Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

Natural Motion Generation for Humanoid Robots

Kensuke Harada*†, Kris Hauser*, Tim Bretl*, and Jean-Claude Latombe*
*Computer Science Department, Stanford University
Stanford, CA 94305, USA
†National Institute of Advanced Industrial Science and Technology(AIST)
1-1-1 Umezono, Tsukuba 305-8568, JAPAN

***Abstract*— This paper presents a method of generating natural-looking motion primitives for humanoid robots. An optimization-based approach is used to generate these primitives, but the objective function is tailored to each one and complexity is reduced by identifying relevant degrees of freedom. Several examples are shown in simulation: for an arm movement to reach an object, it is better to minimize the acceleration of key parts of the robot over its entire trajectory; for a single step on flat ground, it is better to minimize the torque and instantaneous angular momentum at every posture. The primitives are precomputed off-line, but might be used by on-line planner either to provide a fixed set of maneuvers or to bias a probabilistic, sample-based search for motions.**

I. Introduction

Because humanoid robots are designed to interact closely with humans, it is important that their motion look natural. However, generating natural-looking humanoid motion from scratch is hard. One reason is that we have no clear idea what "natural" actually means. Intuitively, we might want the robot to minimize erratic arm and leg movements or to move with a consistent style, but it is difficult to quantify this intuition.

A standard approach in computer animation bypasses this issue by stitching together pieces of captured human motion to generate the motion of digital actors [3], [4], [5], [6]. This approach has also been applied to humanoid robots [1], [2]. But due to differences in governing kinematics and dynamics, what makes the motion of a digital actor look human-like may not make the motion of a humanoid look natural, or even feasible.

An alternative approach generates motion that is optimal with respect to a particular objective function [7], [8], [9], [10]. Rather than copy human motion, this approach uses the objective function to capture the qualities of human motion that make it look natural. However, it is difficult to define an objective function that works well in general. Further, because a humanoid has many degrees of freedom (DOF), it is difficult to solve the resulting optimization problem quickly.

In this paper, we also take an optimization-based approach. But rather than try (on-line) to make arbitrary motion look natural, we focus instead on precomputing (off-line) a library of specific motion primitives. These primitives might include a single step on flat ground, an arm movement that places a hand on a wall for balance, or a bending motion to pick an object up off the ground. We tailor our objective function to generate each primitive and reduce the complexity of the resulting optimization by identifying degrees of freedom that are relevant to the task at hand. In particular, we will discuss an example for which it is better to minimize the acceleration of key parts of the robot over an entire trajectory as well as an example for which it is better to minimize the torque and instantaneous angular momentum at every posture.

Our library of humanoid motion primitives could be used by several existing on-line planners and controllers. For example, it might define a fixed set of feasible steps for the planner of [20] or a fixed set of postural objectives for the controller of [16]. However, we intend our library to be used in particular by a planner developed for the humanoid HRP-2 (Fig. 1) on varied terrain [17] (based on earlier work for a free-climbing robot [19]). This planner first chooses a candidate sequence of steps (potential placements of hands or feet on the terrain), then generates the motion to take each step with a probabilistic, sample-based algorithm [21]. It works well on irregular and steep terrain, but on flat terrain may generate needless motions of the arms or other DOF that are not required for balance. We have used our library to address this limitation – instead of sampling across all of configuration space to plan each step, we sample in a growing distribution around the nominal path (a joint-angle trajectory) defined by a chosen primitive. For example, Fig 1 shows the humanoid HRP-2 climbing a ladder, a motion computed using the planner of [17] combined with a library of four primitives. This motion looks very natural – many of the unnecessary arm and leg movements shown in [17] have been eliminated.

This paper is organized as follows; In Section 3, we introduce the change of coordinates to formulate the equation of motion in the operational space. Due to the physical meaning of each variable, we show that we can significantly decrease the number of search variables. We also show that this change of coordinate is effective in dealing with the change of stance foot. In Section 4, we formulate two optimization problems; one is the trajectory optimization where the motion trajectory during a certain period of time is optimized, and the other is the instantaneous posture optimization where the posture of a humanoid robot at each instant of time is optimized. In Section 5, we show that, while the joint torque has been used in the trajectory

