論文 / 著書情報
Article / Book Information

Title	An Empirical Comparison of a Free Dynamics Simulator “ Open Dynamics Engine ” with TITAN-VIII Hardware Torque/Power Measurements
Author	Gen Endo, Keisuke Arikawa, Shigeo Hirose
Journal/Book name	, , ,
Issue date	2011, 5
DOI	http://dx.doi.org/10.1109/ICRA.2011.5980182
URL	http://www.ieee.org/index.html
Copyright	(c)2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.
Note	このファイルは著者（最終）版です。 This file is author (final) version.

An Empirical Comparison of a Free Dynamics Simulator "Open Dynamics Engine" with TITAN-VIII Hardware Torque/Power Measurements

Gen Endo, Keisuke Arikawa and Shigeo Hirose

Abstract— **A free rigid body dynamics simulator "Open Dynamics Engine" is empirically compared with the results of hardware measurements in joint torque/power dimensions for a quadruped robot "TITAN-VIII". We developed the quadruped robot model on the simulator and compared simulated joint torques and powers with hardware experiments during walking which takes four different walking postures. The results suggest that the direct acquisitions of simulated joint torques include a large latency in our application but this defect is avoidable by using joint force information. Careful parameter settings and tunings permit the simulator to match hardware experimental data with sufficient accuracy.**

I. INTRODUCTION

In recent years, the performance of the numerical physics engines for video games has rapidly improved thanks to a diffusion of powerful and cost effective personal computers and home video game consoles. The examples of these physics calculation engine are Havok[1], PhysX[2] and Bullet Physics[3]. In particular, a free open source dynamics library "Open Dynamics Engine (ODE)"[4] originally developed by Russell Smith has been widely used by robotics researches. ODE is very stable and easy to get started because the detailed user guide and various sample source codes are available online[4]. Therefore ODE has been incorporated into various software system such as PyODE[5], OgreODE[6], Webots[7], SimSpark[8] and so on. Especially a commercial simulator Webots for robotics research has been already sold to over 700 universities and SimSpark becomes a common platform of 3D soccer simulation league in Robocup.

Fig. 1(left) shows a survey result[1] of the usage of ODE and Webots. The result shows that 20-30 papers are presented every year and the total number of published papers is growing. Thus ODE has already played an important role in robotics research.

G. Endo and S. Hirose are with the Dept. of Mechanical and Aerospace Engineering, Graduate School of Science and Engineering, Tokyo Institute of Technology, 2-12-1-I1-60 Ookayama, Meguro-ku, Tokyo 152-8552, Japan {gendo, hirose}@mes.titech.ac.jp

K. Arikawa is with the Department of Mechanical Engineering, Kanagawa Institute of Technology, 1030 Shimo-ogino, Atsugi, Kanagawa, 243-0292, Japan arikawa@me.kanagawa-it.ac.jp

[1]We investigated two major robotics conference proceedings, IEEE International Conference on Robotics and Automation (ICRA) and IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) and searched for "Open Dynamics Engine" or "ODE" or "Webot" through all PDF documents in the conference DVD by using Adobe Acrobat PDF searching function. All candidate papers are individually checked whether ODE or Webots is actually used in the paper.

However, ODE was originally developed for video games and the user manual actually points out that ODE is not sufficiently accurate for quantitative engineering[9]. Thus we are concerned about the current situation where many papers are being published without a detailed physical investigation of the relationship between ODE to reality.

Fig. 1(right) shows how ODE is used by the 89 presented papers in the last four years. 42 papers focus on algorithm development and ODE is used to test their algorithm. 12 papers use ODE as a part of virtual reality system. 35 papers use both ODE simulation and a hardware experiment setup. However their comparisons of ODE with hardware experiment basically remain a qualitative consideration in position and/or velocity (for example [10][11]). To the best of our knowledge, there is no detailed investigation of ODE in force and power, which are among the most important physical quantities in robotics research especially for a hardware development.

