# Extraction of Person-specific Motion Style based on a Task Model and Imitation by Humanoid Robot

T. Okamoto† T. Shiratori* M. Glisson⋆ K. Yamane⋆ S. Kudoh‡ K. Ikeuchi†

***Abstract*— In this paper, we present a humanoid robot which extracts and imitates the person-specific differences in motions, which we will call *style*. Synthesizing human-like and stylistic motion variations according to specific scenarios is becoming important for entertainment robots, and imitation of styles is one variation which makes robots more amiable. Our approach extends a learning from observation (LFO) paradigm which enables robots to understand what a human is doing and to extract reusable essences to be learned. The focus is on styles in the domain of LFO and the representation of them using the reusable essences. In this paper, we design an abstract model of a target motion defined in LFO, observe human demonstrations through the model, and formulate the representation of styles in the context of LFO. Then we introduce a framework of generating robot motions that reflect styles which are automatically extracted from human demonstrations. To verify our proposed method we applied it to a ring toss game, and generated robot motions for a physical humanoid robot. Styles from each of three random players were extracted automatically from their demonstrations, and used for generating robot motions. The robot imitates the styles of each player without exceeding the limitation of its physical constraints, while tossing the rings to the goal.**

## I. Introduction

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. To achieve such capability our group has introduced a learning from observation (LFO) paradigm [1] and developed an abstract model called *task model* in the concept. In contrast to a burst of reinforcement learning approach [2], this model 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 [1] [3] [4]. 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 [5]. These examples illustrate a potential of task models in a wide range of applications. Accumulation of such applications is one of our ultimate goals.

†Takahiro Okamoto and Katsushi Ikeuchi are with the University of Tokyo, Japan. {tokamoto, ki}@cvl.iis.u-tokyo.ac.jp

*Takaaki Shiratori is with Microsoft Research Asia, Microsoft Corporation, China. takaakis@microsoft.com

⋆Matthew Glisson and Katsu Yamane are with Disney Research, Pittsburgh, USA. {matt.glisson, kyamane}@disneyresearch.com

‡Shunsuke Kudoh is with the University of Electro-Communications, Japan. kudoh@is.uec.ac.jp

Fig. 1. An example of person-specific 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.

Synthesizing human-like and stylistic robot behavior [6][7] 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 [8][9]. Similarly we expect such technologies that generate expressive motions according to scenarios can make robots very human-like and amiable. In this paper we especially focus on styles in terms of person-specific differences in motions. As shown in Fig. 1, even when we perform a simple task such as tossing rings to the goal, details of the throwing motions vary according to individuals. We are interested in formulating this vaguely defined concept and imitating it using humanoid robots. We expect that such ability would expand the capability of entertainment robots.

In the robotics community synthesizing human-like motions from motion capture data has been investigated. To absorb kinematic differences, Pollard *et al* [10] modified joint-angle trajectories preserving the wave pattern of them within

the constraints. On the other hand, Nakaoka *et al* [5] abstracted dance motions based on task models. Deriving motions from a pre-segmented motion capture database [7] [11] also have been actively developed. To make robot motions look as much like original human motions as possible, optimization-based methods [12] [13] [14] 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 [15] extracts correlations between components of motion for an interactive editing tool of motion styles. Torresani *et al* [16] 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 [17], PCA [18], ICA [9], and DP matching [19]. Comparing styles in terms of mood/emotion-specific variations, few studies consider person-specific styles.

The proposed method in this paper allows a humanoid robot to extract person-specific styles from human demonstrations. Moreover, 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 [5] 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 focus on and extract styles that describe the tendencies of how to do each task from multiple demonstrations of one particular person.

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 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.

This paper is organized as follows: Sec. II gives the design of the style based on a task model. Sec. III describes the process to generate robot motions considering styles. Sec. IV reports the experiment with a physical humanoid robot for validation. Finally, Sec. V concludes this paper.

