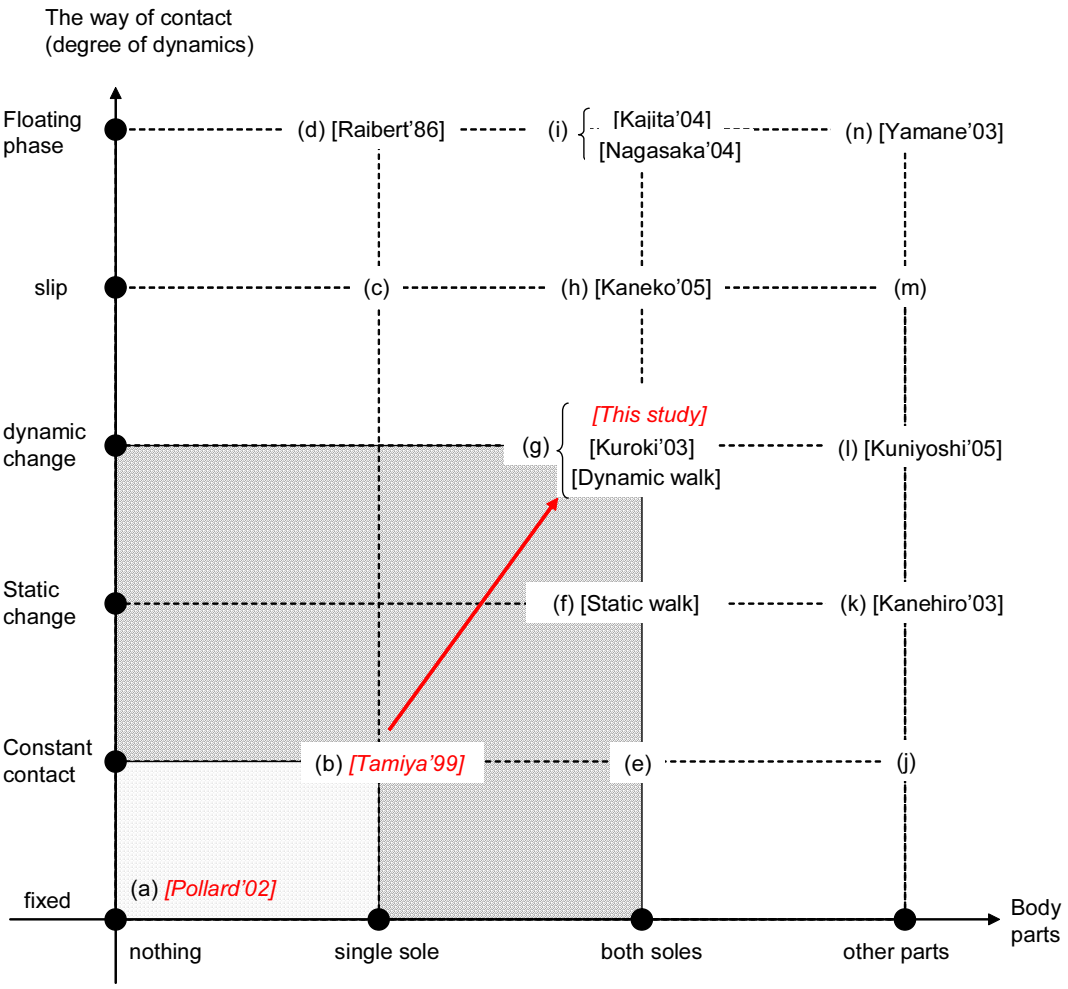


Figure 2.3: Related studies in terms of the generality of whole body motions. Each label in the graph shows major studies that deal with the corresponding motion class. The red italic labels show studies that achieved reproducing human motions in actual robots.

2. Factors that determine important characteristics of the motions

In the previous section, we showed that required factors of task models vary according to the motion classes based on the interaction between the body and the floor. We define the target domain in this classification. We also define the target domain of dances by focusing on their characteristic factors. Through these definitions, we clarify the concrete factors required for the task models for the target task domain.

2.4.1 Target Performers

One of the important advantages of our framework is that it does not depend on a particular body that has its own characteristics. However, if a body does not satisfy the prerequisite condition of the task domain, it is fundamentally impossible for the body to perform tasks according to task models. To cite an extreme case, a mobile robot that can only actuate its wheels cannot perform assembly tasks. Hence the prerequisite conditions of the body must be clarified for the task domain.

Since the target task of this thesis is whole body motions of the human body in a broad sense, we define the prerequisite conditions of the robots that perform tasks as follows:

1. The robot is designed so that its whole body can express human-like motions
2. The robot has legs that can support and move the body on the floor

This kind of robot is called a "biped humanoid robot." Although this definition involves ambiguous factors, this is natural for our purpose. Considering a more concrete definition is a problem of the generation process.

2.4.2 Target Motion Class

In motion classes described in Section 2.3, our target class is (g) in Fig.2.2. The conditions of this motion class are summarized as follows:

1. Motions are performed on the floor.
2. The only object with which the body interacts is the floor.
3. The only body parts that contact the floor are the soles.
4. A sole does not slip on the floor when it contacts the floor.
5. Motions do not include a period in which both soles float in the air at the same time.

For this motion class, it is important for task models to clearly represent the essence of the contact change between soles and the floor. This enables a robot to perform the contact change stably, while maintaining dynamic body balance.

We propose a framework that enables robots to learn and reproduce whole body motions in this motion class from human demonstrations. From the viewpoint of reproducing human motions on actual robots, the existing study that achieved the highest level of the motion class has remained class (b) in Fig.2.2 as described in Section 2.3.6. One of the remarkable achievements of this thesis is extending the level of the motion class from this viewpoint. Extending the level from (b) to (g) significantly enhances the usefulness of the system.

2.4.3 Leg Task Models

In the target motion class (g), all the body parts that contact the floor are only the soles. In other words, no part of the upper body contacts the floor. In this situation, if the assumption is made that the upper body can perform motions independent of the lower body, the upper body motions can be decoupled from whole body motions. In this case, the upper body can employ its own task models according to its purposes. In fact, the upper body motions can be regarded as motion class (a) if the decoupling is achieved. Since this class is significantly different from motion class (g), it is reasonable to employ different task models for the upper body as compared with the lower body.

On the basis of this consideration, our objective is to achieve the framework that satisfies the following conditions:

- The upper body can represent motions that are independent of the lower body motions.
- The lower body compensates for the effect of upper body motions on the interaction with the floor so that motion class (g) can be performed.

This thesis proposes *leg task models*, which deal with the lower body motions so that the above conditions are satisfied. In this thesis, the design of task models concentrates on the leg task models. Although we achieve the framework that allows independent motions of the upper body, this thesis does not deal with task models for upper body motions.

2.4.4 Target Dances

The task domain must also clarify characteristic factors of the target tasks. This thesis sets dances on the target task from this point of view. However, there are many

categories of dances that are based on different characteristic factors. We specify the target category of dances in order to define the task domain.

First, dances are basically classified into the following two types:

- Dances in which a dancer intends to perform the same choreography patterns to the same music
- Dances in which a dancer intends to vary the choreography according to situations

This thesis deals with the former type of dances. We call these dances *choreography-oriented dances*. For this type of dances, to reproduce one demonstration of a human dancer is meaningful for robots from the viewpoint of the original intention. On the other hand, in the latter type of dances, a dancer goes as far as to change global motion patterns every time the dancer performs. For these dances, to reproduce one demonstration is not significantly meaningful because it cannot achieve the original intention. Also, this type of dance performance requires interactive factors other than interaction with the floor. The motions that include those interactions exceed the domain of the target motion class. Therefore this thesis does not deal with the latter type of dances.

Now the target category of dances is choreography-oriented dances that are performed on the conditions of motion class (g). Since our goal is to define the leg task models, we focus on leg motions of this category.

As described in Section 2.3.3, task models for motion class (g) must represent the change of contact between soles and the floor. This indicates that the change of the contact state is essential in this motion class. In fact, only the change of the contact state can changes the global position of the whole body. The change of the contact state also plays an important role in changing a pose of legs in a period in which both soles contact the floor. In this way, the change of the contact state is a factor that causes considerable change in motions. This is hence a general, important factor for expressing characteristics of leg motions. Since choreography-oriented dances intend to express a given choreographic pattern according to a tempo of music, it is important to preserve the pattern and timings of changes in the contact state. These factors are called *the contact pattern factors*.

In addition to the contact pattern factors, dances generally require more specific factors for expressing their important characteristics. Those factors vary depending on more specific dance categories. In this thesis, we focus on the dance category that is based on *stopping key poses*.

In this category, a basic unit of choreography is represented as a transition to a particular key pose. In the dances of this category, the key poses that form basic units of choreography are important to the dance expression. Yukawa *et al.* [89] have

developed a *Buyo-fu* system that enables a user to view and edit a structured form of dance motions based on those basic units. A theater company Warabi-za [1] has used this system. They created a database of various Japanese folk dances using this system. In this data base, they manually structuralized motion data according to the opinion of specialists on the dances. They found that the representation by basic units based on a transition to a particular key pose is valid for various Japanese folk dances. On the basis of this representation, Nakazawa *et al.* [52] have proposed a method that automatically extracts a structured form of dance motions. Their attempt revealed the fact that dancers momentarily stop their motions at key poses. Shiratori *et al.* [71] have improved this method to increase the accuracy of the extraction. Their attempt revealed the fact that a dancer makes key poses at a beat of dance music. From the results of these studies, the following facts are concluded:

- There is the category of dances in which a motion is represented as a transition to key poses that are important to expression.
- Dancers momentarily stop their motions at key poses.
- Timings of key poses are important to expression.
- This category includes various Japanese folk dances.

Stopping key poses means key poses in this dance category.

For the lower body, its pose is determined by the position and attitude of the feet and the position of the waist. Since the positions and attitude of a foot when its sole contacts the floor are represented by the contact pattern factors, expressing stopping key poses requires the following additional factors:

- To represent the positions and attitude of a foot at a particular time when it floats in air
- To represent the positions of the waist at a particular time

2.5 Components of the Framework

In this chapter, we discussed the issue of how to design task models. Since this is the central issue of the LFO paradigm, the discussion also clarified the essence of LFO. At the same time, we defined the target task of this thesis in detail, and made the concrete problem clear.

On the basis of the consideration of this chapter, we propose a framework based on LFO that deals with the target task domain. Fig.2.4 shows the components of the framework. The concrete components we propose in this thesis are as follows.

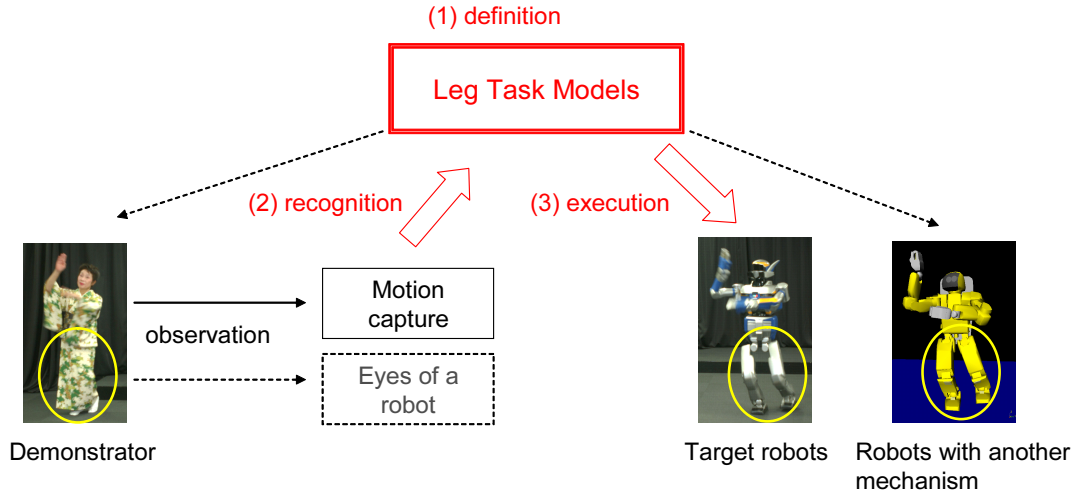


Figure 2.4: Components we propose in this thesis: (1) We define leg task models, which represent lower body motions that are performed according to the condition described in Section 2.4.2. (2) We propose a recognition method that uses a motion capture as an observation device. (3) We propose a generation method that assumes the existing biped humanoid robots with standard mechanisms.

First, we define leg task models, which represent lower body motions that are performed on the condition described in Section 2.4.2. The concrete definition of the leg task models is described in Chapter 3. The framework is constructed on the basis of the leg task models.

As described in Section 2.1, the framework can employ various methods and devices in the recognition process and the execution process. Fig.2.4 includes those possible components. Since the main purpose of this study is to prove the validity of the framework based on LFO, our goal of these processes is to enable actual robots to reproduce actual human dances.

For the recognition process, we employ a motion capture as an observation device. We propose a method for recognizing elements of the leg task models from motion data obtained by a motion capture. This method is described in Chapter 4.

For the execution process, we propose a method that assumes the existing biped humanoid robots with the standard mechanisms. We must solve many problems in order to enable the robots to perform tasks stably. Concrete problems and details of the method are described in Chapter 5. It should be noted that we do not develop or improve the hardware of a robot because such an approach does not serve our purpose.

As for the upper body, the current framework does not deal with its motions in the form of the LFO paradigm. It is desirable that the framework provides another set

of task models for upper body motions so that the advantages of LFO are available for upper body motions. In Appendix D, we show a simple implementation that converts human motions into motions of a robot.

Chapter3

Definition of Leg Task Models

3.1 Overview

Leg task models are motion models that represent basic leg motions and their essential characteristics. The models consists of *tasks* and *skills*; task means *what to do*, and skill means *how to do a task*.

Classes of basic motions are modeled as tasks, and they are symbolized. The characteristics of the tasks are modeled as skills so that each task can support a variety of expressions. Skills are described by *skill parameters* defined for each task. In other words, an instance of a task is determined by the values of the skill parameters.

Leg task models are the fundamental element of our framework for reproducing human motions. Recognizing human motions and generating motions for a robot are based on leg task models. The target motions of this thesis are those described in Section 2.4.

3.2 Defined Tasks

Fig.3.1 shows the defined tasks. These four classes of basic motions are modeled as tasks. They are symbolized as R-STEP, L-STEP, STAND, and SQUAT.

The *R-STEP* represents one stepping motion by a right foot, and the *L-STEP* represents that by a left foot. To be more accurate, they represent a motion in which one foot detaches from the floor once and lands again while the other foot is on the floor. The former foot is called a *swing foot* and the latter foot is called a *support foot*. We use a symbol *STEP* for representing both R-STEP and L-STEP. Using STEP tasks, various leg motions including footfalls, side- or back-stepping, and kicks can be expressed.

The *STAND* represents a motion in which the upper body is supported by both feet.

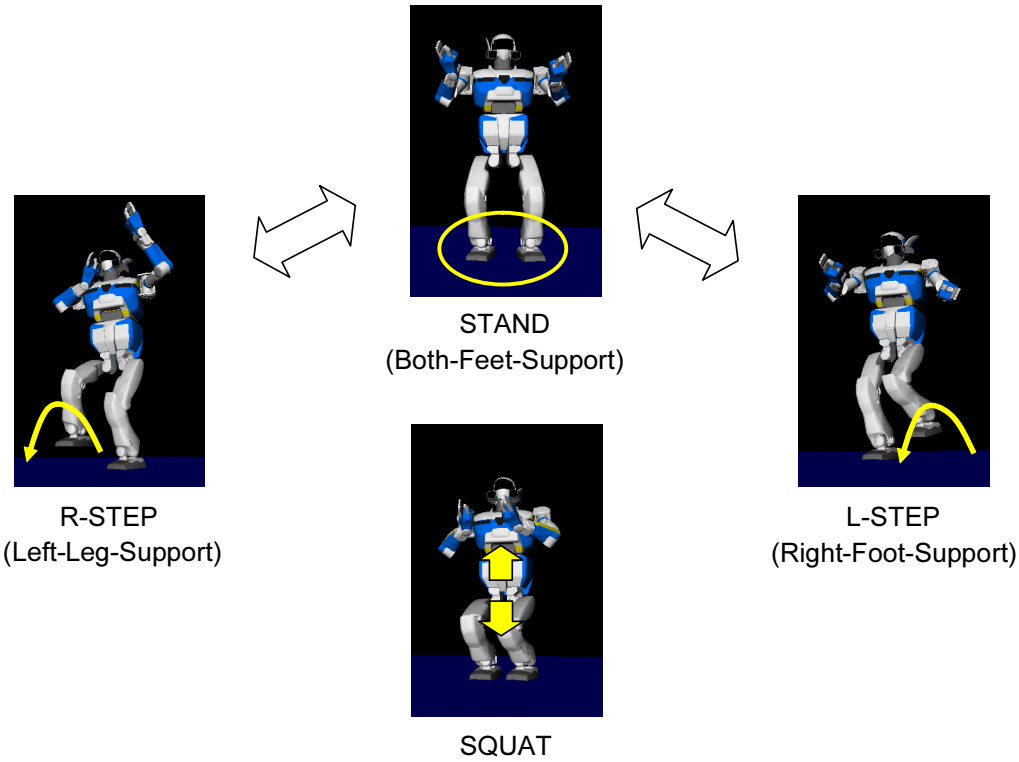


Figure 3.1: The defined tasks. These four classes of basic motions are modeled and symbolized as tasks. Arrowheads indicate transition relationship between tasks. Labels in parentheses show a support state.

Table 3.1: Skill parameters. Σ_{su} is the local coordinate from the support foot.

Common	t_0	Beginning time
	t_f	Finishing time
STAND		-
STEP	Parameters of the swing foot	
	$\boldsymbol{r}_f$	Horizontal position on Σ_{su} at t_f
	ψ_f	Angle around the z-axis (yaw-axis) of Σ_{su} at t_f
	Parameters of the waist	
	ψ_w	Yaw orientation on Σ_{su} at t_f
	Parameters of the mid-state (option)	
	t_1	Time of the mid-state
	$\boldsymbol{r}_1, \boldsymbol{R}_1$	Position and attitude of the swing foot on Σ_{su} at t_1
SQUAT	t_1	Time of the mid-state
	d_1	Waist height distance

The R-STEP, L-STEP, and STAND are the tasks that focus on the *support states*. The states indicate which foot supports the body: only the left foot supports the body, only the right foot, or both feet support the body. These are called *Left-Foot-Support*, *Right-Foot-Support*, and *Both-Feet-Support*, respectively. As shown in Fig.3.1, the support state changes according to the transition of these three tasks, and the transition takes place through the medium of a STAND task.

The *SQUAT* represents a motion in which the waist lowers once and rises again. In other words, the SQUAT is a task that changes the vertical position of the waist.

3.3 Skill Parameters

Each task has its own set of skill parameters that makes the instance of a task. A set of skill parameters consists of timings and positions of body parts at the timings. Those timings and positions represent the essential states in a task. Table 3.1 shows sets of skill parameters defined for each task.

Common Parameters

All the tasks have the beginning time t_0 and the finishing time t_f as skill parameters. These values determine the running period of a task. The STAND task has only the common parameters, and the STEP and SQUAT tasks have other parameters specific to each task.

As for operations such as assembly, operation processes are not necessary to determine the running period of each process in most cases. On the other hand, dance performances are necessary to determine the accurate timings of each task so that performances can keep the rhythm of dances. Thus these timing parameters are important to leg task models for dance reproduction.

Parameters specific to the STEP task

In the STEP task, skill parameters are defined for describing positions and attitudes of a swing foot, which determines the characteristics of a stepping motion.

The positions and attitudes are described as values based on Σ_{su}, which is the local coordinate from the support foot. In this thesis, the positive direction of the z-axis represents the upper direction. The z-axis on Σ_{su} is set at the same direction as that on the world coordinate. It is assumed that the support foot remains still on the floor during a STEP task, so that Σ_{su} is constant during the task.

Parameter $\boldsymbol{r}_f$ is a two-dimensional vector that represents the horizontal position of the swing foot when it lands on the floor at the timing of t_f. The parameter $\boldsymbol{r}_f$ does not include an element of the z-axis because it is forced to be the height of the floor at the landing. Parameter ψ_f is the angle that represents the attitude of the swing foot around the z-axis (yaw-axis) at t_f. These parameters determine the finishing position of the swing foot. According to these parameters, the motion of the swing foot is basically represented as a trajectory from the beginning position at t_0 to the finishing position at t_f. The STEP does not have parameters on the beginning positions. The beginning positions are not determined by the running task but determined by the result of the previous tasks.

A swing foot could perform not only a simple trajectory but a complex one such as raising the foot to a significant height. In such a case, the mid-state is enabled; the time of the mid-state is described by parameter t_1, and the position and attitude of the swing foot at t_1 are described by parameters $\boldsymbol{r}_1$ and rotation matrix $\boldsymbol{R}_1$, respectively. In contrast to $\boldsymbol{r}_f$ and θ_f, parameter $\boldsymbol{r}_1$ is a three-dimensional vector that includes the z-axis, and the attitude is described by the rotation matrix. This is because the swing foot at the mid-state is in the air. If the mid-state is enabled, the motion of the swing foot is represented as a trajectory that passes $\boldsymbol{r}_1$ and $\boldsymbol{R}_1$ at t_1. Whether the mid-state is required or not is determined by the recognition process, which is described in Section 4.2.

Parameter ψ_w is a yaw angle of the waist at t_f. This parameter represents the orientation of the whole body around the vertical axis. This parameter is required for expressing turning motions with a step.

It should be noted that a state of a foot is not necessarily described by a single

position and attitude because the human foot is not a rigid object and some robots have joints in their feet. In leg task models, parameters on the feet are just a representative value of a foot part.

Fig.3.2 shows the correlation between the skill parameters defined for the STEP task.

Parameters specific to the SQUAT task

The SQUAT task has the mid-state at which the waist reaches the lowest position in a motion. Parameter t_1 is the time at the mid-state, and parameter d_1 is the vertical distance of the waist from the original position and the position at the mid-state. Fig.3.3 shows the parameters of the SQUAT tasks.

3.4 Task Sequence

Tasks have their running period from t_0 to t_f so that they can be put on a time axis. A continuous leg motion of a whole performance is represented by a *task sequence*, in which a number of tasks are put on the time axis. Fig.3.4 shows an example of task sequence.

In a task sequence, the same tasks do not overlap. The R-STEP, L-STEP and STAND tasks also do not overlap. One of these tasks is placed on the time axis at the same time according to the transition diagram shown in Fig.3.1. On the other hand, the SQUAT tasks are independent of this transition, so that a SQUAT can overlap with other tasks.

If the R-STEP and L-STEP can overlap, a task sequence can represent motions in which both feet are in the air at the same time. That is to say, motions such as running and jumping can be represented. However, this thesis does not deal with such motions as stated in Section 3.1.

In a task sequence, a task can be appended, modified, and removed without changing other tasks in the sequence as long as the above conditions are satisfied. This is because tasks do not have parameters that determine their beginning state, and all the parameters on positions and attitudes are described as relative values from a base body part at the beginning state. This enables a task to adapt itself to the change of tasks before it. This feature is necessary for the process of skill refinement, which is described in Section 5.7.

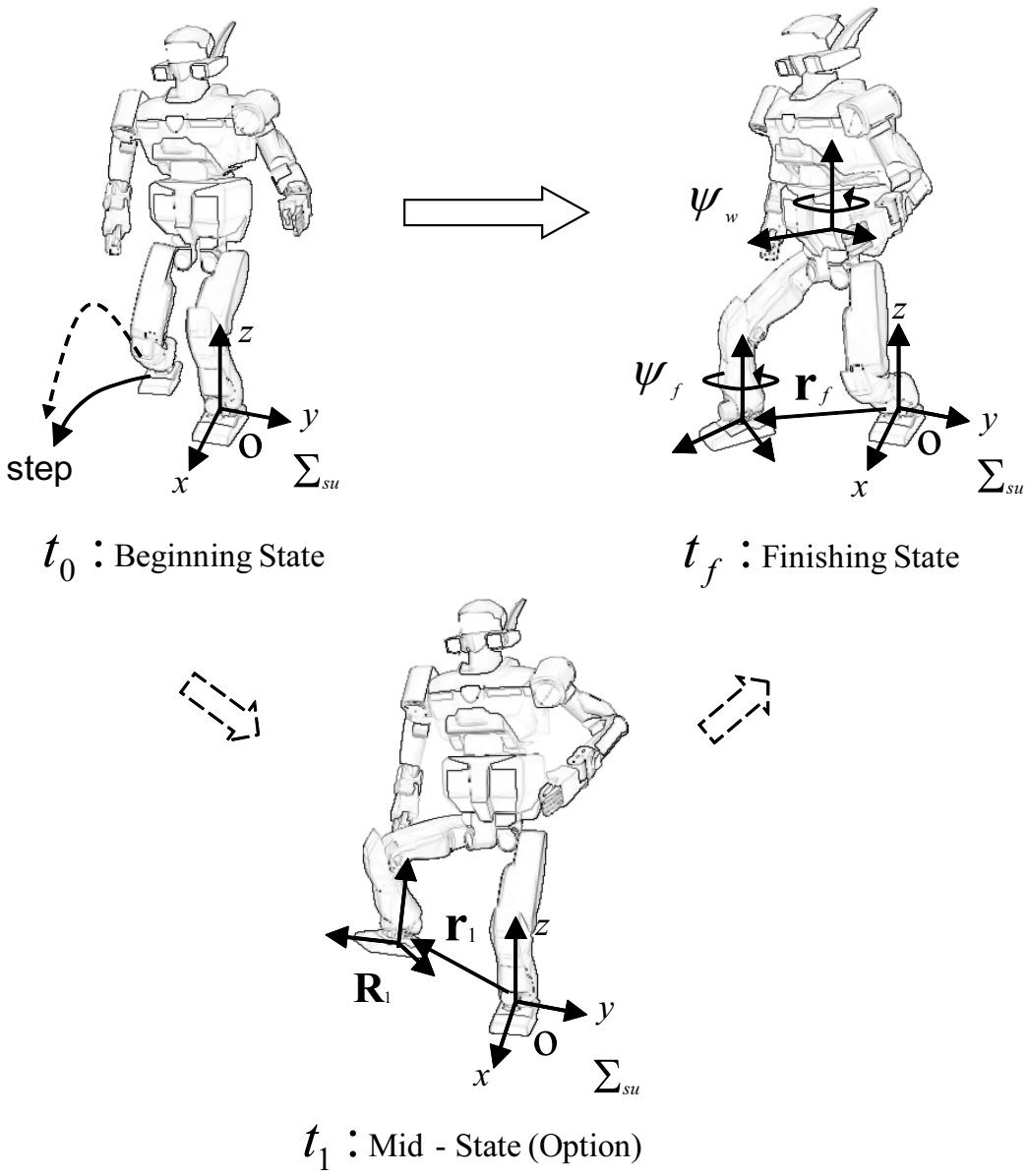


Figure 3.2: Skill parameters defined for the STEP task.

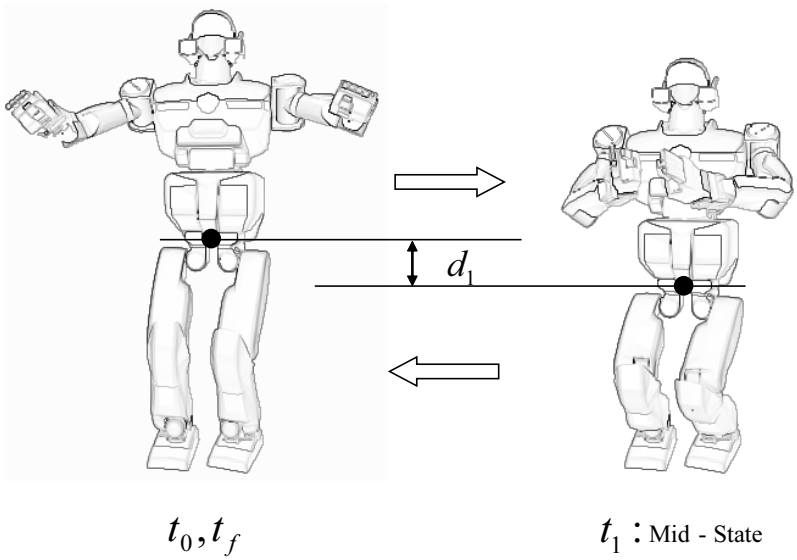


Figure 3.3: Skill parameters defined for the SQUAT task.

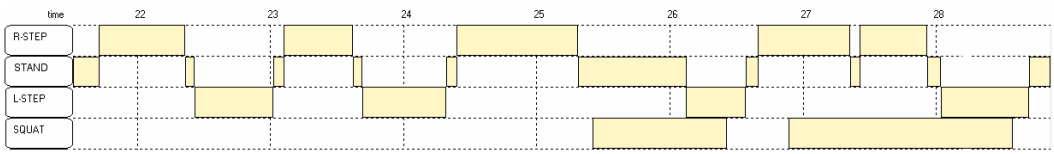


Figure 3.4: An example of a task sequence. Tasks are put on the time axis.

Chapter4

Recognition of Leg Tasks

4.1 Overview

In this chapter, we describe the method for recognizing leg tasks from human motions.

A sequence of leg tasks is recognized from marker trajectories obtained by a motion capture. For each task class, first, temporal segments corresponding to a task are detected. The time values of skill parameters are also obtained in the same process. Then, for each detected task, the position values of skill parameters are extracted from a correlation between several markers related to that task. Fig.4.1 shows the overview of the recognition process.