In this paper, our primal motivation is to empirically assess the ODE performance to simulate a quadruped walking robot as a research tool especially in force and power dimension. We developed a simulation model of a quadruped walking robot "TITAN-VIII" and conducted walking simulations with four different walking postures to measure joint torques and powers. We also carried out experiments using hardware TITAN-VIII with 12 degrees of freedom torque/velocity measurement system. The results are compared and the difference is discussed. The results suggest that ODE requires careful parameter tunings and elaborate derivation of joint torque in our application, but it can provide useful information with us to understand physical phenomena of the quadruped walking.

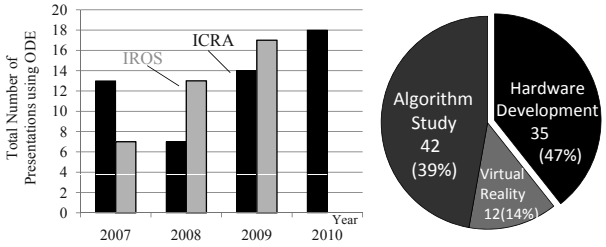

Fig. 1. Result of publication survey: number of publication (left), purpose of the usage (right)

II. TITAN-VIII SIMULATOR

A. Computational Environment

We built ode-0.11.1 with double precision by using Microsoft Visual C++ Express on Windows XP SP3 PC (Core2Quad 3.0GHz, 4GB memory). We use dWorldStep() API to update internal states of dynamics calculation because the API has higher accuracy without using approximate iterations. Simulation time step is set to $1msec$ and it takes $60sec$ to simulate $10sec$ crawl gait as we are to discuss later. The calculation speed is acceptable if we do not intend to use ODE for the study with heavy computational iterations such as a reinforcement learning algorithm or a genetic algorithm.

B. TITAN-VIII Link Model

TITAN-VIII has 4 legs and each leg has 3 active degrees of freedom (DOF). Fig. 2(a) illustrates the arrangement of joints. The parallel link mechanisms using a wire-pulley system are introduced to drive knee joint and ankle joint. The ankle joint is passively adjusted to keep the sole parallel to the main body[12].

The knee joint driving mechanism can be modeled by a parallel four-bar linkage system shown in Fig. 2(b). However it is not computationally efficient to increase the total number of rigid bodies. Thus we model the leg mechanism by a serial link system (Fig. 2(c)), and convert the derived joint velocity and torque into those quantities of the parallel linkage system as follows,

$$\dot{\theta}_p = \mathbf{J_p}^{-1}\mathbf{J_s}\dot{\theta}_s, \quad (1)$$

$$\tau_p = \mathbf{J_p^T}(\mathbf{J_s^T})^{-1}\tau_s. \quad (2)$$

Here $\dot{\theta}_s, \tau_s, \mathbf{J_s}$ are joint velocity, torque and Jacobian matrix of the serial link leg mechanism and $\dot{\theta}_p, \tau_p, \mathbf{J_p}$ are those of the parallel link leg mechanism, respectively.

The rigid links are created by rectangular solids and cylinders and they are connected by a hinge joint as we are to discuss later. TITAN-VIII model uses 25 rigid bodies, 4 fixed joints and 20 hinge joints in total. The size and mass of the link as well as joint positions are adopted from the reference [12] and the design drawings. We assume that the center of gravity is located at the center of geometry. The total weight of the robot including a battery is $22kg$.

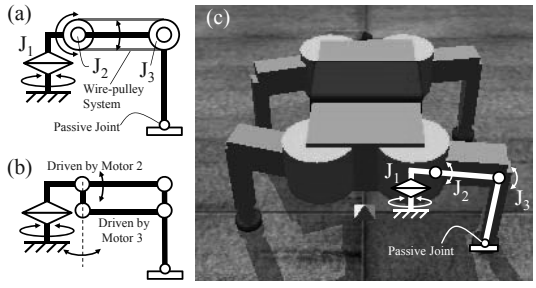

Fig. 2. TITAN-VIII model: (a)actual wire-pulley parallel link mechanism, (b)equivalent parallel link mechanism, (c)screen shot of ODE with serial link mechanism

