2017 IEEE-RAS 17th International Conference on
Humanoid Robotics (Humanoids)
Birmingham, UK, November 15-17, 2017

Torque Based Stabilization Control for Torque Sensorless Humanoid Robots

Hiroto Suzuki[1], Yuya Nagamatsu[1], Takuma Shirai[1],
Shunichi Nozawa[1], Yohei Kakiuchi[1], Kei Okada[1] and Masayuki Inaba[1]

Abstract— **This paper proposes a stabilization control for joint torque controlled humanoid robots without joint torque sensors. The proposed controller's role is to compute all joint torques from the desired positions/postures of floating end effectors, the desired Center Of Gravity (COG), the desired root link posture and the reference Zero Moment Point (ZMP). The controller uses a very simple strategy based on the Single Degree Of Freedom (SDOF) model and statics. First, it calculates the wrenches at the COG or end effectors to keep their positions/postures. Second, it calculates the contact end effector wrenches by solving a constrained optimization problem. Finally, it computes joint torques from all end effector wrenches. Furthermore, we propose a method by which we control joint torques without joint torque sensors by compensating inertia torques and frictions. We implement the stabilization controller and the joint torque controller in a high-power biped robot. We confirm the effectiveness of the proposed controller through balancing and walking experiments on rough terrain.**

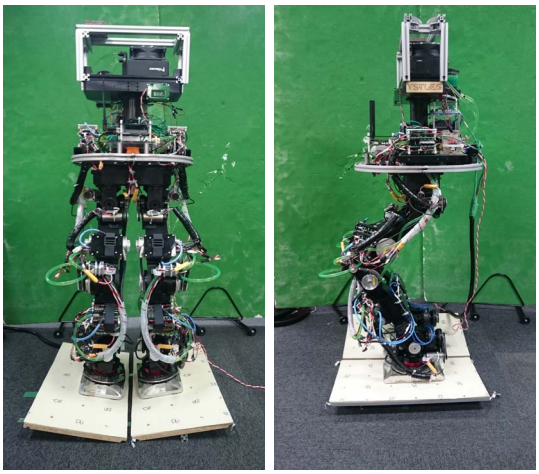


Fig. 1. High-power biped robot used in the experiments

I. Introduction

When humanoid robots perform various tasks instead of humans, it is vitally important that they avoid falling down during performing tasks.

There are many studies about balancing and walking controls for avoiding falling down. Kajita et al. proposed a stabilization control by modifying the desired Zero Moment Point (ZMP) based on the Linear Inverted Pendulum Model and realizing the desired ZMP by using 6-axis force sensors on feet [1]. DRC-HUBO Humanoid Robot also measures the ZMP by using 6-axis force sensors on feet and modifies its pelvis position to realize the desired ZMP [2].

These humanoid robots are joint position controlled robots and modify their desired joint angles depending on a change of 6-axis force sensors values, the IMU value and so on. In case of these robots, there is a possibility that they cannot respond to the instantaneous force and the change of environment because the control flow is

1) change of 6-axis force sensors values, the IMU value
2) regeneration of desired joint angles
3) control of joint angles on motor drivers

and the response delay occurs. Moreover, when links without force sensors, such as shanks and thighs, are pushed, the robots cannot detect external forces and they fall down, or their joints are broken because of the overload.

On the other hand, there are many studies about joint torque controlled humanoid robots [3], [4]. They can rapidly respond to the instantaneous force and the change of the environment because each joint detects the external torque and reacts to it with the motor driver low-latency control.

There are several types of joint torque controlled humanoid robots. Atlas, developed by Boston Dynamics, is a hydraulic humanoid robot. There are pressure sensors in hydraulic cylinders and the robot can measure joint torques. Hyon et al. also developed a hydraulic torque controlled humanoid robot [5]. WALK-MAN, built by IIT, uses a Series Elastic Actuator (SEA) [6]. A torsion bar is attached after the reduction gear and there are two position sensors per joint; one is attached after the reduction gear and the other after the torsion bar. The robot measures the relative deflection of the torsion bar from two position sensors and calculates the joint torque. These robots, which use mechanical elastic parts, can absorb impact forces by the deformation of them. However, using elastic parts causes a lack of controllability for precise motions.

On the other hand, TORO, built by DLR, is a gear-driven humanoid robot with torque sensors and can control joint torques [7]. Nagasaka et al. [8] also proposed the joint torque controller with a gear-driven humanoid robot. It is necessary for sensing joint torques precisely to make torque sensors large, but it is difficult to equip large torque sensors with life-size humanoid robots.