The recognition method assumes the configuration of body markers shown in Fig.4.2. A marker is attached to the toe, heel, and knee of each leg, and four markers are attached to the edges of the waist. For each marker, the trajectory of the 3-D position is obtained, and these trajectories are passed to the method. In the following description, the trajectories of these markers are expressed with their labels. For example, the position of the right toe at time t is expressed as $\boldsymbol{p}_{rtoe}(t)$, which represents a three-dimension vector. As shown in Fig.4.2, the positions are described by the global coordinate, in which the origin is on the floor and the z-axis corresponds to the upper direction.

The recognition method refer to *recognition parameters* shown in Table 4.1. Most parameters are threshold values for detection. Details of these parameters are described in the following section.

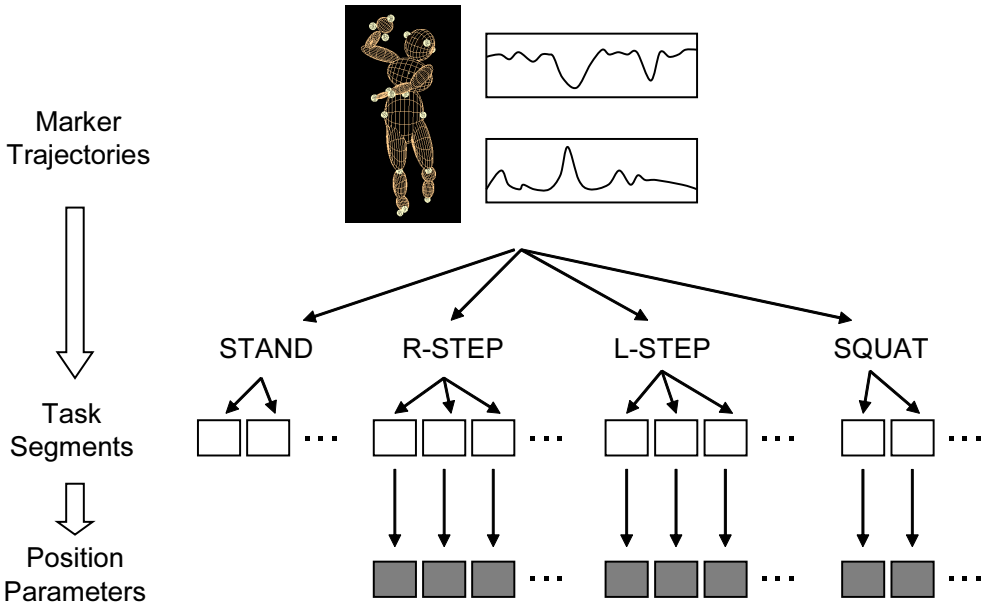


Figure 4.1: Overview of the recognition process. For each task class, temporal segments are detected first. Then the position values of skill parameters are extracted for each task segment.

4.2 Detecting Task Segments

4.2.1 Detecting STEP tasks

STEP tasks are detected by focusing on the speed of a foot motion. For the R-STEP, the speed of the right foot

$$v_{rtoe}(t) = |\,\dot{\boldsymbol{p}}_{rtoe}(t)\,| \tag{4.1}$$

is analyzed.

Fig.4.3 shows an example trajectory of the foot. Graph (a) shows z-axis of $\boldsymbol{p}_{rtoe}(t)$, and graph (b) shows $v_{rote}(t)$. Since our target motions do not include slipping of soles

Table 4.1: Recognition parameters

v_{step}	Threshold of the velocity in the STEP detection
l_{step}	Threshold of the moving distance in the STEP detection
d_{step}	Threshold that determines enabling the mid-state of a STEP task
t_{iv}	Interval between STEP tasks
l_{squat}	Threshold of the moving distance in SQUAT detection
c_f	Ratio that determines the center of a foot

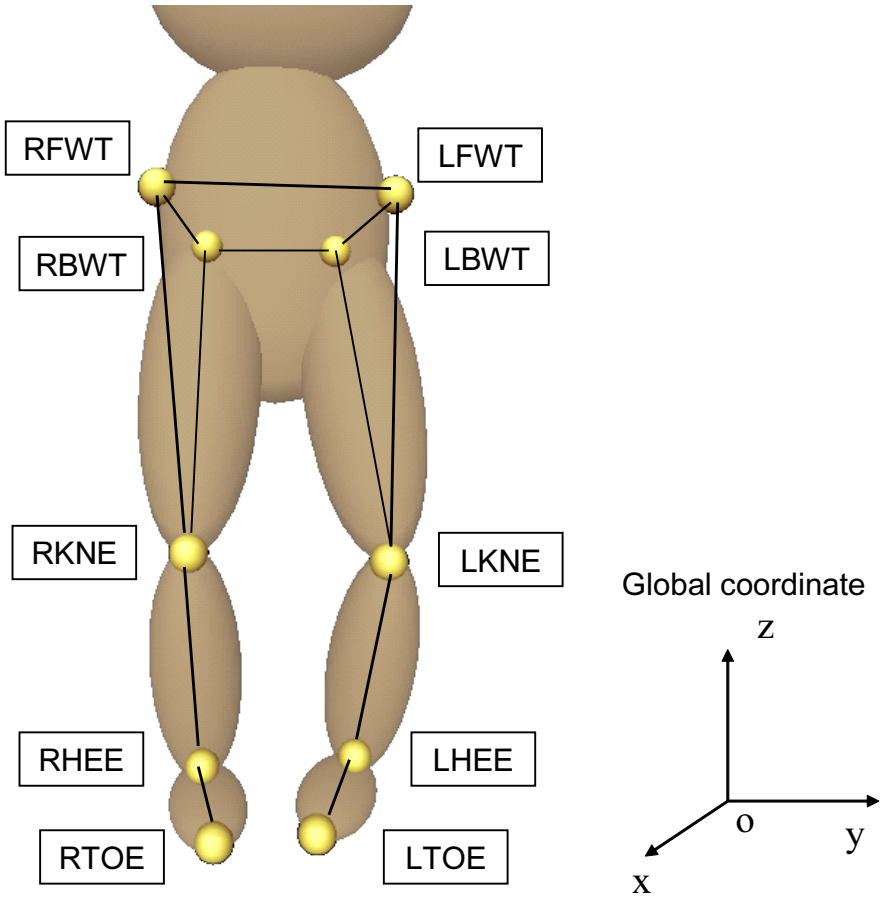


Figure 4.2: Configuration of leg markers. A marker is attached to each toe, heel, and knee, and four markers are attached to the edges of the waist. Each marker is represented by its own label.

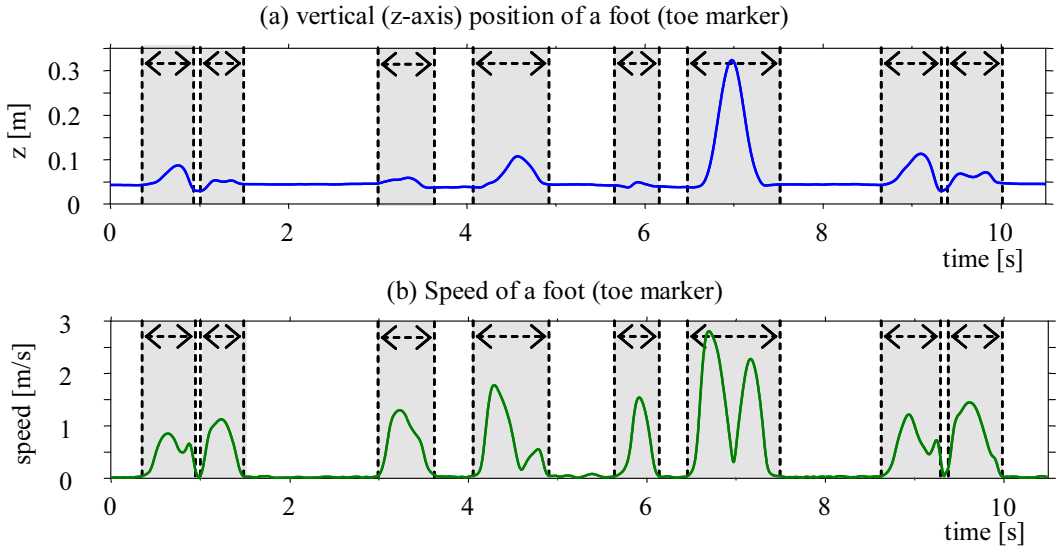


Figure 4.3: Trajectories of a human foot motion, which are analyzed in the process of detecting STEP tasks. Each filled area corresponds to one STEP task in proper thresholds.

as described in Section 3.1, all the movements of a foot are basically regarded as a movement of a foot in stepping. In a trajectory of $v_{rtoe}(t)$, a segment that continuously has positive values is one continuous movement of the foot. A segment from t_0 to t_f that satisfies the following conditions is detected as one R-STEP task:

$$\begin{aligned} \forall t \ \ (t_0 \leq t \leq t_f), \ \ v_{rtoe}(t) \geq v_{step} \\ \int_{t0}^{tf} v_{rtoe}(t) \ dt \geq l_{step}. \end{aligned} \tag{4.2}$$

where v_{step} and l_{step} are threshold values in terms of velocity and moving distance respectively. These thresholds eliminate a noisy slipping motion when the foot is a support foot. By extracting all the segments that satisfy the above conditions, all the R-STEP tasks in the given motion are obtained.

It should be noted that the toe marker is preferable to the heel marker in detection because the heel often floats in the air even when the foot is supporting the body. These movements during a support foot appear in the speed graph, so that analyzing the heel marker results in incorrect detection.

L-STEP tasks are also detected by applying the same process for $\boldsymbol{p}_{ltoe}(t)$, the marker trajectory of the left foot.

After task segments of both the R-STEP and L-STEP are detected, it is possible that a segment of R-STEP overlaps with a segment of L-STEP. Such overlaps do not satisfy the condition of a task sequence.

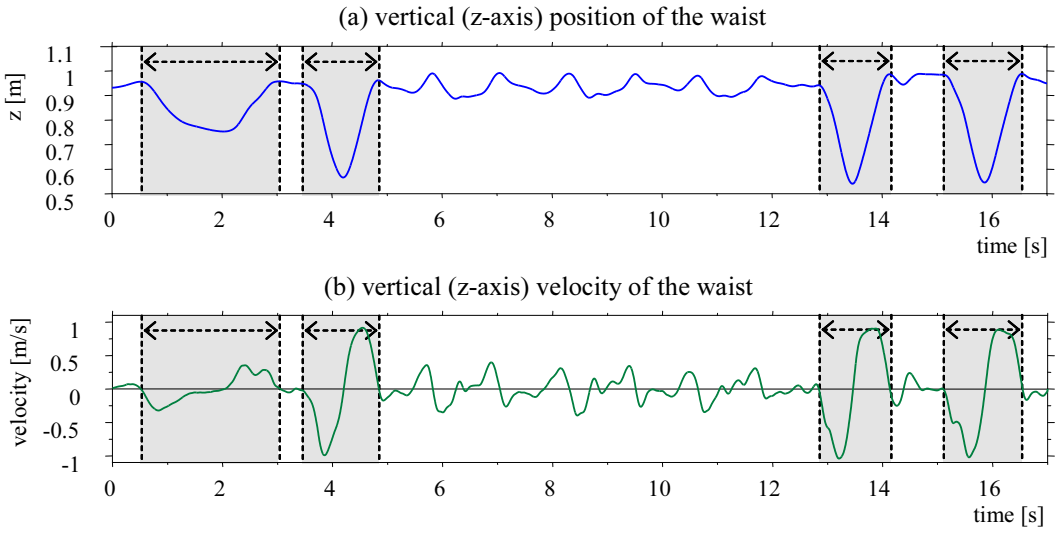


Figure 4.4: Vertical motion trajectories of the waist. Each filled area corresponds to one SQUAT task in a proper threshold.

If the soles do not slip on the floor at all, overlaps do not appear. However, a sole frequently slips slightly when it lands on the floor. A slight slip itself can be ignored because of its slightness, but it can also cause a slight overlap if the next step by another foot begins immediately after the landing. Although motions that include slips of soles are not our target motions that satisfy the condition in a narrow sense, the system should support such slight slips of landing in a practical sense.

The recognition method thereby eliminates such slight overlaps between an R-STEP and an L-STEP. When an overlap is detected, the overlapping segments are shortened so that the segments have an interval between them. The length of the interval is represented by parameter t_{iv}.

4.2.2 Detecting STAND tasks

After segments of the R-STEP and L-STEP are detected, STAND tasks are detected as the segments in which neither R-STEPs nor L-STEPs appear.

4.2.3 Detecting SQUAT tasks

SQUAT tasks are detected by analyzing a vertical trajectory of the waist. First, the trajectory of the waist center

$$\boldsymbol{p}_w(t) = (p_{wx}(t)\ p_{wy}(t)\ p_{wz}(t))^T \tag{4.3}$$

is obtained from the captured markers as follows:

$$\boldsymbol{p}_w(t) = \frac{\boldsymbol{p}_{lfwt}(t) + \boldsymbol{p}_{rfwt}(t) + \boldsymbol{p}_{lbwt}(t) + \boldsymbol{p}_{rbwt}(t)}{4}. \tag{4.4}$$

In Fig.4.4, graph (a) shows the vertical position of the waist $p_{wz}(t)$, and graph (b) shows its velocity $\dot{p}_{wz}(t)$. A SQUAT task corresponds to a motion in which the waist lowers and rises again. This kind of motion is detected as a segment from t_0 to t_f that satisfies the following condition:

$$\begin{cases} \dot{p}_{wz}(t) < 0 & (t_0 \leq t < t_1) \\ \dot{p}_{wz}(t) > 0 & (t_1 < t \leq t_f) \end{cases}, \tag{4.5}$$

$$\int_{t_0}^{t_f} |\dot{p}_{wz}(t)| dt \geq l_{squat}. \tag{4.6}$$

where t_1 corresponds to the timing of the lowest waist position, and l_{squat} is a threshold value for the vertical moving distance, which eliminates slight vertical motions that are not regarded as a SQUAT.

4.3 Extracting Skill Parameters

4.3.1 Extracting Parameters of a STEP task

For a STEP task, skill parameters t_0 and t_f are automatically determined by conditional expression (4.2). Then position parameters are extracted from the correlation between leg markers at t_0 and t_f. The following section describes the case of the L-STEP. For the R-STEP, the parameters are obtained by exchanging the right and left.

First, the base local coordinate Σ_{su} is determined. The attitude matrix $\boldsymbol{R}_{su}$ and origin $\boldsymbol{p}_{su}$ of Σ_{su} are obtained from the markers of the right foot, RTOE, RHEE and RKNE. The x-axis of $\boldsymbol{R}_{su}$ is determined as follows:

$$\begin{aligned} \boldsymbol{a} &= (a_x \ a_y \ a_z)^T \\ &= \boldsymbol{p}_{rtoe}(t_0) - \boldsymbol{p}_{rhee}(t_0) \ , \\ \boldsymbol{a}_{prj} &= (a_x \ a_y \ 0)^T \ , \\ \boldsymbol{b} &= \boldsymbol{p}_{rkne}(t_0) - \boldsymbol{p}_{rhee}(t_0) \ , \\ \boldsymbol{b}_{prj} &= (b_x \ b_y \ 0)^T \ , \\ \boldsymbol{x}_{su} &= \begin{cases} \frac{\boldsymbol{a}_{prj}}{|\boldsymbol{a}_{prj}|} & (|\boldsymbol{a}_{prj}| \geq |\boldsymbol{b}_{prj}|) \\ \frac{\boldsymbol{b}_{prj}}{|\boldsymbol{b}_{prj}|} & (|\boldsymbol{a}_{prj}| < |\boldsymbol{b}_{prj}|) \end{cases} . \end{aligned} \tag{4.7}$$

When the foot makes a pose close to the vertical attitude, the knee direction is employed as shown in these equations. Although Σ_{su} basically represents the position

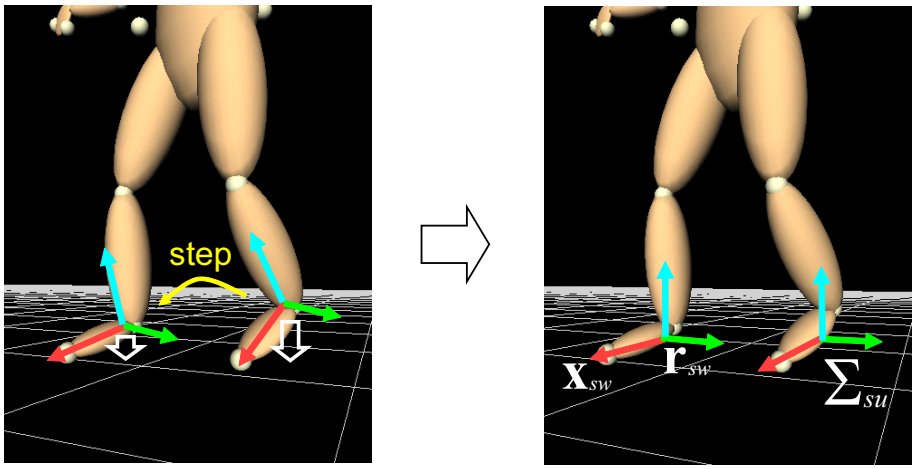


Figure 4.5: Attitudes and positions obtained from the soles. As shown in the left figure, actual attitudes and positions of human soles are not necessarily level with the floor when the soles land on the floor. In the skill parameters, Σ_{su} and the parameters at the landing are recognized as values level with the floor as shown in the right figure.

and attitude of the support foot, its upper direction must correspond to that of the global coordinate. Therefore $\boldsymbol{R}_{su}$ is determined from $\boldsymbol{x}_{su}$ as follows:

$$\begin{aligned} \boldsymbol{z}_{su} &= (0\ 0\ 1)^T\ , \\ \boldsymbol{y}_{su} &= \boldsymbol{z}_{su} \times \boldsymbol{x}_{su}\ , \\ \boldsymbol{R}_{su} &= (\boldsymbol{x}_{su}\ \boldsymbol{y}_{su}\ \boldsymbol{z}_{su})\ . \end{aligned} \tag{4.8}$$

The origin $\boldsymbol{r}_{su}$ is obtained as follows:

$$\begin{aligned} \boldsymbol{c}_{su} &= (1 - c_f)\ \boldsymbol{p}_{rtoe}(t_0) + c_f\ \boldsymbol{p}_{rhee}(t_0) \qquad (0 < c_f \leq 1)\ , \\ \boldsymbol{r}_{su} &= (c_{su\cdot x}\ c_{su\cdot y}\ 0)^T. \end{aligned} \tag{4.9}$$

where c_f determines the foot center between the toe and heel. c_f is usually set near the heel. Like $\boldsymbol{R}_{su}$, z-element of $\boldsymbol{r}_{su}$ must correspond to the global coordinate, in which the vertical position is the same as the floor.

Then the state of the swing foot at landing time t_f is obtained. The direction $\boldsymbol{x}_{sw}$ is determined by equations (4.7) for t_f and markers LTOE, LHEE and LKNE. The position $\boldsymbol{r}_{sw}$ is determined by equations (4.9) for t_f and markers LTOE and LHEE.

Parameter $\boldsymbol{r}_f$ and ψ_f are determined from those values as follows:

$$\begin{aligned} \boldsymbol{r}_f &= \boldsymbol{R}_{su}^T\ (\boldsymbol{r}_{sw} - \boldsymbol{r}_{su})\ , \\ \boldsymbol{x}'_{sw} &= \boldsymbol{R}_{su}^T\ \boldsymbol{x}_{sw}\ , \\ \psi_f &= atan2(x'_{sw\cdot y},\ x'_{sw\cdot x})\ . \end{aligned} \tag{4.10}$$

As shown in Fig.4.5, Σ_{su}, $\boldsymbol{r}_f$ and ψ_f are recognized as values that are level with the floor.

Parameter ψ_f, the waist orientation at time t_f, is obtained as follows:

$$\begin{aligned} \boldsymbol{x}_{wt} &= \boldsymbol{R}_{su}^T \ \{ \ (\ \boldsymbol{p}_{rfwt}(t_f) + \boldsymbol{p}_{lfwt}(t_f) \ ) - (\ \boldsymbol{p}_{rbwt}(t_f) + \boldsymbol{p}_{lbwt}(t_f) \ ) \ \} \ , \\ \psi_f &= atan2(x_{wt\cdot y}, \ x_{wt\cdot x}) \ . \end{aligned} \tag{4.11}$$

In a STEP task, whether the mid-state is enabled or not must be determined. For this purpose, a typical trajectory of the swing foot is generated by interpolation of the beginning position to the finishing position. If the distance between the typical trajectory and the actual trajectory is considerable, the mid-state is enabled so that the trajectory can be expressed.

The interpolated trajectory is generated as follows by using the function (B.1) in Appendix B:

$$\boldsymbol{q}(t) = f_2 \ \langle \ (t_0, \ \boldsymbol{p}_{ltoe}(t_0)), \ (t_f, \ \boldsymbol{p}_{ltoe}(t_f)) \ \rangle \ (t). \tag{4.12}$$

The difference between two trajectories is defined as follows:

$$d(t) = |\boldsymbol{q}(t) - \boldsymbol{p}_{ltoe}(t)|. \tag{4.13}$$

If the following condition is satisfied, the mid-state is enabled at time t_1:

$$\begin{aligned} d(t_1) &= \max_{t_0<t<t_f} d(t) \ , \\ d(t_1) &> d_{step} \ . \end{aligned} \tag{4.14}$$

where d_{step} is a threshold in distance.

If the mid-state is enabled, parameter $\boldsymbol{R}_1$ is determined as follows:

$$\begin{aligned} \boldsymbol{x}_m &= \boldsymbol{r}_{ltoe}(t_1) - \boldsymbol{r}_{rhee}(t_1) \ , \\ \boldsymbol{x}'_m &= \frac{\boldsymbol{x}_m}{|\boldsymbol{x}_m|} \ , \\ \boldsymbol{z}_m &= \boldsymbol{r}_{lkne}(t_1) - \boldsymbol{r}_{rhee}(t_1) \ , \\ \boldsymbol{y}_m &= \boldsymbol{z}_m \times \boldsymbol{x}_m \ , \\ \boldsymbol{y}'_m &= \frac{\boldsymbol{y}_m}{|\boldsymbol{y}_m|} \ , \\ \boldsymbol{z}'_m &= \boldsymbol{x}'_m \times \boldsymbol{y}'_m \ , \\ \boldsymbol{R}_1 &= \boldsymbol{R}_{su}^T \ (\boldsymbol{x}'_m \ \boldsymbol{y}'_m \ \boldsymbol{z}'_m) \ . \end{aligned} \tag{4.15}$$

Parameter $\boldsymbol{r}_1$ is also determined as follows:

$$\begin{aligned} \boldsymbol{c}_{sw1} &= (1 - c_f) \ \boldsymbol{p}_{ltoe}(t_1) + c_f \ \boldsymbol{p}_{lhee}(t_1) \ , \\ \boldsymbol{r}_1 &= \boldsymbol{R}_{su}^T \ (\boldsymbol{c}_{sw1} - \boldsymbol{r}_{su}) \ . \end{aligned} \tag{4.16}$$

4.3.2 Extracting Parameters of a SQUAT task

For a SQUAT task, skill parameters t_0, t_1 and t_f are automatically determined by conditional expression 4.6. Parameter d_1, which represents a vertical difference of the waist in squatting, is determined from the waist markers as follows:

$$\begin{aligned} \boldsymbol{c}_w(t) &= (c_{wx}(t)\ c_{wy}(t)\ c_{wz}(t))^T \\ &= \frac{\boldsymbol{p}_{rfwt}(t) + \boldsymbol{p}_{lfwt}(t) + \boldsymbol{p}_{rfbt}(t) + \boldsymbol{p}_{lfbt}(t)}{4} \ , \\ d_1 &= c_{wz}(t_1) - c_{wz}(t_0) \ . \end{aligned} \tag{4.17}$$

Chapter5

Generation of Leg Tasks

5.1 Introduction

In this chapter, we describe how to generate motions of a robot that executes given tasks.

Fig.5.1 shows the overview of the generation. The generation requires three kinds of data: a sequence of leg tasks, motion trajectories of the upper body, and model data of a robot. These data items are input into a *task generation system*, and the system outputs motion trajectories of the robot whose model was input.

Task sequences used in this system are those extracted from human motions by the method described in Chapter 4.

Motion trajectories of the upper body consist of two data items: one is joint angle trajectories and the other is trajectories of the waist roll and pitch. The latter represents the global inclination of the upper body on the floor. These data items for the upper body are obtained from human motions by the method described in

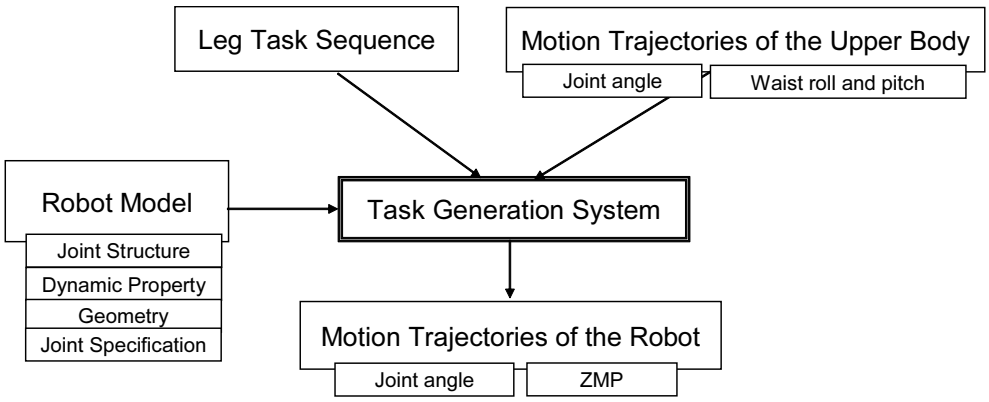


Figure 5.1: Generation Overview

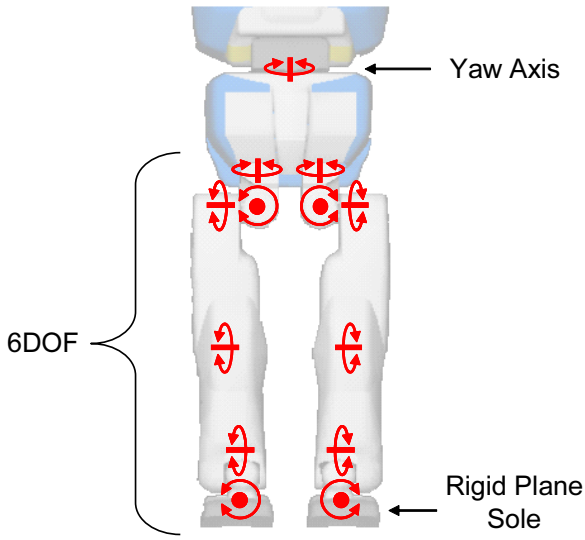


Figure 5.2: Model of the Target Robots

Appendix D.

Model data is required for a robot that executes tasks. The data includes a joint structure, dynamic properties, geometry, and joint specification. A joint structure is required for determining foot trajectory and joint angles of legs. Dynamic properties include mass, center of mass, and an inertia matrix for each body part; they are used for calculating dynamic balance of the robot. Geometry represents shapes of body parts; they are used for checking self-collisions between the body parts. Joint specification includes angle limits and velocity limits of the joints; they are used for checking excess of those limits.

The system assumes that a robot has the structure of the lower body shown in Fig.5.2. A robot must have at least 6-DOF joints in a leg, which enables the soles to move along six axes. The sole of the robot is a rigid plane with a certain area. The robot must also have a yaw axis in its body trunk; it is used for compensating yaw moment so that the robot does not spin on the floor. As long as a robot satisfies these conditions, the system can generate its motion if its model data is given.

The system outputs motion trajectories of the robot. The trajectories consist of joint angle trajectories and a ZMP trajectory. The robot follows the generated joint angle trajectories to execute tasks. The robot also uses the ZMP trajectory to correct errors in dynamic body balance.

The system can generate motions that satisfy the conditions described in Section 3.1. The motions are also assumed to be performed on a flat, level floor.

5.2 Task Generation System

Fig.5.3 shows components of the task generation system and their structure. The components and processes of the system are as follows.

5.2.1 Body State Variables

In the task generation system, the state of the whole body in motion is represented by several *body state variables* shown in Fig.5.3-(b). The whole state is decomposed into the following state variables, which are used as the core information in the generation process.

Position of a swing sole: $\boldsymbol{p}_{sw} = (p_{sw \cdot x}\ p_{sw \cdot y}\ p_{sw \cdot z})^T$. This position is defined as an appropriate point on the sole plane.

Attitude of a swing sole: $\boldsymbol{q}_{sw} = (\phi_{sw}\ \theta_{sw}\ \psi_{sw})^T$, which represents the attitude of the three-axis rotation $\boldsymbol{R}_z(\psi_{sw})\boldsymbol{R}_y(\theta_{sw})\boldsymbol{R}_x(\phi_{sw})$

Waist Position: $\boldsymbol{p}_{wt} = (p_{wt \cdot x}\ p_{wt \cdot y}\ p_{wt \cdot z})^T$. The following elements are separately controlled by different processes:

- Vertical waist position: $p_{wt \cdot z}$
- Horizontal waist position: $p_{wt \cdot x},\ p_{wt \cdot y}$

Waist Attitude: $\boldsymbol{q}_{wt} = (\phi_{wt}\ \theta_{wt}\ \psi_{wt})$, which represents the attitude of the three-axis rotation similar to $\boldsymbol{q}_{sw}$. The following elements are separately controlled by different processes:

- Yaw of the waist: ψ_{wt}. This represents the global orientation of the body on the floor.
- Roll and pitch of the waist: $\phi_{wt},\ \theta_{wt}$. They represent the global inclination of the upper body.

