博士論文

TEMPORAL AND SPATIAL MOTION STYLE IMITATION BASED ON PHYSICAL CONSTRASINTS FOR HUMANOID ROBOT

(人型ロボットのための物理的制約に基づいた
時間的ならびに空間的動作スタイルの模倣)

BY

TAKAHIRO OKAMOTO
岡元 崇紘

A DOCTORAL DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL OF
THE UNIVERSITY OF TOKYO

東京大学
THE UNIVERSITY OF TOKYO

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF INFORMATION SCIENCE AND TECHNOLOGY

DECEMBER 2014

Committee:

Kiyoharu AIZAWA (Chair)
Takeshi NAEMURA
Yoichi SATO
Takeshi Oishi
Toshihiko YAMASAKI

Supervisor:

Katsushi IKEUCHI

ABSTRACT

Recently entertainment area is becoming one of promising applications of robotics technologies. Humanoid robots have appearance resembling a human being, therefore, they are suitable for entertainments such as singing, dancing, and natural interaction with guests at theme parks or stage shows.

For such applications, methods for creating contents such as natural and human-like motions for humanoid robots are important. One of those methods is capturing human motions, using a motion capture system, and applying them to robots.

On the other hand, just repeating the performances will soon become obsolete. To avoid this and to improve the live aspect of entertainment robots, capability of showing various behaviors according to situation will be desirable. In case of dance performance, autonomously synchronizing the whole-body dance motions with the musical tempo is an important task for a dancing robot. To create such motion variation, in the CG community, there are many studies to reuse captured motion data by editing and processing as necessary. But unfortunately these are unavailable on physical humanoid robots with severe constraints.

In this thesis, we propose methods to automatically generate variations of whole-body motions feasible for humanoid robots. As approaches, we observe how human motions vary according to situations, and then use insights from the observation for robot motion generation. Those motion variations specific to particular constraints are called *motion styles* in this thesis. Proposed method first analyze the given human motions via a learning-from-observation (LFO) paradigm, and then extract parameters which abstract those motions. Finally, natural and human-like robot motions are generated by optimizing those parameters to represent the humans motion styles well within the physical constraints of robots.

In thesis, we propose two type of motion styles and frame works to imitate those styles using physical humanoid robots.

We present a method to generate dance motions according to various musical tempos. When a same human dancer performs a dance to a same musical piece, the details of the dance movements vary tempo to tempo. It would appear that dance motions are abbreviated, preserving only essential factors as dance, to follow tempos within the ability to exercise. We call these motion styles, which are specific to temporal constraints such as music tempos, as *Temporal Motion Styles*. In this research, we observed temporal motion styles, and present a method to imitate those styles using a physical humanoid robot. First of all, we observed a same dance by three dancers in various tempos. As a result of observation, we obtained insights that specific postures, which we call *keypose*, tend to be preferentially preserved even if tempos becomes faster. Based on the insights we designed a framework to generate the dance motion variations according to musical tempos and within the physical constraints of the robot platform. Additionally, as an application of the temporal motion styles, we present a method to generate dance motions in real-time to realize a dancing robot that can dance to live music with changeable tempo.

We present a method to generate motion variations by imitating person-specific styles in motions. We assume such person-specific motion styles derive from difference in physical characteristics, and call them *Spatial Motion Styles*. In this thesis we chose a ring toss game as a target motion and observed demonstrations, captured from seven players whose motion style are different from each other, representing them as abstract parameters defined in a LFO paradigm. As a result of observation, we found that statistical distribution of the parameter differ according to personality. Based on the insights, we propose a framework for generating robot motions that reflect person-specific motion styles which are automatically extracted from human demonstrations. To verify our proposed method we applied it to a ring toss game, and generated motions for a physical humanoid robot. Styles from each of three random players were extracted automatically from their demonstrations, and used for motion generation. Finally the robot could imitate the styles of each player without exceeding the limitation of its physical constraints, while tossing the rings to the goal.

Thus, in this thesis, we focused on motion styles which are not actively developed in the filed of robotics, proposed methods to generate motion variations by extending a LFO paradigm, and finally validated via experiments with physical humanoid robots.

Imitation of temporal motion styles described in this thesis will be applicable to other dances. Imitation of spatial motion styles will also applied widely to other activities other than ring toss motions. Additionally, such kind of generation of motion variations by imitating motion styles are not only extend the expression of entertainment robots but also applicable to such as digital archive of intangible cultural heritages which are heirless and vanishing.

論文要旨

近年，エンターテイメントはロボット技術の応用として有望な分野の一つになっている．特に，ヒューマノイドロボットは人間に近い外見を有しているため，テーマパークやステージショーなどにおいて，歌や踊り，人との自然なインタラクションといった娯楽を提供するなど様々な活用が期待される．

そのような応用のために，ヒューマノイドロボットの自然で人間らしい動作といったコンテンツを作成する手法は重要である．例えば，モーションキャプチャシステムによって取得した人間の動作をロボットに適用する事は，そうした解決方法の一つである．

一方で，事前にプログラムされたパフォーマンスを繰り返すだけではすぐに陳腐化してしまう．これを避け，エンターテイメントロボットとしてのライブ感を向上させるためには，オンボードのカメラやマイクロフォンなどを通じて実環境をセンシングしながら，ロボットが状況に応じた様々な振る舞いを提示出来ることが望ましい．ダンスパフォーマンスの場合であれば，全身動作の動きと音楽のテンポを自律的に同期させる事は，踊りロボットにとって重要なタスクである．こうした動作のバリエーションを作成するために，コンピュータグラフィックスの分野では，必要に応じてモーションキャプチャデータを加工・編集し，再利用するための研究が多くある．しかし残念ながら，これらは厳しい制約条件を持つロボット実機のヒューマノイドロボットには適用できない．

そこで本研究では，ヒューマノイドロボットで実行可能な全身動作のバリエーションを自動的に生成する手法を提案する．アプローチとして，我々は状況に応じてどのように人間の動きが変化するかを観察し，その観察から得られた知見をロボットの動作生成に利用する．本稿では特定の制約条件に固有な動作のバリエーションを「動作スタイル」と呼ぶ．提案手法はまず，与えられた人間の動きを観察学習パラダイムによって解析し，それらの動作を抽象化するパラメータを抽出する．それらのパラメータを，ロボットの物理的制約の範囲内で人間の動作スタイルをよく表現するように最適化する事で，自然で人間らしいロボットの動作が生成される．

本論文では，2 つのタイプの動作スタイルと，実機ロボットを使ってそれらのスタイルを模倣するためのフレームワークを提案する．

一つ目の研究では，我々は様々な音楽のテンポに応じた舞踊動作を生成する手法を提案する．本手法は，ある同一の楽曲に合わせて様々なテンポで踊る，人間の変形ストラテジーの観察に基づいている．ある同一の舞踊者が同一の楽曲に合わせて踊る場合，動きの細部はテンポによって異なっている．これは，運動能力の範囲内でテンポに追従するために，舞踊によって本質的な部分を残しながら，舞踊動作を省略しているためであると考えられる．我々はこれら音楽のテンポなど，時間的な制約に固有の動作スタイルを「時間的動作スタイル」と呼ぶ．本研究では，我々は時間的な動作スタイルを観察し，実機ロボットで模倣する方法を提案する．まずは

じめに，我々は様々なテンポにおける3人の舞踊者の同一の舞踊を観察した．観察の結果，我々がキーポーズと呼ぶ特定の姿勢が，テンポが速くなっても優先的に保存される傾向があるという知見を得た．この知見に基づいて，我々は音楽のテンポに応じてロボットの物理的な制約の範囲内で舞踊動作のバリエーションを生成するフレームワークを設計した．加えて，時間的な動作スタイルの応用例として，テンポが変化する生演奏に合わせて踊る舞踊度ロボットを実現するための，リアルタイムな舞踊動作生成手法を提案する．

二つ目の研究では，我々は人に固有な動作スタイルによるバリエーションの生成手法を提案する．本手法では，タスクモデルによってある人の複数回の動作を解析し，共通する振る舞いをその人に固有なスタイルとして抽出する．我々はそのような人に固有の動作スタイルが身体特性の違いによるものであると仮定し,「空間的動作スタイル」と呼ぶ．本稿では我々は輪投げ動作を対象の動作として選択し，スタイルの異なる7人のプレイヤーから抽出された動作をLFOで定義される抽象パラメータによって表現し観察した．観察の結果我々は統計的な分布が個性によって異なるという知見を得た．この知見に基づき，我々は人間の動作から自動的に抽出された人に固有の動作スタイル反映したロボットの動きを生成するフレームワークを提案する．提案手法の妥当性を検証するために，我々はそれを輪投げ動作に適用し，実機ロボットの動作を生成した．ランダムに選ばれた3人のプレイヤーの動作スタイルが動作生成に用いられ，最終的にロボットは輪を目標に向かって投げながら，物理的な制約の範囲内でそれぞれのプレイヤーのスタイルを模倣することが出来た．

以上，本稿ではこれまでにロボティクスの分野であまり扱われて来なかった動作スタイルの模倣について着目し，LFOのフレームワークの拡張によって，ロボットの全身動作パターンのバリエーションとして表現する手法を提案，実際のヒューマノイドロボットを用いたデモンストレーションによってその有効性を示した．

本研究で提案する時間的スタイルの模倣は，舞踊動作というドメインの中であれば多くの異なる舞踊にも有効であり，空間的な動作スタイルの模倣は輪投げ動作以外の様々な全身動作について広く適用可能であると考えられる．また，このような動作スタイルの模倣によるバリエーションの生成は，エンターテイメントにおけるヒューマノイドロボットの表現の幅を広げるだけでなく伝統舞踊など後継者不足により失われつつある無形文化財のアーカイブ化においても活用が期待されるものである．

Acknowledgements

First and foremost, I want to express my profound gratitude to my advisor, Professor Katsushi Ikeuchi, who have supervised me for totally six years of my master's and doctoral program. His dedicated supervision through weekly meeting was essential to complete my doctoral dissertation, and have a helpful influence how I think. I demonstrate my gratitude to Associate Professor Takeshi Oishi. He also supervised me with my advisor through weekly meeting.

I am also grateful to Dr. Shunsuke Kudoh. It may be no exaggeration to say that he was my second advisor. He also have supervised me with my advisor, and I have taken lesson from him. I want to say a big thankyou to Dr. Takaaki Shiratori. He gave me a chance to work as an intern at Disney Research Pittsburgh, and supervised me as a mentor at Disney. The days at Pittsburgh provided invaluable experiences to me.

I demonstrate my gratitude to Dr. Shinichiro Nakaoka. He have provided a sophisticated software, called Choreonoid, for kinetics simulation of humanoid robots. The software provides many features which are necessary to compute robot motions, and I could not have developed my dancing robot without the help of them. Additionally he have unusually helped me to conduct experiments with physical humanoid robots.

I extend a special thank you to Dr(Expected). Yoshihiro Sato. He have taught me the basis of how to research and make a presentation. He have helped me as a most accessible mentor in my academic life. I'd like to sat thank you to Dr(Expected). Yoshie Kobayashi who started her doctoral program in the same year. I got spiritual sustenance from her. I also express my thanks to secretaries and staff in Ikeuchi Lab. They have supported my basis of academic life. I want to show my gratitude to people from Warabi-za. Dr. Takaaki Kaiga and staff have supported capturing human dancing via motion capture system.

I want to show my gratitude to people from Disney Research. Professor Katsu Yamane, Matthew Glisson, Jared Goerner, Jens Kober, Gor Nilanon, Yu Zheng, and Michael Mistry have supported my research with a humanoid robot. I want to express my profound gratitude to Marek Vondrak, Seung-Chan Kim, Munehiko Sato, Ashwini Kanth, and Akihiko Murai for helping my life in Pittsburgh.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Recently entertainment area is becoming one of promising applications of robotics technologies. Animal robots and humanoids are becoming popular as a personal hobby. Humanoid robots have appearance resembling a human being, therefore, they are suitable for entertainments such as singing, dancing, and natural interaction with guests at theme parks or stage shows [Got07, KFI*03, SWA*02, Got07, KKM*09].

For such applications, methods for creating contents such as natural and human-like motions for humanoid robots are important. For example, one of those methods is capturing human motions, using a motion capture system, and applying them to robots [NNK*07, PHRA02, YRA13, JM02, SYK*08, NK12, MYN13]. However, in most cases, direct mapping of human motions to robots does not work because of the differences in physical characteristics such as body type, mass distribution, degrees of freedom, angular range, and speed range. Therefore, in previous works, modification of human motions according to physical constraints of the robot and dynamic compensations of motions, such as balance adjustment, have focused on and actively studied.

On the other hand, just repeating the performances, which are preliminarily programmed, will soon become obsolete. To avoid this and to improve the live aspect of entertainment robots, capability of showing various behaviors according to situation, sensing the actual environment via on-board camera/microphones, will be desirable. In case of dance performance, autonomously

	Temporal Motion Style	Spatial Motion Style
Specific to	Temporal constraints	Spatial constraints
Instance of constraints	Music tempo	Physical feature

Table 1.1: The definition of temporal and spatial motion styles.

synchronizing the whole-body dance motions with the musical tempo is an important task for a dancing robot; When tempos of music become faster, the dancing robot have to dance more faster according to recognized tempos without losing balance. In this case the robot needs variation of dance motions whose tempos are different from each other. To create such motion variation, in the CG community, there are many studies to reuse captured motion data by editing and processing as necessary [BW95, HPP05, HKG06, MPS06]. But unfortunately these are unavailable on physical humanoid robots with severe constraints.

In this thesis, we propose methods to automatically generate variations of whole-body motions feasible for humanoid robots. As approaches, we observe how human motions vary according to situations, and then use insights from the observation for robot motion generation. Those motion variations specific to particular constraints are called *motion styles* in this thesis. Proposed method first analyze the given human motions via a learning-from-observation (LFO) paradigm, and then extract parameters which abstract those motions. Finally, natural and human-like robot motions are generated by optimizing those parameters to represent the humans motion styles well within the physical constraints of robots.

In thesis, we propose two type of motion styles as shown in table 1.1 and frame works to imitate those styles using physical humanoid robots.

Temporal Motion Styles and Imitation by a Humanoid Robot

We present a method to generate dance motions according to various musical tempos. This method is based on insights from observation of humans modification strategies to dance to a same musical piece in various tempos. When a same human dancer performs a dance to a same musical piece, the details of the dance movements vary tempo to tempo. It would appear that dance motions are abbreviated, preserving only essential factors as dance, to follow tempos within the ability to exercise. We call these motion styles, which are specific to temporal constraints such as music tempos, as *Temporal Motion Styles*. In this research, we observed temporal motion styles, and present a method to imitate those styles using a physical humanoid robot. First of all, we observed a same dance by three dancers in various tempos. As a result of observation, we obtained insights that specific postures, which we call *keypose*, tend to be preferentially preserved even if tempos becomes faster. Based on the insights we designed a framework to generate the dance motion variations according to musical tempos and within the physical constraints of the robot platform. Additionally, as an application of the temporal motion styles, we present a method to generate dance motions in real-time to realize a dancing robot that can dance to live music with changeable tempo.

Spatial Motion Styles and Imitation by a Humanoid Robot

We present a method to generate motion variations by imitating person-specific styles in motions. In this method, a robot analyzes multiple demonstrations performed by a person using task models, and then extracts the common behaviors as styles for that particular person. We assume such person-specific motion styles derive from difference in physical characteristics, and call them *Spatial Motion Styles*. In this thesis we chose a ring toss game as a target motion and observed demonstrations, captured from seven players whose motion style are different from each other, representing them as abstract parameters defined in a LFO paradigm. As a result of observation, we found that statistical distribution of the parameter differ according to personality. Based on the insights, we propose a framework for generating robot motions that reflect person-specific motion styles which are automatically extracted from human demonstrations. To verify our proposed method we applied it to a ring toss game, and generated motions for a physical humanoid robot. Styles from each of three random players were extracted automatically from their demonstrations, and used for motion generation. Finally the robot could imitate the styles of each player without exceeding the limitation of its physical constraints, while tossing the rings to the goal.

1.2 Thesis Overview

Chapter 2 reviews the concept of a LFO paradigm, and then describes a framework of robot motion generation based on task models defined in a LFO paradigm. Proposed method in this thesis are extension of task models and described in the context of LFO.

Chapter 3 describes about observation of temporal motion styles. First of all, we report the analysis of dance motions to various musical tempos. Then we present a method to modify dance motions based on insights from analysis, and generate variations according to arbitrary musical tempos. This is done by extending a dance motion generation based on task model introduced in chapter 2. Proposed method in this chapter is applied to the Don-pan dance, a Japanese folk dance from Akita prefecture. Experiments with a physical humanoid robot HRP-4C validated that dance motions generated by our proposed method are feasible for robots. Additionally, as an application of temporal motion styles, we present a framework for on-line dance motion generation based on temporal motion styles. This is a key-component for a robot dancer that can dance to live music with changeable tempos

Chapter 4 describes about observation of spatial motion styles. First of all, we observe demonstrations of a target motion performed by several persons and then model the common structure of the target motion based on task models. Then multiple demonstrations from each person are analyzed based on the model and parameterized using abstract parameters. The person-specific statistical distribution of the abstract parameters is characterized as spatial motion styles of the person. Then we introduce a framework to generate robot motions that are considered similar in motions styles automatically extracted from multiple demonstrations of a particular person.

Finally, Chapter 5 concludes this thesis. We summarize this thesis, discuss our contribution, and then mention about future work.

Chapter 2

Learning-from-Observation Paradigm

Our proposed method in this thesis are extension of a concept of a LFO paradigm. This chapter first gives an explanation of the concept of the paradigm, and then describes how to imitate human motions in those context.

2.1 Learning from Observation

Teaching robots has been one of the most important issues in the field of robotics. Learning skills through observing human demonstrations is an intellectual ability we desire from intelligent robots. Especially for versatile robots which work in our living environment for end-users, such capability will be essential. Because end-users does not necessarily have special expertise in robotics enough to program the robot.

To achieve such learning capability Ikeuchi *et al* [IS94, SI92] has introduced a learning from observation (LFO) paradigm and developed an abstract model called *task model* in the concept. In contrast to a burst of reinforcement learning approach [CGB07, MK98, Sch99, Ude99], this model first gives robots the prior knowledge to understand what a human is doing and to extract reusable essences within a specific task domain. The concept of task models has been successfully applied to complex manipulation tasks [IS94, KI97, TMO*06a]. The use of robots is expanding beyond industrial purposes to the entertainment area. It is also used to imitate full-body human motions such as a dance performance [NNK*07]. These examples illustrate a potential of task models in a wide

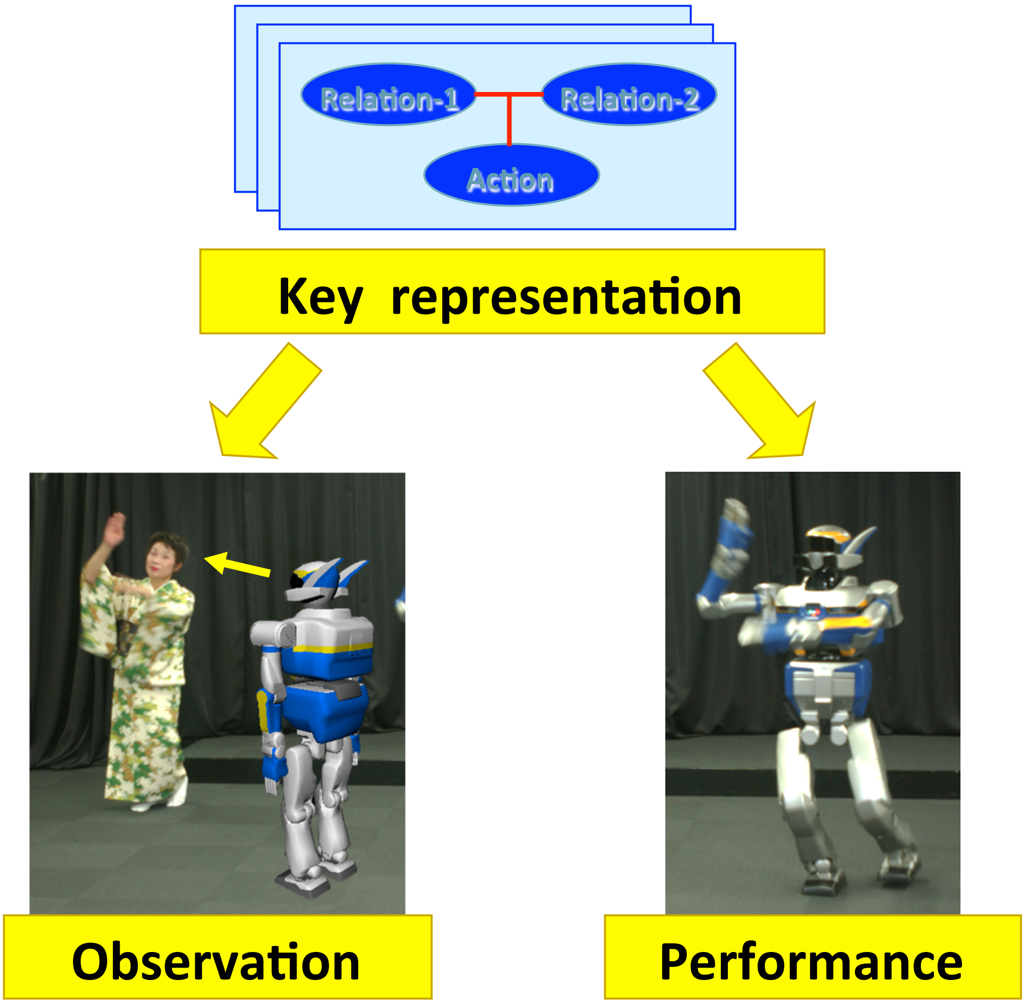


Figure 2.1: Learning from Observation.

range of applications, and accumulation of such applications to demonstrate the potential is one of our ultimate goals.

2.2 Task Models

As mentioned above, the LFO enables robots to learn how to perform various tasks from observing human performance [IS94, SI92]. As shown in Figure 2.1, the LFO generates robot actions through the following three steps:

1) A human dancer performs actions in front of the robot (Figure 2.1 left).

2) The robot recognizes those demonstrated actions based on pre-defined abstract task models, and constructs a series of task models (Figure 2.1 middle).

3) The robot converts those recognized task models into robot physical actions (Figure 2.1 right).

In general, performing the same action does not require mimicking the entire action performed. It is difficult, if not impossible, to repeat the same trajectories to be mimicked, because the humanoid robot has different dimensions from those of the human dancer. Instead, for this purpose, characteristics or important features of the actions are extracted and performed.

Essential and nonessential parts in each action are defined based on the knowledge of task domains. This top-down approach of designing domain-specific task models distinguishes our approach from other bottom-up learning approaches such as those developed by the Nakamura group [ITN03b, ITTN03, ITN03a] or the Kawato group [MK98, Sch99, Ude99]. Our top-down approach first defines task domains such as polyhedral-world operations [IKS93], flexible rope handling [TMO*06b], grasping motions[KI97], and whole body motions [OTKI03]. Then we define task models to represent all necessary essential actions based on domain knowledge.

The LFO introduces abstract task models to represent essential parts in a sequence of actions. Each abstract task model describes the task, i.e., what to do. Each task model also contains skill parameters that explain how to do the specific task. Usually recognizing tasks and extracting skill parameters are done automatically from an input data.

2.3 Motion Imitation Based on Task Models

Our humanoid robots are based on the LFO. In this section we explain how to imitate human motions based on task models taking a dancing robot as an example.

Our method handles upper-body, middle-body, and lower-body motions separately by defining different types of task models. This separation is natural because although the whole body dance motion is conducted simultaneously, the lower, middle, and upper bodies have different roles or constraints to play in the performance of the dance. Considering how human dancers are taught a dance in lessons, those body motions are often taught separately. Thus this separation is natural also for humans, and does not destroy the basic structure of the dance.

The constraint of lower-body motion is to stably support the whole body while performing a dance. The lower-body task model [NNK*07] is defined based on two foot-floor contacting conditions, STEP and STAND tasks (Figure 2.2). For the lower-body motion a continuous foot motion is segmented and recognized using these defined task models. The skill parameters defined for each task model characterize the trajectory of the foot, such as highest positions and length of stride (Figure 2.3). The obtained skill parameters modify the default trajectory of STEP tasks while stably supporting the whole body. Inverse-kinematics provides the joint angles of the robot's foot.

The aim of middle-body, i.e. the waist, motion is two-fold: expression of the dance, and balance maintenance by controlling zero moment point (ZMP). For dance expression, the SQUAT task [NNK*07] is defined to lower the waist position. The skill parameters attributed to this SQUAT task are: how deep the squat is, and the duration of each squat (Figure 2.4). The horizontal trajectory of the middle-body is generated by computing the balance of the whole body [NkK*02]. Although some dance categories may include artistic expressions with the horizontal movements of middle-body, those are out of our current scope to maintain dynamic stability.

The purpose of upper-body motion is to express the dance. Shiratori *et al* [SNI04] introduced keyposes for representing such dance characteristics. A keypose is defined as a fixed posture of a dancer for the purpose of providing the viewers with expressions and meanings of a dance. Figure 2.5 shows some of the keyposes in the Aizu-bandaisan dance, a Japanese traditional folk dance, depicted by a dance teacher. Some expert dancers indicate that these keyposes are the main points during the dance, and to mimic these keyposes is one of the important tasks in showing the beauty of the dance. Thus, we define the performance of keyposes as the upper-body task models. Shiratori *et al* [SNI04] have developed a method of extracting keyposes from continuous dance motions by detecting brief stop motions of dancers corresponding to music beats as shown in Figure 2.6. Upper body motions of a robot follow exactly its configurations at the keypose timing. The trajectory between keyposes, regarded as a skill parameter, is represented with a hierarchical B-spline.

By concatenating lower-body, middle-body, and upper-body motions using Nakaoka system [NNK*07], we can obtain the entire robot motion.

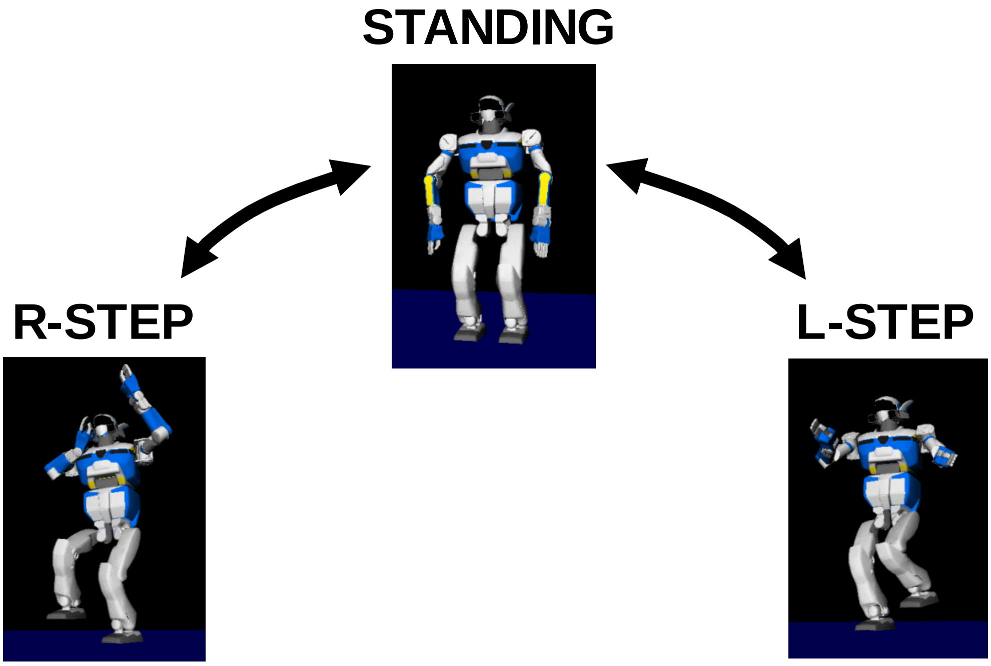
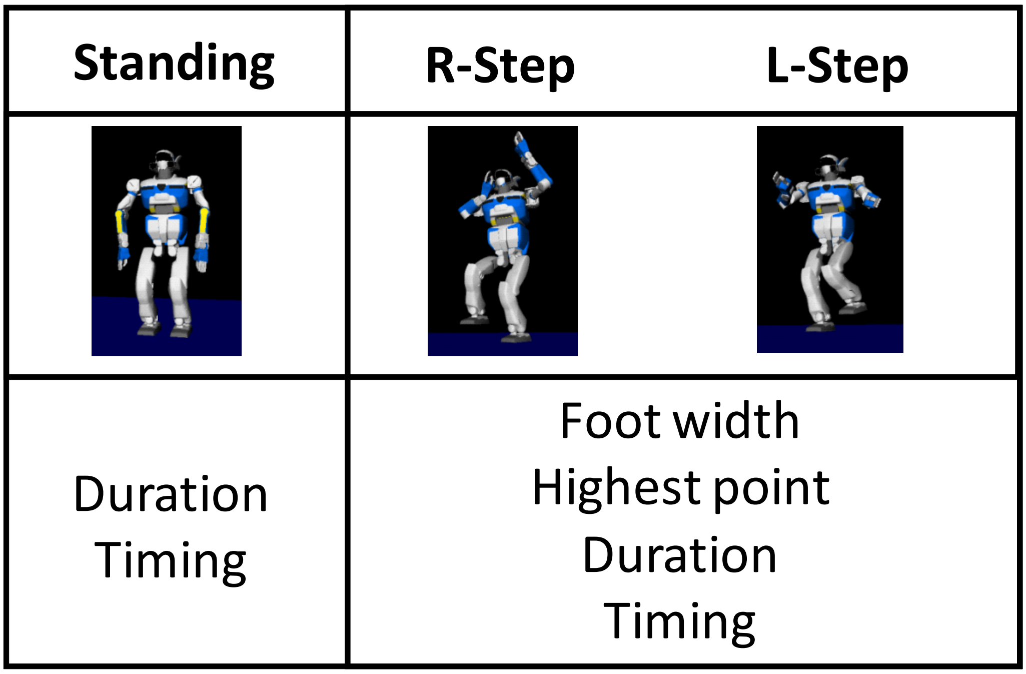


Figure 2.2: Lower-body task models. This models are based on transitions of foot-floor contact states. The STEP tasks and STAND tasks are complementary to each other in lower-body motions.



Standing	R-Step	L-Step
Duration Timing	Foot width Highest point Duration Timing	

Figure 2.3: Lower-body skill parameters. The skill parameters obtained from human motions modify the default trajectory of each task.

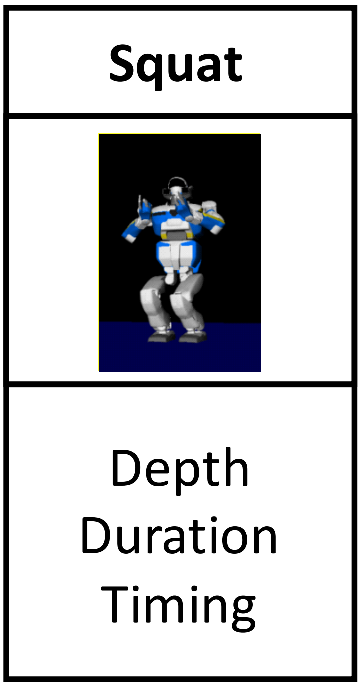


Figure 2.4: Middle-body skill parameters. The skill parameters obtained from human motions modify the default trajectory of SQUAT tasks.

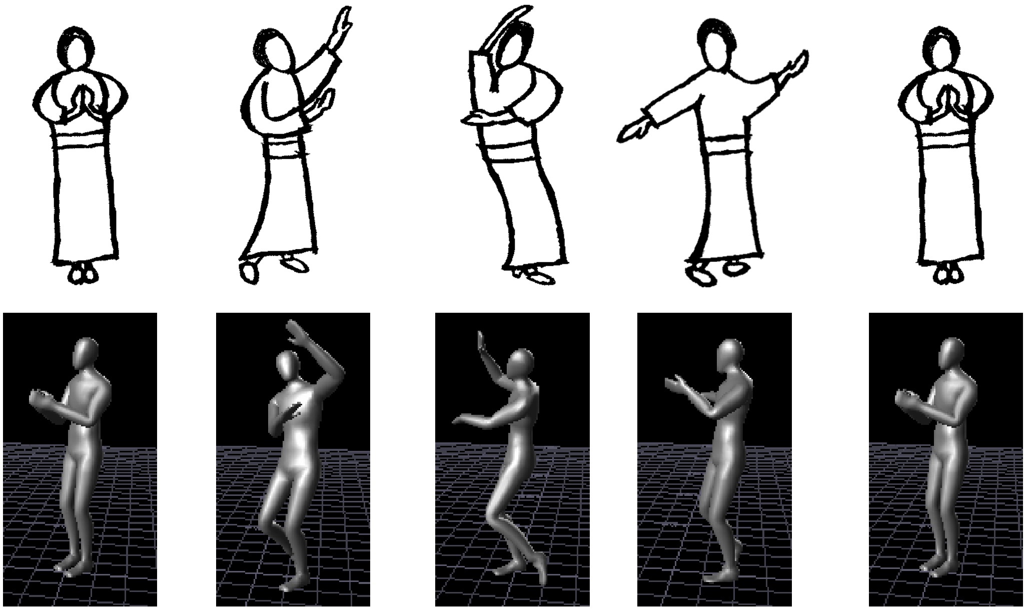


Figure 2.5: Keyposes in Aizu-bandaisan dance. upper row: keyposes depicted by a dance teacher. bottom row: brief stop motions of dancers corresponding to music beats extracted by [SNI04].

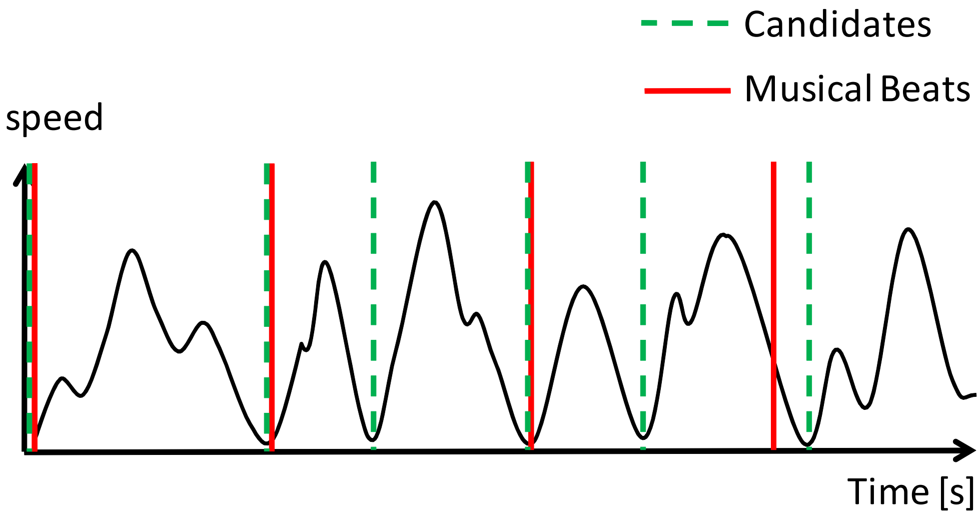


Figure 2.6: A method to detect keyposes. Keyposes are detected by extracting motion frames where the speed of movement become locally minimum at the timing of musical beats [SNI04].

Chapter 3

Temporal Motion Style

3.1 Introduction

The entertainment area is one of the promising application areas of humanoid robots. Entertainment applications such as dancer robots or musician robots, which even a human performer needs special practice to perform, require special skills for robots, and, as a result, push the horizon of robotics technologies. When successful, the performance of entertainment robots fascinates an audience.

Recently, development of such entertainment robots has accelerated as a showcase of robotics technologies [KFI*03, SWA*02, Got07, KKM*09]. Toyota introduced musician robots at a recent AICHI expo [Got07]. Kosuge developed a dance partner robot for dance practice [KHHT03]. At the University of Tokyo, our group have been developing a dancing robot based on the LFO paradigm [NNK*07, ISK*08]. In a different context, AIST developed a healing robot, PARO, for elderly people. As reported in psychological user studies relevant to robots [SW11], if people are to become interested in a robot, they expect it to achieve intelligent interaction, as if the robot were an actual human or animal. Therefore, an entertainment robot needs to have intelligent interaction with its environment as well as skillful motions.

Our dancing humanoid robot is based on *task models*, defined in the paradigm of LFO. The robot has the capability of observing human dance motion, analyzing such human dance motions using task models, and, finally, generating imitation motions with balance maintenance within motor limitations.

This chapter focuses on a human's capability for dancing to music performances of varying tempos, and proposes an algorithm to realize this capability in a humanoid robot. The previous algorithm, proposed by Nalaoka *et al.* [NNK*07] realized a pre-defined static interaction with the environment; the resulting robot can only perform a dance at a pre-fixed tempo. Proposed method in this chapter considers dynamic interactions for the robot to be able to modify its motion according to tempos of a given music piece.

