The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Contact Dynamics Modeling of a Humanoid Robot for Tasks Utilizing Impact Dynamics

Teppei Tsujita, Atsushi Konno and Masaru Uchiyama

***Abstract*— In order to exert a large force on the environment, it is effective to apply impulsive force. We describe the motions that perform tasks by applying impulsive force as "impact motion." This paper presents a contact dynamics model of a humanoid robot for such a motion. Multibody dynamics and effect of a servo controller on impulsive force are also considered in the proposed model. The proposed model can estimate impulsive force at low computation cost compared with full-featured dynamics computation methods. The estimation results of each motion are compared with simulation results by OpenHRP3. The maximum error of impulse is about 6 (%). Therefore, the proposed model is useful for estimating dynamics behavior of a humanoid robot.**

I. INTRODUCTION

When a robot applies force statically on the environment, the magnitude of the force is limited by the maximum torque of the actuators. In order to exert a large force on the environment beyond this limitation, it is effective to apply impulsive force. We describe the motions that perform tasks by applying impulsive force as "impact motion." There are difficult problems introduced by impacts between robots and environments.

Uchiyama proposed a control algorithm constitution method and dynamic control modes for performing a nailing task by a 3 DOF manipulator [1]. Zheng and Hemami discussed mathematical modeling of a robot that collides with the environment [2]. Asada and Ogawa proposed the *virtual mass* for analyzing dynamic behavior of a manipulator arm and its end effector that interacts with the environment [3]. Around the same time, Khatib and Burdick proposed the *effective mass* [4]. Walker investigated the effect of different configurations of kinematically redundant arms with impact force at their end effectors during contact [5]. Tsujio et al. evaluated the *virtual mass* of a 3 DOF planar manipulator empirically [6]. These works mentioned above used robotic manipulators fixed on the ground.

Yoshida et al. investigated impact dynamics in free-floating multibody systems in space [7]. Yoshikawa and Yamada evaluated effects of the joint stiffness against impulse by considering frequency domain [8]. These studies focused on trying to minimize the impulsive force since the force causes fatal problems in space.

A few attempts on tasks applying impulsive force by a humanoid robot have been reported in recent years. Arisumi and Yokoi investigated a method to push a door utilizing impulsive force by a humanoid robot [9]. In the research, the humanoid robot is treated as a rigid single body. However, when a humanoid robot exerts a impulsive force by its end effector, the effect of the servo stiffness cannot be ignored. This paper presents a contact dynamics model of a humanoid robot considering multibody dynamics and effect of a servo stiffness to impulsive force.

II. ESTIMATION OF IMPULSIVE FORCE

When a robot exerts impulsive force on the environment, the reaction force may bring the humanoid robot down. However, the impulsive force is objective value in an impact motion. In this research, the impact motion is planned to be optimized under these objective functions and constraint conditions by SQP(Sequential Quadratic Programming). Since SQP execute the objective function which computes the impulsive force exerted by the robot numerous times, it is indispensable to estimate the force exerted by impact motions at low computation cost.

The concept of *virtual mass* is the projection of the robot's inertia onto the contact point by using the Jacobian matrix. The behavior of the robot during the impact phase is approximated by point mass dynamics. The concept lowers the computational cost of contact dynamics of manipulators fixed on the earth or space robots drastically. The contact dynamics analysis based on the *virtual mass* is for high frequency collisions, e.g. collisions with rigid objects. However, the contact time of the desired practical tasks, e.g., nailing task, in this research is around 10^{-2} (s) as stated in [10]. In this frequency domain, the servo stiffness cannot be negligible. Moreover, the point mass approximation poses a difficulty to compute ZMP(Zero-Moment Point) [11] since behavior of the robot's links is not computed. In this section, a dynamics computation scheme considering the servo stiffness is proposed. Since this scheme computes behavior of the links, ZMP can be estimated during collision phase.

A. A Model for Contact Dynamics

A humanoid robot can have two phases while standing. It can either have both feet on the ground or one foot lifted. In addition, it interacts with the environment using its hands. In full-featured dynamics computation, all the constraint forces are calculated. The spring-damper model is commonly used for humanoid robot simulations [12], [13]. When compared to the LCP(Linear Complementarity Problem) based method [14], the computational cost is lower. Still, the computational complexity of the method is linearly

This research was supported by NEDO Industrial Technology Research Grant Program (project ID: 05A30703a) and JSPS Grant-in-Aid for JSPS Fellows (20·6273).

T. Tsujita, A. Konno and M. Uchiyama are with Department of Aerospace Engineering, Tohoku University, Aoba-yama 6-6-01, Japan. tsujita@space.mech.tohoku.ac.jp

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

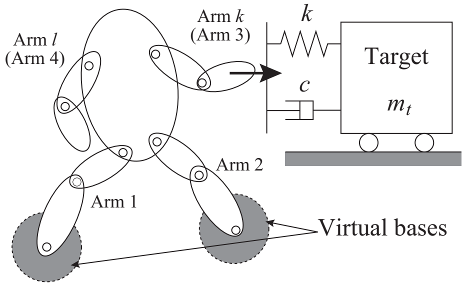


Fig. 1. Modeling of a humanoid robot and a target object.

proportional to the number of contact points. Moreover, the stability of the dynamics simulation is highly dependent on configurations of the spring-damper or the integral methods.

