2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Hierarchical Motor Learning and Synthesis with Passivity-Based Controller and Phase Oscillator

Sang-Ho Hyon, Jun Morimoto and Gordon Cheng

***Abstract*— In this paper, we propose a simple framework for learning and synthesis of fast and complex motor tasks. Where a passivity-based task-space controller acts not only as a full-body force control module, but also as an important module to generate phasic joint patterns. The generated joint patterns are encoded into the paramaters of phase oscillators and form the synergy of the task. Then, similar and/or faster motions are synthesized by superposing the task space controller output and the oscillator output with the modified oscillator amplitudes and/or frequencies. We present some examples of whole-body motion synthesis on a human-sized biped humanoid robot including squatting, dancing and stepping while bipedal balancing. The simulation and experimental videos are supplemented.**

I. Introduction

A. Motivation

Our motivation to study humanoid robots is to develop novel human-friendly assistive devices such as exoskeleton systems. For such systems, it is very important to establish a suitable framework for motor learning and synthesis, which can be seamlessly coupled to various level of human motor control, from joint-level to task-level. At the joint-level, controlling force/torque is critical so that the robot can accept external forces applied from human and environment. As well known from many physiological studies, musculoskeletal systems can control the joint stiffness and the load simultaneously.

At the task-level, how to coordinate whole-body motions, *synergy* [1], is important because there are many degrees of freedom in human body, and transforming task-space force into the joint-space torque is an ill-posed problem. For example, controlling center of mass (CoM) is the most fundamental task objective especially for human balancing, which can be achieved by appropriately controlling the contact forces between human and environment. Without a suitable synergy, it is impossible to obtain a human-friendly balance-assist function.

Modular control architecture will be one of the promising approach for learning and synthesis of multiple complex motor tasks [2], where the output of each module is summed up with the responsibility weights. For position control tasks, an imitation learning paradigm has been successfully adopted to generate human-like movements [3], where the synergy has been extracted from observation. For force control tasks such as balancing, however, no such imitation is available because we cannot observe others forces.

The authors are with JST, ICORP, Computational Brain Project, Honcho 4-1-8, Kawaguchi, Saitama 332-0012, Japan, and ATR Computational Neuroscience Laboratories, Hikaridai 2-2-2, Souraku-gun, Kyoto 619-0288, Japan. {sangho, xmorimo, gordon}@atr.jp

This work was supported by the Keihanna branch of the National Institute of Communication Telecommunication (NiCT), Japan

This paper has supplementary material provided by the authors. This is a single MPEG4 format movie which shows the empirical results. The movie is 5 MB in size. For the details, see “ReadMe.txt”.

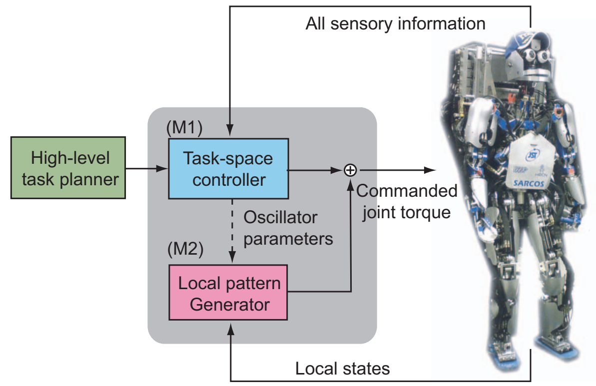


Fig. 1. The conceptual organization of our learning/synthesis framework. The task-space control module (M1) generates whole-body motions, while achieving task-space control objectives given by a high-level motor center. The pattern generator module (M2) encodes the generated motions into the oscillator parameters. Faster and similar motor tasks can be achieved by the superposition of (M1) and (M2) with the modulated oscillator parameters.

On the other hand, we have developed a passivity-based contact force controller for redundant humanoid robots and experimentally validated [4]. The key idea was to suppress *internal motions* (or *self-motions*) [5] of un-modeled nonlinear dynamics by simple dissipation terms [6]. Although this passivity-based redundancy resolution is robust to modeling and sensing errors compared to inverse dynamics-based solutions, the method cannot achieve fast dynamic task because of the simple dissipation term.

