Proceedings of the 2015
IEEE Conference on Robotics and Biomimetics
Zhuhai, China, December 6-9, 2015

Sparsely Extracting Stored Movements to Construct Interfaces for Humanoid End-effector Control

Yuka Ariki[1], Tetsunari Inamura[2], Shiro Ikeda[3] and Jun Morimoto[4]

Abstract— **This paper proposes a robot interface design method by which we can control humanoid end-effector movements with such a low-dimensional input device as a gamepad. In our proposed method, first, the numbers of movement trajectories to accomplish different tasks are generated using a simulated robot model and stored in a database. Second, a human user demonstrates the current task-related behavior. Third, the corresponding stored movements for the demonstrated human behavior are sparsely extracted by a sparse coding method. Finally, the sparsely extracted movement bases are linearly combined to generate a novel movement to accomplish a new target task where the linear weight parameters are modulated by the gamepad. We easily generated such complicated hand movements as spiral motions on a small humanoid robot with our proposed interface.**

I. INTRODUCTION

As robotic technologies continue to progress, robots are expected to support daily human activities, engage in manufacturing, and assist disaster-recovery efforts. To use robotic devices for supporting human activities, we must properly design an interface that connects human users and robots. Using vision systems to monitor human gestures for recognizing user intentions is a popular approach [1], [2], [3]. A hand movement recognition device such as a Wii controller has also been used as a robot interface [4]. Measuring myoelectric signals is another promising approach to design an intuitive interface to control robotic devices [5], [6], [7]. Brain machine interfaces are also getting attention as a potentially very useful robot control interface [8], [9].

On the other hand, one of the most popular interfaces to control high-dimensional systems is a gamepad. Using such a low-dimensional input device as a gamepad to control high-dimensional systems, including in-game characters, is a very common interface implementation.

Interestingly, users do not need to spend a significant amount of time for training to achieve a certain skill level to properly control the in-game characters. This is probably because the design that connects the input device to the high-dimensional system is proper. The development of an proper robot interface, which allows users to control a complicated robotic device without significant effort, is critical.

To develop such a useful robot interface, we need to find proper constraints to connect low-dimensional input devices with a robot system. In previous studies, manifold learning methods, e.g., [10], [11], have been used to find low-dimensional representations of high-dimensional human movements as proper constraints [12], [13], [14], [15], [16], [17]. However, in these previous approaches, since an appropriate low-dimensional manifold needs to be extracted for each target behavior, they are not suitable for generating a wide variety of movements.

On the other hand, task-space control is a standard approach to generate high-dimensional whole-body humanoid movements that correspond to task-relevant end-effector trajectories [18], [19], [20]. However, it is not very easy to generate three-dimensional task-space hand movements by a gamepad when users need to generate complicated hand trajectories. In addition, since users need to continuously control a humanoid end-effector using the input device, generating fast movements is also difficult.

In this paper, we consider a different approach for a robot interface design. We adopt the ideas of *movement primitives* [21] or *motor tapes* [22] to generate robot movements. Previous studies showed that these approaches are useful for generating a variety of movements from previously learned motions to adapt to the surrounding environments and the given tasks [23], [22]. In our approach, we first store different trajectories as basis movements that can be considered movement primitives. Then we use a sparse coding approach by which a newly observed movement is represented with sparsely selected basis trajectories. By linearly combining the selected basis trajectories, users can generate different types of movements, where the combinations of the bases is determined using a low-dimensional input device such as a gamepad.

This paper is organized as follows. In Section II, we explain our strategy for constructing a low-dimensional interface for humanoid end-effector control with a sparse coding approach. In Section III, we introduce our experimental setups that are composed of a small humanoid robot platform and a low-dimensional input device. In Section IV, we show our experimental results using the simulations and a real humanoid robot. We also present control performances with a designed humanoid interface, which we compare with a standard task-space control interface.