In this paper, we propose a stabilization control for joint torque controlled robots. From the above-mentioned reasons, we developed the joint torque estimation method without torque sensors and implemented it in the high-power biped

[1]Hiroto Suzuki, Yuya Nagamatsu, Takuma Shirai, Shunichi Nozawa, Yohei Kakiuchi, Kei Okada and Masayuki Inaba are with Department of Mechano-Informatics, the University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan suzuki@jsk.imi.i.u-tokyo.ac.jp

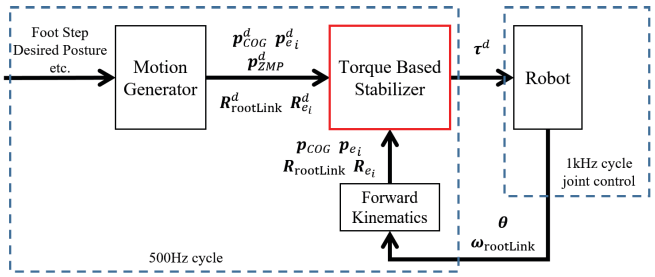


Fig. 2. Overview of the proposed control system for joint torque controlled humanoid robots

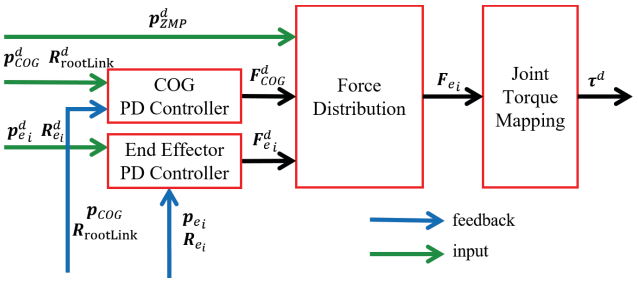


Fig. 3. Components of Torque Based Stabilizer

robot shown in Fig.1 [9]. We implement the proposed controllers in the biped robot. Further, we compare one of the proposed controllers with joint position controlled robots to show high responsiveness against disturbances.

This paper is organized as follows. Section II explains a stabilization control, which calculates the desired joint torques taking balance and various constraints into account. Section III explains the high-power biped robot, the joint torque estimation, and the joint control. Section IV shows balancing and walking experiments with the biped robot. Section V concludes this paper.

II. Torque Based Stabilizer

Fig.2 shows the overview of the proposed system. First, we generate the desired positions/postures of the floating end effectors, the desired Center Of Gravity (COG), the desired root link posture, and the reference ZMP in Motion Generator. For example, we use Kajita's preview controller [10] as Motion Generator for walking. Second, we generate desired joint torques from the computed desired values in the Torque Based Stabilizer.

The major method by which we calculate joint torques is to solve Inverse Dynamics (ID). It is difficult to calculate the analytic solution of ID of a humanoid robot because the root link is floating and multiple end effectors are in contact with the ground, walls or other environments. There are many studies about calculating joint torques [11] [12]. Feng et al. [13] and Herzon et al. [14] calculate not only the joint torques but also the joint accelerations and the contact wrenches by solving a quadratic programming (QP). However, methods using ID and optimization like QP spend high calculation costs because they have many optimization variables. Furthermore, in order to control the joints to follow the desired values precisely, we need to identify mass and inertia of each link precisely.

Therefore, we propose the method of calculating joint torques based on the Single Degree Of Freedom (SDOF) model and statics. Though this method is not suitable for high-speed motions because it ignores each link acceleration, but it can calculate the optimization faster then ID, and we can increase the control cycle. Actually, the calculation of this method took average 150μs on 2.1GHz CPU in Sec. IV experiments. Ott et al. use a similar method [15], but our method can be used for not only balancing but also moving the end effectors or the COG, for example in static walking.

Fig.3 shows the overview of Torque Based Stabilizer. This controller consists of the following four components.

A. End Effector PD Controller

This component calculates the desired wrenches to be generated at the end effectors such as hands, heads, and swing feet to keep their positions and postures.

The desired end effector forces are calculated by using the PD control law.

$$\boldsymbol{f}^d_{e_i} = -\boldsymbol{K}_{f_{e_i}}(\boldsymbol{p}_{e_i} - \boldsymbol{p}^d_{e_i}) - \boldsymbol{D}_{f_{e_i}}(\dot{\boldsymbol{p}}_{e_i} - \dot{\boldsymbol{p}}^d_{e_i}) \quad (1)$$

where $\boldsymbol{K}_{f_{e_i}}, \boldsymbol{D}_{f_{e_i}} \in \mathbb{R}^{3\times3}$ are positive definite gain matrices. $\boldsymbol{p}_{e_i}$ and $\dot{\boldsymbol{p}}_{e_i}$ are the real end effector position and velocity represented in the world coordinate system and $\boldsymbol{p}^d_{e_i}$ and $\dot{\boldsymbol{p}}^d_{e_i}$ are the desired end effector position and velocity.