C. Joint Servo

Since ODE provides dJointAddHingeTorque() API to add joint torque and dJointSetHingeParam() API to set desired joint velocity, we can control each joint by either torque control or velocity control. The active joints in hardware TITAN-VIII are controlled by PID control in position level by each local servo driver. Thus ODE can model this servo by using torque adding API and position/velocity measurement of the joint. However we observed the straightforward implementation of PID joint control became unstable and it was difficult to track a desired position trajectory. The reason is still unclear and we need more investigation. So we took the second best way to avoid this problem. We applied joint velocity control, which is very stable, in order to track a desired position trajectory as follows,

$$\dot{\theta}_d = -k_d(\theta - \theta_d), \quad (3)$$

where $\dot{\theta}_d, \theta_d, \theta, k_d$ are desired joint velocity, desired joint position, measured joint position and velocity gain, respectively. We set the maximum joint torque τ_{max} via velocity control API as in Table I. τ_{max} are calculated from the reduction ratio of each joint[12] and the stalling torque of the equipped actuator. We set small k_d for passive ankle joint and large k_d for other active joints. (Additionally, we observed a noisy ankle torque oscillation when we set the ankle joint completely passive. Thus we introduce a low gain ankle servo and control ankle joint to keep the sole parallel to the main body.)

D. Contact Model Setting

ODE automatically generates a virtual joint at the colliding point between the rigid bodies and applies a force at the joint in order to simulate the collision and friction. We carefully tuned the parameters not to diverse the numerical calculation and to minimize the foot slip as follows.

```
dContact contact[10];
contact[i].surface.mode =
  dContactSlip1 | dContactSlip2 | dContactSoftERP
   | dContactSoftCFM | dContactApprox1;
contact[i].surface.mu       = 0.5;
contact[i].surface.slip1    = 0.001;
contact[i].surface.slip2    = 0.001;
contact[i].surface.soft_erp = 0.2;
contact[i].surface.soft_cfm = 0.0001;
```

We observed an unstable torque oscillation in the joint torque data when we only set the friction pyramid approximation. Thus in order to increase the numerical stability, we introduced a small force-dependent-slip (dContactSlip1, 2) whose magnitude is 0.001, which are originally for the simulation of a wheel slip in the normal direction.

TABLE I
PARAMETERS FOR JOINT SERVO CONTROL

	k_d [1/s]	τ_{max} [Nm]
Hip J1	4.0	32.3
Thigh J2	4.0	40.2
Knee J3	4.0	25.7
Ankle J4	1.0	1.0

III. Simulated Torque Measurement

In this section, the joint torques standing with the standard posture on the horizontal ground are evaluated to verify whether ODE can accurately derive the joint torques or not. We discuss the joint torques τ_s of the serial link mechanism. Fig. 3 shows the standard posture and the body-fixed coordinate system. (The nominal walking direction is in the X direction.) Circular arrows indicate positive direction of the joint torques and velocities. FL, FR, HL, HR are leg's name, for example FL means frontal-left leg and HR means hinder-right leg. We do not discuss the torque of ankle joint $J4$ because there is no actuator in $J4$.

A. 6-axis Force-torque Measurement API

We can not explicitly obtain the joint torques because the joints are controlled by velocity control. However dJointGetFeedback() API permits us to derive the 6-axis force-torque when the rigid body 1 is connected to the rigid body 2 shown in Fig. 4. The argument of this API, dJointFeedback structure, is defined as follows[9].

```
typdef struct dJointFeedback{
 dVector3 f1; //force that joint applies to body 1
 dVector3 t1; //torque that joint applies to body 1
 dVector3 f2; //force that joint applies to body 2
 dVector3 t2; //torque that joint applies to body 2
}
```

Note that derived force and torque are measured at the center of mass (COM) of each rigid body, not at the joint axis shown in Fig. 4. Therefore **t1** and **t2** are not joint torques. Here, the following equilibrium equations at joint position are satisfied based on the Newton's 3rd law.

$$\mathbf{f1} = -\mathbf{f2}, \tag{4}$$

$$\mathbf{t1} + (\mathbf{P_1} - \mathbf{P_j}) \times \mathbf{f1} = -(\mathbf{t2} + (\mathbf{P_2} - \mathbf{P_j}) \times \mathbf{f2}), \tag{5}$$

where $\mathbf{P_1}$, $\mathbf{P_2}$ are position of the rigid body 1 and body 2, and $\mathbf{P_j}$ is the position of the joint. The joint force and torque are the left side of above equations with sign inversion.