1-4244-0259-X/06/$20.00 ©2006 IEEE

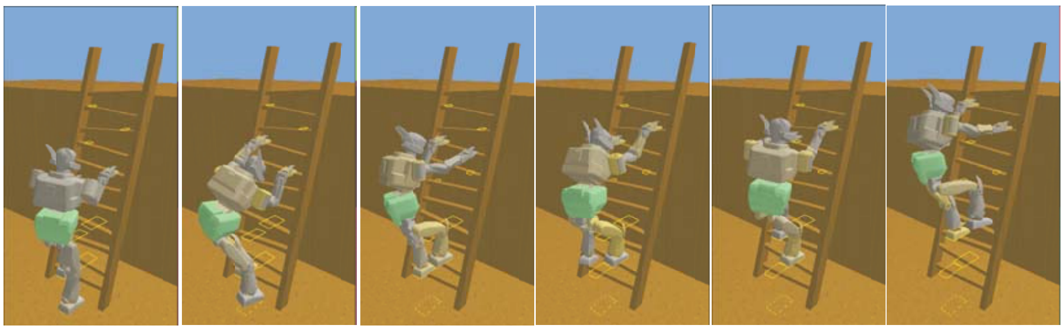


Fig. 1. Four ladder-climbing primitives adapted to a ladder with uneven rungs.

optimization of planar simple manipulators, this index function often fails in generating a natural motion under gravity. On the other hand, we generate the natural-looking reaching motion using the operational space acceleration. We also generate the natural-looking walking and climbing motion by optimizing the joint torque at each instant of time.

II. Related Works

Several humanoid robots have been developed recently, including [12], [13], [14]. The following three approaches have been used to generate natural motion for these robots:

a) Methods based on motion capture: Widely used in computer animation, these methods generate the motion of digital actors by stiching together pieces of captured human motion. Many techniques have been proposed, including one that is interactive [4], one that is constraint-based [5], and one that combines different captured motions for each limb [6]. A similar technique was even used to quantify, with a statistical model, what it means for human motion to be natural [3]. This approach has also been applied to humanoid robots [2], [1], but because of differences in governing kinematics and dynamics, what makes the motion of a digital actor look human-like may not make the motion of a humanoid look natural.

b) Methods based on optimization: Rather than copy human motion, these methods generate motion that is optimal with respect to a particular objective function. A number of objective functions have been proposed, including minimization of hand jerk [7], [9], minimization of the change of joint torque [8], and minimization of joint torque itself [10]. In [10], the gradient of the objective function was computed analytically, with application to a planar model of a humanoid. It has also been observed that angular momentum about the CoG is highly regulated in human walking motion [11], although this quantity has not been used previously as an objective function to generate natural humanoid motion.

c) Methods based on control: These methods use an onboard control strategy to achieve natural motion. One strategy is based on tracking linear and angular momentum [15]. Another is based on achieving a hierarchy of task-level objectives and postural constraints using an operational space forumulation [16]. In fact, our work could be used to define a set of appropriate postural constraints to be used by [16].

III. Formulation

Our goal is to generate "natural-looking" motion primitives for a humanoid robot. Our approach is to generate motion that is optimal with respect to some objective function. In this section we first describe the equations of motion governing a particular humanoid (HRP2), then use these to formulate the optimization problem with respect to several different objective functions, each tailored to one of the motion primitives we will discuss in Section IV.

A. Equation of Motion

We denote the position and orientation of each part of the robot by $\xi = (p \ \phi)$. The elements of $p \in R^3$ and $\phi \in R^3$ are $p = (x \ y \ z)$ and $\phi = (a \ b \ c)$, respectively. We use subscripts to indicate the part of the robot to which p and ϕ refer. The subscripts h, f, b and c denote the hand, the foot, the body and the CoG, respectively. The subscripts r and l denote right and left, respectively. For example, p_{rh} denotes the position of the right hand. Let q be the joint angle vector of the robot. If q is used without any subscripts, it expresses the collection of all joints of the robot.

The motion of the humanoid robot is characterized by the following equation of motion:

$$M \begin{bmatrix} \ddot{\xi}_b \\ \ddot{q} \end{bmatrix} + h + g = \begin{bmatrix} 0 \\ \tau \end{bmatrix} + T_q^T f. \tag{1}$$

In this equation, M, h, g, τ and f denote the mass matrix, the centrifugal and Coriolis force, the gravity force, joint torque and the contact force and moment, respectively. Rather than consider all degrees of freedom, we will focus on the dynamics of key parts of the robot, which we can compute by applying a coordinate transformation to (1). First, the velocity of the hands and the feet can be expressed as

$$\dot{\xi}_i = J_{bi}\dot{\xi}_b + J_{qi}\dot{q}, \quad i = rf, lf, rh, lh. \tag{2}$$