According to [16], the moment $\boldsymbol{n}$ of the virtual spring which acts to align the real end effector posture $\boldsymbol{R}_{e_i}$ to the desired end effector posture $\boldsymbol{R}^d_{e_i}$ represented in the world coordinate system is the following.

$$\boldsymbol{n} = -2(\delta_{e_i}\boldsymbol{I} + \hat{\boldsymbol{\epsilon}}_{e_i})\boldsymbol{K}_{n_{e_i}}\boldsymbol{\epsilon}_{e_i} \quad (2)$$

where δ_{e_i} and $\boldsymbol{\epsilon}_{e_i}$ are the scalar and vector part of the quaternion of ${\boldsymbol{R}^d_{e_i}}^T\boldsymbol{R}_{e_i}$, and $\boldsymbol{K}_{n_{e_i}} \in \mathbb{R}^{3\times3}$ is a positive definite stiffness matrix. $\boldsymbol{I}$ is an identity matrix and $\hat{\boldsymbol{\epsilon}}_{e_i}$ is the cross product matrix of $\boldsymbol{\epsilon}_{e_i}$. By using this virtual spring, the PD control law for keeping the end effector posture is given by

$$\boldsymbol{n}^d_{e_i} = -2(\delta_{e_i}\boldsymbol{I} + \hat{\boldsymbol{\epsilon}}_{e_i})\boldsymbol{K}_{n_{e_i}}\boldsymbol{\epsilon}_{e_i} - \boldsymbol{D}_{n_{e_i}}(\boldsymbol{\omega}_{e_i} - \boldsymbol{\omega}^d_{e_i}) \quad (3)$$

where $\boldsymbol{D}_{n_{e_i}} \in \mathbb{R}^{3\times3}$ is a positive definite gain matrix and $\boldsymbol{\omega}_{e_i}$ and $\boldsymbol{\omega}^d_{e_i}$ are the real and desired angular velocity represented in the world coordinate system.

B. COG PD Controller

In the same way as calculating the desired end effector wrenches, this component calculates the desired wrench generated at the COG to keep the COG position and the root link postures.

The desired COG force are calculated by using the PD control law and the gravity compensation.

$$\boldsymbol{f}^d_g = -\boldsymbol{K}_{f_g}(\boldsymbol{p}_g - \boldsymbol{p}^d_g) - \boldsymbol{D}_{f_g}(\dot{\boldsymbol{p}}_g - \dot{\boldsymbol{p}}^d_g) - m\boldsymbol{g} \quad (4)$$

where $\boldsymbol{K}_{f_g}, \boldsymbol{D}_{f_g} \in \mathbb{R}^{3\times3}$ are positive definite gain matrices. $\boldsymbol{p}_g$ and $\dot{\boldsymbol{p}}_g$ are the real COG position and velocity represented in the world coordinate system and $\boldsymbol{p}_g^d$ and $\dot{\boldsymbol{p}}_g^d$ are the desired COG position and velocity. m is the total mass of the robot and $\boldsymbol{g}$ is the gravitational acceleration.

The desired moment for keeping the root link posture is calculated by the same law as (3).

$$\boldsymbol{n}_g^d = -2(\delta_g \boldsymbol{I} + \hat{\boldsymbol{\epsilon}}_g)\boldsymbol{K}_{n_g}\boldsymbol{\epsilon}_g - \boldsymbol{D}_{n_g}(\boldsymbol{\omega}_g - \boldsymbol{\omega}_g^d) \tag{5}$$

C. Force Distribution

This component calculates all end effector wrenches to realize the desired wrenches calculated in II-A and II-B components. There are constraints in the end effectors which make contact with the ground, such as the friction, the center of pressure and the direction of the reaction force.

Therefore, we solve this problem as QP. In general, QP is represented with the following problem.

$$\underset{x}{\text{minimize}}\ \frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{g} \tag{6}$$

$$\text{subject to}\ \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{0} \tag{7}$$

where $\boldsymbol{x}$ is the optimization variable. $\boldsymbol{H}$ is a positive definite matrix and $\boldsymbol{g}$ is a vector with the same length as $\boldsymbol{x}$. $\boldsymbol{A}$ is a matrix where its number of columns is the same as the length of $\boldsymbol{x}$. As a QP solver, we use qpOASES [17]. In this controller, the end effector wrenches $\boldsymbol{F}_{e_i} = (\boldsymbol{f}_{e_i}^T, \boldsymbol{n}_{e_i}^T)^T$ are the optimization variables $\boldsymbol{x}$. $\boldsymbol{H}$ and $\boldsymbol{g}$ is obtained from cost functions, and $\boldsymbol{A}$ from constraints. Next, we explain the cost functions and constraints used in this controller.

1) COG Force Cost: The relationship between the COG wrench $\boldsymbol{F}_g = (\boldsymbol{f}_g^T, \boldsymbol{n}_g^T)^T$ and the end effector wrenches $\boldsymbol{F}_{e_i}$ is

$$\boldsymbol{F}_g = \sum_i \boldsymbol{H}_{e_i g}\boldsymbol{F}_{e_i} \tag{8}$$

$$\boldsymbol{H}_{e_i g} = \begin{pmatrix} \boldsymbol{R}_{e_i} & \boldsymbol{O} \\ \hat{\boldsymbol{r}}_{ge_i}\boldsymbol{R}_{e_i} & \boldsymbol{R}_{e_i} \end{pmatrix} \tag{9}$$

where $\boldsymbol{R}_{e_i}$ is the end effector posture and $\boldsymbol{r}_{ge_i}$ is the end effector position with respect to COG.

