



# Synthesis of kinematically constrained full-body motion from stochastic motion model

Wataru Takano<sup>1</sup> · Yoshihiko Nakamura<sup>1</sup>

Received: 28 January 2017 / Accepted: 21 January 2019 / Published online: 9 February 2019  
© The Author(s) 2019

## Abstract

Humanoid robots are expected not only to understand human behaviors but also to perform human-like actions in order to be integrated into our daily lives. Learning by imitation is a powerful framework that can allow robots to generate the same motions that humans do. However, generation of motions by robots that are precisely the same as learned motions is not typically helpful in real environments, which are likely to be different from the environment where the motions were learned. For example, a robot may learn to reach for a glass on a table, but this motion cannot be used as is to reach for a cup on a cupboard shelf because the location of the cup is different from the location of the glass. Objects manipulated by robots and humans can be located in a variety of places. The robots therefore have to synthesize motions that depend on the current environment in order to reach a target object. Adaptive motion synthesis from memorized motions is an essential technique for allowing robots to perform human-like motions and accomplish motion tasks. This paper proposes a novel approach to synthesize full body motion by using both motions encoded as Hidden Markov Models and kinematic task constraints. We design an objective function that evaluates similarity between synthesized and memorized motions, satisfaction of the kinematic constraints, and smoothness of the generated motion. We develop an algorithm to find a motion trajectory that maximizes this objective function. The experiments demonstrate the utility of the proposed framework for the synthesis of full body motions by robots.

**Keywords** Motion synthesis · Motion primitive · Stochastic model

## 1 Introduction

Humanoid robots are expected to understand gesture and language, communicate with humans, and be integrated into our daily lives. Recently, approaches for acquisition by robots of motions and their relevant language have been proposed. These approaches are based on mimesis theory, suggested by Donald, and support the hypothesis that humans communicated with each other by using gestures before they acquired language and that gestures underlie symbolic and linguistic mechanisms (Donald 1991).

Representations of motion primitives can be grouped according to two approaches. One approach reduces motions into a set of parameters of a dynamical system, such as systems of differential equations or neural networks (Okada et al. 2002; Ijspeert et al. 2003; Kadone and Nakamura 2005; Ito et al. 2006). Another approach is based on stochastic models, as typified by Hidden Markov Models (HMMs) (Inamura et al. 2004; Asfour et al. 2006; Billard et al. 2006; Kulic et al. 2008). In both approaches, motions are encoded as parameters in dynamical systems or stochastic models. Robots can memorize motions as stochastic models and subsequently reuse the models for the motion recognition and synthesis. In particular, efficient algorithms for stochastic motion models have been developed to compute the likelihood that a stochastic model generates a specific data sequence and to optimize the parameters such that they can maximize the likelihood that the model generates the sequences present in training data. These algorithms can be used to recognize observed motions and to learn motion primitives. Stochastic approaches have the additional advantage that they can be

---

This research was supported by a Grant-in-Aid for Scientific Research (A) (No. 17H00766) from the Japan Society for the Promotion of Science.

---

✉ Wataru Takano  
takano@ynl.t.u-tokyo.ac.jp  
Yoshihiko Nakamura  
nakamura@ynl.t.u-tokyo.ac.jp

<sup>1</sup> The University of Tokyo, Bunkyo Hongo, Tokyo, Japan

combined with natural language models. In the field of natural language processing, many natural language models have been developed by using stochastic models based on the same graphical structures as the motion models. The combination of the motion models and the natural language makes it possible for robots to understand motions as a form of language (Takano et al. 2007; Takano and Nakamura 2008; Takano et al. 2018).

The stochastic motion model is efficiently used for the motion recognition. Robots is expected only to recognize observed motion but also to synthesize human-like motions during interacting with human partner. Motion synthesis from the stochastic motion model is a key technology to improve the human–robot interaction. Several approaches to synthesizing motions from HMMs according to the distribution of training motions has been developed (Inamura et al. 2004; Calinon et al. 2010). Because the synthesized motions are likely to be the average of the training motions, they cannot be used to adaptively synthesize motions in environments different from where the motions were originally learned. For example, if a robot learns a motion for grasping an object on a table, then the robot can reach to an object located at the same position by using the motions as learned. However, the robot cannot reach to an object on a shelf without modifying the learned motion since the object is located at a different position on the shelf than the learned position on the table. The adaptation of synthesized motion to the environment by using demonstrated motions is a required ability for robots to perform flexible actions for manipulation motions.

This paper proposes a novel approach to synthesizing full body motion that satisfies kinematic constraints from HMMs of motion models for humanoid robots. The proposed approach introduces an objective function consisting of three terms: a likelihood that an HMM generates the motion, distances between the synthesized motion and kinematic constraints, and continuity of the synthesized motion. Computation for motion synthesis can be formulated by optimization of the motion such that the motion maximizes the objective function. Thus, the optimization results in motion that not only resembles the learned motion but also satisfies kinematic task constraints. This approach makes it possible for robots both to recognize human motions and to synthesize human-like motions adaptive to the environment only using a stochastic motion model.

## 2 Related work

Several techniques have been developed to synthesize motions from motion models (Inamura et al. 2004; Herzog et al. 2008; Sugiura and Iwahashi 2008; Kunori et al. 2009; Calinon et al. 2010). Inamura et al. have proposed a novel

method to synthesize motions from HMMs. This method consists of two steps. The first step samples a node according to probabilities of node transitions in the HMM. The second step samples a vector of joint angles according to distribution of joint angles on the sampled node. The iteration of these two steps results in a motion trajectory that is represented by a sequence of the joint angles (Inamura et al. 2004). Calinon et al. (2010) have extended HMMs to joint angles and joint angular velocities, using a method that calculates the distribution of the joint angular velocities given the joint angles at the previous frame, samples the joint angular velocities based on the distribution, and updates the joint angles at the current frame. This framework can establish a dynamical system from demonstrations of motion in the system. Although these types approaches can reproduce motion trajectories, they cannot be used to adaptively synthesize motions in current environments.

Herzog et al. (2008) used a Parametric Hidden Markov Model (PHMM) to recognize and synthesize object manipulation motions. PHMMs allow representing typical motions and compensating for different motions of a particular type. A set of motions is encoded into an HMM as a representative motion, and each motion variant is encoded into its own HMM. The parameters of each motion variant's HMM are represented as a linear function of the parameters of the representative motion's HMM and the position of the object. Thus, the individual motions can be parameterized by the position of the object. The performance of PHMM critically depends on the number of demonstrations of manipulation motions. It is difficult to apply the synthesized motion to situations that are markedly different from those in the demonstration. Calinon et al. (2013) proposes an approach to synthesize adaptive motions by using descriptors of both the motion and the manipulated object. The complexity of the descriptor is critical for systems with high degrees of freedom, such as humanoid robots. Paraschos et al. developed Probabilistic Movement Primitives (ProMP) for representing distributions over motions. A feedback controller can be stochastically derived that synthesizes motion reaching the target positions while maintaining the given motion distribution. In Paraschos et al. (2013), the distribution of parameters is characterized by a single Gaussian distribution across the whole motion trajectory. This specialization is necessary to allow efficient searching of the configuration space for useful parameters. However, the specialization restricts the variations of motions that can be synthesized. Additionally, the performance of the ProMP on motion recognition has not been investigated.

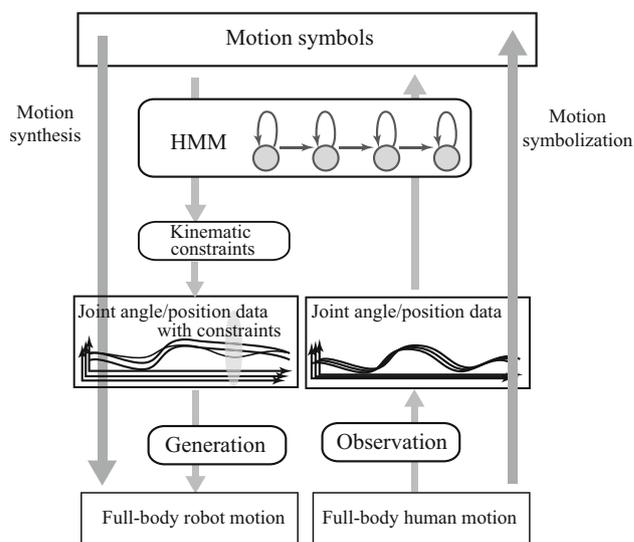
A dynamical system with states represented by nonlinear differential equations is widely used for encoding a training motion trajectory and subsequently synthesizing robot motion adapted to different configurations. Okada et al. defined the dynamics as a vector field. In that framework,

a motion pattern is expressed by a closed curve in the vector field, and they proposed a design for such vector fields that would attract a robot’s motion to a reference motion (Okada et al. 2002). This attractor field will stabilize the robot’s motion even when the robot is perturbed by the external environment. This framework can be used to synthesize a closed motion trajectory, but it does not aim at modifying training motions to allow reaching target configurations at specified times. Additionally, it is difficult to use this framework to recognize observed motions. The Dynamic Movement Primitives (DMPs) framework is a popular approach for learning a single trajectory by using a spring–damping model that includes a forcing term used to attract a synthesized trajectory to a goal configuration (Ijsspeert et al. 2001). The single-trajectory nature of the framework results in limited scalability and makes it difficult to apply the framework in real environments, where robots are required to perform various motions and must adapt to them. Matsubara et al. (2011) extended the DMPs framework to allow learning multiple motions and demonstrated the highly scalable nature of the modified framework by trials on a dual-arm robot. Prada et al. (2013) also specialized the DMPs framework, developing a robot that can pass an object to a human partner. Their approach enables handling of a moving goal. In these frameworks, only a single goal is given for the final phase. Although the resultant trajectory reaches this goal, it is difficult to modify the motion across the whole trajectory. More specifically, the setting of multiple sequential goals (i.e., via-points) is not supported. This restriction leads to limited adaptivity. Additionally, the DMPs framework has difficulty with recognizing human demonstrations. Noda et al. (2014) developed a framework for representation of sensory–motor data, using deep neural networks for this purpose. In that framework, a robot learns a sequence of multimodal data from vision, motor, and audio sensors. An auto-encoder model then extracts low-dimensional features from the sensory images, and these image features and motion data are integrated by another auto-encoder model. Neural networks trained in this way enable a humanoid robot to perform whole body motion that is responsive to sensory perceptions. However, the synthesized motion is not guaranteed to reach target positions.