[1]Y. Ariki is with Dept. of Science and Engineering, Ritsumeikan University, 1-1-1 Kusatsu NoziHigashi, Shiga, Japan yayukaar@fc.ritsumei.ac.jp

[2]T. Inamura with National Institute of Informatics, The Graduate School for Advanced Studies, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan inamura@nii.ac.jp

[3]S. Ikeda with The Institute of Statistical Mathematics, 10-3 Midori-cho, Tachikawa, Tokyo 190-8562, Japan shiro@ism.ac.jp

[4]J. Morimoto is with Dept. of Brain Robot Interface, ATR Computational Neuroscience Laboratories, 2-2-2 Hikaridai Seikacho Sorakugun, Kyoto, Japan xmorimo@atr.jp

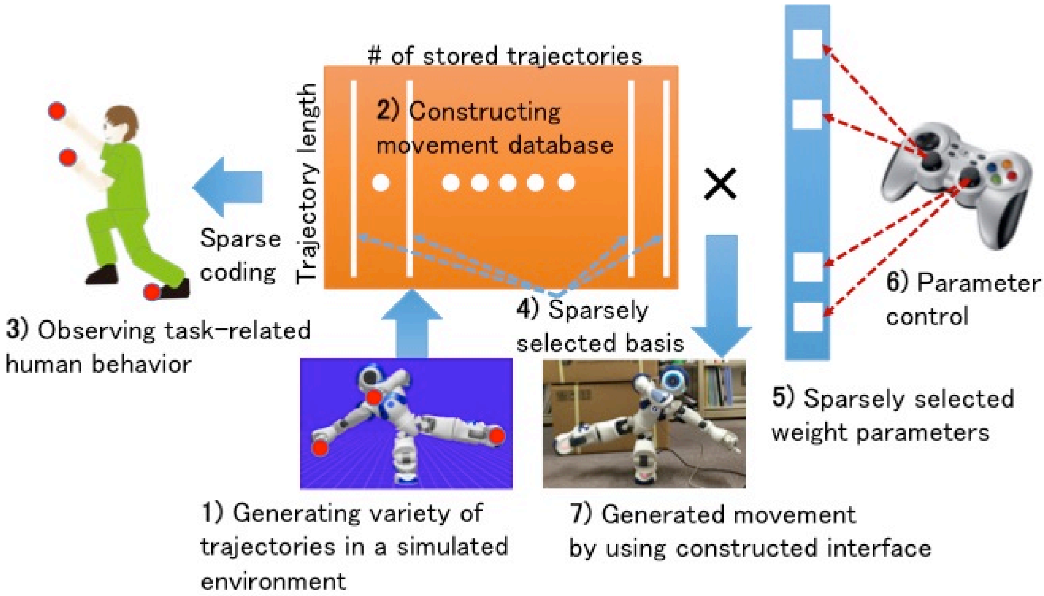

Fig. 1. Schematic diagram of our proposed framework. In our proposed method, 1) numbers of movement trajectories to accomplish different tasks are generated with a simulated robot model and 2) stored in a database; 3) human user demonstrates current task-related behavior; 4), 5) corresponding stored movements and parameters to demonstrated behavior are sparsely extracted by a sparse coding method; 6), 7) sparsely extracted movement bases are linearly combined to generate novel movement to accomplish a new target task where linear weights for combinations are controlled by gamepad.

II. Methods

Here we introduce our interface design method for humanoid end-effector control. Figure 1 shows the interface design strategy. First, we present how we store basis movements into a database by using a simulated environment. Second, we introduce how to extract basis movement trajectories from the stored movements. Finally, we describe how to generate humanoid end-effector movements by using a low-dimensional input device such as a gamepad based on the extracted bases.

A. Storing movement bases

We first store numbers of end-effector movement trajectories in a movement database by using a simulated humanoid robot model. It is easy to design wide variety of movements for accomplish different tasks in a simulated environment. We then solve inverse kinematics problems to generate corresponding joint movements when we control a humanoid platform. A simulated robot generates numbers of task-related hand movements and these end-effector trajectories are stored as movement bases. Here we assume that we have N trajectory bases, where each end-effector trajectory basis has m sample points. Then, the movement basis storage can be represented as:

$$\mathbf{D} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_N], \tag{1}$$

where $\mathbf{b}_i \in \mathbf{R}^m$ denotes each trajectory basis.