Also, the velocity of the CoG can be expressed as

$$\dot{p}_c = \dot{p}_b + J_c\dot{q}. \tag{3}$$

Using these equations, we define a coordinate transformation

$$\begin{bmatrix} \dot{\eta} \\ 0 \end{bmatrix} = \begin{bmatrix} T_x \\ T_q \end{bmatrix} \begin{bmatrix} \dot{\xi}_b \\ \dot{q} \end{bmatrix}. \tag{4}$$

The exact form of this coordinate transformation depends on the motion primitive, in particular on the contact conditions while performing the primitive. For example, if the robot is using both hands while standing on two legs, then we choose coordinate variables

$$\eta = (p_c \ \phi_b \ \xi_{rh} \ \xi_{lh})$$

that encode the position and orientation of the hands and the position of the CoG, require that $\dot{\xi}_{rf}$ and $\dot{\xi}_{lf}$ are always zero, and define T_x and T_q appropriately. As a second example, consider the robot walking on flat ground with swinging arms. In this case, the contact condition switches between left-foot support, double support, and right-foot support phases. Also, we care about the joint angles of both arms, but not about the position and orientation of the hands. We require that each foot has zero velocity when contacting the floor, and choose coordinate variables

$$\eta = (p_c \ \phi_b \ q_{rh} \ q_{lh} \ \xi_{rf})$$

for the left-foot support phase,

$$\eta = (p_c \ \phi_b \ q_{rh} \ q_{lh})$$

for the double support phase, and

$$\eta = (p_c \ \phi_b \ q_{rh} \ q_{lh} \ \xi_{lf})$$

for the right-foot support phase. By selecting relevant degrees of freedom in this way, we reduce the dimension of the resulting optimization. Applying the coordinate transformation, we obtain the new equations of motion

$$\tilde{M}\ddot{\eta} + \tilde{h} + \tilde{g} = \tilde{D}\tau + \tilde{T}^T f, \tag{5}$$

where

$$\begin{aligned} \tilde{M} &= T_x M U_x, \\ \tilde{h} &= T_x \left\{ h - M T^{-1} \dot{T} U_x \dot{\eta} \right\}, \\ \tilde{g} &= T_x g, \\ \tilde{D} &= T_x \begin{bmatrix} 0 \\ I \end{bmatrix}, \\ T^{-1} &= \begin{bmatrix} U_x & U_q \end{bmatrix}, \\ \tilde{T}^T &= T_x T_q^T. \end{aligned}$$

If we consider the physical meaning of each coordinate, we can further reduce the number of variables. For example, when a robot stands on the floor, the horizontal position of the CoG, x_c and y_c, must lie above a support polygon. To maximize safety, we choose to fix the horizontal position of the CoG at the center of support polygon. Also, since there are some coordinate variables having less effect on the objective function, we do not need to optimize these variables. This further splits η into two parts, η_1 and η_2, where η_1 is fixed before optimization. In the next section, we will also consider how this change of coordinate variables contributes to the change of objective function.

In addition to the position and velocity of various parts of the robot, we will also include the joint torque in our objective function. The joint torque can be expressed as a function of the acceleration of the coordinate variables:

$$\tau = (T_x M^{-1} D)^{+} \left\{ \ddot{\eta} + T_x M^{-1}(h + g - T_q^T f) - \dot{T} U_x \dot{\eta} \right\}, \tag{6}$$

where

$$D = \begin{bmatrix} 0 \\ I \end{bmatrix}.$$

In this equation, the symbol $+$ denotes the pseudo-inverse. We use a constraint-based method to determine the contact force f [23].

B. Optimization Problem

Two different optimization problems will be useful for different motion primitives: trajectory optimization and instantaneous posture optimization. In the trajectory optimization, the entire motion trajectory is considered all at once. In the instantaneous posture optimization, the posture of the robot at each instant of time is considered separately. Although both optimization should be implimented at the same time to achieve the complete optimized motion, we apply them separately and consider the feature of both optimization methods in this paper.

1) Trajectory Optimization: To formulate the trajectory optimization problem, the trajectory of η_2 is expressed as a function of via points $\eta_{20}, \cdots, \eta_{2m}$ (Fig.2). If we write the trajectory as n-th order polynomials with respect to time, the coefficients of polynomial can be determined as a function of these via points.