The contribution of this paper is the presentation of a motion synthesis that uses motion models of HMMs to allow creating a motion trajectory that will reach target positions while maintaining a profile of training motions. Additionally, the motion models of the HMMs can be efficiently used for motion recognition. This capability means that robots can not only recognize observed motions but also synthesize human-like motions from the stochastic motion models alone, without the need for additional motion representations.

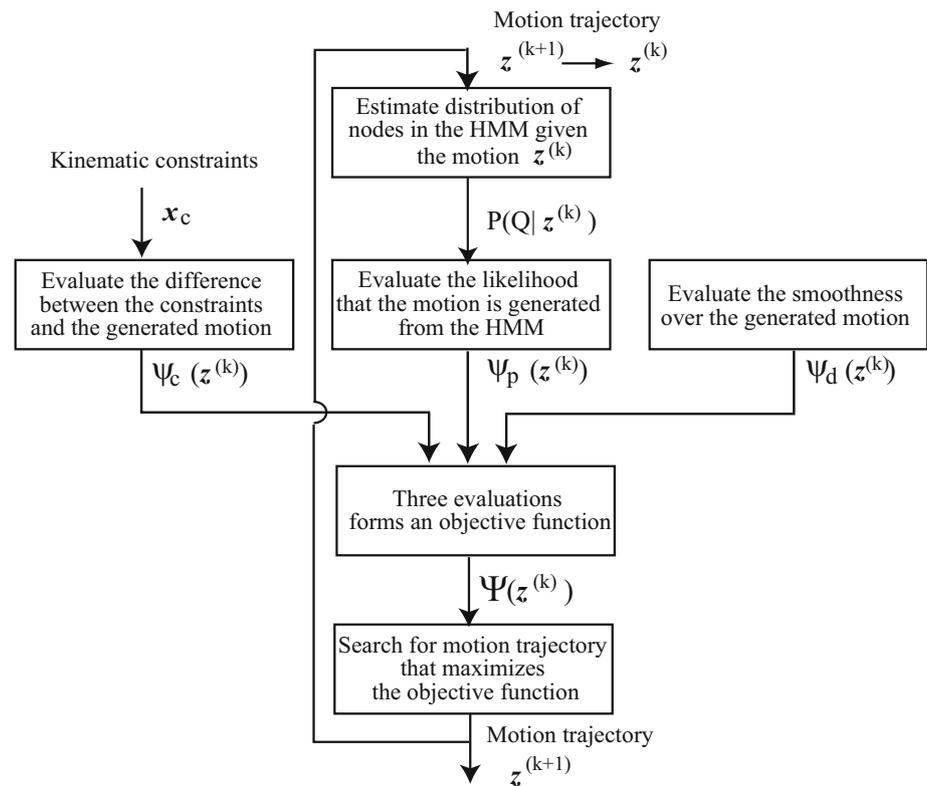
### 3 Synthesis of motions with constraints from motion models

A full-body motion of a human or a humanoid robot is represented by a vector of states,  $\mathbf{x}_t$  at time  $t$ , such as joint angles or positions. A motion segment is expressed by a motion trajectory  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ . The motion segment is encoded into a set of parameters of an HMM, and these sets of parameters are hereinafter referred to as a “motion model”. The HMM is a stochastic model and can be defined in a compact notation  $\lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$ , where  $\mathbf{A} = \{a_{ij}\}$  is the matrix whose entries  $a_{ij}$  are the probability of transitioning from the  $i$ th node to the  $j$ th node,  $\mathbf{B}$  is the set of output probability density functions at the nodes, and  $\boldsymbol{\pi} = \{\pi_1, \pi_2, \dots, \pi_n\}$  is the vector whose entries  $\pi_i$  are probability of starting at the  $i$ th node. Encoding of the motion segment optimizes these parameters of an HMM such that the likelihood that the HMM generates the motion segment is maximized. This optimization is achieved using the Baum–Welch algorithm, which is an expectation maximization (EM) algorithm (Rabiner and Juang 1993). HMMs can be used for motion recognition. Motion recognition finds the HMM from among a set of HMMs that is the most likely to generate the observed motion. This means that the observation is recognized as a motion model corresponding to an HMM. HMMs can be also used to generate motion trajectories. Thus, a humanoid robot memorizes motion segments in the form of motion models through observations and demonstrations, recognizes observed sequences, and generates its own motions by using the motion models. This section describes a novel approach to synthesizing kinemat-



**Fig. 1** Motion segments represented by a sequence of joint angles are encoded into HMMs. Such encodings are referred to as “motion models” in this context. The HMMs make it possible for a humanoid robot to encode full-body human motions and to synthesize human-like full-body motions

**Fig. 2** Motion synthesis searches for a motion trajectory through iterative estimation and maximization. Estimates give the distribution of node sequences corresponding to the motion trajectory. Maximization optimizes the motion trajectory with respect to the objective function, which depends on the estimated distribution of node sequences



ically constrained motion from a motion model in a way that allows humanoid robots to make human-like motions and achieve tasks dependent on the environment, as shown in Fig. 1.

Figure 2 shows a flow diagram of the synthesizing motion from motion models. This framework introduces an objective function that integrates three functions. The first function calculates the likelihood that a synthesized motion is generated by the HMM corresponding to a particular motion model. The second function calculates the error between the synthesized motion and kinematic constraints, which are manually given or determined by the environment. The third function calculates the smoothness of the synthesized motion. The large first function leads to a synthesized motion more similar to training motion, the small second function derives a motion reaching to target configurations, and the small third function derives a motion without abrupt acceleration. The motion synthesis computation searches for the motion trajectory that maximizes the objective function.

The log likelihood that a motion trajectory  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$  is generated by the motion model  $\lambda$  can be written as

$$\psi_p = \ln P(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n | \lambda). \quad (1)$$

This likelihood can be efficiently computed by a forward algorithm (Rabiner 1989). The first term in the objective function can be written as (1).

We assume that  $\delta_t = 1$  if kinematic constraints are given to the motion at time  $t$  and that  $\delta_t = 0$  otherwise. The second term in the objective function is derived from the difference between the state  $\mathbf{x}_t$  and the constraint  $\mathbf{x}_{t,c}$  at time  $t$ . The second term can be written as a weighted quadratic form of the differences as follows:

$$\psi_c = -\frac{w_c}{2} \sum_{t=1}^n \delta_t (\mathbf{x}_t - \mathbf{x}_{t,c})^T (\mathbf{x}_t - \mathbf{x}_{t,c}), \quad (2)$$

where the positive weight parameter  $w_c$  is manually specified. Note that the kinematic constraints are limited to the state of joint angles or positions, and does not handle their velocities or the body structure of the humanoid robot in this paper.

Ideally, smooth motion should be synthesized for a humanoid robot. The smoothness can be measured by joint-angular or positional jerks (Flash and Hogan 1985). The third term is also written in a weighted quadratic form as

$$\psi_d = -\frac{w_d}{2} \sum_{t=4}^n \ddot{\mathbf{x}}_t^T \ddot{\mathbf{x}}_t \quad (3)$$

$$\ddot{\mathbf{x}}_t = \mathbf{x}_t - 3\mathbf{x}_{t-1} + 3\mathbf{x}_{t-2} - \mathbf{x}_{t-3},$$

where the positive weight parameter  $w_d$  also is manually specified.

Hence, these three terms yield the following objective function for the motion synthesis.

$$\Psi = \psi_p + \psi_c + \psi_d. \tag{4}$$

Three terms in this objective function are in different units, but the weight parameters,  $w_c$  and  $w_d$ , include the unit conversion.