B. Extracting movement bases

We then linearly combine end-effector trajectory basis, where the trajectories are sparsely extracted from the movement database. For selecting these basis, we consider using a sparse coding method based on L_1-norm regularization [24]. Concretely, we solve a sparse linear regression problem:

$$\mathbf{x}^* = \text{argmin}_{\mathbf{x}} ||\mathbf{y} - \mathbf{D}\mathbf{x}||_2^2 + \lambda ||\mathbf{x}||_1, \tag{2}$$

where $\mathbf{x}$ denotes weight parameter vector and $\mathbf{y}$ denotes an observed hand-movement trajectory of a human demonstrator. Sparsity of the solution depends on the parameter λ. $\mathbf{x}^*$ denotes optimized weight parameter.

C. Generating movements by constructed interface

Finally, we generate task-related end-effector movements by controlling the weight parameters that correspond to the extracted basis trajectories. To control the weight parameters, we use low-dimensional input device such as a gamepad. By using the modified weight parameters, end-effector movement trajectories are represented as:

$$\tilde{\mathbf{y}} = \mathbf{D}\tilde{\mathbf{x}}, \tag{3}$$

where

$$\tilde{\mathbf{x}} = \mathbf{x}^* + \Delta\mathbf{x} \tag{4}$$

denotes the modified input parameter vector composed of the parameter find in the reconstruction process $\mathbf{x}^*$ in (2) and the low-dimensional control input $\Delta\mathbf{x}$. $\tilde{\mathbf{y}}$ denotes the newly generated robot movement trajectory.

III. Experimental setups

A. Small humanoid robot platform and Low-dimensional input device

We use a 25-DOF small humanoid platform (see also Fig. 2(A)) to evaluate our proposed interface design method. As an input device, we use a gamepad which has two of

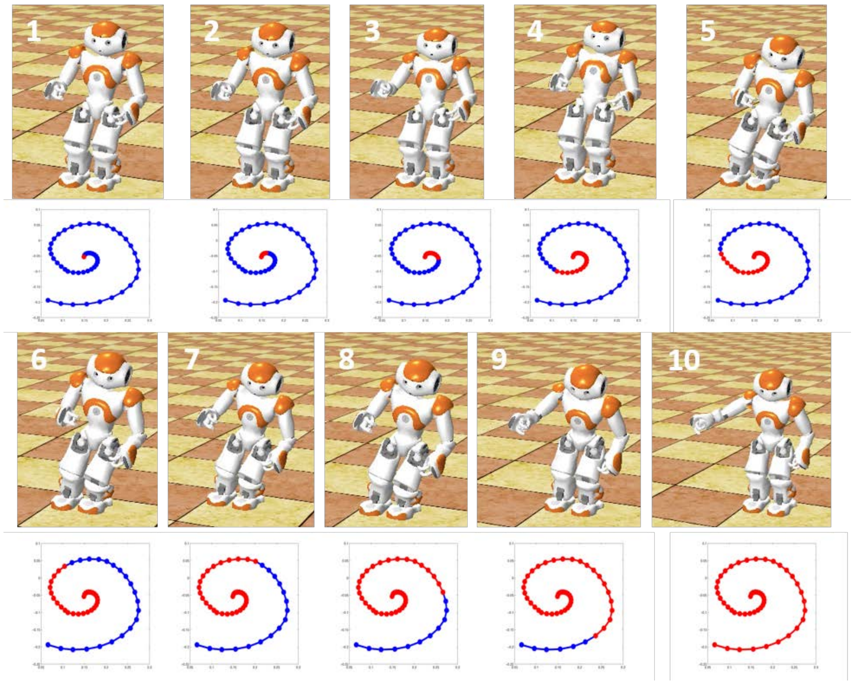

Fig. 3. An example of the stored spiral trajectories that are used in our experiment, and corresponding robot postures.

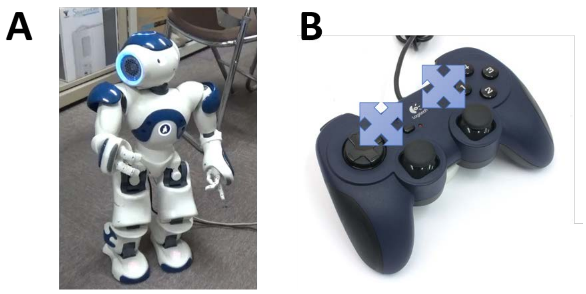

Fig. 2. (A) 25-DOF small humanoid platform, (B) Low-dimensional input device. The device has two of two-DOF analog joysticks.

two-DOF analog joysticks (see also Fig. 2(B)). The analog joystick inputs were used to determine the parameter inputs $\Delta\mathbf{x}$ in (4). Then, a trajectory composed of the linearly combined bases was used to control the small humanoid platform.

