Iterative Learning of Dynamic Full-Body Motions Anchored by Joint Trajectories

Sang-Ho Hyon

***Abstract*— We propose a new supervised learning and synthesis framework for fast and complex motor tasks. Wherein, a statics-based task-space controller acts not only as a full-body motion control module, but also as a module to generate synergetic joint patterns. The generated joint patterns are encoded into the parameters of phase trajectories of attractors and form the synergy of the task. Similar, but faster motions are synthesized by superposing the task-space controller output and the trajectory attractor output with the modified parameters, while learning dynamics and stiffness according to the task error. We demonstrate the proposed framework by simulating a balanced fast squat on a humanoid robot model.**

I. INTRODUCTION

There have been proposed several methods to compute the joint torques from some desired task space trajectories for real-time motor control application. However, most of the methods require exact dynamics computation, which becomes less realistic for complex humanoid robot. On the other hand, learning or adaptive control achieve dynamic motion control without computing detailed dynamics [1].

For humanoid robot, motor learning on task-space (Cartesian space) would be desirable because, for example, balancing is defined in that space. Recently, Arimoto proposed a passivity-based control framework for redundant robots [2], then extended it to iterative learning control in task space [3]. This work implies that complex redundant robots can achieve task-space motion trajectories without having exact model. The passivity-based methods can effectively solve the redundancy problem by way of the uniquely determined joint torques through Jacobian transpose and the local damping injection to joint space. However, there remains problem how to properly design the local damping according to the complexity and the speed of the task. Also, past studies on task-space control or learning mainly focused on the final performance of the tracking, but did not pay attention to the *generated* joint trajectories, which are too valuable information for redundant robots to be thrown away.

We proposes a online learning algorithm to achieve the dynamic motion tasks from the slow execution recursively by effectively combining the task-space controller and the *experienced* joint trajectories. The idea was first proposed in [4], and now we newly implement two learning algorithms; 1) task-space iterative learning for dynamics learning, 2) joint stiffness adaptation for input optimization.

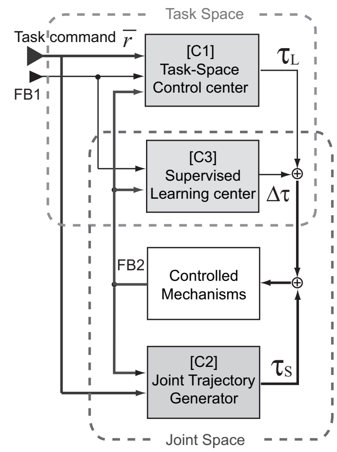


Fig. 1. Overview of the proposed motor learning framework

II. OVERVIEW

Fig. 1 shows the proposed framework. This comprises three modules: **C1** task-space control center, **C2** joint pattern generator, **C3** supervised adaptive/learning center.

C1 computes the necessary joint torques via Jacobian transpose, which are required to follow some position of force trajectories given in the task-space (desired trajectory). This requires forward kinematics based on internal or external sensory feedback, indicated by FB1 and FB2 in the figure. As being static, this controller has difficulty in executing dynamic tasks *when used alone*.

C2 is arranged to the joint space and learns the joint trajectories while the robot is slowly executing tasks. The learned joint trajectories are called reference trajectories in this paper. **C2** also generates attractive force field which pull the joint trajectories to the reference trajectories by specified joint stiffness.

C3 learns feedforward torque to compensate the dynamics, which cannot be treated by **C1**, according to the task-space tracking error. According to which space we focus on, we can use several learning/adaptive control schemes. In this paper, we introduce task-space iterative learning algorithm proposed in [3], because full-body motion tasks with balancing are defined in the task space. **C3** also adaptively tunes the joint stiffness according to the task-space tracking errors.

The author is with JST, ICORP, Computational Brain Project, Saitama, Japan, and National Institute of Information and Communications Technology (NICT), Kyoto, Japan, and ATR Computational Neuroscience Laboratories, Kyoto, Japan. `sangho@atr.jp`

These three components work together to achieve dynamic tasks recursively as follows.

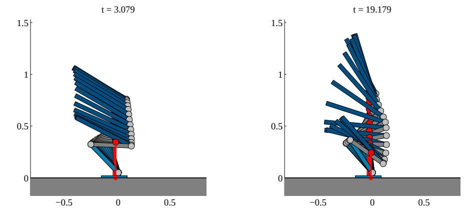
1) **C1** statically achieves desired trajectories with low-frequency.
2) **C2** memorizes the generated joint trajectories during the motion (called reference trajectories), which are parameterized by the phase.
3) Increase the frequency, and superpose **C1** output and **C2** output.
4) **C3** learns the feedforward (dynamic compensation) torque according to the tracking error, while adapting the stiffness around the reference trajectory.
5) **C2** learns the final reference trajectory and the feed-forward torque.