The motion trajectory  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$  is rewritten as the following vector:

$$\mathbf{z} = [\mathbf{x}_1^T, \dots, \mathbf{x}_n^T]^T, \tag{5}$$

where column vectors of the state  $\mathbf{x}_t$  are concatenated to form one vector  $\mathbf{z}$ , and  $^T$  specifies the transpose operation. By using the expression of motion trajectory  $\mathbf{z}$ , the objective function in (4) can be rewritten as the following:

$$\Psi(\mathbf{z}) = \ln P(\mathbf{z}|\lambda) - \frac{w_c}{2} (\mathbf{z} - \mathbf{z}_c)^T \mathbf{A}_c (\mathbf{z} - \mathbf{z}_c) - \frac{w_d}{2} \mathbf{z}^T \mathbf{L} \mathbf{z}, \tag{6}$$

where  $\mathbf{A}_c$  is a matrix with binary elements taking value 1 for constraints that are present and 0 for constraints that are not. The vector  $\mathbf{z}_c$  represents the kinematic constraints  $\mathbf{x}_{t,c}$  at time  $t$  aligned in the column direction,

$$\mathbf{A}_c = \begin{bmatrix} \delta_1 \mathbf{I}_d & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \delta_n \mathbf{I}_d \end{bmatrix}, \tag{7}$$

$$\mathbf{z}_c = [\mathbf{x}_{1,c}^T, \dots, \mathbf{x}_{n,c}^T]^T. \tag{8}$$

Here,  $\mathbf{I}_d$  and  $\mathbf{O}_d$  are the  $d \times d$  identity matrix and the zero matrix respectively, where  $d$  is the number of dimensions of state  $\mathbf{x}_t$ . The matrix  $\mathbf{L}$  is written as the  $dn \times dn$  Laplacian matrix.

$$\mathbf{L} = \begin{bmatrix} & & & & \mathbf{O}_{3d,dn} \\ \mathbf{A}_{d,4d} & \mathbf{O}_d & \dots & & \\ \mathbf{O}_d & \mathbf{A}_{d,4d} & \mathbf{O}_d & \dots & \\ \mathbf{O}_d & \mathbf{O}_d & \mathbf{A}_{d,4d} & \mathbf{O}_d & \dots \\ & & & \ddots & \\ \dots & \mathbf{O}_d & \mathbf{O}_d & \mathbf{A}_{d,4d} & \end{bmatrix}, \tag{9}$$

$$\mathbf{A}_{d,4d} = [-\mathbf{I}_d \ 3\mathbf{I}_d \ -3\mathbf{I}_d \ \mathbf{I}_d]. \tag{10}$$

Motion trajectory  $\mathbf{z}^{(k)}$  is derived from iterative application of an estimation process and a maximization process. The estimation process estimates the distribution  $P(q_t = i|\mathbf{z}^{(k)}, \lambda)$  of the motion at time  $t$  in the motion trajectory  $\mathbf{z}^{(k)}$  as generated from the  $i$ th node in HMM  $\lambda$ , where the motion trajectory  $\mathbf{z}^{(k)}$  is derived during the  $k$ th maximization process.

The maximization process optimizes the motion trajectory such that it maximizes the objective function  $\Psi$ , which is based on the estimated node distribution  $P(q_t = i|\mathbf{z}^{(k)}, \lambda)$ . The maximization process results in a motion trajectory represented as (11).

$$\mathbf{z}^{(k+1)} = \left( \sum_{i=1}^s \mathbf{V}_i \tilde{\boldsymbol{\mu}}_i + w_c \mathbf{A}_c \mathbf{z}_c \right) \times \left( \sum_{i=1}^s \mathbf{V}_i + w_d \mathbf{L} + w_c \mathbf{A}_c \right)^{-1}. \tag{11}$$

Here,  $\mathbf{V}_i$  is the  $d \times n$  matrix defined by using the node distribution  $P(q_t = i|\mathbf{z}^{(k)}, \lambda)$  and the covariance matrix  $\boldsymbol{\Sigma}_i$  at the  $i$ th node in the HMM  $\lambda$ .

$$\mathbf{V}_i = \begin{bmatrix} \boldsymbol{\Sigma}_i^{-1} P(q_1 = i|\mathbf{z}^{(k)}, \lambda) & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \boldsymbol{\Sigma}_i^{-1} P(q_T = i|\mathbf{z}^{(k)}, \lambda) \end{bmatrix}.$$

Additionally,  $\tilde{\boldsymbol{\mu}}_i$  is the  $dn$ -dimensional vector consisting of  $n$  copies of the mean vector  $\boldsymbol{\mu}_i$  at the  $i$ th node of the HMM.

$$\tilde{\boldsymbol{\mu}}_i = [\boldsymbol{\mu}_i^T, \dots, \boldsymbol{\mu}_i^T]^T. \tag{12}$$

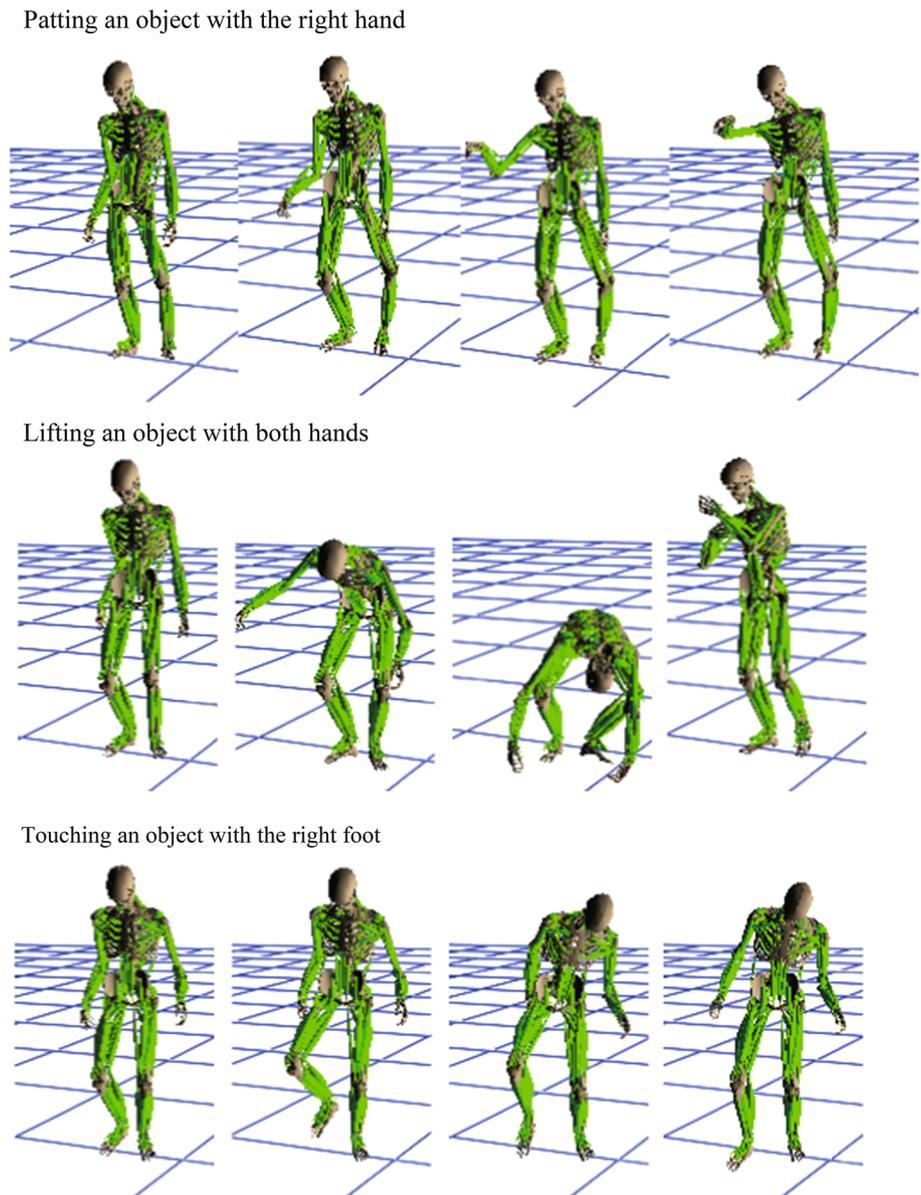
A full derivation of the motion trajectory  $\mathbf{z}^{(k+1)}$  is given in the ‘‘Appendix’’.

The derived motion trajectory does not necessarily avoid the self collision. When a self collision with two specified link is detected, the joint angles relevant to their configurations at the collision are modified such that the distance between the two links becomes larger, and these joint angles are added to the constraints in (2). The motion trajectory to avoid the self collision can be subsequently derived.

### 4 Experiments

An experiment on the practical synthesis of full-body motion was conducted by using our proposed framework, which generates a motion trajectory satisfying kinematic constraints from a motion model. We measured human full-body motions by using an optical motion capture system. The capture system acquired positions of 34 markers attached to a performer. These measured position data were converted into joint angles of a humanoid robot, HRP4, by inverse kinematics. Sequences of joint angles were obtained. We recorded three different motions for each action: ‘‘patting an object with the right hand’’, ‘‘lifting an object with both hands’’, and ‘‘touching an object with the right foot’’. Only one trial from measuring the motion of a single individual is used for training each HMM. These actions are shown in Fig. 3. In

**Fig. 3** Positions of markers attached to a performer were recorded by a motion-capture system. The measured positions were converted into joint angles of a humanoid robot. Human full-body motions—“patting an object with the right hand”, “lifting an object with both hands”, and “touching an object with the right foot”—were recorded. These motion data were encoded into HMMs, each of which represents a motion model



this figure, the captured human full-body motion data were transformed to human character motions. Each sequence of joint angles was encoded into a 30-node left-to-right HMM ( $N = 30$ ). In this way, three motion models represented by the HMMs were acquired from the captured data.

In this experiment, kinematic constraints were given for the final postures of these three motions. Kinematic constraints to reach up with the right hand for an object at a high position and to reach down with the right hand for an object at a low position were added to the motion of “patting an object with the right hand”. The kinematic constraints to lift with both hands to a high or low position were added to the motion of “lifting an object with both hands”. The motion of “touching an object with the right foot” was given the added

kinematic constraints of touching an object near the feet with a foot and touching an object on the ground ahead with a foot.