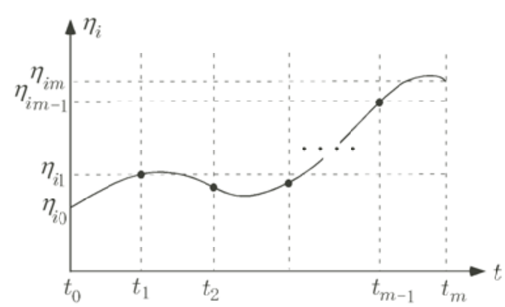


Fig. 2. Trajectory expressed by via points

Given the initial and final values of coordinate variables, the trajectory optimization problem is turned into a nonlinear optimization of via points. To minimize torque, we use the objective function

$$J = \int_0^T \tau^T \tau dt. \tag{7}$$

To minimize the acceleration of parts of the robot's body, we use the objective function

$$J = \int_0^T \ddot{\eta}_2^T \ddot{\eta}_2 dt. \tag{8}$$

To minimize joint acceleration, we use the objective function

$$J = \int_0^T (\ddot{\xi}_b^T \ddot{\xi}_b + \ddot{q}^T \ddot{q}) dt. \tag{9}$$

2) Instantaneous Posture Optimization: To formulate the instantaneous posture optimization problem, we also use via points. But rather than consider the entire trajectory at once, we optimize the robot's posture at each via point. To minimize torque, we use the objective function

$$J = \tau(t)^T \tau(t) \tag{10}$$
$$t = t_0 \cdots t_m.$$

It has been observed that, during human walking motion, the angular momentum about the CoG is regulated [11]. So we also consider minimizing the angular momentum about the CoG, using the objective function

$$J = \mathcal{L}(t)^T \mathcal{L}(t) \tag{11}$$
$$t = t_0 \cdots t_m$$

where $\mathcal{L}$ denotes the angular momentum about the CoG of the robot (defined as the finite difference with respect to neighboring via points).

IV. Simulation

A. Trajectory Optimization

Through the simulation study, we often focus on the optimization of joint torque. If the torque optimization is not effective, we then consider which objective function should be used.

1) 1 DOF Examples: First, we consider simple 1 DOF examples to show that there is a case where conventional joint torque optimization fails in generating natural motion under a gravitational field. In particular, consider the two examples shown in Fig. 3. In both cases, we used (7) as the objective function and used the MATLAB Riots95 toolbox to compute the optimal solution. As shown in Fig. 4, the planar 1-DOF manipulator generates natural motion (moving straight toward the goal) by using the torque minimization criteria. However, under a gravitational field (Fig. 5), the manipulator does not move straight toward the goal – this is because the torque is minimzed when the arm is straight down.

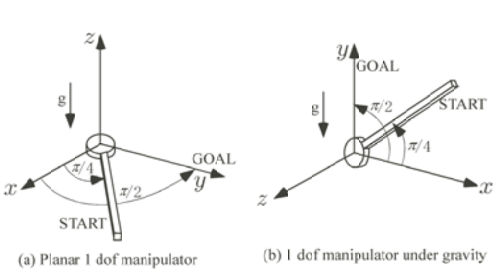


Fig. 3. 1 DOF examples with/without gravity

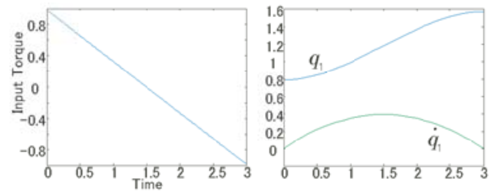


Fig. 4. Result of numerical calculation without gravity

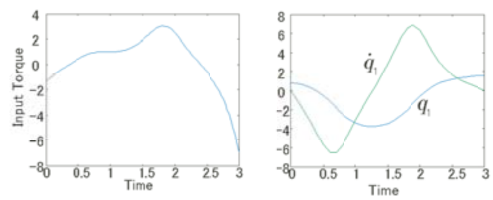


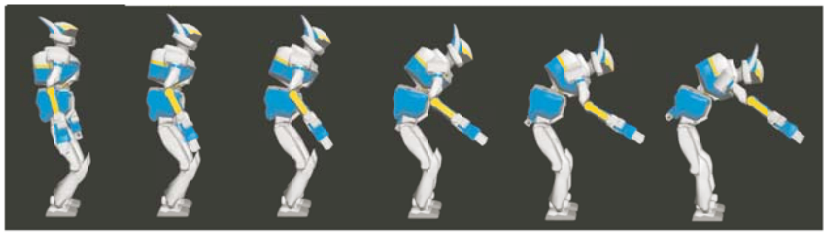
Fig. 5. Result of numerical calculation with gravity