Yaw of the torso joint: ψ_{ts}

Desired ZMP: $\boldsymbol{p}_{zmp} = (p_{zmp \cdot x}\ p_{zmp \cdot y})^T$. Details of ZMP are described in Section 5.5.1.

Fig.5.4 shows the coordinates of these variables in the body.

These state variables are controlled by other components in the system. In Fig.5.3, arrowheads toward a state variable show which process controls that state variable.

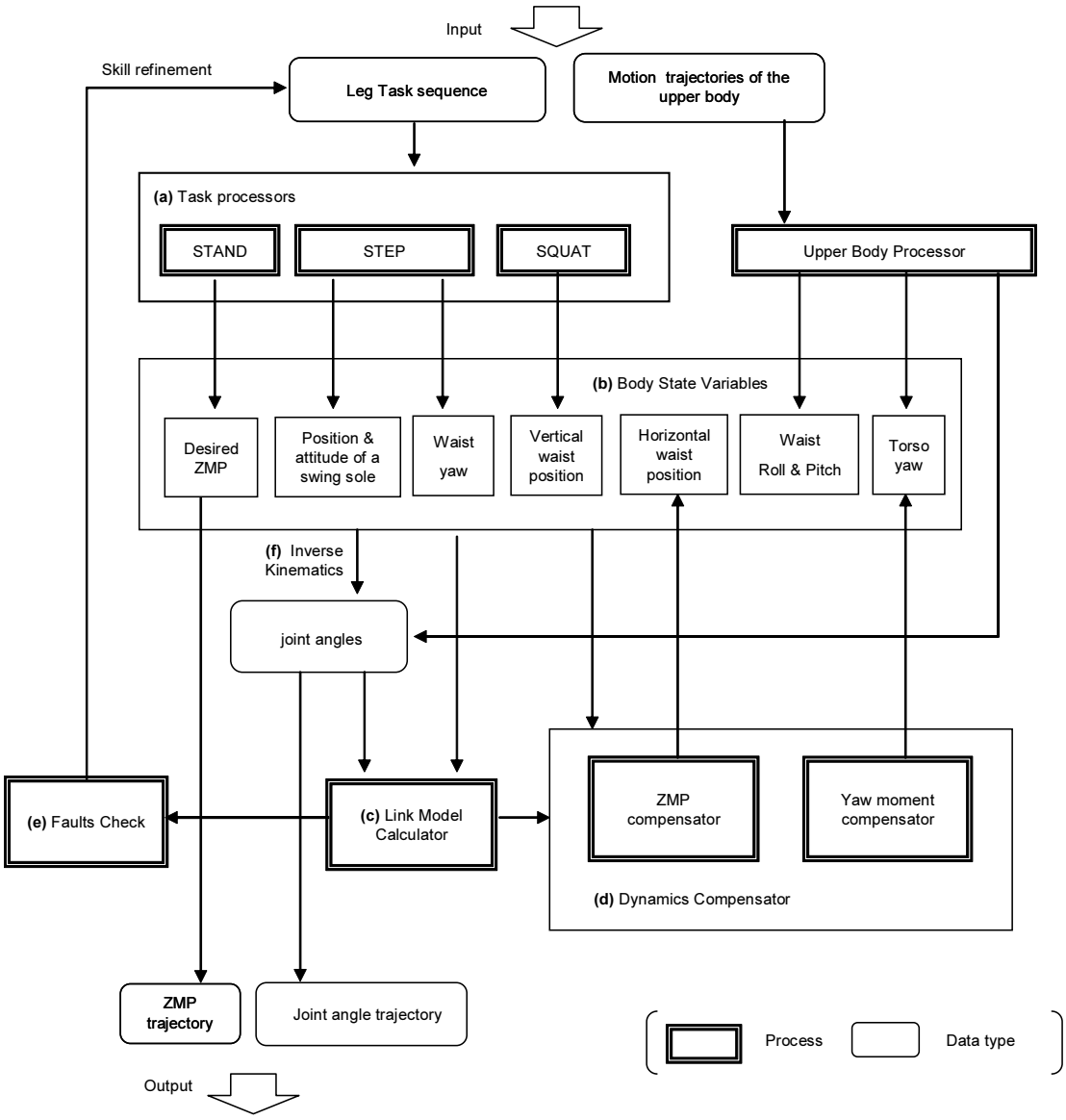


Figure 5.3: Structure of the task generation system

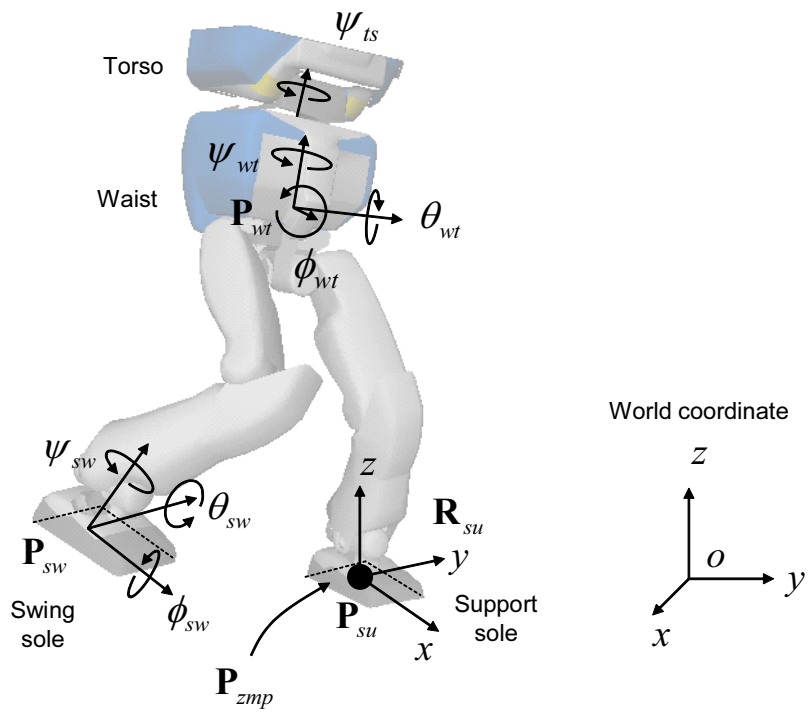


Figure 5.4: Coordinates of the body state variables.

5.2.2 Task Processors

Fig.5.3-(a) shows *task processors*. Task processors control body state variables indicated by the arrowheads from the processor according to the input motion data. For leg task sequences, three processors work for a particular task class. For motion trajectories of the upper body, the upper body processor works. The functions of these processors are summarized as follows.

STAND processor This processor works during a running period of a STAND task. It moves the desired ZMP $\boldsymbol{p}_{zmp}$.

STEP processor This processor works during a running period of a STEP task. It controls a swing foot and global waist yaw, $\boldsymbol{p}_{sw}$, $\boldsymbol{q}_{sw}$ and ψ_{wt}, according to the values of the skill parameters.

SQUAT processor This processor works during a running period of a SQUAT task. It controls the vertical waist position $p_{wt \cdot z}$ according to the values of the skill parameters.

Upper body processor This processor sets poses of the upper body according to the input motion data of the upper body. It also sets the roll and pitch of the

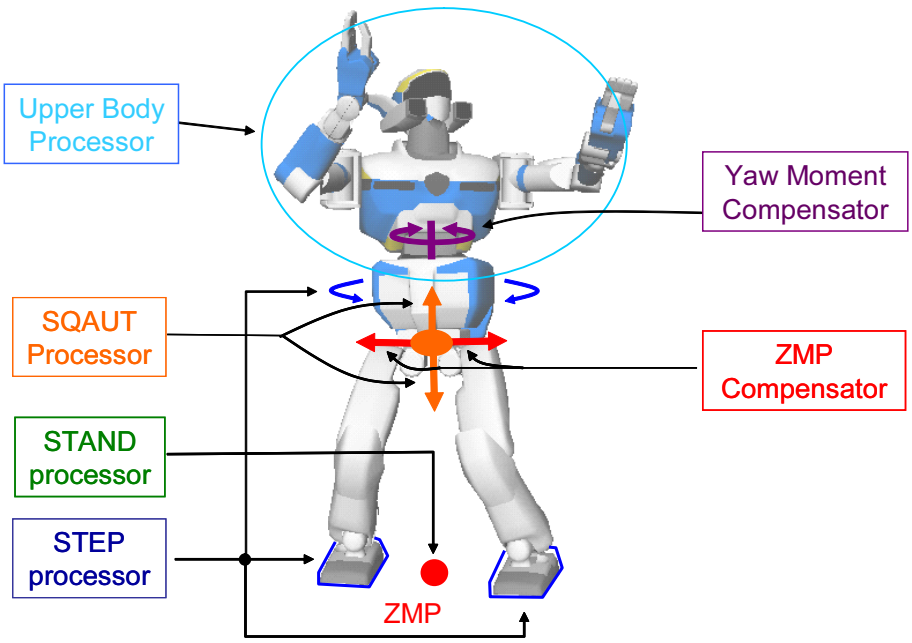


Figure 5.5: Body parts controlled by processors and filters.

waist attitude, ϕ_{wt} and θ_{wt}, according to the data. It works all the time until the input data is finished. Although this processor is not associated with leg tasks, it is classified into task processors for the sake of simplicity. In other words, the function of it is the "task" of the upper body.

Since STAND tasks and STEP tasks do not overlap with each other in a task sequence, the STAND processor and the STEP processor also do not work at the same time. On the other hand, the SQUAT processor and the upper body processor can work simultaneously with the other processors.

Fig.5.5 shows the body parts that are controlled by these processors.

5.2.3 Dynamics Compensator

Body state variables are also controlled by compensators of the whole body dynamics shown in Fig.5.3-(d). The following two compensators work in the system:

ZMP compensator This compensator works to prevent the body from falling down. It compensates ZMP with the horizontal position of the waist, $p_{wt \cdot x}$ and $p_{wt \cdot y}$, according to the desired ZMP.

Yaw moment compensator This compensator works to prevent spins between the soles and the floor. It compensates the yaw moment of the whole body with the

yaw joint of the torso ψ_{ts}.

In contrast to the task processors, these compensators do not directly refer to the input motion data. Their functions focus on adapting the input tasks to the dynamic property of the whole body.

Fig.5.5 also shows the body parts that are controlled by these compensators.

5.2.4 Generation Loop

The task processors and the dynamics compensators work in *generation loops* of a proper time resolution Δt, which is usually determined from specifications of a run-time control system on the robot. In every loop, body state variables are updated, and joint angles of the legs are calculated by inverse kinematics between the feet and the waist. The generation loop is iterated until the final task in an input task sequence is finished, and joint-angle trajectories of the robot are generated.

In a generation loop, faults for executing the motion on the robot such as a collision between the legs are checked (Fig.5.3-(e)). If some faults are detected, the system eliminates the faults by modifying the skill parameters that cause the faults. This process is called *skill refinement*.

5.2.5 Link Model

The system has model data of a robot as a form of *link model*. The link model is expressed by a tree structure in which a node represents a body segment and an edge represents a joint. The generation process refers to the link model in various ways. In Fig.5.3-(c), calculations on the link model such as kinematics simulation and inverse dynamics are processed. These results are used by the dynamics compensators and the process of skill refinement.

5.2.6 Execution Parameters

Table 5.1 shows *execution parameters*, which determine motion behaviors that are independent of tasks. Proper values of these parameters depend not on human motions but on motions determined by the body of a robot. This feature is opposite to skill parameters, the values of which are acquired from human motions. Details of each type of parameter are described in the following sections.

Table 5.1: Execution parameters

h_h	Reference sole height for horizontal impact reduction
h_v	Reference sole height for vertical impact reduction
v_v	Velocity threshold for vertical impact reduction
h_w	Normal waist height
h_s	Normal stepping height
t_z	Time margin of ZMP transition
t_p	Time threshold for pausing ZMP
θ_s	Angle margin of the knee joints for preventing the singular point
Δt	Time resolution of motion trajectories

5.3 Task Processors

A task processor begins to work when the generation loop goes to the beginning time t_0 of a succeeding task. It works until the finishing time of the task, and it stops. In the generation loop, this process is iterated by the number of input tasks.

5.3.1 STEP Processor

The STEP processor controls the position and attitude of the swing sole and the yaw orientation of the waist during task execution. The processor does not control the support sole; it remains in the same position because it is assumed that the sole on the floor does not slip. Fig.5.6 shows the basic function of the STEP processor. The swing sole basically moves along a trajectory generated by interpolation.

The step processor determines stability when a sole contacts the floor. The processor focuses on the following factors for stable contact:

- Satisfying the geometric condition in the contact
- Reducing the impact from the floor in actual performance

The geometric condition means that a sole must perform plane contact with the floor when it supports the body. In order to satisfy the condition, the trajectory of a swing foot is generated so that it satisfies the following condition upon landing:

$$\begin{aligned} p_{sw\cdot z} &= 0 \ , \\ \phi_{sw} = \theta_{sw} &= 0 \ . \end{aligned} \tag{5.1}$$

The condition of the attitude must also be satisfied in positions near the floor. The factors of the impact are described later.

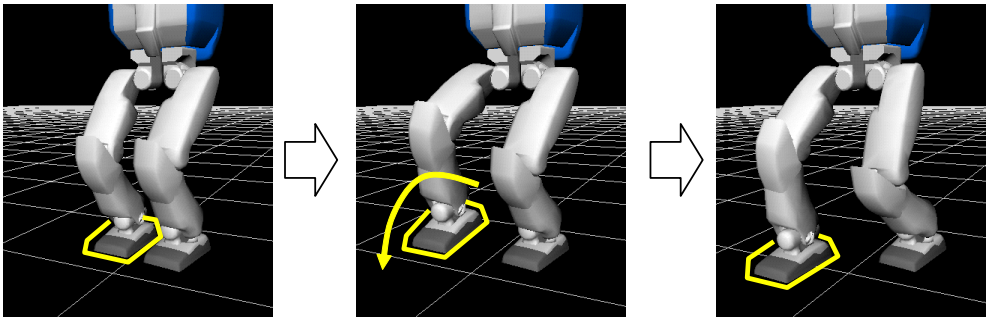


Figure 5.6: Basic function of the STEP processor. The processor moves the swing sole from the beginning position to the finishing position during task execution. The position just before the execution determines the beginning position, and the skill parameters of the task determine the finishing position. The sole basically moves along the trajectory that interpolates those positions.

A concrete trajectory of the swing sole is generated as follows. First, let

$$\boldsymbol{p}_{sw}(t) = (p_{sw\cdot x}(t)\ p_{sw\cdot y}(t)\ p_{sw\cdot z}(t))^T \tag{5.2}$$

be the position of the swing sole at time t during the task execution. This value is set to the state variable $\boldsymbol{p}_{sw}$ in every generation loop. This trajectory is generated from the skill parameters of STEP: t_0, t_1, t_f, $\boldsymbol{r}_1$, $\boldsymbol{r}_f$. Also, let $\boldsymbol{p}_{su}$ and $\boldsymbol{R}_{su}$ be the position and attitude of the support sole at the beginning time t_0, as shown in Fig.5.4. In the following description, values in terms of positions and attitudes are expressed by the world coordinate.

First, we describe the case in which the mid-state of a STEP task is not enabled. A vertical element of a sole trajectory is generated as follows:

$$\begin{aligned} t'_1 &= \frac{t_0 + t_f}{2}\ , \\ p_{sw\cdot z}(t) &= f_3\langle (t_0, 0), (t'_1, h_s), (t_f, 0)\rangle\,(t)\ . \end{aligned} \tag{5.3}$$

where h_s is an execution parameter, which represents a normal stepping height.

This trajectory could cause impact force from the floor upon landing because the vertical velocity of the sole is not adequately reduced before that time. The robot may land the sole just before the planned time because of small errors in control. In this case, the sole collides with the floor at a certain velocity, so that the impact from the floor arises. The impact makes the performance of the robot unstable.

This actual impact can be reduced by generating a trajectory as follows. If the following condition is not true for execution parameters h_v and v_v, the trajectory is

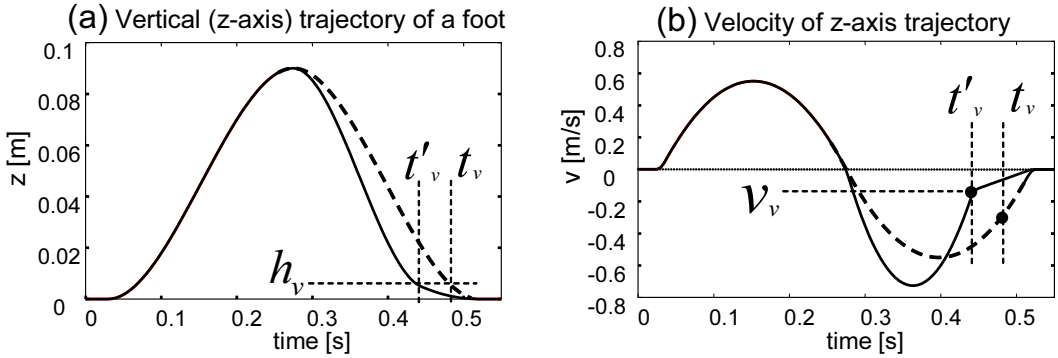


Figure 5.7: Examples of vertical trajectories of a swing sole. Graph (a) shows trajectories of the vertical position, and graph (b) shows their velocities. Dashed lines show trajectories in which the vertical smooth factor is not employed, On the other hand, solid lines show trajectories in which the factor is employed. The latter trajectories can reduce the vertical impact upon landing in an actual performance.

judged that to cause an impact upon landing:

$$\dot{p}_{sw\cdot z}(t_v) \geq -v_v \ , \tag{5.4}$$

where t_v is a value that satisfies

$$p_{sw\cdot z}(t_v) = h_v, \qquad (t'_1 < t_v < t_f). \tag{5.5}$$

In this case, the following trajectory is employed:

$$\begin{aligned} t'_v &= t_f - \frac{2h_v}{v_v} \ , \\ p_{sw\cdot z}(t) &= \begin{cases} f_3\langle (t_0, 0), (t'_1, h_s), (t'_v, h_v, -v_v)\rangle(t) & (t_0 \leq t \leq t'_v) \\ \frac{v_v^2}{4h_v}(t - t_f)^2 & (t'_v < t \leq t_f) \ . \end{cases} \end{aligned} \tag{5.6}$$

In this trajectory, the *vertical smooth factor* is enabled. The trajectory satisfies conditional expression (5.4), and in the final segment ($t'_v < t \leq t_f$), velocity decreases with constant acceleration. Graph (a) of Fig.5.7 shows a trajectory example generated by equation (5.3) (dashed-line) and that by equation (5.6) (solid-line). Graph (b) shows their velocities.

Horizontal elements of the sole trajectory are generated as the trajectory from the beginning position to the finishing position in the STEP task by the following interpolation:

$$\begin{aligned} \boldsymbol{r}'_f &= \boldsymbol{R}_{su}\boldsymbol{r}_f + \boldsymbol{r}_{su} \ , \\ p_{sw\cdot x}(t) &= f_3\langle (t_0, r'_{0\cdot x}), (t_h, r'_{f\cdot x}), (t_f, r'_{f\cdot x})\rangle\,(t) \ , \\ p_{sw\cdot y}(t) &= f_3\langle (t_0, r'_{0\cdot y}), (t_h, r'_{f\cdot y}), (t_f, r'_{f\cdot y})\rangle\,(t) \ . \end{aligned} \tag{5.7}$$

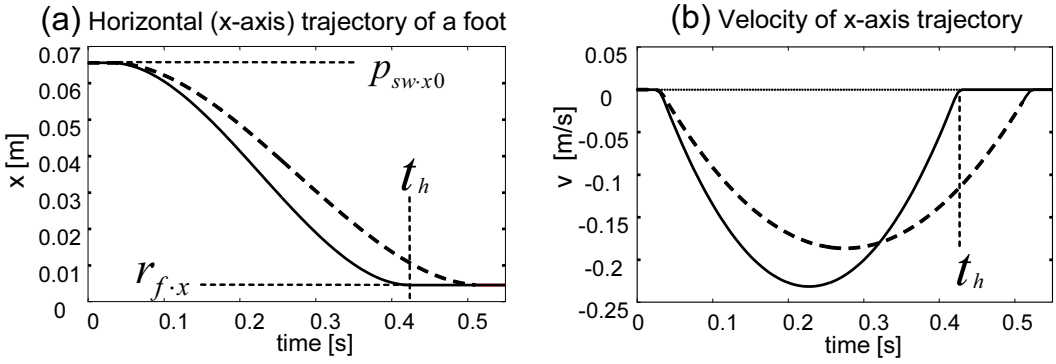


Figure 5.8: Examples of horizontal trajectories of a swing sole. Graph (a) shows trajectories of x-axis position, and graph (b) shows their velocities. Dashed lines show trajectories to which the horizontal smooth factor is not applied, and solid lines show trajectories to which the factor is applied. The latter trajectories can reduce the horizontal impact upon landing in an actual performance.

where $\boldsymbol{r}'_0 = (r'_{0\cdot x}\ r'_{0\cdot y}\ r'_{0\cdot z})$ is the beginning position of the swing sole, which is determined from the previous STEP task. t_h is a value that satisfies the following condition for execution parameter h_h:

$$p_{sw\cdot z}(t_h) = h_h \qquad (t_1 < t_h < t_f)\ . \tag{5.8}$$

The interpolation points at time t_h in equation (5.7), is inserted for reducing horizontal impact upon landing. This is called the *horizontal smooth factor*, which is controlled by parameter h_h. In actual performances, the sole could land on the floor before the horizontal movement of the sole completely stops if this point is not inserted. This is caused by small errors in control, which are similar to the case of vertical impact. The interpolation point at t_h resolves the impact by taking a margin between the end of horizontal movement and the landing. In other words, the sole lands on the floor vertically just before the landing. This way of landing can reduce the horizontal impact from the floor in actual performances. Graphs (a) and (b) of Fig.5.8 show the difference caused by this point. Fig.5.9 shows difference in leg motions.

For the attitude of the swing sole, trajectories of the yaw axis $\psi_{sw}(t)$ are generated as follows. First, values of ψ_f on the world coordinate are obtained:

$$\begin{aligned}(\boldsymbol{e}_x\ \boldsymbol{e}_y\ \boldsymbol{e}_z) &= \boldsymbol{R}_{su}\ ,\\ \psi_{su} &= atan2(e_{x\cdot y},\ e_{x\cdot x})\ ,\\ \psi'_f &= \psi_{su} + \psi_f.\end{aligned} \tag{5.9}$$

Then the trajectory is obtained by interpolation similar to function (5.7).

$$\psi_{sw}(t) = f_3\langle(t_0, \psi'_0), (t_h, \psi'_f), (t_f, \psi'_f)\rangle\,(t)\ , \tag{5.10}$$

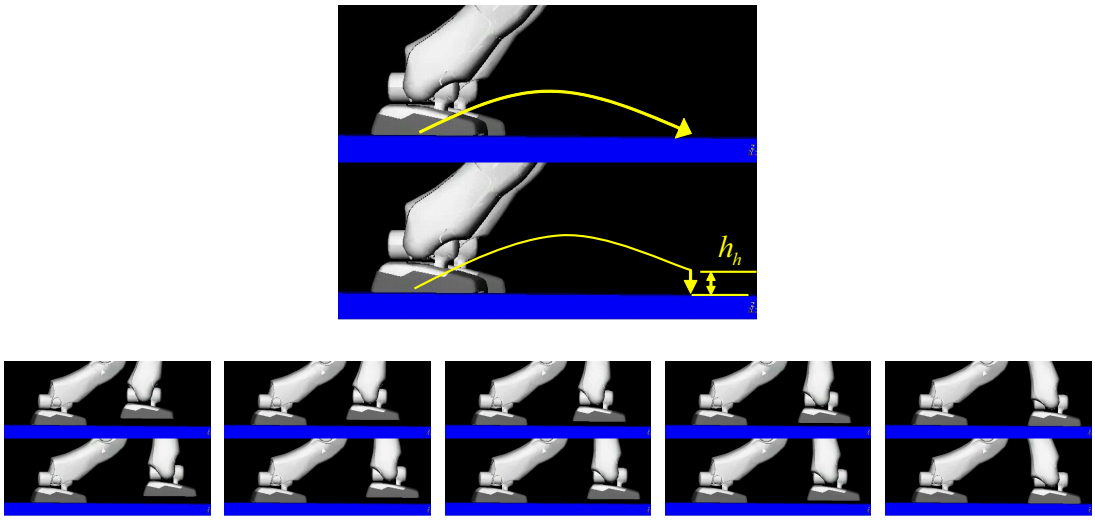


Figure 5.9: Trajectories of a swing foot in leg motions. The upper image shows global trajectories. In this image, the upper part shows a trajectory without the horizontal smooth factor, and the lower part shows that with the factor. The lower images show a sequence of both motions. In the upper part without the factor, the swing foot lands on the floor in a slanting direction. On the other hand, in the lower part with the factor, the swing foot lands on the floor vertically just before the landing.

where ψ'_0 is the yaw orientation of the swing sole at the beginning. Attitude around the horizontal axes is determined as follows:

$$\theta_{sw} = \phi_{sw} = 0 \tag{5.11}$$

Next, we describe the case in which the mid-state is enabled. For the vertical position of the swing sole, equations similar to (5.3) to (5.6) are used. A trajectory is generated by replacing interpolation point (t'_1, h_s) in these equations by $(t_1, r_{1\cdot z})$.

For the horizontal position, the position of the mid-state is inserted into equation 5.7 as follows:

$$\begin{aligned} \boldsymbol{r}'_1 &= \boldsymbol{R}_{su}\boldsymbol{r}_1 + \boldsymbol{p}_{su} \\ p_{sw\cdot x}(t) &= f_4\langle (t_0, r'_{0\cdot x}), (t_1, r'_{1\cdot x}), (t_h, r'_{f\cdot x}), (t_f, r'_{f\cdot x})\rangle\,(t) \\ p_{sw\cdot y}(t) &= f_4\langle (t_0, r'_{0\cdot y}), (t_1, r'_{1\cdot y}), (t_h, r'_{f\cdot y}), (t_f, r'_{f\cdot y})\rangle\,(t). \end{aligned} \tag{5.12}$$

The attitude of the mid-state $\boldsymbol{R}_1$ is decomposed into angles ψ_1, θ_1, ϕ_1, which represent the following three-axis rotation:

$$\boldsymbol{R}_1 = \boldsymbol{R}_z(\psi_1)\boldsymbol{R}_y(\theta_1)\boldsymbol{R}_x(\phi_1)\ . \tag{5.13}$$

Then trajectories of these rotational axes are generated as follows:

$$\begin{aligned}
\psi_1' &= \psi_{su} + \psi_1 \ , \\
\psi_{sw}(t) &= f_4\langle (t_0, \psi_0'), (t_1, \psi_1'), (t_h, \psi_f'), (t_f, \psi_f') \rangle \, (t) \ , \\
\theta_{sw}(t) &= f_5\langle (t_0, 0), (t_h', 0), (t_1, \theta_1), (t_h, 0), (t_f, 0) \rangle \, (t) \ , \\
\phi_{sw}(t) &= f_5\langle (t_0, 0), (t_h', 0), (t_1, \phi_1), (t_h, 0), (t_f, 0) \rangle \, (t) \ .
\end{aligned} \tag{5.14}$$

where t_h and ψ_{su} are the same values as those of equations (5.7) and (5.9), respectively. t_h' is a value that satisfies

$$p_{sw \cdot z}(t_h') = h_h \qquad (t_0 < t_h < t_1). \tag{5.15}$$

The generated values are set to $\boldsymbol{q}_{sw}$ in every generation loop.

Although the final values of the horizontal waist position are determined by the ZMP compensator, the STEP processor temporarily sets a trajectory that is used as the initial trajectory for the compensator. This trajectory is determined to be the center between the feet as follows:

$$p_{wt \cdot x}(t) = f_2\Big\langle \Big(t_0, \frac{p_{sw \cdot x}(t_0) + p_{su \cdot x}}{2}\Big), \Big(t_f, \frac{p_{sw \cdot x}(t_f) + p_{su \cdot x}}{2}\Big) \Big\rangle (t) \ . \tag{5.16}$$

The trajectory of the y-axis is also generated in the same way.

5.3.2 STAND Processor

The STAND processor works while the STEP processor is not working. The main function of the STAND processor is to control the *desired ZMP*. Details of this function are described in Section 5.5.3. This processor does not control the feet or the waist.

5.3.3 SQUAT Processor

The SQUAT processor controls the vertical waist position $p_{wt \cdot z}$. Let $p_{wt \cdot z}(t)$ be the value at time t. Variable $p_{wt \cdot z}$ is usually the value of execution parameter h_w, which represents the normal waist height of the robot. The SQUAT processor changes this value according to its skill parameters during its execution. By using the interpolation function (B.1), the processor generates the following trajectory from skill parameters t_0, t_1, t_f, d_1:

$$p_{wt \cdot z}(t) = f_3\langle (t_0, h_w), (t_1, h_w + d_1), (t_f, h_w) \rangle \, (t). \tag{5.17}$$

The processor sets the value of $p_{wt \cdot z}(t)$ to the body state variable $p_{wt \cdot z}$ in every time frame in the generation loop.

5.3.4 Upper Body Processor

The upper body processor works all the time until the input motion data of the upper body is finished. The motion data sets used by this processor consist of two items: one is joint angle trajectories of the upper body joints, and the other is trajectories of the waist roll and pitch. By using these data items, the processor performs the following two functions:

- To set joint angles of the upper body
- To set roll and pitch of the waist, which are body state variables ϕ_{wt} and θ_{wt}

The processor sets values according to the input data in every generation loop.

