

# Active Stabilization of a Humanoid Robot for Real-Time Imitation of a Human Operator

Seung-Joon Yi<sup>\*†</sup>, Stephen G. McGill<sup>\*</sup>, Byoung-Tak Zhang<sup>†</sup>, Dennis Hong<sup>‡</sup> and Daniel D. Lee<sup>\*</sup>

**Abstract**—Imitating the motion of a human operator is an intuitive and efficient way to make humanoid robots perform complex, human-like behaviors. With the help of recently introduced affordable and real-time depth sensors, the real time imitation of human behavior has become more feasible. However, due to their small footprint and high center of mass, humanoid robots are not inherently stable. The momentum generated by dynamic upper body movements can induce instabilities that are often large enough to make the robot fall down. In this work, we describe a motion controller for a humanoid robot where the upper body is controlled in real time to imitate a human teacher, and the lower body is reactively stabilized based on the current measured state of the robot. Instead of relying on the accuracy of robot dynamics, we use biomechanically motivated push recovery controllers to stabilize the robot against unknown perturbations that include possible impacts. We demonstrate our approach experimentally on a small humanoid robot platform.

**Keywords:** humanoid robot, real-time imitation, biomechanically motivated push recovery

## I. INTRODUCTION

Designing human-like behaviors for humanoid robots with large degrees of freedom is a challenging task, yet making these robots imitate recorded human motions can be an easy and efficient solution. There have been a number of approaches that use captured human motion data to generate corresponding humanoid robot behaviors.

Due to the differences in the dynamics of humans and humanoid robots, full body motion can lead to instability on the robot. In general, achieving stable human motion replication requires a significant amount of offline processing.

One popular technique to allow a humanoid robot to retain its balance includes enforcing ZMP constraints during execution [1], [2], [3]. Researchers have identified the ZMP point of the human, and mapped this information to the robot, in addition to joint angles [4]. However, while ZMP is being controlled in real time, there is always an offline planning component before playing motion back on the robot. In addition to ZMP, angular momentum from limb motion can be controlled in [5].

To capture full body motion in real time application, researchers have used 2D vision images to identify skeletons [6], [7]. Using markers that were placed on the human, the system could identify crucial joints to mimic. However, this



Fig. 1. The DARwIn-OP robot imitating the full body behavior of human operator in real time

requires intrusive changes to a human. With point clouds becoming more prevalent, groups have extracted human figures from this information to mimic skeletons [8]. With a quarter second delay, this work is near real time. However, neither of these approaches seeks to not incorporate walking engines to drive the lower body.

Although the ultimate goal of imitation-based control is the real time imitation of human behavior on a free-walking humanoid robot, most of the previous approaches are confined to either imitation with offline conditioning or externally supported humanoid robot with perfect upper body imitation. There are two problems inhibiting real-time imitation.

The first problem is that tracking human motion required large, stationary motion capture systems, which makes real-time control cumbersome in general. However, the recent introduction of compact depth sensors that can non-invasively track human limbs made imitation a very practical way to control the robots in real time. The second, more pressing, problem is that unsupported humanoid robots are not very stable due to their upright posture. Momentum generated by dynamic movement of limbs then can destabilize the robot if this momentum is not handled properly.

In this work, we focus on the problem of making a humanoid robot imitate human operator's motion in real time while keeping balance. In contrast to offline approaches which record human motion and optimize it for a humanoid

<sup>\*</sup> GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104 {yiseung, smcgill13, ddlee}@seas.upenn.edu

<sup>†</sup> BI Laboratory, Seoul National University, Seoul, Korea. btzhang@ceas.snu.ac.kr