For verification, we measured 6-axis joint force-torque on the knee joint ($J3$) during a static standing with the standard posture. We set the initial Z positions of TITAN-VIII in order to have a clearance of $10mm$ between feet and the ground, and then dynamics simulation started. Based on a physical consideration of the static standing with the standard posture, acting force/torque on $J3$ is a vertical reaction force only in the Z direction and there is no torque. We verified that F_z converged to the gravitational force of $49.1N$ acting on each knee joint[2] and the other forces and torques converged to zero after a sufficient standing duration.

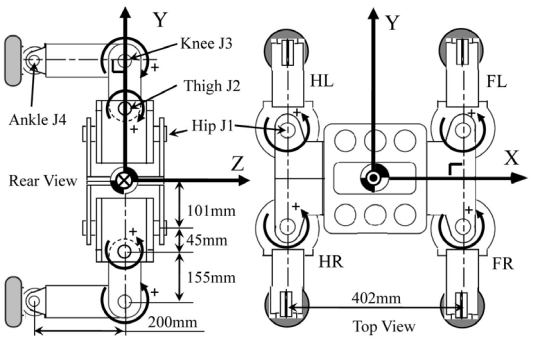

Fig. 3. Standard posture and body-fixed coordinate system

However the speed of convergence of joint torque was very slow. The solid line in Fig. 5 shows the time course of torque convergence. We cut out the initial $30msec$ after legs' touchdown to eliminate the initial impulsive torque of -$25Nm$. As we show later in Fig.8, operational range of J3 torque is about $10Nm$. Even if we accept 5% tolerance of torque error, it takes about $2sec$ to converge. Thus, it is difficult to accurately measure joint torque with Eqn(5) for a walking movement.

We investigated the convergence speed of each term in Eqn(5) and found that **t1** and **t2** made slow convergence. On the other hand, **f1** and **f2** converged rapidly compared with **t1** and **t2**. So far, the reason is unclear and there is a possibility that this problem occurs due to a particular configuration of TITAN-VIII, which forms redundant closed-loops between the 4 legs and the ground.

B. Derivation of the Joint Torque

In the previous section, the result suggests the derived torque **t1**, **t2** in Eqn(5) has a large latency. Thus we propose to derive joint torques by using only translational force information with Jacobian matrix as follows.

$$\tau_s = \mathbf{J_s^T}(\theta)\mathbf{F} \tag{6}$$

where τ_s is a joint torque vector for a serial link leg and $\mathbf{J_s^T}(\theta)$ is its transposed Jacobian matrix and $\mathbf{F}$ is an external force applied to the foot. This method utilizes

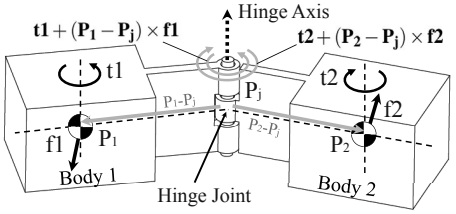

Fig. 4. A hinge joint connects body 1 and body 2

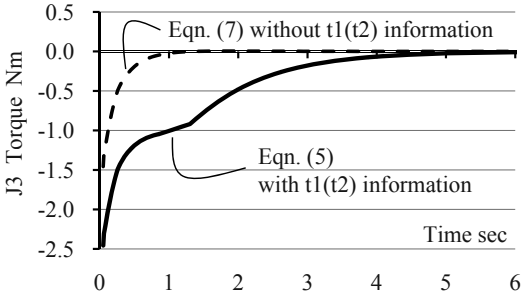

Fig. 5. Joint torque (J3) convergence during a static standing with the standard posture

[2]Total weight is $22.0kg$, total weight of the shank and feet is $2.0kg$, thus (22.0-2.0)*9.81/4=49.1N

a static relationship between a translational external force and a joint torque, and omits **internal** dynamic forces and torques due to inertia of the links. Thus the derived torque is not a true dynamic torque. However $\mathbf{F}$ includes dynamics effect of the overall system such as swing leg touch down, main body tumbling and so on. In the case of quadruped walking, these effects are dominant for joint forces and torques compared with the internal dynamic effect between the leg links. Moreover, even when the quadruped robot walks with a quasi-static gait with a duty factor larger than 0.75, $\mathbf{F}$ becomes a statically-indeterminate problem because of four legs standing. Thus it is beneficial to use ODE to derive $\mathbf{F}$ even in a static situation.