5.4 Inverse Kinematics between the Waist and Soles

After the body state variables are determined in a generation loop, joint angles of the legs are calculated by inverse kinematics between the waist and soles. This process is shown in Fig.5.3-(f).

As shown in Fig.5.4, if all the positions and attitudes of the waist and soles are determined, joint angles between them are also determined uniquely because the target robots have 6-DOF joints in a leg.

Usually the inverse kinematics in this process are calculated by analytical equations rather than a numerical solution using Jacobian because the analytical solution is more efficient.

5.5 ZMP Compensator

Task processors determine leg motions by focusing on stability when the soles contact the floor so that a robot can perform motions stably. However, this is not sufficient for total stability. The robot must also take account of consistency in the dynamics between the body and the floor so that the robot does not fall down. The consistency in the robot cannot be obtained from human motions because the weight distribution of the body and the way soles contact the floor are different from those of the human body. Therefore the robot must achieve the consistency in its own body.

In the task generation system, the horizontal waist position is used for this purpose. The position is determined by the ZMP compensator, which is based on the concept of *Zero Moment Point (ZMP)* proposed by Vukobratovic *et al.* [81, 80]. The ZMP compensator works according to the desired ZMP that is determined by the STAND processor.

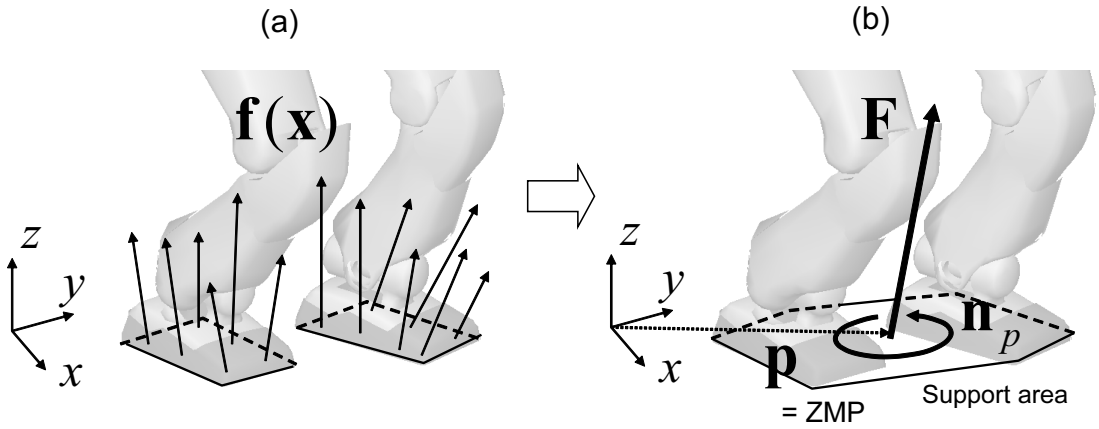


Figure 5.10: Zero Moment Point (ZMP). The floor reaction force is exerted on the soles as shown in (a). This force over the sole area is represented by the force and moment acting on the ZMP as shown in (b). The ZMP is always inside the support area, which is the convex hull constructed by the soles that contact the floor.

In this section, we first describe the basis of the ZMP concept. Then we describe the processes of the ZMP compensator and the STAND processor.

5.5.1 Zero Moment Point

ZMP is a point in which the horizontal elements of the moment caused by the floor reaction force is zero on the floor.

When a body is on the floor, the floor exerts its reaction force on the body. Usually this force is distributed over the sole area, as shown in Fig.5.10-(a). In this situation, the floor reaction force exerts the following moment on point $\boldsymbol{p} = (p_x\ p_y\ p_z)^T$:

$$\boldsymbol{n}_p = \int_S (\boldsymbol{x} - \boldsymbol{p}) \times \boldsymbol{f}(\boldsymbol{x}) d\boldsymbol{x} \ , \tag{5.18}$$

where $\boldsymbol{n}_p = (n_{px}\ n_{py}\ n_{pz})^T$ is the moment, $\mathcal{S}$ is the area in which the feet contact the floor, $\boldsymbol{x}$ is a point in $\mathcal{S}$, and $\boldsymbol{f}(\boldsymbol{x})$ is the force acting on $\boldsymbol{x}$.

If the floor height is zero on the coordinate, the ZMP is a point that satisfies

$$\begin{aligned} n_{px} &= 0 \ , \\ n_{py} &= 0 \ , \\ p_z &= 0. \end{aligned} \tag{5.19}$$

As shown in Fig.5.10-(b), the floor reaction force $\boldsymbol{f}(\boldsymbol{x})$ is represented by the force $\boldsymbol{F}$ and horizontal moment $\boldsymbol{n}_p = (0\ 0\ n_{pz})^T$ that act on the ZMP. The ZMP is always

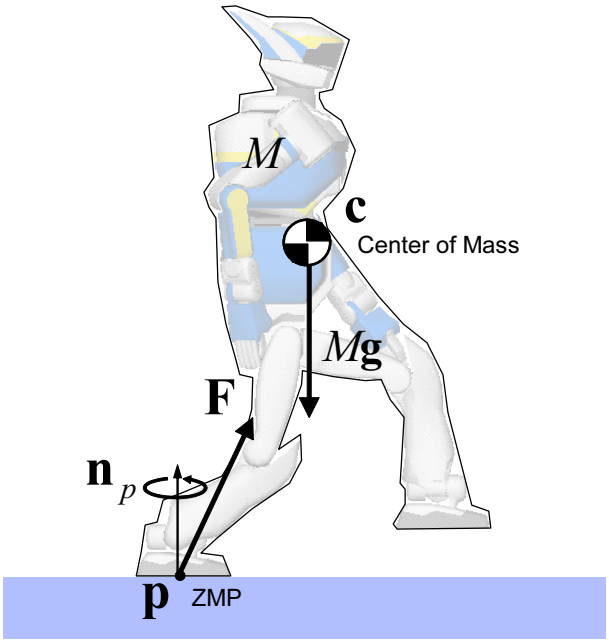


Figure 5.11: The force and moment that act on the body.

inside the convex hull constructed by the soles that contact the floor. This convex hull is called a *support area*.

If the robot contacts only the floor, all the forces that act on the body are the gravitational force and the floor reaction force as shown in Fig.5.11. Since the floor reaction force is represented by force $\boldsymbol{F}$ and moment $\boldsymbol{n}_p$ that act on ZMP $\boldsymbol{p}$, the moment around the origin is written as

$$\boldsymbol{n} = \boldsymbol{p} \times \boldsymbol{F} + \boldsymbol{n}_p \ . \tag{5.20}$$

The motion of the body is written as follows:

$$\begin{aligned} \dot{P} &= M\boldsymbol{g} + \boldsymbol{F} \ , \\ \dot{L} &= \boldsymbol{c} \times M\boldsymbol{g} + \boldsymbol{n} \ . \end{aligned} \tag{5.21}$$

where P and L are the momentum and angular momentum of the whole body respectively, $\boldsymbol{c}$ is the center of mass (CM), M is the total mass, and g is the gravitational constant. Since $\boldsymbol{n}_p$ is at the ZMP, the following equations are satisfied:

$$\begin{aligned} n_{px} &= 0 \ , \\ n_{py} &= 0 \ . \end{aligned} \tag{5.22}$$

Then the horizontal elements of $\boldsymbol{p}$, which means the ZMP, are obtained as follows:

$$
\begin{aligned}
p_x &= \frac{Mgc_x + p_z\dot{P}_x - \dot{L}_y}{Mg + \dot{P}_z} \ , \\
p_x &= \frac{Mgc_y + p_z\dot{P}_y - \dot{L}_x}{Mg + \dot{P}_z} \ .
\end{aligned}
\tag{5.23}
$$

This means that the ZMP can be calculated from the body motion represented by $\dot{\boldsymbol{P}}$ and $\dot{\boldsymbol{L}}$.

5.5.2 ZMP and Dynamic Stability

ZMP is useful for considering the dynamic stability of the body in motion.

The task generation system assumes *the contact condition*, which means that the robot contacts the whole plane of a sole to the floor when the sole is supporting the body. The task processors generate motions so that the motions satisfy this condition with regard to geometry. If this condition is also satisfied with regard to dynamics, the robot can perform the motions stably.

If a support sole does not rotate around the horizontal axes, the sole keeps the plane contact and the robot can perform the intended motion. This is achieved if the ZMP is inside the support area except for the edge of the area. We use "inside the support area" as this sense in the following description. In this case, the moment around the horizontal axes is balanced at the ZMP inside the area, so that the sole does not rotate. Consequently, to keep the ZMP inside the support area satisfies the contact condition with regard to dynamics.

If the dynamic properties of the robot are given, equation (5.23) can calculate the ZMP even from motions that do not necessarily satisfy consistency in the dynamics between the body and the floor. In this case, the calculated ZMP is called the *Imaginary ZMP (IZMP)*. Although the actual ZMP is always inside the support area including its edge, the IZMP can be outside the support area.

Motions in which the IZMP is outside the support area are impossible in the real world. In fact, if the robot tries to perform such motions, the sole of the robot rotates around the horizontal axes and the planned motion cannot be achieved.

In this way, consistency in dynamics can be evaluated from the IZMP, as shown in Fig.5.12. Conversely, if motions are generated so that the IZMP is always inside the support area, the motions can keep the contact condition.

In the task generation system, the *desired ZMP* is determined first. Then the horizontal movement of the waist is determined so that the IZMP calculated from the motion corresponds to the movement of the desired ZMP. In order to achieve this, the system must solve the following problems.

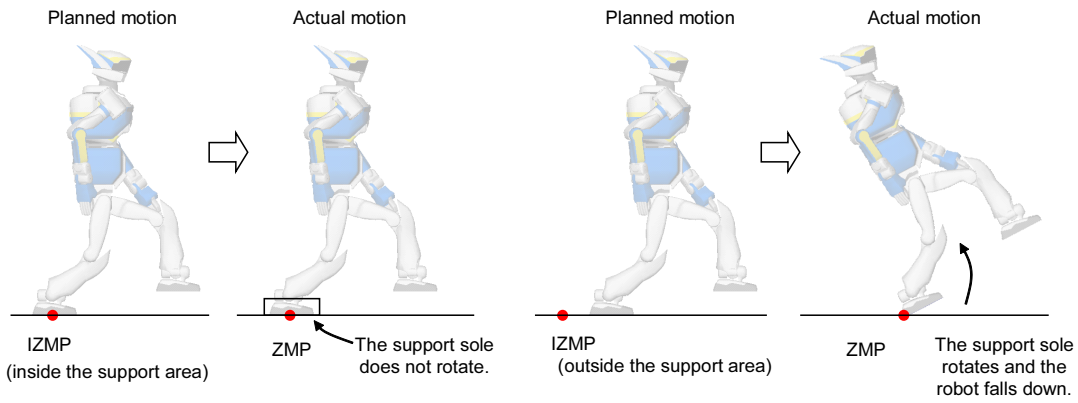


Figure 5.12: The Imaginary ZMP (IZMP) and the actual ZMP. If the IZMP is inside the support area, the motion is possible in the real world. If the IZMP is outside the support area, the motion is impossible. In this case, the sole rotates, so that the robot falls down. The IZMP can evaluate consistency in dynamics in this way.

- How to determine the desired ZMP
- How to make the IZMP consistent with the desired ZMP

5.5.3 Desired ZMP Determined by the STAND Processor

Since the ZMP basically moves from a sole to another sole while the body is supported by both feet, the ZMP moves during a STAND task. This movement of the desired ZMP is controlled by the STAND processor. The STAND processor can determine the desired ZMP because it knows the current and next support area and the time it takes to change from the current body state to the next task in the sequence. As shown in Fig.5.13, a basic transition of the desired ZMP is determined from a task sequence.

Under the condition in which the ZMP is always inside the support area, various trajectories of the desired ZMP are possible. However, this condition is not sufficient for producing stable performances because actual performances involve uncertain factors caused by errors in physical models and errors in controlling the robot. Actual stability therefore depends on the movement of the desired ZMP even if it is always inside the support area.

In order to achieve stability in actual performances, the STAND processor determines the desired ZMP considering the following factors:

- The ZMP is not near the edge of the support area

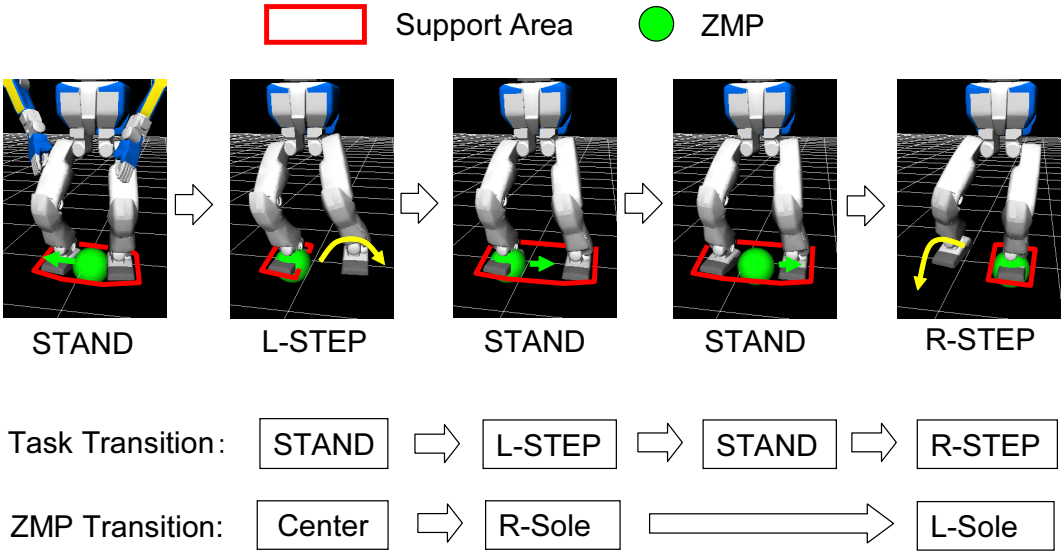


Figure 5.13: Example of a basic transition of the desired ZMP. It is determined from a task sequence.

- The ZMP moves regularly
- The movement of the desired ZMP is similar to that expected in the actual ZMP

First, the desired ZMP should not be near the edge of the support area. In theory, the soles can rotate vertically only when the ZMP is on the edge of the support area. However, if the desired ZMP is near the edge, the possibility of the rotation increases in actual performances because slight errors can move the ZMP to the edge so that the soles cannot maintain the plane contact. Hence it is desirable that the desired ZMP allows a margin of distance from the edge.

Second, the ZMP should move regularly. An irregular movement of the ZMP causes irregular acceleration of the body, which can increase the errors in control. Conversely, regular movement is easier to control, so the errors will decrease.

Finally, the movement of the desired ZMP should be similar to that expected in the actual ZMP. To be more specific, when a sole lands on the floor, the ZMP will move toward the landing sole because floor reaction force is produced at the landing point. If the desired ZMP does not move in such a way, the difference between the desired ZMP and the actual ZMP will increase, so that the actual motion will be unstable.

Considering these factors, the STAND processor determines a concrete trajectory of the desired ZMP as follows.

The desired ZMP at time t is represented as

$$\boldsymbol{p}_{zmp}(t) = (p_{zmp\cdot x}\ p_{zmp\cdot y}\ 0)^T. \tag{5.24}$$

This trajectory is set to the body state variable $\boldsymbol{p}_{zmp}$ in every generation loop.

First, when the support foot of the STEP task executed after a STAND task is different from that executed before it, the STAND processor sets the following trajectory for a ZMP moving on the line between the centers of both the support soles:

$$\begin{aligned}&\boldsymbol{p}_{zmp}(t) = \\ &\quad \boldsymbol{f}_4\langle\ (t_0,\ \boldsymbol{p}_{zmp\cdot 0}),\ (t_0+t_z,\ \boldsymbol{p}_{zmp\cdot 0}),\ (t_f-t_z,\ \boldsymbol{p}_{zmp\cdot f}),\ (t_f,\ \boldsymbol{p}_{zmp\cdot f})\ \rangle\,(t)\ ,\end{aligned} \tag{5.25}$$

where t_0 and t_f are the beginning time and the finishing time of the STAND, respectively. The value $\boldsymbol{p}_{zmp\cdot 0}$ is the center of the support sole in the preceding STEP, $\boldsymbol{p}_{zmp\cdot f}$ is that in the next STEP.

This trajectory pauses at the beginning and ending for the time represented by the execution parameter t_z. In general, the change of the support state can increase uncertain errors in control, so that the stability decreases. The parameter t_z slightly modifies the behavior around changes of the support state. This adjustment can increase stability if the value is appropriate for a specific robot.

When the condition

$$(t_f - t_0) - 4t_z \geq t_p \tag{5.26}$$

is true for the execution parameter t_p, the following points are inserted into the interpolation function in eq.(5.25):

$$(t_0+2t_z,\ \boldsymbol{p}_c),\ \ (t_f-2t_z,\ \boldsymbol{p}_c)\ , \tag{5.27}$$

where $\boldsymbol{p}_c$ is the midpoint between the centers of both soles. This process inserts a motion in which the robot makes a static standing pose for an instant when the STEP task has sufficient execution time.

When the support foot of the STEP task executed after a STAND task is the same as that executed before it, the STAND processor sets the following trajectory:

$$\begin{aligned}\boldsymbol{p}_{zmp}(t) = \boldsymbol{f}_5\Big\langle\ &(t_0,\ \boldsymbol{p}_{zmp\cdot 0}),\ (t_0+t_z,\ \boldsymbol{p}_{zmp\cdot 0}),\\ &\Big(\frac{t_0+t_f}{2},\ \boldsymbol{p}_c\Big),\ (t_f-t_z,\ \boldsymbol{p}_{zmp\cdot 0}),\ (t_f,\ \boldsymbol{p}_{zmp\cdot 0})\ \Big\rangle\,(t)\ .\end{aligned} \tag{5.28}$$

In this case, the source and the destination of the desired ZMP are the same position, but the ZMP moves to the center of the support area and returns. Regardless of the given desired ZMP, the ZMP of the actual robot tends to move toward a swing sole when it lands on the floor because of the reaction force from the floor around the landing position. The desired ZMP trajectory that is close to this movement achieves more stable control on the actual robot. If condition (5.26) is true, a pausing motion is inserted by replacing point $(\frac{t_0+t_f}{2},\ \boldsymbol{p}_c)$ in eq.(5.28) with two points in (5.27), as well as the first case.

In this way, the desired ZMP is determined by the STAND processor. On the other hand, for the period of single-leg support, the desired ZMP remains under the center of the support sole. Since transition to that position is processed by the STAND processor, the STEP processor need not control the desired ZMP.

5.5.4 ZMP Compensation

After the desired ZMP is determined by the STAND processor, the ZMP compensator determines the horizontal waist position so that the IZMP calculated from the motion corresponds to the desired ZMP.

Methods that determines the horizontal movement of the CM from the desired ZMP were proposed by Nishiwaki *et al.* [56] and Kajita *et al.*[16]. We employ Nishiwaki's method because of its efficiency and simplicity. Details of the method are described in Appendix C.

The method requires the following inputs:

- A desired ZMP trajectory
- The CM trajectory of the original motion
- The IZMP trajectory calculated from the original motion

The CM trajectory can be obtained by the forward kinematics of the original motion. In order to calculate the IZMP, first the momentum and angular momentum of the whole body are required. These values are also calculated by the process of the forward kinematics. Then the IZMP can be calculated by equation (5.23).

The method outputs the difference in the horizontal waist position. By applying the difference, the IZMP is approximately close to the desired ZMP. Applying this method can be repeated until the results converge sufficiently.

The ZMP compensator modifies the horizontal waist position as follows. For each time frame in Δt, angles and angular velocities of all the joints including the upper body are set to the link model by a generation loop. The position and attitude of the root link (waist link) is also set. Then forward kinematics is calculated for the link model. The calculation gives center of gravity, momentum, and angular momentum of the whole body; the IZMP is calculated from these values.

By applying the process described above for all the time frames, an IZMP trajectory is obtained. A desired ZMP trajectory is generated by the STAND processor. The compensator inputs these trajectories into Nishiwaki's method, and modifies the waist trajectory according to the output of the method. Finally, the system calculates inverse kinematics between the soles and the waist again, and updates joint angle trajectories of the legs.

The ZMP compensator completely preserves joint angle trajectories of the upper body. It also preserves the original timings and positions of foot steps. Thus, the compensator effectively works for dance motions without losing their characteristics of expression.

5.6 Yaw Moment Compensator

When the yaw-axis moment that a robot exerts on the floor exceeds that exerted by the friction between the soles and the floor, the robot slips on the floor. Even when the original human performance does not include spins, preventing spins must be considered in connection with the robot body.

Tamiya *et al.* [78] proposed a method for constraining the whole body moment under a constant value by a compensation technique. We use the yaw-moment compensation part of this method in the task generation system in order to generate motion that does not cause spins in the actual robot. This process is performed by the yaw moment compensator.

Tamiya's method can use any set of joints for compensation with arbitrary weights. The compensator uses only a yaw-axis joint between the chest and the waist for compensating yaw-moment of the whole body. This joint is represented by a body state variable ψ_{ts}.

Tamiya's method can deal with only single-leg support; this is not a great problem because the moment by friction is sufficient for preventing spins in most cases when the robot is supported by both legs. Thus the compensation is applied only for the period of support by both legs.

5.7 Skill Refinement

The task processors and the compensators of whole body dynamics mainly pay attention to the contact condition that is described in Section 5.5. However, for other kinds of problems, the generated motion can include faults that make executing the motion on the robot impossible. Possible faults include stepping over the movable distance, self-collisions, overruns in angular ranges of the joints, and over-speed in angular velocity limit. These faults are due to the fact that the skill parameters obtained from human motions cannot necessarily be executed on the robot because of the mechanical constraints and difference of body shape. These faults must be eliminated by adjusting skill parameters to the robot body.

To find all the faults in advance without executing the motion is difficult because a humanoid robot is a significantly complicated system that has a high degree of freedom

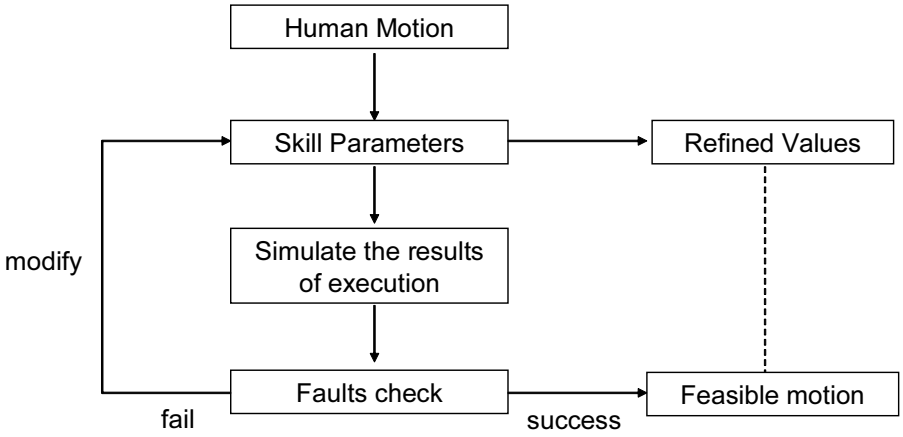


Figure 5.14: Chart of the skill refinement.

and complex body shape, and because the motion is generated through many processes with many factors. In order to find the faults, the task generation system simulates the result of executing the motion The simulation is processed by kinematics calculation in Fig.5.3-(c).

If faults are detected from the simulation, the system modifies values of the skill parameters of the task related to the faults. The modification is processed according to given rules defined for each kind of fault. Then the system runs the time frame back to the time before executing the task, and repeats the generation process to eliminate the faults. Fig.5.14 shows this process.

The following features of the skill parameters make it possible to resolve most faults by simple rules for modifying skill parameters:

- In most of the possible faults, a solution of the skill parameters that does not cause the fault is close to the original value because the value is actually executed by the human dancer.
- Since the number of skill parameters is not great, the number of candidates for modification is limited.

The rules are constructed so that faults are resolved by the smallest modification possible.

Observing case examples of faults is first required for constructing concrete rules for skill refinement. In Section 6.5, we present rules constructed from the faults that actually occurred in the current target dance motions.

5.8 Summary

In this chapter, we described how to generate motions of a robot from a given task sequence. We constructed the task generation system, which integrates various complex factors for determining motions of a robot. The leg task models enable this kind of generation system, which can generate a feasible motion for robots. We summarize how the task generation system solves the problems in reproducing human motion with the legs of a robot.

- The way the soles contact the floor is constrained to plane contact.
- The soles must land on the floor so that the landing does not cause a large impact.

These factors were solved by generating a foot trajectory of a robot from leg tasks. Task processors recomposed motions on the robot according to skill parameters. In this process, the STEP processor considered the problem of sole contact so that these constraints were satisfied.

For reducing the impact of landing, control methods that reduce the impact have been proposed. Takanishi [76] proposed a method that uses torque feedback control to reduce the impact when a heel or a toe lands on the floor. In the control system of H7 [14], the impact is reduced by decreasing the gains of the PD control when a sole lands on the floor. These methods are useful, but the methods cannot necessarily absorb the impact of any foot motions. Especially for dance performances that include wide, high-speed motions, trajectories of the feet strongly affect the impact.

On the other hand, we proposed a trajectory-based solution for this problem. In our method, planned motions themselves include the factor that reduces the impact. This fact achieves the several advantages. First, the method can reduce the impact reliably. Second, the method does not require special hardware or control system. Finally, the method can produce accurate motions required for dances.

- The legs of the robot are not fully extended because of the singular point.

This constraints are also solved by motion recomposition. The normal height of the waist is controlled by the execution parameter h_w. If this value is appropriately determined, the robot can perform motions with poses that stay away from the singular point.

- The leg motions are affected by dynamic body balance under weight distribution different from that of the human body.

The leg task models consider the horizontal waist position as factors that maintain the body balance. The task generation system determines the horizontal waist motion independently of human motions so that the robot can maintain its own body balance. The system employs a ZMP-based method. The ZMP-based method can achieve accurate step patterns, which is desirable for dance performances. We proposed the method that determines the desired ZMP from a task sequence. This method is processed by the STAND processor.

- In most joints, movable ranges are narrower than that of the human body.
- The maximum joint speed of the robot is slower than that of human joints.
- Human motions can cause self-collisions on the robot because of difference in body shape. The robot must avoid self-collisions even if the collisions are not a problem in human motions.

These factors are solved by the skill refinement process, which modifies the values of skill parameters so that the motions meet the limitations of the robot. This process greatly depends on the simplicity of the leg task models. The original values of skill parameters are those actually performed by a human, and the number of skill parameters is not great. These features reduce the complexity of finding a solution, which makes this process practical.

Chapter6

Experimental Results

6.1 Overview

In this chapter, we show the experimental results using human dance motions of an existing dance and an actual robot in order to investigate our methods. Reproduction of dance performances follows the process as shown in Fig.1.5. The process of the experiment is summarized as follows.

1. A human dancer performs a dance in the environment of a motion capture.
2. Position trajectories of body markers are obtained from the motion capture.
3. For the upper body, the marker trajectories are converted into trajectories of joint angles and waist attitude for a robot by the method described in Appendix D.
4. For the lower body, a sequence of leg tasks is recognized from the marker trajectories by the method described in Chapter 4.
5. The task sequence and motion trajectories of the upper body are input into the task generation system described in Chapter 5. The system outputs joint angle trajectories of the whole body and a referential ZMP trajectory.
6. The generated motion data is verified through dynamic simulation.
7. The generated motion data is input into built-in control software of the robot.
8. The robot is controlled according to the given motion data. The robot performs the reproduced dance motions on the floor.

After presenting the experimental results, we discuss our method on the basis of the results in Section 6.7.

6.2 Platform of Experiment

Motion Capture

The motion capture we used in the experiment was an optical-type system named "Vicon." Our system consists of eight infra-red video cameras and 34 body markers. Fig.A.1 in Appendix A shows this system. The system can capture three-dimensional positions of markers at a rate of 120 frames/sec. The number of markers is sufficient to express dance motions except for movements of the fingers. The system basically captures one person at the same time. The frame rate is also sufficient to restore motions smoothly. The system cannot change the directions of the cameras, so that the capturing area is limited. In our setting, the area was about $2.5 \times 2.5[m^2]$.

Humanoid Robot HRP-2

We used a humanoid robot HRP-2 [28] in experiments. HRP-2 is a biped humanoid robot consisting of a whole body with 30-DOF joints. Its height (1.54[m]) and weight (56[kg]) are similar to those of humans. The detailed hardware characteristics of HRP-2 are described in Section 1.3.1.

Built-in Controller

The generated motion trajectories are used as the referential joint trajectories for controlling the robot. They are input into built-in control software for the robot, and the robot is controlled so that it follows the referential joint trajectories.

We used a control system that features on-line stabilization [27]. The controller requires a referential ZMP trajectory in addition to the referential joint trajectories. The controller corrects errors between a given desired ZMP and an actual ZMP during a performance. The actual ZMP is obtained by using the six-axes force sensors embedded in the feet. The errors of ZMP are corrected by slight horizontal movements of the body trunk.

Although a given motion is consistent in dynamics, this kind of feedback stabilization control is necessary for maintaining body balance because actual performances are exposed to various uncertain factors such as errors between the physical models and actual behaviors. Note that the controller can only correct small errors; it cannot correct referential motions that do not satisfy conditions of the physical models.