<sup>‡</sup> RoMeLa Laboratory, Virginia Tech, Blacksburg, VA 24061 dhong@vt.edu

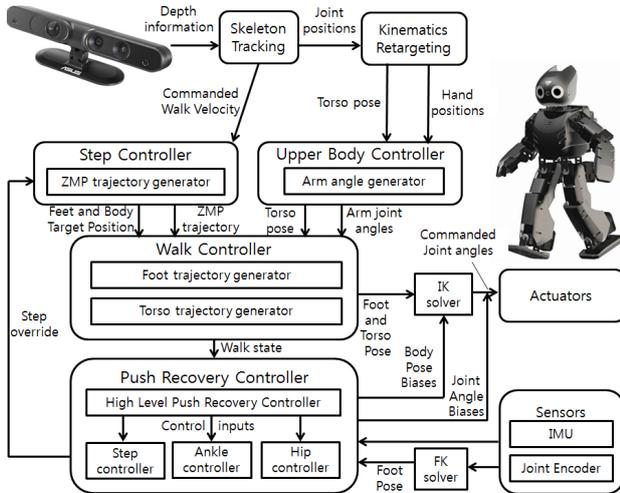


Fig. 2. Overview of the system.

robot to satisfy stability criteria, we use a biomechanically motivated reactive push recovery controller to cope with unknown perturbations from dynamic movement of upper body. We validate our approach using physically realistic simulations, as well as experimentally on the DARwIn-OP small humanoid robot platform. Experimental results show that our method can successfully imitate dynamic human behavior in real time without falling down.

The remainder of the paper proceeds as follows. Section II describes the details of our human imitation controller using a depth sensor. Section III reviews our hierarchical biomechanically motivated push recovery controller using empirically acquired stability boundaries. Section IV shows results using a physics-based simulation. Section V describes and presents experiments on the DARwIn-OP humanoid robot. Finally, we conclude with a discussion of outstanding issues and potential future directions arising from this work.

## II. REAL-TIME IMITATION OF HUMAN MOTION

We use the Asus Xtion RGB-D depth sensor and proprietary software to capture a human's pose at 30 frames per second. Given this skeleton information, we record from the device the coordinates, in meters, of the human's hands, shoulders, elbows, head, and torso. We retarget the coordinated given to desired joint angles for the humanoid robot using inverse kinematics to acquired the end effector offset.

The Asus Xtion RGB-D sensor sends information to a computer external to the robot. Both the robot's onboard computer and the external computer are connected to the same wireless network as the robot. Once the desired joint angles are calculated but the external computer, they are broadcast via UDP packets that are received by the robot's onboard computer. As packets are received, the robot commands its joints, which are PID controlled, to the network-received desired position.

Figure 2 shows the system diagram of how data flows in the system. When evaluating the system in simulation,

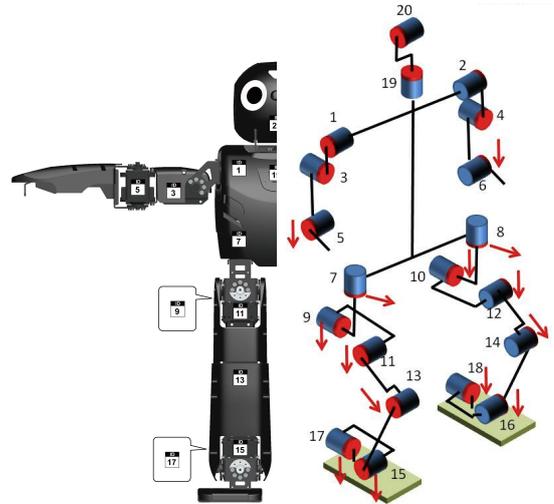


Fig. 3. Left: Joint IDs assigned to the physical DARwIn-OP. Right: Kinematic model of the DARwIn-OP

the joint angles from the inverse kinematics calculation are stored directly into the buffer.

### A. Kinematic Retargeting

Human degrees of freedom are much larger than that of small scale humanoids. Figure 3 shows the kinematic model of the DARwIn-OP, as compared to a human. Since the kinematics of the small size humanoid cannot capture the full range of motion a human operator, the offset of the human's hand from the shoulder is scaled to an offset from the humanoid's shoulder to hand size. This scale is defined as the ratio of the sum of the upper arm and lower arm of the robot to the sum of the arm lengths of the human.