To minimize the error between $\boldsymbol{F}_g^d$ and $\sum_i \boldsymbol{H}_{e_i}\boldsymbol{F}_{e_i}$, we select the following function as a cost function.

$$\left|\left|\boldsymbol{W}_g\left(\boldsymbol{F}_g^d - \sum_i \boldsymbol{H}_{e_i g}\boldsymbol{F}_{e_i}\right)\right|\right|^2 \tag{10}$$

where $\boldsymbol{W}_g \in \mathbb{R}^{6\times6}$ is a positive definite weight matrix.

2) ZMP Cost: There are many studies about walking or balancing control based on ZMP [1] [2] [11]. To follow the ZMP reference, the cost function which control real ZMP is necessary.

The relationship between ZMP and foot wrenches is

$$p_{\mathrm{zmp_x}} = \frac{\sum_{i=\mathrm{Lfoot,Rfoot}}(-n_{e_i y} + f_{e_i z}p_{e_i x})}{\sum_{i=\mathrm{Lfoot,Rfoot}} f_{e_i z}} \tag{11}$$

$$p_{\mathrm{zmp_y}} = \frac{\sum_{i=\mathrm{Lfoot,Rfoot}}(n_{e_i x} + f_{e_i z}p_{e_i y})}{\sum_{i=\mathrm{Lfoot,Rfoot}} f_{e_i z}} \tag{12}$$

The origin of the end effector is at the center of the contact surface. The matrix representation is

$$\sum_i \begin{pmatrix} 0 & 0 & p_{\mathrm{zmp_x}} - p_{e_i x} & 1 & 0 & 0 \\ 0 & 0 & p_{\mathrm{zmp_y}} - p_{e_i y} & 0 & -1 & 0 \end{pmatrix} \boldsymbol{F}_{e_i} = \boldsymbol{0} \tag{13}$$

For ZMP tracking, we add the following cost function.

$$\left|\left|\boldsymbol{W}_{\mathrm{ZMP}}\sum_i \begin{pmatrix} 0 & 0 & p_{\mathrm{zmp_x}} - p_{e_i x} & 1 & 0 & 0 \\ 0 & 0 & p_{\mathrm{zmp_y}} - p_{e_i y} & 0 & -1 & 0 \end{pmatrix} \boldsymbol{F}_{e_i}\right|\right|^2 \tag{14}$$

where $\boldsymbol{W}_{ZMP} \in \mathbb{R}^{2\times2}$ is a positive definite weight matrix.

3) End Effector Force Cost: To realize the desired end effector wrenches calculated in II-A, we add the cost function which minimize the error between desired and real end effector wrenches.

$$\sum_i \left|\left|\boldsymbol{W}_{e_i}\left(\boldsymbol{F}_{e_i}^d - \boldsymbol{F}_{e_i}\right)\right|\right|^2 \tag{15}$$

where is $\boldsymbol{W}_{e_i} \in \mathbb{R}^{6\times6}$ is positive definite weight matrix.

4) Contact Force Minimizing Cost: We add the cost function to minimize the end effector wrenches.

$$\sum_i ||\boldsymbol{W}_{min_i}\boldsymbol{F}_{e_i}||^2 \tag{16}$$

where $\boldsymbol{W}_{min_i} \in \mathbb{R}^{6\times6}$ is positive definite weight matrix. To prioritize the position and posture of COG, and the end effectors, it is necessary to make $\boldsymbol{W}_{min_i}$ smaller than $\boldsymbol{W}_g$ and $\boldsymbol{W}_{e_i}$

5) Friction Constraint: The end effector which contacts with the environment has a friction constraint. The friction constraint is as follows.

$$-\mu f_{e_i z} \leq f_{e_i x} \leq \mu f_{e_i z} \tag{17}$$

$$-\mu f_{e_i z} \leq f_{e_i y} \leq \mu f_{e_i z} \tag{18}$$

$$f_{e_i z} \geq 0 \tag{19}$$

where μ is a friction coefficient.