Dynamics Simulator OpenHRP

We used OpenHRP [23] in dynamics simulation for verifying generated motion data before experiments on the actual robot. OpenHRP is a dynamics simulation system for humanoid robots, which has the following features:

- Many parts of its architecture are open to the public so that users can define their original robot model and can program a new control system.
- The physical body model of robots can be given by a file that is an extension of H-Anim format based on VRML 2.0. This file can be easily edited because VRML 2.0 is an easy-to-understand and well-used format.
- The simulator uses a method proposed by Nakamura and Yamane [50] for calculating the forward dynamics in simulation. This achieves a flexible simulation that allows changes in kinematic structure.
- A spring-dumper model [8] can be used for calculating behavior of contact. This model can produce more realistic results with regard to contact between soles and the floor. This is important in simulating motions that include steps.
- The same control programs can be used between virtual robots for simulation and actual robots [24]. This can reduce extra work in operating a simulation.

Software Implementation

The programs of the method of task recognition, the method of converting upper body motions, and the task generation system were implemented into our software application [51], which integrates functions for dealing with whole body motions of biped humanoid robots. Users can use GUI for operations and see motions by computer graphics. Fig.6.1 shows a screen-shot of the software.

The system can present human motions, task sequences, and motions of robots with both graphical and numerical forms. The animation of these data can be synchronized with each other. In this way, the methods of this study can be processed and verified easily on the GUI and CG. The software has compatibility with OpenHRP [23] in file format of physical body models and motion data. This feature enables us to share those files with OpenHRP.

6.3 Target Dance

We chose a dance called 'Aizu-Bandaisan,' which is a traditional Japanese folk dance, for an experimental dance. The motion speed of this dance is not very fast, and

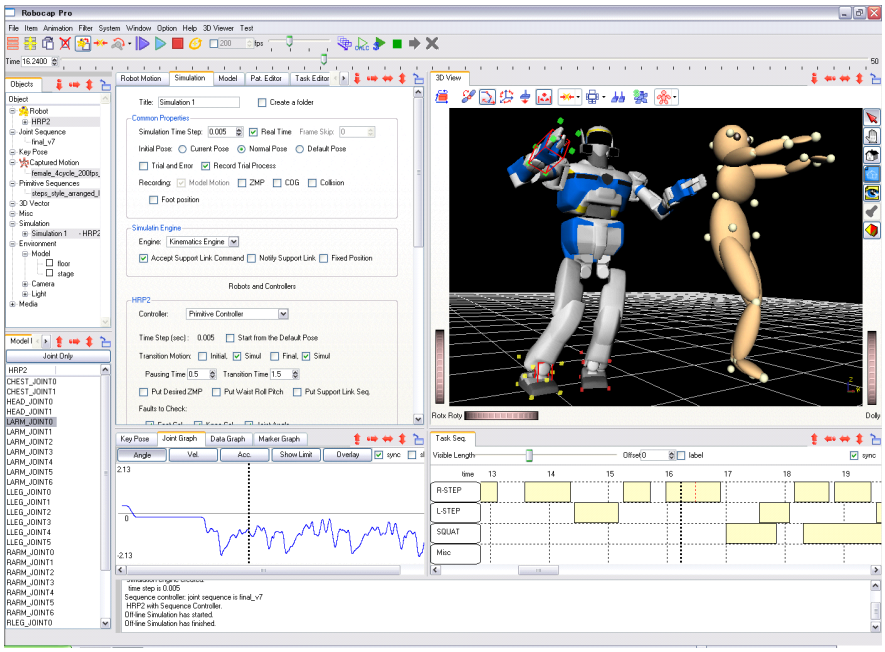


Figure 6.1: Integrated software application developed by us. Our methods were implemented on this software. The software provides various functions including GUI operations and 3D viewers, so that the methods can be easily tested on the software.

the dance includes characteristic motions of the whole body with many leg motions.

We captured the marker trajectories of the dance motion performed by two dancers: a female dancer A and a male dancer B. Fig.6.2 shows the captured motions of these performances. They performed in time to the same dance music. We obtained two items of motion data from their performances; both items are 170 seconds long and have 19 repetitions of the same choreography pattern. We extracted the initial 35 seconds with four repetitions for experimental data.

6.4 Results of Task Recognition

The method of task recognition requires determining the threshold values described in Section 4.2. Graph (a) in Fig.6.3 shows the correlation between the threshold l_{step} and the number of STEP tasks detected. This indicates that 43 STEPs were detected in dancer A in the range $0.02[m] \leq l_{step} \leq 0.13[m]$, which is a stable result. When l_{step} was less than that range, the result rapidly increased because even slight slips of a support foot were recognized as tasks. Conversely, if l_{step} was larger, some of the true STEPs were eliminated. Likewise, 44 STEPs were stably detected in dancer B in the range $0.02[m] \leq l_{step} \leq 0.24[m]$. We determined l_{step} to be $0.04[m]$ for the

Figure 6.2: Motions of Aizu-Bandaisan captured from two dancers. The right body in the images is dancer A, and the left body is dancer B.

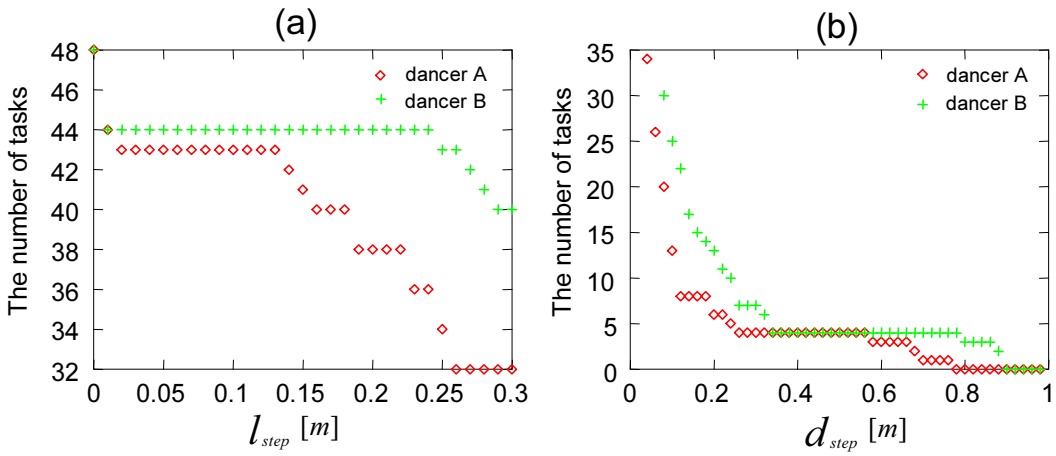


Figure 6.3: Correlation between thresholds and detection results.

value correctly detecting STEP tasks. The threshold v_{step} does not affect the result significantly; it was determined to be $0.04[m/s]$.

The threshold d_{step} determines whether a stepping motion is a characteristic one and a robot pays attention to its expression, or the motion is a simple one and the robot concentrates on its stable execution. Graph (b) in Fig.6.3 shows the correlation between d_{step} and the number of STEPs in which the mid-state was recognized as valid. This indicates that four is the stable result of dancer A in the range $0.26[m] \leq d_{step} \leq 0.56[m]$. The same number of STEPs was stably obtained in dancer B in the range $0.34[m] \leq d_{step} \leq 0.78[m]$. We determined d_{step} to be $0.4[m]$ from these results. All the STEP tasks in which the mid-state was recognized as valid were motions like a kick, an example of which is shown in Fig.6.4-(a), and all the motions like a kick in the dance were recognized as a STEP with the mid-state.

The threshold l_{squat} in the STEP detection determines resolution of vertical waist motion. We determined l_{squat} to be $0.12[m]$, and the motions shown in Fig.6.4-(b) were recognized as a SQUAT from dancer A. On the other hand, dancer B performed the motions shown in Fig.6.4-(c) in the same timings, and these motions were not recognized as a SQUAT. The number of SQUATs detected was 7 in dancer A, and 1 in dancer B.

The numbers of tasks detected were different between dancer A and B. This is due to choreography details that differed with individual dancers.

Fig.6.5 shows the whole task sequence on the time line that was recognized from dancer A.

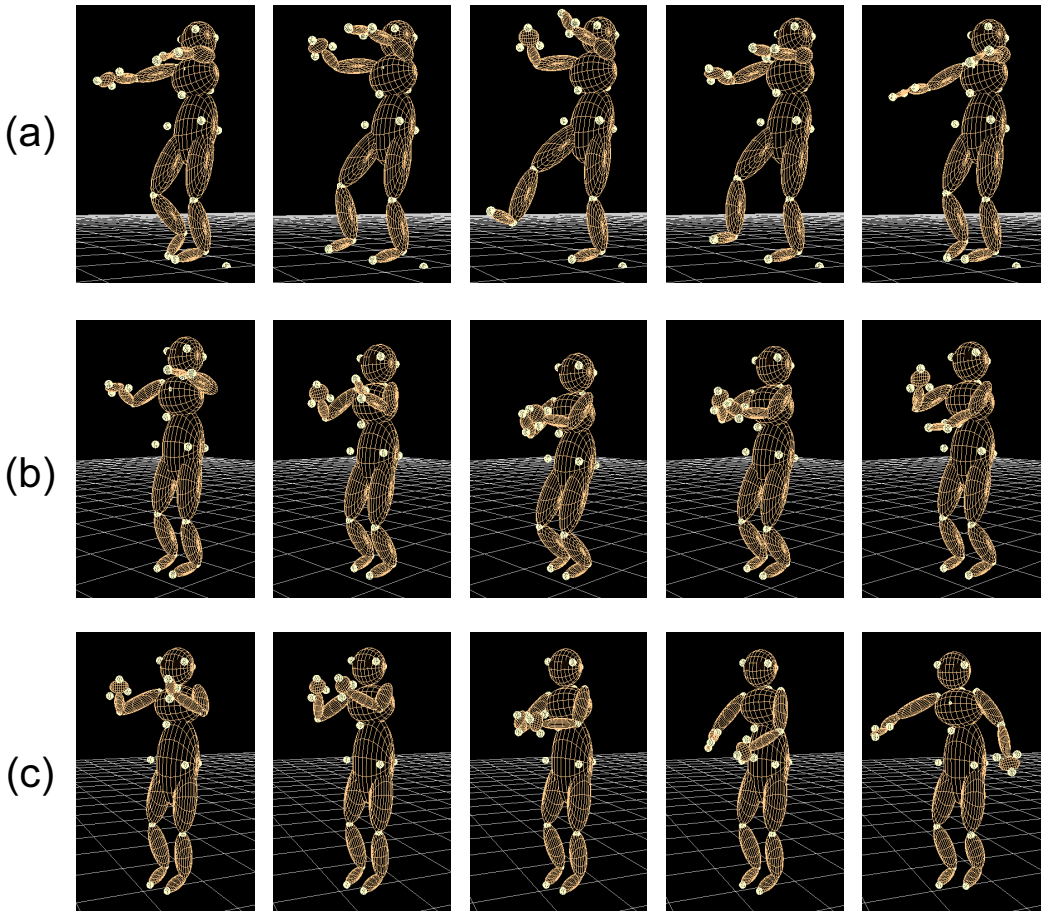


Figure 6.4: Motion examples. (a) is a kick-up motion, which is detected as a STEP with a mid-state. (b) is detected as a SQUAT. (c) is a motion of another dancer at the same timing as (b), which is not detected as a SQUAT.

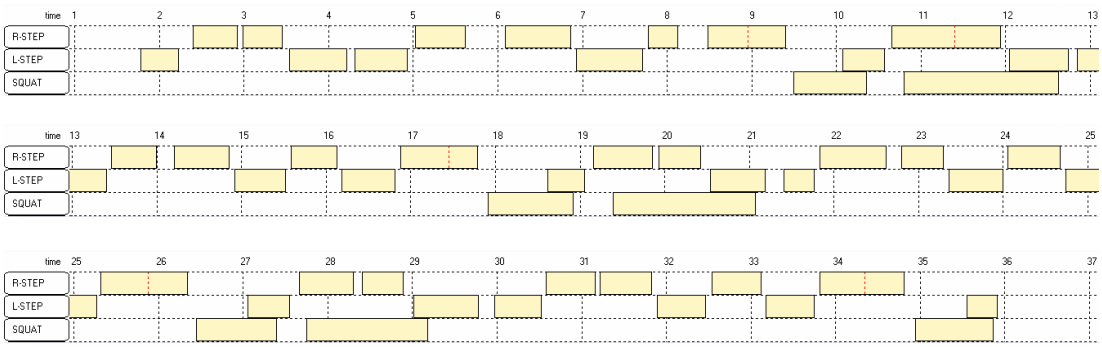


Figure 6.5: The whole task sequence recognized from the dancer A. STAND tasks are omitted from a sequence in this figure.

6.5 Generating Motion Data of HRP-2

6.5.1 Values of the Execution Parameters

The task generation system requires determining the appropriate values of execution parameters for a specific robot.

The parameter h_w, which represents a normal height of the waist, is closely related to the problem of a singular point in the leg joint structure. The generation system does not suppose the singular point in which the knee is fully extended. Parameter h_w must be lower than the waist height in extending the legs, which is $0.71[m]$ in HRP-2. A value close to the height of the extended pose makes the possible stepping distance shorter, and the scale of the original stepping motion cannot be sufficiently realized. Conversely, too low a height makes leg poses too different from those in human performances. We determined h_w to be $0.61[m]$ considering these respects.

The other parameters were determined as follows: h_s was $0.05[m]$, t_p was $0.4[s]$, θ_s was $38[deg]$ and Δ_t was $0.005[s]$. The left parameters were determined through executing motion on the robot, which is described in Section 6.6.

The compensators of whole body dynamics in the generation system were applied as follows; first, the ZMP compensator was applied, and next, the yaw moment compensator was applied. Finally the ZMP compensator was applied again. Through this process, well-converged results in whole body dynamics were obtained.

6.5.2 Observed Faults and Refinement Rules

For the skill refinement described in Section 5.7, faults shown in Fig.6.6 were observed in the original skill values.

(a) is an overrunning of the angle limit of a coxa yaw joint. This fault can be eliminated by modifying the parameter ψ_f of a STEP task within the joint limit.

(b) is an overrunning of the possible step distance. In this case, the swing sole cannot reach the goal position $\boldsymbol{r}_f$, and the knee joint gets into the singular point. This fault can be eliminated by modifying $\boldsymbol{r}_f$ toward the beginning position of the swing sole in a STEP. When this fault occurs on the way to the mid-state, the fault is eliminated by moving $\boldsymbol{r}_1$, the position of the mid-state, toward the direction that shortens distances both to the beginning and finishing positions of the swing sole. This fault is detected by checking whether the angles of the knee joints are close to the singular point; the angles less than execution parameter θ_s are recognized as the fault.

(c) is a self-collision between the knee joints. This fault can be eliminated by modifying the yaw-element of $\boldsymbol{R}_f$, which is the finishing attitude of the swing sole in the last STEP.

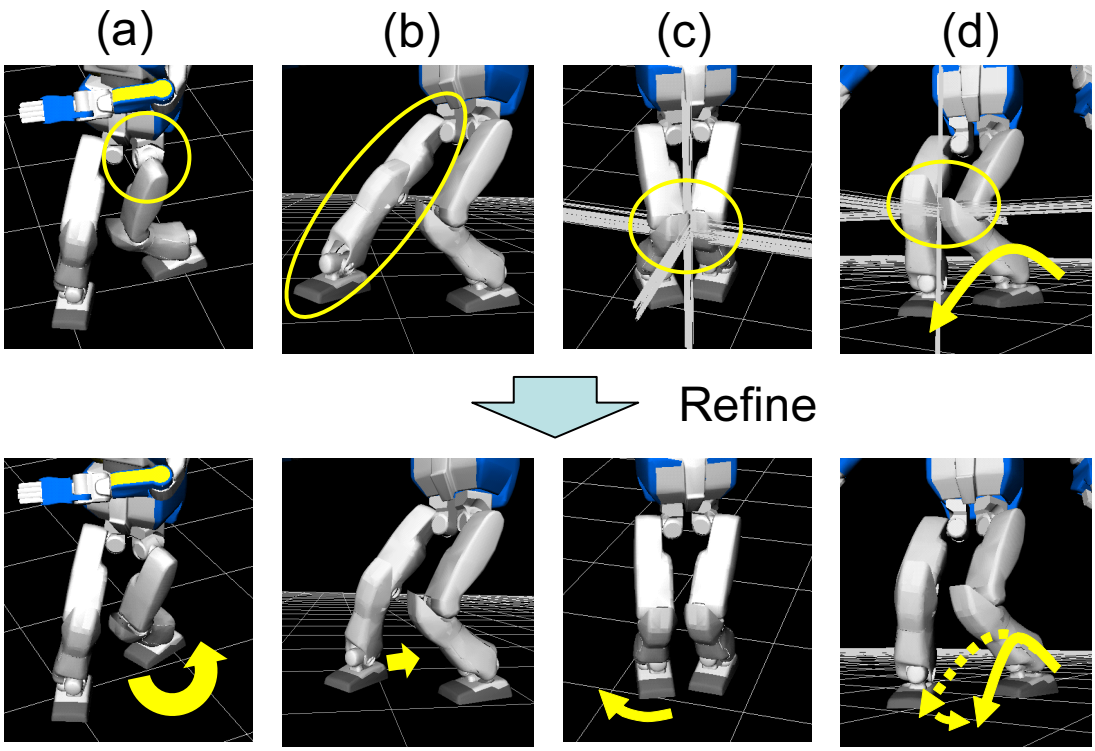


Figure 6.6: Fault examples. (a) is an overrunning of the angle limit of a coxa yaw joint. (b) is an overrunning of the possible step distance. (c) is a self-collision between the knee joints. (d) is a self-collision during stepping.

(d) is a self-collision during stepping. This fault can be eliminated by rotating $\boldsymbol{r}_f$ and $\boldsymbol{R}_f$ away from the support sole on the coordinate of the swing sole at the beginning in the STEP.

In addition to these faults, overruns of the angular velocities in the knee joints were observed. This fault is eliminated by applying the same modification as in (b).

In dancer A, the following numbers of faults were observed: (a) was 3, (b) was 8, (c) was 1, and (d) was 2. In dancer B, (a) was 0, (b) was 20, (c) was 2, and (d) was 0. A number of fault (b)s were observed in dancer B, because dancer B tends to step with long strides. Note that the frequency of fault (b) is a trade-off against the normal waist height h_w; lower h_w reduces the frequency of this fault. After eliminating these faults, a number of overruns of the angular velocities in the knee joints were still observed; the number of them was 14 in all 43 STEPs in dancer A, and 11 in all 44 STEPs in dancer B.

The ways for eliminating the faults described above have been implemented as rules of skill refinement. The implementation automatically eliminated all the faults observed in the experimental data.

Finally, the task generation system generated motion data that satisfies the contact condition in geometry and dynamics, and causes no faults on HRP-2.

6.6 Dance Performance by HRP-2

The execution parameters related to the stability of a robot were determined by experimenting with actual behavior. We obtained the following values for producing stable performances on HRP-2: $h_h = 0.006[m]$, $h_v = 0.005[m]$, $v_v = 0.13[m/s]$, $t_z = 0.025[s]$.

Finally, HRP-2 successfully performed the dance motions of dancers A and B at the original motion speed. Fig.6.7 shows the performance by HRP-2, which imitated dancer A, and the performance by dancer A. Fig.6.8 shows both the performances reproduced from dancer A and B.

6.7 Discussion

In this section, we discuss feasibility, efficiency, and fidelity of the reproduction by considering the experimental results.

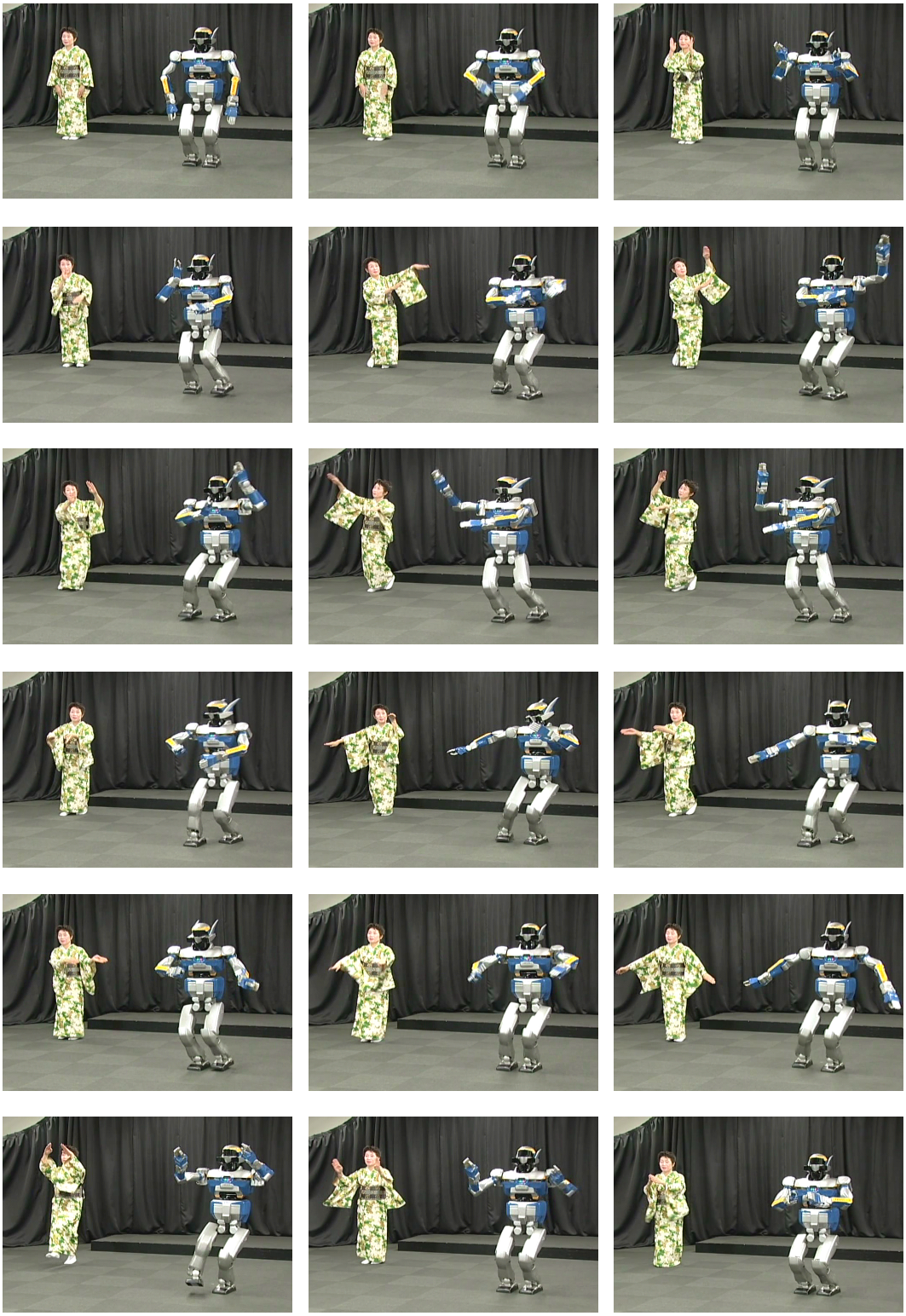


Figure 6.7: Dance performance of Aizu-Bandaisan by HRP-2 and a human master. HRP-2 successfully performed the imitative dance motion of the original music tempo.

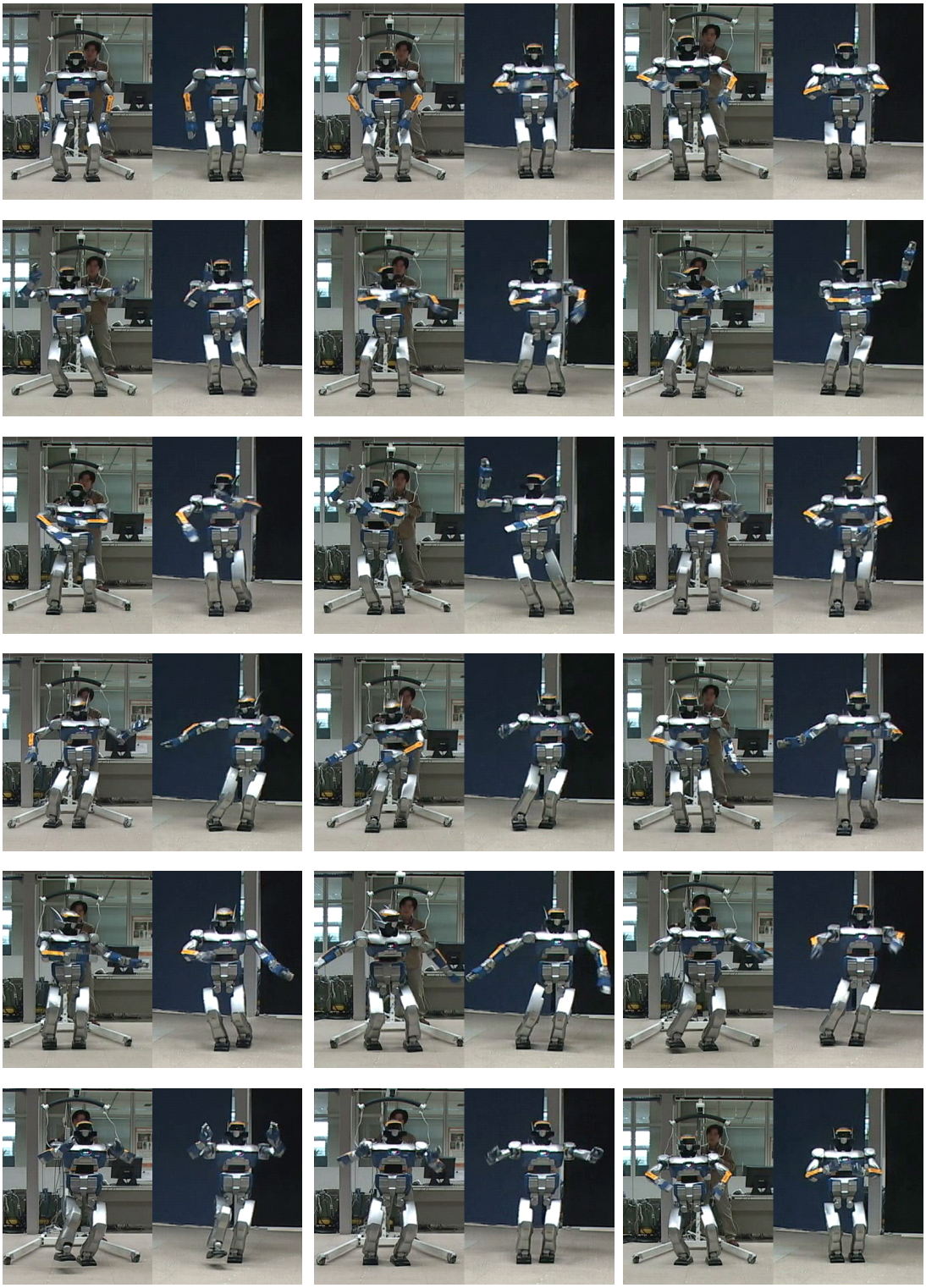


Figure 6.8: Comparison of the performances reproduced from dancer A and B. The right motion in the images is reproduced from dancer A, and the left from dancer B.

6.7.1 Feasibility

The dance we used for the experiments was anything but an easy-to-follow motion. As shown in Fig.6.2, the dance includes unstable, dynamic motions such as a dancer standing on one leg raising the other leg, while extending arms widely. The dance also includes a number of continuous steps with various styles. For this motion, the defined leg tasks were sufficient to cover all motions with regard to kinds of basic motions, and tasks were correctly detected by our method. As for expression of tasks, we discuss this in Section 6.7.3.

HRP-2 stably performed this dance motion without falling down. To be precise, the whole plane of a sole was not detached from the floor while it was a support sole. This means that the contact condition described in Section 5.5.2 was always satisfied, which is a reliable criterion upon which to verify the stability. On the other hand, when we tried to simulate leg motions generated by the method for upper body motions, which is described in Appendix D, the robot immediately fell down as shown in Fig.1.4. These facts indicate that our method is reliable in generating feasible motions for robots in stability.

It should be noted that the setting of execution parameters described in Section 6.6 is important for producing stable performances. When the setting of the execution parameters was not appropriate for the robot and the environment, unstable behaviors were observed; for example, significant impacts arose when a sole landed on the floor, or a part of a support sole was detached from the floor. These events sometimes caused the robot to fall down. These facts indicate that the stability of a robot strongly depends on small, delicate motion factors, which must be appropriately determined for a specific robot. These factors are not obtained from human motions at all, because the difference in legs between humans and robots are not small. Our framework based on recognition and generation is reasonable for controlling those factors of stability. This means that the LFO paradigm works effectively for the problem of feasibility related to stability.

As described in Section 6.5.2, the process of the skill refinement is necessary for generating feasible motions. For the dance we use, we verified that skill refinement can eliminate the faults in motion that are caused by body constraints. The observed faults were eliminated by several simple rules defined for the faults. The simplicity was achieved by the framework based on the leg task models in which motions are represented by minimum skill parameters required for leg expression. This means the LFO paradigm works effectively for the problem of feasibility related to the body constraints.

However, the rules of skill refinement currently implemented are ad-hoc ones constructed only from the faults observed in particular dance motions. It is not certain

that the rules have sufficient generality for various kinds of dance motions. To construct a rule set that has sufficient generality is a research area we must explore as future work.

6.7.2 Efficiency

One of the significant advantages of the LFO paradigm is efficiency in teaching a robot. Our final goal is that a robot reproduces dances motions only by observing human demonstrations.

For this goal, most of our system has been automated. In the recognition process, a task sequence is automatically recognized from human motion data. In the generation process, motion data of a robot is automatically generated from a given task sequence. Although our current implementation is not built-in software of a robot but a PC application with GUI, the implementation enables a user to carry out these processes with some button clicks. For reference, the time consumed by the processes of experimental data that is 35 seconds long are less than a second for recognition and several seconds for generation on a 1.6[GHz]-Pentium M processor. These facts show that the advantage of efficiency has been achieved for the recognition process and the generation process.