The angle between the upper arm and lower arm is calculated directly from the upper and lower arm positions so as to allow the robot's end effector to reach the correct distance to the desired coordinate. This angle is commanded via joint ID 5, and so the equations that follow will include subscripts for angles to indicate which servo motor is controlled.

$$\theta_5 = \cos^{-1} \left( \frac{\vec{a}_u \cdot \vec{a}_l}{|\vec{a}_u| |\vec{a}_l|} \right) \quad (1)$$

Next, we find the joint angles for the shoulder motors that will move the end effector into place. First, the elevation away from the saggittal plane of the robot.

$$\theta_3 = \cos^{-1} \left( \frac{y}{c} \right) \quad (2)$$

Lastly, the rotation in the saggittal plane is considered. This angle is, however, coupled to the previous calculations.

$$\theta_1 = \tan^{-1} \left( \frac{z}{x} \right) - \theta_5 \quad (3)$$

### B. Torso Rotation

In addition to mimicking the arm positions, we are also mimicking the rotation of the human torso. We determine the body rotation by forming a rotation matrix from the skeleton information. We assume that the vector from shoulder to shoulder ( $\vec{u}$ ) and the vector from waist to head ( $\vec{v}$ ) are

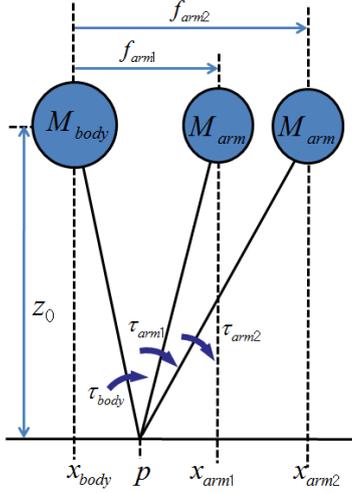


Fig. 4. The extended LIPM with two additional arm masses.

perpendicular. By normalizing these vectors, we can form a three dimensional orthonormal basis by including the cross product of the two vectors ( $\vec{w}$ ). However, in practice, these basis vectors are not quite orthogonal, and thus there cannot be a rotation matrix. We find the nearest rotation matrix ( $R$ ) that describes the human torso rotation in the following way, as described in [9].

$$M = [\vec{u}|\vec{v}|\vec{w}] \quad (4)$$

$$R = M(M^T M)^{-1/2} \quad (5)$$

where  $u$ ,  $v$ , and  $w$  are the basis vectors described above.  $R$  is then a proper rotation matrix that is able to capture the rotation of the human torso. From the rotation matrix, we extract the desired roll ( $r$ ), pitch ( $p$ ), and yaw ( $y$ ) for use in the walk controller.

$$r = \tan^{-1}(R_{32}/R_{33}) \quad (6)$$

$$y = \tan^{-1}(R_{21}/R_{11}) \quad (7)$$

$$p = \tan^{-1}(-R_{31}/(R_{11} \cos^{-1} y + R_{21} \sin^{-1} y)) \quad (8)$$

### III. THE SIMPLIFIED DYNAMICS MODEL

To model the dynamics of the robot imitating human behavior in real time, we use an extended linear inverted pendulum model (LIPM) with two more point masses as in figure 2. This model has center of mass (COM) height  $z_0$ , torso mass  $M_{body}$ , arm mass  $M_{arm}$ , and horizontal COM positions of torso and two arms from support point are denoted by  $x_{body}$ ,  $x_{arm_1}$  and  $x_{arm_2}$ . As we assume LIPM for each mass, the torques at the point  $p$  due to each mass are

$$\tau_{body} = M_{body} (\omega^2 (x_{body} - p) - \ddot{x}_{body}) \quad (9)$$

$$\tau_{arm_1} = M_{arm} (\omega^2 (x_{arm_1} - p) - \ddot{x}_{arm_1}) \quad (10)$$

$$\tau_{arm_2} = M_{arm} (\omega^2 (x_{arm_2} - p) - \ddot{x}_{arm_2}) \quad (11)$$

where  $\omega = \sqrt{\frac{g}{z_0}}$  and  $g$  is the gravitational constant.

For simplicity, we assume that the hands occupy the same height,  $z_0$ . This assumption is flawed if, for instance, up

and down arm motions are executed by the robot. However, these motions have been subjectively observed to be minor disturbances.