2) Reaching motion of humanoid robot model: Next, we consider the humanoid HRP2 [14] performing a reaching motion. This robot was developed by AIST and Kawada Industries. It has 6 DOF arms, 1 DOF fingers, a 2 DOF head, and a 2 DOF body. Although the total number of joints are 30, we do not consider the DOF of the finger and the head. Since two feet contact the floor, we further reduce the number of DOF to 12. So, the total dimension of η becomes 20. Further, we fix the horizontal position of the CoG at the center of the support polygon (the shape of which is approximated as rectangular) and consider neither the motion of the left arm nor the orientation of the right hand. So η can be split into the variables

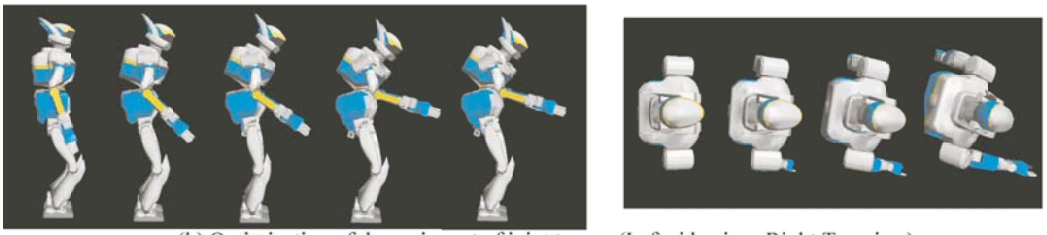
$$\eta_1 = (x_g \; y_g \; \xi_{lh} \; \phi_{rh}) \tag{12}$$
$$\eta_2 = (z_g \; \phi_b \; q_b \; p_{rh}), \tag{13}$$

where we only optimize over η_2. We use five via points for $(z_g \;\; \phi_b \;\; q_b)$, so the trajectory optimization problem is a nonlinear programming problem with a total of 33 variables. We computed the solution using the CFSQP toolbox, which implements the standard SQP method for nonlinear programming. It took about 24 hours to calculate the optimal trajectory on a 2GHz PC.

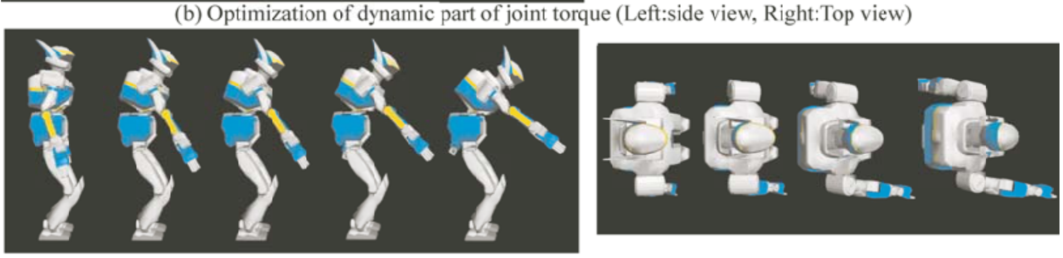
The results of simulation are shown in Fig. 6. Fig. 6(a) shows the motion of the robot used as input (generated by hand). We set the target hand position far from the robot. To reach this target position, the robot stretches its arms and legs. Fig. 6(b) shows the result of gravity-compensated torque minimization. But as in the 1-DOF example, this approach introduces unnatural motion (the robot twists its chest to counter the reaction torque from the arm). Fig. 6(c) shows the result of minimizing $\ddot{\eta}$ by using (8). In this case, the robot stretches its arm to reduce the unnecessary twisting motion (results were not as good when minimizing joint acceleration, given by (9)). So in this case, minimizing the operational space acceleration results in the most natural motion.



(a) Input motion designed by hand



(b) Optimization of dynamic part of joint torque (Left:side view, Right:Top view)



(c) Optimization of operational space acceleration (Left:side view, Right:Top view)

Fig. 6. Trajectory optimization of reaching motion

B. Instantaneous Posture Optimization

In this section, we optimize the walking motion of a humanoid robot. But rather than consider the entire trajectory at once, we consider the posture at each instant of time separately.