However, there are still some parts of the process that are not completely automated. First, the observation process performed by motion capture is not automated. Human demonstrators have to attach a number of markers to their bodies. An operator is required for motion-capture equipment, and the operator has to do some additional operations processing data. Although this problem is not a subject of this thesis, it is certainly desirable that the observation process be automated. As a solution to this problem, an optical-type motion capture that does not require body markers has been studied [3]. This system will reduce the work in the capturing operation. The ideal goal is to use the eyes of the robots for the observation process.

Second, there are several parameters that must be manually determined. Those are the threshold parameters in the recognition process and the execution parameters in the generation process. In order to determine appropriate values for the execution parameters, we needed to have dozens of executions on the robot, which was a great effort for us. However, once appropriate values are determined for a robot through several kinds of motions, the values will also be valid for other performances on the same robot.

In fact, the same setting was valid for the performances of dancers A and B. In our experiments, we first tried to reproduce dancer A. More than ten trials were required to reproduce stable performances because of the initial parameter setting. After that, a stable performance for dancer B was reproduced by the first trial using the same values

of the parameters. In addition to the execution parameters, the threshold values of the task recognition were valid for both the performances. These facts indicates that the recognition and the generation will be completely automatic once those parameters are determined. As for the execution parameters, it may be possible for a robot to autonomously determine those values by observing its body sensors through actual performances.

In summary, we can say that the efficiency of the LFO paradigm was proved for the core process of the reproduction.

6.7.3 Fidelity

Fidelity in expression is also important to dance performances. One of the important factors in expression is rhythm of motions. In this study, key timings of tasks are represented by skill parameters, and the original timings obtained from a human performance are completely preserved in a performance of a robot. This enables a robot to reproduce a high fidelity performance in rhythm.

For overall fidelity including other factors, further investigation is required. For this problem, individual characteristics in dance expression can be key factors in evaluation. In our experimental motions, dancer A and dancer B have their own characteristics in expression. As seen in Fig.6.2, detailed poses are different between two dancers. According to dance masters, they can easily distinguish the original performers just by viewing the movements of the captured markers. The reason for this is that the markers represent individual characteristics of expression. Accordingly, if masters can distinguish the original performers from performances of a robot, we can say that the robot achieved fidelity to the original performances.

We reproduced dancers A and B as shown in Fig.6.8. The question is whether these performances reflected the characteristics of the original dancers or not. In fact, the grand master, dancer A, clearly distinguished the original dancers by observing the performances of HRP-2. Furthermore, she stated that the robot even expressed her habits of leg motions. This means that the fidelity of our method was highly regarded by the specialist in her field. This finding gave strong support to our method.

This way of subjective evaluation is one valid approach to evaluate the method. However, currently only the two performances were evaluated by one person, which is not a sufficient evaluation. More evaluation tests based on this subjective way must be conducted for many dances and dancers.

Development of objective evaluation may also be worth trying. In this approach, fidelity is evaluated by a method that outputs quantitative similarity in dance expression. However, a simple comparison of trajectory differences will not be useful because the grand master recognized the fidelity of motions of the robot, which did not always

follow the same trajectories because of the constraints. Any evaluation must evaluate the essence of expression, so we must first try subjective evaluation in order to know the essence for evaluation.

The fact that the dance performances of the robot were successfully reproduced from human dance motions is a significant achievement of our method. We verified that the leg task models and our proposed methods worked efficiently as we intended. Although the experiments were carried out only for the two items of motion data, good results were obtained with regard to stability, efficiency, and fidelity, as described above. The results of our previous experiments also support the validity of our method (see Appendix E). However, further investigation using many kinds of motion data is required for increasing confidence in our method. This is our future work.

Chapter7

Conclusion

7.1 Summary

In this thesis, we described a method that enables a biped humanoid robot to reproduce human dance motions with its whole body. The method is based on the Learning from Observation paradigm, which is a paradigm for teaching tasks to a robot just by demonstrating the tasks. The paradigm consists of the following three processes: observation, recognition, and execution. In the observation process, the motion of human demonstration is obtained. We use a motion capture for obtaining motion trajectories of human body parts. Then the observed motions are recognized on the basis of task models, and the robot executes the motion according to the recognition result.

This thesis especially focused on these processes for leg motions. Since legs must maintain stable contact of the soles to the floor and maintain the dynamic body balance so that the body does not fall down, leg motions are severely constrained when they try to follow human motion trajectories. The way of contacting the soles is severely constrained because the soles are not flexible like human soles. The weight distribution of the body is different from that of humans; this causes a different dynamic balance. As well as these leg-specific factors, robots are under general constraints caused by their joint mechanism and body shape. Furthermore, dances require a robot to preserve essential characteristics in expression.

We showed that the problem of leg motions is solved by the processes of recognition and generation based on the LFO paradigm. We described our solution that consists of the leg task models, the method of recognizing leg tasks, and the method of generating leg tasks. These components of our reproduction framework are summarized as follows.

Leg Task Models We defined leg task models, which are top-down motion modes that represent essential leg motions. Four basic leg motions, STAND, R-STEP, L-STEP, and SQUAT are modeled as tasks. STAND, R-STEP and L-STEP

are defined by focusing on the state of contact between soles and the floor, and SQUAT focuses on the state of the waist. A robot can clearly know "what to do" from these tasks. The essential characteristics of each task are described by its skill parameters. Skill parameters are designed so that motions are described with a small number of parameters required for expressing motion characteristics. The robots can clearly know "how to do tasks" from values of the skill parameters. Leg motions in a performance are described as a sequence of a number of tasks that have their own values for the skill parameters.

Recognition of Leg Tasks We developed a method that automatically recognized a sequence of leg tasks from human motion trajectories. This method detects tasks by focusing on the velocity of a body part that is associated with a kind of task. STEP tasks are detected by analyzing the velocity of a foot marker. STAND tasks are detected as a period in which any STEPs are not detected. SQUAT tasks are detected by analyzing the vertical velocity of the waist markers. After task segments are detected in this way, skill parameters of each task are extracted. Timing parameters are determined first, then position parameters are determined from the correlation between the associated markers. Our top-down task models made the recognition process efficient and reliable.

Generation of Leg Tasks We developed a method that automatically generates motions of a robot from a given sequence of leg tasks. The method was constructed as the task generation system, which consists of several components. The function of each component is clear in the system, which made it possible to control the complex motion factors of biped humanoid robots. Each component uses clear information represented by leg task models for generating motions that are executable on the robot. Trajectories of feet are generated from STEP tasks so that a foot performs the way of stepping described by skill parameters. This generation also satisfies stable contact between the soles and the floor. The dynamic balance of the whole body is satisfied by the process based on ZMP. STAND tasks move the desired ZMP during its running period considering the stability on dynamic balance. Then the horizontal waist position is determined from the desired ZMP. The vertical waist position is determined by SQUAT tasks. Faults caused by body constraints are eliminated by skill refinement. When faults are detected, this process modifies the skill parameters associated with the faults so that faults are eliminated.

Adapting Upper Body Motions For upper body motions, human motions are converted into motions of a robot by a process independent of leg tasks. The constraints of upper body motions are not as severe as leg motions because the upper

body does not interact with the floor. On the other hand, motion trajectories themselves are important for expression of upper body motions. These features of upper body motions make a contrast with leg motions. It is hence desirable that the upper body can employ its own motion models or processing method. In order to achieve this, the leg task models are designed so that upper body motion can be decoupled from tasks of the lower body. This design enables the task generation system to generate feasible motion data without modifying a given upper body motions. We presented a simple, trajectory-based conversion method for upper body motions in Appendix D.

We implemented our methods, and investigated them through experiments with the dance Aizu-Bandaisan and the humanoid robot HRP-2. First, the task sequences corresponding to the original human performances were correctly recognized from the captured trajectories. Next, our generation method output motion data for HRP-2 from the recognition results. The generated motion data satisfied all the conditions for executing the motion on HRP-2. Finally, HRP-2 successfully performed the reproduced dance motions without falling down at the same motion speed as that of human performances. The dance expression including leg motions was highly regarded by the grand master of the dance.

7.2 Contribution

The contribution of this study is summarized as follows.

- We achieved the Learning from Observation paradigm for whole body performances of biped humanoid robots. Human instructors were able to teach dances to the robot just by demonstrating the dances. The method eliminated work by specialists on robots in the basic sense. The human master was sure that the dance motions of the robot reflected the essence of the original human performances well. These facts means that our method based on the paradigm successfully achieved the advantages of the paradigm. Our method provides the basis of adapting the LFO paradigm to tasks of biped humanoid robots.

- We enabled a biped humanoid robot to reproduce human dance motions using its legs to support the body on the floor. Including leg motions for supporting a body in reproduction is a novel achievement of humanoid robots. It is our method based on the leg task models that has achieved this for the first time.

- We proposed leg task models, which are necessary for achieving the LFO paradigm on biped humanoid robots. Top-down definition of the models provides essential information required for both expression of dances and execution of robots.

Leg-specific motion models that have the features of supporting both the requirements have not been proposed by other studies. The success of the dance reproduction proved the validity of our leg task models.

- We developed a method for recognizing leg tasks from observed human motion trajectories. This method is necessary for the framework of learning based on the leg task models. The validity of the method was certified through experiments on human motions that performed existing dances.

- We developed the task generation system, which generates motion data that are executable on a robot from a given task sequence. This system is necessary for the framework of learning based on leg task models. The validity of the system was certified through experiments using HRP-2 and the task sequence obtained by our recognition method. We proposed the structure of the system that consists of several motion factors and processes that control them. The structure is designed so that it can control detailed parts of complex motion factors of biped humanoid robots. Also, the processes include the following methods proposed by us.

  - We proposed a method for generating a sole trajectory that reduces the impact of landing. This method does not depend on a specific hardware mechanism or a specific control system, so the method is available to any robot that has a standard mechanism and control system. Accurate control can be performed even when impact reduction is enabled. This feature is necessary for dance performances in which motions must keep accurate timings. The framework based on the leg task models makes this method available for reproducing human motions.
  - We proposed a method for determining a desired ZMP based on a given task sequence. The method can clearly determine a desired ZMP required for a stable performance because the contact state of the soles is clearly indicated by running tasks. We proved that the leg task models effectively work for this process.
  - We proposed a framework of skill refinement, which can automatically eliminates the faults caused by the body constraints of robots. In most of the possible faults, a solution of the skill parameters that does not cause the fault is close to the original value because the value is actually executed by the human dancer. Also, since the leg task models represent motions with the minimum number of skill parameters required for expression, searching space for a solution are small. These features makes the skill refinement practical for acquiring solutions with simple modification rules. We showed the validity of this framework through experiments.

- Our method enables biped a humanoid robot to be used as a medium for presenting human performances using the whole body. This is a useful application that exploits the advantage of human-like looks and bodies of humanoid robots. The method can be applicable for various motions because the original goal was reproducing dance motions, which generally includes various, dynamic motions. The use of this medium will increase the practical values of biped humanoid robots.

7.3 Future Work

More Tests on Various Dance Motions

It is desirable that experiments on many other dances and dancers will be carried out. In this thesis, we tested our method by using two motions of the same dance performed by two dancers. We showed the validity of our method on the basis of the results of these motions as described in Section 6.7. However, we did not sufficiently show the generality of our method through experiments. Experiments on various motion data can increase confidence in our method, or reveal required improvements to make the method more general.

Regarding fidelity in dance expression, subjective evaluation as described in Section 6.7.3 should be carried out. First, several motions of the same dance performed by different dancers are reproduced on a robot. Then dancers watch only lower body motions of the reproduced dances. Next, they guess the original human dancer for each performance, and document reasons for confidence in their guess. The result of this evaluation shows how well the original characteristics are reproduced and is valid for evaluating the fidelity of the reproduction in itself. In addition, it provides key knowledge of the essence in dance expression. On the basis of this knowledge, we will be able to consider improvement of our method and development of an objective method for evaluating motions.

Through the tests of various motions, we must also consider rules for skill refinement. We cannot say that the current implementation of the refinement rules has sufficient generality to support various motions. Although we show the validity of the framework of skill refinement, the current rules are ad-hoc one for eliminating the faults that happened in the two experiments we conducted. There will be other kinds of possible faults, and conflicts between different faults can happen unless the rules are well designed. Since humanoid robots are complex systems, possible cases of these problems are not easily clarified. For the first step, it is reasonable to investigate possible cases through tests on various motions.

Robots are expected to master various dances through these tests. This is also a

valuable achievement in a practical sense. The dance performances themselves have a value as interesting displays of humanoid robots. Furthermore, if robots master traditional dances that are regarded as an intangible cultural heritage, they can help to preserve those dances because actual performances by robots are probably more impressive, effective presentation than other existing technologies such as videos or computer graphics. As the result of tests on various dances, a dance archive by humanoid robots will be also achieved [54].

Interactive Performance

If our method enables a robot to perform dances in an interactive way, the method can expand its use.

Dances performances essentially include interactive factors. Dancing to music is the usual way to perform dances. Dancers listen to the music and fit their motion to the music. When several dancers perform a dance together, they often interact with each other. They can flexibly adjust their movement to another dancer, or a spontaneous motion of one dancer may trigger another spontaneous motion by another dancer.

Our method does not enable a robot to perform those interactive dances. However, the method has sufficient potential to support those interactive performances because the method has a feature of flexibility by nature. Since motions are represented by a task sequence that clearly describe motions based on task models, the motions are easily modified as intended by editing the task sequence. In addition to this flexibility, the interactive performance requires the following factors:

- Sensing ability
- Real-time edit of a task sequence
- Flexibility of upper body motions

First, a robot must recognize a subject of interaction. Dancing to music requires the ability to detect the beat or melody of the music. Interaction with another dancer requires the ability to recognize the other dancer's motion in real time. This requires an advanced vision using cameras of a robot, which are installed in the head as eyes of the robot.

Second, the flexibility must be available in real time. A function of task edit that our system provides must be improved so that the robot can edit its own tasks according to interaction and generate the edited tasks in real time.

Finally, the upper body motion must also have flexibility so that the interactive performance can be available not only for leg motions but also for whole body motions.

As a first attempt of an interactive performance, we want to enable a robot to dance to music. Our research group has already developed methods that are useful for this purpose. They proposed a method that detects beats of music and uses them for recognizing a structure of dance motions [71, 52]. They also proposed a method that automatically creates dance motions from given music [55]. We think that dancing to music can be achieved by integrating these methods into the improved system.

More Extensive Tasks

The kinds of motions our method can deal with are limited to those under the conditions described in Section 3.2. In fact, many dances can be reproduced under these conditions, but there are also many dances that cannot be reproduced under the conditions. That is to say, the dances that include jumping, running, or spinning cannot be supported by the current system. It is desirable that these motions are supported so that the generality of our method increases. In order to reproduce those kinds of motions, improvement of the defined tasks or development of additional tasks is required. At the same time, the improvement of the hardware is necessary. It is probably impossible to reproduce those motions on the current hardware no matter how much the software is improved. We will improve our method in step with the hardware improvement so that our method can support more extensive tasks.

Appendix A

Motion Captures

A motion capture is a system for capturing properties of human body posture in a time series. Recently the technology of motion capture has progressed, and these systems are in practical use. Currently motion capture has been used in various fields that include computer graphics, video games, and biomechanics. Robotics is one of those fields.

A.1 Optical Systems

In this thesis, we use an optical-type motion capture. Basic components of the optical-type systems are several video cameras and dozens of markers. The markers are attached to the body; a sufficient number of markers is attached with appropriate body positions so that the body motion can be represented only by the motions of the markers. The system also has lights to illuminate the markers, and a computer system for processing the input data and storing results of capturing. In order to distinguish the markers clearly from the background, many systems force a performer to wear a mono-colored suit that covers almost the whole body and to put on the markers on the suit. Furthermore, many systems use infra-red cameras and lights that project infra-red rays, and use a material that reflects infra-red rays well for the markers. Fig.A.1 shows an example of an optical system.

In optical systems, a sequence of a three-dimension (3D) position is obtained for each marker. Once the placement of the cameras is determined, the system stores the positions, orientations, and internal parameters of the cameras through a calibration process. The cameras shoot an image sequence of a performance from different positions. Then the system tracks the markers on the images obtained. Markers are frequently lost from a camera because they are behind the body and the camera angle, but those markers are expected to be shot by another camera. By integrating the tracking results of all the camera images, the system can calculate the positions of the

Figure A.1: An optical motion capture system. Around the ceiling, infra-red video cameras are attached. A performer puts on markers, which are illuminate by infra-red lights on the ceiling and shot by the cameras. A performance on the floor is captured by the system.

markers with the technique of triangular surveying. For a period when a marker is not shot from any cameras, the system interpolates a trajectory of the marker. The more cameras the system has, the more robust the result is, because loss of markers can be avoided and the positions of the markers are more accurate with additional cameras. Usually at least six cameras are used for capturing one person. In this way, the system outputs trajectories of markers attached to the body. If a sufficient number of markers is attached to appropriate body positions, a sequence of whole body postures can be restored from the marker trajectories by fitting those positions with a model of human body. Fig.A.2 shows the original motion and the captured motion represented by markers.

A.2 Systems with Other Mechanisms

Currently, optical motion capture systems are the most popular systems, but motion capture systems that are based on a different mechanism are also available. They include electro-magnetic and mechanical systems, which have features different from optical systems. Here we briefly describe these two systems.

An electro-magnetic system consists of magnetic field emitters, magneto-metric

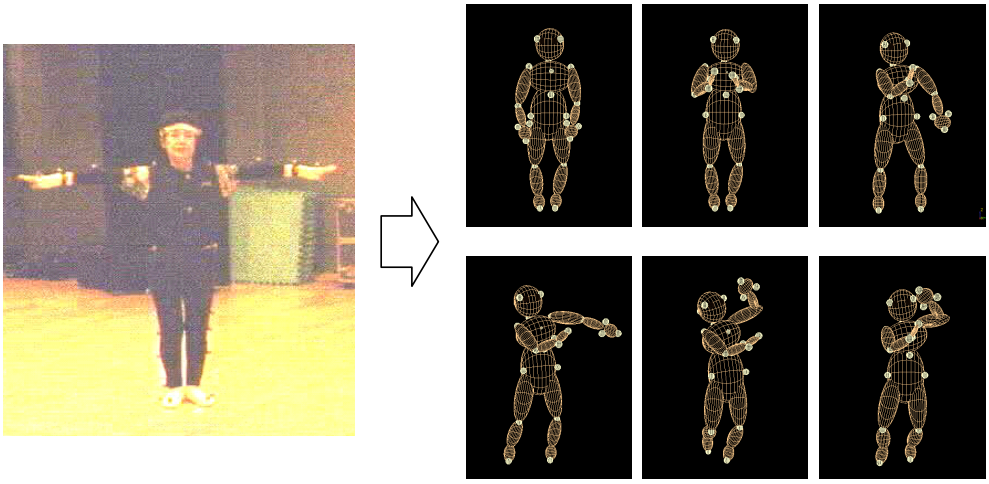


Figure A.2: Obtaining a sequence of poses.

sensors, and a radio transmitter. The emitters are placed on the edge of a stage, and they generate a magnetic field on the stage. The sensors are attached to the body like markers of the optical systems. The sensors are connected to the transmitter, which is put on the back of the performer, and outputs from the sensors are transmitted to the system by air.

The electro-magnetic systems have advantages over the optical systems. First, the attitude of a sensor is obtained as well as its position. This enables obtaining accurate poses with a smaller number of markers. Also, the sensors are not concerned with the occlusion problem, as optical systems are. These features can enhance the robustness of capturing. Furthermore, the systems consist of fewer devices than optical systems, which require a number of cameras. This makes the setup of the system easy.

On the other hand, the magnetic system also has disadvantages. First, the available area of the stage is limited by the power of the emitters. The area tends to be smaller than that of the optical systems. Also, the performer must put the transmitter on the back. The transmitter has a certain size and weight, and it is connected to the sensors by a cord. This can affect the performance. The most critical problem is that the magnetic field generated by the emitters is disturbed easily by metal objects in the environment. Although a small disturbance is calibrated, disturbances typically observed in a building of steel are too big to calibrate. This feature strongly limits the condition of the capture space.

In a mechanical-type motion capture, a performer puts an articulated frame on the body. The joints of the frame fit corresponding joints of the performer, and joint angles are captured by angle sensors embedded in the joints of the frame. The mechanical

systems are easier to use than the two other systems because they do not require other devices such as cameras or emitters and they are available at low cost. However, the current systems do not achieve the degree of accuracy of the other systems, and performances are more affected then other systems by the device on the body.

A.3 Advanced Studies

The technology of motion capture is still progressing; many studies try to achieve more advanced features.

Kurihara *et al.* [36] have developed an optical motion capture called *behavior capture system.* In their system, cameras automatically follow the object by changing their orientations. This enables a performer to perform more dynamic motions over a wider area of the stage. Their system can also output capture data in real time. This feature enables the system to use applications including real-time interaction.

Cheung *et al.* [3] have developed an optical system that does not require markers on a body. This technology removes a burden in performances, and enables the system to be used easily.

Appendix B

Interpolation Function using Third Polynomial Equations

Trajectory interpolation is used by many processes in this thesis. We use a simple interpolation function using third polynomial equations for this purpose. The function is expressed as follows:

$$\boldsymbol{f}_n \left\langle (t_1, \boldsymbol{y}_1, \dot{\boldsymbol{y}}_1), \cdots, (t_n, \boldsymbol{y}_n, \dot{\boldsymbol{y}}_n) \right\rangle (t) \qquad (n \geq 2). \tag{B.1}$$

A set of $(t_i, \boldsymbol{y}_i, \dot{\boldsymbol{y}}_i)$ indicates a point where its time, value, and velocity are t_i, $\boldsymbol{y}_i$, and $\dot{\boldsymbol{y}}_i$, respectively. The function passes all the given points, which means

$$\begin{aligned} &\forall\ i\ (1 \leq i \leq n) \\ &\boldsymbol{f}_n \langle \cdots \rangle (t_i) = \boldsymbol{y}_i \\ &\dot{\boldsymbol{f}}_n \langle \cdots \rangle (t_i) = \dot{\boldsymbol{y}}_i. \end{aligned} \tag{B.2}$$

This function is a kind of the cubic spline interpolation, but the function requires differential values of the mid points in contrast to the cubic spline.

When $\dot{\boldsymbol{y}}_i$ is omitted, $\dot{\boldsymbol{y}}_i$ is supposed to be $\mathbf{0}$. Thus, the expression

$$\boldsymbol{f}_n \left\langle (t_1, \boldsymbol{y}_1), \cdots, (t_n, \boldsymbol{y}_n) \right\rangle (t) \tag{B.3}$$

means

$$\boldsymbol{f}_n \left\langle (t_1, \boldsymbol{y}_1, \mathbf{0}), \cdots, (t_n, \boldsymbol{y}_n, \mathbf{0}) \right\rangle (t). \tag{B.4}$$

In this function, one segment between the two adjacent points is expressed by a third polynomial equation. The segment from t_i to t_{i+1} is expressed by the following

function $\boldsymbol{g}_i(t)$:

$$
\begin{aligned}
\boldsymbol{a}_0 &= \boldsymbol{y}_i \ , \qquad \boldsymbol{a}_1 = \dot{\boldsymbol{y}}_i \ , \\
\boldsymbol{a}_2 &= \frac{3}{t_{i+1}^2}(\boldsymbol{y}_{i+1} - \boldsymbol{y}_i) - \frac{2}{t_{i+1}}\dot{\boldsymbol{y}}_i - \frac{1}{t_{i+1}}\dot{\boldsymbol{y}}_{i+1} \ , \\
\boldsymbol{a}_3 &= -\frac{2}{t_{i+1}^3}(\boldsymbol{y}_{i+1} - \boldsymbol{y}_i) + \frac{1}{t_{i+1}^2}(\dot{\boldsymbol{y}}_{i+1} + \dot{\boldsymbol{y}}_i) \ , \\
\boldsymbol{g}_i(t) &= \boldsymbol{a}_0 + \boldsymbol{a}_1(t - t_i) + \boldsymbol{a}_2(t - t_i)^2 + \boldsymbol{a}_3(t - t_i)^3 \ .
\end{aligned}
\tag{B.5}
$$

Therefore $\boldsymbol{f}_n(t)$ is expressed with $\boldsymbol{g}_i(t)$ as follows:

$$
\boldsymbol{f}_n \left\langle (t_1, \boldsymbol{y}_1, \dot{\boldsymbol{y}}_1), \cdots, (t_n, \boldsymbol{y}_n, \dot{\boldsymbol{y}}_n) \right\rangle (t) = \begin{cases} \boldsymbol{g}_1(t) & (t_1 \leq t < t_2) \\ \cdots & \\ \boldsymbol{g}_i(t) & (t_i \leq t < t_{i+1}) \\ \cdots & \\ \boldsymbol{g}_{n-1}(t) & (t_{n-1} \leq t \leq t_n) \ . \end{cases}
\tag{B.6}
$$

Appendix C

Method for Achieving a Desired ZMP

Nishiwaki *et al.* [56] proposed a method that determines the movement of the center of mass (CM) of the body according to a given ZMP movement. The method is also used for modifying a given CM motion to achieve a given desired ZMP.

This method assumes a simple model of interaction between a robot and the floor. Fig.C.1 shows this model. In this model, the truck on the table corresponds to the center of mass of the robot, and the stand of the table corresponds to a sole of the robot. The body of the robot is represented by a unique mass point, and the mass point moves horizontally at the same height.

Moment at a point on the stand is calculated as follows:

$$\tau_P = Mg(x - p) - M\ddot{x}h \ , \tag{C.1}$$

where τ_P is moment at point p, M is the total mass of the robot, x is the horizontal position of the CM, and h is height of the CM.

When the table maintains balance, the stand does not rotate. That is, the point at which τ_P is zero is on the stand. In this case, the following equations are derived:

$$\begin{aligned} Mg(x - p) - M\ddot{x}h &= 0 \ , \\ p &= x - \frac{h}{g}\ddot{x} \ . \end{aligned} \tag{C.2}$$

The ZMP corresponds to the position p.

On a discrete system in which one frame has time Δt, the acceleration is written as

$$\ddot{x}_i = \frac{x_{i+1} - 2x_i + x_{i-1}}{\Delta t^2} \ , \tag{C.3}$$

where $\ddot{x}_i$ and x_i mean values at time $i\Delta t$.

From equations (C.2) and (C.3), the following equation is derived:

$$p_i = \frac{-hx_{i+1} + (2h + g\Delta t^2)x_i - hx_{i-1}}{g\Delta t^2} \ , \tag{C.4}$$

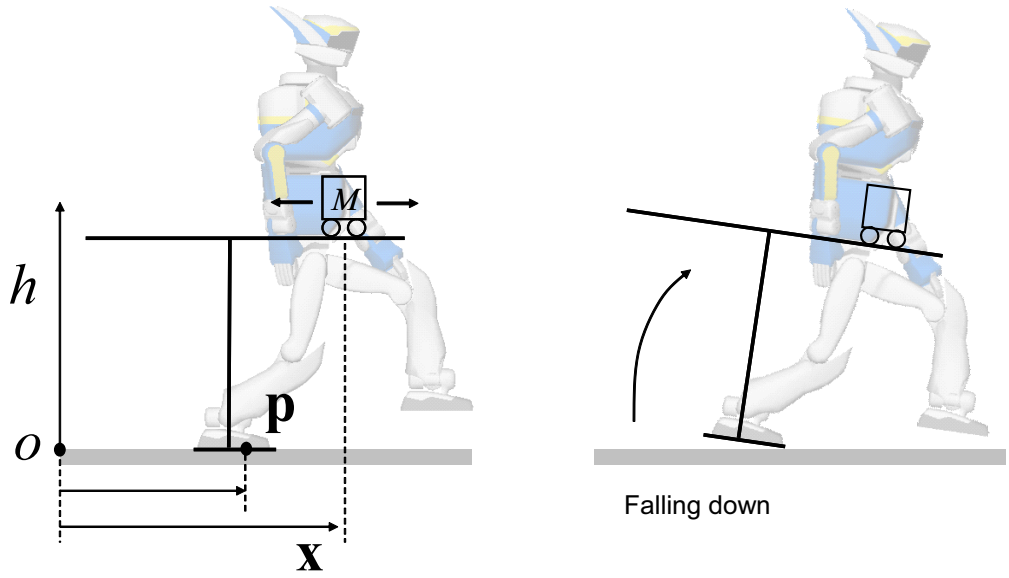


Figure C.1: Simple Model for ZMP calculation

where p_i means values of p at time $i\Delta t$.

This equation is expressed by information of three consecutive frames. These types of equations are solved as tridiagonal simultaneous linear equations [66]. x_i can be calculated mechanically.

The equation (C.4) can also be expressed as

$$\begin{aligned} \Delta p_i &= p_i - p'_i \ , \\ \Delta x_i &= x_i - x'_i \ , \\ \Delta p_i &= \frac{-h\Delta x_{i+1} + (2h + g\Delta t^2)\Delta x_i - h\Delta x_{i-1}}{g\Delta t^2} \ , \end{aligned} \tag{C.5}$$

where p'_i is the IZMP calculated from a given motion, and x'_i is the CM trajectory of a given motion. This form of equation can modify a given motion so that it achieves a given desired ZMP.

This method cannot figure out a result that completely follows the desired ZMP trajectory in one calculation because the model is an approximation of the actual body that consists of a number of body parts, and the model assumes the constraint in which the height of CM is a constant.

However, a result easily converges by iterating the calculation. This method is useful and efficient for a ZMP-based motion generation.