3.1.1 Prior Works

Prior Works in the CG Community

Adjusting timings or time warping of captured human motions is popular in the field of computer graphics (CG) and animation. Bruderlin *et al.* [BW95] proposed a method to synthesize a new walking motion from the original one at different speeds based on a *dynamic time warping* (DTW) [SC78] technique, and many researchers are applying DTW for adjustment of motion [HPP05, HKG06]. In addition, methods for synchronizing a motion pattern with music [ABB05, LL05, SNI06] or retiming based on physical properties such as gravity [MPS06] are proposed for editing the motion of CG characters. However, these methods cannot be applied to the motions of a physical humanoid robot, which need consideration of joint limitations and maintaining balance. Since even slow motions such as those of a Japanese traditional dance in its original tempo need consideration of joint and balance limitation, faster motions proportionally scaled would easily exceed the limitations. Additionally, keeping artistic expression of faster dance motion within the limitations is necessary in our case. Shiratori *et al.* [SKNI07, SI08] proposed a method to temporally scale the captured motion data with such constraints; Their approach is based on observation of human dancing at various music tempos and keeps artistic expressions within the constraints. The only problem is, however, their method can not be applied to lower-body motions because it may cause a slip of the foot. So another method is required for lower-body motions.

Prior Works in the Robotics Community

Interaction via musical expression has been investigated in the robotics field. Murata *et al.* [YNT*07, MNY*08] have described a humanoid robot that can step to musical tempos using robot audition. Their work is focused on catching the target sound in a noisy environment including footstep or motor noise, and differs from our goal. Mizumoto *et al*. [MLO10] proposed a robot musician that mutually interacts with a human musician using musical instruments, such as a flute. Kojima *et al*. proposed a method to recognize various type of dance step performed by a user and to create imitation motions off-line for a humanoid robot with balance maintenance. Oliveria *et al*. [OGR08] proposed a small robot that moves its limbs according to detected onset timings and features of music in real time. Similarly, Grunberg *et al*. [GEKO09] proposed a humanoid robot that reacts to musical tempo, beat, and style. Additionally proposed are a robot that interacts with a human via simple dance [TS04, TFAM05, KMN09], a robotic percussionist [WDP05] that listens to human playing in real time and playback, a dance partner robot [KHHT03], and multiple quadrocopters flying to music [SAD10]. However, complicated whole body motion such as human dance and balance maintenance using legs were not dealt with in their works. Our target is more challenging in the way that our humanoid robot not only imitate artistic dance patterns using the whole body but also changes the speed of dance motions according to musical tempos via robot audition. A temporal scaling technique is necessary for the achievement of such a humanoid robot; dance motions have to be modified according to arbitrary musical tempos.

3.1.2 Approach

This chapter, first, analyzes human dancing, and extracts modification strategies, which we call *temporal motion styles*, used by skillful dancers to adapt motions to music tempos. Then motion modification strategies, based on the temporal motion styles, is proposed to create a dance motions at an arbitrary musical tempo.

Here, before modification, the algorithm assumes that robot motion at a certain music tempo is generated by using the Nakaoka system [NNK*07]. The validation of our proposed algorithm is conducted using a physical humanoid robot HRP-4C.

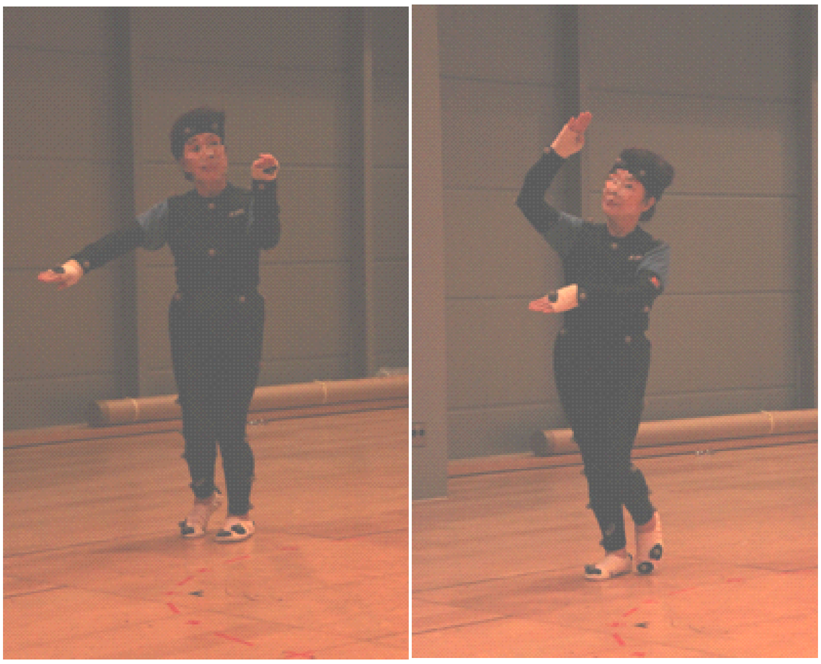


Figure 3.1: Observation of the Aizu-bandaisan dance performed by a dance master. Dance motions are captured using an optical motion capture system from VICON.

3.2 Observation of Temporal Motion Styles

The LFO method [NNK*07] provides a new way for a robot to learn how to dance at a fixed tempo when a human performance is observed. These learned dance motions are, unfortunately, fixed to that particular tempo when learned. One of the important features in performing a dance is for a dancer to vary its body motions along with the tempo performed at that moment. When the music tempo increases, the robot should dance more quickly; when the music tempo decreases, the robot should dance more slowly. The research described in this chapter aims to build such capability.

The variation strategy needs to be consistent with that of humans. This is because we aim to design a dance robot that gives an impression similar to that of a human dancer. For this purpose we first observed and analyzed how a human dancer modifies his or her motions along with music tempos [OSKI10].

Dance performances by human dancers at several different music tempos were captured through an optical motion capture system, VICON (See Figure 3.1). We sampled them at the original tempo, 1.2 times faster, 1.5 times faster, 1.8 times faster, and 2.0 times faster. Music at double speed is actually not practical for the dance, but, because human can move more quickly than robots, we need to investigate human performance to the music whose tempo is too fast for human to move perfectly. We used the Aizu-bandaisan dance, a Japanese traditional folk dance, as a dance example. This dance consists of cyclic patterns, each of which takes about 10 seconds. Three dancers performed the 10-15 cycles of the dance to each music tempo.

As was done in the previous task model design [NNK*07], we assumed that lower-body, middle-body, and upper-body would have different modification strategies. As for upper-body motions, observation of temporal motion styles for the Aizu-bandaisan dance and a modification algorithm are reported by Shiratori *et al*. [SKNI07, SI08]. So we conduct observation of middle and lower body motions separately, and then review the observation of the upper-body motions [SKNI07, SI08] in the following subsection for consistency.

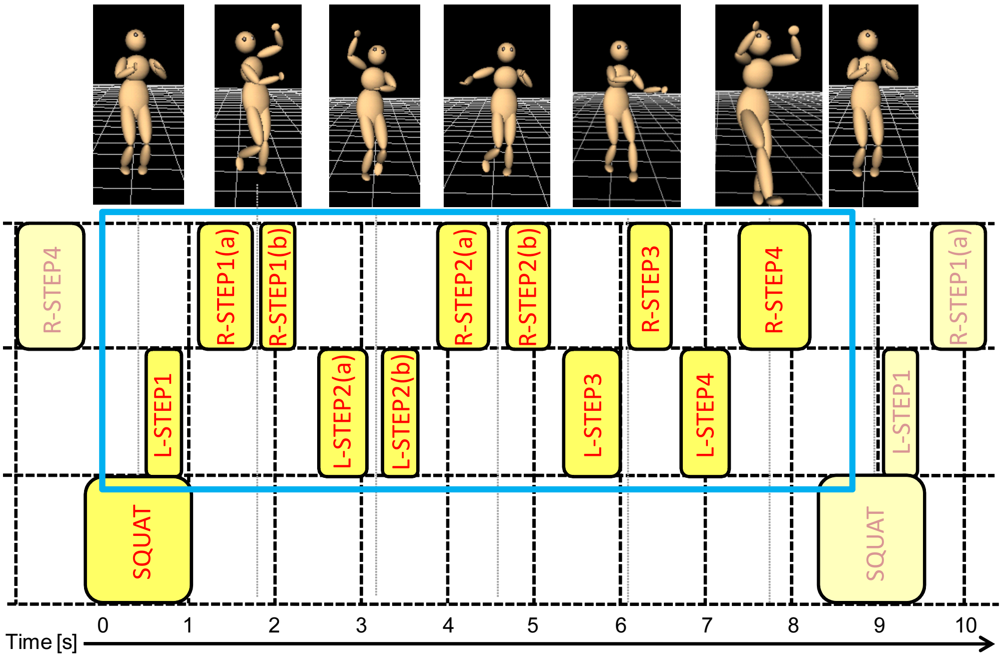


Figure 3.2: STEP tasks in a cycle of the Aizu-bandaisan dance. We extracted 11 STEP tasks for one cycle of 9 seconds. Here, R-STEP*n* and L-STEP*n* denote the *n*-th right and the left foot steps, respectively.

3.2.1 Lower-body Motions

The lower-body motions consist of STEP and STAND tasks. The STEP tasks and STAND tasks are complementary to each other in the lower-body motions. In the entire duration of one cycle of the dance, when the STEP tasks occur, the STAND tasks do not occur, and when the STEP tasks do not occur, the STAND tasks occur.

The Aizu-bandaisan dance consists of cycles of a sequence of tasks. In Figure 3.2, we extracted 11 STEP tasks for one cycle of 9 seconds. Here, R-STEP*n* and L-STEP*n* denote the *n*-th right and the left foot steps, respectively.

We observed start and end timing, length of stride, and the maximum speed of foot tips as well as the trajectories for STEP tasks, and only the start and end timing of the STAND tasks, as was done in our preliminary experiments reported in [OSKI10, OSK*14].

Maximum Speed of the Foot-tip

Figure 3.3 shows how the maximum speed of the foot-tip varies with the music tempo. As the music tempo become faster, the maximum speed of the foot-tip also become faster. However, the maximum speed is 1.5 times faster than original even in double tempo. From this observation, we learned that the maximum speed of the foot-tip does not vary as much as that of the musical tempo, probably due to the physical limitation of the dancer. Thus, the following discussion focuses on timing, length, and trajectories.

Timing

Figure 3.4 shows the variance of start and end timings of step motions depending on music tempos. Lines of different colors in the graph represent different tempos. Here we have normalized the horizontal axis so that one cycle of dance is always depicted from 0 to 1, independent of the music tempos. As can be seen in the graph, when the tempo of music increases, the variance of non-keypose steps become larger, while keypose steps have lower variance. Thus, we can conclude:

L-1 Timing of a STEP near a keypose will be maintained.

L-2 Timing of a STEP far from any keypose will be adjusted when necessary to accommodate music tempos.

Length of Stride

Figure 3.5 shows how the stride varies with the music tempo. The length of a stride is maintained even though the tempo increases. We have also observed a couple of examples of breakdown at a higher tempo. Thus, we can summarize that:

L-3 The length of a stride will be maintained as much as possible up to a certain threshold. Over this threshold, it will be reduced accordingly.

Trajectories of a Foot Tip

Figure 3.6 shows the trajectories of a foot tip in a STEP tasks at each tempo. The STEP task has a special trajectory like kicking up at the last part of each cycle of the dance. The trajectories become smaller when the music tempo increases.

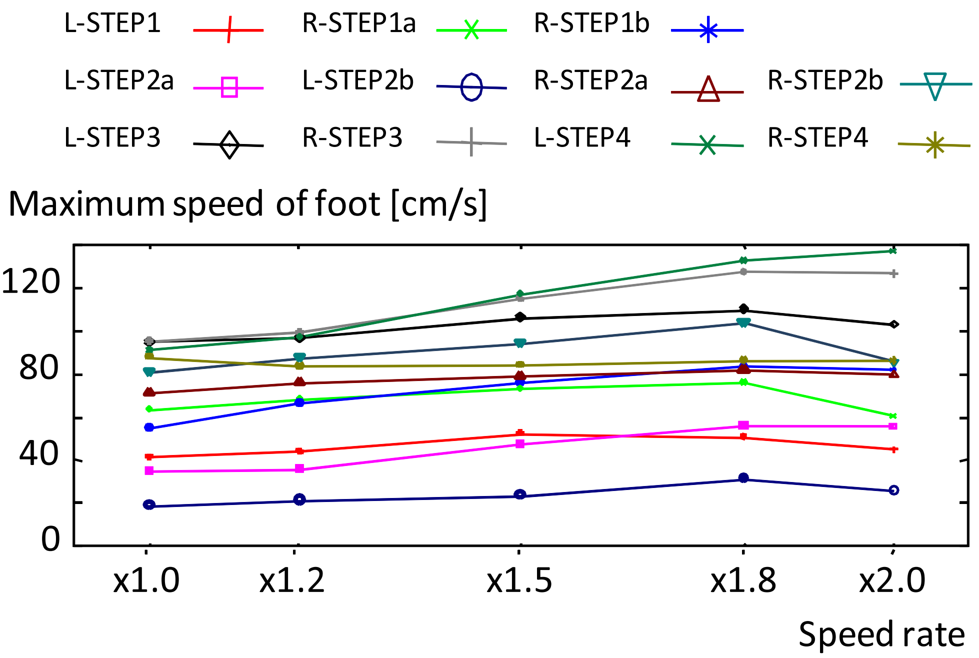


Figure 3.3: Maximum speed of foot-tip: each marker represents the average maximum speed of foot-tip and length of stride in STEP tasks at each musical tempo.

L-4 Trajectories of the foot tip become compact with increased music tempo.

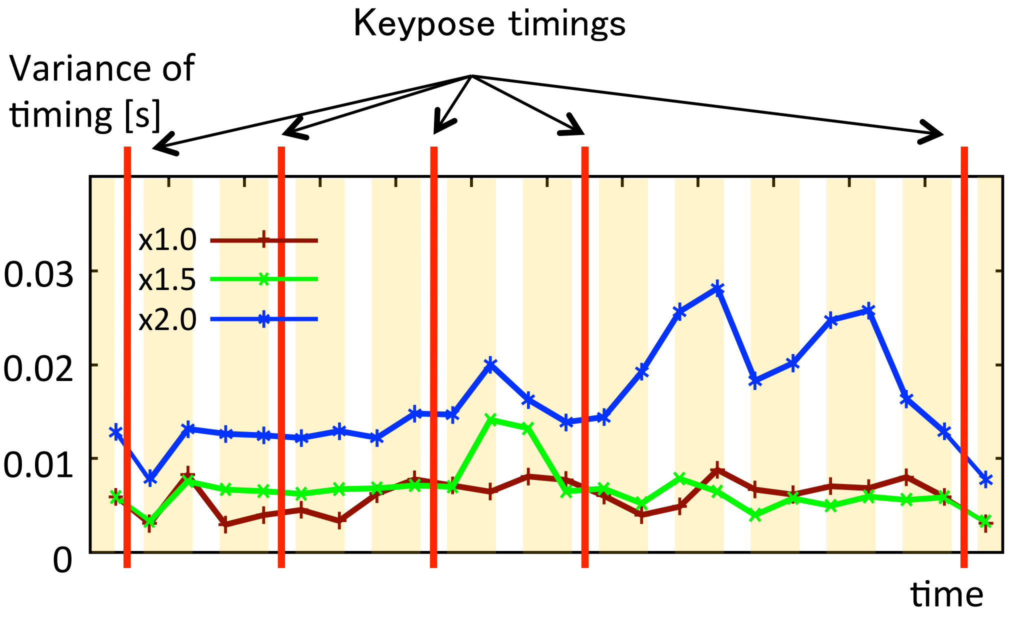


Figure 3.4: Variance of start/end timings of each STEP task: red, green, and blue markers provide variances at original tempo, 1.5 times faster, and 2.0 times faster.

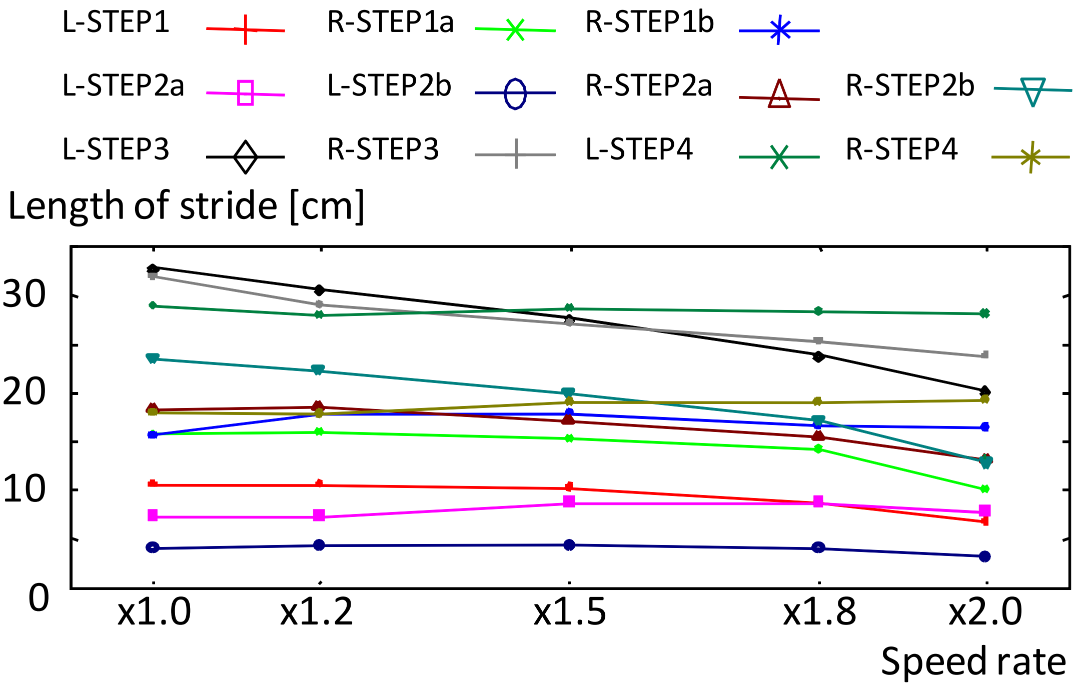


Figure 3.5: Length of stride: each marker represents the average maximum speed of foot-tip and length of stride in STEP tasks at each musical tempo.

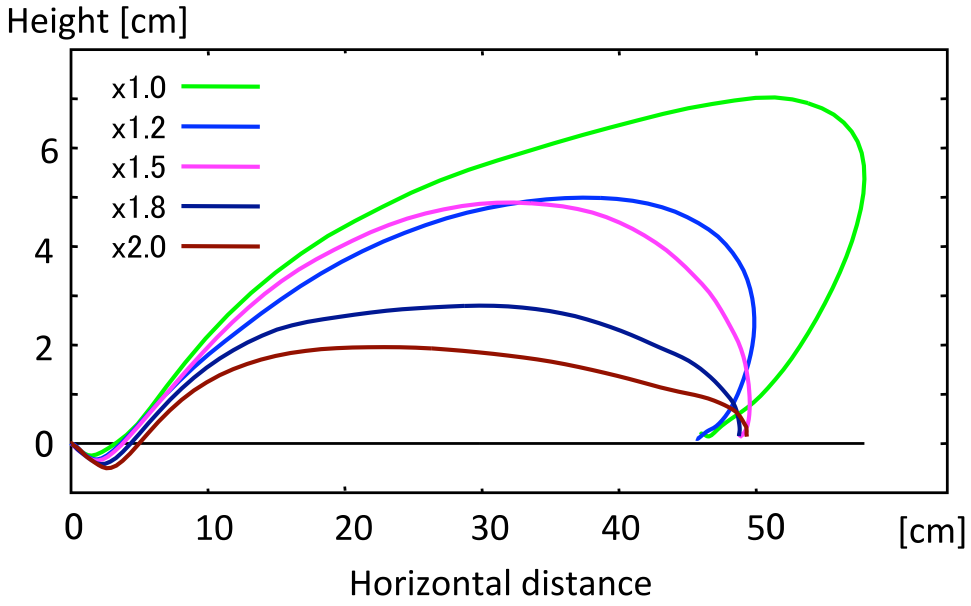


Figure 3.6: Trajectories of a foot tip in a STEP task labeled as R-STEP4 at each tempo. The STEP task has a special trajectory like kicking up at the last part of each cycle of the dance.

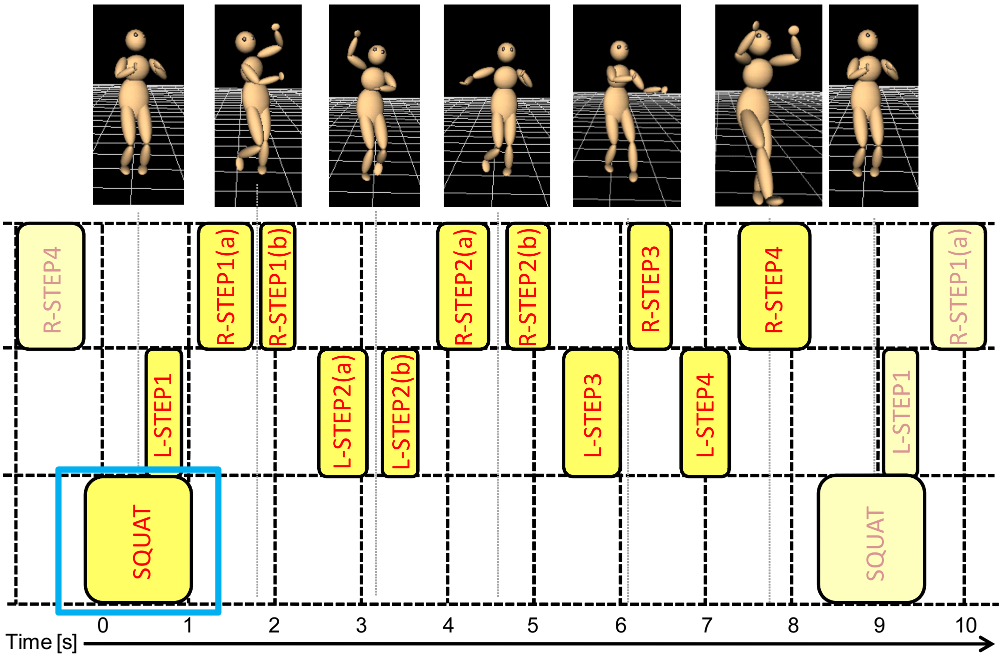


Figure 3.7: a SQUAT task in a cycle of the Aizu-bandaisan dance

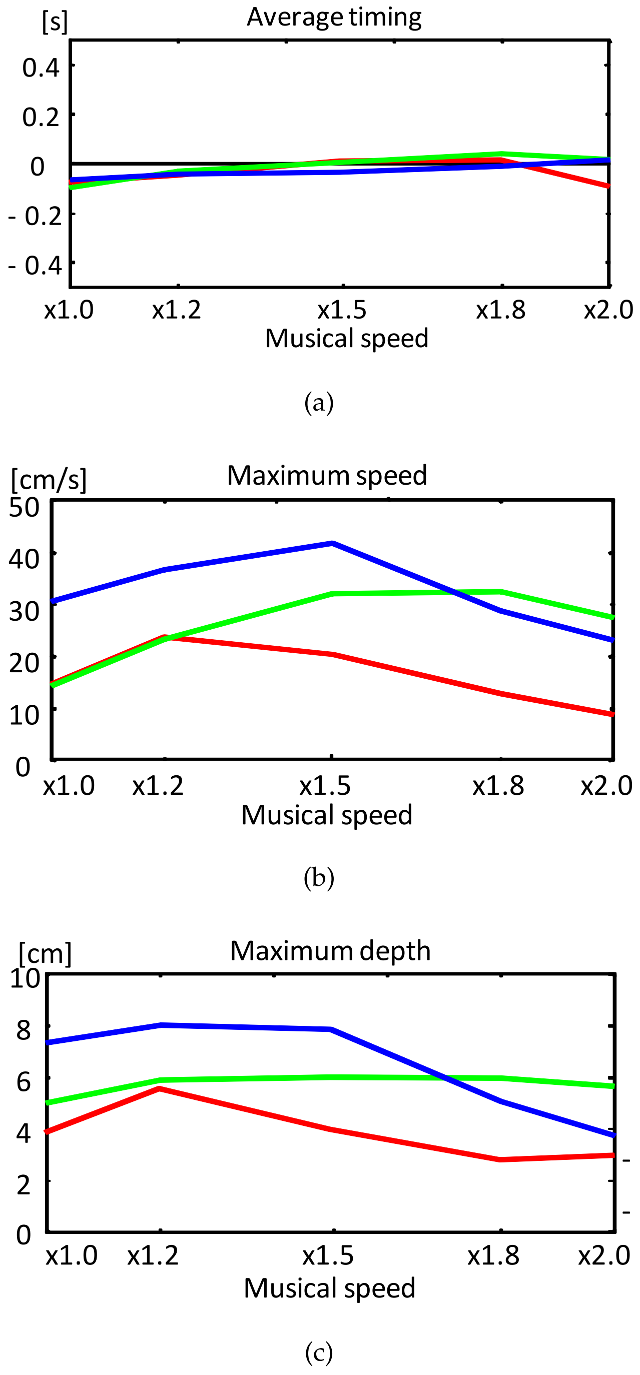
3.2.2 Middle-body Motions

For SQUAT tasks we observe timing to reach the maximum depth, the maximum depth, and the maximum speed of the waist. For the timing, as shown in Figure 3.8 (a), we find that even though the music tempo increases as depicted in the horizontal axis, the average timing does not change. The dancers try to maintain the SQUAT timing as much as possible. In the Aizu-bandaisan dance, this SQUAT corresponds to the keypose. The speed of the SQUAT and the depth of the SQUAT are depicted in Figure 3.8 (b) and (c), respectively. As shown in Figure 3.8 (b) as the music tempo increases, the speed of the SQUAT increases and the depth of the SQUAT is maintained. However, beyond a certain point, when it becomes difficult to increase the speed, the depth of the SQUAT gradually decreases.

From observation, we found the following characteristics:

M-1 The timing of the SQUAT, which usually occurs at the keypose, is maintained independently of the music tempo. There is no difference in timing even when the music speed increases to twice the original.

M-2 Dancers try to maintain the depth of the SQUAT by increasing the speed of the waist up to a certain music tempo. However, beyond this threshold tempo, the dancers accommodate the faster music tempos by reducing the depth.



(a)

(b)

(c)

Figure 3.8: Maximum speed, maximum depth, and average timing of SQUAT task with varying music tempos: red, green, and blue lines represent three dancers.

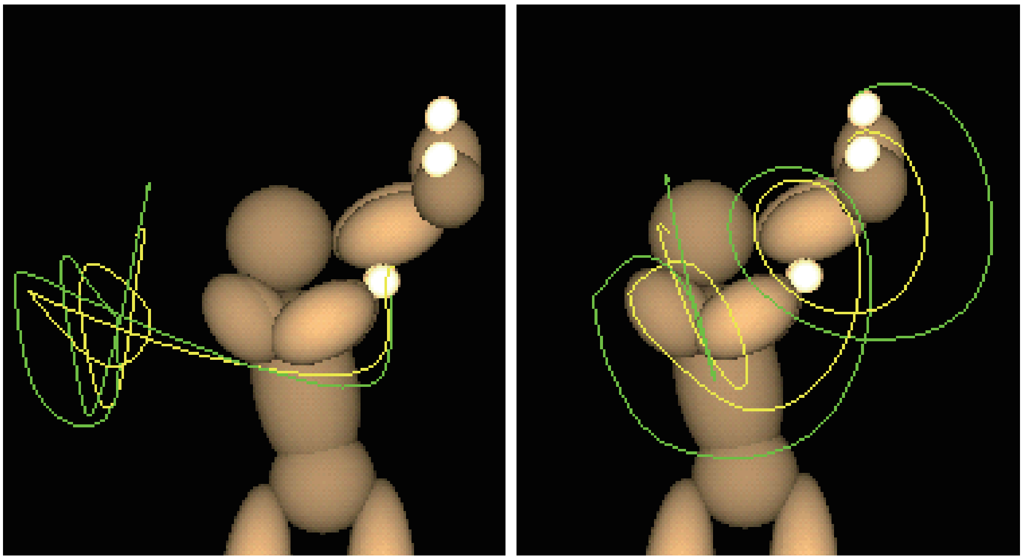


Figure 3.9: Comparison of hand trajectory differences depending on music speed. The green and yellow curves represent the hand trajectories at a normal musical speed and a 1.3 times faster musical speed, respectively [SKNI07, SI08].

3.2.3 Upper-body Motions

Here, we describe the observation of the upper-body motions conducted by Shiratori *et al*. [SKNI07, SI08]. The setup for observation is same as ours. They depicted the trajectories of hand motions of various music beats as shown in Figure 3.9. Apparently, as the music tempo becomes faster, the trajectories becomes more compact as was the case in the foot tip trajectories. It is reasonable for the dancer to move his or her hands along a shorter path in order to accommodate the music tempo.

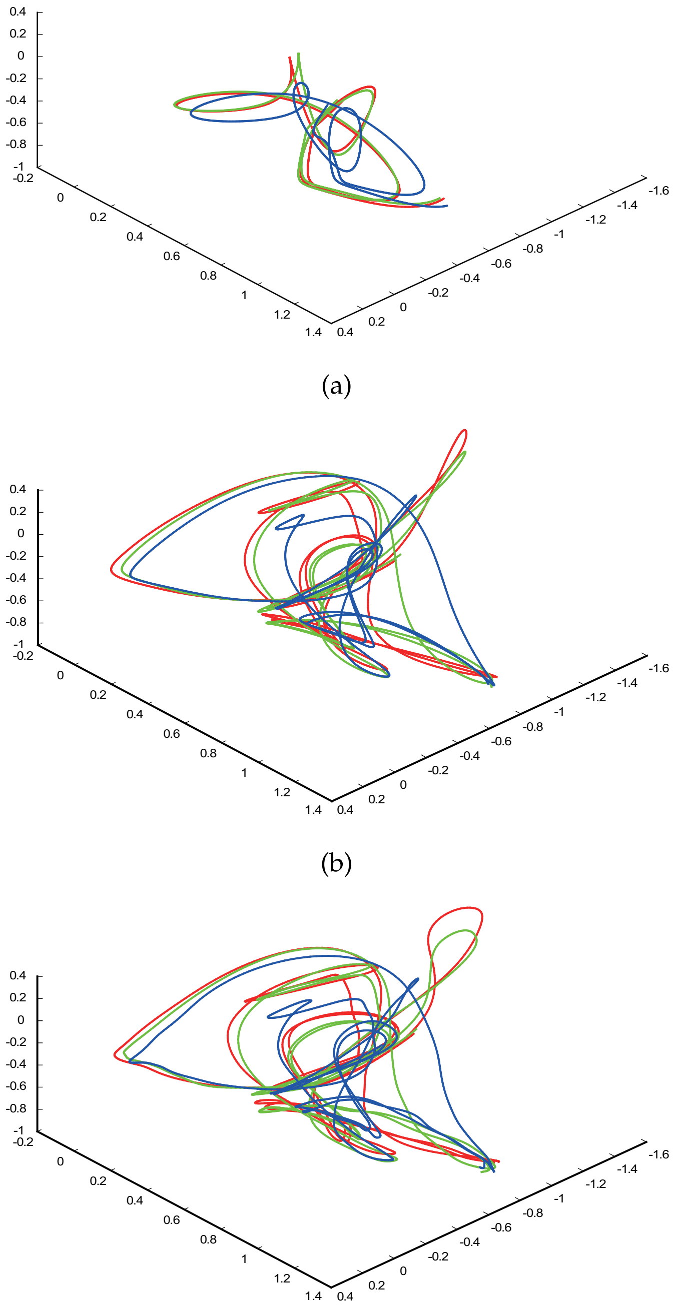
Along this line of observation, they decomposed a dancer's motion using the hierarchical B-spline technique [LS99, SKNI07, SI08] (See Figure3.10). (a) mean motion using a single-layer B-spline, (b) mean motion using a three-layer hierarchical B-spline, and (c) mean motion using a five-layer hierarchical B-spline. These trajectories are in the logarithmic space of a quaternion. Variation of trajectory according to tempos in Figure (c) is greater than that in Figure (a); Higher order motions are omitted preferentially with increased music tempo. At the first layer, by using a certain number of knots points sets based on the original music tempo, the dancer motion was represented by using B-Spline as shown in Figure3.10(a). Then, the difference of the original motion and resulting B-Spline is further represented by using a B-Spline of a finer interval of knots as shown in Figure3.10(b). This process is repeated iteratively as shown in Figure3.10(c). As expected, when the music tempo becomes faster, the higher order motion is omitted.

For a different aspect, how the variance of motion coincides with the music tempo was plotted, as shown in Figure 3.11. The top row shows the variance sequences of a joint angle at various musical tempos, and the bottom row shows a sequence of corresponding postures to the common local minimum. This figure shows that the local minimum of variance occurs at certain musical points, and in fact, those postures at those timings correspond to the keyposes defined by a human dancer.

We can summarize their findings as follows:

U-1 Keypose timings and postures will be preserved even if the musical tempo becomes faster.

U-2 High-frequency components of motion will decrease when the musical tempo becomes faster.



(a)

(b)

(c)

Figure 3.10: Comparison of mean joint angle trajectories at the original musical tempo (red), 1.2 times faster tempo (green), and 1.5 times faster tempo (blue).

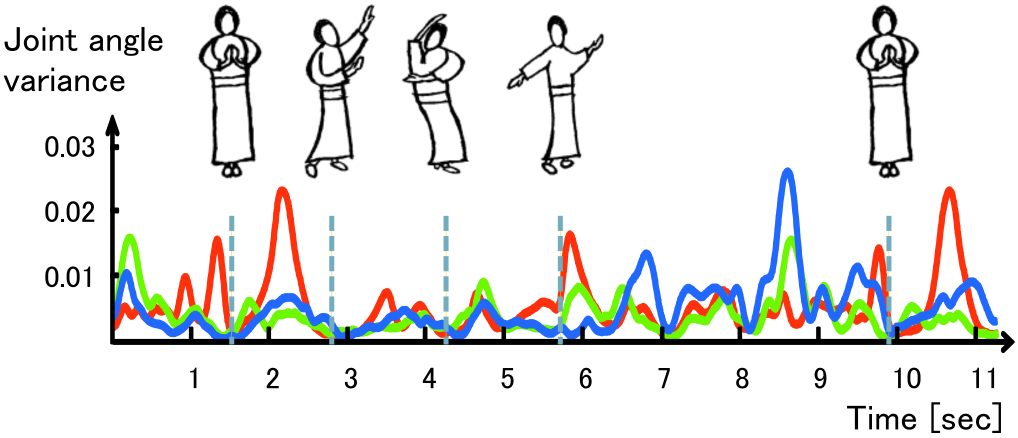


Figure 3.11: Comparison of variance sequences. Top row: postures corresponding to the common local minimum. Bottom row: variance sequences at the original musical tempo (red), 1.2 times faster tempo (green), and 1.5 times faster tempo (blue). Variance sequences for each speed tend to become local minimum at keyposes.

3.3 Imitation of Temporal Motion Styles

This section presents modification strategies of dance motions of a humanoid robot to accommodate various music speeds. The assumption before we begin this motion adjustment is that the human dance motions at a standard music tempo has been learned as a sequence of task, which we call *a task sequence*, based on the LFO method [NNK*07].

This section is organized as follow: First subsection presents temporal scaling algorithms, i.e., methods to create a cycle of dance motions for whole-body at a certain music tempo, from human motion at an original music tempo, based on keyposes. Then following subsection presents a method to realize a dancing-to-music capability for varying music tempos as an application; when robots dance to time-varying music tempo, they need to be able to change dancing speed satisfying required constraints. To achieve this we focus on a method to create robot motions on-line.

3.3.1 Temporal Scaling Algorithms Based on Motion Styles

This subsection presents a method to create whole-body motions for robots at a certain music tempo, from human motion at an original music tempo. When a music tempo becomes slower than the standard tempo, modifying the trajectories is relatively easy; we simply make each joint rotate more slowly by adjusting skill parameters of start/end timings and reconstructing the whole body motions based on the skill parameters. When the music tempo becomes faster than the standard tempo, a robot needs to make joints rotate faster in the same way. The payload of motors increases and sometimes may exceed the limit of the motor as shown in Figure 3.12. In order to avoid this situation, we derive modification strategies based on the observations in the previous section.

Keypose-based Integration of Lower-, Middle-, and Upper-body Motions

From the observation results in the previous section it would appear that the keypose is an essential factor for the dance performance as was found in L-1, M-1, and U-1. As stated earlier, a keypose is defined as a fixed posture of a dancer for the purpose of providing the viewers with expression and meanings of the dance. Other Japanese dances, such as Nou and Kabuki, also have keyposes that are often referred to as *Kime*, *Tome*, or *Mie*. In these traditional dances, dance masters regard it as very important to represent these keyposes

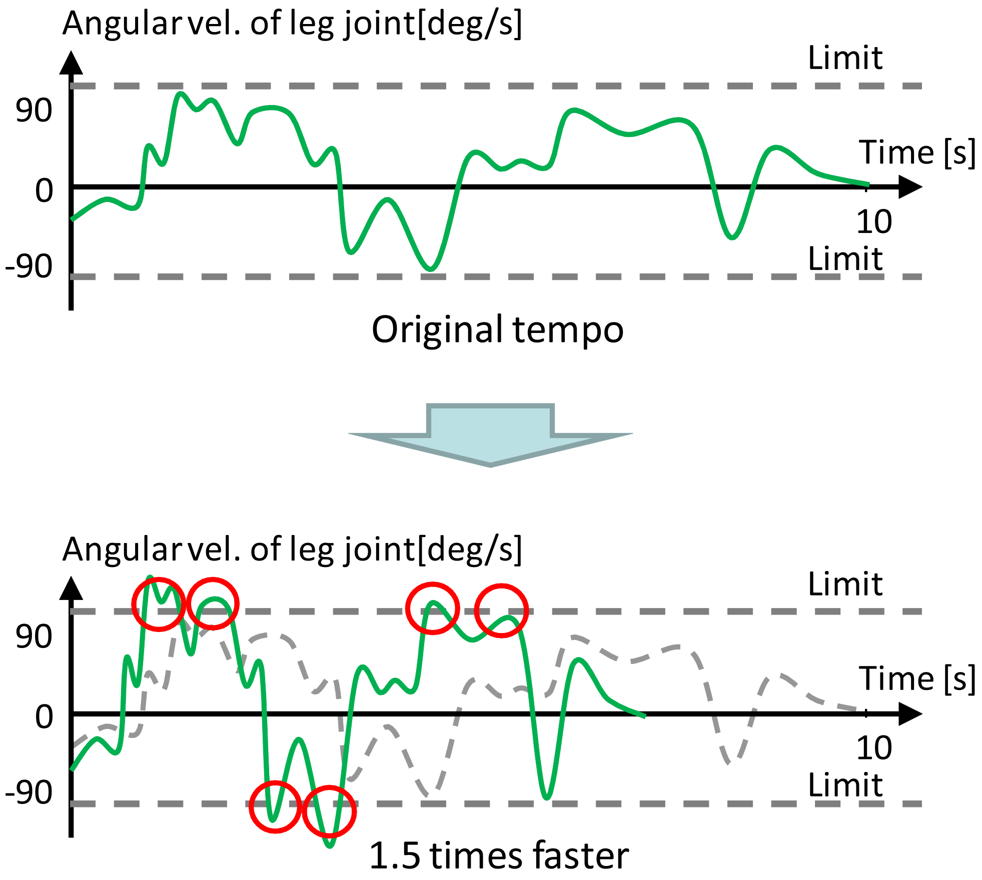


Figure 3.12: Excess of motor limitation by proportional temporal shrinkage of performance durations.