If we denote the total robot mass as  $M = M_{body} + 2M_{arm}$ , then they should satisfy the following equation, from work in [14], to make net moment zero at point  $p$ .

$$x_i(\phi) = \begin{cases} p_i(\phi) + a_i^p e^{\phi/\phi_{ZMP}} + a_i^n e^{-\phi/\phi_{ZMP}} \\ \quad + m_i t_{ZMP} \left( \frac{\phi - \phi_1}{\phi_{ZMP}} - \sinh \frac{\phi - \phi_1}{\phi_{ZMP}} \right) & 0 \leq \phi < \phi_1 \\ p_i(\phi) + a_i^p e^{\phi/\phi_{ZMP}} + a_i^n e^{-\phi/\phi_{ZMP}} & \phi_1 \leq \phi < \phi_2 \\ p_i(\phi) + a_i^p e^{\phi/\phi_{ZMP}} + a_i^n e^{-\phi/\phi_{ZMP}} \\ \quad + n_i t_{ZMP} \left( \frac{\phi - \phi_2}{\phi_{ZMP}} - \sinh \frac{\phi - \phi_2}{\phi_{ZMP}} \right) & \phi_2 \leq \phi < 1 \end{cases} \quad (12)$$

$$M\omega^2 p = M_{body}(\omega^2 x_{body} - \ddot{x}_{body}) + M_{arm}(\omega^2 x_{arm_1} - \ddot{x}_{arm_1}) + M_{arm}(\omega^2 x_{arm_2} - \ddot{x}_{arm_2}), \quad (13)$$

which ensures the dynamic stability of the robot if the position of  $p$  lies inside the support polygon during the motion. As we assume that the robot imitates human motion in real time, arm positions  $x_{arm_1}$  and  $x_{arm_2}$  are externally given. Then we define both arms as linear joints

$$f_{arm_1} = x_{arm_1} - x_{body} \quad (14)$$

$$f_{arm_2} = x_{arm_2} - x_{body}, \quad (15)$$

to update the torso trajectory using (3)

$$\ddot{x}_{body} = \omega^2 (x_{body} - p) + M_{arm}(\omega^2 f_{arm_1} - \ddot{f}_{arm_1})/M + M_{arm}(\omega^2 f_{arm_2} - \ddot{f}_{arm_2})/M. \quad (16)$$

### IV. THE HIERARCHICAL PUSH RECOVERY CONTROLLER

Biomechanical studies show that humans display three distinctive motion patterns in response to sudden external perturbations, which we denote as ankle, hip and step push recovery strategies [10] and are shown in figure 5. The ankle strategy applies a control torque at the ankle joint, the hip strategy uses the angular acceleration of torso to apply a counteractive ground reaction force, and finally the step strategy changes the base of support to a new position by stepping. In this section, we review three push recovery controllers based on those strategies using an abstract model of the robot, and provide selection criteria based on current state and the stability region of each controller.

#### A. Ankle push recovery

The ankle strategy applies a control torque on the ankle joints to keep the center of mass within the base of support. We can assume the abstract model in figure 4 (a), where ankle torque  $\tau_{ankle}$  is applied to a LIPM with mass  $M$ , COM height  $z_0$  and COM horizontal position  $x$  from current support point. Then the resulting linearized dynamic model is