In order to overcome these difficulties, the idea of "virtual base," which is a hypothetical fixed point by adding the huge mass to the links constrained to the environment, e.g., the soles or hands, is suggested [10]. Extra weight is added to both soles when the robot stands on both feet as shown in Fig. 1. This scheme can be applied for various constraint conditions without reconstructing the robot's structure description and computing constraint forces. The mass properties of its end effectors are varied depending on the constraint conditions. The virtual base link(s) is assumed to be part of the ground by the following expression:

$$m_i = M_g, \tag{1}$$

$$\boldsymbol{I}_i = I_g \boldsymbol{E}, \tag{2}$$

where m_i and $\boldsymbol{I}_i \in R^{3\times3}$ are mass of link i and inertia tensor with respect to its mass center respectively. M_g and I_g are chosen to satisfy the conditions described below:

$$M_g \gg M, \tag{3}$$

$$\det[\boldsymbol{I}_i] \gg \det[\boldsymbol{I}_w], \tag{4}$$

where M is total mass of the system and $\boldsymbol{I}_w$ is inertia of the whole system. The dynamics of the robot can be solved in a way similar to space robots dynamics with the above model. A general form of the equation of motion of a n DOF model with respect to joint space can be written as follows:

$$\boldsymbol{H}^*(\boldsymbol{\phi})\ddot{\boldsymbol{\phi}} + \boldsymbol{c}^*(\boldsymbol{\phi}, \dot{\boldsymbol{\phi}}) + \boldsymbol{g}(\boldsymbol{\phi}) + \boldsymbol{F}(\dot{\boldsymbol{\phi}}) = \boldsymbol{\tau} + \left(\boldsymbol{J}^{*k}(\boldsymbol{\phi})\right)^T \boldsymbol{\mathcal{F}}_h, \tag{5}$$

where $\boldsymbol{\phi} \in R^n$, $\boldsymbol{H}^* \in R^{n\times n}$ and $\boldsymbol{J}^{*k} \in R^{6\times n}$ are the joint angle vector, the *generalized inertia matrix* and the *generalized Jacobian matrix* of multiple serial manipulators mounted on a free-flying space robot respectively [15]. The Jacobian matrix is computed with respect to the contact point and superscript k expresses the number of the colliding arm as shown in Fig. 1. $\boldsymbol{c}^*$, $\boldsymbol{g}$ and $\boldsymbol{\tau} \in R^n$ are centrifugal and Coriolis term, a gravity term and input torques respectively. $\boldsymbol{\mathcal{F}}_h \in R^6$ is external force. $\boldsymbol{F}(\dot{\boldsymbol{\phi}})$ is viscous and Coulomb friction term.

B. Simplified Dynamics Computation

The concept of the proposed simplified dynamics computation scheme is based on the assumptions described as below.

(i) The impulsive force exerted by the whole body is extremely large, since this research tries to utilize impact rather than reducing it.

(ii) Contact between target object and end effector occurs within a short time.

(iii) The joint angles of the robot are controlled by a high gain PD(Proportional Derivative) controller in the configuration space and the reference trajectory follows time-series data of the joint angles and velocities.

This scheme is applied to Lagrange formulation as an example. However, the scheme can be applied to faster forward dynamics equations.

From (5), the joint acceleration vector can be solved in the following form:

$$\ddot{\boldsymbol{\phi}} = \boldsymbol{H}^{*-1}\left\{\boldsymbol{\tau} + \left(\boldsymbol{J}^{*k}\right)^T \boldsymbol{\mathcal{F}}_h - \boldsymbol{c}^* - \boldsymbol{g} - \boldsymbol{F}\right\}, \tag{6}$$

where $\boldsymbol{H}^*$ contains equivalent inertia of the joints. Assuming (i), the effects of the centrifugal and Coriolis term and the gravity term are relatively small compared with the torques from the contact force.

$$\|\boldsymbol{c}^*(\boldsymbol{\phi}, \dot{\boldsymbol{\phi}})\| \ll \|\left(\boldsymbol{J}^{*k}\right)^T \boldsymbol{\mathcal{F}}_h\|, \tag{7}$$

$$\|\boldsymbol{g}(\boldsymbol{\phi})\| \ll \|\left(\boldsymbol{J}^{*k}\right)^T \boldsymbol{\mathcal{F}}_h\|. \tag{8}$$

Hence, (6) can be simplified as follows:

$$\ddot{\boldsymbol{\phi}} = \boldsymbol{H}^*(\boldsymbol{\phi})^{-1}\left\{\boldsymbol{\tau} + \left(\boldsymbol{J}^{*k}\right)^T \boldsymbol{\mathcal{F}}_h - \boldsymbol{F}\right\}, \tag{9}$$

where $\boldsymbol{\tau}$ is the input torque vector exerted by the actuators and calculated as follows on the assumption (iii):

$$\boldsymbol{\tau}_{PD} = \boldsymbol{K}_p\left(\boldsymbol{\phi}_{ref} - \boldsymbol{\phi}_{act}\right) + \boldsymbol{K}_d\left(\dot{\boldsymbol{\phi}}_{ref} - \dot{\boldsymbol{\phi}}_{act}\right). \tag{10}$$

$\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are the proportional and derivative gain matrices and $\boldsymbol{\phi}_{ref} \in R^n$ and $\boldsymbol{\phi}_{act} \in R^n$ denote the reference and actual joint angles respectively.