A naive way of reducing the effect of the dissipation terms is to increase the task-space feedback gain (task-space stiffness). As we observed from our experiments on humanoid robots, however, a large task-space gain easily results in awkward resonance, due to the forward kinematics errors which are caused by the accumulation of the joint posture sensing. This is one of the serious and real problems of all task-space controllers, which cannot be seen in ideal simulation environments. The instability due to the high feedback gain can also occur when there is considerable transmission delay in sensory motor systems. Even if some fast task-space tracking is *fortunately* achieved by increasing the task gain, but again the internal motions may appear. This trade-off shows that there are obvious limitations in the passivity-based task-space control *when it is used alone*.

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

B. Overview of the proposed framework

In this paper, we propose a simple framework for learning and synthesis of fast and complex motor tasks for humanoid robots via superposition of a passivity-based task-space controller with a local joint pattern generator. The conceptual organization is illustrated in Fig. 1. We use the passivity-based task-space controller not only as a task-space position/force control module, but also as an important module to generate joint-space motion patterns.

For the learning stage, we assume a control hierarchy between the task-space controller and the local pattern generator by:

(M1) Task-space control module achieves *slow* but *complex* task objectives;

(M2) Local pattern generator module encodes the generated joint patterns into the parameters of some oscillator models.

For the synthesis stage, we use the learned *task-related* oscillator simultaneously with the task-space control by simple superposition. In this way, we expect the generated motions being valid, at least statically, because the superposition is done for the experienced (slow) motions. The advantages of this synthesis are:

- The ill-posedness is solved by the explicit joint patterns learned in the oscillators;
- Some critical task-space control objectives such as balancing is imposed all the time;
- Fast motions are possible by modulating oscillator parameters;
- Kinematic constrains such as joint limits can be met.

By this simple synthesis, we try to achieve *fast* and *complex* motions effectively. Although the oscillator outputs statically-valid motions, the oscillator *alone* cannot outputs dynamically-valid motions for fast motor tasks in general. Therefore, the task-space control plays important role for modulating the oscillator outputs. Balancing control will be the strongest modulation. Once the target motor task has been achieved by the synthesis, the obtained motions are again encoded into the oscillator parameters, which are stored into a motion library. In this way, we expect various fast and complex motions can be learned.

We demonstrate our motor learning and synthesis framework on simulations and experiments on a full-sized biped humanoid robot. Because of the page limit, the extended simulation and experimental results are supplemented by the conference video.

II. Hierarchical motor control by superposition

A. Task space controller

The task-space control is thought to be a part of the central nervous system (CNS) of humans [7], where CNS composites necessary control inputs to the lower level from proprioceptive, graviceptrive/vestibular and visual feedback signal. As the task control module, we employ our whole-body contact force controller [4]. To focus on the main subject of this paper, here we give the simplest form of the controller. Consider a multi-DoF humanoid robot as shown in Fig. 2. Let $r_C = [x_C,\ y_C,\ z_C]^T \in \mathcal{R}^3$ be the position vector of CoM in the world coordinate frame Σ_W, $q \in \mathcal{R}^n$ be the joint angles and $\phi \in \mathcal{R}^3$ be the attitude of the base. Suppose the foot is in contact with the ground, and the *center of pressure* (CoP), whose position vector from the CoM is indicated by $r_P = [x_P,\ y_P,\ z_P]^T \in \mathcal{R}^3$, lies within the supporting region. We introduce a *gross applied force*, or *ground applied force* (GAF) $f_P = [f_{xP},\ f_{yP},\ f_{zP}]^T$, defined as $f_P := -f_R$, where f_R is the ground reaction force (GRF). The GAF represents the gross force that the robot applies to the environment. The control objective here is to bring f_P to the desired value $\overline{f}_P$, which is give by a task, such as balancing and/or walking.

The simplest form[1] of the passivity-based contact force control is given by

$$\tau = J_P^T \overline{f}_P - D\dot{q} \quad (1)$$

$$\overline{f}_P = f_u + Mg \quad (2)$$

where $J_P(\phi, q) \in \mathcal{R}^{3\times n}$ is the Jacobian from CoM to the *desired* CoP and $f_u = [f_{ux},\ f_{uy},\ f_{uz}]^T \in \mathcal{R}^3$ is a certain new force input. This yields the convergence of GAF, $f_P \to f_u + Mg$ as $t \to \infty$, provided the joint-wise damping D (positive diagonal matrix) is designed so that the internal dynamics is stable as shown in the Appendix of [4]. The optimal contact force distribution for multiple contact cases are also presented in [4].