Figure 4 shows the full-body motions synthesized from the motion models under the given kinematic constraints. In the figures on the top, a humanoid robot reaches out its right hand toward a high position and toward a low position by modifying the original motion trajectory. In the middle figure sequence, the humanoid robot lifts an object to over its head and to chest height with both hands. In the figures on the bottom, the humanoid robot uses the right foot to touch an object near the feet and to touch an object on the ground ahead. This motion synthesis does not deal with the robot dynamics. More specifically the synthesized motion is not stable and a controller to stabilize the posture of the robot is additionally needed.

**Fig. 4** Motions optimized by taking into account both the memorized motion pattern and task constraints. The humanoid robot performs motions—patting a high object, patting a low object, lifting an object to a high position, lifting an object to a low position, touching an object near the feet with a foot, and touching an object on the ground ahead with a foot—in a way that satisfies the given task constraints

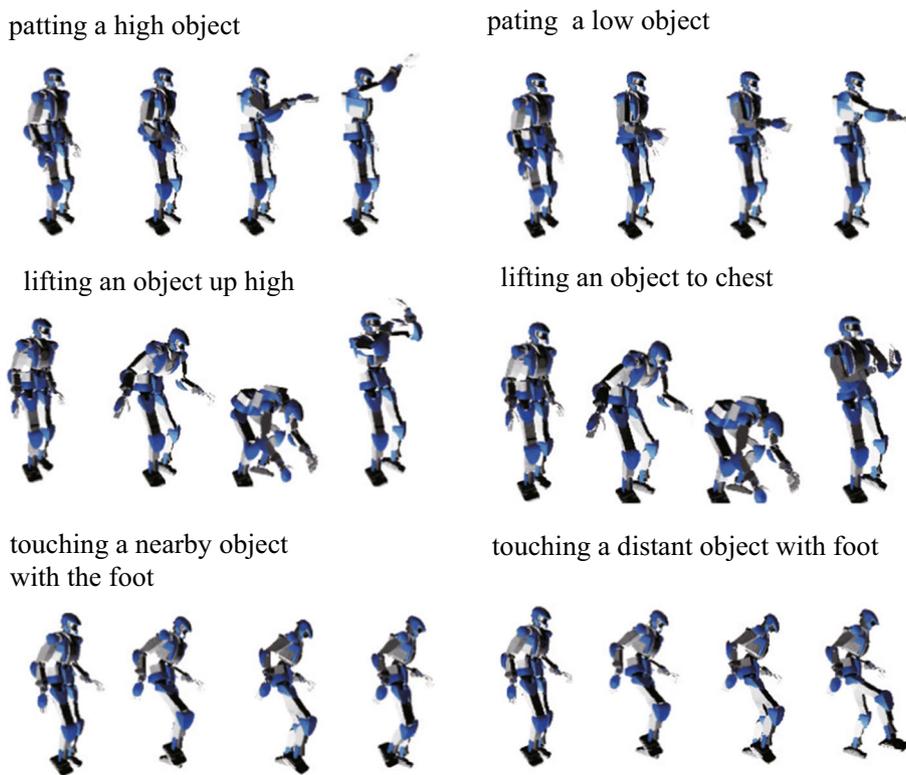
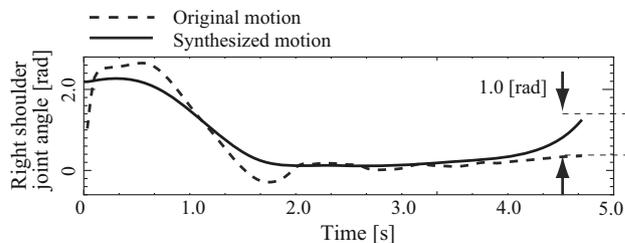


Figure 5 compares the original motion trajectory of “patting an object with the right hand” with the synthesized motion trajectory that was given the first kinematic constraint. The kinematic constraint set the joint angle of the robot’s right shoulder to become 1.0 rad larger than that of the final posture in the original motion. The solid line and dashed line represent trajectories of the right shoulder joint angle in the synthesized motion and the original motion, respectively. The profiles of the trajectories are similar from 0 to 4.0 s. The synthesized motion maintains the characteristic change in the original motion that was embedded in the motion model. The motion is synthesized such that the joint angle of the right shoulder can achieve the kinematic constraint of 1.4 rad from 4.0 to 5.0 s. At the final frame, the joint angle of the right shoulder becomes 1.0 rad larger than that of the original motion. The trajectory in Fig. 5 demonstrates that the humanoid robot can perform an action of “patting an object with the right hand” not only by maintaining the profile of the original motion but also by partially modifying the motion such that it can reach out its right hand toward a high position. This algorithm for full-body motion synthesis makes it possible for a humanoid robot to behave or manipulate objects in a way that adapts to the environment, even if the target objects are differently located than in the training phase.

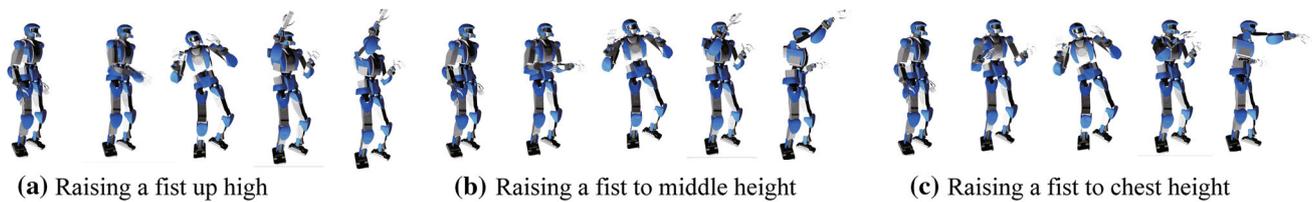
We tested our proposed framework on another motion, “raising the right fist”. Kinematic constraints were given to this motion such that the humanoid robot can raise its right



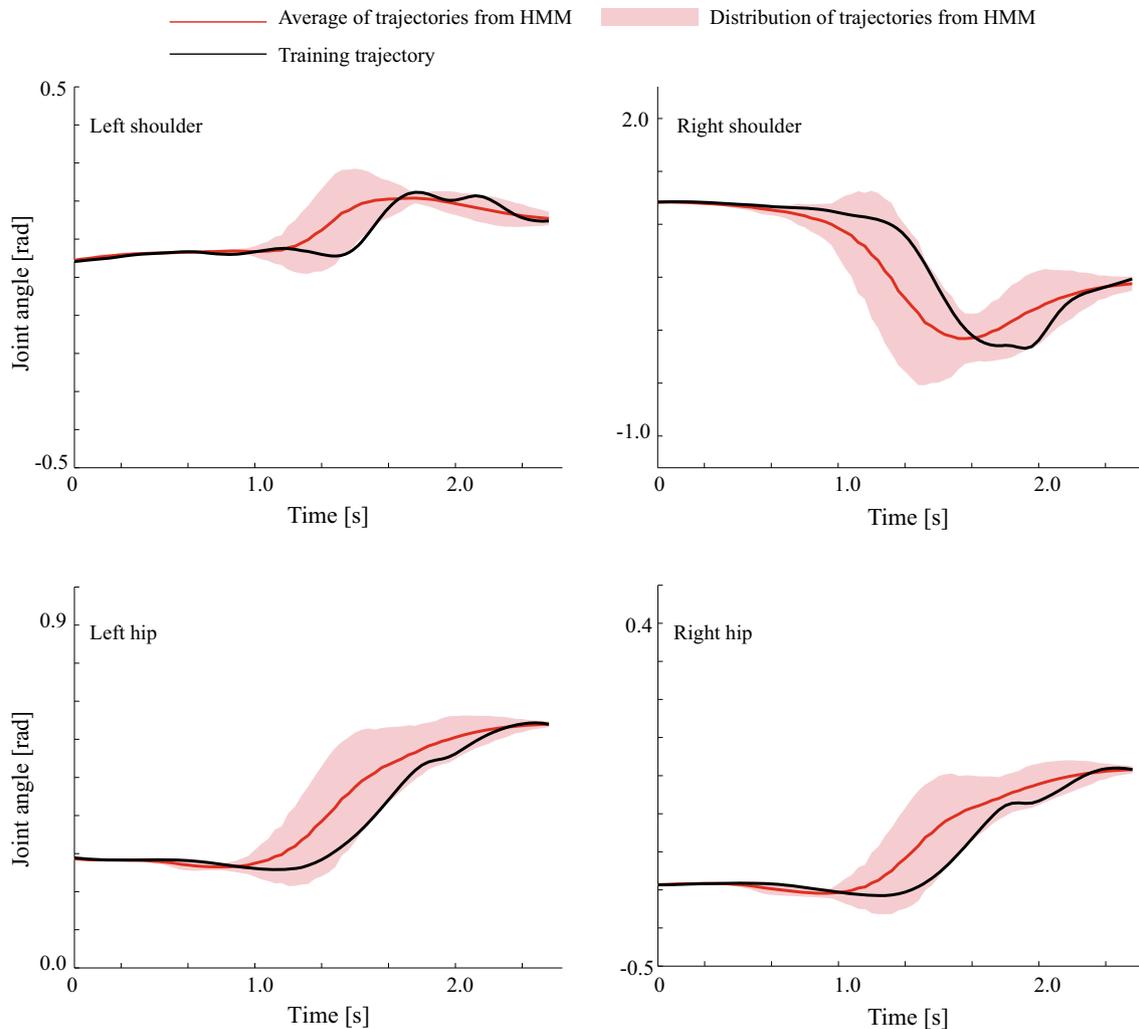
**Fig. 5** Profile of joint angles of a right shoulder of a trained motion and a generated motion for “patting” motion pattern

fist to three levels. The first kinematic constraint set the joint angle of the robot’s right shoulder to become 1.0 rad larger than that of the final posture in the original motion; the second kinematic constraint set the joint angle of robot’s right shoulder to become the same as that of the final posture in the original motion; and the third kinematic constraint set the joint angle of robot’s right shoulder to 0.0 rad (i.e., upper arm pointing down) at the final posture. Figure 6 shows the synthesized full-body motions. In the figures on the left, the humanoid robot is raising its right fist vertically. In the middle figures, the robot is raising its right fist to a high position, and in the figures on the right the robot is raising its right fist horizontally.