Appendix D

Conversion of Upper Body Motions

D.1 Overview

As described in Chapter 5, the task generation system requires motion trajectories of the upper body, which include joint angle trajectories and trajectories of the waist attitude (for roll and pitch). This data must be prepared by a process independent of leg tasks. The motions must be adapted to the constraints of the upper body motions. In this chapter, we describe a method for generating upper body motions for robots from human motions.

In contrast to leg tasks, trajectory-based conversion is employed for the upper body. That is, the conversion process directly modifies the original motion trajectories so that they are adapted to the constraints of robots.

This approach is reasonable for upper body motions because of the following factors:

- Constraints of upper body motions are not as severe as those of leg motions because the upper body does not interact with the floor directly.
- Generally upper body motions emphasize their detailed trajectories for dance expression, while leg motions emphasize the functions of supporting and moving the body rather than detailed motion trajectories.

The task generation system accepts an upper body motion as data independent of leg tasks, and the system generates leg motions that can perform the given motion of the upper body without modification. This feature of the system allows our approach to upper body motions, and the approach works well for dance performances.

The outline of the conversion process is as follows.

1. For a pose in motion, joint angles and the waist attitude (roll and pitch) of a robot are determined from captured marker positions.

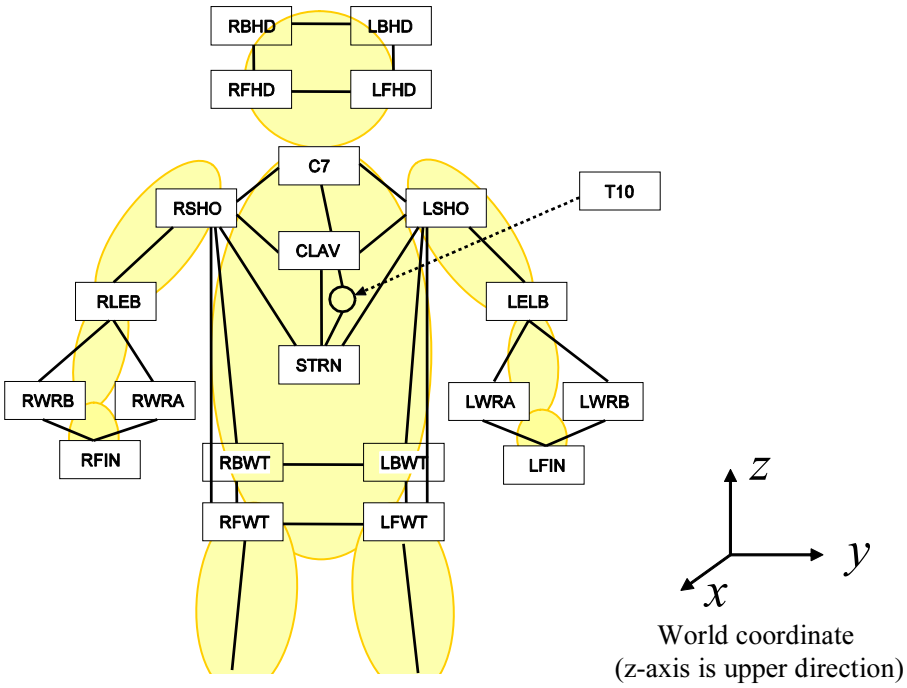


Figure D.1: Marker configuration of the human upper body. A pose is represented by 3-D positions of these markers.

2. Trajectories of the joint angles and waist attitude are obtained by applying the above process for the whole motion sequence.

3. Angular velocities of the joints are constrained so that they are under the limits of actuators.

4. Joint angles are constrained so that they are under the limits of ranges.

D.2 Motion Data

A pose of the human body is represented by the markers as shown in Fig.D.1. 3-D positions of these markers are obtained by a motion capture, and they are described by the world coordinate. For example, position of the right hand is expressed as

$$\boldsymbol{p}_{rfin} = (x_{rfin}\ y_{rfin}\ z_{rfin})^T \ . \tag{D.1}$$

Human motions are described as trajectories of these positions. A trajectory is stored as an array of discrete time sequence. For example, the trajectory of the right hand is expressed as

$$\boldsymbol{p}_{rfin}(i) = (x_{rfin}(i)\ y_{rfin}(i)\ z_{rfin}(i))^T \ , \tag{D.2}$$

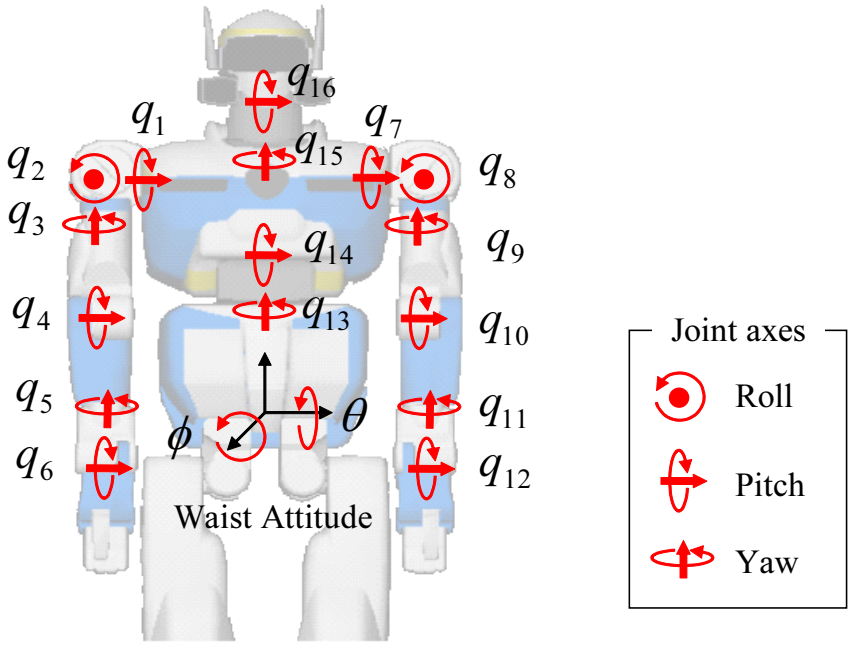


Figure D.2: Joint structure of the upper body of the robot. Each joint has one rotational axis. A pose is represented by angles of these joints. Global attitude of the upper body is determined by the roll and pitch of the waist attitude.

where i is an index of the array, which means time $i\Delta t$.

A pose of the robot is represented by angles of the joints shown in Fig.D.2. The pose is also determined by the roll and pitch of the waist attitude; these values determine the global inclination of the body.

The angle of the j-th joint is represented as q_j, and the roll and pitch of the waist are represented as θ and ϕ, respectively.

Trajectories of these values are described as $q_j(i)$, $\theta(i)$, and $\phi(i)$, which are similar to the marker trajectories.

D.3 Mapping Markers on Joints

Joint angles of the robot are determined from the human markers. This is processed by mapping the marker positions on the joint structure of the robot. In the following, we show an example of this process.

First, the centers of the four body parts are determined as follows:

$$
\begin{aligned}
\boldsymbol{c}_w &= r_w \frac{\boldsymbol{p}_{lfwt} + \boldsymbol{p}_{rfwt}}{2} + (1 - r_w) \frac{\boldsymbol{p}_{lbwt} + \boldsymbol{p}_{rbwt}}{2} \qquad (0 < r_w < 1) \ , \\
\boldsymbol{c}_c &= r_c \ \boldsymbol{p}_{strn} + (1 - r_c) \ \boldsymbol{p}_{t10} \qquad (0 < r_c < 1) \ , \\
\boldsymbol{c}_s &= \frac{\boldsymbol{p}_{lsho} + \boldsymbol{p}_{rsho}}{2} \ , \\
\boldsymbol{c}_h &= r_h \frac{\boldsymbol{p}_{lfhd} + \boldsymbol{p}_{rfhd}}{2} + (1 - r_h) \frac{\boldsymbol{p}_{lbhd} + \boldsymbol{p}_{rbhd}}{2} \qquad (0 < r_h < 1) \ ,
\end{aligned}
\tag{D.3}
$$

where $\boldsymbol{c}_w$, $\boldsymbol{c}_c$, $\boldsymbol{c}_s$ and $\boldsymbol{c}_h$ are the centers of the waist, chest, shoulder, and head, respectively, and r_w, r_c, and r_h are ratios that determine the centers of each part.

The attitude of the waist is determined from these values:

$$
\boldsymbol{R}_w = \boldsymbol{R}_{cross}((\boldsymbol{p}_{lfwt} + \boldsymbol{p}_{lbwt}) - (\boldsymbol{p}_{rfwt} + \boldsymbol{p}_{rbwt}), \ \boldsymbol{c}_s - \boldsymbol{c}_w) \ , \tag{D.4}
$$

where $\boldsymbol{R}_w$ is the attitude matrix of the waist. $\boldsymbol{R}_{cross}(\boldsymbol{y}, \ \boldsymbol{z})$ is a matrix that is defined as

$$
\begin{aligned}
\boldsymbol{e}_x &= |\boldsymbol{y} \times \boldsymbol{z}| \ , \\
\boldsymbol{e}_y &= |\boldsymbol{y}| \ , \\
\boldsymbol{e}_z &= \boldsymbol{e}_x \times \boldsymbol{e}_y \ , \\
\boldsymbol{R}_{cross}(\boldsymbol{y}, \ \boldsymbol{z}) &= (\boldsymbol{e}_x \ \boldsymbol{e}_y \ \boldsymbol{e}_z) \ .
\end{aligned}
\tag{D.5}
$$

In a way similar to the waist, the attitude of the chest and head are determined as

$$
\begin{aligned}
\boldsymbol{R}_c &= \boldsymbol{R}_{cross}(\boldsymbol{p}_{lsho} - \boldsymbol{p}_{rsho}, \ \boldsymbol{c}_s - \boldsymbol{c}_c) \ , \\
\boldsymbol{R}_h &= \boldsymbol{R}_{cross}((\boldsymbol{p}_{lfhd} + \boldsymbol{p}_{lbhd}) - (\boldsymbol{p}_{rfhd} + \boldsymbol{p}_{rbhd}), \ \boldsymbol{c}_h - \boldsymbol{c}_s) \ ,
\end{aligned}
\tag{D.6}
$$

where $\boldsymbol{R}_c$ and $\boldsymbol{R}_h$ are the attitude matrices of the chest and head, respectively.

Now we define a function that extracts angles of three-axis rotation from a matrix as follows:

$$
\begin{aligned}
\boldsymbol{R} &= (\boldsymbol{e}_x \ \boldsymbol{e}_y \ \boldsymbol{e}_z) \ , \\
f_\psi(\boldsymbol{R}) &= atan2(\boldsymbol{e}_{x \cdot y}, \ \boldsymbol{e}_{x \cdot x}) \ , \\
\boldsymbol{R}' &= (\boldsymbol{e}'_x \ \boldsymbol{e}'_y \ \boldsymbol{e}'_z) \\
&= \boldsymbol{R}_z(-f_\psi(\boldsymbol{R})) \ , \\
f_\theta(\boldsymbol{R}) &= atan2(\boldsymbol{e}'_{z \cdot x}, \ \boldsymbol{e}'_{z \cdot z}) \ , \\
\boldsymbol{R}'' &= (\boldsymbol{e}''_x \ \boldsymbol{e}''_y \ \boldsymbol{e}''_z) \\
&= \boldsymbol{R}_y(-f_\phi(\boldsymbol{R})) \ , \\
f_\phi(\boldsymbol{R}) &= atan2(\boldsymbol{e}''_{y \cdot z}, \ \boldsymbol{e}''_{y \cdot y}) \ ,
\end{aligned}
\tag{D.7}
$$

where $\boldsymbol{R}_y(\theta)$ and $\boldsymbol{R}_z(\theta)$ are the rotation matrices that represent rotations around y-axis and z-axis, respectively. Functions $f_\phi(\boldsymbol{R})$, $f_\theta(\boldsymbol{R})$ and $f_\psi(\boldsymbol{R})$ represents the roll, pitch, and yaw angles of the rotation matrix $\boldsymbol{R}$, respectively.

The roll and pitch of the waist are determined as

$$\begin{aligned} \phi &= f_\phi(\boldsymbol{R}_w) \ , \\ \theta &= f_\theta(\boldsymbol{R}_w) \ . \end{aligned} \tag{D.8}$$

Angles of the joints between the waist and chest are determined as follows:

$$\begin{aligned} {}^w\boldsymbol{R}_c &= \boldsymbol{R}_w^T \boldsymbol{R}_c \ , \\ q_{13} &= f_\psi({}^w\boldsymbol{R}_c) \ , \\ q_{14} &= f_\theta({}^w\boldsymbol{R}_c) \ . \end{aligned} \tag{D.9}$$

Angles of the neck joints are determined as follows:

$$\begin{aligned} {}^c\boldsymbol{R}_h &= \boldsymbol{R}_c^T \boldsymbol{R}_h \ , \\ q_{15} &= f_\psi({}^c\boldsymbol{R}_h) \ , \\ q_{16} &= f_\theta({}^c\boldsymbol{R}_h) \ . \end{aligned} \tag{D.10}$$

The roll and pitch of the right shoulder are determined as follows:

$$\begin{aligned} \boldsymbol{a} &= \boldsymbol{p}_{relb} - \boldsymbol{p}_{rsho} \ , \\ \boldsymbol{a}_0 &= \boldsymbol{R}_c^T \boldsymbol{a} \ , \\ q_1 &= atan2(-a_0 x, \ -a_0 z) \ , \\ \boldsymbol{a}_1 &= \boldsymbol{R}_y(-q_1)\boldsymbol{a}_0 \ , \\ q_2 &= atan2(a_1 y, \ -a_1 z) \ . \end{aligned} \tag{D.11}$$

The yaw of the shoulder and the elbow joint are determined as follows:

$$\begin{aligned} \boldsymbol{R}_2 &= \boldsymbol{R}_c \boldsymbol{R}_y(q_1) \boldsymbol{R}_x(q_2) \ , \\ \boldsymbol{b} &= \left| \frac{\boldsymbol{p}_{rwra} + \boldsymbol{p}_{rwrb}}{2} - \boldsymbol{p}_{relb} \right| \ , \\ \boldsymbol{b}_2 &= \boldsymbol{R}_2^T \ \boldsymbol{b} \ , \\ q_3' &= atan2(-b_{2y}, \ b_{2x}) \ , \\ \boldsymbol{b}_3' &= \boldsymbol{R}_z(-q_3') \ \boldsymbol{b}_2 \ , \\ \alpha &= acos(-b_{3z}') \ , \\ q_3 &= \begin{cases} q_3' & \alpha \geq \alpha_0 \\ prev(q_3) & \alpha < \alpha_0 \ , \end{cases} \\ \boldsymbol{b}_3 &= \boldsymbol{R}_z(-q_3) \ \boldsymbol{b}_2 \ , \\ q_4 &= atan2(b_{3x}, \ -b_{3z}) \ , \end{aligned} \tag{D.12}$$

where α_0 is a threshold value that determines the boundary of the singular point, and $prev(q_i)$ is a function that outputs the previous angle of the joint.

Angles of the wrist joints are determined as follows:

$$
\begin{aligned}
\boldsymbol{R}_4 &= \boldsymbol{R}_2 \boldsymbol{R}_z(q_3) \boldsymbol{R}_y(q_4) \ , \\
\boldsymbol{s} &= \boldsymbol{p}_{rwra} - \boldsymbol{p}_{rwrb} \ , \\
\boldsymbol{s}_4 &= \boldsymbol{R}_4^T \ \boldsymbol{s} \ , \\
q_5 &= atan2(s_{4y}, s_{4x}) \ , \\
\boldsymbol{R}_5 &= \boldsymbol{R}_4 \boldsymbol{R}_z(q_5) \ , \\
\boldsymbol{u} &= \boldsymbol{p}_{rfin} - \frac{\boldsymbol{p}_{rwra} + \boldsymbol{p}_{rwrb}}{2} \ , \\
\boldsymbol{u}_5 &= \boldsymbol{R}_5^T \ \boldsymbol{u} \ , \\
q_6 &= atan2(-u_{5x}, \ -u_{5z}) \ .
\end{aligned}
\tag{D.13}
$$

Joint angles of the left arm are also determined in a similar way.

In this way, joint angles of the upper body are obtained from captured markers. By applying this process to the whole motion sequence, joint angle trajectories of the upper body are obtained.

D.4 Constraining Angular Velocity

In joint angle trajectories obtained by the above process, angular velocities may become an impossible value over the constraints of the robot.

In addition to the problem of each joint trajectory, the motion may imply poses that are in the neighborhood of singular points. At poses near the singular point, valid moving patterns are limited and movement may be locked. Such motions cannot be performed by the actual robot. This problem can be considered as a problem of angular velocity because a non-continuous curve of velocity around a locked frame can be considered to be a high gradient curve on a discrete system.

Angular velocity is constrained by a method proposed by Pollard *et al.* [65]. This method preserves timings of the global vibrations well.

The basic idea of this method is that a new angle trajectory is created by following the original trajectory under the limit of angular velocity. The process is similar to the PD control method.

First, a new trajectory is created by the following equations:

$$
\begin{aligned}
\dot{q}_j(i) &= q_j(i) - q_j(i-1) \ , \\
\ddot{q}_{Fj}(i+1) &= 2\sqrt{K_s}\{\dot{q}_j(i) - \dot{q}_{Fj}(i)\} + K_s\{q_j(i) - q_{Fj}(i)\} \ , \\
\dot{q}_{Fj}(i+1) &= max(\ \omega_{Lj}, min(\omega_{Uj}, \ \dot{q}_{Fj}(i) + \ddot{q}_{Fj}(i+1)\ ) \ , \\
q_{Fj}(i+1) &= q_{Fj}(i) + \dot{q}_{Fj}(i+1) \ ,
\end{aligned}
\tag{D.14}
$$

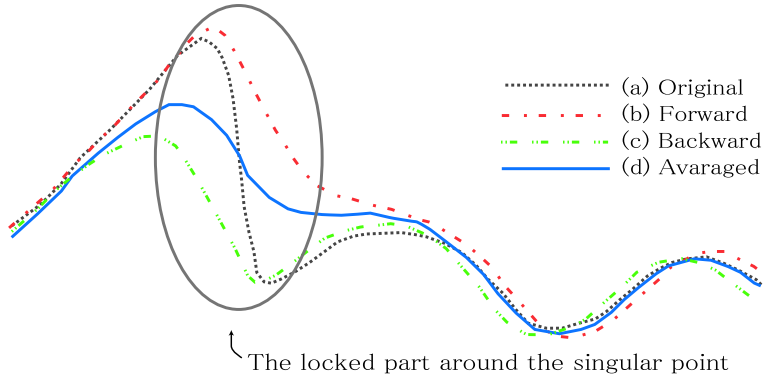


Figure D.3: Trajectories in the process of constraining angular velocity

where $q_j(i)$ is the original value at time frame i, $q_{Fj}(i)$ is new joint angle, j is a joint number, and ω_{Lj} and ω_{Uj} are the lower and upper velocity limits of the j-th joint. K_s is the parameter which controls stiffness of motion.

The result becomes a trajectory similar to the original one within the limit. However, it must delay from the original one in ranges over the velocity limit. (Fig.D.3-(b)).

To solve this problem, another trajectory is created by the inverse process of the above equations from end to start as follows:

$$\begin{aligned}
\dot{q}_j(i) &= q_j(i) - q_j(i+1) \ , \\
\ddot{q}_{Bj}(i-1) &= 2\sqrt{K_s}\{\dot{q}_j(i) - \dot{q_{Bj}}(i)\} + K_s\{q_j(i) - q_{Bj}(i)\} \ , \\
\dot{q}_{Bj}(i-1) &= max(\omega_{Lj}, min(\omega_{Uj}, \dot{q}_{Bj}(i) + \ddot{q}_{Bj}(i-1)) \ , \\
q_{Bj}(i-1) &= q_{Bj}(i) + \dot{q}_{Bj}(i-1) \ .
\end{aligned} \tag{D.15}$$

The result becomes a trajectory of the future instance compared with the original one (Fig.D.3-(c)).

Finally, both trajectories are averaged to obtain a trajectory whose shape overlaps the original one as follows:

$$q_{Vj}(i) = \frac{q_{Fj}(i) + q_{Bj}(i)}{2} \ . \tag{D.16}$$

Fig.D.3-(d) shows this trajectory. In this final trajectory, the rhythm of curve approximately matches the original one, and values are within the limit, at the same time. The locked parts around singular points in the initial motion are also changed into a smooth curve. Fig.D.3 shows the initial motion that must lock on the actual robot, and the motion after the filter, which can be performed on the actual robot.

D.5 Constraining Angle

Joint angles are basically constrained by the following function that scales down values of the trajectories:

$$f_{i_0,i_1}(x) = \begin{cases} a_u(x-c_u)^3 + b_u(x-c_u)^2 + (x-c_u) + c_u & c_u < x \\ x_i & c_l \le x \le c_u \\ a_l(x-c_l)^3 + b_l(x-c_l)^2 + (x-c_l) + c_l & x < c_l \ . \end{cases} \tag{D.17}$$

where i_0 and i_1 determine a range to which this function is applied. Coefficients of these equations are calculated as follows. First, the range of values is determined as

$$\begin{aligned} x_{max} &= max\{x_i \mid i_0 \le i \le i_1\} \ , \\ u &= \begin{cases} x_{max} & x_{max} > U \\ U & x_{max} \le U \ , \end{cases} \\ x_{min} &= min\{x_i \mid i_0 \le i \le i_1\} \ , \\ l &= \begin{cases} x_{min} & x_{min} < L \\ L & x_{min} \ge L \ , \end{cases} \end{aligned} \tag{D.18}$$

where u and l are the upper and lower bounds of the range, and U and L are the original upper and lower limits.

Then the coefficients c_u and c_l are determined as follows:

$$\begin{aligned} c_{u0} &= \frac{U-ku}{1-k} \ , \\ c_{l0} &= \frac{L-kl}{1-k} \ , \\ &(0 < k < 1) \ , \\ c_m &= \frac{Lu-Ul}{u-l+L-U} \ , \\ c_u &= \begin{cases} c_{u0} & c_l \le c_u \\ c_m & c_l > c_u \ , \end{cases} \\ c_l &= \begin{cases} c_{l0} & c_l \le c_u \\ c_m & c_l > c_u \ , \end{cases} \end{aligned} \tag{D.19}$$

where k is a parameter that determines the average scaling ratio.

Coefficients a_u and b_u are determined as follows. If $u \neq c_u$, the following values

are employed:

$$
\begin{aligned}
r_u &= \alpha \frac{U - c_u}{u - c_u} \qquad (0 < \alpha < 1) \ , \\
d_u &= u - c_u \ , \\
b_u &= \frac{d_u(1 - r_u) - 3(u - U)}{d_u^2} \ , \\
a_u &= \frac{-2d_u b_u - 1 + r_u}{3d_u^2} \ ,
\end{aligned}
\tag{D.20}
$$

where α is a parameter that determines the scaling ratio at the upper and lower bounds. If $u = c_u$, the following values are employed:

$$
\begin{aligned}
a_u &= 0 \ , \\
b_u &= 0.
\end{aligned}
\tag{D.21}
$$

In a similar way, coefficients a_l and b_l are determined. If $l \neq c_l$, the following values are employed:

$$
\begin{aligned}
r_l &= \alpha \frac{L - c_l}{l - c_l} \ , \\
d_l &= l - c_l \ , \\
b_l &= \frac{d_l(1 - r_l) - 3(l - L)}{d_l^2} \ , \\
a_l &= \frac{-2d_l b_l - 1 + r_l}{3d_l^2}.
\end{aligned}
\tag{D.22}
$$

If $l = c_l$, the following values are employed:

$$
\begin{aligned}
a_l &= 0 \ , \\
b_l &= 0.
\end{aligned}
\tag{D.23}
$$

Although function (D.17) scales down values in local space, it is desired that the scaling down is applied to temporary local parts. For this purpose, a whole trajectory is constrained by applying functions to a number of local time segments. The j-th segment is determined to be the minimum segment from a_j to b_j that satisfies the following conditions in the definition of $f_{a_j,b_j}(x)$:

$$
\begin{aligned}
u > U \ \ or \ \ l < L \ , \\
\forall \ i \ (a_j \leq i \leq b_j) \quad x_i \geq c_u \ \ or \ \ x_i \leq c_l \ , \\
c_l < x_{a_j - 1} < c_u \ , \\
c_l < x_{b_j + 1} < c_u \ .
\end{aligned}
\tag{D.24}
$$

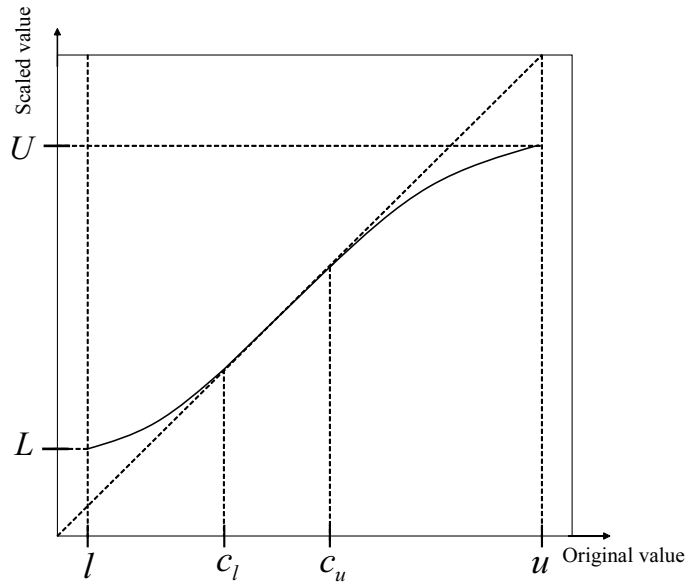


Figure D.4: Scaling function $f_{i_0,i_1}(x)$. The horizontal axis represents the original values, and the vertical axis represents the scaled values. Only local ranges near the limits are scaled in this function.

Finally, constrained values are obtained by n functions for n segments as follows:

$$x'_i = \begin{cases} f_{a_1,b_1}(x_i) & a_1 \leq i \leq b_1 \\ \dots & \\ f_{a_j,b_j}(x_i) & a_j \leq i \leq b_j \\ \dots & \\ f_{a_n,b_n}(x_i) & a_n \leq i \leq b_n \\ x_i & otherwise\ , \end{cases} \tag{D.25}$$

where x'_i is a constrained value.

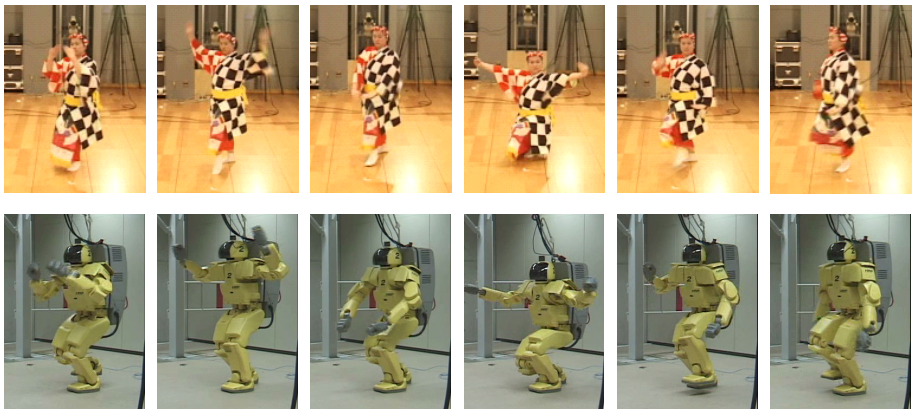
Appendix E

Jongara-bushi Dance by HRP-1S

In Chapter 6, we presented the Aizu-bandaisan dance performed by HRP-2. In addition to this experiment, we had also carried out the experiment using another dance and another robot before we completed the development of the methods described in this thesis.

In the experiment, a humanoid robot HRP-1S performed a dance called "Tsugaru Jongara-bushi." HRP-1S could not perform a motion at the same tempo of the original dance music because the dance include motions that are too fast for HRP-1S to follow. Consequently, the tempo was decreased to 40% of the original tempo. Also, the impact when a sole lands on the floor was considerable in this experiment because the method for reducing the impact was not developed in this time.

Although the reproduced performance was not complete in these respects, HRP-1S performed the motion without falling down and expressed the characteristics of the original dance motion. Since this result partly indicates the generality of our method, we introduce the pictures of the result for reference.



"Tsugaru Jongara-bushi". The performance by a human dancer and the performance reproduced by HRP-1S.