Furthermore, the yaw moment is affected by the friction constraint because n_z relates to f_x and f_y. Fig.4 shows the distribution of the end effector force when the yaw moment is the maximum and minimum value. The inequality about yaw moment friction is given by

$$\begin{aligned} n_{e_i z} &\leq \frac{1}{2}(\mu f_{e_i z} + f_{e_i x})y_r + \frac{1}{2}(\mu f_{e_i z} - f_{e_i x})y_l \\ &+ \frac{1}{2}(\mu f_{e_i z} + f_{e_i y})x_f + \frac{1}{2}(\mu f_{e_i z} - f_{e_i y})x_r \end{aligned} \tag{20}$$

$$\begin{aligned} n_{e_i z} &\geq -\frac{1}{2}(\mu f_{e_i z} + f_{e_i x})y_l - \frac{1}{2}(\mu f_{e_i z} - f_{e_i x})y_r \\ &- \frac{1}{2}(\mu f_{e_i z} + f_{e_i y})x_r - \frac{1}{2}(\mu f_{e_i z} - f_{e_i y})x_f \end{aligned} \tag{21}$$

where x_f, x_r, y_l, y_r is the lengths from the center of the end effector to the front, rear, left and right edge. The matrix

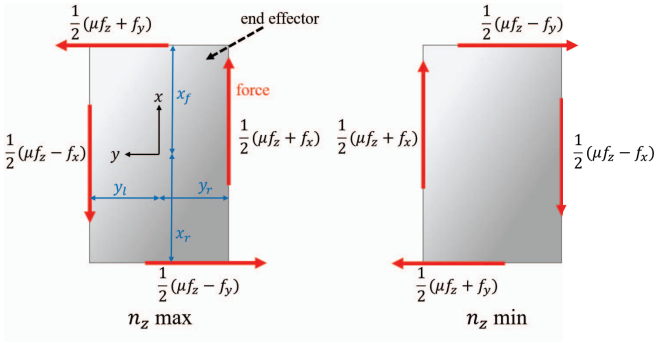


Fig. 4. The rotational friction constraint related to the translational friction

representation is

$$\begin{pmatrix} 1 & 0 & \mu & 0 & 0 & 0 \\ -1 & 0 & \mu & 0 & 0 & 0 \\ 0 & 1 & \mu & 0 & 0 & 0 \\ 0 & -1 & \mu & 0 & 0 & 0 \\ \frac{y_l - y_r}{2} & \frac{x_r - x_f}{2} & \frac{\mu(x_f + x_r + y_l + y_r)}{2} & 0 & 0 & 1 \\ \frac{y_r - y_l}{2} & \frac{x_f - x_r}{2} & \frac{\mu(x_f + x_r + y_l + y_r)}{2} & 0 & 0 & -1 \end{pmatrix} \boldsymbol{F}_{e_i} \geq \boldsymbol{0} \tag{22}$$

6) COP Constraint: The constraint that the Center Of Pressure (COP) is in the supporting polygon is necessary for balancing control. The inequality is given by

$$-x_r \leq \frac{-n_{e_i y}}{f_{e_i z}} \leq x_f \tag{23}$$

$$-y_r \leq \frac{n_{e_i x}}{f_{e_i z}} \leq y_l \tag{24}$$

The matrix representation is

$$\begin{pmatrix} 0 & 0 & x_f & 0 & 1 & 0 \\ 0 & 0 & x_b & 0 & 1 & 0 \\ 0 & 0 & y_l & -1 & 0 & 0 \\ 0 & 0 & y_r & -1 & 0 & 0 \end{pmatrix} \boldsymbol{F}_{e_i} \geq \boldsymbol{0} \tag{25}$$

D. Joint Torque Mapping

This component calculates joint torques to generate the end effector wrenches calculated in II-C.

Wrenches which act on the robot consist of the end effector wrenches, the gravity, and the joint torques, which are balanced. The principle of virtual work leads to the following equation.

$$-\boldsymbol{F}_g^T \delta \boldsymbol{p}_g + \boldsymbol{\tau}^T \delta \boldsymbol{q} + \sum_i \boldsymbol{F}_{e_i}^T \delta \boldsymbol{p}_{e_i} = 0 \tag{26}$$

Furthermore, the relationship between the virtual displacement of the COG position and the end effector position is

$$\delta \boldsymbol{p}_g = \boldsymbol{J}_g \delta \boldsymbol{q} \tag{27}$$

$$\delta \boldsymbol{p}_{e_i} = \boldsymbol{J}_{e_i} \delta \boldsymbol{q} \tag{28}$$

where $\boldsymbol{J}_g, \boldsymbol{J}_{e_i}$ are Jacobian matrices. Finally, the following equation is obtained by combining Eq.(8),(26),(27) and (28)

$$\left\{ -\left(\sum_i \boldsymbol{H}_{e_i g} \boldsymbol{F}_{e_i} \right)^T \boldsymbol{J}_g + \boldsymbol{\tau}^T + \sum_i \boldsymbol{F}_{e_i}^T \boldsymbol{J}_{e_i} \right\} \delta \boldsymbol{q} = 0 \tag{29}$$

By solving this equation for τ, the relationship between joint torques and end effector wrenches is given by

$$\boldsymbol{\tau}_d = \sum_i \left(\boldsymbol{J}_g^T \boldsymbol{H}_{e_i g} - \boldsymbol{J}_{e_i g} \right) \boldsymbol{F}_{e_i} \tag{30}$$

III. Robot and Joint Control

We implement the proposed stabilization control system in the high-power biped robot. In this section, we explain the robot and the joint control implemented in it.