$$\ddot{x} = \omega^2 (x - \tau_{ankle}/Mg), \quad (17)$$

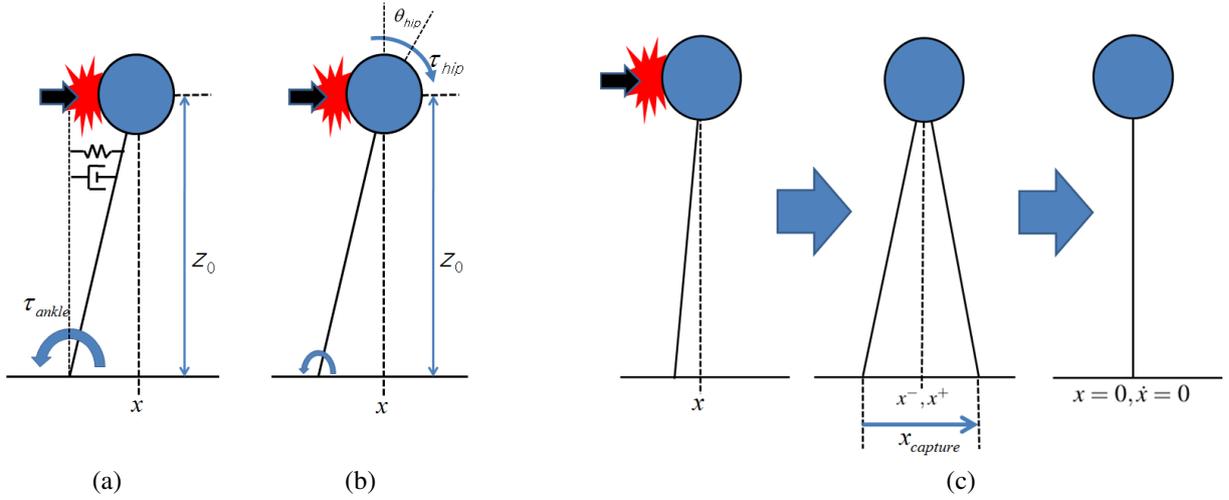


Fig. 5. Three push recovery strategies. (a) Ankle strategy that applies a control torque at the ankle joint. (b) Hip strategy which uses the angular acceleration of the torso and limbs to apply counteractive ground reaction forces. (c) Step strategy that changes the support point by stepping.

which can be controlled by a PD-control on  $x$  with control gains  $K_p$  and  $K_d$ .

$$\tau_{ankle} = K_p x + K_d \dot{x}, \quad (18)$$

### B. Hip push recovery

The hip strategy uses angular acceleration of the torso and limbs to generate a backward ground reaction force (GRF) to pull the center of mass back towards the base of support. Abstract model in figure 4 (b) includes a flywheel with point mass at height  $z_0$  and rotational inertia  $I$ , and control torque  $\tau_{hip}$  at the COM. Then the resulting linearized dynamic model is

$$\ddot{x} = \omega^2 (x - \tau_{hip}/Mg) \quad (19)$$

$$\ddot{\theta}_{hip} = \tau_{hip}/I. \quad (20)$$

However we should stop the flywheel from exceeding joint limits. In this case, following the bang-bang profile [11] can be used for applying hip torque to maximize the effect while satisfying the joint angle constraint

$$\tau_{hip}(t) = \begin{cases} \tau_{hip}^{MAX} & 0 \leq t < T_{H1} \\ -\tau_{hip}^{MAX} & T_{H1} \leq t < 2T_{H1}, \end{cases} \quad (21)$$

where  $\tau_{hip}^{MAX}$  is the maximum torque that can be applied on torso and  $T_{H1}$  is the time the torso stops accelerating.

### C. Step push recovery

The step strategy moves the base of support towards the direction of push by taking a step, as shown in figure 4 (c). If we assume the support point transition occurs instantly preserving the linear momentum, we can get the following landing position from the initial support point [11]:

$$x_{capture} = \dot{x}/\omega + x. \quad (22)$$

### D. High-level push recovery controller

When pushed, humans perform a combination of push recovery behaviors according to the particular situation. To select the appropriate set of push recovery behaviors as humans do, we use a hierarchical controller where ankle, hip and step push recovery controllers work as low-level subcontrollers and the high-level push recovery controller triggers each according to the current sensory input [12].