If we can assume that all leg masses are negligibly-small for simplicity, the joint torque τ_{si} (where $i = 1,2,3$ from the proximal link to the distal link) can be obtained by the following equation.

$$\tau_{si} = ((\mathbf{r_a} - \mathbf{r_i}) \times \mathbf{F}) \cdot \mathbf{u_i} \tag{7}$$

Here, $\mathbf{r_a}$ is the foot position where the external force applied, and $\mathbf{r_i}$ is the joint position and $\mathbf{u_i}$ is the unit vector of the joint axis. In the following section, we use Eqn(7) because the leg structure of TITAN-VIII is relatively lightweight compared with the body structure. We set $\mathbf{r_a}, \mathbf{F}$ as $J4$ position and force, respectively. We also measured the knee joint torque during the static standing using Eqn(7) shown in dashed line in Fig. 5. This method has almost 5 times faster convergence compared to Eqn(5).

IV. Hardware Torque Measurement

The relationship between the walking postures and power consumptions using TITAN-VIII has been well studied in the reference[13]. In this section we briefly introduce the outline.

A. Measurement Device

We measure the joint velocity and torque in order to derive the mechanical power during walking. The joint velocity is measured by a local servo driver using an electric governor circuit, which measures counter-electromotive force of the motor coil.

As for torque measurement, the most tractable method to obtain the joint torque is to measure an actuator current. However the actuator of a legged robot usually requires a reduction gear system with a large reduction ratio to generate sufficient torque to sustain the weight of the robot. Thus a current measurement is not accurate to estimate the mechanical joint torque because the reduction gear system usually suffers from a large frictional resistance and the transmission efficiency changes depending on a temperature. Therefore we developed a new torque sensing mechanism "Float Differential (FD) torque sensor"[14], which can measure a pure mechanical torque.

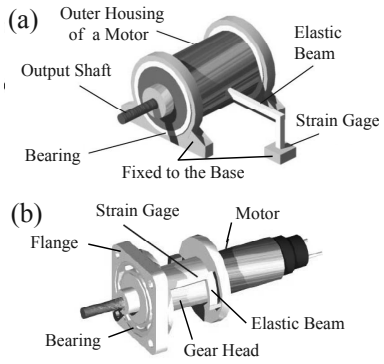

Fig. 6. Float differential torque sensor: (a) basic mechanical structure, (b) actual implementation for a conventional geared motor

Fig. 6(a) illustrates the basic mechanism of FD torque sensor. The outer housing of the motor is supported by a bearing whose outer ring is fixed to the base structure. The outer housing is also supported by an elastic beam whose extremity is also attached to the base structure. A strain gage is mounted on the elastic beam. When the output shaft of the motor generates a torque, a reaction torque rotates the outer housing in the opposing direction with respect to the output rotation because the housing is free to rotate against the base structure. The reaction torque causes a deformation of the elastic beam that can be measured by the strain gage. FD torque sensor can measure a pure torque of the output shaft and eliminates the friction of the gear head. Fig. 6(b) is an example of the actual implementation for an ordinary DC actuator with a cylindrical gear head. This torque sensor can be easily incorporated into the existing robot system because the mechanism requires slight increase in diameter only.

We installed 12 FD torque sensors and all joint torques were precisely measured without suffering from the frictional resistance of the gear reduction system.