A. High-Power Biped Robot

The high-power biped robot shown in Fig.1 is a 12 DOF (Degree Of Freedom) gear-driven robot. Each joint has a harmonic drive and a pulley and the total reduction ratio is 240. For the joint torque control, we developed a method by which we can estimate the joint torque without joint torque sensors from the motor current, the motor velocity and the acceleration [9]. This joint torque estimation is implemented in the biped robot.

B. Feed Forward Joint Torque Control

We propose the feed forward joint torque control. This method can control joint torques without joint torque sensors. This controller calculates the motor torque to generate the desired joint torque without the joint torque estimation feedback. There are some studies about joint torque controls without torque sensors. Yoshikawa et al. proposed a method to control joint torques with a joint position controller [18]. Katsura et al. developed a disturbance observer and realize a joint torque control without joint torque sensors [19].

The relationship between the joint torque and the motor torque is

$$\tau = R\tau_{\text{motor}} - RI_{\text{motor}}\ddot{\theta}_{\text{motor}} - \tau_{\text{fric}} \tag{31}$$

where τ and τ_{motor} are joint and motor torques. R is the reduction ratio. I_{motor} is the motor inertia. θ_{motor} is the motor angle. τ_{fric} is the friction. We use the Coulomb and viscous friction model as a harmonic drive friction model.

The proposed controller compensates the friction and the inertial torque and generate the joint torque without a joint torque feedback loop. Eq.(31) leads to the following control law.

$$\begin{aligned} \tau_{\text{motor}} &= \frac{1}{R}\tau_d + I_{\text{motor}}\ddot{\theta}_{\text{motor}} + \frac{1}{R}\tau_{\text{fric}} + G_{\theta \text{D}}\dot{\tilde{\theta}}_{\text{motor}} \\ \tilde{\theta}_{\text{motor}} &= \theta_{\text{motor\_ref}} - \theta_{\text{motor}} \end{aligned} \tag{32}$$

The last term prevents the control from becoming unstable. We identify the motor inertia and parameters of the friction model in advance.

C. Joint Compliance Control

There is a possibility that the control using only a torque reference becomes unstable because of the difference between the friction model and the actual friction, the uncertainty of the robot model and so on.

Therefore, we developed Joint Compliance Control combining PD control and LPF (Low Pass Filter) [9]. The control law is

$$\begin{aligned} \tau_{\rm motor} &= G_{\theta \rm P}\tilde{\theta}_{\rm motor} + G_{\theta \rm D}\dot{\tilde{\theta}}_{\rm motor} + G_{\tau \rm P}\tilde{\tau} + G_{\tau \rm D}\dot{\tilde{\tau}} \\ &+ G_{\rm L}{\rm LPF}\left(\tilde{\theta}_{\rm motor} + \frac{\rm G_{\tau P}}{\rm G_{\theta P}}\tilde{\tau}\right) \qquad (33) \\ \tilde{\tau} &= \tau_d - \tau \end{aligned}$$

where $G_{\theta \rm P}, G_{\theta \rm D}, G_{\tau \rm P}, G_{\tau \rm D}, G_{\rm L}$ are gains. The motor torque τ is obtained by using our joint torque estimator [9].

This controller can keep the humanoid posture even if there is an error of the torque estimation or a model error because it includes the position PD control. However, this controller may not be able to generate the desired joint torque if the position gain is large. Therefore, in our biped robot, we make the torque gain large and the position gain small.

In this paper, we use Feed Forward Joint Torque Control for the ankle pitch and roll joints, and Joint Compliance Control for the other joints.

D. Desired Joint Position

The Joint Compliance Control needs not only the desired joint torque but also the desired joint position as a input of the controller. However, The Torque Based Stabilizer calculates only the desired joint torque.

In general joint position controlled robots, a stabilization controller calculates the modification of the desired joint position by using the feedback of 6-axis force sensors and IMU. In the proposed stabilization control, the effect corresponding to the feedback of these sensors appears in the desired joint torques. Thus, it is not necessary to modify the desired joint positions.

For this reason, we input directly to the robot the desired joint positions calculated by the Motion Generator.

IV. Experiments

We conduct two experiments, balancing and walking experiments on a tilting board.

A. Balancing Experiment

We conduct the balancing experiment on a tilting board with the biped robot. The board tilts in one direction like a seesaw. Two contact switches and an acceleration sensor are attached to the board so that we can measure the board inclination. Furthermore, these sensors are connected to the biped robot and we can get board states and robot states synchronously. In this experiment, we put the board to tilt in the pitch axial direction. Then, we apply 80 N·m to the board instantaneously and tilt it to an opposite side.

First, we conducted this experiment with joint position control and Kajita's stabilizer [1]. This stabilizer uses 6-axis force sensors on feet and modifies the robot posture depending on the sensor values. As shown in Fig.5, the biped robot was not able to respond to the change of the board inclination and fell down. Second, we conducted this experiment with joint torque control and the proposed stabilizer. As shown in Fig.6, the biped robot adapted its ankles to the inclination of the board and avoided falling down. Fig.9 shows the pitch posture of root link calculated using the IMU. The proposed controller was able to keep the variation of the pitch posture of root link below 2.1 deg.