Based on the assumptions (i) and (ii), the behavior of the joint angles is considered below:

Joint angles $\boldsymbol{\phi}$:

The values are taken as constant immediately before and after the impact,

Joint velocities $\dot{\boldsymbol{\phi}}$:

The changes of the values are discontinuous and can not be ignored,

Joint acceleration $\ddot{\boldsymbol{\phi}}$:

The changes of the values are huge.

In addition, the error between $\boldsymbol{\phi}_{ref}$ and $\boldsymbol{\phi}_{act}$ is assumed small just before collision because the high gain servo controller pursues the reference angle. Therefore, (10) can be simplified as follows since the joint angles are taken as constant during the impact:

$$\boldsymbol{\tau}_{PD} = \boldsymbol{K}_d\left(\dot{\boldsymbol{\phi}}_{ref} - \dot{\boldsymbol{\phi}}_{act}\right). \tag{11}$$

In order to consider the motor characteristics, the input torques are limited by the maximum torques as follows:

$$\tau_i = \begin{cases} \tau_{maxi} & \tau_{PDi} > \tau_{maxi} \\ \tau_{PDi} & \|\tau_{PDi}\| < \tau_{maxi} \\ -\tau_{maxi} & \tau_{PDi} < -\tau_{maxi} \end{cases}, \tag{12}$$

where τ_{maxi} and τ_{PDi} are maximum torque and computed torque of the i joint.

Fig. 1 shows the simulation model of the contact force using the proposed simplified dynamics computation scheme. As an example, the contact model between the robot and the target is expressed by the commonly-used spring and damper model. However, this scheme is not dependent on the contact model and any contact model can be implemented. In order to calculate the spring or damper force, the relative position and velocity between the end effector and the target object are necessary. The position and velocity of the end effector can be obtained by integrating the acceleration of the tip. The acceleration of the point in the inertial coordinate system can be calculated using the *generalized Jacobian matrix*, the relation between $\dot{\boldsymbol{\phi}}$ and the velocity $\boldsymbol{v}_h$ or angular velocity $\boldsymbol{\omega}_h$ of the contact point as follows:

$$\begin{bmatrix} \boldsymbol{v}_h \\ \boldsymbol{\omega}_h \end{bmatrix} = \boldsymbol{J}^{*k}\dot{\boldsymbol{\phi}}. \tag{13}$$

Differentiating (13) by time yields:

$$\begin{bmatrix} \dot{\boldsymbol{v}}_h \\ \dot{\boldsymbol{\omega}}_h \end{bmatrix} = \dot{\boldsymbol{J}^{*k}}\dot{\boldsymbol{\phi}} + \boldsymbol{J}^{*k}\ddot{\boldsymbol{\phi}}. \tag{14}$$

When the Jacobian matrix is not singular, the second term of the right hand of (14) is dominant since the joint acceleration is assumed huge. From (9), the acceleration of the contact point is obtained in following form:

$$\begin{bmatrix} \dot{\boldsymbol{v}}_h \\ \dot{\boldsymbol{\omega}}_h \end{bmatrix} = \boldsymbol{J}^{*k}\ddot{\boldsymbol{\phi}} = \boldsymbol{J}^{*k}\boldsymbol{H}^{*-1}\left\{\boldsymbol{\tau} + \left(\boldsymbol{J}^{*k}\right)^T \boldsymbol{\mathcal{F}}_h - \boldsymbol{F}\right\}. \tag{15}$$

The acceleration of the contact point in the collision direction can be expressed as follows:

$$\dot{v}_c = \boldsymbol{n}^T \dot{\boldsymbol{v}}_h, \tag{16}$$

where $\boldsymbol{n} \in R^3$ is a normal vector of the collision direction. By integrating the acceleration $\dot{v}_c$, the position and velocity in the collision direction can be obtained and $\boldsymbol{\mathcal{F}}_h$ is denoted as follows using the contact force f_r calculated by the spring-damper model:

$$\boldsymbol{\mathcal{F}}_h = \begin{bmatrix} f_r \boldsymbol{n} \\ \boldsymbol{0} \end{bmatrix}. \tag{17}$$