B. Walking Posture

The power consumption of a steady walking changes depending on the walking posture even if the hight of the center of gravity and moving velocity of the robot are constant. Especially the power consumption increases when a joint generates a negative power due to gravitational force. Because the negative power has to be compensated by another actuator's positive power. If the robot system has a regenerative circuit, this problem does not matter because the negative power can be stored and utilized for the positive power. However it is not plausible to implement a regenerative system because the control system of a legged

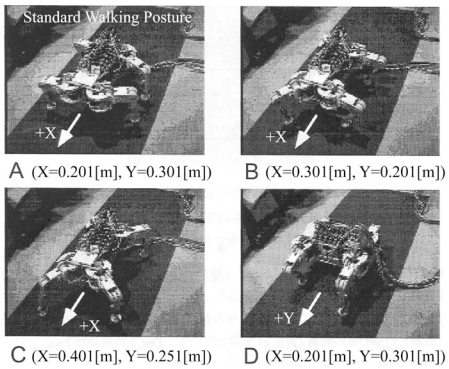

Fig. 7. Walking posture A, B, C and D

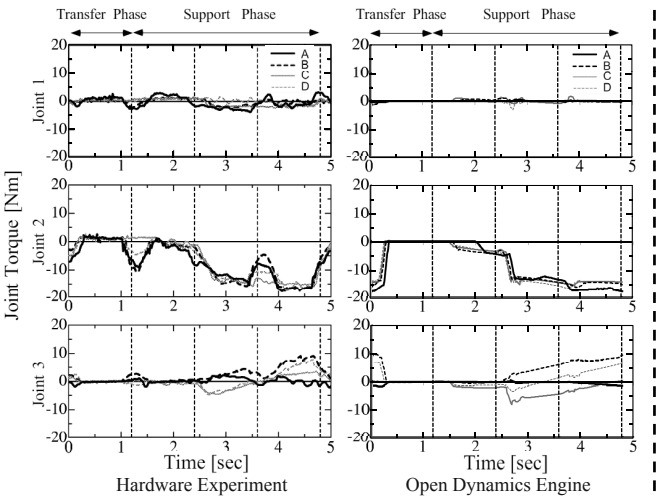

Fig. 8. Joint torque comparison

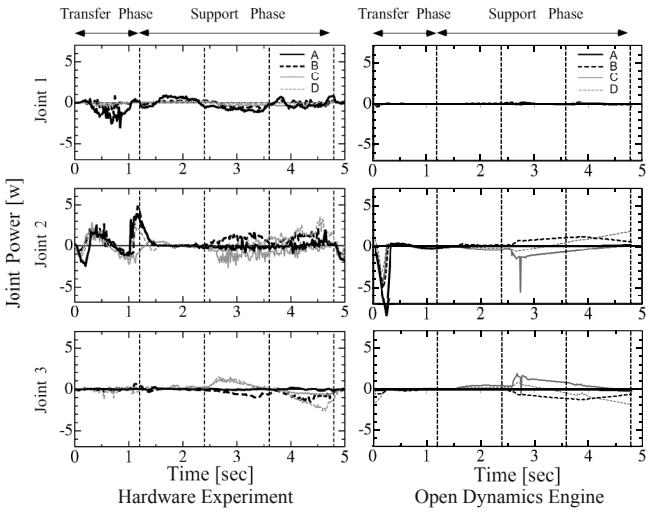

Fig. 9. Joint power comparison

robot is already complicated. Thus the negative power is usually dissipated as heat. Therefore the generation of the negative power should be minimized in order to increase the efficiency of walking. So we propose "Gravitationally Decoupled Actuation (GDA)" to avoid the generation of the negative power[15][16]. TITAN-VIII was designed to satisfy GDA conditions by selecting the walking posture as well as the arrangement of the actuators[12][17].

We measured power consumption during walking with four different walking postures, A, B, C and D, shown in Fig. 7 to verify the change of power consumption. Posture A satisfies GDA conditions. X, Y denotes the initial position of the FL leg in Fig. 7 coordinates where initial positions of four legs are symmetric. Z is constant at -0.243m.

The supporting leg trajectory is a linear trajectory whose stroke is 0.18m and the swing leg trajectory is described by a cycloid, whose maximum height is 0.05m, to minimize leg's touch down velocity. Both trajectories are described in Fig. 7 coordinates. We implemented a crawl gait with a duty factor of 0.75 and a period of 4.8sec and its theoretical walking velocity becomes $0.05m/s$.