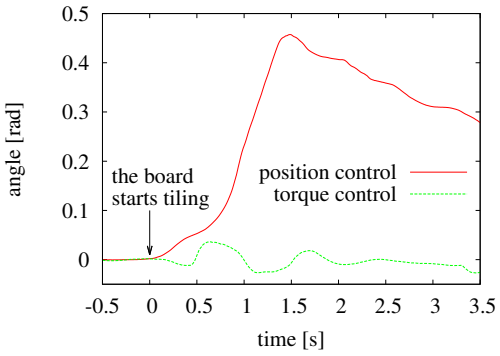


Fig. 9. Root link pitch posture while the board is tilting in the balancing experiment.

B. Walking Experiment

We conduct the walking experiment on rough terrain with the biped robot. There are many studies about a rough terrain walking [20], [21]. The feature of our walking experiment is that the floor tilts during walking. We put the tilting board, which is used in the balancing experiment, on the walking path as a rough terrain. We set the step height as 70 mm and length as 100 mm. As with the balancing experiment, we conduct the walking experiment under two conditions, the joint position control with Kajita's stabilizer and the joint torque control with the proposed stabilizer.

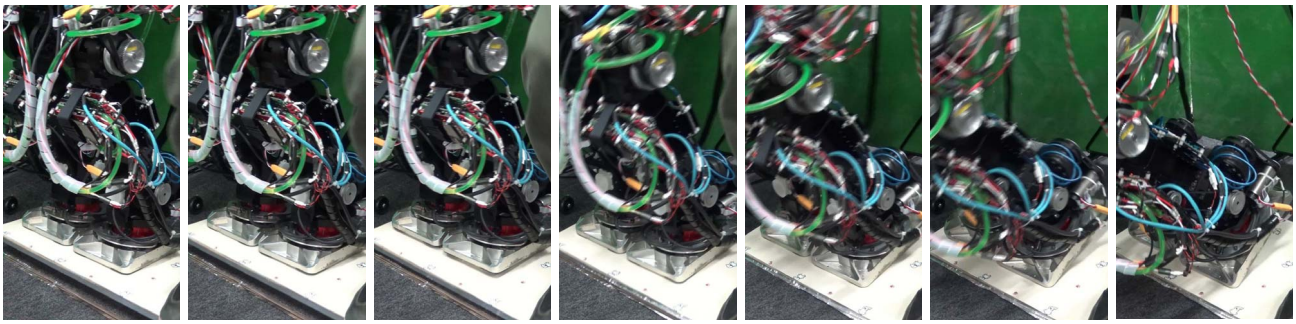
As Fig.7 and Fig.8 show, the robot with the joint position control fell down when the board tilted, but the robot with joint torque control was able to overcome the tilting board. Fig.10 shows the desired and actual COG trajectories while the robot was walking with the joint torque control. The proposed controller was able to cause the COG to follow the desired trajectory even when the board tilted.

V. Conclusions

In this paper, we have proposed a stabilization control for joint torque controlled robots without joint torque sensors.

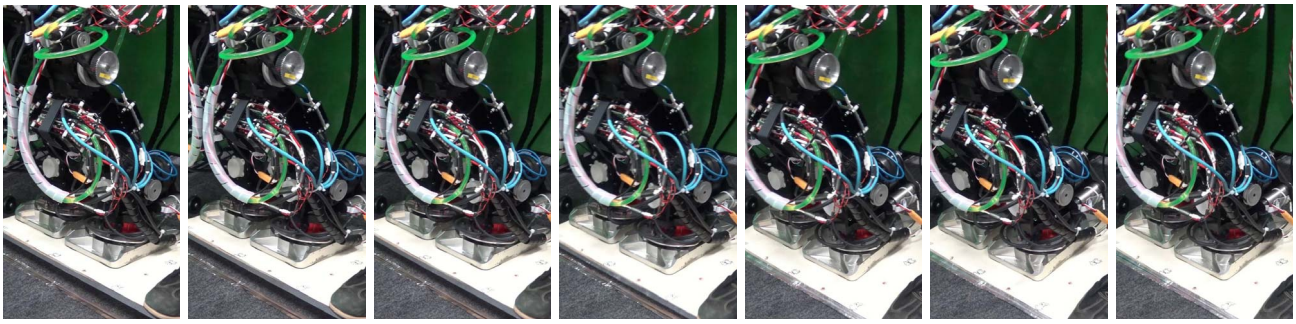
This controller is based on the SDOF model and statics, and can be calculated in high-speed cycle of 150us.This controller consists of four components: End Effector PD Controller, COG PD Controller, Force Distribution and Joint Torque Mapping. End Effector PD Controller and COG PD Controller calculate wrenches generated at end effectors and COG to keep their positions and postures by using the PD control law and the gravity compensation. Force Distribution calculates contact wrenches taking various constraints into account by using a QP. Joint Torque Mapping calculates joint torques to generate the desired wrenches.