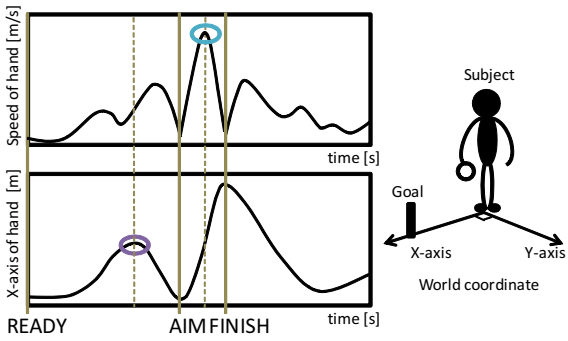


Fig. 2. Movements of the dominant hand in a sample motion of a player. Upper graph shows time-series data of hand speeds, and lower graph shows that of hand positions represented in the X-axis of the world coordinate. To define the world coordinate, the standing position of a human player is considered 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.

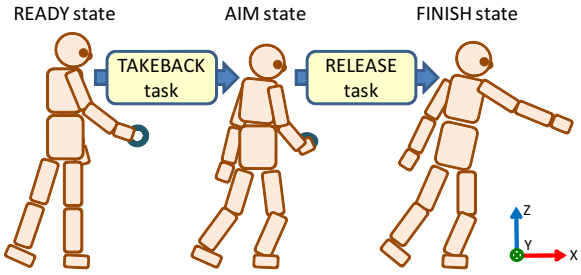


Fig. 3. Design of task sequence in the ring toss motion.

## II. Observation and Representation of Style in a Task Model

This section begins with the description of a task model that is used to represent motions in a ring toss game. Then, a method to extend the task model to represent styles is given.

### A. Task Model

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.

*1) Task:* 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. 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. 2). The upper graph suggests that the player stops the hand just anterior

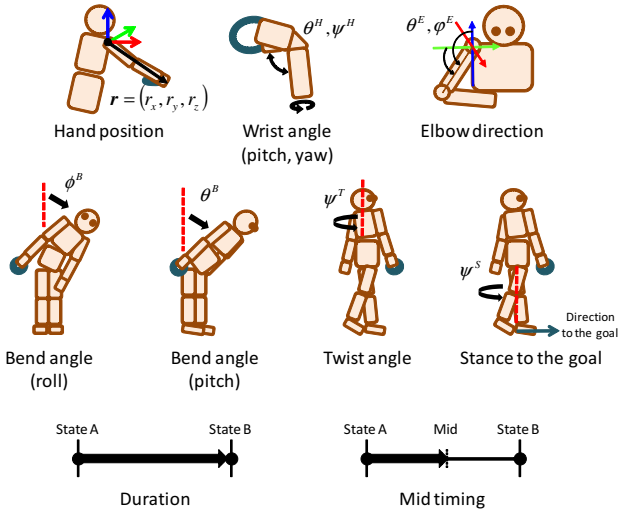


Fig. 4. Design of skill parameters for each task in the ring toss. These skill parameters characterize the trajectory of each task by giving details at end timing and a specific intermediate timing of each task.

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 state of the whole motion as READY state. This paper 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.

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. 3).

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.

Based on the definitions, all of the human demonstrations we captured could be automatically segmented. This supports the generality of our task representation in the ring toss game. There might be other alternative motion structures to toss the ring; however, we assume that the motions of human players from this observation covers all patterns of tossing.

*2) Skill parameter:* We chose skill parameters based on an observation of the ring toss motions as shown in Fig. 4. Skill parameters are designed 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. Every status during the task execution can be interpolated using a cubic spline. 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.

$\theta^H$, $\psi^H$ : 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.

$\theta^E$, $\varphi^E$ : 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.

$\phi^B$, $\theta^B$, $\psi^T$ : 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.

$\psi^S$ : 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. 2 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 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 $\boldsymbol{r}$, $\theta^H$, and $\psi^H$. Although our skill parameters might not be enough to represent the motions perfectly for each player, describing