For the simplified models shown in Figure 4, previous analysis have shown the decision boundaries of each controller based on the current state [13]. If we assume maximum ankle torque as  $\tau_{MAX}^{ankle}$ , then the stability region for ankle push recovery controller is derived as

$$|Mg(\dot{x}/\omega + x)| < \tau_{ankle}^{MAX} \quad (23)$$

which is increased by combining the hip strategy plus ankle strategy

$$|Mg(\dot{x}/\omega + x)| < \tau_{ankle}^{MAX} + \tau_{hip}^{MAX} (e^{\omega T_{H1}} - 1)^2. \quad (24)$$

Finally, if we assume instantaneous support point transition without loss of linear momentum, we have the following stability region for using all three strategies at once:

$$|Mg(\dot{x}/\omega + x)| < \tau_{ankle}^{MAX} + \tau_{hip}^{MAX} (e^{\omega T_{H1}} - 1)^2 + Mg x_{capture}^{MAX}, \quad (25)$$

where  $x_{capture}^{MAX}$  is the maximum step size available. In this case we can use two boundary conditions in (14) and (15) to select between controllers based on current state. Phase space trajectory plots and stability regions for each controller are shown in Figure 5. For the more realistic case with a multi-segmented body with motor dynamics as on a physical robot, we can use an empirical stability boundaries trained from experience [12], [14].

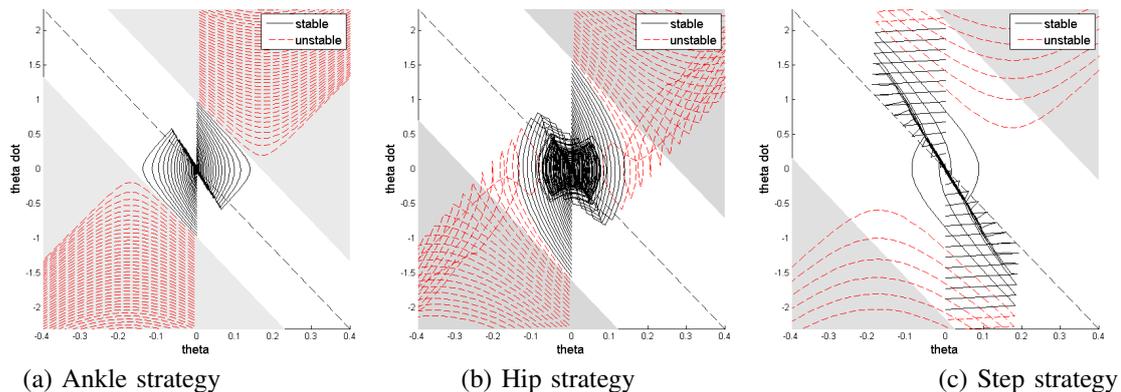


Fig. 6. Stability regions for each push recovery controller. White and gray region denotes stable and unstable region of state space. Black and red lines denote stable and unstable state trajectories from various initial states.

## V. EXPERIMENTAL RESULTS

### A. Hardware setup

We use the DARwIn-OP commercial humanoid robot to validate our approach experimentally. The DARwIn-OP robot is 45cm tall, weighs 2.8kg, and has a 3-axis accelerometer and a 3-axis gyroscope for inertial sensing. It has position-controlled Dynamixel servos for actuators, which are controlled by a custom microcontroller connected to an Intel Atom-based embedded PC at a control frequency of 100Hz.

### B. Real-time imitation performance

Figure 7 shows the DARwIn-OP robot imitating the motion of human operator. We have found that the robot can convincingly imitate a number of human motions using its arms and torso movements, even if it has much lower degree of freedom compared to human. The overall latency of the system was below 0.05 s, and the robot motion was only limited by the velocity limit of the servomotors.

### C. Effect of active stabilization controller

To quantitatively test the effectiveness of our active stabilization controller, we generate data logs from human motion and play these logs on the robot with and without active stabilization controller. To better compare the effect of stabilization, we let the robot walk in place during motion, which makes the robot more unstable than standing still. The momentum generated by the robot's limb movements are large enough to make the robot fall down without push recovery controller.

Table 1 shows the result of the trials. We found that the active controller clearly improved the performance for 4 of the 5 motions we tested, except for one motion which was stable without active stabilization.