B. Constructing spiral movement database

We constructed end-effector movement database using a simulation environment of the small humanoid platform depicted in Fig. 2(A). In this study, we consider generating complicated hand movements such as spiral trajectories. In our experiment, 96 spiral trajectories are generated in the simulated environments, where the spirals with four different sizes, three different length, and eight different rotation angles were considered, i.e., $4 \times 3 \times 8 = 96$. Then, the generated trajectories were stored and represented in a matrix from $\mathbf{D}$ in (1). Figure 3 shows an example of the stored spiral trajectories that were used in our experiment, and corresponding robot postures.

C. Spiral movement demonstration

From the constructed movement database, we sparsely extracted four movement bases by observing a hand movement $\mathbf{y}$ demonstrated by a human user. As examples, a human user demonstrated three different types of spiral motions as depicted in Figs. 4(A),(D) and (G). Then, for each observed spiral movement, corresponding trajectories in the movement database are sparsely selected by the basis selection method presented in (2). The four movement bases are selected based on the absolute value of the element of the optimized parameter vector $\mathbf{x}^*$ in (2). Note that we used lambda parameter $\lambda = 0.001$ so that less than ten movement bases were sparsely selected from the database which includes 96 bases. In other words, based on the absolute value of the element of the optimized parameter vector, we finally chose the four bases from the less than ten bases selected by the sparse coding method.

D. Spiral movement generation

The parameters for the selected basis trajectories are controlled to generate modified spiral movements. These parameters were controlled by using the gamepad (see also Fig.2(B)). The two of the two-DOF joysticks were assigned to control the selected four parameters. We show that the small humanoid platform can generate similar but different spiral hand movements from the observed spiral trajectories by controlling parameters with using the gamepad. We then also show that we can control the small humanoid robot to generate the modulated spiral movement.