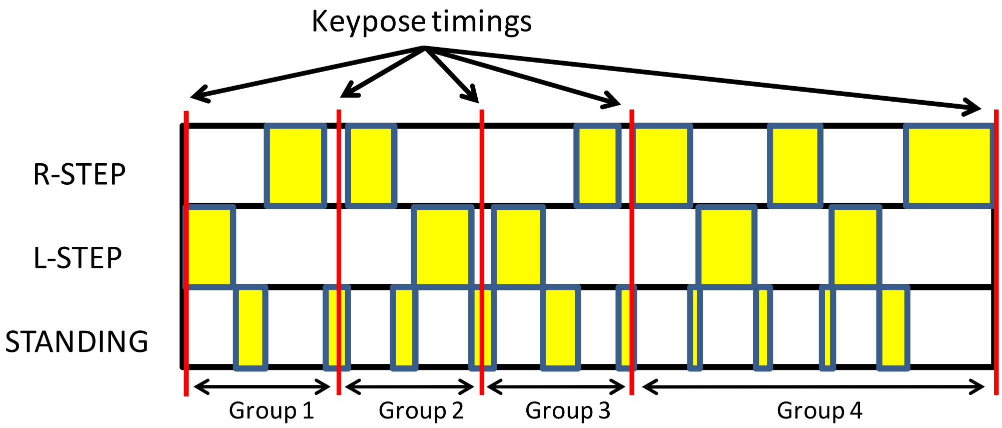


Figure 3.13: STEP and STAND tasks grouped by keypose timings. Tasks between two adjacent keypose times are considered as a group. Such a group consists of several STEP and STAND tasks

in appropriate timings. A dance performance with sophisticated keyposes is considered a skillful performance. The dancers tend to keep keypose postures in the appropriate relative timings in the cycle as much as possible, even if they have to attenuate the motions to follow a faster tempo of music.

Thus, we use these keyposes as anchor points when synchronizing the lower, middle, and upper-body motions for generating the whole-body motions on a humanoid robot.

Lower-body Motions

Our algorithm for lower-body motions consists of the following three phases:

- **Phase 1:** the task sequence of lower-body motions is temporally scaled proportionally, and STAND tasks are adjusted if they are shorter than a certain threshold (L-1).
- **Phase 2:** STEP tasks in which joint angular velocity exceeds the limit are detected by calculating inverse kinematics .
- **Phase 3:** skill parameters of the STEP tasks are modified by changing parameters of duration and stride (L-2,3,4).

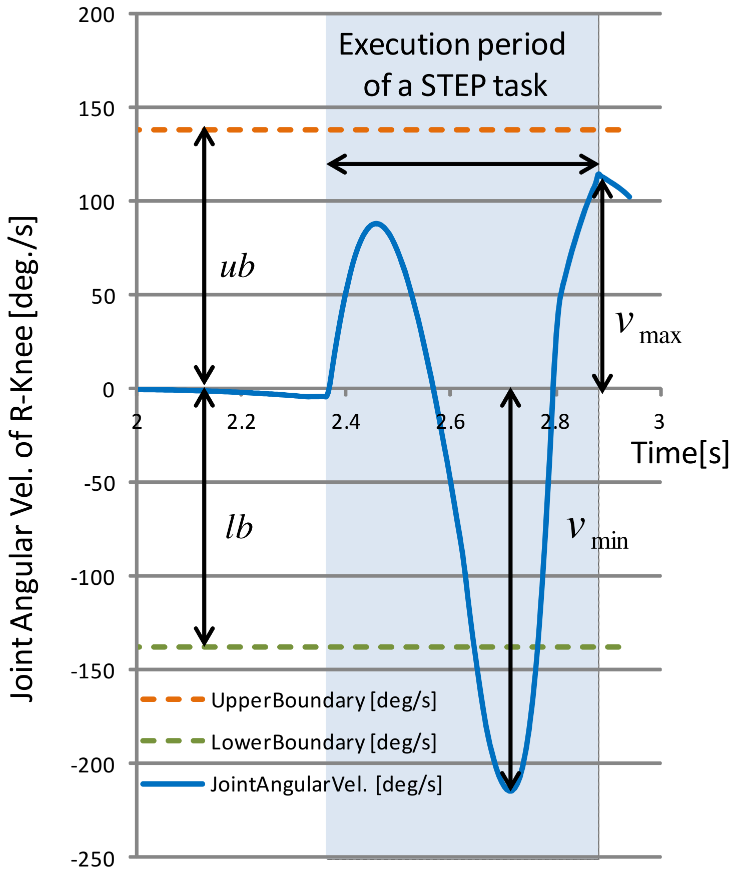


Figure 3.14: Joint angular velocities of a knee joint for a STEP task.

Phase 1: This operation is based on the observation of L-1. As the first operation, proportional temporal shrinkage of performance durations along with the music tempo is applied to the task sequence of lower-body motions. The start and end timings for all tasks are divided by the speed rate of the musical tempo. As a result, all the lower-body motions corresponding to keyposes occurred at the appropriate music timings. However, such shrinkage causes an overload of joint motors. Avoidance of such overload is the issue in this subsection.

Tasks between two adjacent keypose times are considered as a group. Such a group consists of several STEP and STAND tasks as shown in Figure 3.13; the start of the first STAND task and end timing of the last STAND task among the group are fixed so as to maintain the keypose timings. Note, however, that, execution periods of STAND tasks are need to be longer than certain threshold for stable balance maintenance. The minimal length of the execution periods is called *Minimum Non-Step Interval* (MNSI), and we use empirical value of 0.07 sec. MNSI is a time interval required in order for the Zero Moment Point (ZMP) can move stably in the support polygon during STAND tasks. If the execution periods of STAND tasks are not enough as a result of the first operation, Each of period are preferentially extended by adjusting start/end timings of adjacent STEP tasks. For the first and the last STAND task among the group, we use the half of MNSI exceptionally. Although modification may change the end timing of tasks around keyposes, the modification is not noticeable because MNSI is significantly small.

Phase 2: For all the STEP tasks in a group, the system increases the speeds of the joint motors to achieve shortened execution periods. First, the speeds of the joint motors are computed using the inverse kinematics method at each newly created start and ending timing. Then, those speeds are examined to determine whether they exceed the motor capability limit or not. This inspection is executed on all of joints of both legs. For each of STEP tasks, a velocity excess ratio E is calculated as follow:

$$e_j = \max\{\frac{v_{max}}{ub}, \frac{v_{min}}{lb}\}, \tag{3.1}$$

$$E = \max\{e_j, j = 1, 2, \cdots, numJoints\}, \tag{3.2}$$

where definitions of v_{max}, v_{min}, ub, and lb are given in Figure 3.14.

Phase 3a: This process is based on the observation of L-2. From among those STEP tasks in the group, the duration of the tasks, with the exceeding speed limitation, are extended so as to satisfy the motor limitation. This is achieved by first reducing the period of the following STAND task. If this is not enough, the STEP and STAND tasks with capacity allowance in the group are considered as

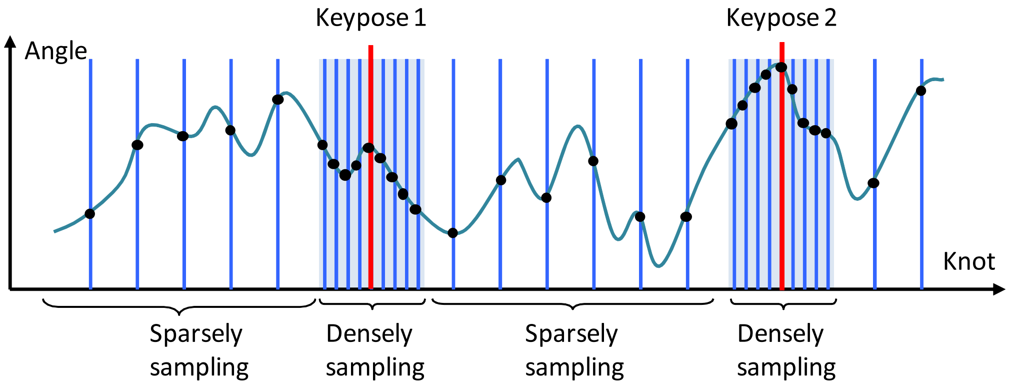


Figure 3.15: Sampling method to consider keypose information for hierarchical motion decomposition [SKNI07, SI08]. Vertical lines represent sampled time instants, and a dashed curve represents ground truth of a continuous joint angle trajectory. Data sampled by our method (black dots) are used.

candidates for duration reduction. This operation is conducted iteratively along the descending order of the exceeding tasks within the group.

Phase 3b: This process is based on the observations of L-3 and L-4. If this period adjustment is not enough, the strides and trajectories of all the exceeding tasks are reduced iteratively so as to satisfy the motor limitation.

$$\boldsymbol{r}_f \leftarrow \alpha(E)(\boldsymbol{r}_f - \boldsymbol{r}_0) + \boldsymbol{r}_0, \tag{3.3}$$

$$h \leftarrow \alpha(E) \cdot h, \tag{3.4}$$

where α is given according to the velocity excess ratio provided by Phase 2. $\boldsymbol{r}_0$, $\boldsymbol{r}_f$, and h represents a starting position, a landing position, and a maximum height of the swing foot while stepping respectively. This stride reduction satisfies the limitation, because eventually the strides and a maximum height of the swing foot will become under an executable threshold, and the robot is simply standing with upper body motions only.

Upper-body Motions

For generation of upper-body motions, we employ Shiratori *et al.*'s method based on observation of temporal motion styles for upper-body [SKNI07, SI08].

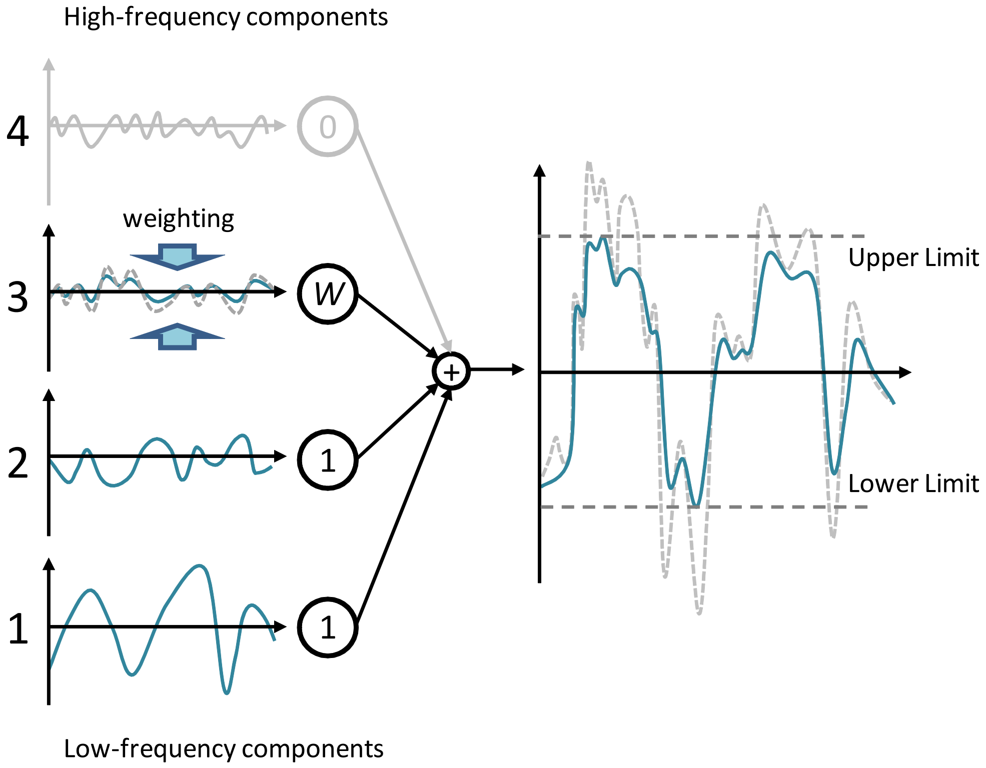


Figure 3.16: Skill parameter adjustment for temporal scaling of upper body motion [SKNI07, SI08]. This adjustment process gradually decreases the weighting factors from the finest layer of the hierarchical B-Spline.

Joint trajectories are represented by hierarchical B-splines. Here, in order to preserve posture information of keyposes in the following operations, The algorithm sample joint trajectories around keyposes more densely than those in other parts. This sampling method is illustrated in Figure 3.15. Then, proportional temporal scaling, fitting to the new music tempo, is applied to each layer of B-spline representation so that the resulting motion is consistent with the new music tempo. This operation satisfies the observation U-1; whatever the music tempo is, keyposes occur at particular music timings. However, the resulting motion may exceed motor capability. Thus, this excess joint speed is amended in the hierarchical manner.

The algorithm examine the motor capacity by examining the layers of the hierarchical B-spline iteratively from higher to lower layers. It recalculate the motor load by reducing the amplitude of the highest level of the hierarchical B-spline. If it cannot achieve the motor load within the capacity by setting the amplitude of this highest level to zero, it repeat the same operation in the next layer of the hierarchical B-spline, iteratively. This adjustment is illustrated in Figure 3.16.

Middle-body Motions

As we explained above, we consider vertical and horizontal movements separately for middle-body motions.

For vertical movements, abstracted as SQUAT tasks, we found that the most important factor is to maintain the timing of squat (M-1) while adjusting the depth (M-2). Thus, along with lower and upper-body motions, proportional temporal shrinkage of performance durations along with the music tempo is applied to the task sequence of middle-body motions. The start and end timings for all tasks are divided by the speed rate of the musical tempo.

Then SQUAT tasks in which joint angular velocity exceeds the limit are detected by calculating inverse kinematics. Unlike in the case of lower-body motions, SQUAT tasks in which acceleration of the waist link exceeds that of free fall are detected as unrealistic tasks. The depth of SQUAT, with the exceeding speed limitation and acceleration of free fall, is gradually reduced up to a certain value to satisfy the motor limitation and dynamic limitation.

If the reduction of the depth to the limit does not provide the allowance of motor capability, then we expand the period of the SQUAT task. If this is still not sufficient, we eliminate the SQUAT task from the task sequence. These actions are not based on the observation, but it is inevitable to avoid exceeding the payload limit of the motors.

For horizontal movements, the desired horizontal trajectories of the waist are calculated using a ZMP compensation filter [NkK*02]. This calculation is done considering lower- and upper-body motions.

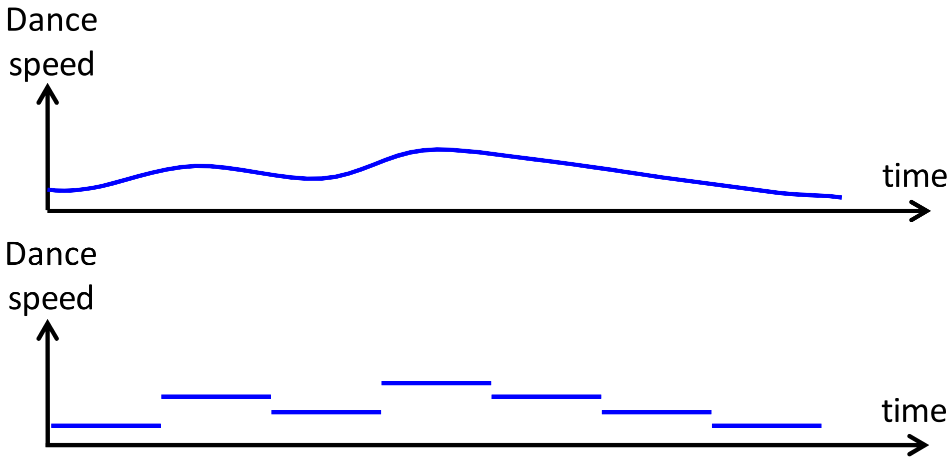


Figure 3.17: Two options for how to change the speed of dancing.

3.3.2 Dancing-to-music Capability for Varying Music Tempos

In previous subsection, we presented algorithms to create dance motion variations at arbitrary music tempo, from human motion at an original tempo. As an application of the algorithms, this subsection presents a strategy for on-line dance motion generation on the assumption that a robot is dancing to music with time-varying tempo such as live-music. Our method make it possible for the robot to modify timings of dance motions, even if the robot motions are not synchronizing with music tempos. Such dynamic interactions with time-varying tempos of a given music piece is the focus of this subsection. Here, we assume that music tempos will increase/decrease by 20 percent at most.

Keypose-based On-line Generation

When music tempos are changed while a robot is dancing, the robot also need to change the speed of dance motions, to keep up with the music. To change the speed of dancing, there are at least two options for how to change the speed; continuous one or discrete one as shown in Figure 3.17. Continuous means that the robot can change the speed continuously, even during stepping.

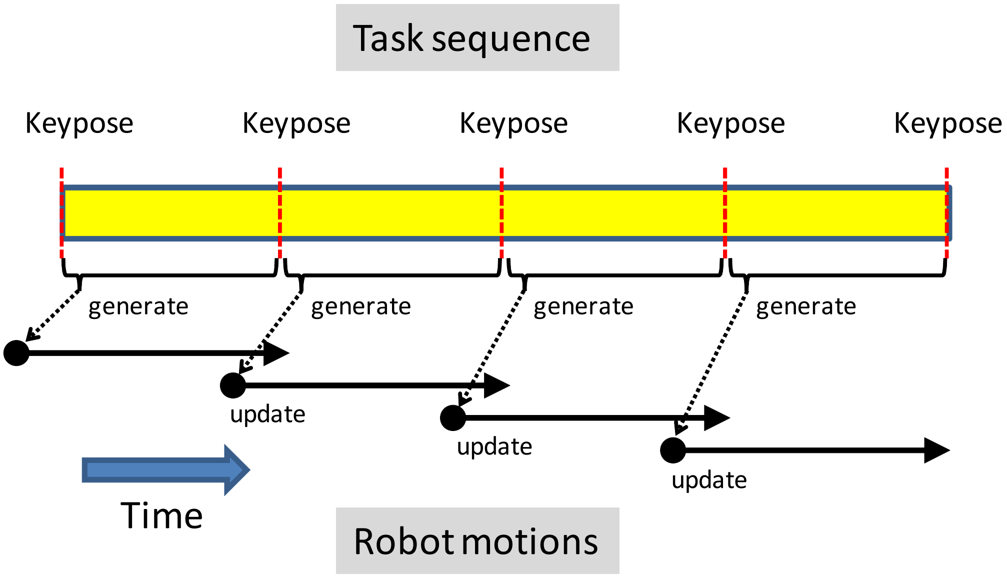


Figure 3.18: On-line dance motion generation based on keyposes. Gray broken curves are trajectories generated for each segment. To avoid discontinuity at keyposes, we modify trajectories as shown by green curves.

Discrete means that the robot update the speed at some intervals, beats or bars for instance.

In this research, we employ the latter one. Because, from the observation, we found that keyposes are essential in the dance and assume that a dancer changes the speed of dance based on keyposes too. Additionally our strategies for dance motion generation described in this thesis is based on keyposes, which correspond to music beats. As described above, we employed keyposes for segmentation of a continuous dance motions and use as anchor points for integration of whole-body dance motions. Thus, we generate motion segments, defined based on keyposes, separately and update on-line as illustrated in Figure 3.18.

On the other hand, two motion segments generated for different music tempos are need to be connected continuously. A method for such connection is described as follow.

Continuity of Motion Trajectories

Upper-body Motions: Our method is illustrated in Figure 3.19. Now, we generate upper body motions for the segment k at a certain music tempo and connect to motions for the previous segment $k-1$ at a keypose timing. However, as a result of generation based on Shiratori *et al.*'s method [SKNI07, SI08], a discontinuity of trajectory may occur at the keypose timings. Therefore we need to modify the trajectory for segment k to avoid such discontinuity and to connect continuously.

Constraints for modification here are only C^2 continuous and joint limitations of the robot. We use a trajectory tracking method proposed by Pollard *et al.* [PHRA02] and generate a new trajectory (the green curve in Figure 3.19) within the joint limitations. For all time step i in k-th segment, a new trajectory $\theta_{F,i}$ is obtained as follow:

$$\dot{\theta}_i = \theta_i - \theta_{i-1}, \tag{3.5}$$

$$\ddot{\theta}_{F,i+1} = 2k_s\left(\dot{\theta}_i - \dot{\theta}_{F,i}\right) + k_s^2\left(\theta_i - \theta_{F,i}\right), \tag{3.6}$$

$$\dot{\theta}_{F,i+1} = \max\left(\dot{\theta}_L(\theta_{F,i}), \min\left(\dot{\theta}_U(\theta_{F,i}), \dot{\theta}_{F,i} + \ddot{\theta}_{F,i+1}\right)\right), \tag{3.7}$$

$$\theta_{F,i+1} = \theta_{F,i} + \dot{\theta}_{F,i+1}, \tag{3.8}$$

where θ_i represents a original trajectory (the gray broken curve in Figure 3.19), and the discontinuity frame corresponds to a frame with $i = 0$. Additionally, our refinement on a stiffness parameter k_s increase it while $i < t'$. This avoid precipitous tracking. $\dot{\theta}_L(\theta_{F,i})$ and $\dot{\theta}_U(\theta_{F,i})$ represent the lower and upper limits of the joint angular velocity at $\theta_{F,i}$ given as follow:

$$\dot{\theta}_U(\theta_{F,i}) = \begin{cases} \dot{\theta}_U^{max} \cdot \frac{\theta_U - \theta_{F,i}}{\alpha} & (\theta_U - \theta_{F,i} \leq \alpha) \\ \dot{\theta}_U^{max} & (otherwise) \end{cases} \tag{3.9}$$

$$\dot{\theta}_L(\theta_{F,i}) = \begin{cases} \dot{\theta}_L^{min} \cdot \frac{\theta_{F,i} - \theta_L}{\alpha} & (\theta_{F,i} - \theta_L \leq \alpha) \\ \dot{\theta}_L^{min} & (otherwise), \end{cases} \tag{3.10}$$

where θ_L and θ_U represent the lower and upper limits of the joint angle, respectively. $\dot{\theta}_U^{max}$ and $\dot{\theta}_L^{min}$ represent the minimum and maximum values of $\dot{\theta}_U^{max}$ and $\dot{\theta}_L^{min}$, which are depend on the actuator.

Lower-body Motions: In lower-body motions, we generate foot trajectories for a new segment. In leg task models [NNK*07], a trajectory of a swing foot in

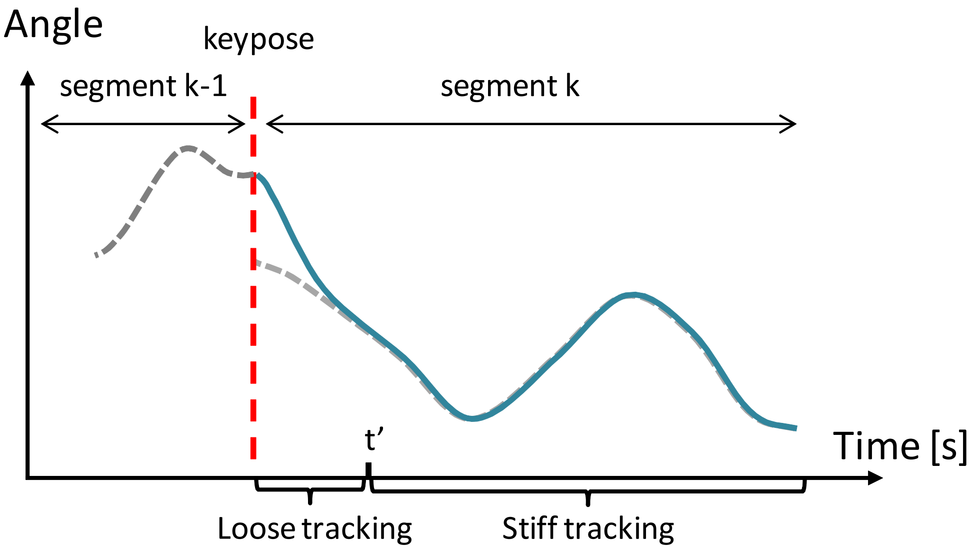


Figure 3.19: Connection of upper body motions.

a STEP task is reconstructed using a smooth interpolation function based on a cubic polynomial that passes along three points computed from skill parameters: the starting point $(t_0, \boldsymbol{r}_0)$, the middle point $(t_1, \boldsymbol{r}_1)$, and the landing point$(t_f, \boldsymbol{r}_f)$. This function is expressed as follows:

$$\boldsymbol{f}_n \langle (t_0, \boldsymbol{r}_0), (t_1, \boldsymbol{r}_1), (t_f, \boldsymbol{r}_f) \rangle (t). \tag{3.11}$$

There are two ways to determine the middle point. One is the case of a *normal step*, in which a captured trajectory of the swing foot is similar to a step in usual walking. For a normal step, the middle point calculated by the following equations is used:

$$t_1 = \frac{t_0 + t_f}{2}, \tag{3.12}$$

$$\boldsymbol{r}_1 = \left(\frac{r_x^0 + r_x^f}{2}, \frac{r_y^0 + r_y^f}{2}, h \right)^T, \tag{3.13}$$

where h is a predefined value as the normal step height. The other case is when the captured trajectory differs largely from the trajectory of a normal step.

For such a *stylistic* action, the middle point is determined directly from the foot position of human motion data at the timings of the middle point. In our implementation, these two cases are distinguished by a degree of the difference between the middle point of the captured trajectory and the point calculated by Equation (3.13).

In this context, constraints on generation will be consistency of foot positions; the starting point $(t_0, \boldsymbol{r}_0)$ of the first STEP task in the new segment of task sequence is computed from the finishing state of the last STAND task in the previous segment.

Middle-body Motions: Vertical positions of the middle-body are reconstructed using skill parameters like foot trajectories described above. Velocities and accelerations at keyposes are always 0, therefore, these elements are satisfy the boundary conditions in the same way as lower-body.

The horizontal positions of the middle-body, on the other hand, is not becoming zero at keyposes except for start/end timings of the whole dance motions. This is because the balance maintenance of biped robots is basically different from that of human beings and based on ZMP. The ZMP positions, while standing with foot, need to be within the support polygon defined using a foot or both feet contacting floor. The transition of ZMP to under the next support foot is always executed while STAND tasks. This transition is controlled indirectly by translation of the horizontal positions of the middle-body.

Considering the observation, it might be best if the robot can explicitly stop the horizontal movements of the middle-body. To translate horizontal positions within the execution period stably, however, it is not in a realistic way.

Thus we connect the horizontal positions continuously without stopping at keyposes. And the ZMP need to be within the support polygons in this connection.

In our implementation, as shown in Figure 3.20, positions of the middle-body from before the beginning of the last STEP task to the end of next segment (the green curve) are computed for the next segment k. The positions are actually computed by modifying an initial positions, which are calculated from foot positions, using an approximate computation method proposed by Nishiwaki *et al.* [NkK*02]; The ZMP positions are, first, computed based on the initial positions of the middle-body and then positions of the middle-body are modified iteratively so that the ZMP positions are corresponds to the desired positions

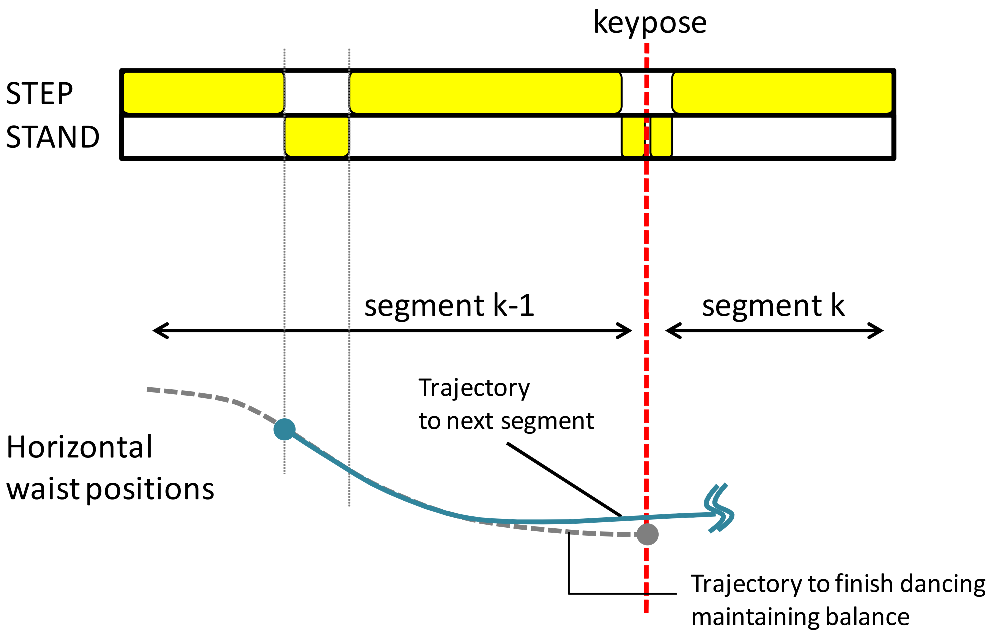


Figure 3.20: Generation of horizontal waist trajectory.

within the support polygon. The minute discontinuity at the beginning frame of the green curve, as a result of the approximate computation, is interpolated using an interpolation function based on the quintic polynomial equation.

The quintic polynomial equation is as follow:

$$r(t) = 6t^5 - 15t^4 + 10t^3, \tag{3.14}$$

Strictly speaking, as a result of this interpolation, the ZMP positions are actually not correspond to the desired position. However, the error is within the acceptable range for actual use as shown in our experiments in the following section.

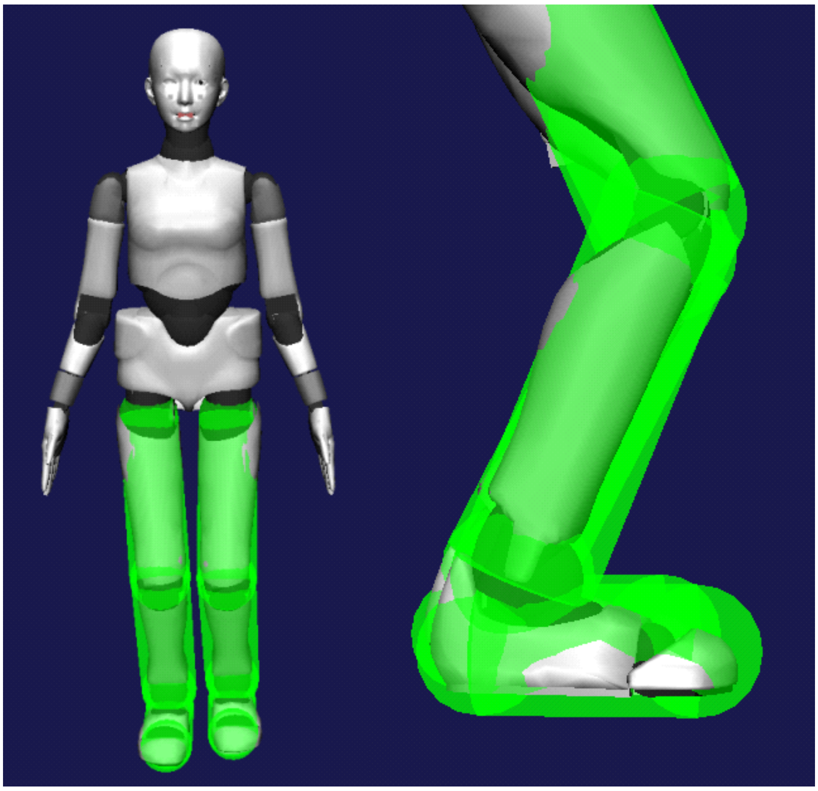


Figure 3.21: Capsule-shaped volumes for collision detection.

Physical Constraints

When a robot execute a task on-line, generated motions must be feasible for the humanoid robot. One of factors to be considered for feasibility is self-collisions. Especially, to imitate stylistic human motions such as dance, body links of the robot need to be always close each other in peril of collisions. This make our on-line generation more complicated comparing to other researches such as on-line working pattern generation or real-time control of dance patterns preliminarily designed for the robot.

We assume that task sequences executable for the original music tempos are given as input via previous works. However, adaptation to a certain music tempo modify horizontal positions of the waist link for balance maintenance, and as a result, self-collision may occur. Therefore we need to inspect the generated

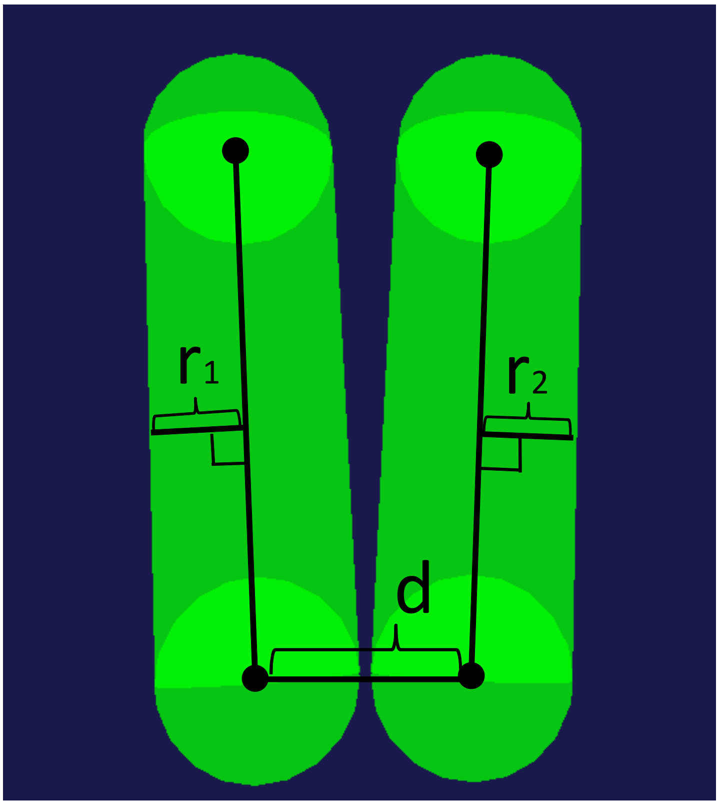


Figure 3.22: Parameters for collision detection.

motion segment for collisions and modify the skill parameters to avoid it.

To execute collision avoidance in a low-cost way, we approximate the link to be inspected by capsule-shaped virtual volumes. In this research we handle the self-collision between legs only, because we assume that modification strategy for upper-body is free from care about collisions. Six links of lower-body is wrapped by capsule-shaped virtual volume as shown in Figure 3.21, and we defined link pairs to be checked collision as follow:

Pair 1 Right thigh - Left thigh

Pair 2 Right thigh - Left shin

Pair 3 Right shin - Left thigh

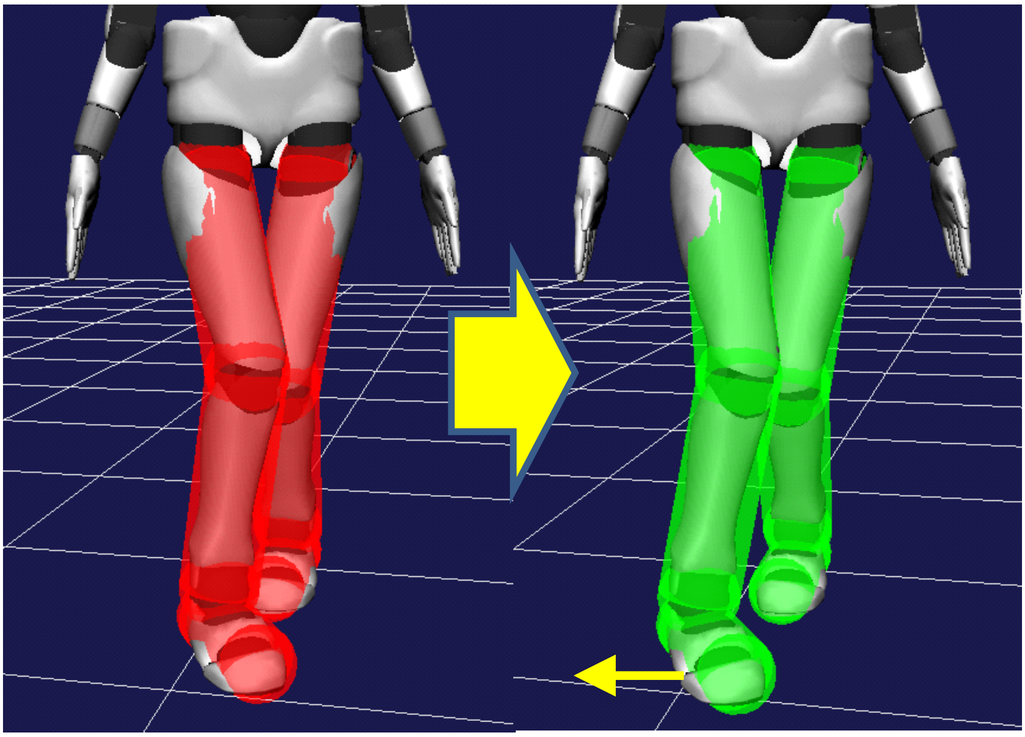


Figure 3.23: An example of the result of collision avoidance.