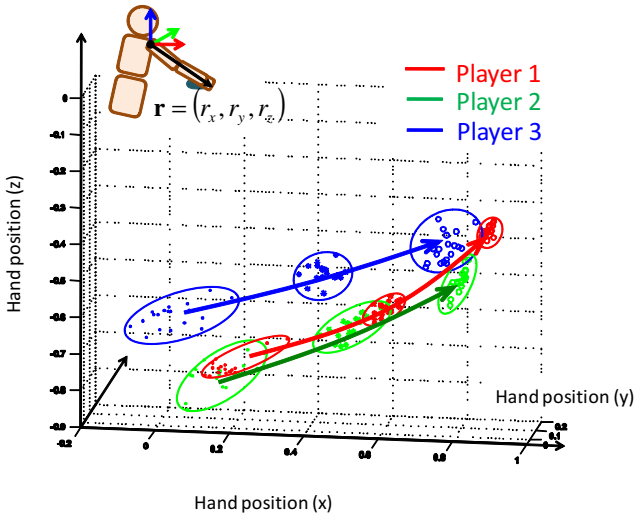


Fig. 5. Distributions of skill parameters $\boldsymbol{r}$ in RELEASE tasks in a total of 60 ring toss motions from three human players (20 motions per player). Note that $\boldsymbol{r}$ is defined in a Cartesian coordinate with the origin at the right shoulder. Clusters circled in the same color represent 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.

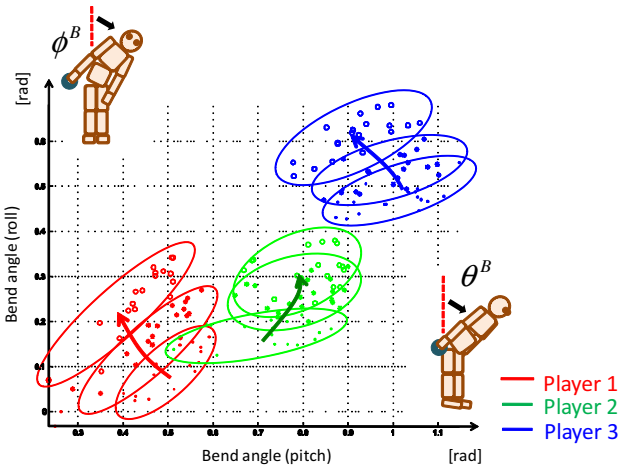


Fig. 6. Distributions of skill parameters $\phi^B$ and $\theta^B$ in RELEASE tasks in a total of 60 ring toss motions from three human players (20 motions per player). Clusters circled in the same color represent statistical distributions of hand positions $\boldsymbol{r}$ at start, a specific intermediate, 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.

motions perfectly using too many skill parameter is out of our focus. The purpose of task model is to abstract the target motion by will or choice of elements to be described. 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.

### *B. Style Parameter*

This subsection describes the representation of person-specific 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 representation to represent the individual differences in the context of the task model.

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. 5). 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 $\phi^B$ and $\theta^B$ in RELEASE tasks were plotted for each player in Fig. 6.

These figures show that the statistical distribution of each skill parameter varies from player to player. Fig. 5 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. 6 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.

As illustrated in the observation, distributions of skill parameters of a task vary from player to player. In fact, distributions describe the personal styles in the domain of the specific task. We use the distributions of skill parameters to represent person-specific style. To describe the distributions, we introduce style parameters in the concept of task model, one style parameter for each task. 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:

$$D = (\bar{\boldsymbol{s}}, \boldsymbol{\sigma}), \tag{1}$$

$$\bar{\boldsymbol{s}} = (\bar{s_1}, \bar{s_2}, \bar{s_3}, \cdots, \bar{s_k}, \cdots, \bar{s_N})^T, \tag{2}$$

$$\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \sigma_3, \cdots, \sigma_k, \cdots, \sigma_N)^T, \tag{3}$$

where $N$ is the number of skill parameters for the task, $\bar{\boldsymbol{s}}$ 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 $\sigma_k$ 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. $\bar{\boldsymbol{s}}$ 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. A set of skill parameters which consist of $\bar{\boldsymbol{s}}$ can be used to reconstruct a typical motion that reflects a personal style. However, it is difficult to use them directly to generate robot motion due to the physical constraints of the robot. In this case, the refinement of skill parameters according to the robot platform is required during the process of robot motion generation. The refinement in previous works [5] were based on ad-hoc rules, and could impair the features of style. $\boldsymbol{\sigma}$ suggest to us the importance of each skill parameter for the style. A proposed framework, to generate robot motion, is based on $\bar{\boldsymbol{s}}$ and $\boldsymbol{\sigma}$ effectively, and preserves the personal style in robot motion generation.

## III. Imitation of Person-Specific Motion Style based on Style Parameter

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.