Figure 7 shows the profile of the joint angles of left shoulder, right shoulder left hip and right hip in the original motion data of “raising up the right fist”. In the figure, the black solid



**Fig. 6** Three kinematic constraints were given to the motion of “raising the right fist”. In the left panels, a humanoid robot raises its right fist vertically. In the middle panels, the robot raises its right fist to a high position, and in the right panels, the robot raises its right fist horizontally

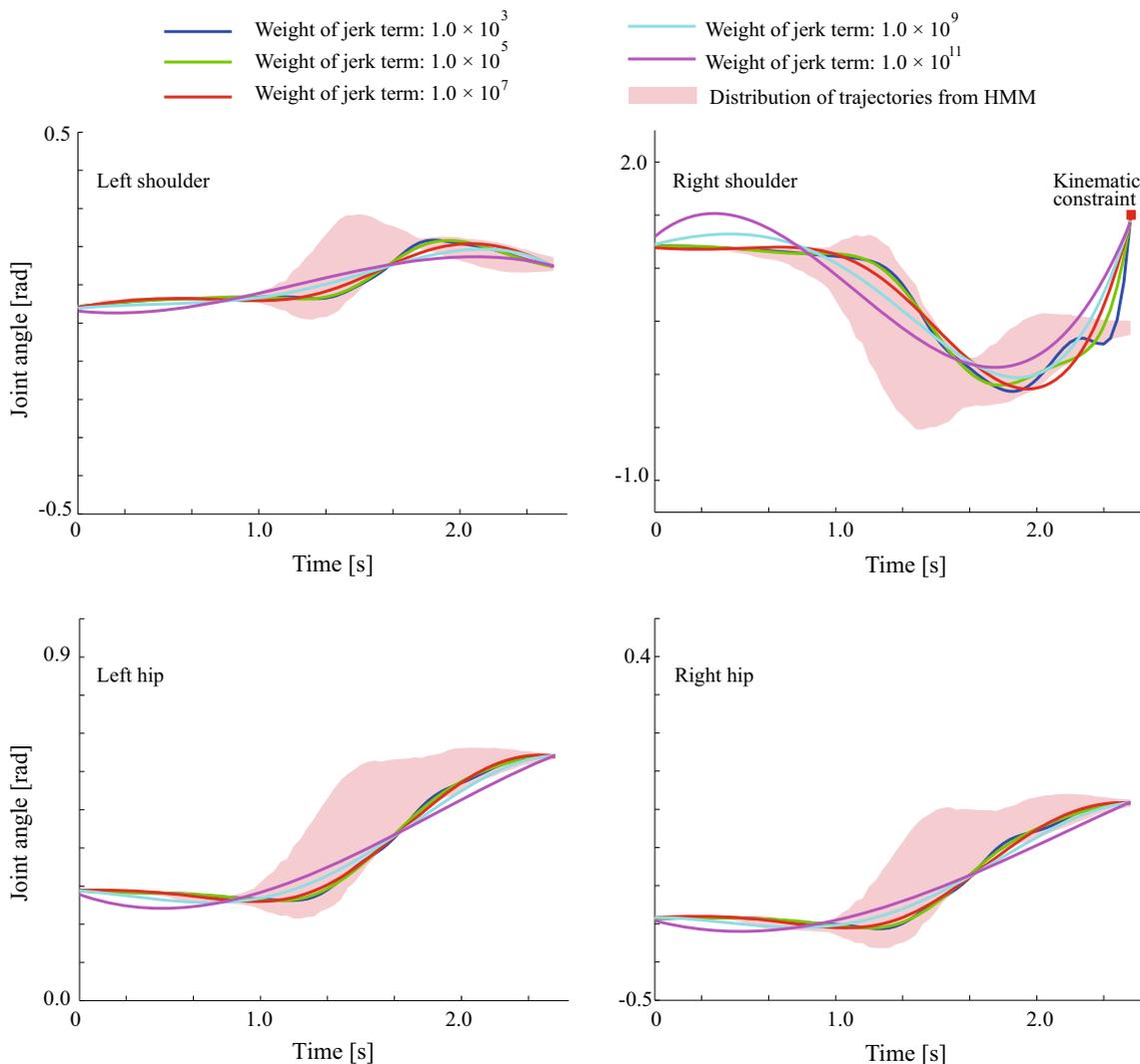


**Fig. 7** Profiles of the joint angles of left shoulder, right shoulder, left hip, and right hip in the captured motion. Black solid lines are training data trajectories. The red solid lines show the averages of the generated motion trajectories. The red-shaded graphs show motion trajectory distributions of the HMM, which the captured motion is encoded into.

These distributions are formed by motion trajectories generated by the HMM. The red solid lines show the averages of the generated motion trajectories (Color figure online)

line represents the training motion trajectory. 100 motion trajectories were generated from the motion model by a Monte Carlo method (Inamura et al. 2004), and the red solid line is the average motion trajectory of them. The red-shaded graph represents the motion trajectory distribution from 100 generated motion trajectories and the average motion trajectory.

The kinematic constraints are not given to this motion synthesis. We synthesized several full-body variants of the motion “raising the right fist to middle height” by varying the positive weight parameter  $w_d$ . Parameter  $w_d$  was set to  $1.0 \times 10^3$ ,  $1.0 \times 10^5$ ,  $1.0 \times 10^7$ ,  $1.0 \times 10^9$ , and  $1.0 \times 10^{11}$  while positive weight parameter  $w_c$  was fixed to the constant value  $1.0 \times 10^3$ .



**Fig. 8** Trajectories of the joint angles of left shoulder, right shoulder, left hip, and right hip in the synthesized motion. Blue, green, red, light blue, and purple solid lines represents the motion trajectories derived by

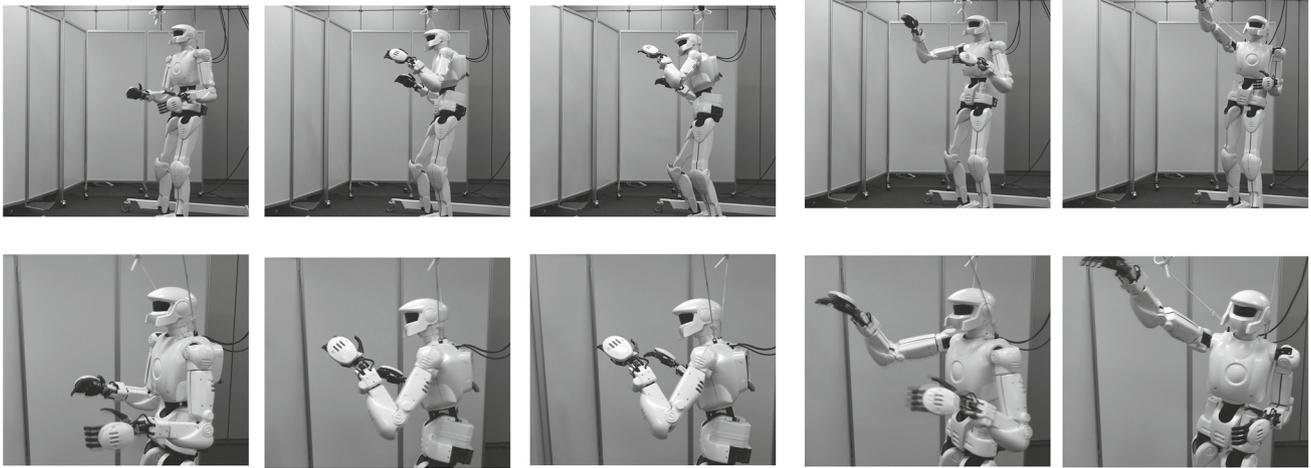
setting  $w_d$  to  $1.0 \times 10^3$ ,  $1.0 \times 10^5$ ,  $1.0 \times 10^7$ ,  $1.0 \times 10^9$ , and  $1.0 \times 10^{11}$ , respectively (Color figure online)

Figure 8 shows the synthesized motion trajectories with the kinematic constraints. All the synthesized motion trajectories are similar to the training motion trajectory and maintain the profile of the motion trajectory distribution of the motion model from 0.0 to 2.0 s. The synthesized motion trajectories come close to the kinematically constrained posture at the final frame. The synthesized motions are also smooth without abrupt acceleration. Additionally, we measured the computational time needed to synthesize the motion “raising up the right fist,” with computation performed by a computer with an Intel Xeon 2.6 GHz CPU with 128 GB of memory and setting the number of maximization processes to five. Computation took 7.3 s to synthesize the motion, which had  $n = 79$  frames. The objective function takes values of  $-102.5$ ,  $-89.3$ , and  $-92.0$  without the maximization pro-