Thus, the proposed method achieves given motion tasks by iterative learning control combined with the joint trajectories with the stiffness adaptation. Rather than computing the necessary feedforward torque directly by the tracking error, we obtain joint trajectory in the static execution of a slow motion task first, then use it as the bootstrap for the faster motion, step by step, which leads to a safe online learning in the presence of internal motions due to the joint redundancy.

III. Example

We test our framework on a simplified model of a human-sized humanoid robot. Fig. 2 shows the result of a high-speed squat. Before the synthesis and learning, the oscillator model has been trained using slow squatting motion; a sinusoidal pattern with the amplitude 0.3 m and the frequency of 0.1 Hz. We put normalized Gaussians on 11 anchor points on the phase coordinate from $-\pi$ to π to learn the trajectory. Then, the synthesis has been performed by setting the target phase frequency to 0.5 Hz, five times faster than the original one. The task-space gain is set to $K_P = 1000, K_D = 300$, and the joint stiffness is set to $K_S = 100$ for all joints. Fig. 3 shows the time series of the simulation. The top two graphs show that the CoM is regulated to zero by the active balancer. The bottom two graphs show the stiffness torque from **C2**, and the feedforward torque from **C3**, respectively. It must be noted that without joint trajectory the robot falls down due to the joint limits. Also, without learning, the task error does not converge.

Currently, we are implementing the learning controller on our humanoid robot. Fig. 4 shows a preliminary experiment.



(a) Before learning (b) After learning

Fig. 2. Simulation of a fast squat. The vertical lines indicate the virtual pendulum connecting CoM and ZMP. The time profile is shown in Fig. 3

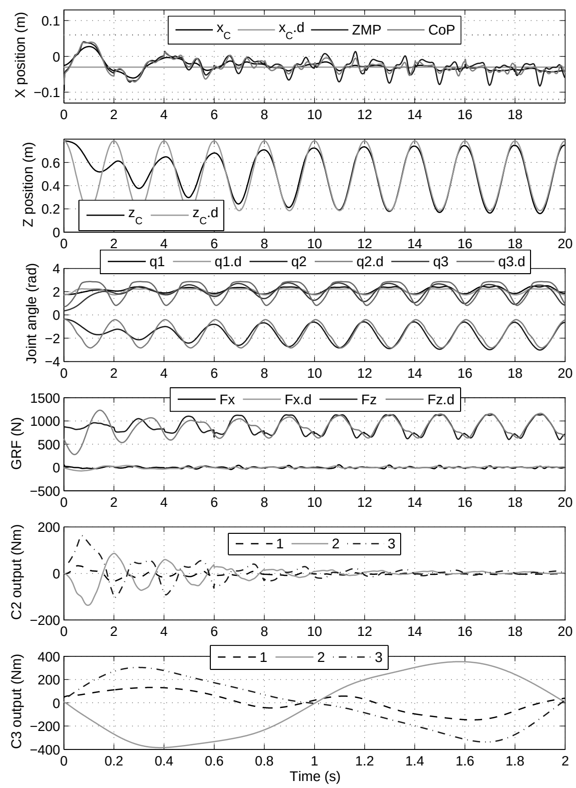


Fig. 3. Simulation data corresponding to Fig. 2. Before learning, the tracking error and the joint stiffness torques are large due to the unmodeled dynamics and small gains. After learning the dynamics, the tracking error becomes small, and the stiffness decreases accordingly.

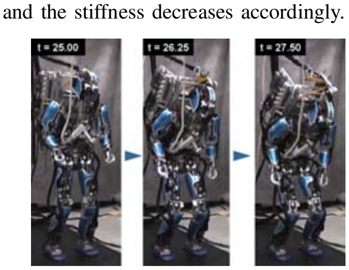


Fig. 4. Periodic squatting motion on a force-controllable humanoid robot

References

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

[2] S. Arimoto, M. Sekimoto, H. Hashiguchi, and R. Ozawa, "Natural resolution of ill-posedness of inverse kinematics for redundant robots: a challenge to bernstein's degrees-of-freedom problem," *Advanced Robotics*, vol. 19, no. 4, pp. 401–434, 2005.

[3] S. Arimoto, M. Sekimoto, and S. Kawamura, "Iterative learning of specified motions in task-space for redundant multi-joint hand-arm robots,," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 2867–2873.

[4] S. Hyon, J. Morimoto, and G. Cheng, "Hierarchical motor learning and synthesis with passivity-based controller and phase oscillator," in *IEEE International Conference on Robotics and Automation*, Pasadena, USA, pp. 2705–2710.