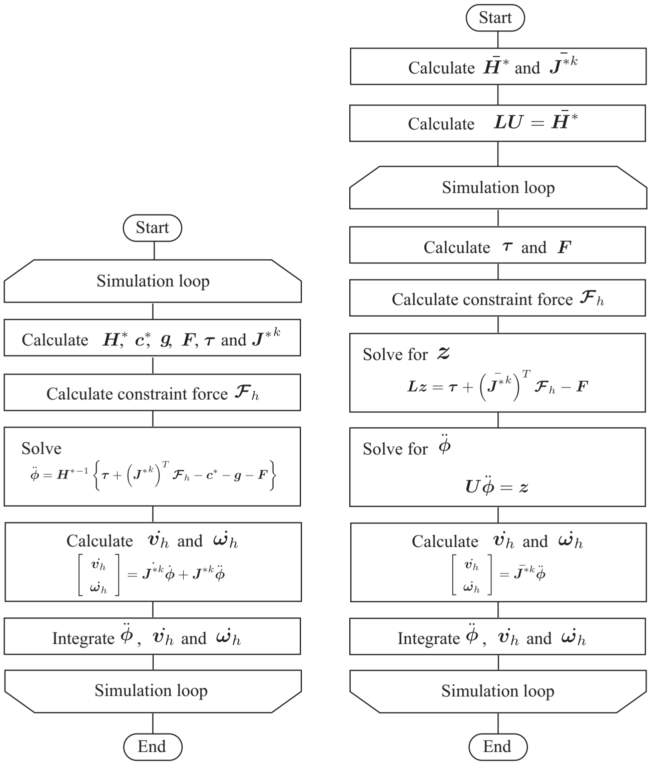
The simulation flow is expressed as follows:

Step 1a: The *generalized Jacobian matrix* $\boldsymbol{J}^{*k}$ and the *generalized inertia matrix* $\boldsymbol{H}^*$ are calculated.

Step 2a: The joint torques $\boldsymbol{\tau}$, $\left(\boldsymbol{J}^{*k}\right)^T \boldsymbol{\mathcal{F}}_h$ and $\boldsymbol{F}$ are calculated.

Step 3a: The joint accelerations are solved using (9).

Step 4a: The accelerations of the contact point are obtained using (15).


Start
Simulation loop
Calculate $\boldsymbol{H}^*, \boldsymbol{c}^*, \boldsymbol{g}, \boldsymbol{F}, \boldsymbol{\tau}$ and $\boldsymbol{J}^{*k}$
Calculate constraint force $\boldsymbol{\mathcal{F}}_h$
Solve
$\ddot{\phi} = \boldsymbol{H}^{*-1}\left\{\boldsymbol{\tau} + \left(\boldsymbol{J}^{*k}\right)^T \boldsymbol{\mathcal{F}}_h - \boldsymbol{c}^* - \boldsymbol{g} - \boldsymbol{F}\right\}$
Calculate $\dot{\boldsymbol{v}}_h$ and $\dot{\boldsymbol{\omega}}_h$
$\begin{bmatrix} \dot{\boldsymbol{v}}_h \\ \dot{\boldsymbol{\omega}}_h \end{bmatrix} = \dot{\boldsymbol{J}^{*k}}\dot{\boldsymbol{\phi}} + \boldsymbol{J}^{*k}\ddot{\boldsymbol{\phi}}$
Integrate $\ddot{\phi}$, $\dot{\boldsymbol{v}}_h$ and $\dot{\boldsymbol{\omega}}_h$
Simulation loop
End
Start
Calculate $\bar{\boldsymbol{H}}^*$ and $\bar{\boldsymbol{J}}^{*k}$
Calculate $\boldsymbol{LU} = \bar{\boldsymbol{H}}^*$
Simulation loop
Calculate $\boldsymbol{\tau}$ and $\boldsymbol{F}$
Calculate constraint force $\boldsymbol{\mathcal{F}}_h$
Solve for $\boldsymbol{z}$
$\boldsymbol{Lz} = \boldsymbol{\tau} + \left(\bar{\boldsymbol{J}}^{*k}\right)^T \boldsymbol{\mathcal{F}}_h - \boldsymbol{F}$
Solve for $\ddot{\phi}$
$\boldsymbol{U}\ddot{\boldsymbol{\phi}} = \boldsymbol{z}$
Calculate $\dot{\boldsymbol{v}}_h$ and $\dot{\boldsymbol{\omega}}_h$
$\begin{bmatrix} \dot{\boldsymbol{v}}_h \\ \dot{\boldsymbol{\omega}}_h \end{bmatrix} = \bar{\boldsymbol{J}}^{*k}\ddot{\boldsymbol{\phi}}$
Integrate $\ddot{\phi}$, $\dot{\boldsymbol{v}}_h$ and $\dot{\boldsymbol{\omega}}_h$
Simulation loop
End

(a) Normal flow

(b) Simplified flow

Fig. 2. Dynamics simulation flowchart.

Step 5a: The positions and velocities of the contact point, the joint angles and velocities are obtained by the integral function.

Step 6a: Returning to Step 1a.

Fig. 2 (a) shows a flowchart of the normal forward dynamics simulation process. The computation cost of step 5a is dependent upon the integral method. The total computation cost of Step 1a $\sim$ 4a can be denoted as follows:

$$\begin{aligned} C_1 \;=\; & \left\{15\frac{2}{3}n^3 + 63n^2 + 254\frac{1}{2}n + 88 \right. \\ & \left. + 6\left(n_1^2 + n_2^2 + \cdots + n_l^2\right) + 18n_k\right\} T_i, \end{aligned} \tag{18}$$

where T_i is an iteration count of the simulation. T_i is derived as follows:

$$T_i = \frac{t_{total}}{t_s}, \tag{19}$$

where t_{total} and t_s are the total simulation time and sampling time respectively. n and n_i are total DOF and DOF of the i-th arm, respectively. Subscripts l and k denote the number of the last arm and colliding arm as shown in Fig. 1, respectively. The total computation cost denotes that sum of number of four arithmetic operations.