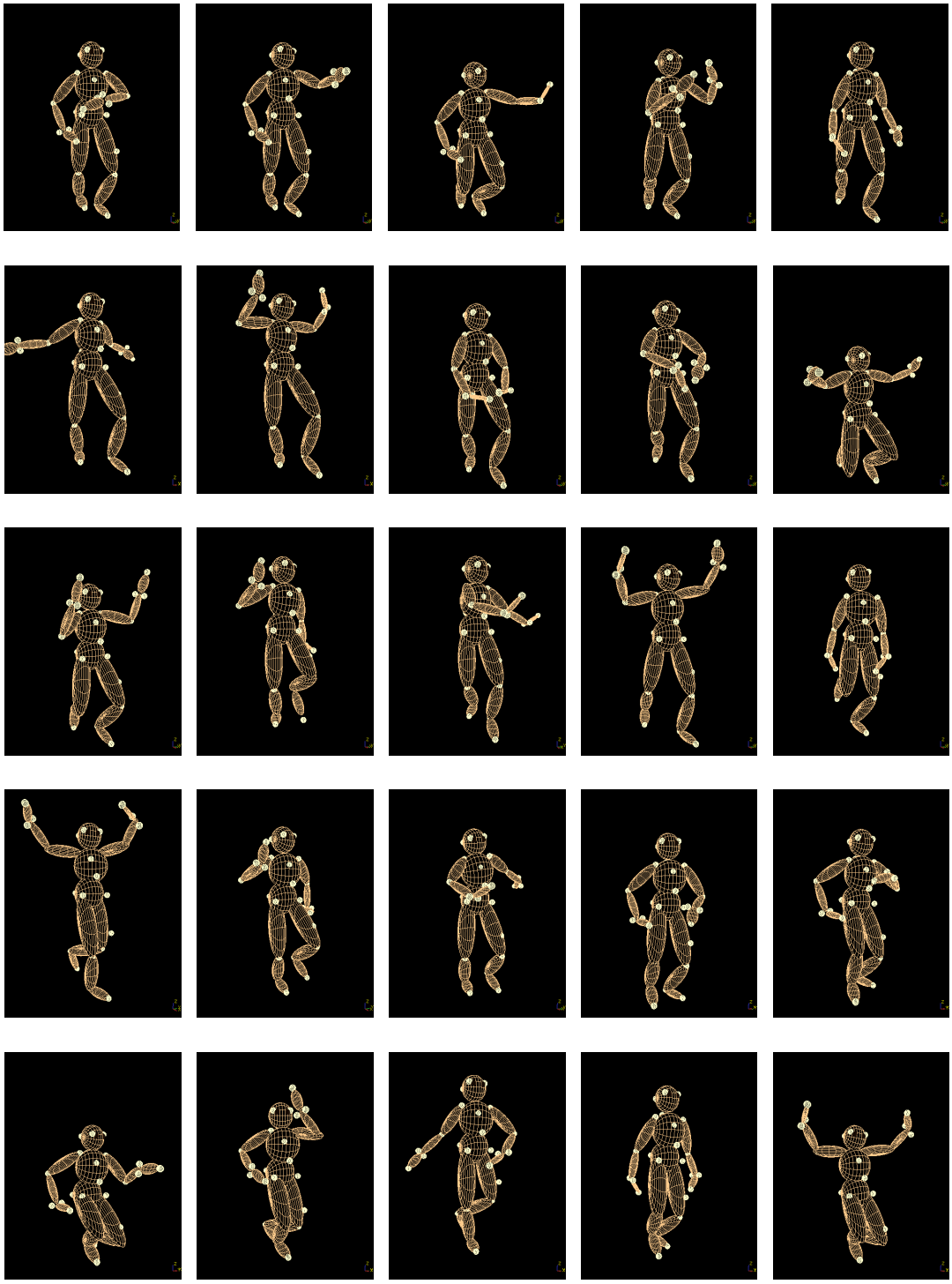


Figure E.1: Captured human motion of Jongara-bush

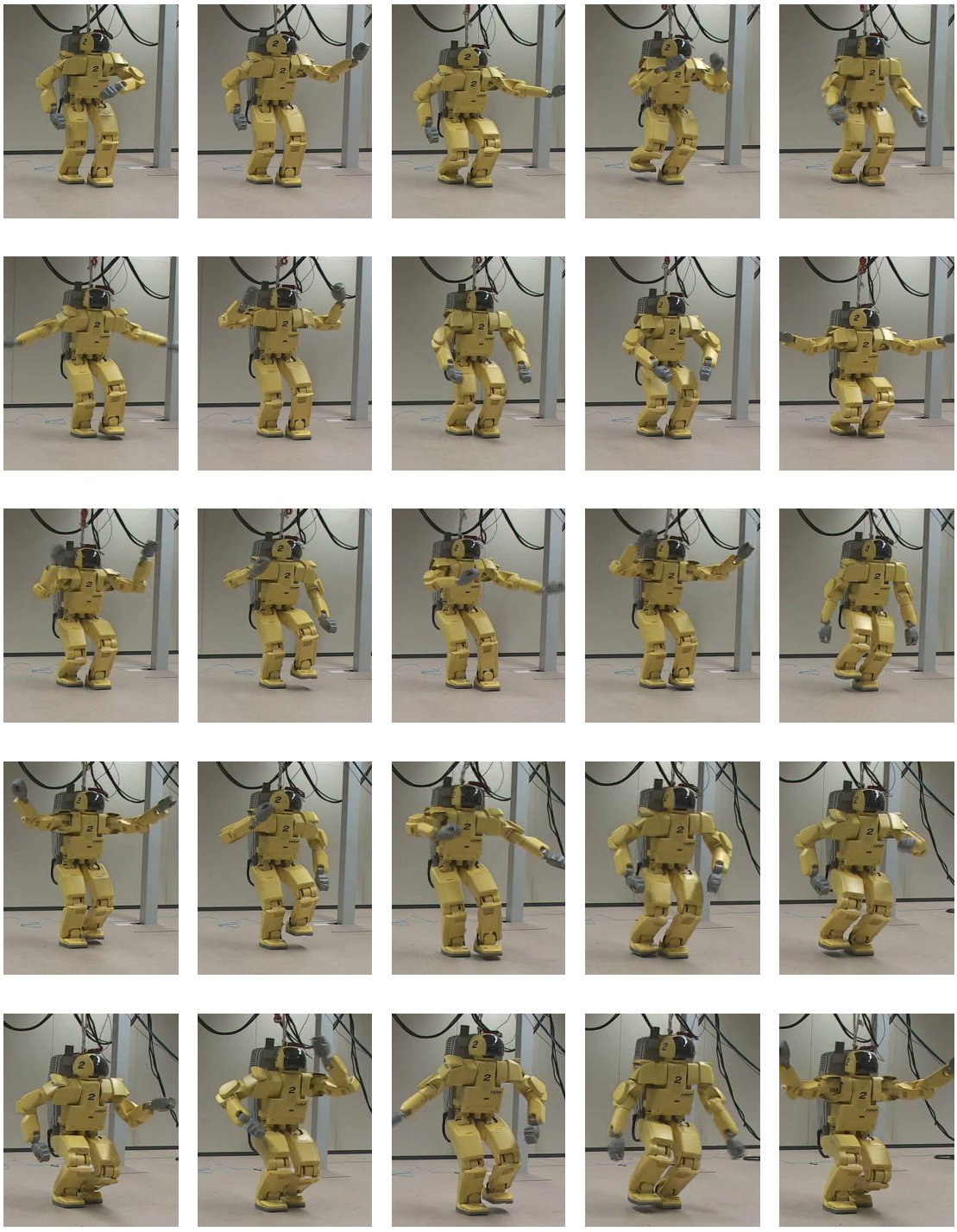


Figure E.2: Jongara-bushi performed by HRP-1S

References

[1] http://www.warabi.or.jp [in Japanse].

[2] Keni Bernardin, Koichi Ogawara, Katsushi Ikeuchi, and Ruediger Dillmann. Sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models. *IEEE Transactions on Robotics*, Vol. 21, No. 1, pp. 47–57, February 2005.

[3] K.M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time: Part ii: Applications to human modeling and markerless motion tracking. *International Journal of Computer Vision*, Vol. 63, No. 3, pp. 225–245, August 2005.

[4] Christopher G.Atkeson, Joshua G.Hale, Frank Pollick, Marcia Riley, Shinya Kotosaka, Stefan Schaal, Tomohiro Shibata, Gaurav Tevatia, Ales Ude, Sethu Vijayakumar, and Mitsuo Kawato. Using humanoid robots to study human behavior. *IEEE Intelligent Systems*, Vol. 15, No. 4, pp. 46–56, 2000.

[5] Michael Gleicher. Retargetting motion to new characters. In *Proceedings of SIGGRAPH 98*, pp. 33–42, 1998.

[6] Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka. The development of honda humanoid robot. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pp. 1321–1326, Leuven, Belgium, May 1998.

[7] Masato Hirose, Toru Takenaka, Hiroshi Gomi, and Nobuaki Ozawa. Humanoid robot. *Journal of the Robotics Society of Japan*, Vol. 15, No. 7, pp. 983–985, 1997. [in Japanese].

[8] Hirohisa Hirukawa, Fumio Kanehiro, Shuji Kajita, Kazuhito Yokoi, Kenji Kaneko, Kiyoshi Fujiwara, Kazuo Tanie, and Takashi Nagasaki. Dynamics simulation of humanoid robots based on a spring-damper model. In *Proceedings of 19th Annual Conference of the Rbotics Society of Japan*, pp. 5–6, Japan, September 2001. [in Japanese].

[9] Katsushi Ikeuchi and Takashi Suehiro. Toward an assembly plan from observation, Part I: Task recognition with polyhedral objects. *IEEE Transaction on Robotics and Automation*, Vol. 10, No. 3, pp. 368–385, Junuary 1994.

[10] Ryojun Ikeura and Hikaru Inooka. Dancing robot. *Journal of the Robotics Society of Japan*, Vol. 14, No. 2, pp. 200–203, 1996. [in Japanese].

[11] Ryojun Ikeura, Hisashi Nakazato, and Hikaur Inooka. Motion planning for dancing robots based on human's dancing motion. *Journal of the Robotics Society of Japan*, Vol. 15, No. 6, pp. 117–123, 1997. [in Japanese].

[12] Tetsunari Inamura, Yoshihiko Nakamura, and Iwaki Toshima. Embodied symbol emergence based on mimesis theory. *International Journal of Robotics Research*, Vol. 23, No. 4, pp. 363–377, 2004.

[13] Hirochika Inoue, Susumu Tachi, Kazuo Tanie, Kazuhito Yokoi, Shigeoki Hirai, Hirohisa Hirukawa, Kazuo Hirai, Shigeto Nakayama, Kazuya Sawada, Takashi Nishiyama, Osamu Miki, Toshiyuki Itoko, Hajimu Inaba, and Masako Sudo. HRP:humanoid robotics project of MITI. In *Proceedings of the First IEEE-RAS International Conference on Humanoid Robots (Humanoids 2000)*, Massachusetts, September 2000.

[14] Satoshi Kagami, Koichi Nishiwaki, James J. Kuffner Jr., Yasuo Kuniyoshi, Masayuki Inaba, and Hirochika Inoue. Online 3d vision, motion planning and bipedal locomotion control coupling system of humanoid robot: H7. In *Proceedings of International Conference on Intelligent Robots and Systems*, pp. 2557–2562, EPFL, Lausanne, Swizerland, September 2002.

[15] Satoshi Kagami, Koichi Nishiwaki, James J. Kuffner, Tomomichi Sugihara, Masayuki Inaba, and Hirochika Inoue. Design and implementation of humanoid H6 and its application to remote operation. *In Experimental Robotics VII, Lecture Notes in Control and Information Sciences 271*, pp. 41–50, 2001.

[16] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Kazuhito Yokoi, and Hirohisa Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pp. 1620–1626, Taipei, Taiwan, September 2003.

[17] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kazuhito Yokoi, and Hirohisa Hirukawa. A realtime pattern generator for biped walking. In *Proceedings of International Conference on Robotics and Automation*, pp. 31–37, Washington DC, April 2002.

[18] Shuuji Kajita, Mitsuharu Morisawa, Kensuke Harada, Kenji Kaneko, Fumio Kanehiro, Kiyoshi Fujiwara, and Hirohisa Hirukawa. Adaptive walking pattern gener-

ation by modifying future ZMP. In *Proceedings of RSJ 23th Annual Conference of the Robotics Society of Japan*, Tokyo, September 2005. [in Japanese].

[19] Shuuji Kajita, Takashi Nagasaki, Kenji Kaneko, Kazuhito Yokoi, and Kazuo Tanie. A hop towards running humanoid biped. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pp. 629–635, New Orleans, LA, April 2004.

[20] Shuuji Kajita, Takashi Nagasaki, Kazuhito Yokoi, Kenji Kaneko, and Kazuo Tanie. Running pattern generation for a humanoid robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2755–2761, Washington DC, April 2002.

[21] Noriyuki Kanehira, Toshikazu Kawasaki, Shigehiko Ohta, Takakatsu Isozumi, Tadahiro Kawada, Fumio Kanehiro, Shuuji Kajita, and Kenji Kaneko. Design and experiments of advanced leg module (HRP-2L) for humanoid robot (HRP-2) development. In *Proceedings of International Conference on Intelligent Robots and Systems*, pp. 2455–2460, EPFL, Lausanne, Swizerland, October 2002.

[22] F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, S. Kajita, K. Yokoi, H. Hirukawa, K. Akachi, and T. Isozumi. The first humanoid robot that has the same size as a human and that can lie down and get up. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, 2003.

[23] Fumio Kanehiro, Kiyoshi Fujiwara, Shuuji Kajita, Kazuhito Yokoi, Kenji Kaneko, Hirohisa Hirukawa, Yoshihiko Nakamura, and Katsu Yamane. Open architecture humanoid robotics platform. In *Proceedings of the IEEE 2002 International Conference on Robotics and Automation*, pp. 24–30, Washington DC, May 2002.

[24] Fumio Kanehiro, Natsuki Miyata, Shuuji Kajita, Kiyoshi Fujiwara, Hirohisa Hirukawa, Yoshihiko Nakamura, Katsu Yamane, Ichitaro Kohara, Yuichiro Kawamura, and Yoshiyuki Sankai. Virtual humanoid robot platform to develop controllers of real humanoid robots without porting. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1093–1099, Maui, Hawaii, October 2001.

[25] Kanji Kaneko, Fumio Kanehiro, Shuuji Kajita, Mitsuharu Morisawa, Kiyoshi Fujiwara, Kensuke Harada, and Hirohisa Hirukawa. Slip observer for walking on a low friction floor. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1457–1463, Edmonton, Canada, August 2005.

[26] Kenji Kaneko, Fumio Kanehiro, Shuuji Kajita, Hirohisa Hirukawa, Toshikazu Kawasaki, Masaru Hirata, Kazuhiko Akachi, and Takakatsu Isozumi. Humanoid robot HRP-2. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pp. 1083–1090, New Orleans, LA, April 2004.

[27] Kenji Kaneko, Fumio Kanehiro, Shuuji Kajita, Kazuhito Yokoi, Kiyoshi Fujiwara, Kensuke Harada, and Hirohisa Hirukawa. Stabilizer for biped walking. In *Proceedings of the 21st Annual Conference of The Robotics Society of Japan*, p. 3A34, Tokyo, Japan, September 2003. [in Japanese].

[28] Kenji Kaneko, Fumio Kanehiro, Shuuji Kajita, Kazuhiko Yokoyama, Kazuhiko Akachi, Toshikazu Kawasaki, Shigehiko Ota, and Takakatsu Isozumi. Design of prototype humanoid robotics platform for HRP. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2431–2436, EPFL, Lausanne, Swizerland, October 2002.

[29] Sing Bing Kang and Katsushi Ikeuchi. Toward automatic robot instruction from perception–recognizing a grasp from observation. *IEEE Transaction on Robotics and Automation*, Vol. 9, No. 4, pp. 432–443, Augutst 1993.

[30] Sing Bing Kang and Katsushi Ikeuchi. Toward automatic robot instruction from perception–mapping human grasps to manipulator grasps. *IEEE Transaction on Robotics and Automation*, Vol. 13, No. 1, pp. 81–95, Feburary 1997.

[31] Ichiro Kato. Development of WABOT-1. In *Biomechanism 2*, pp. 173–214. The University of Tokyo Press, 1973. [in Japanese].

[32] Kazuhiro Kosuge, Tomohiro Hayashi, Yasuhisa Hirata, and Ryosuke Tobiyama. Dance partner robot -Ms DancerR-. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3459–3464, Las Vegas, Nevada, October 2003.

[33] Lucas Kovar, Michael Gleicher, and Frederic Pighin. Motion graphs. In *Proceedings of SIGGRAPH 2002*, pp. 473–482, 2002.

[34] Yasuo Kuniyoshi, Masayuki Inaba, and Hirochika Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transaction on Robotics and Automation*, Vol. 10, No. 6, pp. 799–822, December 1994.

[35] Yasuo Kuniyoshi, Yoshiyuki Ohmura, and Koji Terada adn Akihiko Nagakubo. Dynamic roll-and-rise motion by an adult-size humanoid robot. *Journal of Robot Sciety of Japan*, Vol. 23, No. 6, pp. 706–717, 2005. [in Japanese].

[36] Kazutaka Kurihara, Shin'ichiro Hoshino, Katsu Yamane, and Yoshihiko Nakamura. Optical motion capture system with pan-tilt camera tracking and realtime data processing. In *Proceedings of International Conference on Robotics and Automation*, pp. 1241–1248, Washington DC, April 2002.

[37] Yoshihiro Kuroki, Bill Blank, Tatsuo Mikami, Patric Mayeux, Atsushi Miyamoto, Robert Playter, Ken'ichiro Nagasaka, Marc Raibert, Masakuni Nagano, and Jin'ichi Yamaguchi. Motion creating system for a small biped entertainment robot. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1394–1399, Las Vegas, Nevada, October 2003.

[38] Yoshihiro Kuroki, Masahiro Fujita, Tatsuo Ishida, Ken'ichiro Nagasaka, and Jun'ichi Yamaguchi. A small biped entertainment robot exploring attractive applications. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pp. 471–476, Taipei, Taiwan, September 2003.

[39] Jehee Lee, Jinxiang Chai, Paul S. A. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. Adapting simulated behaviors for new characters. In *In SIGGRAPH 97 Proceedings*, pp. 153–162. ACM SIGGRAPH, Addison Wesley.

[40] Jehee Lee, Jinxiang Chai, Paul S. A. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics*, Vol. 21, No. 3, pp. 491–500, 2002.

[41] Jun Miura and Katushi Ikeuchi. Task-oriented generation of visual sensing strategies in assembly tasks. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 2, pp. 126–138, February 1998.

[42] Ikuo Mizuuchi, Shigenori Yoshida, Masayuki Inaba, and Hirochika Inoue. The development and control of a flexible-spine for a human-form robot. *Advanced Robotics*, Vol. 17, No. 2, pp. 179–196, 2003.

[43] Ikuo Mizuuchi, Tomoaki Yoshikai, Yuto Nakanishi, and Masayuki Inaba. A reinforceable-muscle flexible-spine humanoid "Kenji". In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 692–697, Edmonton, Canada, August 2005.

[44] Takuma Morita, Jun Takamatsu, Koichi Ogawara, Hiroshi Kimura, and Katsushi Ikeuchi. Knot planning from observation. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pp. 3887–3892, Taipei, Taiwan, September 2003.

[45] Akihiko Nagakubo, Yasuo Kuniyoshi, and Gorgon Cheng. The ETL-humanoid system – a high-performance full body humanoid system for versatile real-world interaction. *Advanced Robotics*, Vol. 17, No. 2, pp. 149–164, 2003.

[46] Ken'ichiro Nagasaka, Masayuki Inaba, and Hirochika Inoue. Stablization of dynamic walk on a humanoid using torso position compliance control. In *In Proceedings of 17th Annual Conference on Robotics Society of Japan*, pp. 1193–1194, 1999. [in Japanese].

[47] Ken'ichiro Nagasaka, Hirochika Inoue, and Masayuki Inaba. Dynamic walking pattern generation for a humanoid robot based on optimal gradient method. In *In Proceedings of the 1999 IEEE Systems, Man, and Cybernetics Conference (SMC'99)*, pp. VI908–VI913, October.

[48] Ken'ichiro Nagasaka, Atsushi Konno, Koichi Nishiwaki, Tomonobu Kitagawa, Tomomichi Sugihara, Masayuki Inaba, and Hirochika Inoue. Development of dynamic humanoid 'H5'. In *Proceedings of 16th Annual Conference of the Rbotics Society of Japan*, pp. 837–838, Hokkaido, Japan, September 1998. [in Japanese].

[49] Ken'ichiro Nagasaka, Yoshihiro Kuroki, Shin'ya Suzuki, Yoshihiro Itoh, and Jun'ichi Yamaguchi. Integrated motion control for walking, jumping and running on a small bipedal entertainment robot. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pp. 3189–3194, New Orleans, LA, April 2004.

[50] Yoshihiko Nakamura and Katsu Yamane. Dynamics computation of structure-varying kinematic chains and its application to human figures. *IEEE Transaction on Robotics and Automation*, Vol. 16, No. 2, pp. 124–134, 2000.

[51] Shinichiro Nakaoka, Atsushi Nakazawa, and Katsushi Ikeuchi. An efficient method for composing whole body motions of a humanoid robot. In *Proceedings of the Tenth International Conference on Virtual Systems and Multimedia*, pp. 1142–1151, Ogaki City, Japan, November 2004.

[52] Atsushi Nakazawa, Shinichiro Nakaoka, and Katsushi Ikeuchi. Imitating human dance motions through motion structure analysis. In *Proceedings of the 2003 IEEE International Conference on Intelligent Robots and Systems*, pp. 2539–2544, EPFL, Lausanne, Swizerland, September 2002.

[53] Atsushi Nakazawa, Shinichiro Nakaoka, and Katsushi Ikeuchi. Matching and blending human motions using temporal scaleable dynamic programming. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 287–294, Sendai, Japan, September 2004.

[54] Atsushi Nakazawa, Shinichiro Nakaoka, Shunsuke Kudoh, and Katsushi Ikeuchi. Digital archive of human dance motions. In *Proceedings of the Eighth International Conference on Virtual Systems and MultiMedia*, pp. 952–960, Gyeongju, Korea, September 2002.

[55] Atsushi Nakazawa, Takaaki Shiratori, and Katsushi Ikeuchi. Bust a move mr. avatar!: Character animation dance to music. In *Proceedings of Meeting on Image Recognition and Understanding 2005 (MIRU2005)*, pp. 1137–1144, Hyogo, Japan, July 2005. [in Japanese].

[56] Koichi Nishiwaki, Satoshi Kagami, Yasuo Kuniyoshi, Masayuki Inaba, and Hirochika Inoue. Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired ZMP. In *Proceedings of International Conference on Intelligent Robots and Systems*, pp. 2684–2689, EPFL, Lausanne, Swizerland, October 2002.

[57] Koichi Nishiwaki, Tomonobu Kitagawa, Tomomichi Sugihara, Satoshi Kagami, Masayuki Inaba, and Hirochika Inoue. Fast generation method of dynamically stable trajectory of humanoid motion base on the characteristics of ZMP. In *Proceedings of 19th Annual Conference of the Rbotics Society of Japan*, pp. 721–722, Japan, September 2000. [in Japanese].

[58] Koichi Nishiwaki, Tomomichi Sugihara, Satoshi Kagami, Masayuki Inaba, and Hirochika Inoue. Online mixture and connection of basic motions for humanoid walking control by footprint specification. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4110–4115, Seoul, Korea, May 2001.

[59] Koichi Ogawara, Kentaro Hashimoto, Jun Takamatsu, and Katsushi Ikeuchi. Grasp recognition using a 3d articulated model and infrared images. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1590–1595, Las Vegas, Nevada, October 2003.

[60] Koichi Ogawara, Jun Takamatsu, Hiroshi Kimura, and Katsushi Ikeuchi. Vision based acquisition of manipulation tasks. *IPSJ Transactions on Computer Vision and Image Med*, Vol. 442, No. SIG17(CVIM8), pp. 13–23, December 2003.

[61] Takashi Ogura, Kei Okada, Masayuki Inaba, and Hirochika Inoue. Posture manipulation interface of life-size humanoid using small humanoid with 37 DOFs "Tot". In *Proceedings of 21th Annual Conference of the Rbotics Society of Japan*, p. 2A21, Japan, September 2003. [in Japanese].

[62] Yu Ogura, Hiroyuki Aikawa, Hun ok Lim, and Atsuo Takanishi. Development of a human-like walking robot having two 7-DOF legs and a 2-DOF waist. In

Proceedings of 2004 IEEE International Conference on Robotics and Automation, pp. 134–139, New Orleans, LA, April 2004.

[63] Masafumi Okada, Yoshihiko Nakamura, and Shin ichiro Hoshino. Development of the cybernetic shoulder - a three DOF mechanism that imitates biological shoulder-motion -. In *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 543–548, Seoul, Korea, October 1999.

[64] Masafumi Okada, Tetsuya Shinohara, Tatsuya Gotoh, Shigeki Ban, and Yoshihiko Nakamura. Double spherical joint and backlash clutch for lower limbs of humanoids. In *Proceeding of IEEE International Conference on Robotics and Automation*, pp. 491–496, Taipei, Taiwan, September 2003.

[65] Nancy S. Pollard, Jessica K.Hodgins, Marcia J. Riley, and Christopher G. Atkeson. Adapting human motion for the control of a humanoid robot. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pp. 1390–1397, Washington DC, April 2002.

[66] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brain P. Flannery. NUMERICAL RECIPES in C. 1988.

[67] Marc H. Raibert. *Legged Robots That Balance*. MIT Press, 1986.

[68] Alla Safonova, Nancy S. Pollard, and Jessica K.Hodgins. Optimizing human motion for the control of a humanoid robot. In *Proceedings of 2nd International Symposium on Adaptive Motion of Animals and Machines*, Kyoto, Japan, March 2003.

[69] Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. The intelligent ASIMO: System overview and integration. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2478–2483, EPFL, Lausanne, Swizerland, October 2002.

[70] Stefan Schaal. Is imitation learning the route to humanoid robots ? *Trends in Cognitive Science*, Vol. 3, pp. 233–242, 1999.

[71] Takaaki Shiratori, Atsushi Nakazawa, and Katsushi Ikeuchi. Detecting dance motion structure through music analysis. In *Proceedings of The 6th International Conference on Automatic Face and Gesture Recognition*, pp. 857–862, May 2004.

[72] Seyoon Tak, Oh young Song, and Hyeong-Seok Ko. Motion balance filtering. In *Proceedings of EUROGRAPHICS*, Vol. 19, pp. 437–446, INTERLAKEN, August 2000.

[73] Jun Takamatsu, Hiroshi Kimura, and Katsushi Ikeuchi. Classifying contact status for recognizing human assembly tasks. In *Proceedings of IEEE /SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 177–182, 1999.

[74] Jun Takamatsu, Takuma Morita, Koichi Ogawara, Hiroshi Kimura, and Katsushi Ikeuchi. Representation of knot tying tasks for robot execution. *Journal of the Robotics Society of Japan*, Vol. 23, No. 5, pp. 572–582, 2005. [in Japanese].

[75] Jun Takamatsu, Koichi Ogawara, Hiroshi Kimura, and Katsushi Ikeuchi. Understanding of human assembly tasks for robot execution generation of optimal trajectories based on transition of contact relations. *Journal of Robot Sciety of Japan*, Vol. 22, No. 6, pp. 62–73, 2004. [in Japanese].

[76] Atsuo Takanishi. Quasi dynamic walking of the biped walking robot. *Journal of the Robotics Society of Japan*, Vol. 11, No. 3, pp. 196–203, 1983. [in Japanese].

[77] Atsuo Takanishi. Biped walking robot compensating moment by trunk motion. *Journal of the Robotics Society of Japan*, Vol. 11, No. 3, pp. 348–353, 1993. [in Japanese].

[78] Yukiharu Tamiya, Masayuki Inaba, and Hirochika Inoue. Realtime balance compensation for dynamic motion of full-body humanoid standing on one leg. *Journal of the Robotics Society of Japan*, Vol. 17, No. 2, pp. 268–274, 1999. [in Japanese].

[79] Koji Terada, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. Analysis and control of whole body dynamic humanoid motion – towards experiments on a roll-and-rise motion. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1382–1387, Las Vegas, Nevada, October 2003.

[80] M. Vukobratovic, B.Borovac, D.Surla, and D.Stokic. *Biped Locomotion: Dynamics, Stability, Control and Application*, Vol. 7 of *Scientific Fundamentals of Robotics*. Springer-Verlag, 1990.

[81] M. Vukobratović and D. Juricic. Contribution to the synthesis of biped gait. *IEEE Transaction on Biomedical Engineering*, Vol. 16, No. 1, pp. 1–6, 1969.

[82] Jinichi Yamaguchi, Atsuo Takanishi, and Ichiro Kato. Development of biped walking robot compensating for three-axis moment by trunk motion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 561, Yokohama, Japan, July 1993.

[83] Katsu Yamane, Jessica K. Hodgins, and H. Benjamin Brown. Controlling a marionette with human motion capture data. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pp. 3834–3841, Taipei, Taiwan, September 2003.

[84] Katsu Yamane and Yoshihiko Nakamura. Dynamics filter - concept and implementation of online motion generator for human figures. *IEEE Transaction on Robotics and Automation*, Vol. 19, No. 3, pp. 421–432, 2003.

[85] Katsu Yamane and Yoshihiko Nakamura. Natural motion animation through constraining and deconstraining at will. *IEEE Transaction on Visualization and Computer Graphics*, Vol. 9, No. 3, pp. 352–360, 2003.

[86] Kazuhito Yokoi, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Shuji Kajita, and Hirohisa Hirukawa. A honda humanoid robot controlled by aist software. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, pp. 259–264, Tokyo, November 2001.

[87] Kazuhito Yokoi, Katsumi Nakashima, Masami Kobayashi, Humisato Mihune, Hitoshi Hasunuma, Yoshitaka Yanagihara, Takao Ueno, Takuya Gokyuu, and Ken Endou. A tele-operated humanoid robot drives a backhoe in the open air. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1117–1122, Las Vegas, Nevada, October 2003.

[88] Kazuhiko Yokoyama, Hiroyuki Handa, Takakatsu Isozumi, Yutaro Fukase, Kenji Kaneko, Fumio Kanehiro, Yoshihiro Kawai, Fumiaki Tomita, and Hirohisa Hirukawa. Cooperative works by a human and a humanoid robot. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pp. 2985–2991, Taipei, Taiwan, September 2003.

[89] Takashi Yukawa, Takaaki Kaiga, Kazuo Nagase, and Hideo Tamamoto. Human motion description system using BUYO-FU. *IPSJ Journal*, Vol. 41, No. 10, pp. 2873–2880, October 2000. [in Japanese].