Pair 4 Right shin - Left shin

Pair 5 Right shin - Left foot

Pair 6 Right foot - Left shin

Pair 7 Right foot - Left foot

Geven link positions for each leg task, we inspect collision between two capsule-shaped volumes by calculating $c_p = (r_{1,p} + r_{2,p}) - d_p$, where d_p, $r_{1,p}$, and $r_{2,p}$ are the minimal distance between axes and radii of each volume of the p-th pair respectively as shown in Figure 3.22.

When a collision between a pair of capsule-shaped volumes is detected at a certain task, we compute the minimal translation of swing foot under the constraints as follow:

$$\begin{aligned} \underset{\dot{\mathbf{r}}}{\text{minimize}} \quad & ||\dot{\mathbf{r}}||^2 \\ \text{subject to} \quad & \mathbf{J}_d \cdot \dot{\mathbf{r}} \leq \mathbf{C}, \end{aligned}$$

where $\dot{\mathbf{r}}$ represents the minimal translation of the foot. $\mathbf{J}_d$ represents a Jacobian determinant for a collision vector $C = c_1, c_2, ..., c_7$.

The minimal translation of the foot is reflected on the skill parameters of the corresponding task. Then we reconstruct the robot motions using the updated skill parameters.

3.4 Experiments

This section reports the implementation of the system on a physical humanoid robot for validating our proposed algorithm.

3.4.1 System Overview

Overview of our implementation of the system is shown in Figure 3.24. Given a dance motion and information about the original music, our system automatically detects each task in the motion and extracts task sequences that is based on the previous method [NNK*07, SKNI07, SI08]. The speed rate and segments of task sequence are passed to the system as inputs.

The temporal scaling algorithm 1 changes the speed of the dance motion by adjusting skill parameters of each task in the segments of task sequence. A task reconstructor create elements of motions, such as foot positions, waist positions, desired ZMP potions, and joint angles of upper-body, for each task in the segments of task sequence. Here, horizontal waist positions are calculated based on foot potions and modified by a ZMP compensation filter [NkK*02] so that actual ZMP positions are close to desired ZMP positions.

Joint angles and link positions of the whole-body is reconstructed from the elements of motions using inverse kinematics. Then a fault inspector detects collisions and overload of actuators and refine the skill parameters based on algorithms described in this chapter. Task reconstructor create elements of motions again using the refined skill parameters. Those process are repeated while a fault inspector find faults in the whole-body motions.

To reduce the computation time of the iterative process, we skip a ZMP compensation filter while faults are found in the motions without the filter. When a fault inspector find no faults, we enable the ZMP compensation filter to get down to the wire. However, modification of waist positions may occur the faults again. To avoid this situation, we use an algorithm described in Algorithm 3.1. Using this algorithm, we obtain initial positions of the waist more similar to a result of modification by a ZMP compensation filter than that calculated using previous works as shown in Figure 3.25, 3.26.

When the generation process is finished, finishing states of the motions are returned to a task reconstructor as the next initial conditions for next segments of task sequence. This system is implemented as a Plug-in of Choreonoid [Nak12], a robot simulation tool.

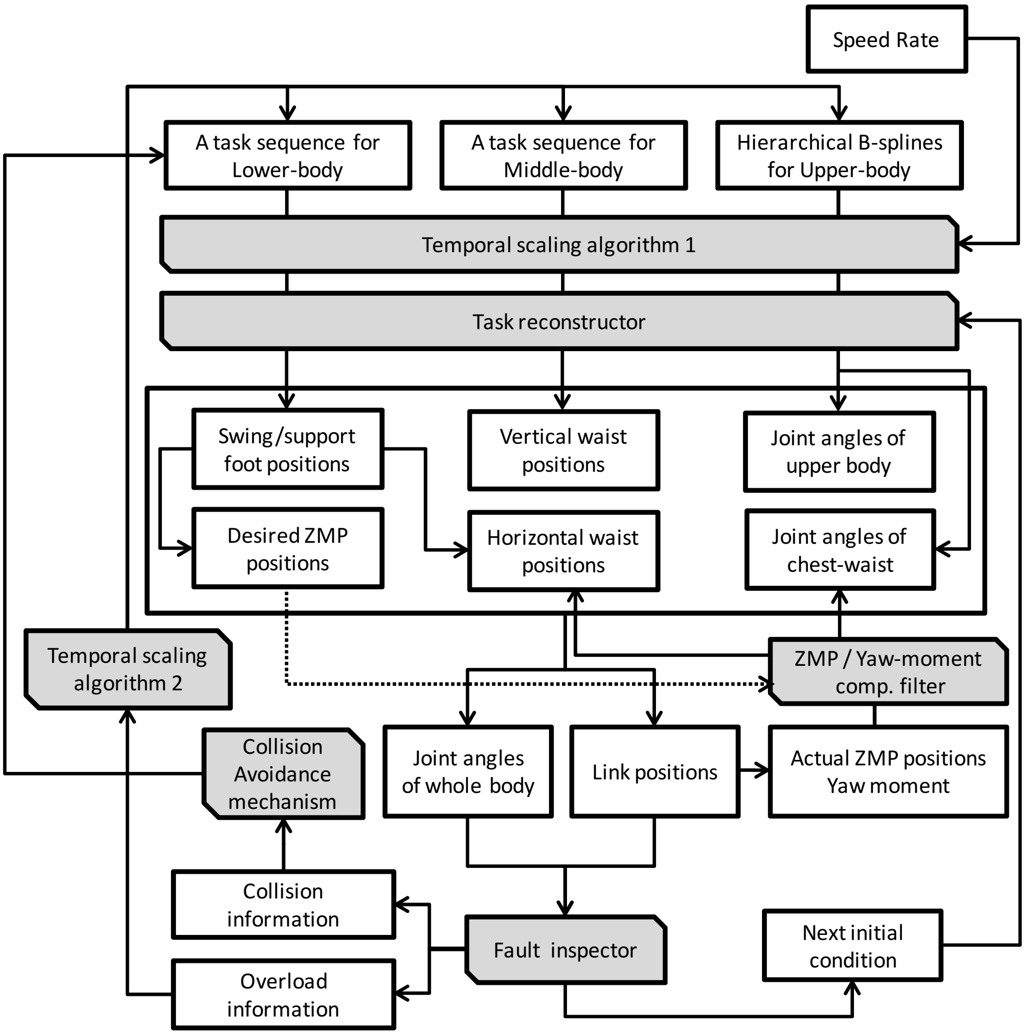

Speed Rate
A task sequence for Lower-body
A task sequence for Middle-body
Hierarchical B-splines for Upper-body
Temporal scaling algorithm 1
Task reconstructor
Swing/support foot positions
Vertical waist positions
Joint angles of upper body
Desired ZMP positions
Horizontal waist positions
Joint angles of chest-waist
Temporal scaling algorithm 2
ZMP / Yaw-moment comp. filter
Collision Avoidance mechanism
Joint angles of whole body
Link positions
Actual ZMP positions Yaw moment
Collision information
Fault inspector
Next initial condition
Overload information

Figure 3.24: System Overview.

Algorithm 3.1 Initial waist positions

```
input: R-Foot positions $\boldsymbol{p}_r$
input: L-Foot potisions $\boldsymbol{p}_l$
input: an array of STEP task $\boldsymbol{S}$
input: end timing of whole motions $t_{last}$
local: weighting factor $w$
local: start timing of a STEP task $t_{st}$
local: mid timing of a STEP task $t_{mid}$
local: interpolation function based on a cubic polynomial $f$
output: waist positions $\boldsymbol{p}_{wt}$
$f \leftarrow \text{SetPoint}(0, \frac{\boldsymbol{p}_r.at(0)+\boldsymbol{p}_l.at(0)}{2})$
for $i$ in 1 : length $\boldsymbol{S} - 1$
$t_{st} \leftarrow \boldsymbol{S}[i].t_{st}$
$t_{mid} \leftarrow \boldsymbol{S}[i].t_{mid}$
  if $\boldsymbol{S}[i].support == leftfoot$ then
    $f \leftarrow \text{SetPoint}(t_{mid}, \boldsymbol{p}_l.at(t_{st}) \cdot w + \boldsymbol{p}_r.at(t_{st}) \cdot (1 - w))$
  else
    $f \leftarrow \text{SetPoint}(t_{mid}, \boldsymbol{p}_r.at(t_{st}) \cdot w + \boldsymbol{p}_l.at(t_{st}) \cdot (1 - w))$
end for
$f \leftarrow \text{SetPoint}(t_{last}, \frac{\boldsymbol{p}_r.at(t_{last})+\boldsymbol{p}_l.at(t_{last})}{2})$
$\boldsymbol{p}_{wt} \leftarrow \text{InterpolatePoints}(f)$
```

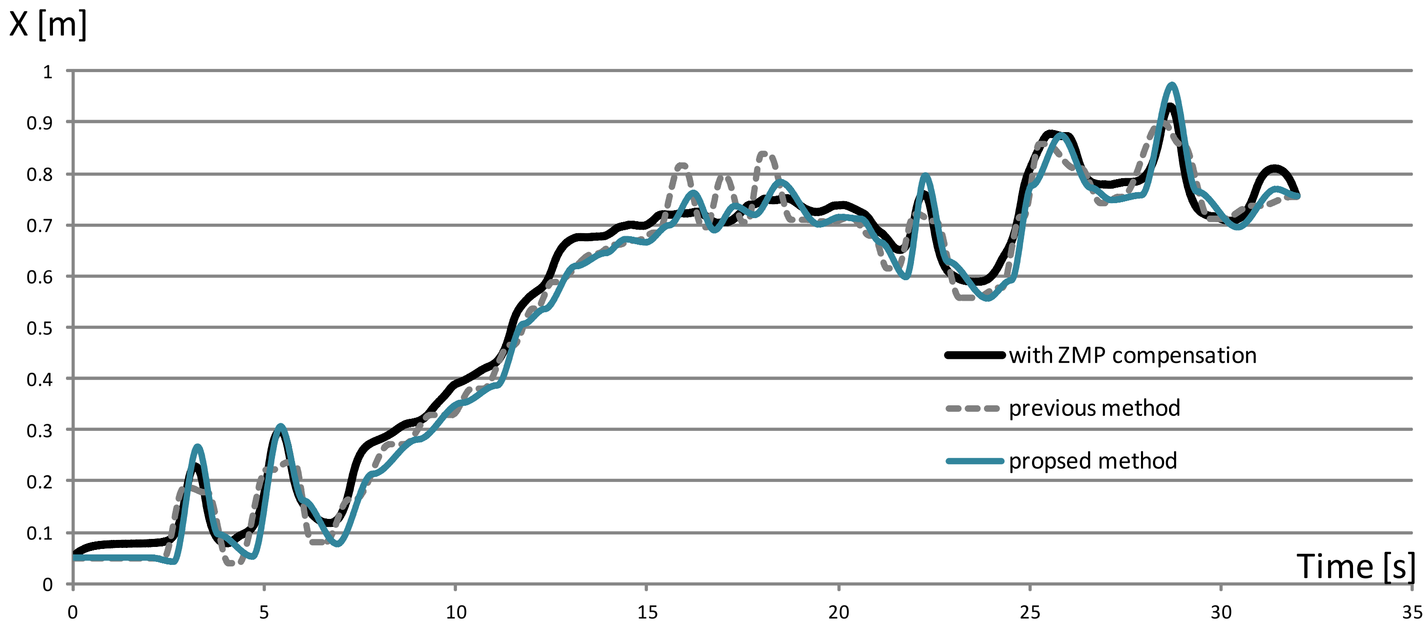


Figure 3.25: X coordinate of waist positions.

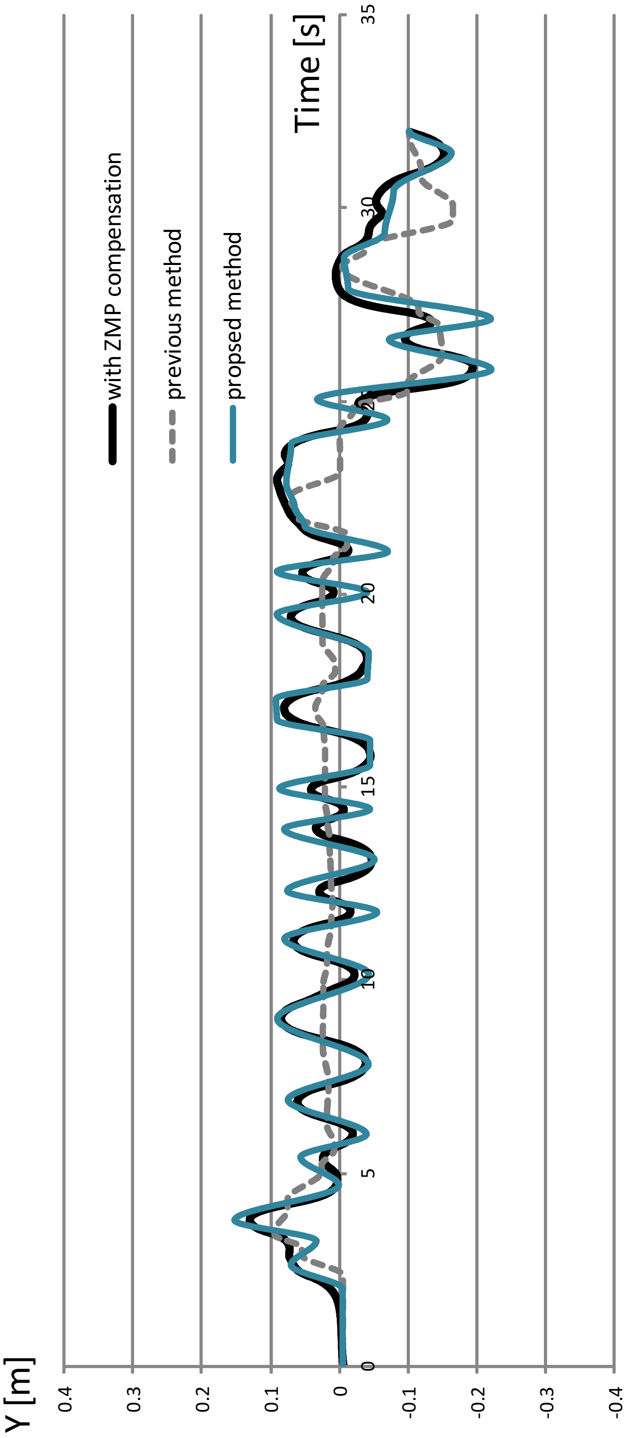


Figure 3.26: Y coordinate of waist positions.

3.4.2 A Robot Platform and a Target Dance

In our experiments, HRP-4C [KKM*09] was selected as a physical robot platform as shown in Figure 3.27. The height and the weight are 1.58[m] and 47[kg] including the weight of batteries in the waist link. Totally 44 DOFs are available. We used 12 DOFs for lower-body motions and 13 DOFs for upper-body motions. Joint angles and angular velocities for each joint of generated motions were always within the 80 percent and 65 percent of the specific limit of actuators, respectively. We always set this value for the sake of safety.

We chose the Don-pan dance, a famous folk dance in Akita prefecture, as a target dance for experiments. The Don-pan dance also consists of cyclic patterns, each of which takes about 32 seconds. Figure 3.29 shows the reference of keyposes in the Don-pan dance captured using a magnetic motion capture system, MotionStar Wireless. Along with the hierarchical B-splines for upper-body motions, a task sequence of lower- and middle-body motions extracted from a skillful dancer as shown in Figure 3.29.

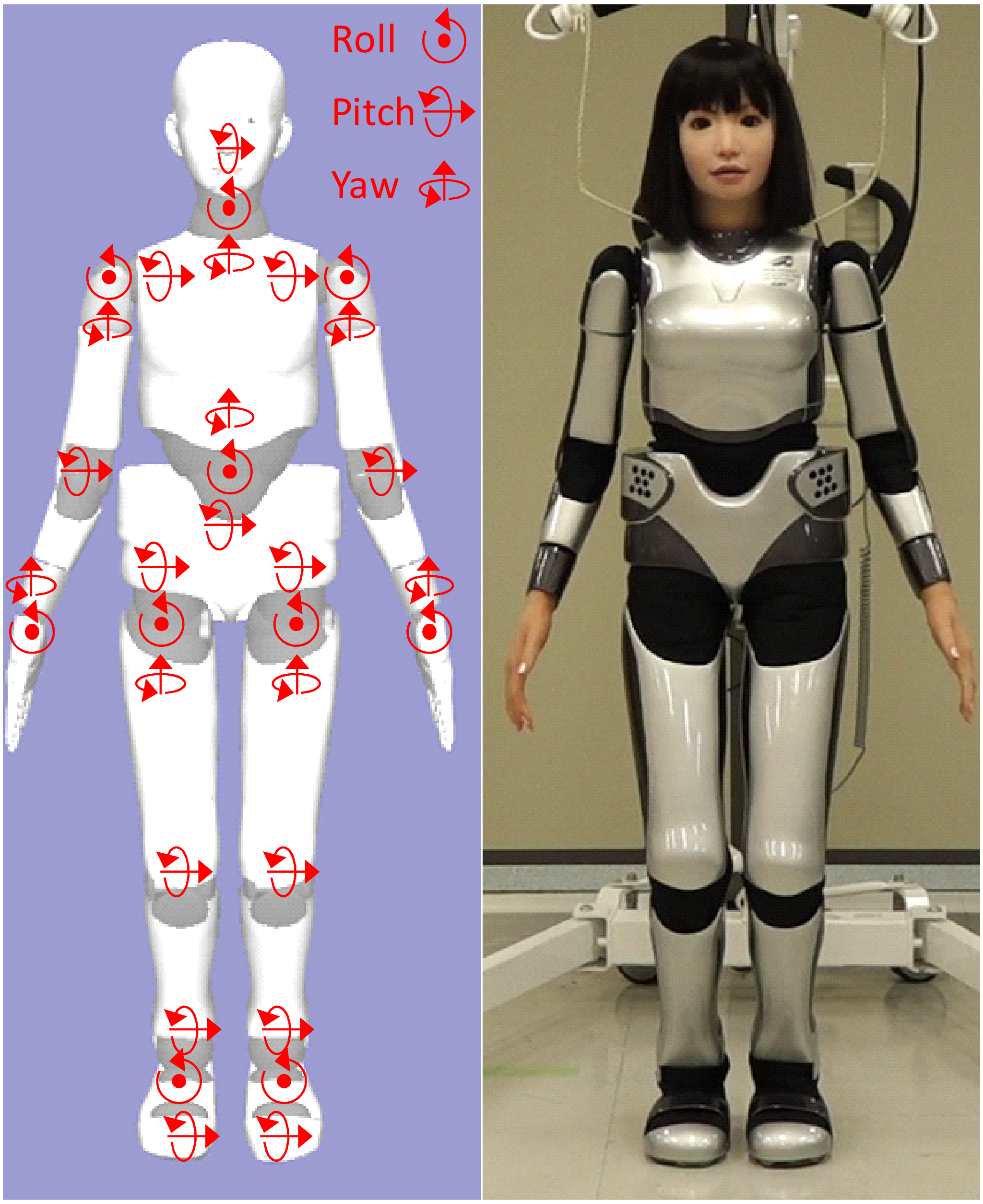


Figure 3.27: A robot platform: HRP-4C. The height and the weight are 1.58[m] and 47[kg] including the weight of batteries in the waist link. Totally 44 degree of freedom are available.

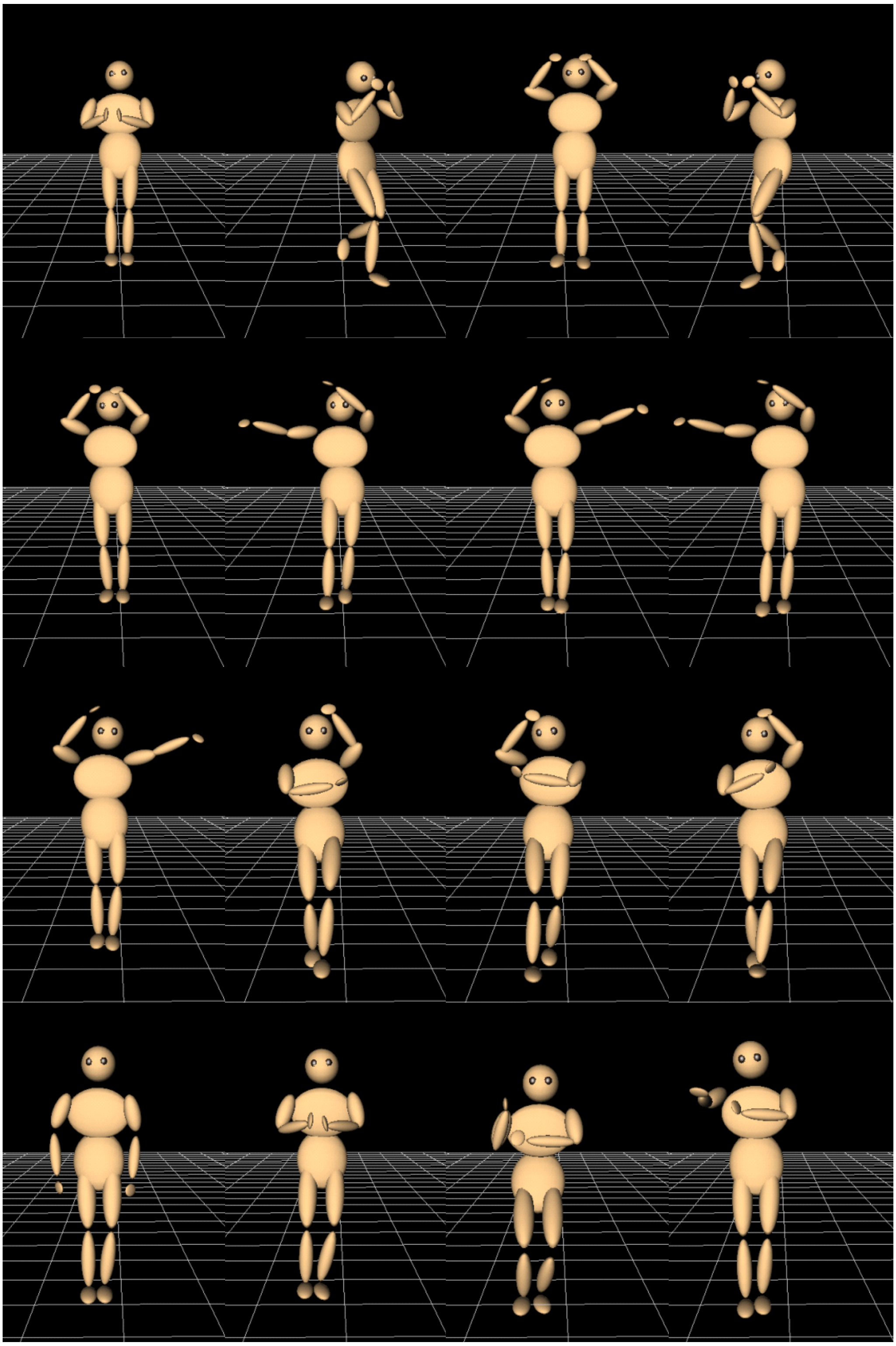


Figure 3.28: Examples of keyposes in the Don-pan dance.

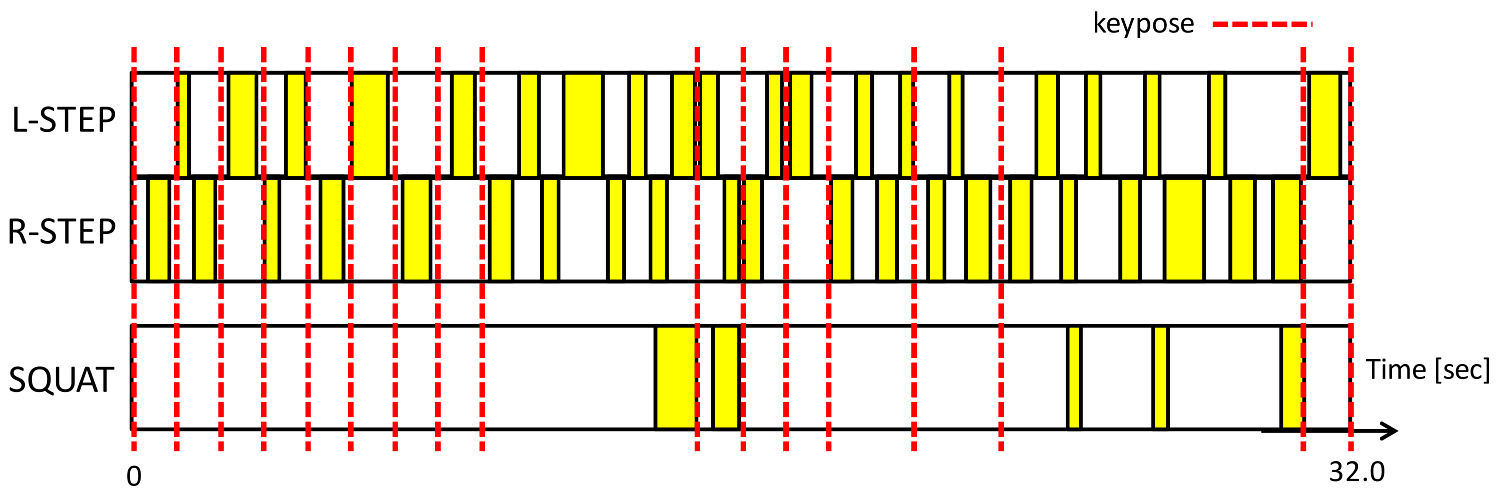


Figure 3.29: A task sequence of don-pan dance for lower- and middle-body motions at the original music tempo.

3.4.3 Preliminary Generation in Off-line

For preliminary experiments, whole-body motions of the Don-pan dance were generated to increase/decrease tempos of music. Target tempos of music were fixed to the original, 0.85 times and 1.2 times faster than the original, respectively. In this experiments, the whole sequences of task are passed to the system at a time.

Figure 3.31, Figure 3.30, and Figure 3.32 show sequences of the Don-pan dance motions whose tempos are fixed to the original, 0.85 times and 1.2 times faster than the original, respectively. Each poses in figures are correspond to the reference of keyposes shown in Figure 3.29. Each sequence of the dance start from the far left in the top row and end at the far right in the bottom row in a Figure. The robot expressed the keyposes using the whole-body in appropriate timings in the sequence, and provided viewers with an artistic dance pattern in which upper-body and leg motions were fully harmonious. Although the dance motions are modified separately according to musical tempos using different strategies, differences between keyposes at each music tempo are difficult to find. In both music tempos, our proposed system generated feasible motions within the joint limitations, and the HRP-4C could perform without falling down.

The joint angular velocities in the above two experiments are shown in Figure 3.34 and Figure 3.33. Velocity sequences of the right knee angle generated for the original tempo and 0.85 times faster tempos than the original one are shown. The green lines represent the joint angular velocities of the motions generated using the proposed system. The orange lines represent log data of the execution recorded by a sensor of the robot. The gray lines represent the upper/lower limits of the velocity. Motions generated using our method satisfy the limitations and were feasible for the physical humanoid robot.

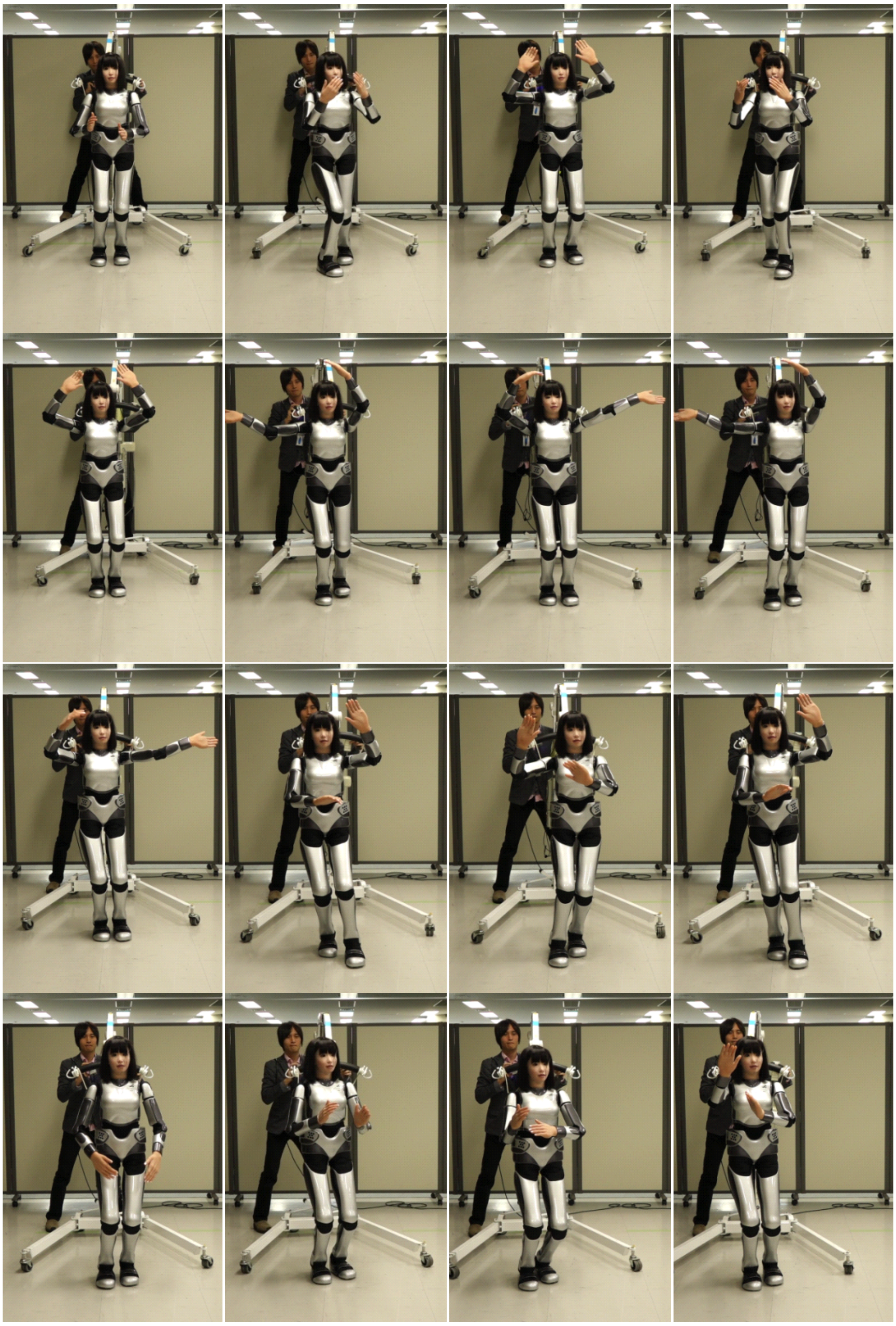


Figure 3.30: The Don-pan dance at the tempo 0.85 times faster than the original.

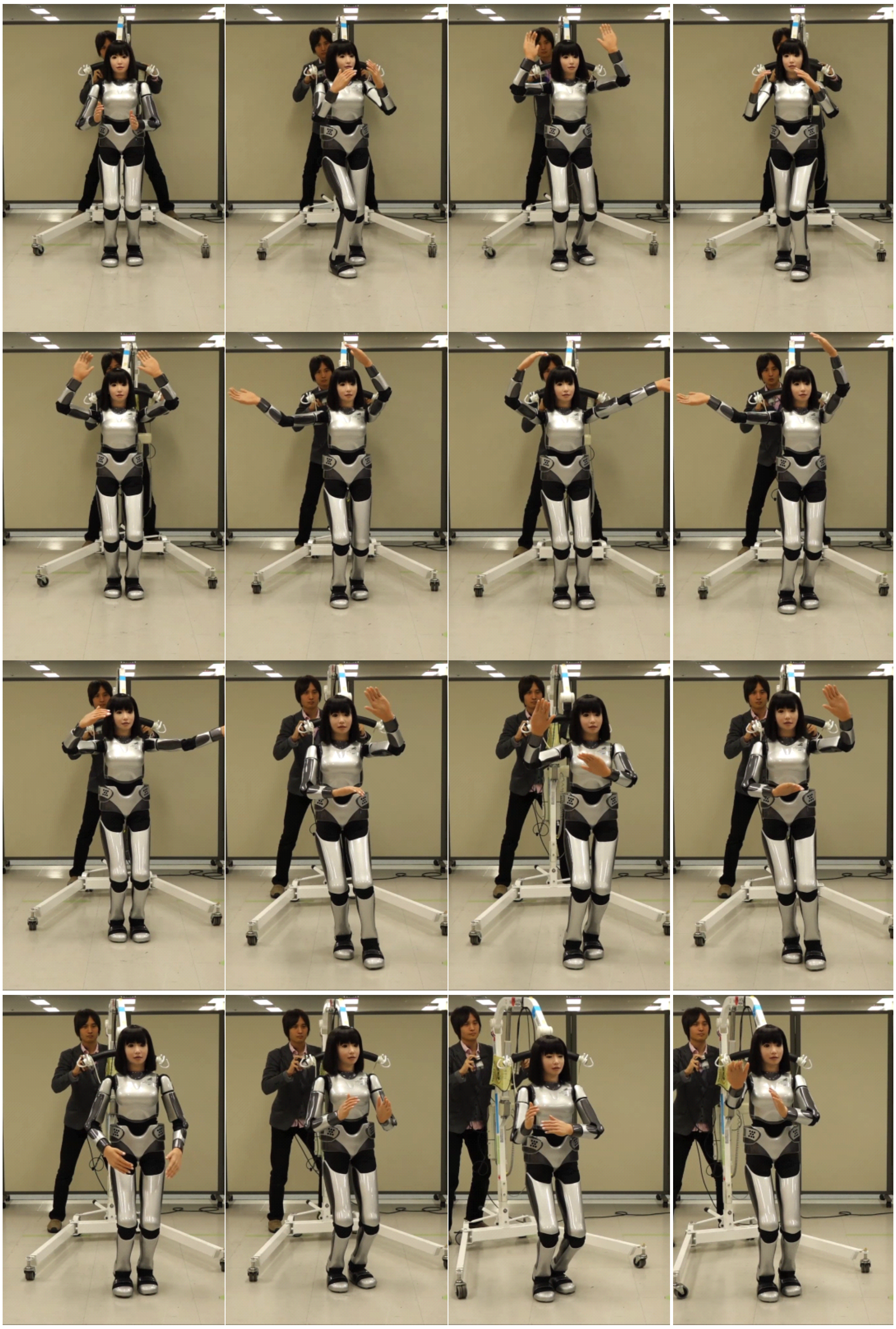


Figure 3.31: The Don-pan dance at the original tempo.