Equation (15) is computed every sampling time of the simulation. The most complex computations are calculating the *generalized inertia matrix* and solving the simultaneous equations by the Gaussian elimination method and these computation costs are $O(n^3)$. In order to reduce the simulation cost, $\boldsymbol{H}^*$ is regarded as a constant value in the simulation loop due to assumption (ii). In addition, the LU decomposition of $\boldsymbol{H}^*$ is calculated before starting the loop. The modified simulation scheme is as follows.

Step 1b: The *generalized Jacobian matrix* $\boldsymbol{J}^{*k}$ and the *generalized inertia matrix* $\boldsymbol{H}^*$ are calculated by using the joint angles just before contact and stored in the computer's main memory as $\bar{\boldsymbol{J}}^{*k}$ and $\bar{\boldsymbol{H}}^*$ respectively.

Step 2b: The LU decomposition of the $\bar{\boldsymbol{H}}^*$ ($\boldsymbol{LU} = \bar{\boldsymbol{H}}^*$) is computed and the $\boldsymbol{L}$ and $\boldsymbol{U}$ are stored.

Step 3b: The joint torques $\boldsymbol{\tau}$, $\left(\bar{\boldsymbol{J}}^{*k}\right)^T \boldsymbol{\mathcal{F}}_h$ and $\boldsymbol{F}$ are calculated.

Step 4b: Solving $\boldsymbol{Lz} = \boldsymbol{\tau} + \left(\bar{\boldsymbol{J}}^{*k}\right)^T \boldsymbol{\mathcal{F}}_h - \boldsymbol{F}$ for $\boldsymbol{z}$.

Step 5b: Solving $\boldsymbol{U}\ddot{\boldsymbol{\phi}} = \boldsymbol{z}$ for $\ddot{\boldsymbol{\phi}}$.

Step 6b: The accelerations of the contact point are obtained by $\boldsymbol{J}^{*k}\ddot{\boldsymbol{\phi}}$.

Step 7b: The positions and velocities of the contact point and the joint angles and velocities are obtained by the integral function.

Step 8b: Return to Step 3b.

Fig. 2 (b) shows a flowchart of the simplified forward dynamics simulation process. The total computation cost of Step 1b $\sim$ 6b can be denoted as follows:

$$C_2 = \left\{15\frac{2}{3}n^3 + 63n^2 + 242n + 73 + 6\left(n_1^2 + n_2^2 + \cdots + n_l^2\right) + 18n_k\right\} + \left(2n^2 + 25n - 6\right)T_i. \quad (20)$$

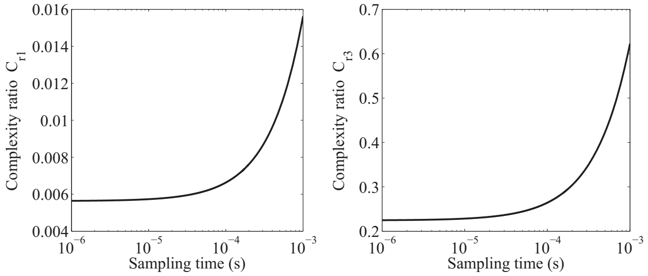
In Step 5b and 7b, the joint angle, velocity and acceleration can be obtained and force and moment in the inertial coordinate system can be computed using inverse dynamics. Therefore, ZMP can be calculated by the force and moment.

C. Computation Cost

In order to compare the complexity, the complexity ratio C_{r1} is defined as follows:

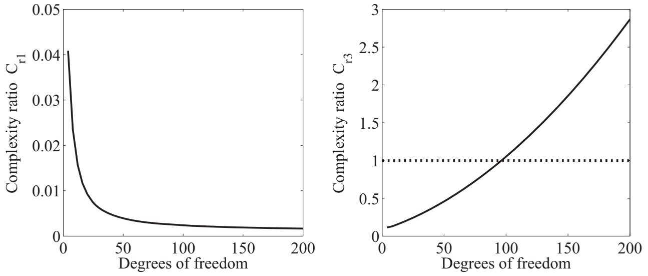
$$C_{r1} = \frac{C_2}{C_1}. \quad (21)$$

Fig. 3 (a) shows relationship between sampling time and the complexity ratio C_{r1}. The total simulation time t_{total} is 0.1 (s). In order to compute (18) and (20), the total DOF n is based on a humanoid robot HRP-2 [16] model as shown in Fig. 5. HRP-2 has 28 DOF except hand joints. The computation cost ratio is about 0.016 at the sampling time 10^{-3} (s). From 10^{-3} (s) to 10^{-5} (s), the computation cost is reduced as the sampling time is decreased. Under the sampling time 10^{-5}, the computation ratio is almost constant. Fig. 4 (a) shows relationship between DOF and the complexity ratio C_{r1} under the assumption that $l = 4$ and $n_k = n/4$ at the sampling time 10^{-4} (s).



(a) C_{r1} (b) C_{r3}

Fig. 3. Relationship between sampling time and the complexity ratios.



(a) C_{r1} (b) C_{r3}

Fig. 4. Relationship between DOF and the complexity ratios.