To decide which coordinate variables to optimize, we looked at the contribution of each coordinate variable to the change in objective function at the robot's initial posture, shown in Table I (as computed by the Jacobian matrix). From this table, we conclude that the vertical position of the body most affects the change of torque. This result is not surprising, since the torque in each leg is minimized when it is straight (that is, when the robot has an upright posture). Likewise, we see that the inclination of the body affects the angular momentum of the robot. This result also makes sense, since if the robot bends over, we expect the angular momentum about the CoG to increase. Using both results as a heuristic, we choose to minimize torque by considering only z_c and to minimize angular momentum by considering only a_b, b_b, q_{b1}, q_{rh1}, q_{rh4}, q_{lh1} and q_{lh4}.

Results are shown in Fig. 7. With 70 via points, the calculation of each step took about 10 minutes on a 2GHz PC. Since we minimize joint torque, the robot tends to stretch its knees while walking (similar to a human). Since we minmize angular momentum, the robot swings its body and its arms about the CoG (also similar to a human). These results are much better than if we had used the same objective function as for the reaching motion example.

We have computed a number of other motion primitives as well. For example, Fig. 8 shows a climbing motion of the robot (similar to the primitive we used to plan the ladder-climb in Section I). We used the objective function (11) and chose the following variables: the x and z coordinates of the CoG, the inclination of the body, and the angle of the waist. In this simulation, we also modeled the effect of static friction at each contact point. Notice that, as before, the robot tends to stretch its arms and legs to reduce joint torque.

V. Conclusions

In this paper, we presented a method of generating natural-looking motion primitives for humanoid robots. We used an optimization-based approach, tailoring our objective function (joint torque, acceleration, or angular momentum), our coordinate variables (position of the CoG, position and orientation of hands and feet, or body inclination), and our formulation (considering the entire trajectory at once or the posture at each instant of time separately) to suit each motion primitive. We showed several example results in simulation, and briefly described how our library of primitives could be used to help plan longer motions. Future work might focus on reducing the amount of user interaction required, on decreasing computation time, or on the design of new planning and control algorithms that take better advantage of motion primitives.

Acknowledgments

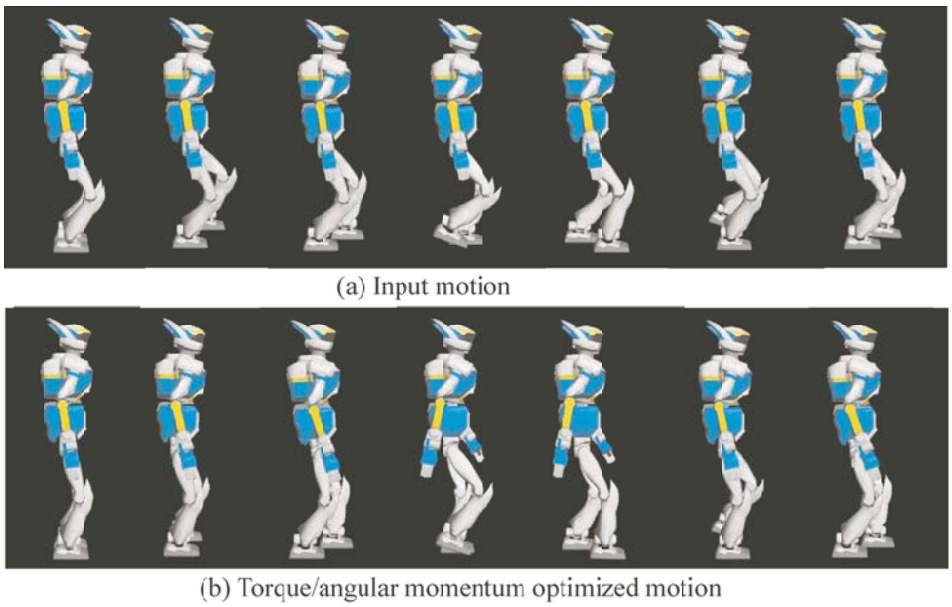
The first author would like to express his sincere gratitude to Dr. Shigeoki Hirai, Dr. Hirohisa Hirukawa, Dr. Shuuji Kajita, Dr. Kenji Kaneko, Dr. Fumio Kanehiro, Mr.