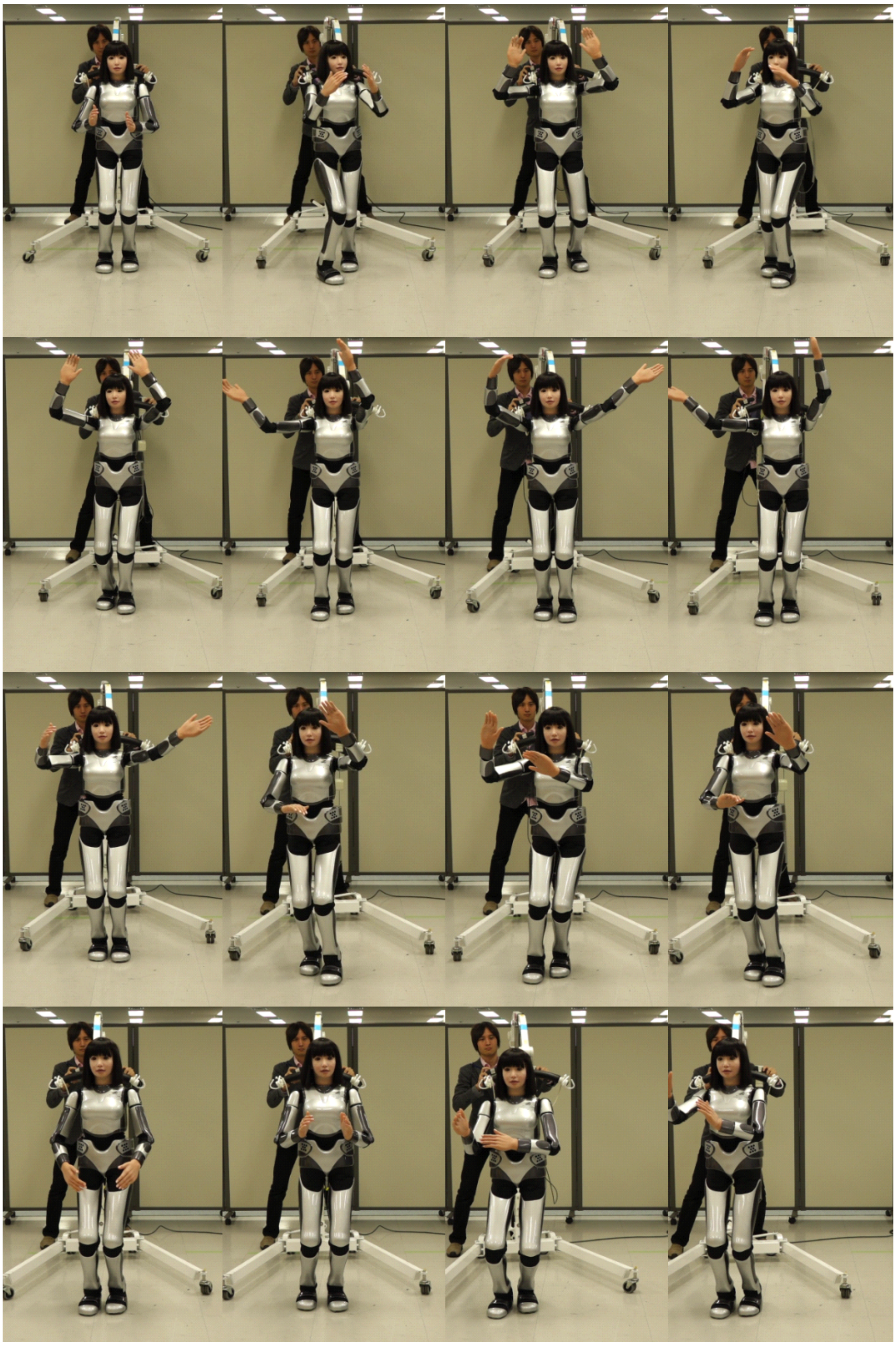


Figure 3.32: The Don-pan dance at the tempo 1.2 times faster than the original

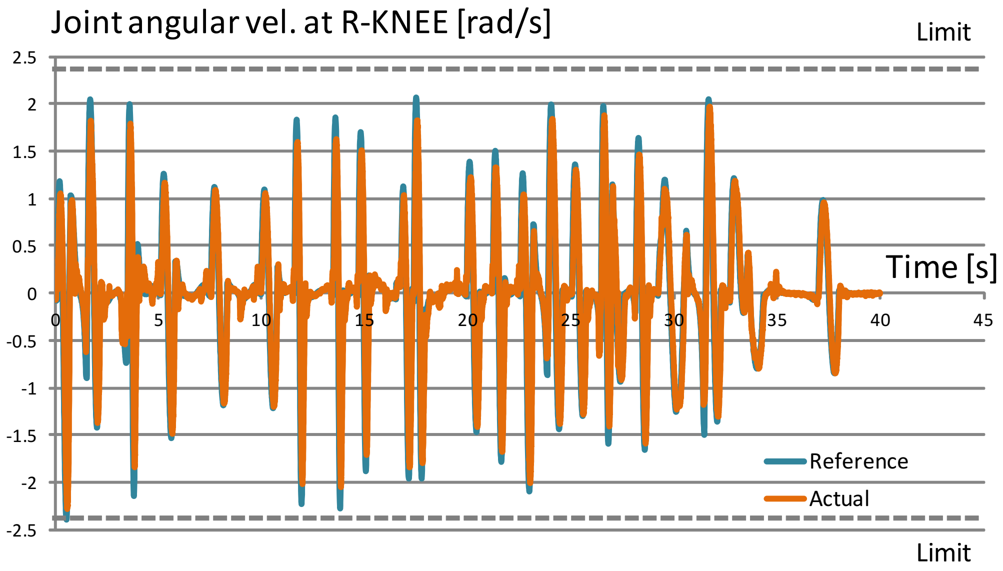


Figure 3.33: Velocity sequences of the right knee angle generated for the tempo 0.85 times faster than the original.

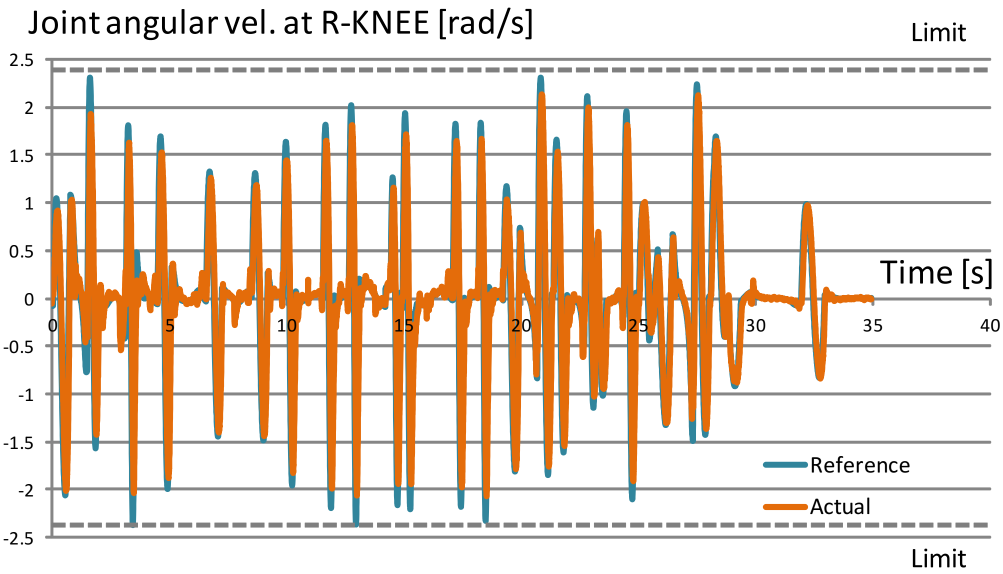


Figure 3.34: Velocity sequences of the right knee angle generated for the original tempo.

3.4.4 Quasi On-line Generation

To validate our method for keypose-based on-line generation, the dance motions whose tempo are different from each other are created for segments defined by keyposes and grafted one segment after another. As an input, a task sequence which is optimized for original tempo is given to the system. As shown in Figure 3.35, in this experiment, the task sequence is divided into three segments based on keyposes.

Our system, first of all, processed a segment surrounded by first and 9th keyposes and created dance motions whose tempo is fixed to 0.85 times faster than original. The computation time was 1.97 sec for a 10.08-second motion. The system processed a segment surrounded by 9th and 14th keyposes and created dance motions whose tempo is fixed to the original and grafted to the previous segment. The computation time was 1.227 sec for a 10.71-second motion. Finally, the system processed a segment by 14th and the last keyposes and created dance motions whose tempo is fixed to 1.2 times faster than original and grafted to the previous segment. The computation time was 4.85 sec for a 8.933-second motion. The time intervals for blending and grafting segments are represented by green-colored area. This experiment shows that the computation time is small enough to use on-line.

As a result of grafting segments using an interpolation function, there are fluctuation of the actual ZMP positions during time intervals for interpolation. Error between desired and actual ZMP positions is shown in Figure 3.36. At the last few seconds of each segment, actual ZMP positions fluctuate, however, the errors are within a few centimeters and sufficiently-small for feasibility. Although two precipitous peaks of error are found around 5.67 sec and 27.255 sec, the error come from translation of actual ZMP little earlier than the desired one. To validate this, we executed the motions generated via experiments above using a physical humanoid robot HRP-4C. The result of the experiment is shown in Figure 3.37. The robot could perform the motions without falling down.

The joint angular velocities in the experiment are shown in Figure 3.38. Velocity sequences of the right knee angle are shown. The green lines represent the joint angular velocities of the motions generated using the proposed system. The orange lines represent log data of the execution recorded by a sensor of the robot. The gray lines represent the upper/lower limits of the velocity. The time intervals for grafting segments are represented by light-green-colored area. Motions generated by our proposed method satisfy the limitations and were feasible for the physical humanoid robot.

From these experiments, we believe that our system can actually generate feasible robot motions on-line, and will make it possible for the robot to modify timings of dance motions, even if the robot motions are not synchronizing with music tempos.

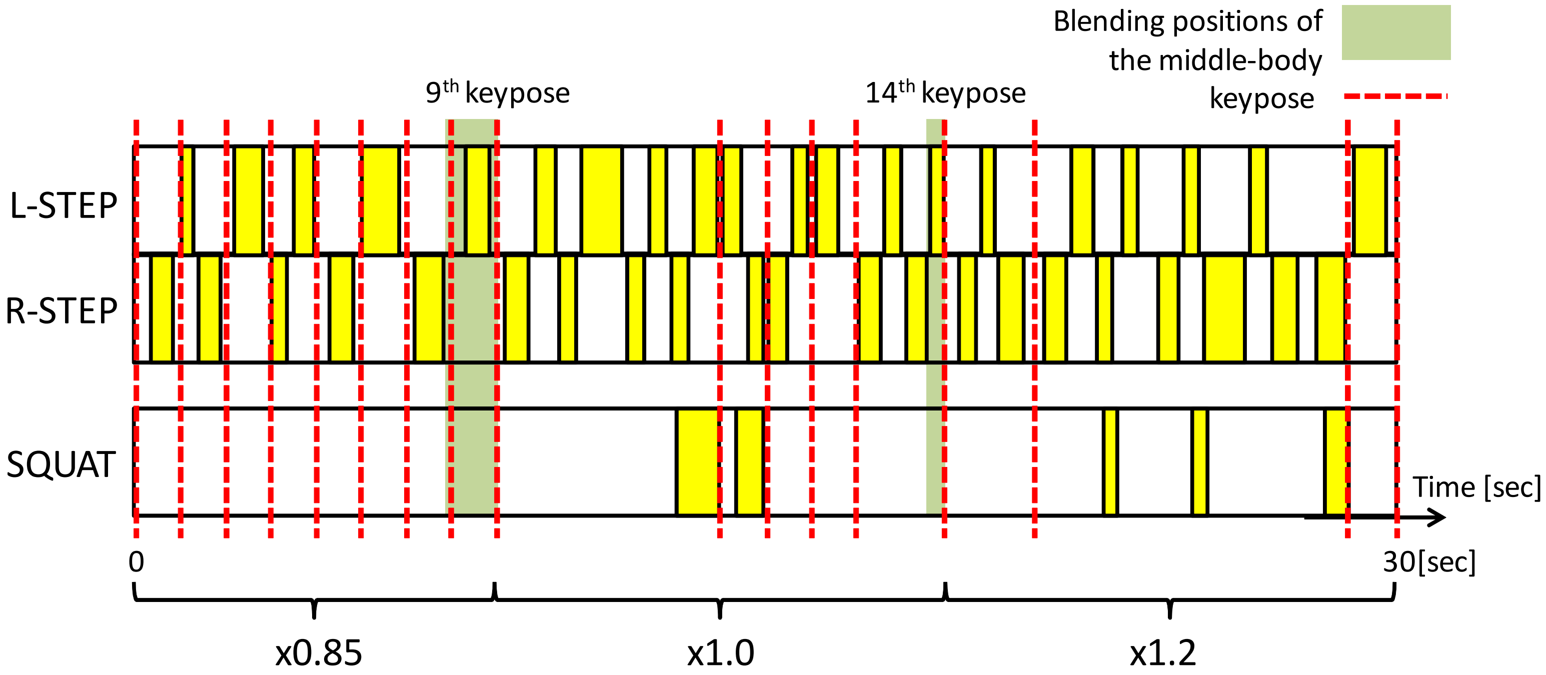


Figure 3.35: The task sequence divided into three segments based on keyposes.

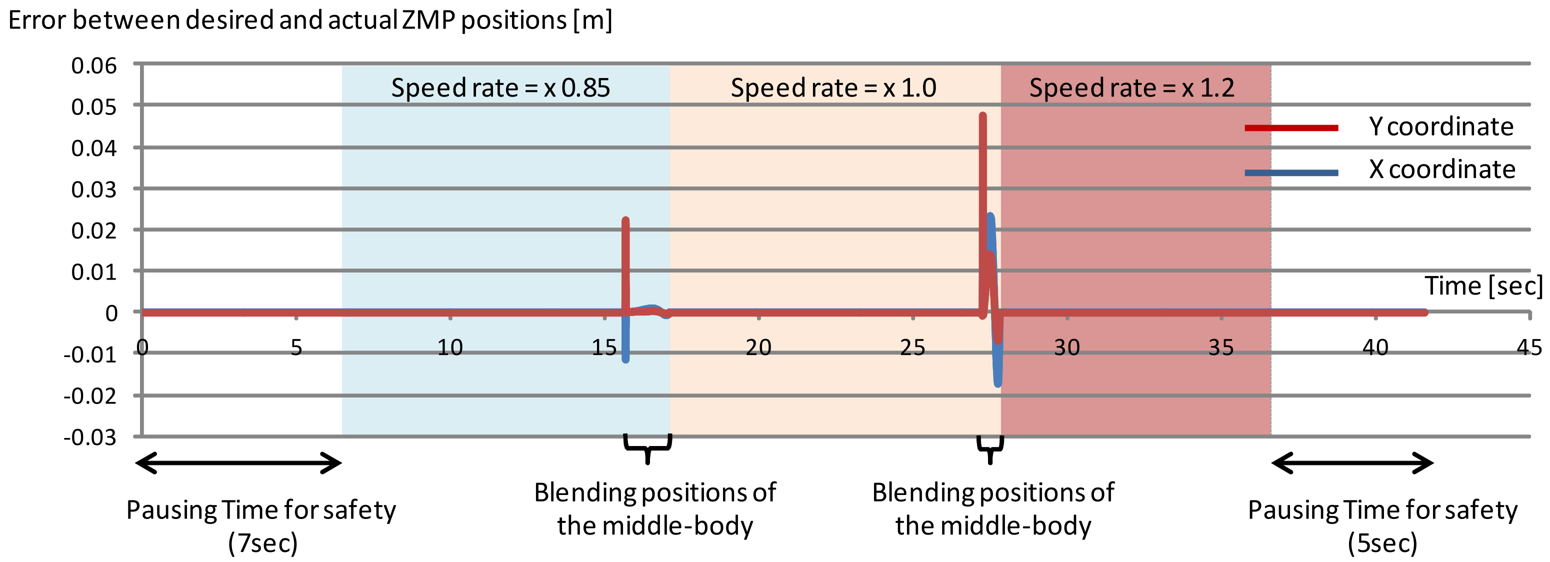


Figure 3.36: Trajectory of error between actual and desired ZMP.

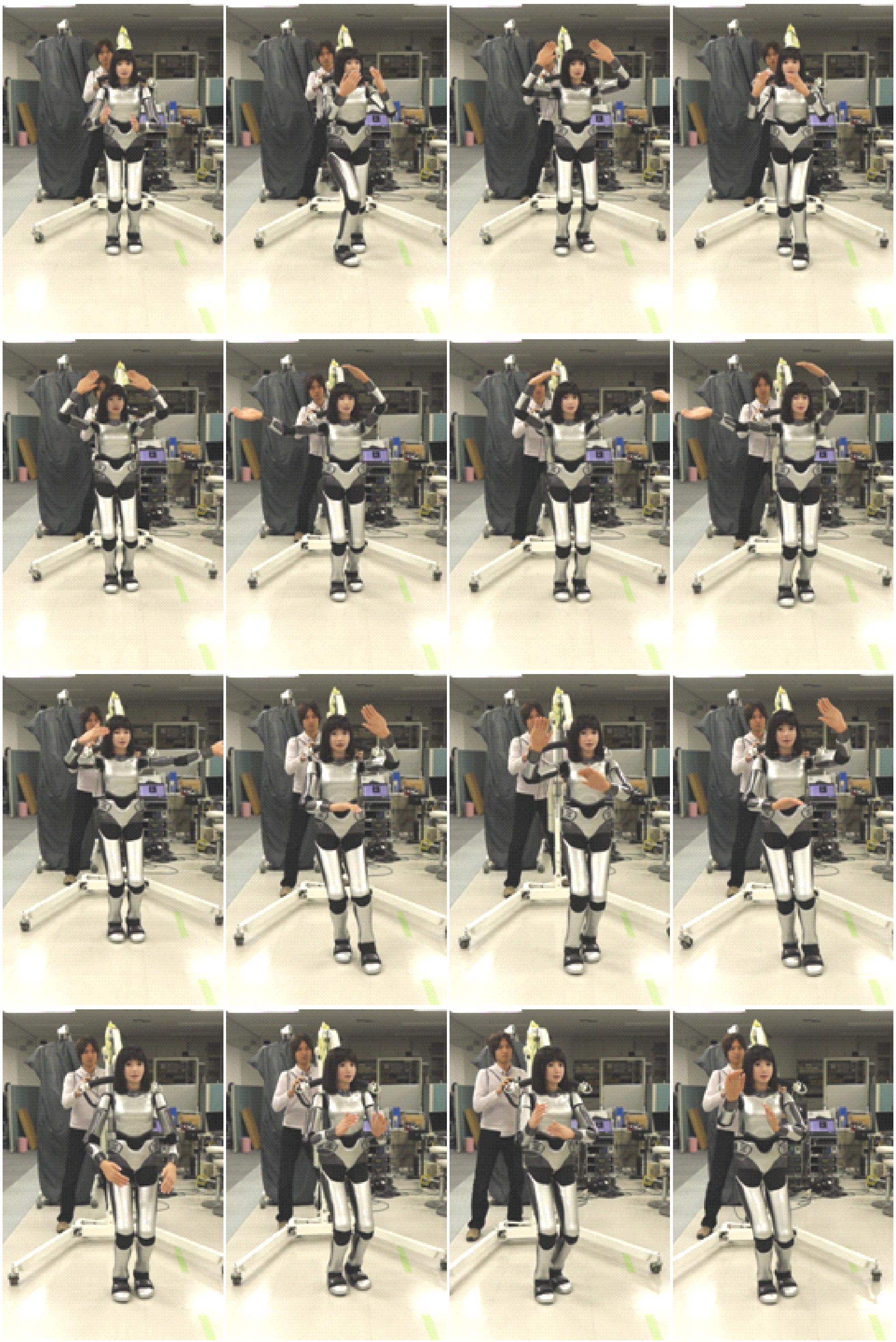


Figure 3.37: The Don-pan dance motion, whose tempo varies twice, performed by HRP-4C.

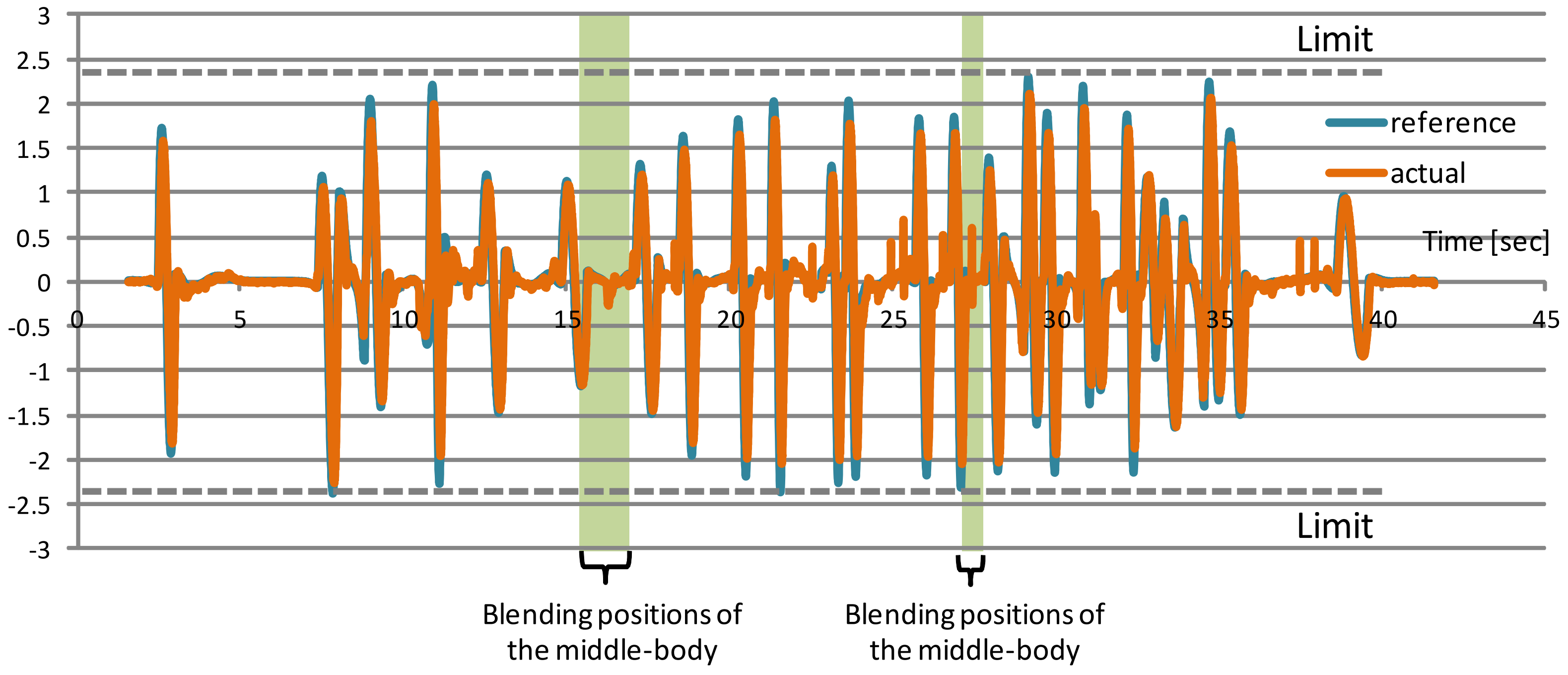


Figure 3.38: Velocity sequences of the right knee angle for motions whose tempo varies twice.

3.5 Discussion

Although temporal motion styles are extracted from observation of the Aizu-bandaisan dance in various music tempo, our algorithm was applicable to another folk dance, the Don-pan dance, and generated dance motions feasible for a physical humanoid robot. This shows the generality of our method to a certain extent.

Our algorithm for on-line generation is based-on keyposes and does not handle the real-time control such as changing speed while stepping foot. This is because we found that keyposes are essential in the dance from observation and assume that a dancer changes speed of dance based on keyposes too. As a result of experiments, our method did not make unnatural impression, and moment of changing of dancing speed was unnoticeable.

Because physical humanoid robots have some noise in motor control, we equipped a security crane in case of failure for experiments. Our dancing robot successfully performed without a falling down and the security crane was not used in the experiments. However, the control noise often cause landing of the swing foot on a tilt. This make the robot unstable if the swing foot is used for next support foot, especially in motions such as turning the body trunk. For safe control, additional compensation mechanism will be desirable.

3.6 Summary

In this chapter, we focused on a human's capability for dancing to music performances of varying tempos, and proposed an algorithm to realize this capability in a humanoid robot. As approach, we first, analyzed human dancing, and extracts modification strategies, which we call *temporal motion styles*, used by humans to adapt to music tempos, and then, proposed motion modification strategies, based on the temporal motion styles, to create a dance motions at an arbitrary musical tempo. From the observation, we found that keyposes are essential in the dance and can be employed as anchor points for integration of whole-body dance motions. We integrated individual temporal scaling algorithms for lower-body, middle-body, and upper-body motions obtained from modeling of human capability of dancing to various musical tempos. Then we presented a method for on-line dance motion generation on the assumption that a robot is dancing to music with time-varying tempo such as live-music. Our on-line generation is also based on keypose, because our observation shows that keyposes are essential in the dance and proposed temporal scaling techniques are also based on keyposes. Validation of our algorithm via experiments using a physical humanoid robot HRP-4C is conducted. In the experiments, the Don-pan dance was generated to the original, 0.85 times, and 1.2 times faster tempo. The robot expressed the keyposes using the whole-body in appropriate timings in the sequence, and provided viewers with an artistic dance pattern in which upper-body and leg motions were fully harmonious. This experiments demonstrated that our algorithm based on temporal motion style is effective to generate motion variations according to various music tempos. The temporal motion styles obtained from observation of a Japanese folk dance, the Aizu-bandaisan dance, was applicable to another folk dance, the Don-pan dance.

Chapter 4

Spatial Motion Style

4.1 Introduction

Synthesizing human-like and stylistic robot behavior [KNG*11][YRA13] is becoming more important as entertainment robots become popular. This kind of topic has been tackled in the animation community to synthesize realistic, emotional and animated motions of CG characters [TH00][SCF06]. Similarly we expect such technologies that generate expressive motions according to scenarios can make robots very human-like and amiable.

Developments of learning-from-demonstration approaches have enabled robots to learn and imitate human motions in various task domains. The only problem is that a few of those approaches explicitly consider the person-specific differences in motions: To achieve a specific task, observed human demonstrations are often generalized and used to generate an instance of robot motion. However, as shown in Figure 4.1, even when we achieve a simple task such as tossing rings to the goal, details of the throwing motions vary according to individuals. In the Aizu-bandaisan dance, a Japanese traditional folk dance, it is no secret that there are many variations in motions according to gender or individual as shown in Figure 4.2. Thus, human motions to specific tasks varies according individuals and are refereed as *styles*. We are interested in characterizing this vaguely defined concept and imitating it using humanoid robots. We expect that such ability would expand the capability of entertainment robots.

In this chapter we focus on person-specific styles in motions, which we call *spatial motion style*. The proposed method in this chapter allows a humanoid

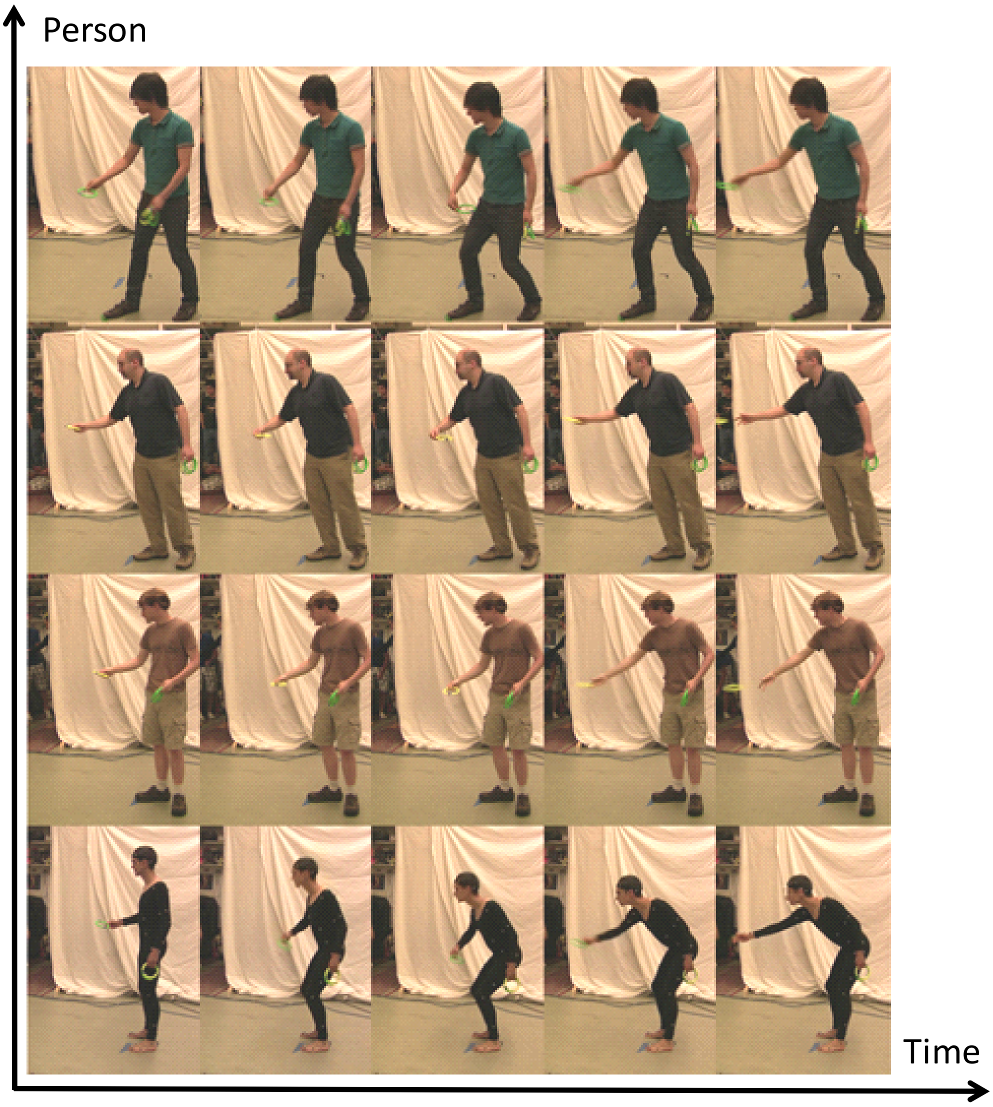


Figure 4.1: An example of person-specific motion styles in ring toss motions. Four players are tossing rings in their own way. Differences in hand position, attitude of body trunk, and bend angle are especially noticeable.

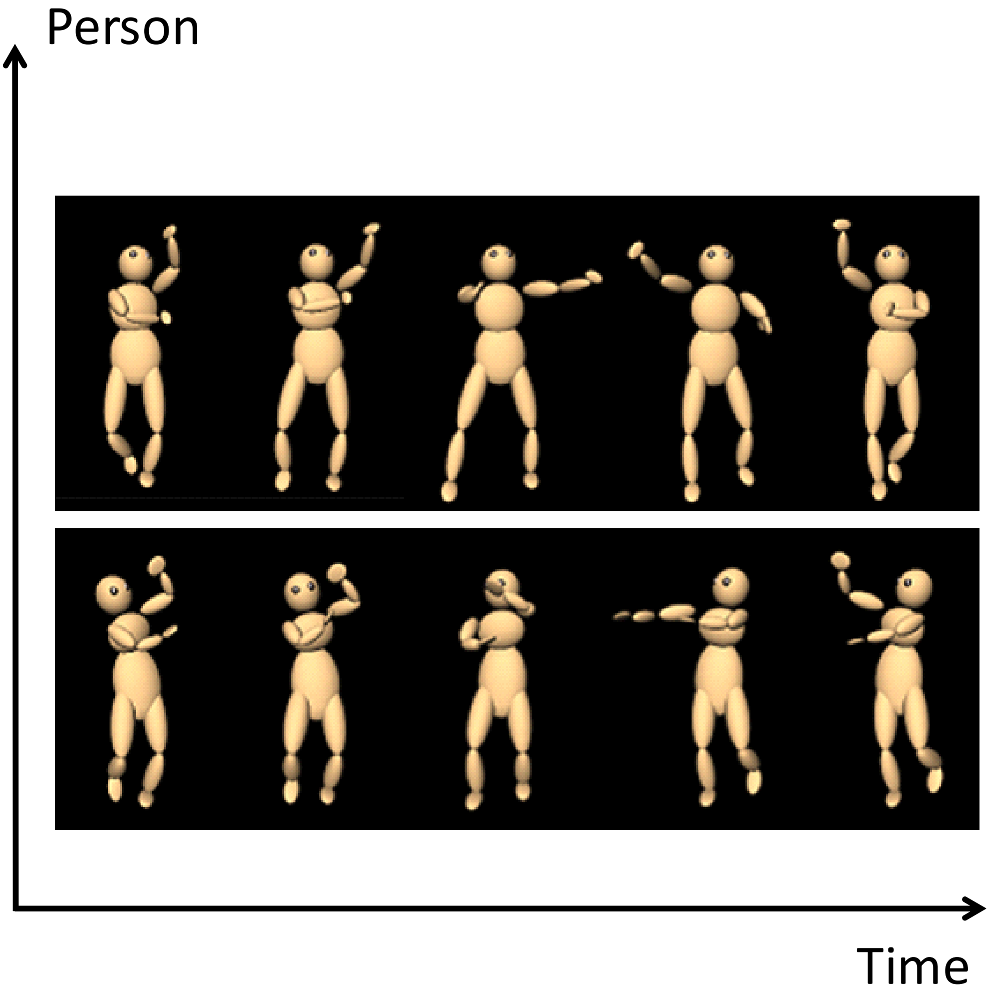


Figure 4.2: An example of person-specific motion styles in the Aizu-bandaisan dance.

robot to extract spatial motion styles from human demonstrations. And it allows the robot to imitate the motion based on the extracted styles within the physical limitations of robots. This is done automatically by extending the task model representation [NNK*07] without losing its high applicability. In our method a robot analyzes multiple demonstrations performed by a person, and then extracts the common behaviors as styles for that particular person.

In a motion analysis, a human demonstration is decomposed into a sequence of predefined primitive actions called *task*, which describe "what to do". *Skill parameters* for each task describe "how to do" it. We characterize the tendencies of how to do each task from multiple demonstrations of one particular person as styles and focus on them.

Our framework for a robot motion generation first extracts skill parameters for all demonstrations of a person. Then a robot motion is computed by solving a non-linear optimization problem. The set of skill parameters, together with other constraints, is used in the objective function to generate the motion that is considered similar in style.

To verify the proposed framework, we used a ring toss game. The task model for a ring toss game is designed by analyzing multiple demonstrations of various players. The statistical distribution of all sets of skill parameters that are extracted from the same player, defines the spatial motion styles for that player. The generated motions based on the styles were actually performed by a physical humanoid robot, and compared with each original motion of the players. The robot could imitate their style of tossing the rings to the goal within the limitation of its physical constraints.

4.1.1 Prior Works

In the robotics community synthesizing human-like motions from motion capture data has been investigated. To absorb kinematic differences, Pollard *et al*[PHRA02] modified joint-angle trajectories preserving the wave pattern of them within the constraints. On the other hand, Nakaoka *et al* [NNK*07] abstracted dance motions based on task models. Deriving motions from a pre-segmented motion capture database [YRA13] [JM02] also have been actively developed. To make robot motions look as much like original human motions as possible, optimization-based methods [SYK*08] [NK12] [MYN13] have been developed. However their cost functions in optimization are fixed regardless of target tasks; There is no guarantee that those functions are essential for any other motions. In addition to those factors, our proposed method considers the variability in motions of one particular person.

In the animation community there are also a number of studies on stylistic motion synthesis. Neff [NK09] extracts correlations between components of motion for an interactive editing tool of motion styles. Torresani *et al* [THB06] used Laban Movement Analysis to describe styles in the domain of three-dimensional perceptual space: flow, weight, and time. These factors are quantified manually by the designer. Various studies analyze and learn time-varying vectors in joint angles using HMMs [BH00], PCA [UGB*04], ICA [SCF06], and DP matching [NNI04]. Comparing styles in terms of mood/emotion-specific variations, few studies consider spatial motion styles.

4.2 Characterization of Spatial Motion Styles

This section characterize the spatial motion style by extending a concept of a LFO paradigm. Our focus is the tendencies of how to do each task from multiple demonstrations of one particular person, and in this thesis those tendencies are characterized as the styles. The following subsections begin with the description of a task model that is used to represent motions in a ring toss game. Then, a detail of the method to extend the task model to represent the styles is given.

4.2.1 Task Models

In task models, a series of movements are segmented based on transitions of state, and a segment is recognized as a primitive action called task. Skill parameters of a task explain how this is done. Whole motions are abstracted into a sequence of tasks and then reused to generate robot motions.

Tasks

First, to design tasks in a ring toss game, we asked seven human players chosen at random to toss the ring to the goal from the same standing position without any other specific instructions. Fig. 4.1 shows sample motion sequences from four, out of a total of seven, human players. Each player has their own style of motions, but a common structure also can be discovered among them; they first take the ring back spontaneously and then release it through the air to the goal.