### A. Style preservation in skill optimization

Initial entry of skill parameters is set to the $\bar{\boldsymbol{s}}$ 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:

$$E_{Style} + \lambda_1 E_{Ring} + \lambda_2 (E_{Angle} + E_{Velocity} + E_{Collision}), \quad (4)$$

$$E_{Style} = \sum_{i=1}^{N} \left(\frac{\bar{s_i} - s_i}{\sigma_i}\right)^2, \quad (5)$$

$E_{Style}$ 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.

$$E_{Ring} = |\boldsymbol{r}_{goal} - \boldsymbol{r}_{ring}(\boldsymbol{s})|^2, \quad (6)$$

$E_{Ring}$ 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.

$$E_{Angle} = \sum_{k=0}^{K} \sum_{j=0}^{N} \alpha_{j,k}^2(\boldsymbol{s}), \quad (7)$$

$$\alpha_{j,k}(\boldsymbol{s}) = \begin{cases} q_{j,k}(\boldsymbol{s}) - q_j^{max} & (q_{j,k}(\boldsymbol{s}) > q_j^{max}) \\ q_j^{min} - q_{j,k}(\boldsymbol{s}) & (q_{j,k}(\boldsymbol{s}) < q_j^{min}) \\ 0 & (otherwise) \end{cases}, \quad (8)$$

$E_{Angle}$ 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.

$$E_{Velocity} = \sum_{k=0}^{K} \sum_{j=0}^{N} \beta_{j,k}^2(\boldsymbol{s}), \quad (9)$$

$$\beta_{j,k}(\boldsymbol{s}) = \begin{cases} \dot{q}_{j,k}(\boldsymbol{s}) - \dot{q}_j^{max} & (\dot{q}_{j,k}(\boldsymbol{s}) > \dot{q}_j^{max}) \\ \dot{q}_j^{min} - \dot{q}_{j,k}(\boldsymbol{s}) & (\dot{q}_{j,k}(\boldsymbol{s}) < \dot{q}_j^{min}) \\ 0 & (otherwise) \end{cases}, \quad (10)$$

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

$$E_{Collision} = \sum_{k=0}^{K} \sum_{p=0}^{N} \gamma_{p,k}^2(\boldsymbol{s}), \tag{11}$$

$$\gamma_{p,k}(\boldsymbol{s}) = \begin{cases} r_{1,p} + r_{2,p} - d_p(\boldsymbol{s}) & (r_{1,p} + r_{2,p} > d_p(\boldsymbol{s})) \\ 0 & (otherwise) \end{cases}, \tag{12}$$

$E_{Collision}$ 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.

$E_{Ring}$, $E_{Angle}$, $E_{Velocity}$ 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 [20]. $\lambda_1$ and $\lambda_2$ are given empirically.

### B. Task reconstruction from skill parameters

This subsection describes the process to reconstruct the trajectory of a task from a set of skill parameters to generate the robot motion.

A trajectory of a task is reconstructed by interpolating the state transition in the task. Timings of each key state: 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 $\theta^H$, elbow direction $\theta^E$, twist and lean $\psi^T$ $\phi^B$ $\theta^B$, and stance $\psi^S$ at those key states 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 skill parameters interpolated in tasks. 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.

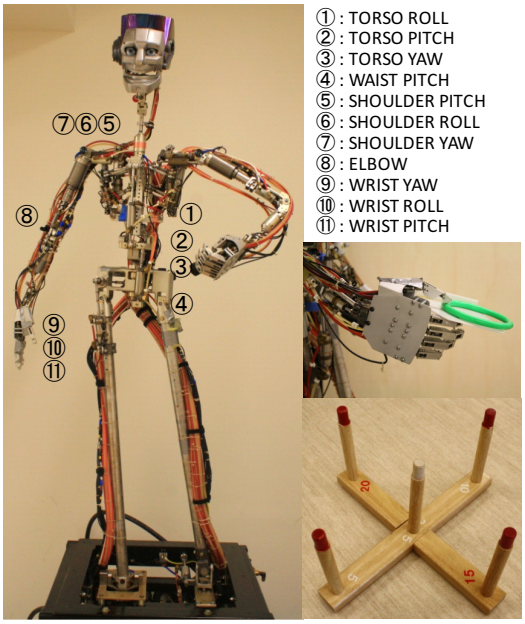


Fig. 7. A physical humanoid robot with its degree of freedoms, a hand plate for grasping, a ring on the plate, and a goal for the ring toss game.