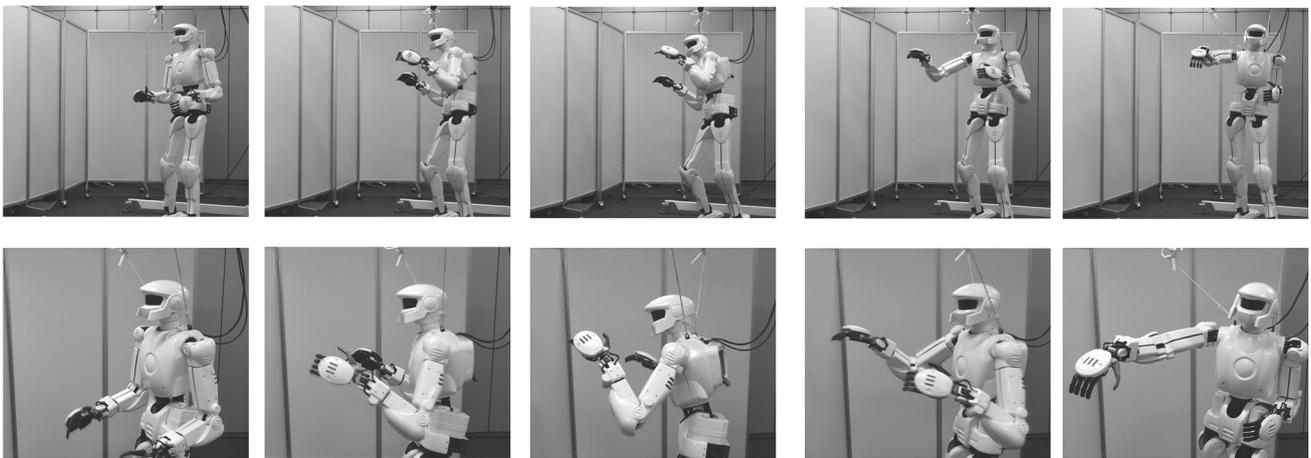
cess, and reached values of 86.2,  $-3.8$ , and  $-8.6$  after the five maximization processes for the motions of “raising the fist up high”, “raising the fist to middle height,” and “raising the fist to chest height,” respectively.

We tested the synthesized motions on the full-size humanoid robot HRP-4. All joint angles of the full body are controlled according to the synthesized motions of “raising the right fist to middle height” and “raising the right fist to chest height.” Because of limits in joint mobility, the motion “raising the fist up high” could be performed by the real humanoid robot. Figure 9 shows the two motions being performed by the humanoid robot. In the photographs on the top, the robot raises its right fist to middle height, and in the photographs on the bottom, the robot raises its right fist to chest height. The joint angle of the right shoulder in the

## Raise a fist to middle height



## Raise a fist to chest height



**Fig. 9** Experiments of motion synthesis using humanoid robot HRP-4. The upper photographs show the motion of raising the right fist to middle height, and the lower photographs show the motion of raising the right fist to chest height

final posture of the motion “raising the fist to middle height” was 0.49 rad, which differs from the given constraints by 0.03 rad. The joint angle of the right shoulder in the motion “raising the fist to chest height” was 0.11 rad in the final posture. The results of this experiment on a real humanoid robot demonstrate that our proposed method can synthesize full-body human-like motions.

In the experiments described above, the kinematic constraints were given manually. This is not practical for real-world applications, where robots will be expected to discover the appropriate constraints on motions and the environments. More specifically: the robots need the ability to extract key features between the motions and objects being manipulated. An HMM that captures not only the sequence of angles of all the joint in the full body but also the sequence of positions of the object in the robot’s coordinate system may be helpful for extracting features of constraints. Additionally,

our proposed framework should be extended to account for the torques of all joints in humanoid robot’s full body, such that the robot can perform human-like motions and maintain the desired points of contact with the environment so that the robot can avoid falling down or suffering contact damage.

## 5 Conclusion

A summary of this research is as follows.

1. We presented a novel approach to synthesizing human-like motion with kinematic constraints for a humanoid robot. Human full-body motions are encoded into HMMs, which can be referred to as motion models. The objective function is designed by using three functions. The first function represents the likelihood that the motion trajec-

tory is generated from the motion model. The second function calculates the distance between the synthesized motion trajectory and kinematic constraints. The third function calculates the total jerk along the synthesized motion, so that minimizing this produces smooth motion.

2. We proposed an algorithm to find the motion trajectory that maximizes the objective function. The algorithm alternates between two processes. One is estimating the distribution of node sequences in the HMM given a motion trajectory that is derived in the other process. The other process finds a motion trajectory that maximizes the objective function given the estimated distribution of the node sequences. Iteration of these two processes results in the motion synthesis.
3. We tested our proposed approach to synthesizing full-body motion in a humanoid robot by using the motion models. We recorded the human full-body motions of “patting an object with the right hand,” “lifting an object with both hands,” “touching an object with the right foot,” and “raising the right fist.” Each recorded motion data was encoded into its corresponding HMM. Kinematic constraints were manually added to the motions. Synthesized motion demonstrated that the humanoid robot can not only perform human-like motions but also satisfy the kinematic constraints. The experiments validated our approach to synthesizing full-body motions in a humanoid robot.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## Appendix

The difference between the objective functions of  $\mathbf{z}^{(k+1)}$  and  $\mathbf{z}^{(k)}$  is represented by

$$\begin{aligned} \Psi(\mathbf{z}^{(k+1)}) - \Psi(\mathbf{z}^{(k)}) &= \ln P(\mathbf{z}^{(k+1)}|\lambda) - \ln P(\mathbf{z}^{(k)}|\lambda) \\ &\quad - \frac{w_c}{2} (\mathbf{z}^{(k+1)} - \mathbf{z}_c)^T \Delta_c (\mathbf{z}^{(k+1)} - \mathbf{z}_c) \\ &\quad + \frac{w_c}{2} (\mathbf{z}^{(k)} - \mathbf{z}_c)^T \Delta_c (\mathbf{z}^{(k)} - \mathbf{z}_c) \\ &\quad - \frac{w_d}{2} \mathbf{z}^{(k+1)T} \mathbf{Lz}^{(k+1)} + \frac{w_d}{2} \mathbf{z}^{(k)T} \mathbf{Lz}^{(k)}. \end{aligned} \tag{13}$$

The first and second terms in (13), which represents the likelihood that the motion trajectory  $\mathbf{z}$  is generated by HMM  $\lambda$ ,

can be rewritten as the following:

$$\ln P(\mathbf{z}|\lambda) = \ln \sum_{\forall \mathbf{q}} P(\mathbf{z}, \mathbf{q}|\lambda) \tag{14}$$

$$\begin{aligned} &= \ln \sum_{\forall \mathbf{q}} P(\mathbf{q}|\mathbf{z}) \frac{P(\mathbf{z}, \mathbf{q}|\lambda)}{P(\mathbf{q}|\mathbf{z})} \\ &= \ln E_{P(\mathbf{q}|\mathbf{z})} \left[ \frac{P(\mathbf{z}, \mathbf{q}|\lambda)}{P(\mathbf{q}|\mathbf{z})} \right], \end{aligned} \tag{15}$$

where  $\mathbf{q} = [q_1, q_2, \dots, q_n]^T$  is a sequence of nodes corresponding to the motion trajectory  $\mathbf{z} = [\mathbf{x}_1^T, \dots, \mathbf{x}_n^T]^T$ , where  $q_t$  is the node at time  $t$  in HMM  $\lambda$ . This means that node  $q_t$  generates the state vector  $\mathbf{x}_t$ . The value  $E_{P(\mathbf{q}|\mathbf{z})} [R]$  denotes the expected value of  $R$  given  $P(\mathbf{q}|\mathbf{z})$ , which is the distribution of the sequence of nodes  $\mathbf{q}$  for the motion trajectory  $\mathbf{z}$ . According to Jensen’s inequality, (15) implies that the following inequality holds.

$$\ln E_{P(\mathbf{q}|\mathbf{z})} \left[ \frac{P(\mathbf{z}, \mathbf{q}|\lambda)}{P(\mathbf{q}|\mathbf{z})} \right] \geq E_{P(\mathbf{q}|\mathbf{z})} \left[ \ln \frac{P(\mathbf{z}, \mathbf{q}|\lambda)}{P(\mathbf{q}|\mathbf{z})} \right]. \tag{16}$$

(14) and (16) yield the following relation:

$$\begin{aligned} \ln P(\mathbf{z}|\lambda) - E_{P(\mathbf{q}|\mathbf{z})} \left[ \ln \frac{P(\mathbf{z}, \mathbf{q}|\lambda)}{P(\mathbf{q}|\mathbf{z})} \right] &= \ln P(\mathbf{z}|\lambda) - E_{P(\mathbf{q}|\mathbf{z})} \left[ \ln P(\mathbf{z}|\lambda) - \ln \frac{P(\mathbf{q}|\mathbf{z}, \lambda)}{P(\mathbf{q}|\mathbf{z})} \right] \\ &= E_{P(\mathbf{q}|\mathbf{z})} \left[ \ln \frac{P(\mathbf{q}|\mathbf{z})}{P(\mathbf{q}|\mathbf{z}, \lambda)} \right] \\ &= KL(P(\mathbf{q}|\mathbf{z}) || P(\mathbf{q}|\mathbf{z}, \lambda)). \end{aligned} \tag{17}$$