The computation cost is compared with the $O(n)$ forward dynamics algorithm based on articulated-body inertias [17] as an example. The method requires $199n - 198$ multiplications and $174n - 173$ additions. Hence, the total computation cost is denoted as follows:

$$C_3 = 373n - 371. \quad (22)$$

The complexity ratio C_{r3} is expressed as follows:

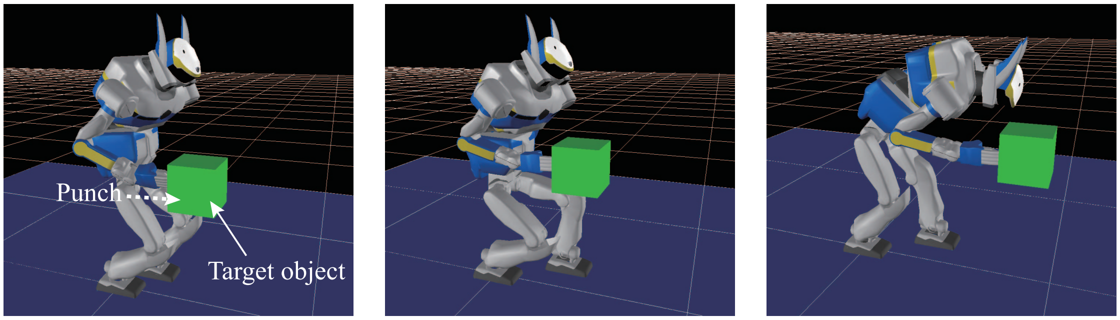
$$C_{r3} = \frac{C_2}{C_3}. \quad (23)$$

Fig. 3 (b) shows relationship between sampling time and the complexity ratio C_{r3} of the HRP-2 model. At the sampling time 10^{-4} (s), the computation cost is about $1/4$. Fig. 4 (b) shows relationship between DOF and the complexity ratio C_{r3} on assumption that $l = 4$ and $n_k = n/4$ at the sampling time 10^{-4} (s). Under 100 DOF, the complexity ratio is under one. Hence, the proposed algorithm has an advantage in the computation cost for a humanoid robot such as HRP-2.

III. EVALUATION OF THE MODEL

In order to evaluate the accuracy of the estimation of impulsive force and ZMP trajectory, nine type of motions are designed heuristically. The *virtual mass* m_a of each posture is computed by the model described in [10]. The estimation results of each motion are compared with the simulation result using OpenHRP3 [14], [18]. The collision speeds of each posture are 0.4 (m/s), 0.7 (m/s) and 1.0 (m/s) respectively. Table I shows the type names of each motion. For example, the *virtual mass* and collision speed v_c of the A-1 motion are 4.0 (kg) and 0.4 (m/s) respectively. Fig. 6 shows the experimented postures.

As an example, Fig. 7 shows the estimated impulsive force history of the Type C series motions and the simulation result in OpenHRP3. The model is expressed as shown in Fig. 1. In order to express the constraint conditions of the floor the extra weights are added to the both soles. All the



(a) $m_a = 4.0$ (kg)
Type A series

(b) $m_a = 6.0$ (kg)
Type B series

(c) $m_a = 8.0$ (kg)
Type C series

Fig. 6. HRP-2 punches the target object by different postures.

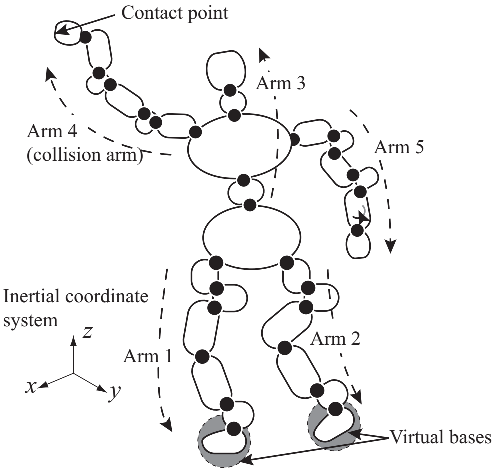
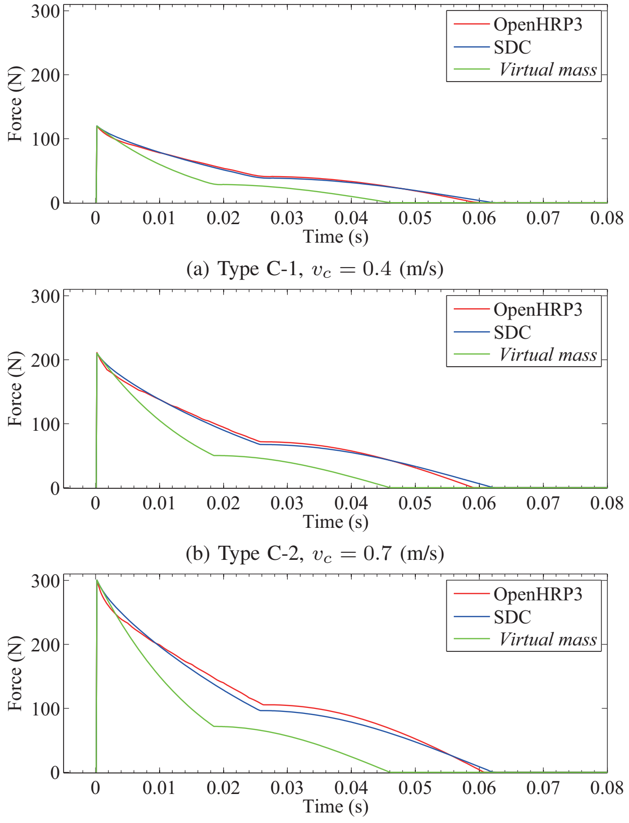


Fig. 5. Notation of the links of HRP-2.