## VI. CONCLUSIONS

We propose a system that uses RGB-D camera to track the motion of human operator to make an untethered humanoid robot imitate that motion in real time. To handle the perturbations caused by unforeseen upper body movements, we use biomechanically motivated hierarchical push

Motion	With Stabilization	Without Stabilization
Swing	3/3	0/3
Stir	2/3	0/3
Conductor	2/2	2/2
Jab	2/3	1/3
Uppercut	3/3	0/3

TABLE I  
NUMBER OF SUCCESSFUL TRIALS FOR IMITATING DIFFERENT UPPER BODY MOTIONS.

recovery controllers to stabilize the robot. Our approach is implemented and demonstrated using a DARwIn-OP small humanoid robot, and the experimental results show that our system can make the robot imitate the human motion in real time, and our active stabilization controller help the robot stabilize itself while doing upper body motion, even during walking.

Possible future work includes extending our algorithm to handle more dynamic full-body motions, as well as implementation of these algorithms on full-size humanoid robots.

## ACKNOWLEDGMENTS

We acknowledge the support of the NSF PIRE program under contract OISE-0730206 and ONR SAFFIR program under contract N00014-11-1-0074. This work was also partially supported by the NRF grant of MEST (2012-0005643), as well as the BK21-IT program funded by MEST.

## REFERENCES

- [1] S. Nakaoka, A. Nakazawa, F. Kanehiro, K. Kaneko, M. Morisawa, and K. Ikeuchi, "Task model of lower body motion for a biped humanoid robot to imitate human," in *Dances, IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.
- [2] L. Boutin, A. Eon, S. Zeghloul, and P. Lacouture, "From human motion capture to humanoid locomotion imitation application to the robots hrp-2 and hoap-3," *Robotica*, vol. 29, no. 2, pp. 325–334. [Online]. Available: <http://dx.doi.org/10.1017/S0263574710000172>
- [3] B. Dariush, M. Gienger, B. Jian, C. Goerick, and K. Fujimura, "Whole body humanoid control from human motion descriptors," in *ICRA*, 2008, pp. 2677–2684.
- [4] S. Kim, C. Kim, B. You, and S. Oh, "Stable whole-body motion generation for humanoid robots to imitate human motions," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, oct. 2009, pp. 2518–2524.



Fig. 7. The real-time imitation of various full-body human motions by the DARwIn-OP robot.

- [5] N. Naksuk, C. Lee, and S. Rietdyk, "Whole-body human-to-humanoid motion transfer," in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, dec. 2005, pp. 104–109.
- [6] M. Riley, A. Ude, K. Wade, and C. G. Atkeson, "Enabling real-time full body imitation: A natural way of transferring human movements to humanoids," in *in: Proceedings of the IEEE International Conference on Robotics and Automation*, 2003, pp. 2368–2374.
- [7] K. Yamane, S. O. Anderson, and J. K. Hodgins, "Controlling humanoid robots with human motion data: Experimental validation," in *Humanoids*, 2010, pp. 504–510.
- [8] B. Dariush, M. Gienger, A. Arumbakkam, Y. Zhu, B. Jian, K. Fujimura, and C. Goerick, "Online transfer of human motion to humanoids," *I. J. Humanoid Robotics*, vol. 6, no. 2, pp. 265–289, 2009.
- [9] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *J. Opt. Soc. Am. A*, vol. 5, no. 7, pp. 1127–1135, Jul 1988. [Online]. Available: <http://josaa.osa.org/abstract.cfm?URI=josaa-5-7-1127>
- [10] A. G. Hofmann, "Robust execution of bipedal walking tasks from biomechanical principles," Ph.D. dissertation, Cambridge, MA, USA, 2006.
- [11] J. Pratt, J. Carff, and S. Drakunov, "Capture point: A step toward humanoid push recovery," in *in 6th IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 200–207.
- [12] S.-J. Yi, B.-T. Zhang, D. Hong, and D. D. Lee, "Learning full body push recovery control for small humanoid robots," in *ICRA*, 2011.
- [13] B. Stephens, "Humanoid push recovery," in *Proceedings of the IEEE RAS International Conference on Humanoid Robots*, 2007.
- [14] S.-J. Yi, B.-T. Zhang, D. Hong, and D. D. Lee, "Online learning of a full body push recovery controller for omnidirectional walking," in *Proceedings of the IEEE RAS International Conference on Humanoid Robots*, 2011, pp. 1–6.