As the basic assumption of the controller, we suppose the linear force mapping between the contact forces and the joint torques, J_P^T, are known. This mapping is described in sinusoidal functions of joint angles and mass distribution, which can be effectively represented by some neural network model to be trained.

In general, however, it is difficult for task-space controllers to solve all the constraints. Therefore, in the learning process, we allow some heuristics to be combined. For example, the joint states are not updated in a singular posture. If the robot is in an upright posture, the knee joint is fully extended, and it becomes difficult to bend the knee. If we allow the posture enters into singularity, we should introduce

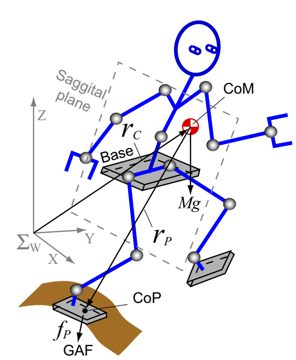


Fig. 2. Definition of positions and forces of a biped humanoid robot

[1]This is the simplest, but approximated formula. See [4] for the exact formulation.

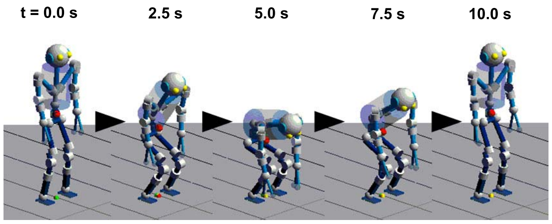


Fig. 3. (**Video. 1**) Slow squat with the frequency of 0.1 Hz while balancing. Joint limits are avoided.

a heuristic *recovery* rule from it, similar to the human's knee locking/release. Moreover, in the learning process we should consider kinematic constraints into its motions. For assistive devices, learning the range of motion of humans is very important. Simple joint limits avoidance are possible by a repulsive force field around the joint limit:

$$U(q) = \frac{\alpha}{(q - q_{min})^2} + \frac{\alpha}{(q - q_{max})^2}, \quad (3)$$

and the resultant task space controller is given by

$$\tau = J_P^T (f_u + Mg) - D\dot{q} + \frac{\partial U}{\partial q} \quad (4)$$

Fig. 3 (and **Video. 1**) shows a slow squatting motion achieved only by (4) only, where two simple feedback controllers are used. The one is the CoM stabilization

$$f_{ux} = -K_{PC} x_C - K_{DC} \dot{x}_C, \quad (5)$$

$$f_{uy} = -K_{PC} y_C - K_{DC} \dot{y}_C, \quad (6)$$

where $K_{PC}, K_{DC} > 0$ are the task-space PD-gains, and the origin is defined as the center of the feet. The other one is the head height control. A vertical PD-feedback force is applied between the head and the center of the feet (CoM height is not controlled: $f_{uz} = 0$). A sinusoidal pattern with the amplitude 0.3 m and the frequency of 0.1Hz has been set. Since the motion is very slow, the joint limits are easily avoided. In this way, the robot may achieve relatively complex joint space trajectories.

B. Local pattern generator

Once some joint motions are generated by the task space control module, we can extract the features for the lower layer module. If the motions are periodic, one can consider them as the outputs of some oscillator models. Even if the motions are one-path trajectories, as long as the terminal velocities are zero, one can consider them as the half paths of oscillations.

In this paper, we adopt a simple oscillator; phase oscillator parameterized by the phase frequency ω_j and the amplitude $r_j(\phi)$ with respect to the phase, for all active joints $j = 1, 2, ..., n$. The amplitude $r(\phi)$ is represented by radial basis function networks and the weight is learned through the motion [8]. In addition, each oscillator has the phase shift ϕ_0 as the parameter. All these parameters represent the synergy for specific tasks.

Since we are using passivity-based task-space controller, internal motion is suppressed. As long as the internal motion is well suppressed, the task-space control torque updates the joint states to an appropriate direction. The update direction is unique. As a result, we can obtain *task-related* inverse kinematics. Thus-obtained inverse kinematics includes not only the instantaneous relationship between the task position and joint angles, but also the tangential vector of the motion because the phase trajectory is continuous. Moreover, we can magnify the phase frequency for faster motions.