V. Comparison with ODE and Hardware

We carried out numerical simulations using ODE under the same conditions of the above-mentioned hardware experiments. The results of numerical simulation using Eqn.(1),(2),(7) are compared with the hardware experiments.

A. Joint Torque and Power

Fig. 8, 9 show the time series of torque and power of the FL leg with four different walking posture A, B, C and D during one gait cycle. The stance phase is from time 0-1.2sec and the swing phase is 1.2-4.8sec, and vertical dashed lines indicate the time of another leg touch down.

As for the joint torques, we observe the peaks in Joint2 at 1.4 and 3.8sec in the hardware experiment while simulated Joint2 torque remains constant. This is because of the difference of the body stiffness. In the hardware system, the main body parts are made of reinforced aluminium sheets and the body leans to the swing leg side due to the deformation of the main body. As a result, relative distance of the ground and the foot of the swing leg decreases and the robot stomps when the swing leg touches the ground. In this situation, Joint2 generates large torque changes. On the other hand, the body structure of ODE is completely rigid and never generates such a deformation. Thus the simulated swing leg gently touches the ground as we planned by using a cycloid trajectory. Except for this difference, ODE generally predict the magnitude and time series of the change in the hardware experimental data regardless of the walking postures.

As for the joint powers, the same discussion can be done for the difference when the legs touch down. The most interesting results are power transitions between Joint2 and Joint3 from 2.4 - 4.8sec in Fig. 9. With the standard posture A which satisfies GDA conditions, the power consumptions of each joints are almost zero. On the contrary, with the other posture B, C and D, Joint2 and Joint3 generate symmetric powers. Joint2 generates positive power when Joint3 generates negative power and these magnitudes are almost the same. In other words, Joint2 power is transmitted and consumed by Joint3 power. This power transition decreases the efficiency of walking which is discussed in the next section. The ODE results of time series of power and its magnitudes nicely follow the experimental data except for the legs' touch down phase.

B. Averaged Power Consumption

This section compares averaged power consumptions with four different walking posture A, B, C and D. We derive the total time-averaged absolute mechanical power $|\overline{P}_{\tau\omega}|$ as follows.

$$|\overline{P}_{\tau\omega}| = \frac{1}{nT}\int_0^{nT} \sum_i \sum_j |P_{\tau\omega_{ij}}| dt \qquad (8)$$

Here, T, n are walking period, number of walking cycle, i, j denote leg and joint number, respectively. Each joint power $P_{\tau\omega_{ij}}$ is calculated by the joint torque multiplied by the joint velocity. The averaged power consumptions $|\overline{P}_{\tau\omega}|$ during 3 periods with four different postures are discussed.

Fig. 10 shows the results. As we designed, the standard

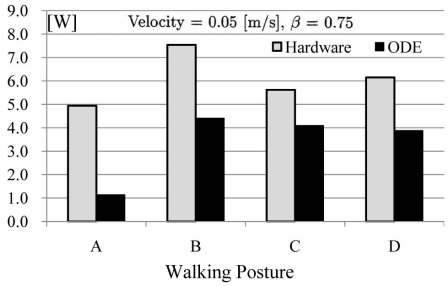

Fig. 10. Comparison of averaged power with different walking posture

walking posture A satisfying GDA conditions consumes the minimum power both in ODE and hardware experiment. Both ODE and hardware experiment results have the same order of single-digit watt and exhibit a similar tendency. However ODE results is smaller than the hardware experiments. The main reason is the power consumptions of swing leg touch down as we mentioned before. Moreover in case of the hardware experiment, there are many inevitable particular error causes such as frictions in a wire driving system for active joints, dragging a relatively heavy tether for controlling the robot and calibration errors in the measurement system. Thus maybe it is reasonable that the results of ODE is smaller than that of hardware experiment.

The reference[13] reports that a total electric power consumption of actuators has similar tendency of the total mechanical power consumption and the electric power is about 3 - 4 times higher than the mechanical power. Although we can not generalize the relationship between mechanical power and electric power because the electric power largely depends on the efficiency of the actuator, this fact suggests that there is a possibility to estimate a total electric power consumption by using ODE before making a real hardware system.