TABLE I
THE NAMES OF EACH MOTION.

		Virtual mass (kg)		
		4.0	6.0	8.0
Collision	0.4	A-1	B-1	C-1
velocity	0.7	A-2	B-2	C-2
v_c (m/s)	1.0	A-3	B-3	C-3



(a) Type C-1, $v_c = 0.4$ (m/s)

(b) Type C-2, $v_c = 0.7$ (m/s)

(c) Type C-3, $v_c = 1.0$ (m/s)

Fig. 7. Estimated impulsive force and OpenHRP3 simulation result (Type C series motions).

extra weights of the simplified model are 10^5 (kg) and all the extra inertias are $\mathrm{diag}\left[10^5, 10^5, 10^5\right]$ (kgm^2). HRP-2 is controlled by the torque controller in OpenHRP3 simulator environment and the block diagram of the controller is shown in Fig. 8. The torque limit is given by the specifications of the robot. The joint angles are controlled by PD(Proportional Derivative) controller. The mass of the target object is 5 (kg). The spring coefficient k and damper coefficient c are 10^4 (N/m) and 300 (Ns/m) respectively. These values are decided in order to simulate 10 $\sim$ 100 (ms) collision. The static and dynamic friction coefficients between the sole and the floor are 5 and 5, respectively. The joint friction is not considered in the proposed scheme and OpenHRP3 simulation in this evaluation because specifications of the joints of HRP-2 is not disclosed.

In addition, in order to see the accuracy of the *virtual mass*, simulation results of all three methods are plotted in these figures. In these simulations, the integral method and its sampling time are Runge-Kutta method and 10^{-4} (s) respectively. Fig. 9 shows the simulation model of the contact force using the *virtual mass*. The contact model is expressed by the same spring and damper model. The *virtual mass* and target object can move in the same straight line freely. The *virtual mass* point, with mass m_a, approaches the target object with collision velocity v_c. The target mass is 5 (kg).

In the Fig. 7, “SDC(Simplified Dynamics Computation)” means the proposed dynamics computation scheme. In all results, the force history by the proposed scheme is quite similar to the OpenHRP3’s result. However, the estimation using the *virtual mass* model is not similar during late phase of collision. A conceivable reason is that the *virtual mass* does not consider the effect of the servo controller. Table II

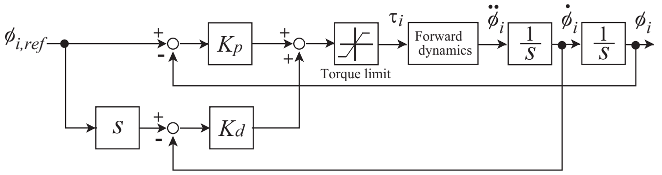


Fig. 8. Block diagram of a feedback control.

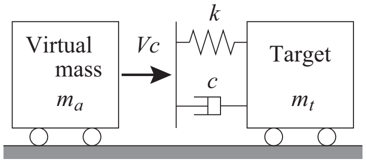


Fig. 9. Modeling of the contact force using the virtual mass.

and Table III show estimation errors of the maximum force and impulse based on OpenHRP3's result. In these tables, f_{max}, f_n, f_{err} and f_{nerr} are maximum force, impulse, maximum force error and impulse error, respectively. The errors are computed in the form:

$$x_{err} = \frac{x_{OH3} - x_{est}}{x_{OH3}} \times 100, \tag{24}$$

where x_{OH3} and x_{est} are OpenHRP3's result and estimated value by the proposed model or the *virtual mass* model, respectively. The maximum force errors are minor in both estimating methods. The impulse error of the *virtual mass* method, however, is not negligible. The maximum error of the *virtual mass* is about 54 (%) as shown in Table III. On the other hand, the impulse error of the proposed method is small. The maximum error is about 6 (%).

TABLE II

COMPARISON BETWEEN THE SIMPLIFIED DYNAMICS COMPUTATION SCHEME AND OPENHRP.

	OpenHRP3		Simplified dynamics computation			
Type	f_{max} (N)	f_n (Ns)	f_{max} (Ns)	f_{err} (%)	f_n (Ns)	f_{nerr} (%)
A-1	122.6	2.6	119.7	2.3	2.6	−0.4
A-2	215.7	4.5	209.9	2.7	4.6	−1.2
A-3	299.3	6.6	300.4	−0.4	6.5	0.7
B-1	121.3	2.5	119.7	1.2	2.7	−6.1
B-2	209.1	4.5	209.9	−0.4	4.7	−4.5
B-3	301.3	6.5	300.4	0.3	6.7	−2.4
C-1	119.6	2.8	120.1	−0.4	2.7	0.6
C-2	211.3	4.8	210.2	0.5	4.8	0.2
C-3	299.5	7.1	300.7	−0.4	6.9	4.0

TABLE III

ESTIMATED MAXIMUM FORCE AND IMPULSE BY THE *virtual mass*.