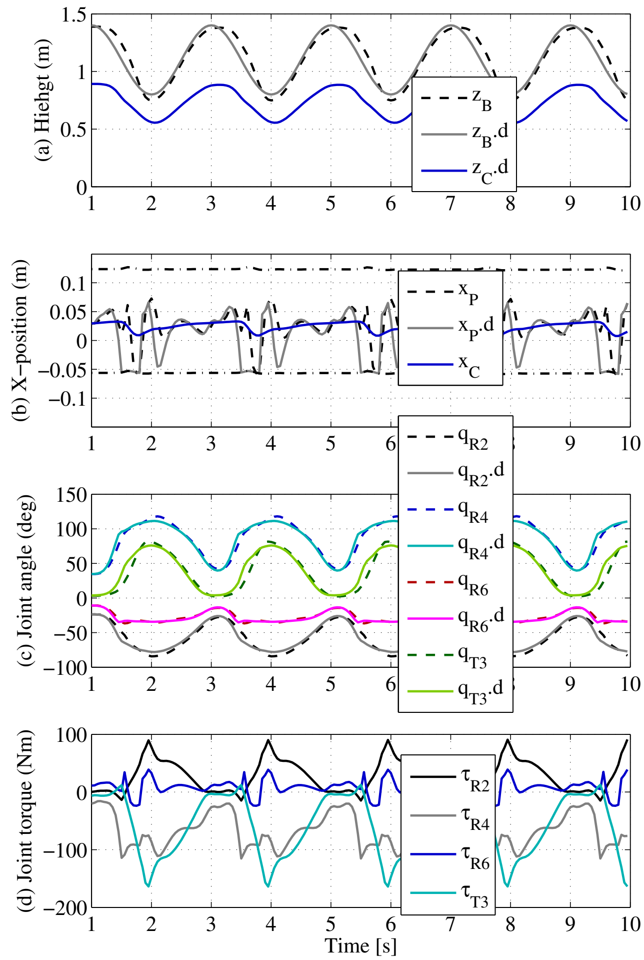


Fig. 4. (**Video. 2**) Fast squat with 0.5 Hz. The frequency is five times higher than the original one. The height of the head is indicated by z_B. The CoM as well as CoP are shifted from the origin because of the balance maintenance. There are tracking errors both in the task-space and joint-space. To make the tracking errors small, we need to re-train the phase oscillator during the motion synthesis. The overall behavior is very similar to Fig. 3. (Subscripts) R2: Right hip, R4: Right knee, R6: Right ankle, T3: Torso bend.

C. Synthesis by weighted superposition

The main purpose of the control framework is to utilize the learned motion pattern for similar but more dynamic tasks. For this purpose, we synthesize the motion by:

(S1) Superposing the task-space control and the stiffness control around the learned joint patterns;

(S2) Modulating the stiffness according to the task difficulty.

For (S1), the total controller can be implemented by:

$$\tau = J_P^T (f_v + Mg) - D\dot{q} - \kappa \left\{ K_P(q - \overline{q}) + K_V(\dot{q} - \dot{\overline{q}}) \right\} \quad (7)$$

where $\overline{q}$ and $\dot{\overline{q}}$ are joint states from the oscillator model, K_P and K_V are the position/velocity gains for tracking, and κ is the stiffness scaling (scalar). The introduction of this superposition is not only required by our control purpose, but also supported by a biological musculoskeletal model [9], because the isometric muscle force produces torque and stiffness around thge joints.

For (S2), the stiffness κ can be adjusted by the task difficulty. If the task requires very complex joint motion or fast joint motion, we need to increase the stiffness. If the task motion is similar to the training data (slow motion), we can reduce the stiffness. In some cases, it will be better to have the stiffness *shaped* rather than keeping it constant during motion. For example, we can increase the stiffness to the normal direction of motion and decrease to the tangential direction.

Moreover, to make the tracking error small for the synthesized motions, it will be necessary to:

(S3) Re-train the oscillator parameters.

This can be achieved by some supervised learning schemes. These adaptation is briefly discussed at the end of this paper.

Fig. 4 (and **Video. 2**) shows the result of a high-speed squat using the learned phase oscillator superposed to task-space control. Before the synthesis, the oscillator model has been trained using slow squatting motion described above (Fig. 3 and Video. 1). We put normalized Gaussians on 11 anchor points on the phase coordinate from $-\pi$ to π. The each local phases are extracted from $\text{atan2}(\dot{q}, -(q - q_0))$. Then, the synthesis has been performed by setting the target phase frequency to 0.5 Hz, five times faster than the original one. The local tracking gains are set to $K_P = 100, K_V = 1$, and the stiffness scaling is set to $\kappa = 1$ for all joints.