## IV. Experiment

### A. Robot Platform

In this experiment, we employed a physical humanoid robot as shown in Figure 7. 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 7. The ring set on the plate is fixed and released by the right thumb.

### B. Experimental condition

Our system extracted style parameters for each of the three players (A, B, and C), from demonstrations captured at Sec. II-B, 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.

### C. Result

The robot imitated a total three types of styles in ring toss motion. The three types of ring toss are shown in Fig. 8, Fig. 9, and Fig. 10. 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. 8 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.

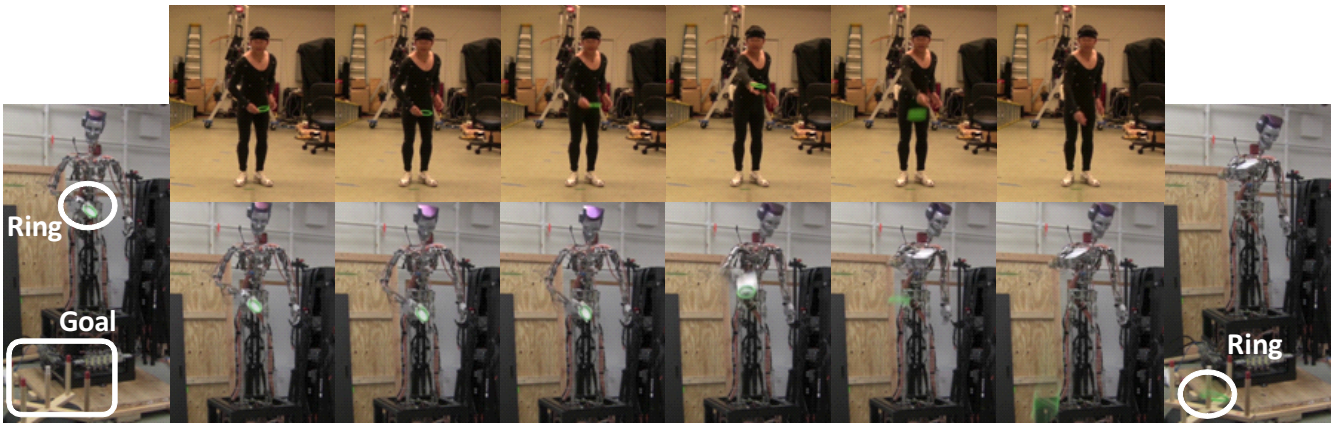


Fig. 8. Top row: the sequence of the player's demonstration. Bottom row: the sequence of robot motion mimicking that player's style. Player A tends to take the ring back slightly, and toss the ring in the front. His bending is relatively small compared to the others', and his hand position, especially at the end timing of RELEASE task, tends to be higher. These features are imitated by the robot as shown in the picture.


Fig. 9. 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.


Fig. 10. 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.

Player B shown in Fig. 9 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

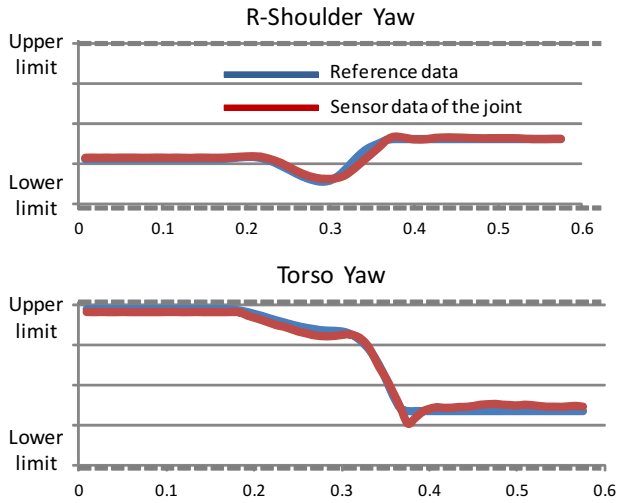


Fig. 11. Command trajectories of generated robot motions and the trajectories executed by robot: Gray lines show the upper and lower limits. Blue lines show the inputted command trajectories for R-shoulder yaw and torso yaw joints. Red lines show the executed trajectories captured via sensors equipped to each joint of the robot.

imitated by the robot. Player C shown in Fig. 10 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.