TABLE I

CONTRIBUTION OF COORDINATE VARIABLE TO INDEX FUNCTIONS

	z_c	a_b	b_b	c_b	q_{b1}	q_{b2}	q_{rh1}	q_{rh2}	q_{rh3}	q_{rh4}	q_{rh5}	q_{rh6}
$\tau^T\tau$	-14387.2	-0.013	3191.2	3.44	0.0	346.643	1057.2	-34.07	-4.07	-82.81	-1.43	1.20
$\mathcal{L}_x$	0.1335	1.3706	0.0824	0.0754	0.3695	0.0049	-0.1103	-0.4035	-0.0382	0.0220	0.0	0.0126
$\mathcal{L}_y$	-0.0496	-0.0123	1.1921	0.8479	0.8479	1.3368	-0.4204	-0.0301	0.0080	-0.1160	-0.0011	-0.0006
$\mathcal{L}_z$	0.0735	0.4109	0.4581	0.7831	0.8208	0.1906	-0.4873	-0.0894	-0.0011	-0.1608	-0.0020	0.0015

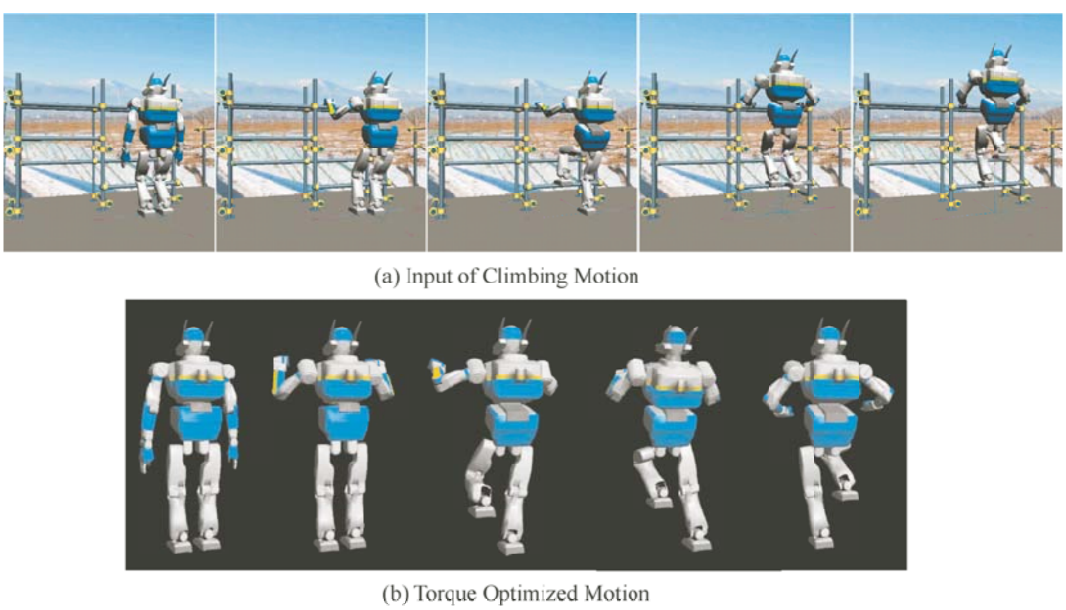
q_{b1}:Body yaw, q_{b2}:Body pitch, q_{rh1}:Shoulder pitch, q_{rh2}:Shoulder roll, q_{rh3}:Shoulder yaw, q_{rh4}:Elbow, q_{rh5}:Wrist yaw, q_{rh6}:Wrist pitch



(a) Input motion

(b) Torque/angular momentum optimized motion

Fig. 7. Instantaneous optimization of walking motion



(a) Input of Climbing Motion

(b) Torque Optimized Motion

Fig. 8. Instantaneous optimization of climbing motion

Kiyoshi Fujiwara, and Dr. Mitsuharu Morisawa at AIST and to Dr. Mikhail Svinin at RIKEN for their helpful discussions.

REFERENCES

[1] S. Nakaoka, A. Nakazawa, K. Yokoi, H. Hirukawa, and K. Ikeuchi: "*Generating Whole Body Motions for a Biped Humanoid Robot from*

Captured Human Dances", Proc. of 2003 IEEE Int. Conf. on Robotics and Automation, 2003.

[2] N. S. Pollard, J. K. Hodgins, M. J. Riley, and C. G. Atkeson: "*Adapting human motion for the control of a humanoid robot*", Proc. of 2002 IEEE Int. Conf. on Robotics and Automation, 2002.

[3] L. Ren, A. Patrick, A.A. Efros, J.K. Hodgins, and J.Rehg: "*A Data-Driven Approach to Quantifying Natural Human Motion*", ACM Transactions on Graphics, Vol. 24, No. 3, 2005.