The top graph shows the height of the head and CoM. There is tracking errors. The second graph shows the balancing performance. Note that the x_C is not maintained at the center of the feet (around 0.03 m in the second graph). This is because of the joint limits (if we remove the joint limits, the CoM error becomes smaller). In this case, (S3), re-training the oscillator parameters, will be required. It should be noted that fast squat was impossible by the task-space controller alone, as shown in **Video. 3**. **Video. 4** shows the 10-times faster squatting, where the stiffness has been increased to $\kappa = 750$. We can apply some on-line learning algorithm to optimize the stiffness κ.

III. Preliminary experiments

This section shows some preliminary experimental results, where the local joint patterns are preset or taught by the users.

A. Dynamic biped stepping

As the first example, we show a knee-stretched biped stepping. In this task, describing pre-planned joint patterns is useful to avoid singularity of the task-space control. Fig. 6 (and **Video. 5**) shows one of the experimental results of continuous stepping achieved by combining task-space contact force controller with sinusoidal joint trajectories imposed on *all* joints. The sinusoidal joint patterns are driven by a *single* phase frequency but different amplitudes. The robot is initially standing still with an upright posture. Then, as the phase evolves the robot makes steps or stops, which can be seen from Fig. 5. The nominal phase frequency is set to 1.5 Hz.

There are many combinations of the nominal frequency and the amplitudes of the sinusoidal patterns, which can make the robot step stably. Anti-gravitational forces are optimally distributed through the CoG (ground projection of CoM) as described in [4]. That is, the more the CoG is shifted to the edge of one foot, the more control torque is supplied to the limb. Therefore, tracking to the joint pattern is greatly improved compared to the case with stiffness only. Furthermore, to obtain better stability, we introduced a method proposed by Morimoto [10] which modulates the desired phase to be synchronized with the actual CoP. The time profile of the modulated phase is shown in Fig. 5(c).

B. Direct motion teaching

As the second example, we show a direct whole-body motion teaching. We allow the robot to follow the external forces while balancing. This compliant behavior is innate in our full-body gravity compensation scheme [4], but this time, we introduce the phase oscillator to learn the motions. Fig. 7 (and **Video. 6**) shows one of the examples. In this experiment, a human operator holds the robot's hands or legs and imposes periodic motions. Then, the oscillator parameters are updated at every knot points, where the joint velocities cross zero. The robot tries to follow the learned motion patterns while keeping its balance.

The time profile is shown in Fig. 8. In this experiment, the duration of teaching is preset in advance. The upper body motions are learned until 20s, and the leg motions are learned until 30s. Finally, from 35s, we have applied a feedback control to synchronize the phase frequency . Once the phases are synchronized, it is also easy to control the phase shift ϕ_0.

IV. Discussion: Learning dyynamic and stiffness

Learning dynamic model during performing task has been proposed in [11], where inverse dynamics is obtained via supervised learning from feedback error signals. This control scheme requires target joint trajectories $\overline{q}$. Therefore, it is natural to introduce this adaptive scheme into our control framework.

Fig. 9 and Fig. 10 (and **Video. 7**) shows one of the simulation examples. In this simulation, we set both task-space and joint-space tracking gains to very small. Therefore, initially, the tracking errors are large in both spaces. As the learning proceeds, the joint-space feedback input is