VI. Discussion

In this paper, we introduced a torque measurement of static standing in Section III before discussing the walking experiment. However, in reality, we intensively did the simulated walking at first and struggled with the difference without knowing the problem of joint torque measurement for a long time.

To avoid a waste of time, we propose to do a kind of "calibration" with a basic motion and/or posture to verify the torque calculation before going into a user-specific complicated target motion. In our case, we confirm a static standing with a standard posture whose joint torque is easily calculated by simple equations. We also did a sinusoidal up-and-down motion of the body in the vertical direction. We can derive the potential energy increase from bottom to top and the increased potential energy must be the same as the integration of the joint power. Moreover, integration of the joint powers over a period should be zero because the body height returns to the same starting height. These physical considerations were quite helpful to check whether ODE could accurately calculate the torque or not.

VII. Conclusion

In this paper, the simulated torque and power of Open Dynamics Engine are empirically compared with quadruped walking robot TITAN-VIII hardware measurements. Although the direct acquisitions of simulated joint torques include a large latency in our case, this defect is avoidable by using the joint forces. Careful parameter settings and tunings permit the simulator to match hardware experimental data in our application.

In this paper, the comparison remains in a quasi-static walking and a verification experiment in a dynamic walking is needed. Investigations in various situations such as a walking on rough terrain are also important to generalize the validity of ODE torque/power simulation. A comparison between ODE and alternative candidates forms one of our important future works.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the contribution of Russel Smith and ODE community. The authors would like to thank Prof. Kosei Demura of Kanazawa Institute of Technology for providing us with detailed information about ODE. The authors also would like to thank Prof. Shoichi Hasegawa, Atsushi Tani and Ewerton Ickowzcy of Tokyo Institute of Technology for invaluable discussions about internal calculation of a dynamics simulation.

References

[1] "Havok" http://www.havok.com/
[2] "NVIDIA PHYSX" http://www.nvidia.co.jp/object/physx_new_jp.html
[3] "Game Physics Simulation" http://bulletphysics.org/wordpress/
[4] "Open Dynamics Engine" http://www.ode.org/
[5] "PyODE" http://pyode.sourceforge.net/
[6] "OgreODE" http://www.ogre3d.org/tikiwiki/OgreODE
[7] "Webots 6" http://www.cyberbotics.com/
[8] "Main Page -Simspark-" http://simspark.sourceforge.net/wiki/index.php/Main_Page
[9] "Open Dynamics Engine Manual" http://opende.sourceforge.net/wiki/index.php/Manual_%28All%29
[10] I.A. Sucan, J.F. Kruse, M. Yim, L.E. KavrakiKinodynamic: "Motion Planning with Hardware Demonstrations" , in Proc. IROS'08, pp.1661-1666, 2008.
[11] T. Kamegawa, T. Harada and A. Gofuku: "Realization of cylinder climbing locomotion with helical form by a snake robot with passive wheels", in Proc. ICRA'09, pp.3067-3072, 2009.
[12] K. Arikawa, S. Hirose: "Development of Quadruped Walking Robot TITAN-VIII", in Proc. IROS'96, pp.208-214, 1996.
[13] K. Arikawa: "Study on the Optimized Walking Posture of a Walking Robot", Ph.D Thesis in Tokyo Institute of Technology, Kou No.4321, 2000 (in Japanese).
[14] S. Hirose, K. Kato: "Development of a Float Differential Type Torque Sensor", Robomec'98, 1CI2-6, 1998 (in Japanese).
[15] K. Arikawa, S. Hirose: "Mechanical design of walking machines" Philosophical Transactions of The Royal Society A, Vol.365, pp.171-183, 2007.
[16] S. Hirose et al.: "Quadruped Walking Robots at Tokyo Institute of Technology", IEEE Robotics and Automation Magazine, Vol.16, No.2, pp.104-114, 2009.
[17] K. Arikawa, S. Hirose: "Study of Walking Robot for 3 Dimensional Terrain", in Proc. ICRA'95, pp.703-708, 1995.