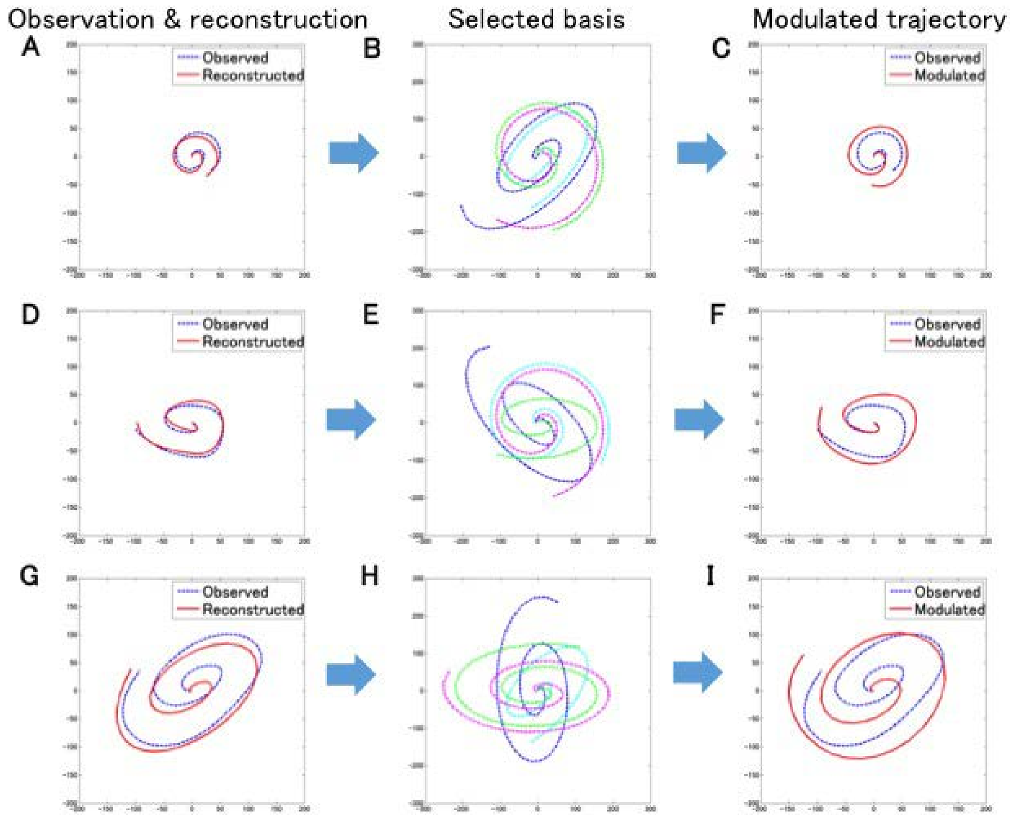

Fig. 4. Generated spiral movements. (A)-(C) correspond to first observed movement, (D)-(F) correspond to second observed movement, (G)-(I) correspond to third observed movement. (A),(D),(G): Demonstrated and reconstructed spirals. Demonstrated movements were measured by using Visualeyez real-time motion capture system (Phoenix Technologies Inc.). (B),(E),(H): Selected basis end-effector trajectories. (C),(F),(I): Modulated spirals by using proposed interface.

IV. Results

A. Generating spiral movements

We first showed that we were able to generate spiral movements by using the low-dimensional input device. Figure 4 shows the results of the interface construction procedure. Human user demonstrated different spiral movements that have three different shapes (see also Figs. 4 (A),(D) and (G)). For each demonstrated movement, the four movement bases were selected by using (2) (see also Figs. 4 (B),(E) and (H)) and the corresponding parameters were controlled by using the gamepad. In Figs. 4(C),(F) and (I), modulated spirals are presented for each different demonstrated movement. By using the constructed robot interface, we showed that the user was able to easily control the end effector of the small humanoid robot by using the gamepad input device based on the sparsely selected basis trajectories. Figure 5 shows the real humanoid robot movement when the robot generated the modulated spiral movement which is presented in Fig.4(C).

B. Spiral movement control to generate a desired trajectory

We then showed that the constructed humanoid interface were able to be used to follow a target spiral movement by controlling the small humanoid robot with using the gamepad input device. Although generating a spiral movement by using the constructed robot interface was easy, still, some training trials were necessary for a user to follow the target spirals. We show the learning performance of a user to generate a target spiral movement in Fig. 6. Within 30 trials, humanoid end-effector control performance was much improved, where each trial only takes around one second.

As a comparison, we also asked the same subject to generate a target spiral movement by using an control interface by which the hand position in a cartesian coordinate were directory controlled by the two-DOF analog joystick. Figure 7 shows the comparison of the generated spiral movements. As in Figure 7(B), direct control of the hand position for generating the spiral trajectory by using the gamepad was very difficult while, as in Fig.7(A), it was easy to generate spiral end-effector movement to follow the target trajectory by using the proposed interface. These results clearly show that our proposed approach was much easier to be used to generate a complicated end-effector trajectory such as a spiral movement.

V. Conclusions

We proposed a humanoid interface design method. We showed that we can easily generate spiral hand movements on a small humanoid platform by using the gamepad input device. Humanoid robot movements were controlled by the