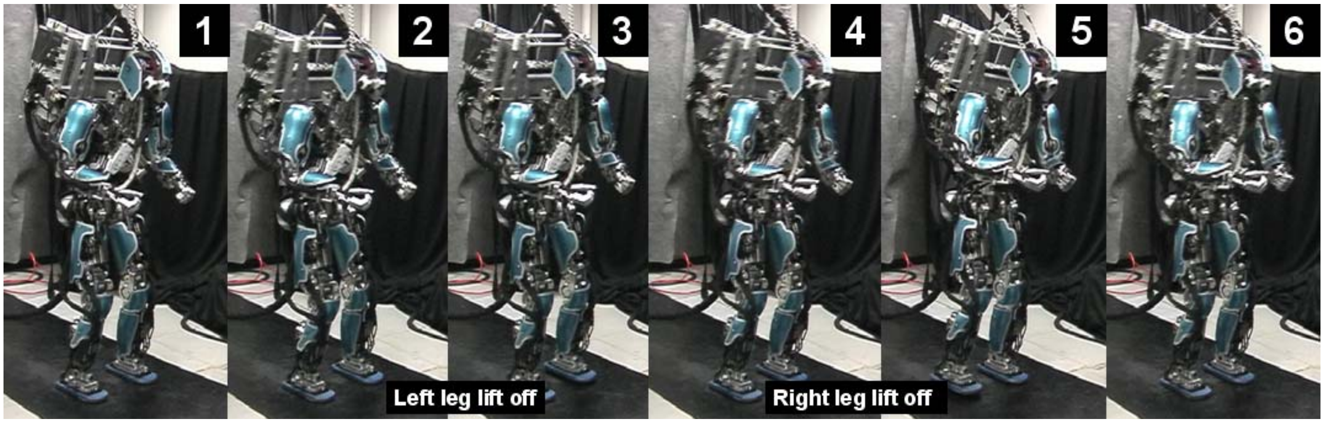


Fig. 6. (**Video. 5**) Continuous stepping achieved by combining task-space controller and the joint stiffness around sinusoidal joint trajectories imposed on "all" joints. The robot initially stands still. Then, as the phase evolves it makes step or stop motions.

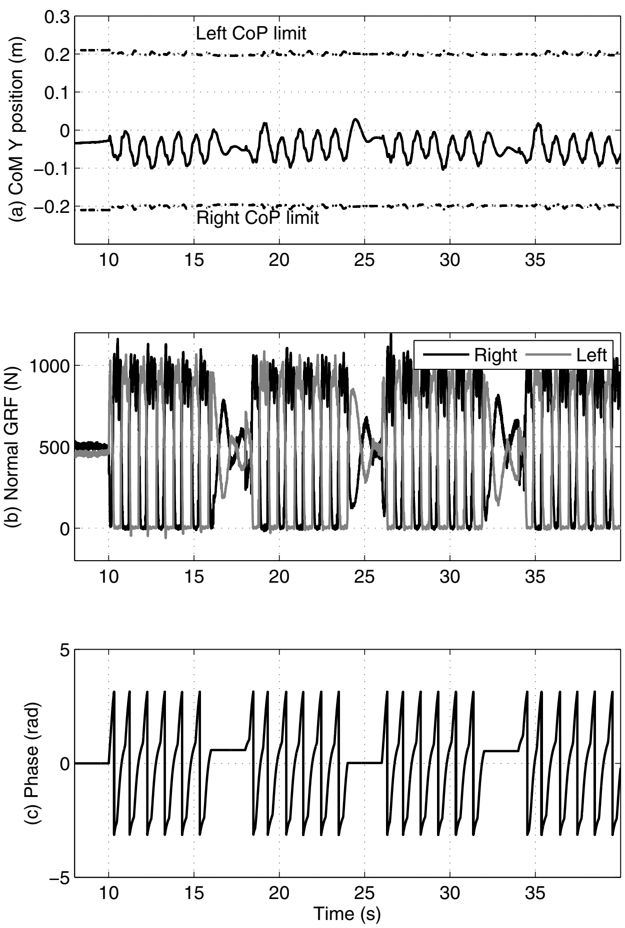


Fig. 5. Experimental data corresponding to Fig. 6. (a) The lateral CoG is oscillating. According to the actual CoG, the full-body joint torque is generated. (b) The normal GRF of each foot, showing the robot is actually stepping and stopping. (c) The step/stop is commanded by the single phase variable. In this case, the robot steps 6-periods, then stops 2-periods, etc. The original frequency of the phase is set to 1.5 Hz, which is then modulated to synchronize with the actual CoP [10].

transferred into the task-space control torque. This is the feedback error learning mechanism. Moreover, we simultaneously adapt the stiffness κ according to the 2-norm of the tracking error. Then, both tracking errors and joint-space feedback inputs are becoming small. However, it is not easy to find adaptation parameters so that the learning converges. Combining some suitable learning/adaptive schemes into the proposed framework, as well as the theoretical analysis, is left for our future studies.

REFERENCES

[1] N. Bernstein, *The control and regulation of movements*. Pergamon Press, 1967.