[4] O. Arikan and D.A. Forsyth: "*Interactive Motion Generation from Examples*", ACM Transactions on Graphics, Vol. 21, No. 3, 2002.

[5] M. Gleicher: "*Comparing Constraint-based Motion Editing Methods*", Graphical Models, vol. 63, no. 2, pp. 107-134, 2001.

[6] L. Ikemoto and D.A. Forsyth: "*Enriching a Motion Collection by Transplanning Limbs*", Proc. of 2004 ACM SIGGRAPH/Eurographics Symp. on Computer Animation, pp. 99-108, 2004.

[7] T. Flash and N. Hogan: "*The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model*", J. Neuroscience, vol. 5, no. 7, pp. 1688-1703, 1985.

[8] Y.Uno, M.Kawato, and R.Suzuki: "*Formation and Control of Optimal Trajectory in Human Multijoint Arm Movement Minimum Torque-Change Model*", Biological Cybernetics, no. 61, pp. 89-101, 1989.

[9] M.Svinin, Y.Masui, Z.-W.Luo, and S.Hosoe: "*On the Dynamic Version of the Minimum Hand Jerk Criterion*", J. of Robotic Systems, vol. 22, no. 11, pp. 661-676, 2005.

[10] J.E.Bobrow, B.Martin, G.Sohl, E.C.Wang, F.C.Park, and J.Kim: "*Optimal Robot Motions for Physical Criteria*", J. Robotic Systems, vol. 18, no. 12, pp. 785-795, Dec 2001.

[11] M.B. Popovic, A. Goswami, and H. Hurr: "*Ground Reference Points in Legged Locomotion: Definitions, Biological Trajectories and Control Implications*", Int. J. of Robotics Research, vol. 24, no. 12, pp. 1013-1032, 2005.

[12] K. Hirai, M. Hirose, Y. Haikawa, T. Takenaka: "The Development of Honda Humanoid Robot", Proc. of 1998 IEEE Int. Conf. on Robotics and Automation, pp. 1321-1326, 1998.

[13] M.Fujita, Y.Kuroki, T.Ishida, and T.Doi: "Autonomous Behavior Control Architecture of Entertainment Humanoid Robot SDR-4X", Proc. of IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 960-967, 2003.

[14] K. Kaneko et al.: "*The Humanoid Robot HRP2*", Proc. of IEEE Int. Conf. on Robotics and Automation, 2004.

[15] S.Kajita et al.: "*Resolved Momentum Control*", Proc. of 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2003.

[16] L. Sentis and O. Khatib,: "*Synthesis of whole-body behaviors through hierarchical control of behavioral primitive*", Int. J. of Humanoid Robotics, 2005.

[17] K. Hauser, T. Bretl, and J.-C. Latombe: "*Non-Gaited Humanoid Locomotion Planning*", Proc. of IEEE-RAS/RSJ Int. Conf. on Humanoid Robotics, 2005.

[18] T. Bretl, S.Lall, J.-C. Latombe, and S.Rock: "*Multi-step Motion Planning for Free-climbing Robots*", Proc. of 6th Int. Workshop on Algorithmic Foundation of Robotics(WAFR), 2004.

[19] T. Bretl: "*Motion Planning of Multi-Limbed Robots Subject to Equilibrium Constraints: The Free-Climbing Robot Problem*", To appear in Int. J. Rob. Res., 2006.

[20] J.J. Kuffner, Jr., K. Nishiwaki, S. Kagami, M. Inaba, H. Inoue,: "Motion Planning for Humanoid Robots", Int. Symp. Rob. Res., 2003.

[21] H. Choset, K.M. Lynch, S. Hutchinson, G. Kanto, W. Burgard, L.E. Kavraki, and S. Thrun: "Principles of Robot Motion: Theory, Algorithms, and Implementations", MIT Press,2005.

[22] K.Harada, S.Kajita, K.Kaneko, and H.Hirukawa: "*Analytical Method on Real Time Gait Planning for a Humanoid Robot*", Proc. of 2004 IEEE-RAS/RSJ Int. Conf. on Humanoid Robotics, 2004.

[23] K.Harada, M.Kaneko, and T.Tsuji: "*Rolling Based Manipulation for Multiple Objects*", IEEE Trans. on Robotics and Automation, vol.16, no.5, pp. 457-468, 2000.

[24] Y.Chen and A.L.Schwartz: "*Riots95–a Matlab toolbox for solving general optimal control problems*", R.Luus(Edit), Transworld Research Publishers, 2002, pp.229-252, 2002.