In these experiments self collisions did not occur and excess of joint limitations are avoided as shown in Fig. 11. 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.

## V. Conclusion

This paper presented a method to extract person-specific 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 detect features of each style in the robot motions.

For future work, we will 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.

## Acknowledgment

This work was supported by JSPS KAKENHI under Grant 23240026.

## References

[1] K. Ikeuchi and T. Suehiro, "Toward an assembly plan from observation. i. task recognition with polyhedral objects," *Trans. Robotics and Automation*, vol. 10, no. 3, pp. 368–385, 1994.

[2] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE trans. systems, man and cybernetics, Part B*, vol. 37, no. 2, pp. 286–298, 2007.

[3] S. B. Kang and K. Ikeuchi, "Toward automatic robot instruction from perception: Mapping human grasps to manipulator grasps," *Trans. Robotics and Automation*, vol. 13, no. 1, pp. 81–95, 1997.

[4] J. Takamatsu, T. Morita, K. Ogawara, H. Kimura, and K. Ikeuchi, "Representation for knot-tying tasks," *Trans. Robotics*, vol. 22, no. 1, pp. 65–78, 2006.

[5] S. Nakaoka, A. Nakazawa, F. Kanehiroand, K. Kaneko, M. Morisawa, H. Hirukawa, and K. Ikeuchi, "Learning from observation paradigm: Leg task models for enabling a biped humanoid robot to imitate human dances," *Int'l J. of Robot. Res.*, vol. 26, no. 8, pp. 829–844, 2007.

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

[7] K. Yamane, M. Revfi, and T. Asfour, "Synthesizing object receiving motions of humanoid robots with human motion database," *Proc. of ICRA-2013*, 2013.

[8] L. M. Tanco and A. Hilton, "Realistic synthesis of novel human movements from a database of motion capture examples," *Proc. of Workshop on HUMO-2000*, pp. 137 – 142, 2000.

[9] A. Shapiro, Y. Cao, and P. Faloutsos, "Style components," *Proc. of GI-2006*, pp. 33 – 39, 2006.

[10] N. S. Pollard, J. K. Hodgins, M. J. Riley, and C. Atkeson, "Adapting human motion for the control of a humanoid robot," *Proc. of ICRA-2002*, vol. 2, pp. 1390–1397, 2002.

[11] O. C. Jenkins and M. J. Mataric, "Deriving action and behavior primitives from human motion data," *Proc. of IROS-2002*, vol. 3, pp. 2551–2556, 2002.

[12] W. Suleiman, E. Yoshida, F. Kanehiro, J. P. Laumond, and A. Monin, "On human motion imitation by humanoid robot," *Proc. of ICRA-2008*, pp. 2697–2704, 2008.

[13] S. Nakaoka and T. Komura, "Interaction mesh based motion adaptation for biped humanoid robots," *Proc. of Humanoids-2012*, pp. 625–631, 2012.

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

[15] M. Neff and Y. Kim, "Interactive editing of motion style using drives and correlations," *Proc. of SCA-2009*, pp. 103–112, 2009.

[16] L. Torresani, P. Hackney, and C. Bregler, "Learning motion style synthesis from perceptual observations," *proc. of NIPS-2006*, pp. 1393–1400, 2006.

[17] M. Brand and A. Hertzmann, "Style machines," *Proc. of SIGGRAPH-2000*, pp. 183–192, 2000.

[18] R. Urtasun, P. Glardon, R. Boulic, D. Thalmann, and P. Fua, "Style-based motion synthesis," *Computer Graphics Forum*, vol. 23, no. 4, pp. 799–812, Dec. 2004.

[19] A. Nakazawa, S. Nakaoka, and K. Ikeuchi, "Matching and blending human motions using temporal scaleable dynamic programming," *Proc. of IROS-2004*, vol. 1, pp. 287–294, 2004.

[20] K. Madsen, H. B. Nielsen, and O. Tingleff, "Methods for non-linear least squares problems (2nd ed.)," p. 60, 2004.