[2] D. M. Wolpert, R. C. Miall, and M. Kawato, "Internal models in the cerebellum," *trends in cognitive sciences*, vol. 2, no. 9, pp. 338–347, 1998.

[3] C. Atkeson, J. G. Hale, F. Pollick, M. Riley, S. Kotosaka, S. Schaal, T. Shibata, G. Tevatia, A. Ude, S. Vijayakumar, and M. Kawato, "Using humanoid robots to study human behavior," *IEEE Intelligent Systems*, vol. 15, no. 4, pp. 46–56, 2000.

[4] S. Hyon, J. G. Hale, and G. Cheng, "Full-body compliant human-humanoid interaction: Balancing in the presence of unknown external forces," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 884–898, 2007.

[5] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[6] S. Arimoto, H. Hashiguchi, and R. Ozawa, "A simple control method coping with a kinematically ill-posed inverse problem of redundant robots: Analysis in case of a handwriting robot," *Asian Journal of Control*, vol. 7, no. 2, pp. 112–123, 2005.

[7] P. G. Morasso, L. Baratto, R. Capra, and G. Spada, "Internal models in the control of posture," *Neural Netw*, vol. 12, no. 7-8, pp. 1173–1180, 1999.

[8] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning rhythmic movements by demonstration using nonlinear oscillators," in *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 2002, pp. 1398–1403.

[9] N. Hogan, "Adaptive control of mechanical impedance by coactivation of antagonist muscles," *Automatic Control, IEEE Transactions on*, vol. 29, no. 8, pp. 681 – 690, Aug 1984.

[10] J. Morimoto, G. Endo, J. Nakanishi, S. Hyon, G. Cheng, D. Bentivegna, and C. G. Atkeson, "Modulation of simple sinusoidal patterns by a coupled oscillator model for biped walking," in *Proc. of IEEE International Conference on Robotics and Automation*, Orlando, USA, 2006, pp. 1579–1584.

[11] M. Kawato, "Feedback-error-learning neural network for supervised motor learning," *Advanced Neural Computers*, pp. 365–373, 1990.

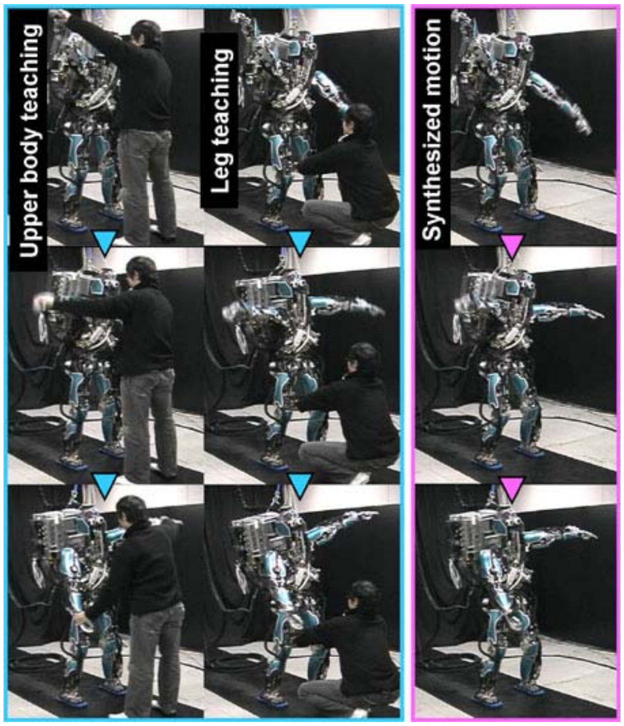


Fig. 7. (**Video. 6**) Direct motion teaching experiment. The human operator is moving upper body and legs while the robot balancing. The robot then follows the learned motion patterns while synchronizing the phase frequencies.