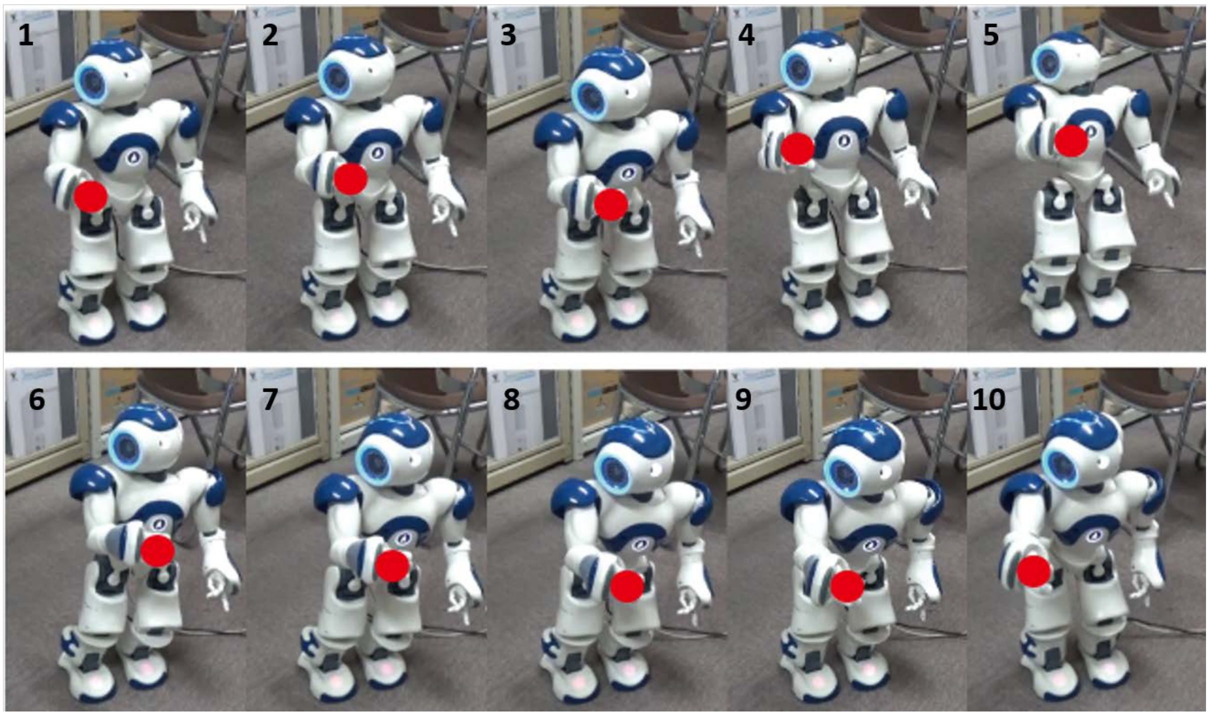

Fig. 5. Generated spiral movements on real humanoid platform. Red disks represent end-effector positions.

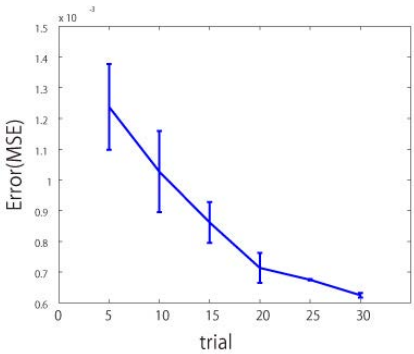

Fig. 6. Spiral control performance using the proposed interface. Within 30 trials, humanoid end-effector control performance was much improved, where each trial only takes around one second. Mean and standard deviation of errors for every five trials are plotted.

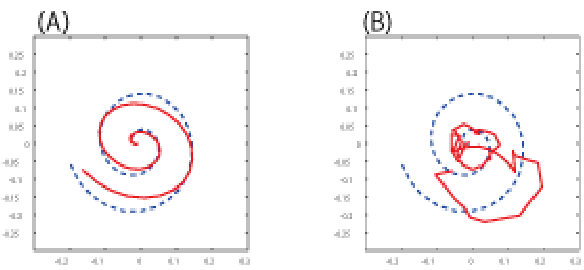

Fig. 7. Comparison of the generated spiral movements. This results clearly show that our proposed approach was much easier to be used to generate a complicated end-effector movement such as a spiral trajectory. (A)Proposed. (B)Control interface by which the hand position in a Cartesian coordinate were directory controlled by the two-DOF analog joystick.

combined basis motion trajectories, where the parameters to combine each basis were determined by the input command specified with the gamepad. The basis trajectories were sparsely extracted from the stored spiral trajectories by using the sparse coding method. Different spiral movements were successfully generated by using the constructed interface. Since the stored basis trajectories were generated in the simulated humanoid robot system, it would not be always suitable to code the observed human movements. In such case, we can possibly use a dictionary learning method [25] to adapt the stored basis motion to observed human movement trajectories. Therefore, in our future study, we consider using this kind of adaptation mechanism to refine the stored basis trajectories to properly represent observed human behaviors.

ACKNOWLEDGMENT

This is the results carried out under MIC-SCOPE. Part of this research was supported by SRPBS, AMED/MEXT. This research was also partially supported by MEXT KAKENHI grant Number 23120004, ImPACT Program of Council for Science, Technology and Innovation, and NEDO.

REFERENCES

[1] H. Do, C. Mouser, Y. Gu, W. Sheng, S. Honarvar, and T. Chen, "An open platform telepresence robot with natural human interface," in *Cyber Technology in Automation, Control and Intelligent Systems (CYBER), 2013 IEEE 3rd Annual International Conference on*, May 2013, pp. 81–86.