(17) measures the difference between two distributions,  $P(\mathbf{q}|\mathbf{z})$  and  $P(\mathbf{q}|\mathbf{z}, \lambda)$ , and represents the Kullback–Leibler information. The Kullback–Leibler information becomes zero when these two distributions are the same and is positive otherwise. (13) can be rewritten as the following equation by using (17):

$$\begin{aligned} \Psi(\mathbf{z}^{(k+1)}) - \Psi(\mathbf{z}^{(k)}) &= E_{P(\mathbf{q}|\mathbf{z}^{(k+1)})} \left[ \ln \frac{P(\mathbf{z}^{(k+1)}, \mathbf{q}|\lambda)}{P(\mathbf{q}|\mathbf{z}^{(k+1)})} \right] \\ &\quad + KL(P(\mathbf{q}|\mathbf{z}^{(k+1)}) || P(\mathbf{q}|\mathbf{z}^{(k+1)}, \lambda)) \\ &\quad - E_{P(\mathbf{q}|\mathbf{z}^{(k)})} \left[ \ln \frac{P(\mathbf{z}^{(k)}, \mathbf{q}|\lambda)}{P(\mathbf{q}|\mathbf{z}^{(k)})} \right] \\ &\quad - KL(P(\mathbf{q}|\mathbf{z}^{(k)}) || P(\mathbf{q}|\mathbf{z}^{(k)}, \lambda)) \\ &\quad - \frac{w_c}{2} (\mathbf{z}^{(k+1)} - \mathbf{z}_c)^T \Delta_c (\mathbf{z}^{(k+1)} - \mathbf{z}_c) \\ &\quad + \frac{w_c}{2} (\mathbf{z}^{(k)} - \mathbf{z}_c)^T \Delta_c (\mathbf{z}^{(k)} - \mathbf{z}_c) \end{aligned}$$

$$-\frac{w_d}{2} \mathbf{z}^{(k+1)T} \mathbf{Lz}^{(k+1)} + \frac{w_d}{2} \mathbf{z}^{(k)T} \mathbf{Lz}^{(k)}. \tag{18}$$

We approximate the distributions of the sequence of node  $\mathbf{q}$  under the motion trajectory using HMM  $\lambda$ . In other words, distributions  $P(\mathbf{q}|\mathbf{z}^{k+1})$  and  $P(\mathbf{q}|\mathbf{z}^k)$  are set to  $P(\mathbf{q}|\mathbf{z}^k, \lambda)$ .

$$\begin{aligned} P(\mathbf{q}|\mathbf{z}^{k+1}) &= P(\mathbf{q}|\mathbf{z}^k, \lambda), \\ P(\mathbf{q}|\mathbf{z}^k) &= P(\mathbf{q}|\mathbf{z}^k, \lambda). \end{aligned} \tag{19}$$

These estimations lead to two relations, that in which the Kullback–Leibler information of the second term in (18) becomes greater than zero, and that in which the Kullback–Leibler information of the fourth term becomes zero. Therefore, if the new motion trajectory  $\mathbf{z}^{(k+1)}$  can be found at the  $k + 1$ th iteration computation such that (18) is positive except for the second and fourth terms, then the motion trajectory  $\mathbf{z}^{(k+1)}$  increases the value on the objective function  $\Psi(\mathbf{z}^{(k+1)})$  because the  $\Psi(\mathbf{z}^{(k+1)})$  is greater than  $\Psi(\mathbf{z}^{(k)})$ . This iteration results in a gradual increase in the value on the objective function. Thus, synthesis of motion trajectory  $\mathbf{z}$  with kinematical constraints from motion symbol  $\lambda$  can be performed by finding the motion trajectory iteratively such that

$$\begin{aligned} &\arg \max_{\mathbf{z}^{(k+1)}} \left\{ E_{P(\mathbf{q}|\mathbf{z}^{(k)}, \lambda)} \left[ \ln \frac{P(\mathbf{z}^{(k+1)}, \mathbf{q}|\lambda)}{P(\mathbf{q}|\mathbf{z}^{(k)}, \lambda)} \right] \right. \\ &\quad - E_{P(\mathbf{q}|\mathbf{z}^{(k)}, \lambda)} \left[ \ln \frac{P(\mathbf{z}^{(k)}, \mathbf{q}|\lambda)}{P(\mathbf{q}|\mathbf{z}^{(k)}, \lambda)} \right] \\ &\quad - \frac{w_c}{2} (\mathbf{z}^{(k+1)} - \mathbf{z}_c)^T \mathbf{A}_c (\mathbf{z}^{(k+1)} - \mathbf{z}_c) \\ &\quad + \frac{w_c}{2} (\mathbf{z}^{(k)} - \mathbf{z}_c)^T \mathbf{A}_c (\mathbf{z}^{(k)} - \mathbf{z}_c) \\ &\quad \left. - \frac{w_d}{2} \mathbf{z}^{(k+1)T} \mathbf{Lz}^{(k+1)} + \frac{w_d}{2} \mathbf{z}^{(k)T} \mathbf{Lz}^{(k)} \right\} \tag{20} \\ &= \arg \max_{\mathbf{z}^{(k+1)}} \left\{ E_{P(\mathbf{q}|\mathbf{z}^{(k)}, \lambda)} \left[ \ln P(\mathbf{z}^{(k+1)}|\mathbf{q}, \lambda) P(\mathbf{q}|\lambda) \right] \right. \\ &\quad - \frac{w_c}{2} (\mathbf{z}^{(k+1)} - \mathbf{z}_c)^T \mathbf{A}_c (\mathbf{z}^{(k+1)} - \mathbf{z}_c) \\ &\quad \left. - \frac{w_d}{2} \mathbf{z}^{(k+1)T} \mathbf{Lz}^{(k+1)} \right\} \\ &= \arg \max_{\mathbf{z}^{(k+1)}} \left\{ \sum_{\forall \mathbf{q}} P(\mathbf{q}|\mathbf{z}^{(k)}, \lambda) \ln P(\mathbf{z}^{(k+1)}|\mathbf{q}, \lambda) \right. \\ &\quad - \frac{w_c}{2} (\mathbf{z}^{(k+1)} - \mathbf{z}_c)^T \mathbf{A}_c (\mathbf{z}^{(k+1)} - \mathbf{z}_c) \\ &\quad \left. - \frac{w_d}{2} \mathbf{z}^{(k+1)T} \mathbf{Lz}^{(k+1)} \right\}. \end{aligned} \tag{21}$$

The likelihood  $P(\mathbf{z}^{(k+1)}|\mathbf{q}, \lambda)$  that motion trajectory  $\mathbf{z}^{(k+1)}$  is generated along the sequence of nodes  $\mathbf{q}$  from HMM  $\lambda$  can be expressed in a closed form by using parameters of the

HMM.

$$\begin{aligned} P(\mathbf{z}^{(k+1)}|\mathbf{q}, \lambda) &= \ln \pi_{q_1} + \sum_{t=1}^{n-1} \ln a_{q_t, q_{t+1}} \\ &\quad + \sum_{t=1}^n \ln b_{q_t}(\mathbf{x}_t^{(k+1)}). \end{aligned} \tag{22}$$

Here,  $\pi_i$  is the initial node probability of the  $i$ th node,  $a_{i,j}$  is the probability of transitioning from the  $i$ th node to the  $j$ th node, and  $b_i(\mathbf{x})$  is the output probability that the  $i$ th node generates the state vector  $\mathbf{x}$ . By substituting (22) into (21), (21) can be rewritten as the following:

$$\begin{aligned} &\arg \max_{\mathbf{z}^{(k+1)}} \left\{ \sum_{\forall \mathbf{q}} P(\mathbf{q}|\mathbf{z}^{(k)}, \lambda) \right. \\ &\quad \times \left\{ \ln \pi_{q_1} + \sum_{t=1}^{n-1} \ln a_{q_t, q_{t+1}} + \sum_{t=1}^n \ln b_{q_t}(\mathbf{x}_t^{(k+1)}) \right\} \\ &\quad - \frac{w_c}{2} (\mathbf{z}^{(k+1)} - \mathbf{z}_c)^T \mathbf{A}_c (\mathbf{z}^{(k+1)} - \mathbf{z}_c) \\ &\quad \left. - \frac{w_d}{2} \mathbf{z}^{(k+1)T} \mathbf{Lz}^{(k+1)} \right\} \\ &= \arg \max_{\mathbf{z}^{(k+1)}} \left\{ \sum_{\forall \mathbf{q}} P(\mathbf{q}|\mathbf{z}^{(k)}, \lambda) \sum_{t=1}^n \ln b_{q_t}(\mathbf{x}_t^{(k+1)}) \right. \\ &\quad - \frac{w_c}{2} (\mathbf{z}^{(k+1)} - \mathbf{z}_c)^T \mathbf{A}_c (\mathbf{z}^{(k+1)} - \mathbf{z}_c) \\ &\quad \left. - \frac{w_d}{2} \mathbf{z}^{(k+1)T} \mathbf{Lz}^{(k+1)} \right\} \\ &= \arg \max_{\mathbf{z}^{(k+1)}} \left\{ \sum_{i=1}^s \sum_{t=1}^T \ln b_i(\mathbf{x}_t^{(k+1)}) P(q_t = i|\mathbf{z}^{(k)}, \lambda) \right. \\ &\quad - \frac{w_c}{2} (\mathbf{z}^{(k+1)} - \mathbf{z}_c)^T \mathbf{A}_c (\mathbf{z}^{(k+1)} - \mathbf{z}_c) \\ &\quad \left. - \frac{w_d}{2} \mathbf{z}^{(k+1)T} \mathbf{Lz}^{(k+1)} \right\}, \end{aligned} \tag{23}$$

where  $P(q_t = i|\mathbf{z}^{(k)}, \lambda)$  is the distribution with staying at the  $i$ th node at time  $t$  given the motion trajectory  $\mathbf{z}^{(k)}$ . The output probability distribution  $b_i(\mathbf{x})$  is assumed to follow the Gaussian distribution:

$$\begin{aligned} \ln b_i(\mathbf{x}) &= -\frac{1}{2} \left[ \ln(2\pi)^d |\Sigma_i| \right. \\ &\quad \left. + (\mathbf{x} - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) \right], \end{aligned} \tag{24}$$

where  $\boldsymbol{\mu}_i, \Sigma_i$  are the  $d$ -dimensional mean vector and the  $d \times d$  covariance matrix at the  $i$ th node in the HMM, respectively. By substituting (24) into (23) and eliminating terms that are

independent of the motion trajectory  $z^{(k+1)}$  in (23), (23) can be reduced to (25).