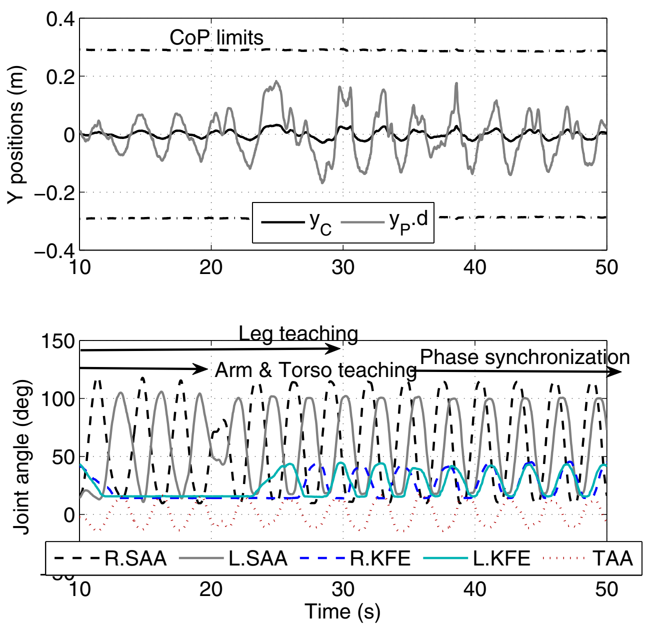
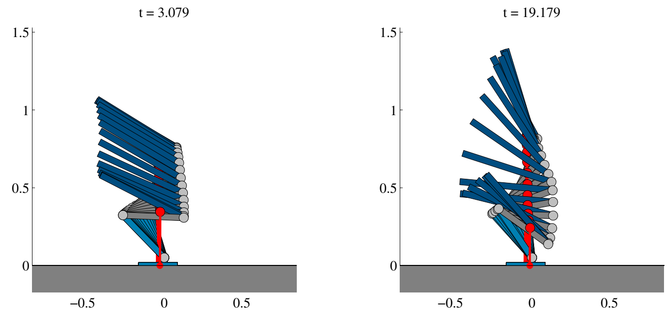


Fig. 8. Experimental data of the direct motion teaching. The upper body motions are learned until 20 s, and the leg motions are learned until 30 s. From 35 s, the phase frequency is synchronized.



(a) Before learning

(b) After learning

Fig. 9. (**Video. 7**) Simulation of the fast squatting with Feedback Error Learning [11]. The vertical lines indicate the virtual pendulum connecting CoM and ZMP. The time profile is shown in Fig. 10

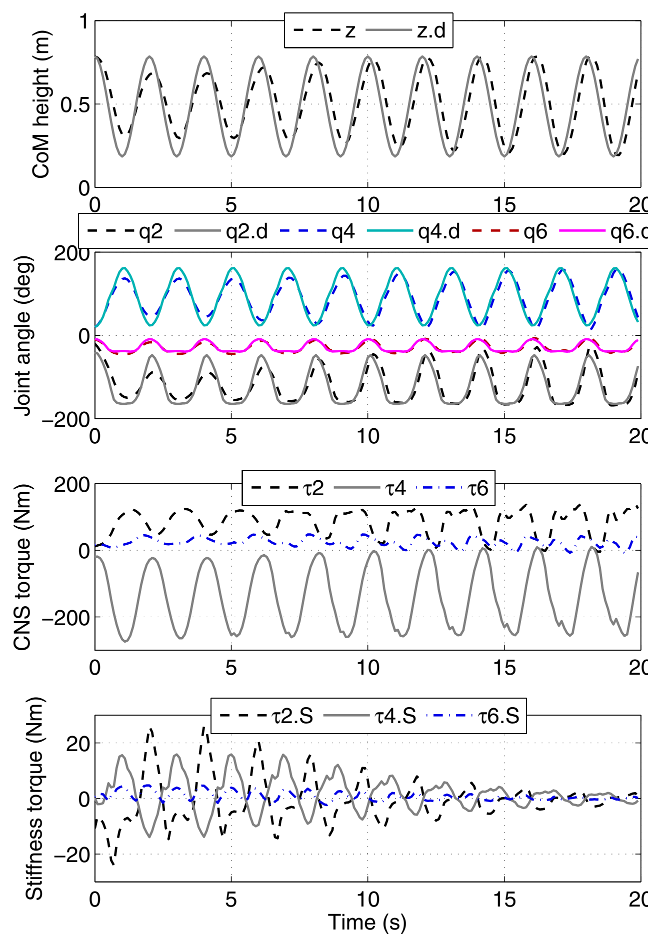


Fig. 10. Simulation data corresponding to Fig. 9. Before learning, the joint stiffness torques are large due to the unmodeled dynamics. The tracking errors are also large due to the small tracking gains. After learning the dynamics and stiffness, the both tracking errors and the stiffness torques become small. Subscripts 2, 4 and 6 indicate the Hip, Knee and Ankle joint, resp.