Secondly we analyzed movements of the dominant hand in a typical sample motion of a player (See Fig. 4.3). Upper graph shows time-series data of hand speeds, and lower graph shows that of hand positions represented in the X-axis of the world coordinate. To define the world coordinate, the standing position of a human player is considered as the origin, and the goal of a ring toss game is assigned to be on the X-axis. Timings circled with purple and blue represent a local maximum of hand positions and the global maximum of hand speeds, respectively. The upper graph suggests that the player stops the hand just anterior to, and behind, the timing of maximum speed (circled with blue). The former is the end timing of a preliminary action before throwing rings, where the hand is pulled closer to the player's body trunk. The latter is that of releasing rings, where the hand is the closest to the goal and then pulled back. We labeled these two stopping states as AIM state and FINISH state, while labeling the initial

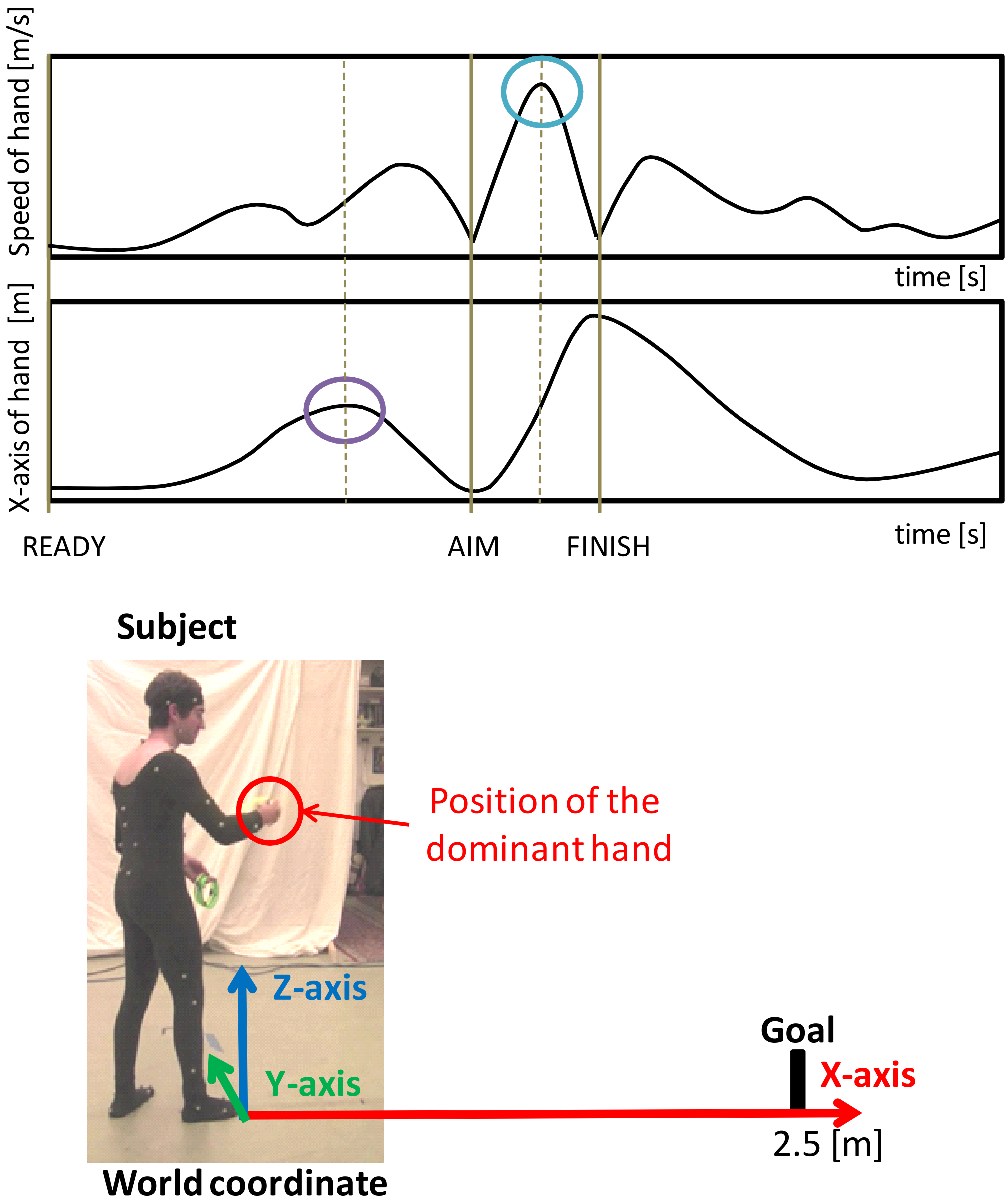


Figure 4.3: Movements of the dominant hand in a sample motion of a player.

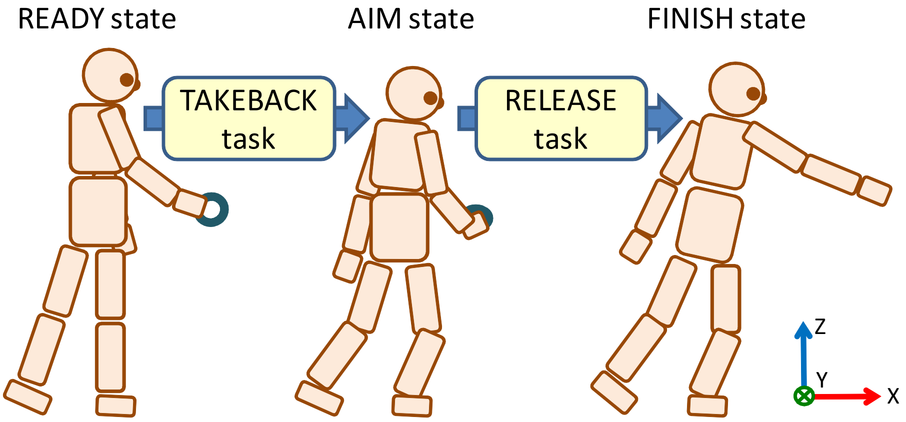


Figure 4.4: Design of task sequence in the ring toss motion.

state of the whole motion as READY state. This chapter focuses on movements from READY state to FINISH state, while movements after FINISH state are not considered as essential behavior for the ring toss game. When a player throw the ring in a row, a FINISH state transits to a READY state for the next trial.

Based on these specific states, the series of movements can be divided into two segments. We defined those two segments as two different tasks: TAKEBACK and RELEASE (See Fig. 4.4).

TAKEBACK

is a preliminary action before throwing rings, and is defined as a transition from READY state to AIM state.

RELEASE

is a throwing action, and is defined as a transition from AIM state to FINISH state.

From this analysis, a series of movements for a trial of the ring toss can be represented as transitions between three states. Actually all of the human demonstrations we captured could be automatically segmented based on the task definitions described above. This supports the generality of our task representation in the domain of the ring toss. There might be other alternative motion structures to toss the ring if we observe other players. however, in this paper, we assume that the motions of human players from this observation covers all patterns of tossing.

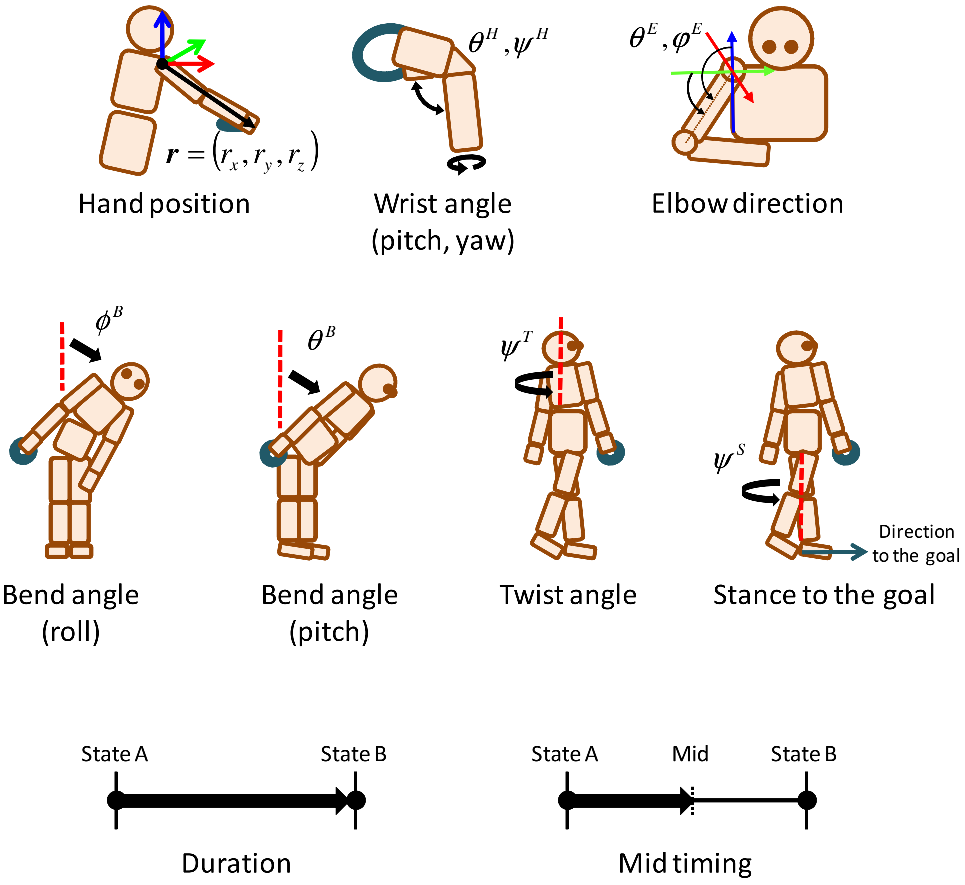


Figure 4.5: Design of skill parameters for each task in the ring toss. These skill parameters characterize the trajectory of each task by giving details at end timing and a specific intermediate timing of each task.

Skill parameters

This subsection gives description about skill parameters.

For tasks we defined in the previous subsection, we defined skill parameters based on an observation of the ring toss motions as shown in Fig. 4.5. Skill parameters are defined in common for each task, TAKEBACK and RELEASE. These skill parameters characterize how to do the task by describing the status at the initial state, the specific intermediate timing, and the finishing state of the task. Concrete definitions are given as follows. Without loss of generality, all players are assumed to be right-handed and throw the ring with their right hand.

$\boldsymbol{r}$: Hand position

represents a position of the right hand in a Cartesian coordinate. It is defined in a Cartesian coordinate with the origin at the right shoulder and each axis is parallel to the corresponding axis of the world coordinate. To neglect the effect from the difference in limb length, the position of the right hand is normalized by the length of the right arm.

θ^H, ψ^H **: Wrist angle**

represents a pitch angle and a yaw angle of the right wrist. The yaw axis corresponds to the direction from the right wrist to the right elbow, and the pitch axis is orthogonal to the yaw axis in a plane parallel to the flat of the right hand.

θ^E, φ^E **: Elbow direction**

represents angles corresponding to the position of the arm from the right shoulder to the right elbow in the spherical coordinate with the origin at the right shoulder. X,Y, and Z-axis of the spherical coordinate are parallel to the world coordinate axes.

ϕ^B, θ^B, ψ^T **: Bend and Twist angle**

A roll angle represents a torso leaning, a pitch angle represents a torso bending, and a yaw angle represents the twisting of the upper body. It should be noted that the order of the torso joint is assumed to be yaw-roll-pitch.

ψ^S **: Stance to the goal**

represents a yaw angle of the attitude of the waist. This is used to represent the stance to the goal of the ring toss.

Duration

represents the interval of time required for the task execution.

Midtiming

represents a specific intermediate timing in the task execution. A mid timing in TAKEBACK is defined as a timing corresponding to a local maximum of hand position in Fig. 4.3 between READY state and AIM state (circled in purple). If candidates are more than two, the closest inflection point to AIM state is chosen as the mid timing in TAKEBACK. If there is no candidate in TAKEBACK, the intermediate timing between READY state and AIM state is chosen. A mid timing in RELEASE is defined as a timing corresponding to a global maximum of hand speed (circled in blue).

Skill parameters described above cover also some features which are not parameterized as skills. For example, a speed of the hand is partially overlapped with $\boldsymbol{r}$ and timings of task execution. A ring position can be described by $\boldsymbol{r}$, θ^H, and ψ^H. Although our skill parameters might not be enough to represent the all of features in motions for each player, skill parameters can be added flexibly according to the capability of the robot platform or according to the belief of the designers on what is important or noticeable from observation.

Based on task models described above, humans demonstrations of ring toss motions are abstracted as a task sequence and corresponding skill parameters. Those abstracted information used for reconstruction of the trajectories, in robot motion generation, to imitate the original human motions. However, imitating features represented by skill parameters perfectly is often difficult for robots because of differences in length of limbs, joint limitations, and performance of actuators. Actually, for such case in previous works, a final check and manual refinement with ad-hoc rules by engineers were often inevitable for feasibility. In those process, preservation of the spatial motion styles of a person will be difficult and the generated motions may become unnatural according to the target motion.

Style parameters newly introduced in the following subsection solve this problem. We observe multiple demonstrations of a person and then parameterize the features of the movement statistically. In execution of a task by a robot, skill parameters with higher priority is estimated and optimized under the constraints of the robot. Based on this approach, we achieve the imitation of essential factors, by absorbing the difference of physical constraints between the robot and human.

4.2.2 Style Parameter

This subsection describes the representation of spatial motion styles in the task model. We first observed the difference in statistical distributions of skill parameters of a task between human players, and then defined a style parameter to represent the individual differences in the context of the task model.

Observation of Individuality in Statistical Distributions of Skill Parameters

For observation, we captured a total of 60 ring toss motions from three human players (20 motions per player) using an optical motion capture system from VICON. The distance between the specified standing point and the goal on the floor was set to 2.5 [m] for each player. We focus on the difference in statistical distributions of skill parameters in RELEASE tasks in this observation. First, distribution of skill parameters of hand positions $\boldsymbol{r}$ in RELEASE tasks were plotted for each player (See Fig. 4.6). Clusters circled in the same color mean statistical distributions of hand positions at start, a specific intermediate, and end timings in RELEASE tasks by a specific human player. The comprehensive transitions from start to end timings in RELEASE tasks are indicated by arrows. The color of the plot, circles, and arrows differentiates human players. Similarly, distributions of bend angle ϕ^B and θ^B in RELEASE tasks were plotted for each player (See Fig. 4.7).

These figures show that the statistical distribution of each skill parameter varies from player to player. Fig. 4.6 shows that a player with blue markers tends to throw rings at higher positions, in a Cartesian coordinate with the origin at the right shoulder, compared to the other players. On the other hand, a player with green markers tends to throw rings from lower positions. A player with red markers has a style which is similar to that of the player with green markers, but the difference appears at the end timings. Additionally, this player moves the hand carefully; variances at each timing tend to be small compared to the others. Fig. 4.7 shows that the player with blue markers tends to bend more both forward and sideways, while the subject with red markers does not bend as much. In this way we can compare and discover the differences and similarities in style of two or more human players based on the distributions of skill parameters. Moreover, the variance of these distributions can also suggest the importance of each skill parameter.

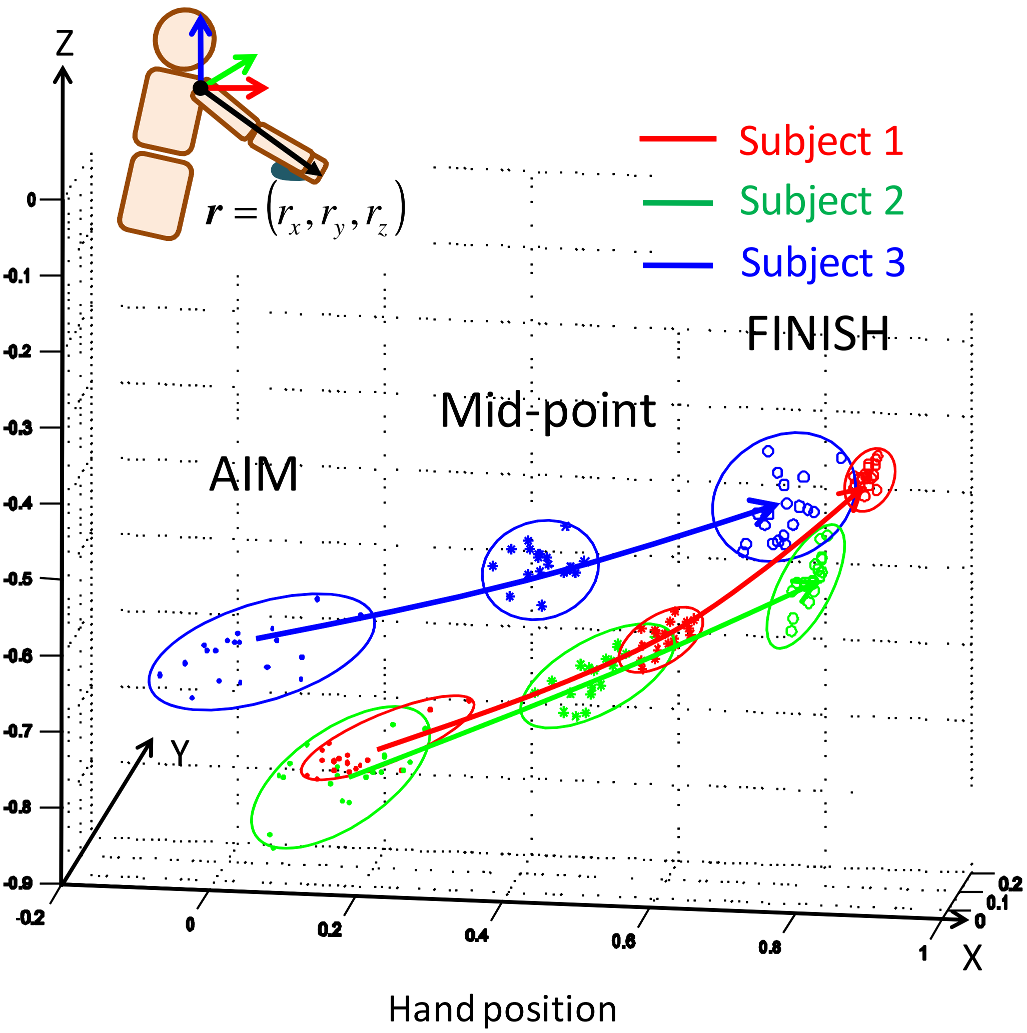


Figure 4.6: Distributions of skill parameters $\boldsymbol{r}$ in RELEASE tasks in a total of 60 ring toss motions from three human players (20 motions per player). Note that $\boldsymbol{r}$ is defined in a Cartesian coordinate with the origin at the right shoulder. Clusters circled in the same color represent statistical distributions of hand positions $\boldsymbol{r}$ at start, a specific intermediate, and end timings in RELEASE tasks by a specific human player. The comprehensive transitions from start to end timings in RELEASE tasks are indicated by arrows. The color of the plot, circles, and arrows differentiates human players.

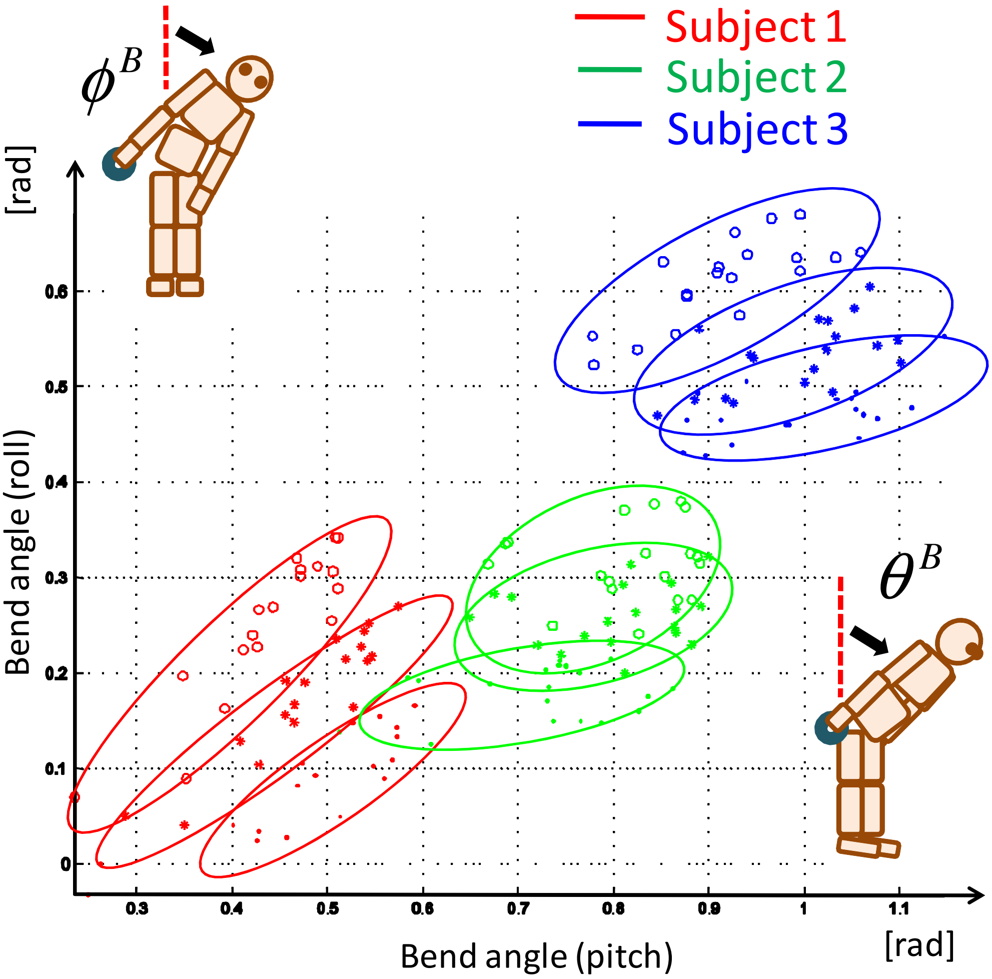


Figure 4.7: Distributions of skill parameters ϕ^B and θ^B in RELEASE tasks in a total of 60 ring toss motions from three human players (20 motions per player). Clusters circled in the same color represent statistical distributions of hand positions ϕ^B and θ^B at start, a specific intermediate, and end timings in RELEASE tasks by a specific human player. The color of the plot, circles, and arrows differentiates human players.

Characterization of Style Parameter

As illustrated in the observation, distributions of skill parameters of a task vary from player to player. In fact, those distributions describe the person-specific differences in the domain of the specific task. We use the distributions of skill parameters to represent spatial motion style. To describe the distributions, we introduce style parameters in the concept of task model, one style parameter for each task (See Figure 4.8). A style parameter consists of vectors of averages and variances of all skill parameters in the task. Skill parameters are assumed to be normally-distributed.

A style parameter d for a task is defined as follow based on mean and variance of skill parameters:

$$d = (\bar{\mathbf{s}}, \boldsymbol{\sigma}), \tag{4.1}$$

$$\bar{\mathbf{s}} = (\bar{s_1}, \bar{s_2}, \bar{s_3}, \cdots, \bar{s_k}, \cdots, \bar{s_N})^T, \tag{4.2}$$

$$\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \sigma_3, \cdots, \sigma_k, \cdots, \sigma_N)^T, \tag{4.3}$$

where N is the number of skill parameters for the task, $\bar{\mathbf{s}}$ is the vector consisting of $\bar{s_k}$ which represents the mean of the k-th skill parameter, and $\boldsymbol{\sigma}$ is the vector consisting of σ_k which represents the variance of the k-th skill parameter.

In the context of this extended task model, a style parameter describes “tend to do” of the skill parameters, while skill parameters describe “how to do” of the task. $\bar{\mathbf{s}}$ represents the most typical set of skill parameters for the task of a specific person. On the other hand, $\boldsymbol{\sigma}$ represents the flexibility of each skill parameter for the style. skill parameters with small $\boldsymbol{\sigma}$ tend to be same value, and can be considered as important factors for the spatial motion stylex of the person. A set of skill parameters which consist of $\bar{\mathbf{s}}$ can be used to reconstruct a typical motion that reflects a style of a person. However, as mentioned above, executing the motions which is the most typical for the style is often impossible because of the physical constraints. Therefore, in our proposed method for robot motion generation, an optimization, which use $\bar{\mathbf{s}}$ and $\boldsymbol{\sigma}$ extracted from the multiple demonstrations of a person effectively, achieve the natural motions imitating the style.

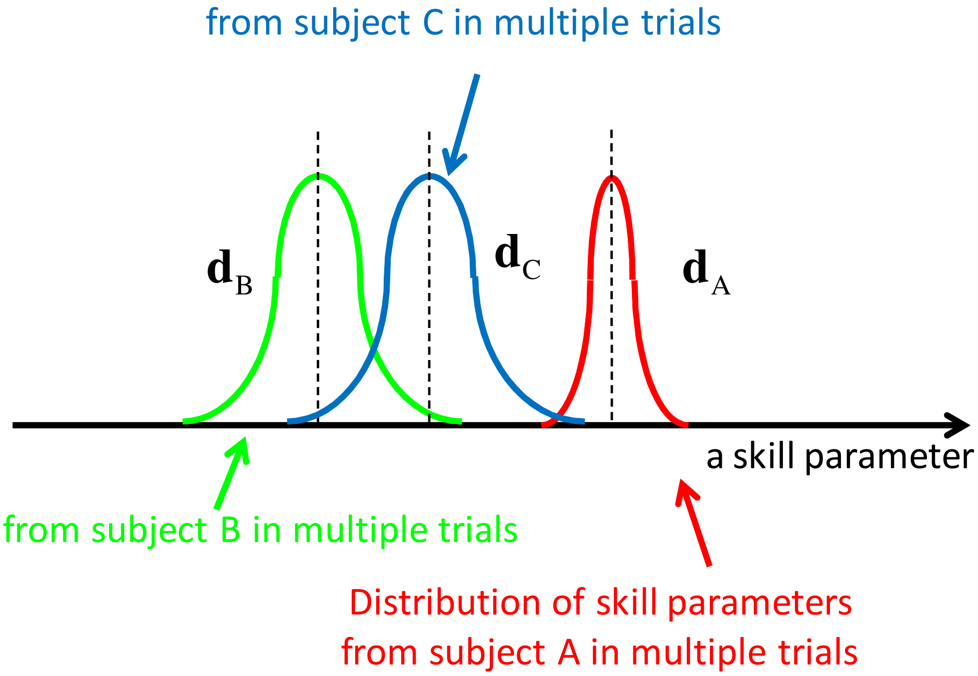

from subject C in multiple trials
$\mathbf{d}_B$
$\mathbf{d}_C$
$\mathbf{d}_A$
a skill parameter
from subject B in multiple trials
Distribution of skill parameters
from subject A in multiple trials

Figure 4.8: Style parameter

4.3 Imitation of Spatial Motion Styles

This section describes a proposed framework to generate robot motion based on style parameters described in the previous section. The framework consists of three phases:

- **Phase 1:** Style parameters are extracted from the demonstrations of a player.
- **Phase 2:** A set of skill parameters are optimized based on the style parameter so that robots can imitate the style while satisfying the constraints.
- **Phase 3:** Whole body motions are reconstructed by the optimized skill parameters.

Phase 1: Using multiple demonstrations of a player as inputs, states are detected based on their definitions. Motions are segmented as tasks and their corresponding skill parameters are extracted. Then, style parameters for each task are calculated from the mean and variance of skill parameters of multiple demonstrations.

Phase 2: Skill parameters for each task are optimized based on the style parameters so that the reconstructed trajectory can mimic the style as closely as possible within the physical constraints. This optimization is executed by minimizing an objective function which consists of terms that are derived from physical constraints of the robots, distance between the ring and the goal, and preservation of the style.

Phase 3: Whole body motions are reconstructed from the fully optimized skill parameters. Kinematics of the robot at each time frame are reconstructed from skill parameters. Then, trajectory in joint angle space of the robot is calculated.

4.3.1 Skill Optimization Based on Style Parameter

Initial entry of skill parameters is set to the $\bar{s}$ of the style parameter. Then values of each component are updated iteratively to minimize the objective function based on physical constraints of the robots, distance between the ring and the goal, and preservation of the style. The objective function to be minimized is designed as follows:

$$E_{Style} + \lambda_1 E_{Ring} + \lambda_2 (E_{Angle} + E_{Velocity} + E_{Collision}), \tag{4.4}$$

$$E_{Style} = \sum_{i=1}^{N} (\frac{\bar{s_i} - s_i}{\sigma_i})^2 , \tag{4.5}$$

E_{Style} is a term for preserving the style. s_i is the i-th skill parameter, and N is the number of skill parameters. The value of E_{Style} is increased in direct proportionality to the difference between s_i and $\bar{s_i}$. In addition, each term is weighted by $1/\sigma_i$. A skill parameter with a smaller variance is preserved by a larger weight coefficient, and that with a larger variance is adjusted preferentially by a smaller weight coefficient.

$$E_{Ring} = |\boldsymbol{r}_{goal} - \boldsymbol{r}_{ring}(\boldsymbol{s})|^2, \tag{4.6}$$

E_{Ring} is a term relevant to a distance between the goal and the landing points of rings thrown by the robot. A landing point $\boldsymbol{r}_{ring}(\boldsymbol{s})$ is simulated from a robot motion generated using a value set of skill parameters $\boldsymbol{s}$. Initial positions and initial velocities of rings are given as those of the hand positions at the moment in which hand speed becomes maximum during RELEASE task. The gravity acceleration is assumed to be 9.8 $[m/s^2]$. The air resistance and frictions are ignored in the calculations of landing points. The larger the difference between a position of goal $\boldsymbol{r}_{goal}$ and a landing point $\boldsymbol{r}_{ring}(\boldsymbol{s})$ is, the more the value of this term will increase.

$$E_{Angle} = \sum_{k=0}^{K} \sum_{j=0}^{N} \alpha_{j,k}^2(\boldsymbol{s}), \tag{4.7}$$

$$\alpha_{j,k}(\boldsymbol{s}) = \begin{cases} q_{j,k}(\boldsymbol{s}) - q_j^{max} & (q_{j,k}(\boldsymbol{s}) > q_j^{max}) \\ q_j^{min} - q_{j,k}(\boldsymbol{s}) & (q_{j,k}(\boldsymbol{s}) < q_j^{min}) \\ 0 & (otherwise) \end{cases} , \tag{4.8}$$

E_{Angle} is a term relevant to constraints of joint angles. First, trajectories in joint angle space generated using the skill parameters $\boldsymbol{s}$ are simulated. Then exceeding of joint limitations is detected for each time frame, and it increases the term according to the level of excess. $q_{j,k}(\boldsymbol{s})$ represents the angle of j-th joint at the k-th time frame. q_j^{max} and q_j^{min} represents the upper boundary and lower boundary of the j-th joint angle. The value becomes zero if the joint angle is within the limit.

$$E_{Velocity} = \sum_{k=0}^{K} \sum_{j=0}^{N} \beta_{j,k}^2(\boldsymbol{s}), \tag{4.9}$$

$$\beta_{j,k}(\boldsymbol{s}) = \begin{cases} \dot{q}_{j,k}(\boldsymbol{s}) - \dot{q}_j^{max} & (\dot{q}_{j,k}(\boldsymbol{s}) > \dot{q}_j^{max}) \\ \dot{q}_j^{min} - \dot{q}_{j,k}(\boldsymbol{s}) & (\dot{q}_{j,k}(\boldsymbol{s}) < \dot{q}_j^{min}) \\ 0 & (otherwise) \end{cases}, \tag{4.10}$$

$E_{Velocity}$ which is relevant to the constraint of joint angular velocity is also treated in the same way as E_{Angle}.

$$E_{Collision} = \sum_{k=0}^{K} \sum_{p=0}^{N} \gamma_{p,k}^2(\boldsymbol{s}), \tag{4.11}$$

$$\gamma_{p,k}(\boldsymbol{s}) = \begin{cases} r_{1,p} + r_{2,p} - d_p(\boldsymbol{s}) & (r_{1,p} + r_{2,p} > d_p(\boldsymbol{s})) \\ 0 & (otherwise) \end{cases}, \tag{4.12}$$

$E_{Collision}$ is a term relevant to the constraint of self collisions. In this implementation, collision is detected by calculating the distance between swept sphere volumes which wrap around body segments. First, positions of each body segment in the whole motion are simulated using the skill parameters $\boldsymbol{s}$. Then upper body, lower body, right upper arm, right lower arm, and right hand are wrapped around by each of the five swept sphere volumes. The radius of each sphere is determined empirically to wrap around each body segment with enough margin. $\gamma_{p,k}(\boldsymbol{s})$ which represents collision level of p-th joint pair at the k-th time frame is calculated by $(r_{1,p} + r_{2,p}) - d_p$. Where d_p is the minimal distance between axes of each swept sphere volume of the p-th pair and $r_{1,p}, r_{2,p}$ are each radii of a sphere. Considering the movement of the ring toss, collision pairs to be checked are set as follow:

Pair 1 Hand - Upper body

Pair 2 Hand - Lower body

Pair 3 Right upper arm - Upper body

Pair 4 Right upper arm - Lower body

Pair 5 Right lower arm - Upper body

Pair 6 Right lower arm - Lower body

E_{Ring}, E_{Angle}, $E_{Velocity}$ and $E_{Collision}$ are factors that should be considered by hard constraints if it is possible. However each component of $\boldsymbol{s}$ does not represent joint angles or landing points of the ring linearly. Therefore, in this framework, these factors are dealt with as soft constraints, and satisfied by adjusting weight coefficients and thresholds.

Optimization of value set of skill parameters is executed by minimizing the objective function described above. To solve this optimization problem, we used the Levenberg-Marquardt method [MNT04]. λ_1 and λ_2 are given empirically.

4.3.2 Motion Generation From Executable Skill Parameters

This subsection describes the process to reconstruct the trajectory of a task from a set of skill parameters for imitation bu humanoid robot.

A trajectory of a task is reconstructed by interpolating the state transition during the task execution. Timings of each key frame: a start, an intermediate, and an end timing of the task, are given by skill parameters *Duration* and *Midtiming*. Then skill parameters of hand position $\boldsymbol{r}$, hand direction θ^H, elbow direction θ^E, twist and lean ψ^T ϕ^B θ^B, and stance ψ^S at those key frames are interpolated. Interpolation of each component is performed using a cubic natural spline function.

Robot motions in the form of joint positions are calculated using the interpolated skill parameters. The postures of the robot in each time frame are determined by the joint positions. Finally, motions in the form of joint positions are converted to trajectories in joint angle space of the robot by solving inverse kinematics. As mentioned above, the exceeding of physical limitations in interpolated motions is also checked during the optimization process.

4.4 Experiments

4.4.1 A Robot Platform

In this experiment, we employed a physical humanoid robot as shown in Figure 4.9. The robot has 39 degrees of freedom and each joint is driven by hydraulic motors. The feet are fixed on the base. In default setting, the posture of the robot is updated at 30 [fps] by the input data. Seven degrees of freedom on the right arm and four degrees of freedom on the torso are used. In addition, an auxiliary plate for grasping the ring is designed as shown in Figure 4.9. The ring set on the plate is fixed and released by the right thumb.

Delay Compensation in Robot Control

Each actuator of robots are controlled so that they follow the each desired value given by inputs. However, there are robots with non-negligible delays for the actuators to be updated. These non-negligible delays make it difficult for the robot to trace inputted desired motions accurately. Therefore there are major gaps between inputted motions and outputted sensor data of the robot as shown in Figure 4.12. Not only temporally delaying but also impairing of the shape also occur. Therefore, to control such type of robots, we developed a delay compensation filter for input motion as follows.

As described above, a hydraulic motor system is employed for the A100 robot. Therefore, time constants are too large to compensate delays by constructing feedback loops, and feed-forward delay compensations are required. So we identified systems from some Gaussian noise inputs and outputs using a system identification technique based on discrete-time state space model, and acquired transfer functions for each actuator. To estimate discrete-time state space models, we applied a prediction error method.

Then, inverse operations for input motions are performed based on identified systems so that output will correspond to the desired motion. Here, input motions for the ring toss in 30[fps] have around 150 frames at most. Therefore, in this situation, compensated input motions for each actuator can be found by solving a constrained optimization problem:

$$\arg\min_{\boldsymbol{q}} \sum_{i=1}^{K} (Ref_i - Sim(\boldsymbol{q})_i)^2, \tag{4.13}$$

$$lb_i < q_i < ub_i \tag{4.14}$$

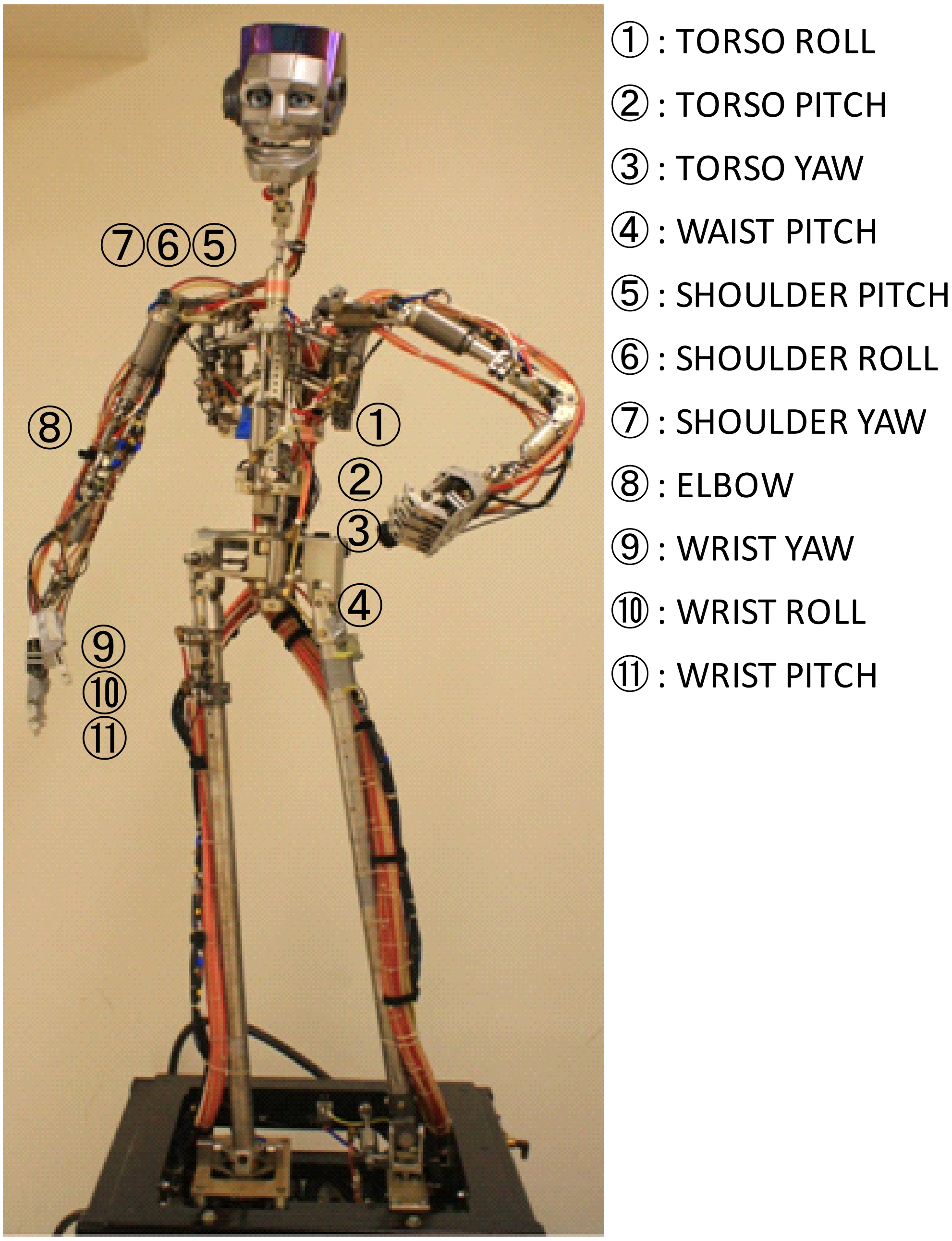


Figure 4.9: A100 robot and its degree of freedoms.