$$\begin{aligned} & \arg \max_{z^{(k+1)}} \left\{ - \sum_{i=1}^s \sum_{t=1}^n \frac{1}{2} \left( x_t^{(k+1)} - \mu_i \right)^T \Sigma_i^{-1} \left( x_t^{(k+1)} - \mu_i \right) \right. \\ & \quad \times P(q_t = i | z^{(k)}, \lambda) \\ & \quad - \frac{w_c}{2} \left( z^{(k+1)} - z_c \right)^T \Delta_c \left( z^{(k+1)} - z_c \right) \\ & \quad \left. - \frac{w_d}{2} z^{(k+1)T} L z^{(k+1)} \right\} \\ & = \arg \max_{z^{(k+1)}} \left\{ - \frac{1}{2} \sum_{i=1}^s \left[ z^{(k+1)} - \tilde{\mu}_i \right]^T V_i \left[ z^{(k+1)} - \tilde{\mu}_i \right] \right. \\ & \quad - \frac{w_c}{2} \left( z^{(k+1)} - z_c \right)^T \Delta_c \left( z^{(k+1)} - z_c \right) \\ & \quad \left. - \frac{w_d}{2} z^{(k+1)T} L z^{(k+1)} \right\}. \end{aligned} \quad (25)$$

Therefore, the motion trajectory  $z^{(k+1)}$  satisfying (25) can be derived as (11).

## References

- Asfour, T., Gyarfas, F., Azad, P., & Dillmann, R. (2006). Imitation learning of dual-arm manipulation task in humanoid robots. In *Proceedings of the IEEE-RAS international conference on humanoid robots* (pp. 40–47).
- Billard, A., Calinon, S., & Guenter, F. (2006). Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics and Autonomous Systems*, 54, 370–384.
- Calinon, S., Alisadeh, T., & Caldwell, D. G. (2013). On improving the extrapolation capability of task-parameterized movement models. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 610–617).
- Calinon, S., D'halluin, F., Sauser, E. L., Caldwell, D. G., & Billard, A. (2010). Learning and reproduction of gestures by imitation. *IEEE Robotics Automation Magazine*, 17(2), 44–54.
- Donald, M. (1991). *Origin of the modern mind*. Cambridge: Harvard University Press.
- Flash, T., & Hogan, N. (1985). The coordination of arm movement: An experimentally confirmed mathematical model. *Neuroscience*, 5, 1688–1703.
- Herzog, D., Krueger, V., & Grest, D. (2008). Parametric hidden markov models for recognition and synthesis of movements. In *Proceedings of the British Machine Vision Conference* (pp. 163–172).
- Ijsspeert, A. J., Nakanishi, J., Shibata, T., & Schaal, S. (2001). Nonlinear dynamical systems for imitation with humanoid robots. In *Proceedings of the IEEE-RAS international conference on humanoid robots*.
- Ijsspeert, A. J., Nakanishi, J., & Schaal, S. (2003). Learning control policies for movement imitation and movement recognition. *Neural Information Processing System*, 15, 1547–1554.
- Inamura, T., Toshima, I., Tanie, H., & Nakamura, Y. (2004). Embodied symbol emergence based on mimesis theory. *International Journal of Robotics Research*, 23(4), 363–377.
- Ito, M., Noda, K., Hoshino, Y., & Tani, J. (2006). Dynamic and interactive generation of object handing behaviors by a small humanoid robot using a dynamic neural network model. *Neural Networks*, 19(3), 323–337.
- Kadone, H., & Nakamura, Y. (2005). Symbolic memory for humanoid robots using hierarchical bifurcations of attractors in nonmonotonic neural networks. In *Proceedings of the 2005 IEEE/RSJ international conference on intelligent robots and systems* (pp. 2900–2905).
- Kulic, D., Takano, W., & Nakamura, Y. (2008). Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. *International journal of Robotics Research*, 27(7), 761–784.
- Kunori, H., Lee, D., & Nakamura, Y. (2009). Associating and reshaping of whole body motions for object manipulation. In *Proceedings of the 2008 IEEE/RSJ international conference on intelligent robots and systems* (pp. 5240–5247).
- Matsubara, T., Hyon, S. H., & Morimoto, J. (2011). Learning parametric dynamic movement primitives from multiple demonstrations. *Neural Networks*, 24(5), 493–500.
- Noda, K., Arie, H., Suga, Y., & Ogata, T. (2014). Multimodal integration learning of robot behavior using deep neural networks. *Robotics and Autonomous Systems*, 62, 721–736.
- Okada, M., Tatani, K., & Nakamura, Y. (2002). Polynomial design of the nonlinear dynamics for the brain-like information processing of whole body motion. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 1410–1415).
- Paraschos, A., Daniel, C., Peters, J., & Neumann, G. (2013). Probabilistic movement primitives. In *Advances in neural information processing systems* (pp. 2616–2624).
- Prada, M., Remazeilles, A., Koene, A., & Endo, S. (2013). Dynamic movement primitives for human-robot interaction: Comparison with human behavioral observation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013* (pp. 1168–1175).
- Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77, 257–286.
- Rabiner, L., & Juang, B. H. (1993). *Fundamentals of speech recognition*. Prentice Hall Signal Processing Series. Englewood Cliffs, NJ: Prentice Hall.
- Sugiura, K., & Iwahashi, N. (2008). Motion recognition and generation by combining reference-point-dependent probabilistic models. In *Proceedings of the 2008 IEEE/RSJ international conference on intelligent robots and systems* (pp. 852–857).
- Takano, W., & Nakamura, Y. (2008). Integrating whole body motion primitives and natural language for humanoid robots. In *Proceedings of the IEEE-RAS international conference on humanoid robots* (pp. 708–713).
- Takano, W., Yamada, Y., & Nakamura, Y. (2018). Linking human motions and objects to language for synthesizing action sentences. *Autonomous Robots*. <https://doi.org/10.1007/s10514-018-9762-1>.
- Takano, W., Yamane, K., & Nakamura, Y. (2007). Capture database through symbolization, recognition and generation of motion patterns. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 3092–3097).

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Wataru Takano** is an Associate Professor at the Department of Mechano-Informatics, School of Information Science and Technology, University of Tokyo. He was born in Kyoto, Japan, in 1976. He received the B.S. and M.S. degrees from Kyoto University, Japan, in precision engineering in 1999 and 2001, Ph.D. degree from Mechano-Informatics, the University of Tokyo, Japan, in 2006. He was an Project Assistant Professor at the University of Tokyo from 2006 to 2007, and a Researcher on

Project of Information Environment and Humans, Presto, Japan Science and Technology Agency from 2010. His field of research includes kinematics, dynamics, artificial intelligence of humanoid robots, and intelligent vehicles. He is a member of IEEE, Robotics Society of Japan, and Information Processing Society of Japan. He has been the chair of Technical Committee of Robot Learning, IEEE RAS.



**Yoshihiko Nakamura** is a Professor at the Department of Mechano-Informatics, School of Information Science and Technology, University of Tokyo. He was born in Osaka, Japan, in 1954. He received the B.S., M.S., and Ph.D. degrees from Kyoto University, Japan, in precision engineering in 1977, 1978, and 1985, respectively. He was an Assistant Professor at the Automation Research Laboratory, Kyoto University, from 1982 to 1987. He joined the Department of Mechanical and

Environmental Engineering, University of California, Santa Barbara, in 1987 as an Assistant Professor, and became an Associate Professor in 1990. He was also a co-director of the Center for Robotic Systems and Manufacturing at UCSB. He moved to University of Tokyo as an Associate Professor of Department of Mechano-Informatics, University of Tokyo, Japan, in 1991. His fields of research include the kinematics, dynamics, control and intelligence of robots ? particularly, robots with non-holonomic constraints, computational brain information processing, humanoid robots, human-figure kinetics, and surgical robots. He is a member of IEEE, ASME, SICE, Robotics Society of Japan, the Institute of Systems, Control, and Information Engineers, and the Japan Society of Computer Aided Surgery. He was honored with a fellowship from the Japan Society of Mechanical Engineers. Since 2005, he has been the president of Japan IFToMM Congress. He is a foreign member of the Academy of Engineering in Serbia and Montenegro.