Type	m_a (kg)	v_c (m/s)	f_{max} (N)	f_{err} (%)	f_n (Ns)	f_{nerr} (%)
A-1	4.0	0.4	119.7	2.3	1.2	53.2
A-2	4.0	0.7	209.9	2.7	2.1	52.8
A-3	4.0	1.0	300.2	−0.3	3.0	53.7
B-1	6.0	0.4	119.7	1.3	1.5	39.9
B-2	6.0	0.7	209.9	−0.4	2.7	40.8
B-3	6.0	1.0	300.2	0.3	3.8	42.1
C-1	8.0	0.4	119.8	−0.2	1.7	37.3
C-2	8.0	0.7	210.4	0.4	3.0	36.9
C-3	8.0	1.0	300.6	−0.4	4.3	39.4

IV. CONCLUSIONS

This paper presents a contact dynamics model of a humanoid robot considering multibody dynamics and the effect of the servo stiffness to impulsive force. The maximum error of impulse is about 6 (%). Therefore, the proposed model is useful for estimating dynamic behavior of a humanoid robot. The proposed scheme is applied to Lagrange formulation as an example, here. However, the scheme can be applied to faster forward dynamics equations and the computation cost is lower than the example described in Subsection II-C. The model will be evaluated using a real robot and used for optimizing impact motions.

REFERENCES

[1] M. Uchiyama, "A control algorithm constitution method for aritificial arm and dynamic control modes," in *Biomechanism 3*. University of Tokyo Press, 1975, (in Japanese).

[2] Y. F. Zheng and H. Hemami, "Mathematical modeling of a robot collision with its environment," *J. of Robotic Systems*, vol. 2, no. 3, pp. 289–307, 1985.

[3] H. Asada and K. Ogawa, "On the dynamic analysis of a manipulator and its end effector interacting with the environment," in *Proc. of the 1987 IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 751–756, 1987.

[4] O. Khatib and J. Burdick, "Motion and force control of robot manipulators," in *Proc. of the 1986 IEEE Int. Conf. on Robotics and Automation*, pp. 1381–1386, 1986.

[5] I. D. Walker, "Impact configurations and measures for kinematically redundant and multiple armed robot systems," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 670–683, 1994.

[6] S. Tsujio and Y. Seo, "Dynamics of manipulator tip collision with its environment –evaluation of theoretical equivalent mass of links at tip position by experiments–," in *Proc. of the 1999 JSME Conf. on Robotics and Mechatronics*, 2A1-32-034, 1999, (in Japanese).

[7] K. Yoshida and N. Sashida, "Modeling of impact dynamics and impulse minimization for spacerobots," in *Proc. of the 1993 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2064–2069, 1993.

[8] S. Yoshikawa and K. Yamada, "Dynamics evaluation for capturing a target by s space robot – effects of the joint stiffness –," *J. of the Robotics Society of Japan*, vol. 13, no. 2, pp. 242–248, 1995.

[9] H. Arisumi and K. Yokoi, "Whole-body motion of a humanoid robot for passing through a door," in *Proc. of the 9th SICE System Integration Division Annual Conf.*, pp. 1067–1068, 2008, (in Japanese).

[10] T. Tsujita *et al.*, "Analysis of nailing task motion for a humanoid robot," in *Proc. of the 2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1570–1575, 2008.

[11] M. Vukobratović *et al.*, *Biped Locomotion – Dynamics, Stability, Control and Application*. Springer-Verlag, 1990.

[12] H. Hirukawa *et al.*, "Experimental evaluation of the dynamic simulation of biped walking of humanoid robots," in *Proc. of the 2003 IEEE Int. Conf. on Robotics and Automation*, pp. 1640–1645, 2003.

[13] Y. Hwang *et al.*, "An order n dynamic simulator for a humanoid robot with a virtual spring-damper contact model," in *Proc. of the 2003 IEEE Int. Conf. on Robotics and Automation*, pp. 31–26, 2003.

[14] S. Nakaoka *et al.*, "Constraint-based dynamics simulator for humanoid robots with shock absorbing mechanisms," in *Proc. of the 2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 3641–3647, 2007.

[15] K. Yoshida, R. Kurazume, and Y. Umetani, "Dual arm coordinate in space free-flying robot," in *Proc. of the 1991 IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 2516–2521, 1991.

[16] K. Kaneko *et al.*, "Humanoid robot HRP-2," in *Proc. of the 2004 IEEE Int. Conf. on Robotics and Automation*, pp. 1083–1090, 2004.

[17] R. Featherstone, "The calculation of robot dynamics using articulated-body inertias," *J. of Robotics Research*, vol. 2, no. 1, pp. 13–30, 1983.

[18] H. Hirukawa *et al.*, "A robot simulator constructed on distributed object modules," in *Proc. of the 25th Annual Conf of the Robotics Society of Japan*, 1L22, 2007, (in Japanese).