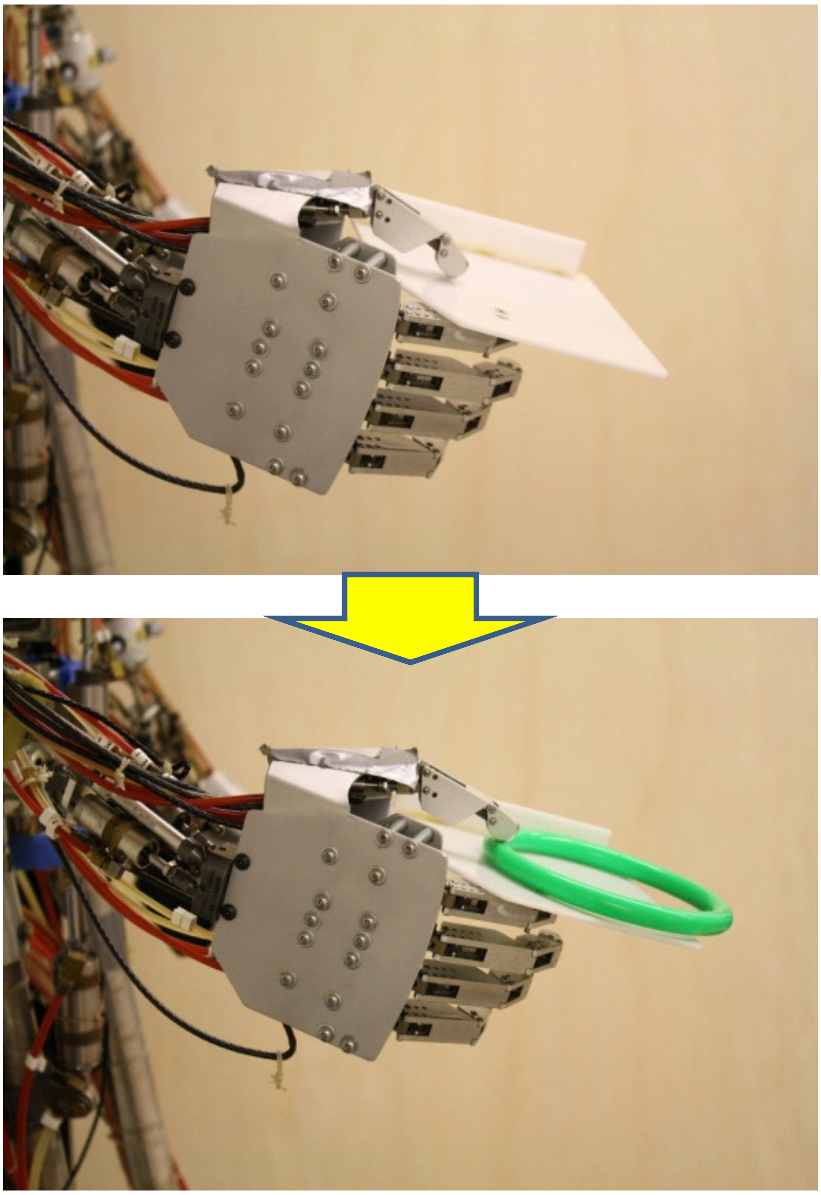


Figure 4.10: A hand plate for grasping on the right hand.

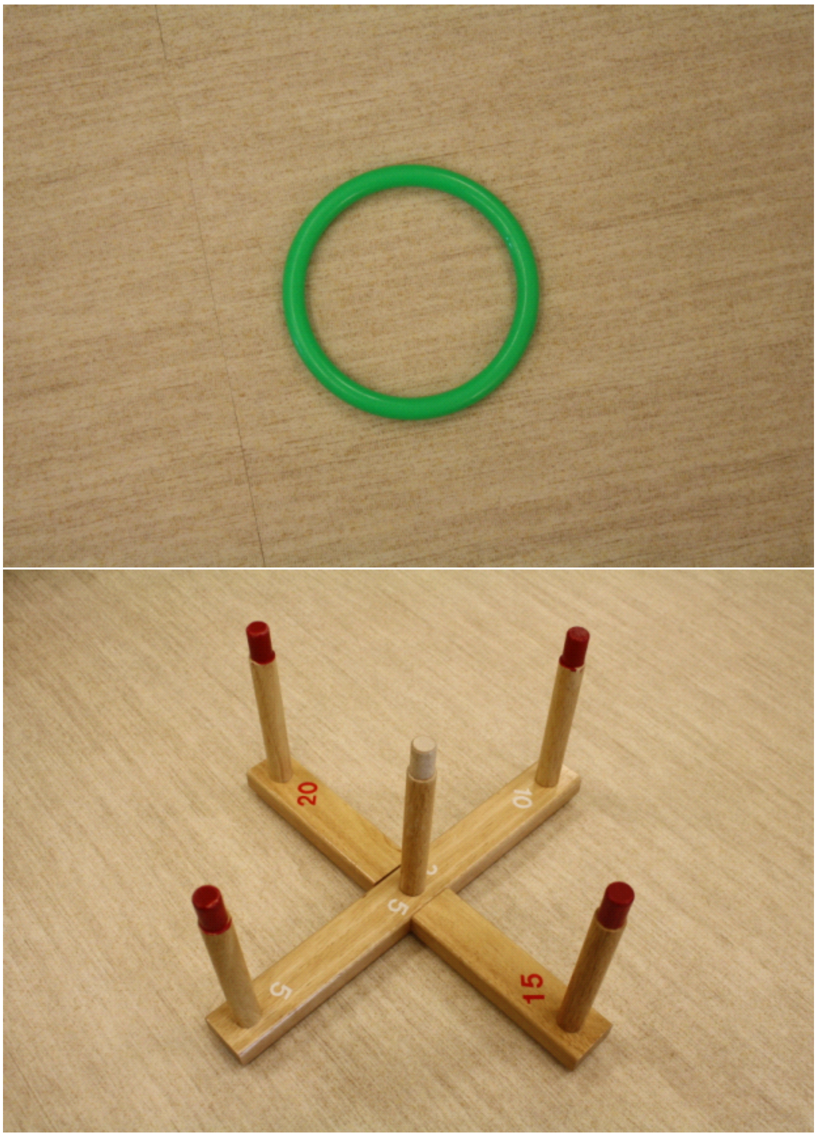


Figure 4.11: A ring and the goal for ring toss.

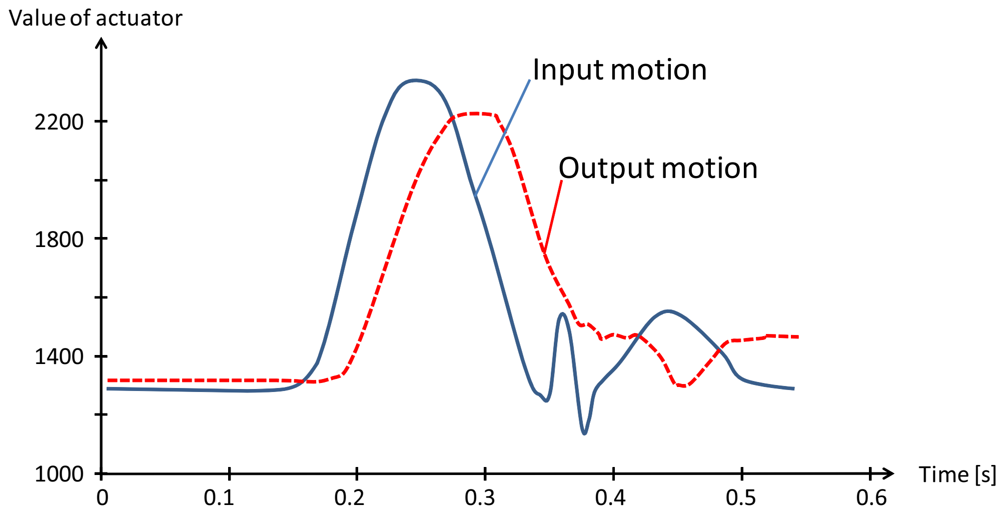


Figure 4.12: Input motion and output motion with delay.

Where, Ref represents desired trajectory of the actuator of a total K frames, and Ref_i represents the desired value of the actuator at i-th frame. $\boldsymbol{q}$ represents the input trajectory of the actuator, and q_i represents the input value of the actuator to i-th frame. $Sim(\boldsymbol{q})$ represents the simulated output of the actuator to the input $\boldsymbol{q}$. $Sim(\boldsymbol{q})_i$ represents the output value of the actuator at i-th frame. In addition, initial entry of $\boldsymbol{q}$ is set to Ref, and lower bounds and upper bounds of values of actuator are set to lb_i and ub_i.

An output trajectory from an compensated input and a desired trajectory for an actuator are shown in Figure 4.13. As the result of the compensation, the output trajectory shown in a green line can be fully following the desired trajectory shown in a blue line. On the other hand, the output from an input without compensation could not follow the desired trajectory as shown in Figure 4.12.

4.4.2 Experimental Condition

Our system extracted style parameters for each of the three players (A, B, and C), from demonstrations captured at Sec. 4.2, then it generated robot motions for each style. Distance between standing position of the robot and the goal is set to 2.5[m]. Instead of an on-board camera system, position of the goal is given manually in the current implementation.

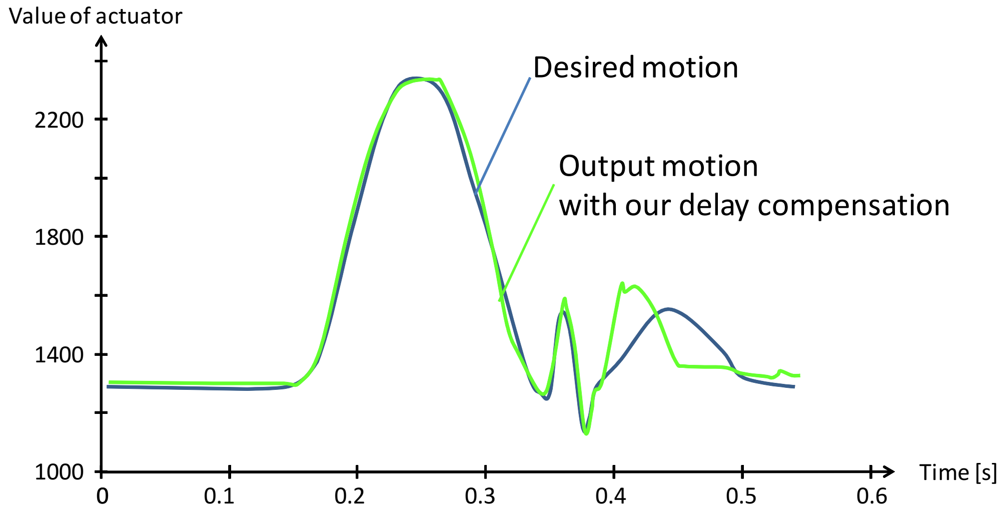


Figure 4.13: Desired motion and output motion with our delay compensation.

4.4.3 Result

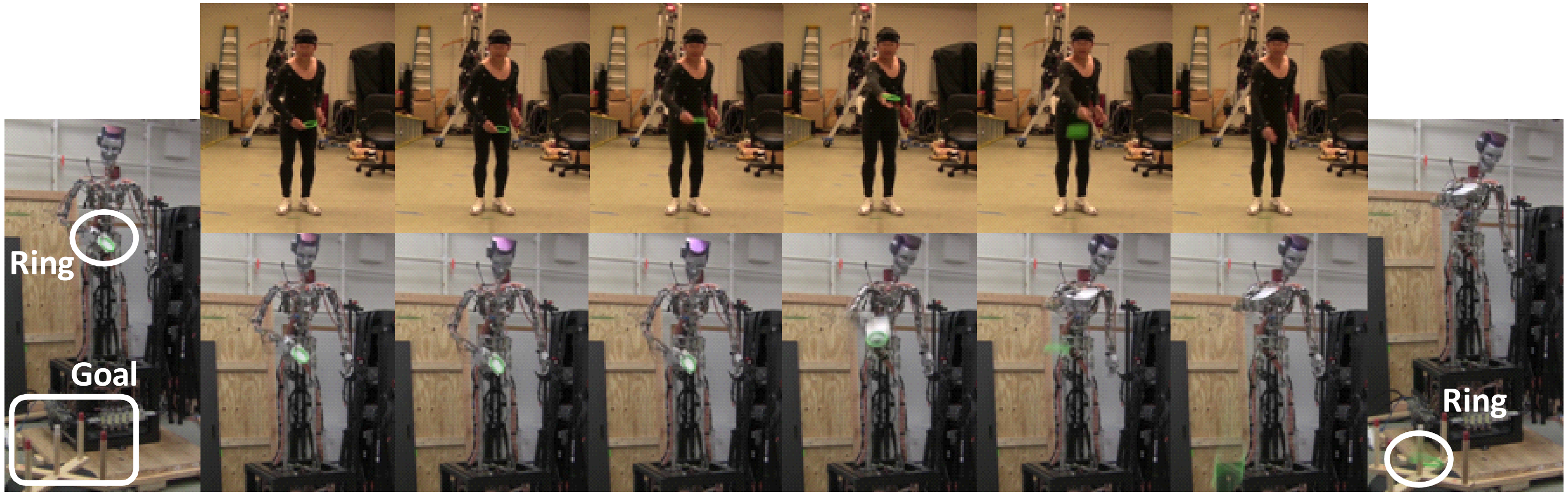


Figure 4.14: Imitation of Motion Style of Player A. Top row: the sequence of the player's demonstration. Bottom row: the sequence of robot motion mimicking that player's style. Player A tends to take the ring back slightly, and toss the ring in the front. His bending is relatively small compared to the others', and his hand position, especially at the end timing of RELEASE task, tends to be higher. These features are imitated by the robot as shown in the picture.

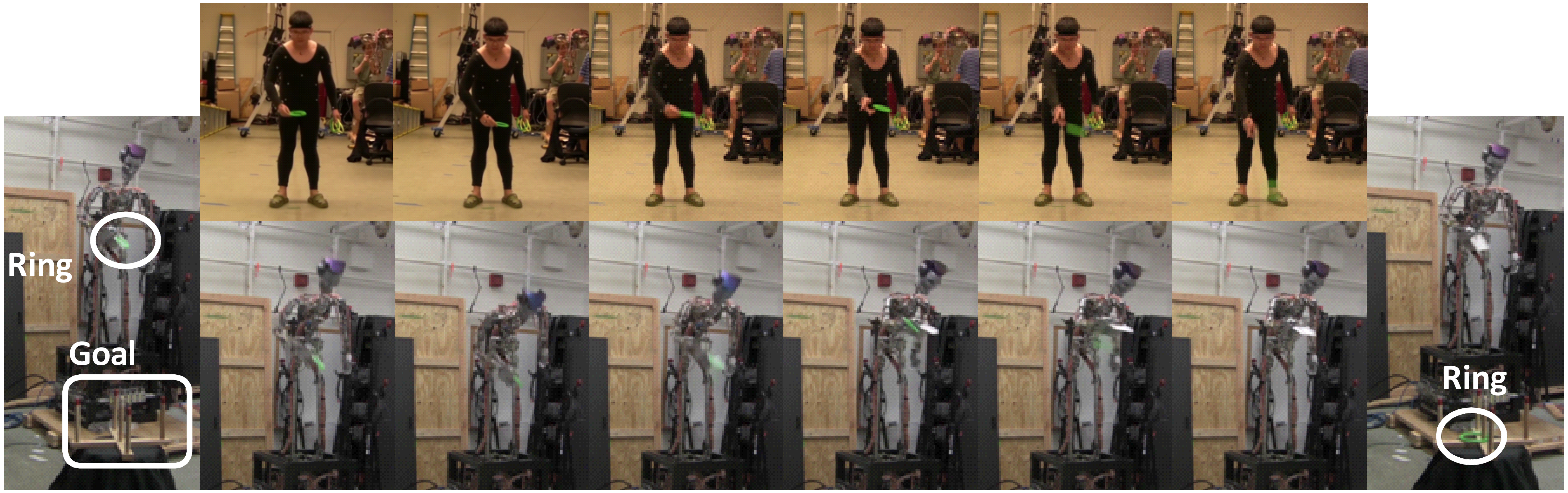


Figure 4.15: Imitation of Motion Style of Player B. Top row: the sequence of the player's demonstration. Bottom row: the sequence of robot motion mimicking that player's style. Player B tends to take the ring back lower, and toss the ring in the front. His bending is relatively deeper than first player, and his hand position during RELEASE task tends to be lower. Although the robot seems to bend excessively, probably to satisfy the required flying distance of the ring within the joint constraints, features described above are imitated by the robot.

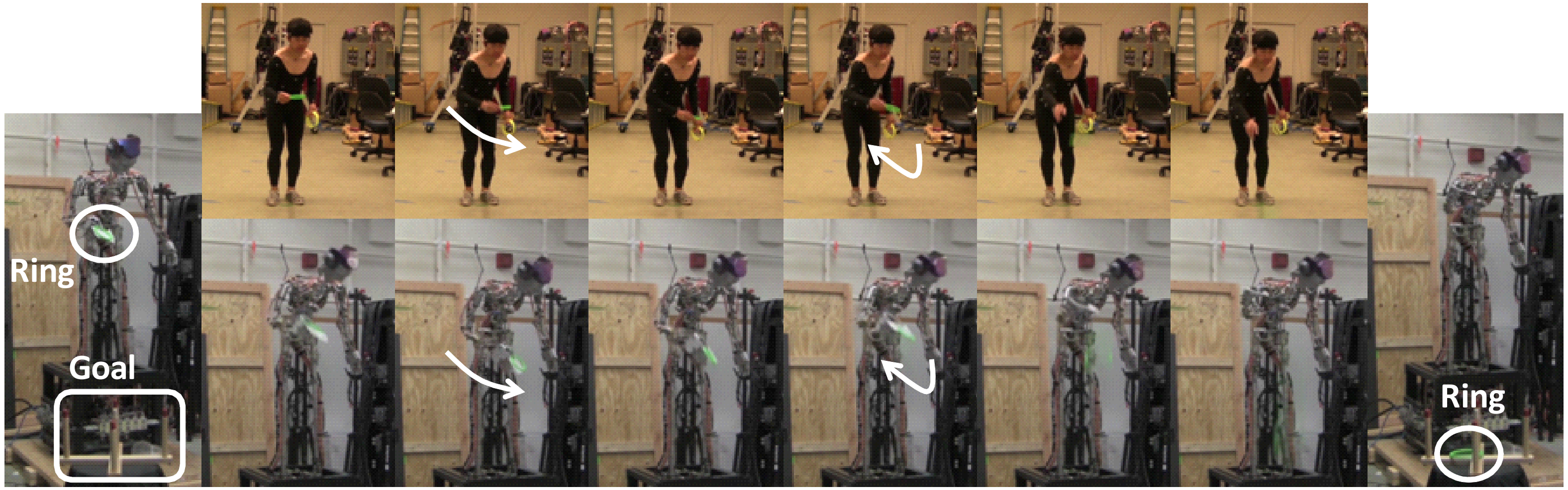


Figure 4.16: Imitation of Motion Style of Player C. Top row: the sequence of the player's demonstration. Bottom row: the sequence of robot motion mimicking that player's style. Player C tends to take the ring back slowly and largely, and throw from the side position. His bending tends to be deep. These features are imitated by the robot as shown in the picture.

The robot imitated a total three types of styles in ring toss motion. The three types of ring toss are shown in Fig. 4.14, Fig. 4.15, and Fig. 4.16. Each upper row shows the sequence of the player's demonstration, and each lower row shows the sequence of robot motion mimicking that player's style.

Player A shown in Fig. 4.14 tends to take the ring back slightly, and toss the ring in the front. His bending is small compared to the others', and his hand position, especially at the end timing of RELEASE task, tends to be higher. These features were imitated by the robot as shown in the picture. Player B shown in Fig. 4.15 tends to take the ring back lower, and toss the ring in the front. His bending is deeper than first player, and his hand position during RELEASE task tends to be lower. Although the robot seems to bend excessively, probably to satisfy the required flying distance of the ring within the joint constraints, features described above were imitated by the robot. Player C shown in Fig. 4.16 tends to take the ring back slowly and largely, and throw from the side position. His bending tends to be deep. These features were imitated by the robot as shown in the picture.

4.5 Discussion

Although we parameterized style using $\bar{s}$ and σ, we may need to improve considering correlation between skill parameters.

Additionally, for validation of our method, we only generated robot motions using style parameters extracted from each of three player and compared with the original motions. But applications such as personal recognitions and interpolation of styles may emphasize our contribution in this chapter. Those will be one of our future work.

In the experiments self collisions did not occur and excess of joint limitations are avoided as shown in Fig. 4.17. However, these factors should be considered as hard constraints in the optimization process for safety. Additionally, success rate of the ring toss was not very high because we did not consider the air and frictional resistances in this implementation. These considerations will be a part of our future work.

To validate the generality of proposed method, we need to apply our method to another kind of target motion and robot platform. We expect our method has equal applicability to previous task models, and we will validate this.

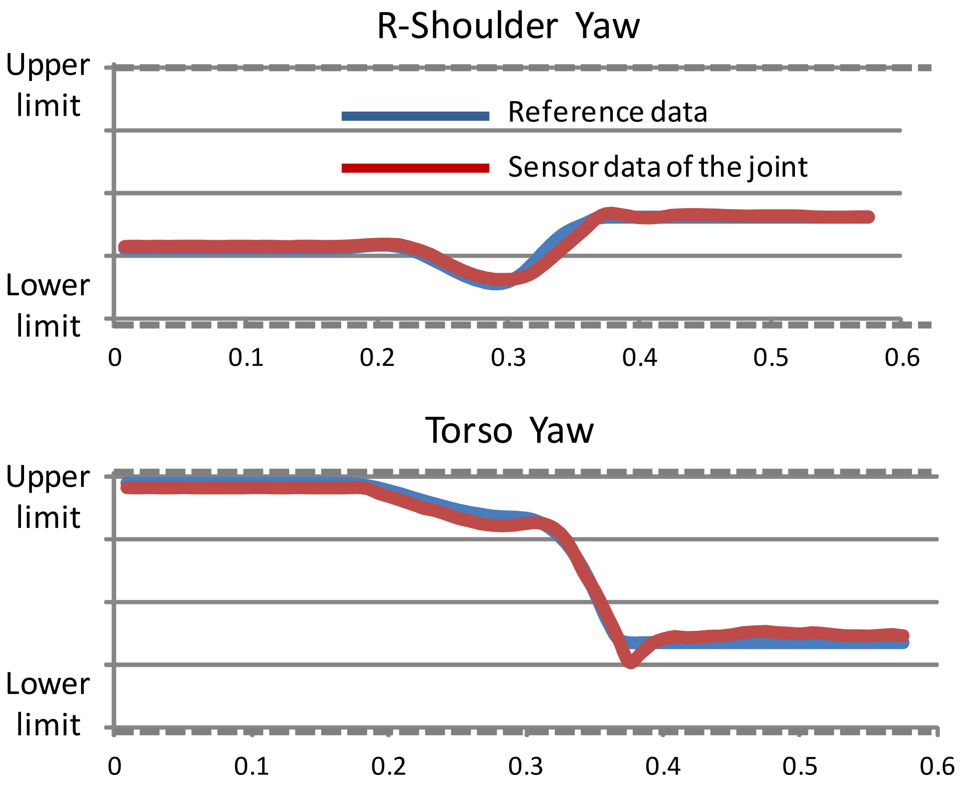


Figure 4.17: Command trajectories of generated robot motions and the trajectories executed by robot: Gray lines show the upper and lower limits. Blue lines show the inputted command trajectories for R-shoulder yaw and torso yaw joints. Red lines show the executed trajectories captured via sensors equipped to each joint of the robot.

4.6 Summary

This chapter presented a method to extract spatial motion styles in motions, and a framework to imitate them using physical humanoid robots. Our approach is an extension of the concept of task model and focuses on such styles in the domain of task representations.

First we chose a ring toss game for a target motion and designed a task model for it. We defined tasks and corresponding skill parameters based on observations. Then we introduced a style parameter to the concept of task model. We observed statistical distributions of skill parameters and used means and variances of them to represent styles. The framework for generation of robot motion consists of three phases: First phase extracts style parameters for each task from human demonstrations. The second phase optimizes a set of skill parameters based on style parameters so that robots can imitate the style while satisfying their physical constraints. The last phase generates whole motions using the fully optimized set of skill parameters.

To verify the proposed framework, we conducted experiments with a physical humanoid robot. The robot performed the ring toss motions imitating each style, while tossing rings to the specific goal. We were able to find features of each style in the robot motions.

Chapter 5

Conclusions

5.1 Summary

In this thesis, we have presented methods to generate variations of whole-body motions, which are feasible for physical humanoid robots. As approaches, we have observed how human motions vary according to constraints such as music tempos or personality. Then we have extracted styles of motions based on insights from the observation. Our proposed method first analyze given human demonstrations via task models defined in a learning-from-observation paradigm, and then extract tasks and skill parameters to be imitated by humanoid robots. Finally, to generate stylistic robot motions, extracted motion styles are used in optimization for executable skill parameters, which represent the motion styles best within physical constraints of the robot.

Temporal Motion Styles and Imitation by a Humanoid Robot

We have presented a method to generate dance motions according to arbitrary musical tempos. Proposed method is based on insights from observation of motion styles specific to temporal constraints. When a same human dancer performs a dance to a same musical piece, the details of the dance movements vary tempo to tempo. It would appear that dance motions are modified, preserving only essential factors as dance, to follow quicker tempos within the ability to exercise. We have focused on these dance variation, which are specific to musical tempos, as temporal motion styles. In this research, we have observed temporal motion styles and presented a method to imitate them using a physical humanoid robot.

First of all, we have observed dance motions performed by three dancers in various tempos of a same musical peace. As a result of observation, we have obtained insights that particular postures, called keyposes, and their timings tend to be preferentially preserved even in extremely quicker tempos. Based on the insights, we have developed a temporal scaling algorithm to generate dance motion variations according to tempos under sever physical constraints of the robot platform.

We have integrated individual temporal scaling algorithms for lower-body, middle-body, and upper-body motions. Those algorithms modify the dance motions by individual process but preserve keyposes in common. Thus we have used keyposes as anchor points of each body parts. Then, we have presented a method for on-line dance motion generation on the assumption that a robot is dancing to a music with time-varying tempo such as live-music. Our on-line generation is also based on keypose, because our observation shows that keyposes are essential in the dance, and proposed temporal scaling techniques are also based on keyposes. In the experiments to validate our algorithm, a physical humanoid robot HRP-4C have been used, and the Don-pan dance have been generated to the original, 0.85 times, and 1.2 times faster tempo. The robot could express keyposes using its whole-body at appropriate timings in the sequence, and provide viewers with an artistic dance pattern in which upper-body and leg motions were fully harmonious. The experiments have demonstrated that our algorithms based on temporal motion style are effective to generate motion variations according to music tempos. Although the temporal motion styles have been obtained from observation of a Japanese folk dance, the Aizu-bandaisan dance, they have been applicable to another folk dance, the Don-pan dance.

Spatial Motion Styles and Imitation by a Humanoid Robot

We have presented a method to generate motion styles which are specific to spatial constraints such as body type, length of limbs, and so on. Especially we have focused on styles, which are specific to the person, as spatial motion styles. Our approach is an extension of the concept of task model and focuses on such styles in the domain of task/skill representations.

First of all, we have chose a ring toss game as an instance of target motion and designed task models for it. We have defined tasks and corresponding skill parameters based on a common structure of ring toss motions, obtained from observation of seven player's demonstrations. Then we have introduced a style parameter into the concept of task model. We have observed statistical distributions of skill parameters and used means and variances of those to represent styles, supposing each of skill parameter are normally-distributed. The framework for generation of robot motion consists of three phases: The first phase extracts style parameters for each task from human demonstrations. The second phase optimizes a set of skill parameters based on style parameters so that robots can imitate the style while satisfying their physical constraints. The last phase generates whole motions using the fully optimized set of skill parameters.

To verify the proposed framework, we have conducted experiments with a physical humanoid robot A100. The robot could perform the ring toss motions imitating each style, while tossing rings to the specific goal. We have found features of each style in the robot motions.

5.2 Contributions

The contributions of this research are as follows:

- We have observed human dancing and analyzed humans modification strategies for the lower- and middle-body motions to dance to a same musical piece in various tempos. This analysis have revealed how human dancing changes according to musical tempos for the whole-body motions, by combining with Shiratori *et al.*'s analysis [SKNI07, SI08].

- We have designed algorithms for temporally scaling of the lower- and the middle-body motions based on the observation. These algorithms allow robots to modify the speed of dancing while avoiding overload of joint motors.

- We have integrated the lower-, middle-, and upper-body which are modified separately according to musical tempos using different strategies, using keyposes as anchor points. This integration allow robots to synchronize the lower, middle, and upper-body motions, modified based on different strategies.

- We have proposed a framework for on-line motion generation based on keyposes and effective refinement of faults in motions.

- We have conducted preliminary experiments with a physical humanoid robot HRP-4C. These experiments validated our system can generate feasible whole-body motions, in which each body parts are fully harmonious, for arbitrary music tempos.

- We have conducted an experiment of keypose-based on-line motion generation with a physical humanoid robot HRP-4C. Although we have changed dancing speed twice in midstream of the whole sequence of dance motions, computation time was enough small and the robot could perform without falling down.

- We have designed task models for ring toss motions and it allowed humanoid robots to imitate the ring toss motions by human players.

- We have observed how skill parameters varies according to individuality. This observation revealed that statistical distributions of skill parameters can be used to identify the individuals.

- We have defined style parameter and it allowed to characterize the person-specific motion styles, which we call spatial motion styles.

- We have designed a framework to optimize the skill parameters based on style parameters. It allowed robots to preserve spatial motion styles in motion imitation as much as possible within the physical constraints.

- We have proposed a method for delay compensation in robot control. It made it possible to compensate the control error of a physical humanoid robot.

- We have conducted experiments with a physical humanoid robot A100. These experiments validated our method can preserve features of each style in the robot motions.

5.3 Discussion

In this thesis we have introduced a concept of motion style in a learning-from-observation paradigm. We have characterized how humans move according to specific constraints. As examples of those constraints, we have handled temporal and spatial constraints.

For examples of temporal motion style, we have focused on styles which are specific to music tempo in the chapter 3. As a result of observation of human dancing in various music tempo, we could extract statistical tendencies which are common among dancers. The fact shows that such tendencies are actually specific to music tempos and are independent on dancers.

Insights relevant to preserving keyposes are corresponds to those reported by Shiratori *et al*. [SKNI07, SI08]. Considering their observation were focusing on upper body motions performed by other dancers, those insights can be quite general in the domain of dance. In the chapter 3, we applied the temporal motion styles extracted from the Aizu-bandaisan dance to Don-pan dance. The generated dance motions were quite natural, and feasible for a humanoid robot. This suggest the applicability of our proposed method in the domain of dance. We believe that our method can be applied to dances which consist of transition between keyposes. To widen the applicability of our temporal motion style, observation of motion styles in slower tempos and other kind of dance motion such as waltz will be necessary in future works.

On ther other hand, we extracted person-specific motion styles, as one of spatial motion styles in chaper 4. Motion styles extracted from multiple demonstrations of a same specific person are parameterized as style parameters, and used for robot motion generation. We could find features of the styles in the statistical distribution of skill parameters and generated robot motions. We believe that proposed method can be applied to dance for parameterizing person-specific keyposes. Additionally statistical distribution of skill parameters shows that human motions by a same person are slightly different every time. Robot motion generation considering such factor can make the robot more human-like. These are also our future work.

5.4 Future Work

Integration with a Real-time Music Analysis Our system for a dancing-to-music capability enabled robots to modify dance motions on-line according to music speed. However, to achieve this, integration with a real-time music analysis will be necessary to recognize the musical beats while dancing. For such purpose, there are number of researches for detection of musical beats in motor noise of a dancing robot.

A Strategy for Real-time Dance Speed Control To achieve a dancing-to-music capability, how to control the dancing speed to keep up with the music will also be issue. To keep up with the music, the robot need to predict musical tempos of a few seconds later and compute the appropriate dancing speed.

Validation of Spatial Motion Styles in Other Motions For validation of spatial motion styles, we only generated robot motions using style parameters extracted from each of three player and compared with the original motions. But applications such as personal recognitions and interpolation of styles may emphasize the possibility of application.

Collision Avoidance in a Non-linear Optimization Problem In the experiments in Chapter 4, self collisions did not occur and excess of joint limitations are avoided as shown in Fig. 4.17. However, these factors should be considered as hard constraints in the optimization process for safety.

Appendix A

Motion Capture System

This chapter describe about motion capture system we used to capture the human motions. In this thesis, we used two type of motion capture system.

To capture the Aizu-bandaisan dance 3 and ring toss motions in chapter 4, we used an optical motion capture system from VICON. We attached over 30 optical markers to the whole-body of subject as shown in Figure A.1, and then the subject performed demonstrations being surrounded by the VICON cameras. The system recorded the positions of the optical markers in a Cartesian coordinate in C3D format. The frame rate was fixed to 120 [fps] in our observation.

To capture the Don-pan dance 3, we used MotionStar Wireless, a magnetic motion capture system from Ascension. We attached 15 markers to the whole-body of subject as shown in Figure A.2, and then the subject performed demonstrations in front of a transmitter. The system recoded the positions in a Cartesian coordinate and rotation of the magnetic markers in FBX format. The frame rate was fixed to 30 [fps] in our observation.

To process each type of input motions in the same manner, we converted the optical/magnetic motion capture data to a common configuration as shown in Figure A.3. The motion data in this unified configuration are used for observation of motion styles and inputted to our system for robot motion generation. The frame rate of input data was fixed to 120 [fps] in this thesis.

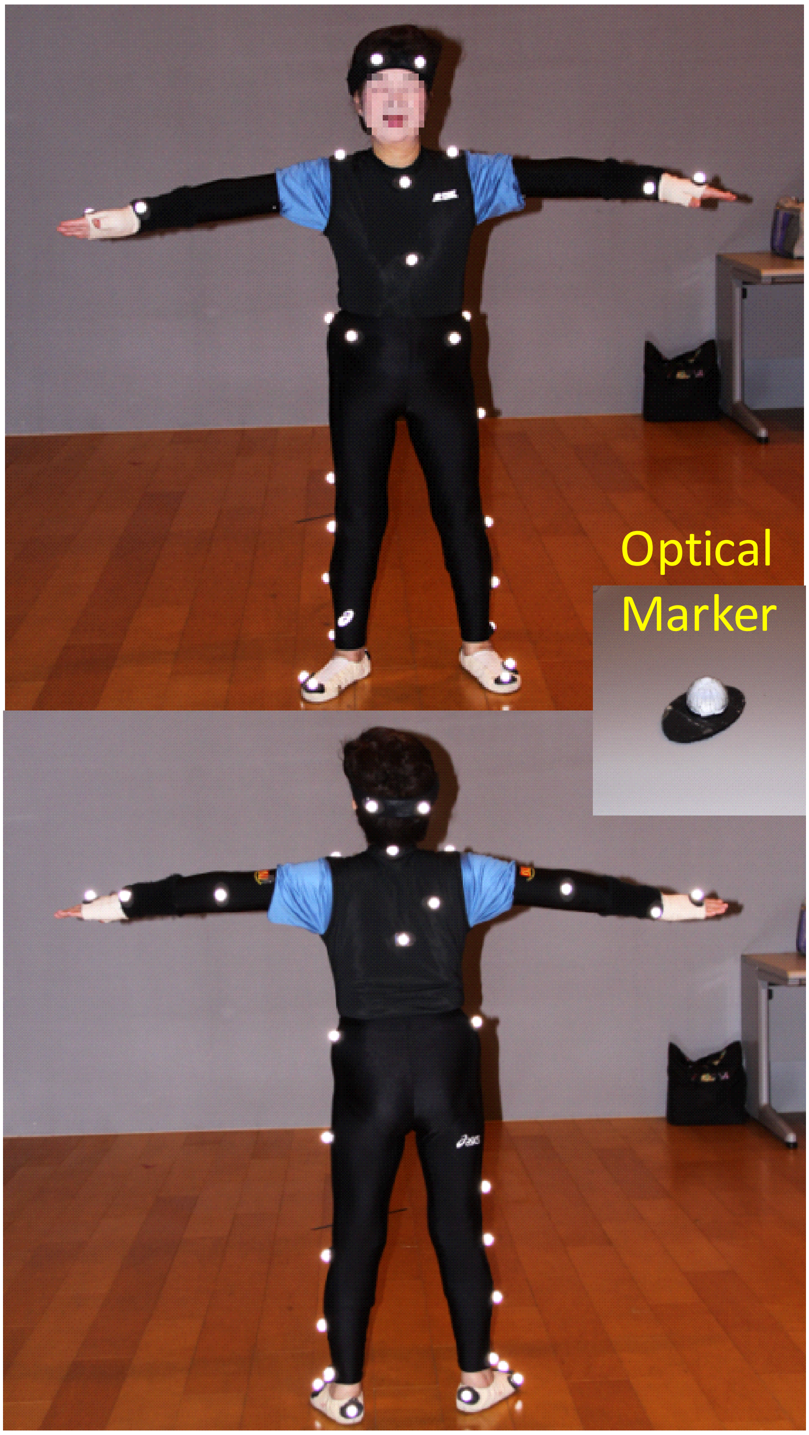


Figure A.1: Optical motion capture system from VICON. We used this system to capture the Aizu-bandaisan dance in chapter 3 and ring toss motions in chapter 4.