[2] J. T. C. Tan and T. Inamura, "Sigverse - a cloud computing architecture simulation platform for social human-robot interaction," pp. 1310–1315, 2012.

[3] R. Yan, K. P. Tee, Y. Chua, and H. Tang, "A gesture recognition system using localist attractor networks for human-robot interaction," in *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*, Dec 2010, pp. 1217–1222.

[4] D. Balakrishna, P. Sailaja, R. Rao, and B. Indurkhya, "A novel human robot interaction using the wiimote," in *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*, Dec 2010, pp. 645–650.

[5] T. Matsubara, S. Hyon, and J. Morimoto, "Learning and adaptation of a stylistic myoelectric interface: Emg-based robotic control with individual user differences," in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, Dec 2011, pp. 390–395.

[6] K. Kiguchi and Y. Hayashi, "An emg-based control for an upper-limb power-assist exoskeleton robot," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 4, pp. 1064–1071, Aug 2012.

[7] J. Furukawa, T. Noda, T. Teramae, and J. Morimoto, "An emg-driven weight support system with pneumatic artificial muscles," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–9, 2014.

[8] J. Zhao, Q. Meng, W. Li, M. Li, F. Sun, and G. Chen, "An openvibe-based brainwave control system for cerebot," in *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*, Dec 2013, pp. 1169–1174.

[9] T. Noda, N. Sugimoto, J. Furukawa, M. Sato, S. Hyon, and J. Morimoto, "Brain-controlled exoskeleton robot for bmi rehabilitation," in *IEEE-RAS International Conference on Humanoid Robots*, 2012, pp. 21–27.

[10] P. Dollar, S. Belongie, and V. Rabaud, "Learning to traverse image manifolds," in *Proceedings of NIPS*, 2006, pp. 361–368.

[11] P. Dollar, V. Rabaud, and S. Belongie, "Non-isometric manifold learning: Analysis and an algorithm," in *Proceedings of the 24th international conference on Machine learning*, 2007.

[12] K. Tatani and Y. Nakamura, "Dimensionality reduction and reproduction with hierarchical nlpca neural networks-extracting common space of multiple humanoid motion patterns," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2003, pp. 1927–1932.

[13] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović, "Style-based inverse kinematics," in *ACM SIGGRAPH 2004 Papers*, ser. SIGGRAPH '04, 2004, pp. 522–531.

[14] C.-S. Lee and A. Elgammal, "Human motion synthesis by motion manifold learning and motion primitive segmentation," in *Proceedings of the 4th international conference on Articulated Motion and Deformable Objects*, ser. AMDO'06, 2006, pp. 464–473.

[15] S. Bitzer, I. Havoutis, and S. Vijayakumar, "Synthesising novel movements through latent space modulation of scalable control policies," in *SAB*, 2008, pp. 199–209.

[16] Y. Ariki, T. Inamura, and J. Morimoto, "Learning a humanoid robot interface by embedding a low-dimensional command manifold into a high-dimensional joint action space," in *Proceedings of IEEE/RAS the International Conference on Humanoid Robots*, 2013, pp. 514–519.

[17] ——, "Observing human movements to construct a humanoid interface," in *Proceedings of IEEE/RAS the International Conference on Humanoid Robots*, 2014, pp. 342–347.

[18] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 1, pp. 43–53, 1987.

[19] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational space control: A theoretical and empirical comparison," *I. J. Robotic Res.*, vol. 27, no. 6, pp. 737–757, 2008.

[20] J. Peters and S. Schaal, "Learning to control in operational space," *I. J. Robotic Res.*, vol. 27, no. 2, pp. 197–212, 2008.

[21] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, no. 6, pp. 233–242, 1999.

[22] C. Atkeson and J. Morimoto, "Nonparametric representation of policies and value functions: A trajectory-based approacha," in *Advances in Neural Information Processing Systems*, vol. 15, 2002, pp. 1611–1618.

[23] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 800–815, 2010.

[24] R. Tibshirani, "Regression shrinkage and selectionvia the lasso," *J. Royal. Statist. Soc B*, vol. 49, pp. 702–709, 1996.

[25] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26t International Conference on Machine Learning*, 2009, pp. 689–696.