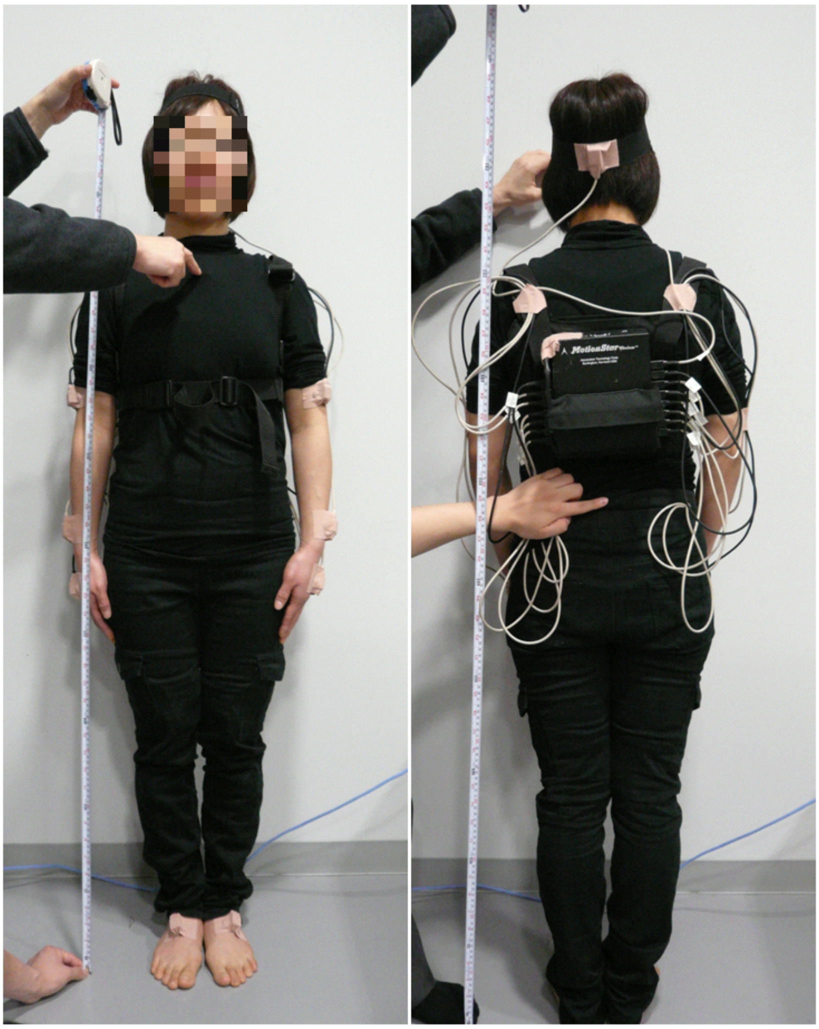


Figure A.2: Magnetic motion capture system from Ascension. To capture the Don-pan dance, totally 15 markers are attached to the dancer in chapter 3.

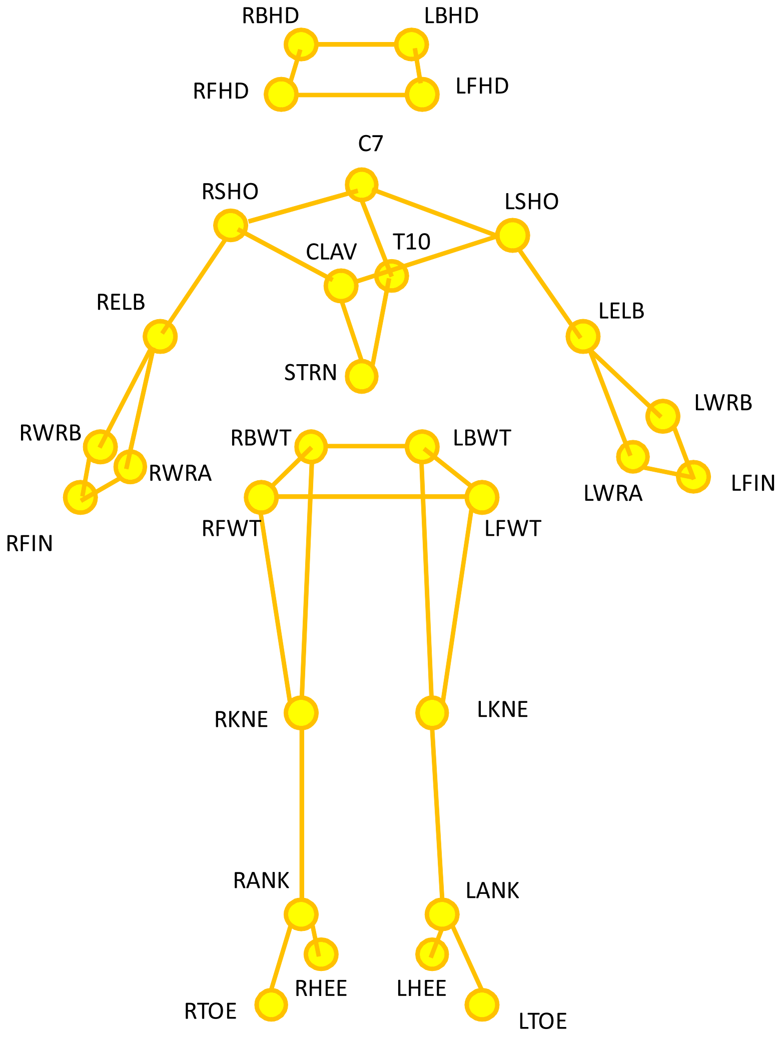


Figure A.3: A common configuration of input data. We unified configuration of input data of our proposed system in this thesis as shown in this figure.

Appendix B

The Aizu-bandaisan Dance by HRP-2

In Chapter 3, we proposed a method to imitate temporal motion styles using a humanoid robot HRP-4C. And we used the Don-pan dance for validation. This appendix shows additional experimental results of our proposed method reported in our preliminary experiments [OSKI10, OSK*14]. In the experiments [OSKI10, OSK*14], the Aizu-bandaisan dance was performed by HRP-2. Target tempo was the original, 1.2 times faster, and 1.5 times faster than the original tempos.

Figure B.1 shows a demonstration of the Aizu-bandaisan dance performed by a dance master. Pictures of No.1, No.2, No.3, No.4, and No.9 correspond keyposes of the Aizu-bandaisan dance.

Figure B.2 shows a demonstration of the Aizu-bandaisan dance at the original tempo performed by HRP-2. The task sequences for dance motions at the original tempo are learned from a demonstration performed by the dance master, and whole body motions for HRP-2 are obtained from the task sequences based on Nakaoka system [NNK*07].

Figure B.3 shows a demonstration of the Aizu-bandaisan dance at the tempo 1.2 times faster than the original performed by HRP-2. The HRP-2 could performed the dance without exceeding the physical constraints, expressed the keyposes using the whole-body in appropriate timings in the sequence, and provided viewers with an artistic dance pattern in which upper-body and leg motions were fully harmonious.

Figure B.4 shows a demonstration of the Aizu-bandaisan dance at the tempo 1.5 times faster than the original performed by HRP-2. At this tempo, strides of

STEP tasks were notedly reduced to follow the faster tempo. However HRP-2 could performed the dance even at this tempo within the limitations.

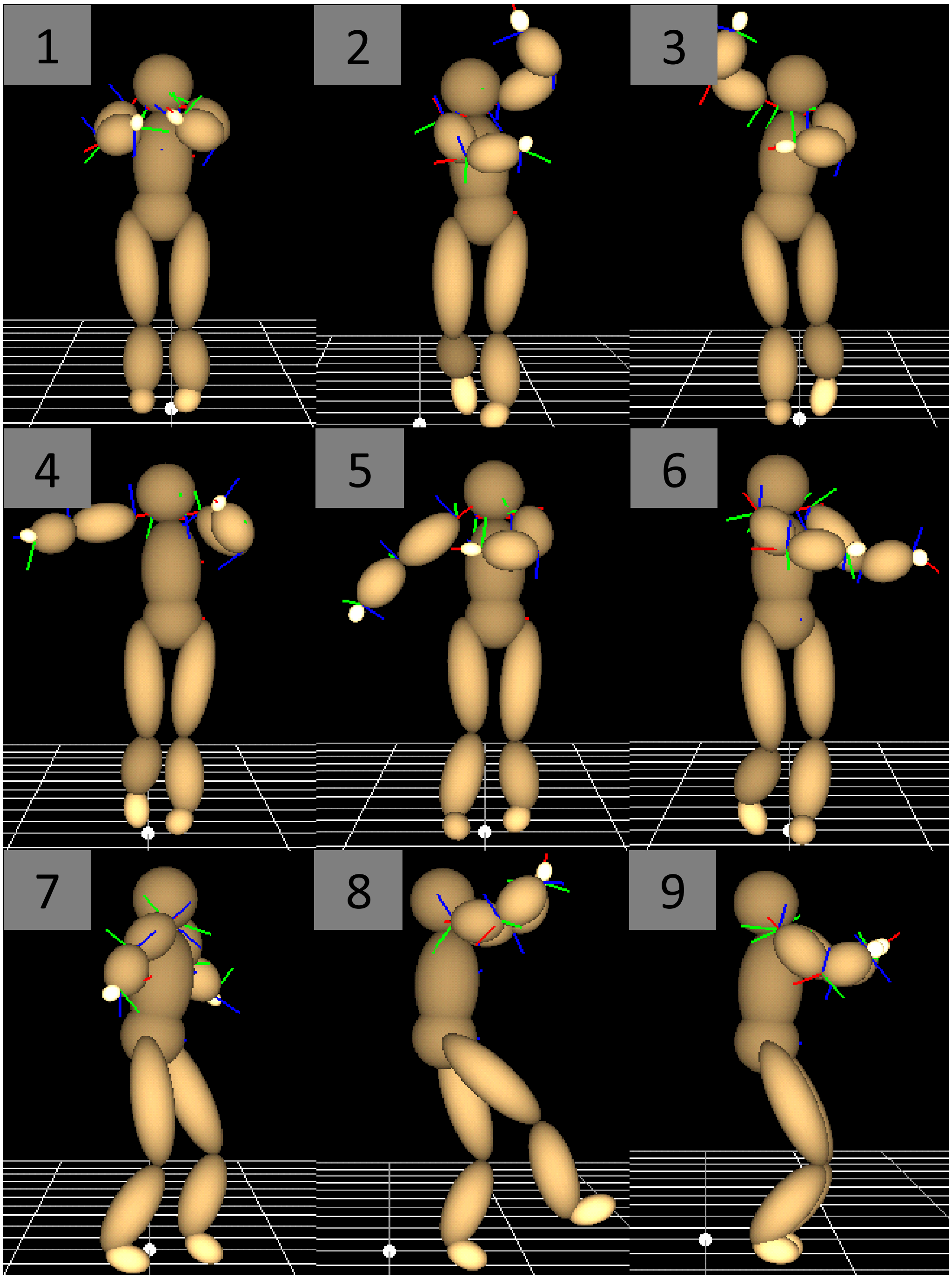


Figure B.1: Demonstration of the Aizu-bandaisan dance performed by a dance master. No.1, No.2, No.3, No.4, and No.9 correspond keyposes of the Aizu-bandaisan dance.

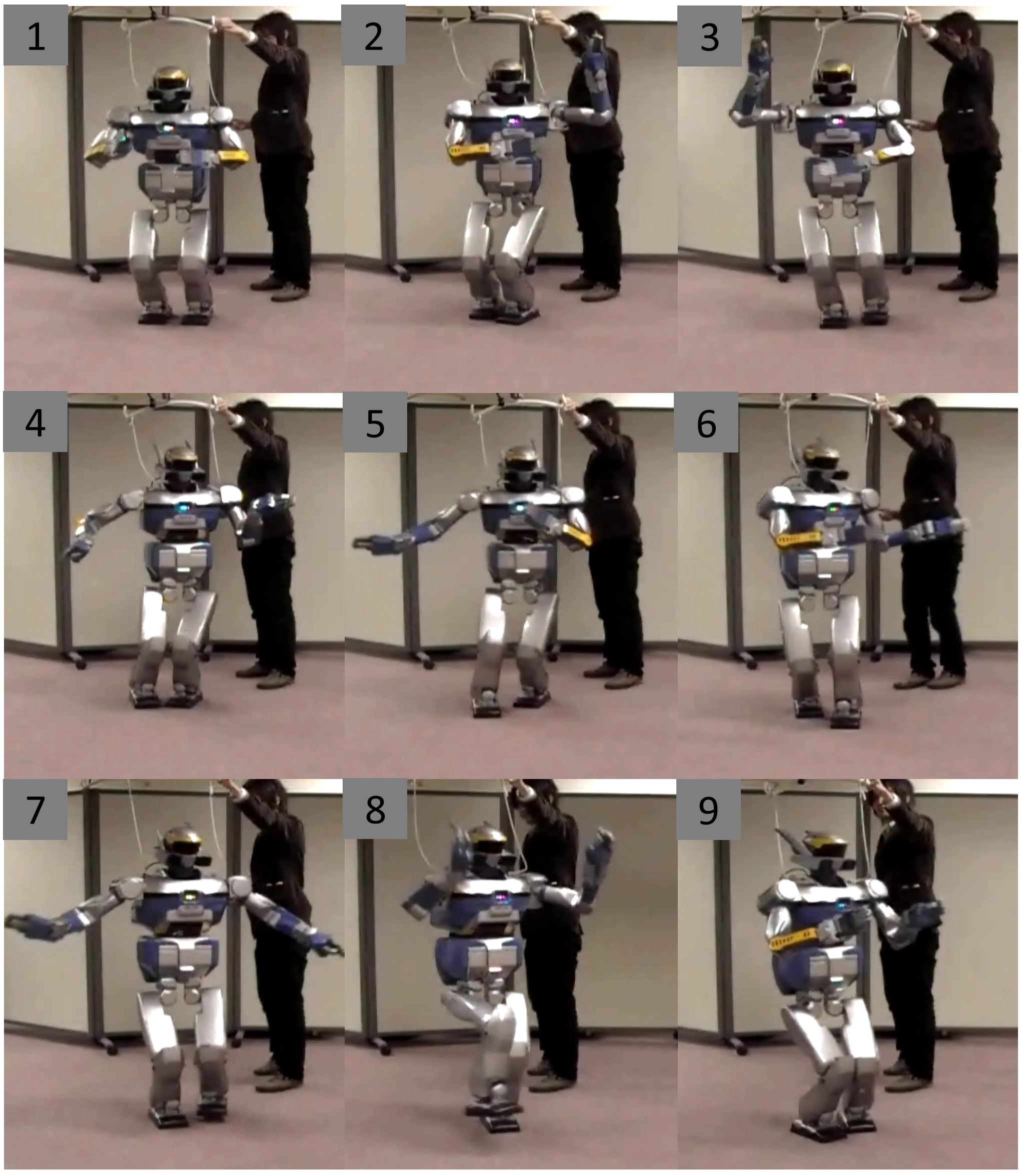


Figure B.2: Demonstration of the Aizu-bandaisan dance at the original tempo performed by HRP-2.

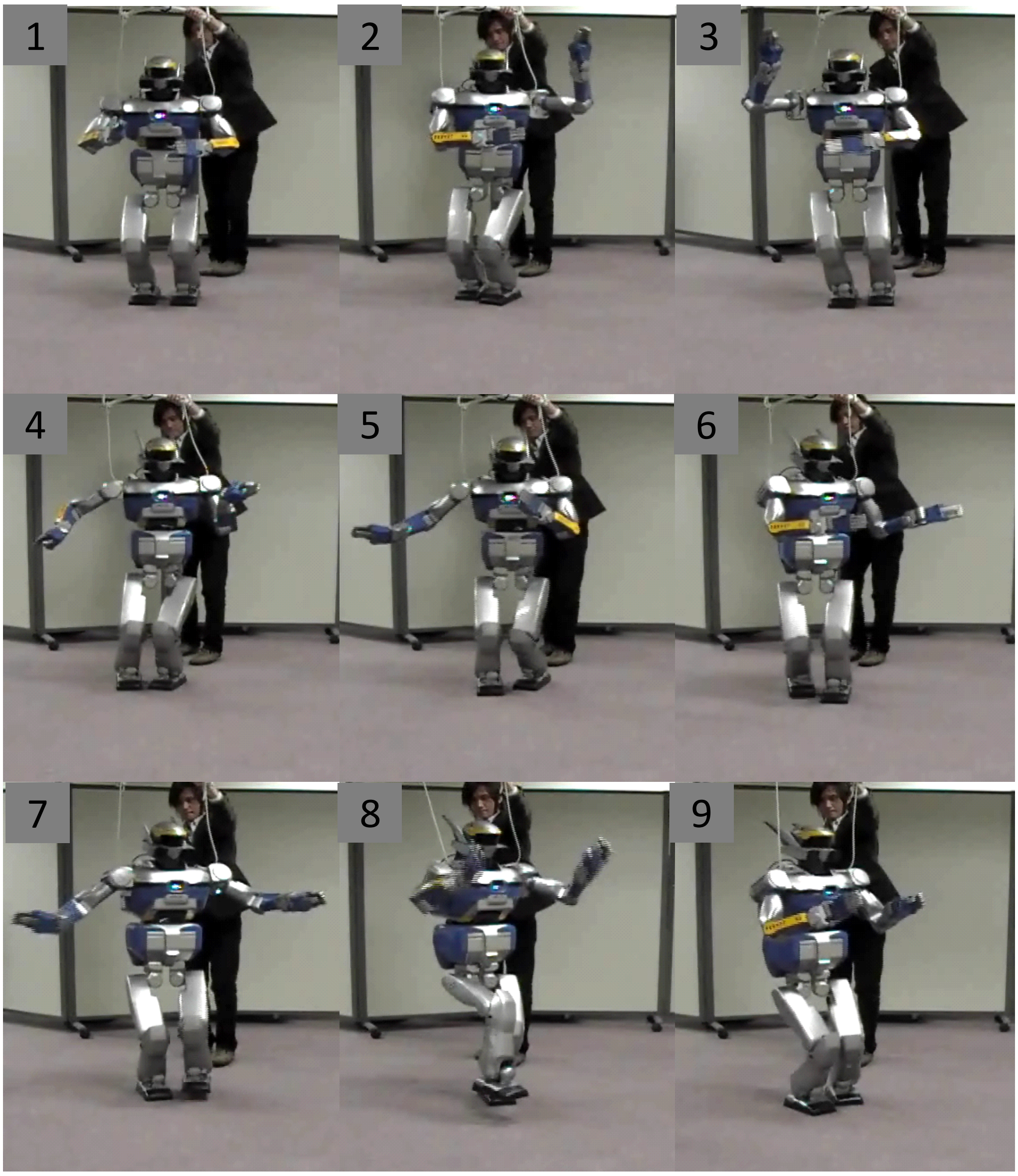


Figure B.3: Demonstration of the Aizu-bandaisan dance at the tempo 1.2 times faster than the original performed by HRP-2.

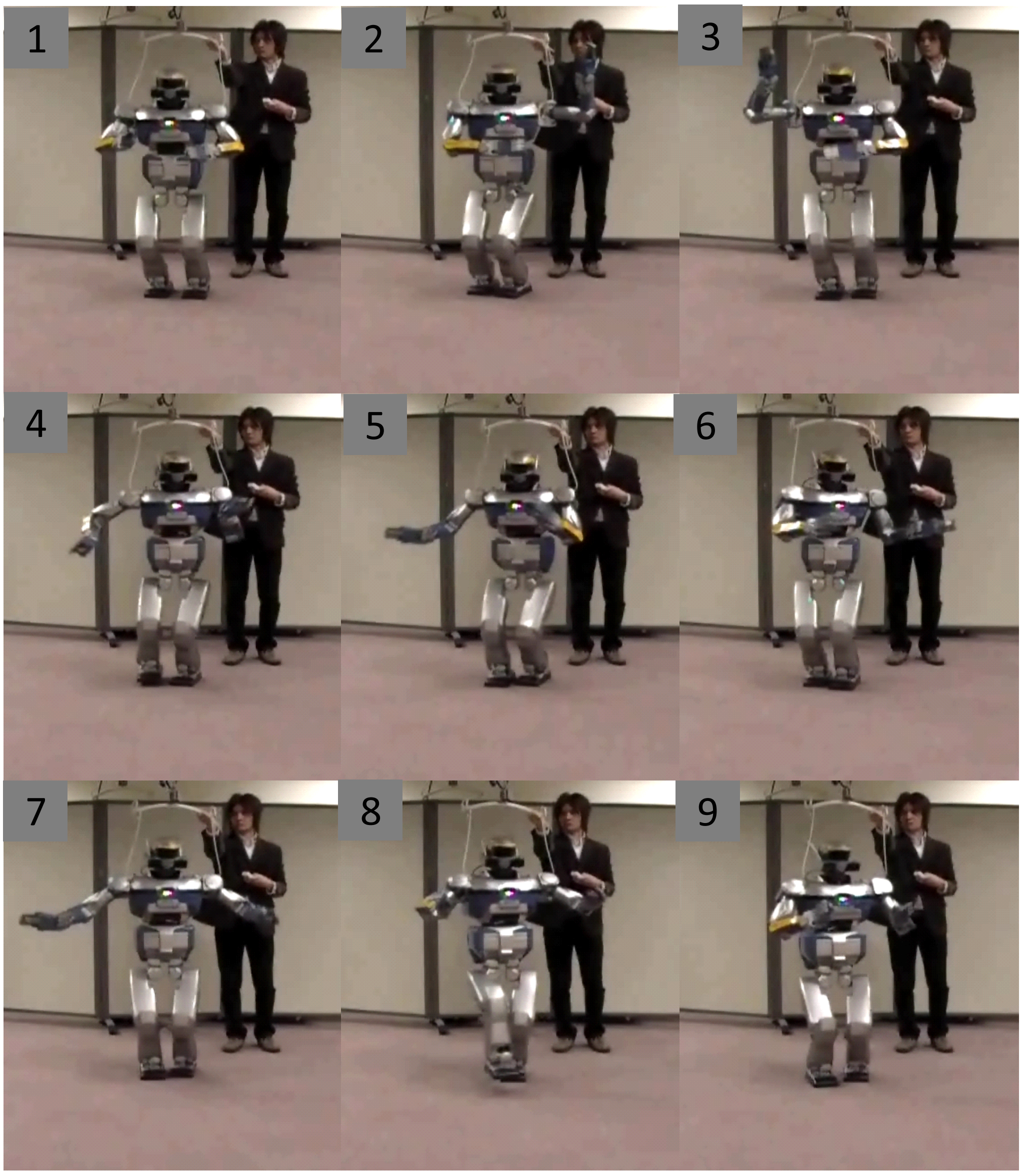


Figure B.4: Demonstration of the Aizu-bandaisan dance at the tempo 1.5 times faster than the original performed by HRP-2.

References

[ABB05] Alankus G., Bayazit A. A., Bayazit O. B.: Automated motion synthesis for dancing characters. *Computer Animation and Virtual Worlds 16*, 3-4 (Sep. 2005), 259–271.

[BH00] Brand M., Hertzmann A.: Style machines. *Proc. of SIGGRAPH-2000* (2000), 183–192.

[BW95] Bruderlin A., Williams L.: Motion signal processing. *Proc. of ACM SIGGRAPH-95* (1995), 97–104.

[CGB07] Calinon S., Guenter F., Billard A.: On learning, representing, and generalizing a task in a humanoid robot. *IEEE trans. systems, man and cybernetics, Part B 37*, 2 (2007), 286–298.

[GEKO09] Grunberg D., Ellenberg R., Kim Y., Oh P.: Creating an autonomous dancing robot. *Proc. of the International Conference on Hybrid Information Technology (ICHIT)* (2009), 221–227.

[Got07] Goto A.: Musical instrument playing robot: Toyota partner robots for the 2005 world exposition, aichi, japan. *Proc. of IROS-2007 Workshop Art and Robots* (Nov. 2007), 41–45.

[HKG06] Heck R., Kovar L., Gleicher M.: Splicing upper-body actions with locomotion. *Computer Graphics Forum 25*, 3 (Sept. 2006), 459–466.

[HPP05] Hsu E., Pulli K., Popovic J.: Style translation for human motion. *Proc. of ACM SIGGRAPH-2005 24* (2005), 1082–1089.

[IKS93] Ikeuchi K., Kawade M., Suehiro T.: Assembly task recognition with planar, curved and mechanical contacts. *Proc. of Robotics and Automation 2* (May 1993), 688–694.

[IS94] Ikeuchi K., Suehiro T.: Toward an assembly plan from observation. i. task recognition with polyhedral objects. *Trans. Robotics and Automation 10*, 3 (1994), 368–385.

[ISK*08] Ikeuchi K., Shiratori T., Kudoh S., Hirukawa H., Nakaoka S., Kanehiro F.: Robots that learn to dance from observation. *Intelligent Systems 23*, 2 (Mar./Apr. 2008), 74–76.

[ITN03a] Inamura T., Tanie H., Nakamura Y.: From stochastic motion generation and recognition to geometric symbol developement and manipulation. *Proc. of Humanoids-2003* (2003).

[ITN03b] Inamura T., Tanie H., Nakamura Y.: Keyframe compression and decompression for time series data based on the continuous hidden markov model. *Proc. of IROS-2003 2* (Oct. 2003), 1487–1492.

[ITTN03] Inamura T., Tanie H., Toshima I., Nakamura Y.: An approach from motion generation/recognition to intelligence based on mimesis principle. *Proc. of AMAM-2003* (Mar. 2003), SaA–II–1.

[JM02] Jenkins O. C., Mataric M. J.: Deriving action and behavior primitives from human motion data. *Proc. of IROS-2002 3* (2002), 2551–2556.

[KFI*03] Kuroki Y., Fujita M., Ishida T., Nagasaka K., Yamaguchi J.: A small biped entertainment robot exploring attractive applications. *Proc. of ICRA-2003 1* (Sept. 2003), 471–476.

[KHHT03] Kosuge K., Hayashi T., Hirata Y., Tobiyama R.: Dance partner robot -Ms DancerR-. *Proc. of IROS-2003 3* (Oct. 2003), 3459–3464.

[KI97] Kang S. B., Ikeuchi K.: Toward automatic robot instruction from perception: Mapping human grasps to manipulator grasps. *Trans. Robotics and Automation 13*, 1 (1997), 81–95.

[KKM*09] Kaneko K., Kanehiro F., Morisawa M., Miura K., Nakaoka S., Kajita S.: Cybernetic human hrp-4c. *Proc. of Humanoids-2009* (Dec. 2009), 7–14.

[KMN09] Kozima H., Michalowski M. P., Nakagawa C.: Keepon: A playful robot for research, therapy, and entertainment. *Int'l Journal of Social Robotics 1*, 1 (Jan. 2009), 3–18.

[KNG*11] Kajita S., Nakano T., Goto M., Matsusaka Y., Nakaoka S., Yokoi K.: Vocawatcher: Natural singing motion generator for a humanoid robot. *Proc. of IROS-2011* (2011), 2000–2007.

[LL05] Lee H. C., Lee I. K.: Automatic synchronization of background music and motion in computer animation. *Computer Graphics Forum 24*, 3 (Sept. 2005), 353–361.

[LS99] Lee J., Shin S. Y.: A hierarchical approach to interactive motion editing for human-like figures. *Proc. of ACM SIGGRAPH-1999* (1999), 39–48.

[MK98] Miyamoto H., Kawato M.: A tennis serve and upswing learning robot based on bi-directional theory. *Neural networks: the official journal of the International Neural Network Society* (1998).

[MLO10] Mizumoto T., Lim A., Otsuka T.: Integration of flutist gesture recognition and beat tracking. *The First IROS 2010 Workshop on Robots and Musical Expressions* (2010).

[MNT04] Madsen K., Nielsen H. B., Tingleff O.: Methods for non-linear least squares problems (2nd ed.). 60.

[MNY*08] Murata K., Nakadai K., Yoshii K., Takeda R., Torii T., Okuno H. G., Hasegawa Y., Tsujino H.: A robot uses its own microphone to synchronize its steps to musical beats while scatting and singing. *Proc. of IROS-2008* (Sept. 2008), 2459–2464.

[MPS06] McCann J., Pollard N. S., Srinivasa S.: Physics-based motion retiming. *Proc. of ACM SIGGRAPH-2006* (2006), 205–214.

[MYN13] Moulard T., Yoshida E., Nakaoka S.: Optimization-based motion retargeting integrating spatial and dynamic constraints for humanoid. *Int. Symposium on Robotics* (2013), 1–6.

[Nak12] Nakaoka S.: Choreonoid: Extensible virtual robot environment built on an integrated gui framework. *Proc. of the 2012 IEEE/SICE Inter. Sympo. on System Integration (SII2012)* (2012), 79–85.

[NK09] Neff M., Kim Y.: Interactive editing of motion style using drives and correlations. *Proc. of SCA-2009* (2009), 103–112.

[NK12] Nakaoka S., Komura T.: Interaction mesh based motion adaptation for biped humanoid robots. *Proc. of Humanoids-2012* (2012), 625–631.

[NkK*02] Nishiwaki K., kagami S., Kuniyoshi Y., Inaba M., Inoue H.: Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired zmp. *Proc. of IROS-2002 3* (Oct. IROS2002), 2684–2689.

[NNI04] Nakazawa A., Nakaoka S., Ikeuchi K.: Matching and blending human motions using temporal scaleable dynamic programming. *Proc. of IROS-2004 1* (2004), 287–294.

[NNK*07] Nakaoka S., Nakazawa A., Kanehiroand F., Kaneko K., Morisawa M., Hirukawa H., Ikeuchi K.: Learning from observation paradigm: Leg task models for enabling a biped humanoid robot to imitate human dances. *Int'l J. of Robot. Res. 26*, 8 (2007), 829–844.

[OGR08] Oliveira J., Gouyon F., Reis L. P.: Towards an interactive framework for robot dancing applications. *Proc. of the International Conference on Digital Arts (ARTECH)* (2008).

[OSK*14] Okamoto T., Shiratori T., Kudoh S., Nakaoka S., Ikeuchi K.: Toward a dancing robot with listening capability: Keypose-based integration of lower, middle, and upper body motions for varying music tempos. *Transactions on Robotics 30* (June 2014), 771–778.

[OSKI10] Okamoto T., Shiratori T., Kudoh S., Ikeuchi K.: Temporal scaling of leg motion for sound feedback system of a dancing humanoid robot. *Proc. of IROS-2010* (Oct. 2010), 2256–2263.

[OTKI03] Ogawara K., Takamatsu J., Kimura H., Ikeuchi K.: Extraction of essential interactions through multiple observations of human demonstrations. *Transaction on Industrial Electronics 50*, 4 (Aug. 2003), 667–675.

[PHRA02] Pollard N. S., Hodgins J. K., Riley M. J., Atkeson C.: Adapting human motion for the control of a humanoid robot. *Proc. of ICRA-2002 2* (2002), 1390–1397.

[SAD10] Schollig A., Augugliaro F., D'Andrea R.: A platform for dance performances with multiple quadrocopters. *Proc. of IROS-2010 Workshop on Robots and Musical Expressions* (Oct. 2010).

[SC78] Sakoe H., Chiba S.: Dynamic programming algorithm optimization for spoken word recognition. *Transactions on Acoustics, Speech and Signal Processing 26* (Feb. 1978), 43–49.

[SCF06] Shapiro A., Cao Y., Faloutsos P.: Style components. *Proc. of GI-2006* (2006), 33 – 39.

[Sch99] Schaal S.: Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences 3*, 6 (Jun. 1999), 233–242.

[SI92] Suehiro T., Ikeuchi K.: Towards an assembly plan from observation: Part ii: Correction of motion parameters based on fact contact constraints. *proc. of IROS-1992 3* (1992), 2095–2102.

[SI08] Shiratori T., Ikeuchi K.: Synthesis of dance performance based on analyses of human motion and music. *Transactions on Computer Vision and Image Media 1*, 1 (June 2008), 34–47.

[SKNI07] Shiratori T., Kudoh S., Nakaoka S., Ikeuchi K.: Temporal scaling of upper body motion for sound feedback system of a dancing humanoid robot. *Proc. of IROS-2007* (Oct. 2007), 3251–3257.

[SNI04] Shiratori T., Nakazawa A., Ikeuchi K.: Detecting dance motion structure through music analysis. *Proc. of FGR-2004* (May. 2004), 857–862.

[SNI06] Shiratori T., Nakazawa A., Ikeuchi K.: Dancing-to-music character animation. *Computer Graphics Forum 25*, 3 (Sept. 2006), 449–458.

[SW11] Shibata T., Wada K.: Introduction of field test on robot therapy by seal robot, paro. *Journal of the Robotics Society of Japan 29*, 3 (2011), 246–249.

[SWA*02] Sakagami Y., Watanabe R., Aoyama C., Matsunaga S., Higaki N., Fujimura K.: The intelligent asimo: System overview and integration. *Proc. of IROS-2002 3* (2002), 2478–2483.

[SYK*08] Suleiman W., Yoshida E., Kanehiro F., Laumond J. P., Monin A.: On human motion imitation by humanoid robot. *Proc. of ICRA-2008* (2008), 2697–2704.

[TFAM05] Tanaka F., Fortenberry B., Aisaka K., Movellan J.: Plans for developing real-time dance interaction between qrio and toddlers in a classroom environment. *Proc. of ICDL-2005* (Jul. 2005), 142–147.

[TH00] Tanco L. M., Hilton A.: Realistic synthesis of novel human movements from a database of motion capture examples. *Proc. of Workshop on HUMO-2000* (2000), 137 – 142.

[THB06] Torresani L., Hackney P., Bregler C.: Learning motion style synthesis from perceptual observations. *proc. of NIPS-2006* (2006), 1393–1400.

[TMO*06a] Takamatsu J., Morita T., Ogawara K., Kimura H., Ikeuchi K.: Representation for knot-tying tasks. *Trans. Robotics 22*, 1 (2006), 65–78.

[TMO*06b] Takamatsu J., Morita T., Ogawara K., Kimura H., Ikeuchi K.: Representation for knot-tying tasks. *Transaction on Robotics 22*, 1 (Feb. 2006), 65–78.

[TS04] Tanaka F., Suzuki H.: Dance interaction with qrio: A case study for non-boring interaction by using an entertainment ensemble model. *Proc. of ROMAN-2004* (Oct. 2004).

[Ude99] Ude A.: Filtering in a unit quaternion space for model-based object tracking. *Robotics and Autonomous Systems 28*, 2-3 (Aug. 1999), 163–172.

[UGB*04] Urtasun R., Glardon P., Boulic R., Thalmann D., Fua P.: Style-based motion synthesis. *Computer Graphics Forum 23*, 4 (Dec. 2004), 799–812.

[WDP05] Weinberg G., Driscoll S., Parry M.: Musical interactions with a perceptual robotic percussionist. *Proc. of ROMAN-2005* (Aug. 2005), 456–461.

[YNT*07] Yoshii K., Nakadai K., Torii T., Hasegawa Y., Tsujino H., Komatani K., Ogata T., Okuno H. G.: A biped robot that keeps steps in time with musical beats while listening to music with its own ears. *Proc. of IROS-2007* (Oct. 2007), 1743–1750.

[YRA13] Yamane K., Revfi M., Asfour T.: Synthesizing object receiving motions of humanoid robots with human motion database. *Proc. of ICRA-2013* (2013).

List of Publications

Journal Papers

1. Takahiro Okamoto, Takaaki Shiratori, Shunsuke Kudoh, Shinichiro Nakaoka, and Katsushi Ikeuchi, “Toward a Dancing Robot with Listening Capability: Keypose-Based Integration of Lower, Middle, and Upper Body Motions for Varying Music Tempos”, *Transactions on Robotics*, Vol.30, pp.771–778, 2014.06

International Conferences

1. Takahiro Okamoto, Takaaki Shiratori, Matthew Glisson, Katsu Yamane, Shunsuke Kudoh, and Katsushi Ikeuchi, “Extraction of Person-specific Motion Style based on a Task Model and Imitation by Humanoid Robot”, In *Proc. IEEE International Conference on Intelligent Robots and Systems* (IROS2014), 2014.9

2. Takahiro Okamoto, Takaaki Shiratori, Shunsuke Kudoh, and Katsushi Ikeuchi, “Temporal Scaling of Leg Motion for Music Feedback System of a Dancing Humanoid Robot”, In *Proc. IEEE International Conference on Intelligent Robots and Systems* (IROS2010), pp.2256–2263, 2010.10

3. Shunsuke Kudoh, Takahiro Okamoto, Takaaki Shiratori, Shinichiro Nakaoka, and Katsushi Ikeuchi, “Towards a Dancing-to-music Humanoid Robot: Temporal Scaling Model of Whole Body Motion for a Dancing Humanoid Robot”, In *Proc. IROS2010 workshop “Robots and Musical Expressions”*, pp.9–16, 2010.10

Domestic Conferences

1. 岡元崇紘, 白鳥貴亮, 工藤俊亮, 池内克史, “舞踊ロボットの音楽フィードバックシステムためのSQUATを含む脚動作時間伸縮”, 第28回日本ロボット学会学術講演会, 2010.09