We implemented this stabilization controller with the high-power biped robot. In order to control joint torques without joint torque sensors, we propose Feed Forward Joint Torque Control. This controller compensates the inertia torque and



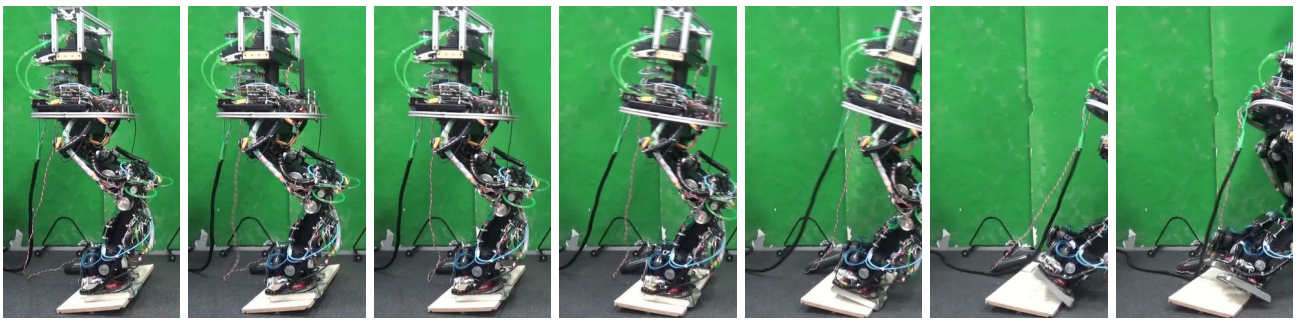
(a) $t = 0.0\,\mathrm{s}$ (b) $t = 0.2\,\mathrm{s}$ (c) $t = 0.4\,\mathrm{s}$ (d) $t = 0.6\,\mathrm{s}$ (e) $t = 0.8\,\mathrm{s}$ (f) $t = 1.0\,\mathrm{s}$ (g) $t = 1.2\,\mathrm{s}$

Fig. 5. Joint position control on tilting the board: The modification of joint angles did not follow rapidly the change of the board inclination. The heel left from the board at $t = 1.0$s and the robot fell down.



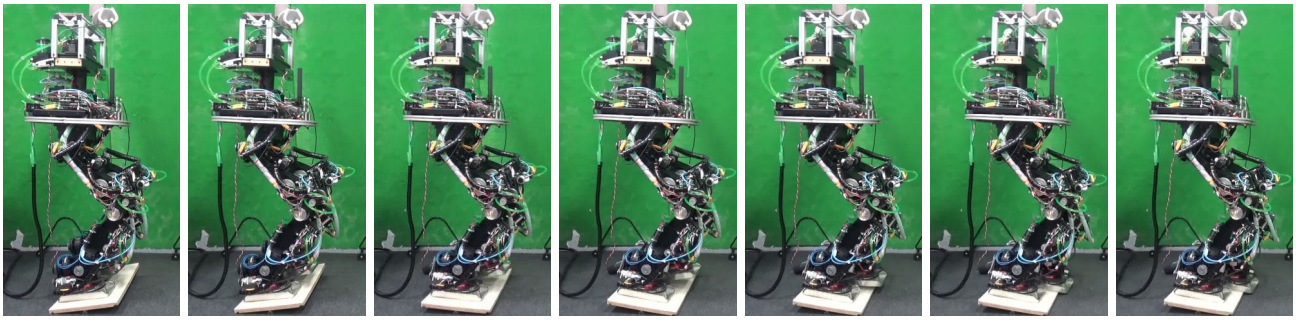
(a) $t = 0.0\,\mathrm{s}$ (b) $t = 0.2\,\mathrm{s}$ (c) $t = 0.4\,\mathrm{s}$ (d) $t = 0.6\,\mathrm{s}$ (e) $t = 0.8\,\mathrm{s}$ (f) $t = 1.0\,\mathrm{s}$ (g) $t = 1.2\,\mathrm{s}$

Fig. 6. Proposed balancing control on tilting the board with joint torque control: The robot changed joint angles depending on the change of the board inclination and was able to keep its posture.



(a) $t = 0.0\,\mathrm{s}$ (b) $t = 0.5\,\mathrm{s}$ (c) $t = 1.0\,\mathrm{s}$ (d) $t = 1.5\,\mathrm{s}$ (e) $t = 2.0\,\mathrm{s}$ (f) $t = 2.5\,\mathrm{s}$ (g) $t = 3.0\,\mathrm{s}$

Fig. 7. The biped robot walked with joint position control. The robot fell down during moving its COG.



(a) $t = 0.0\,\mathrm{s}$ (b) $t = 0.5\,\mathrm{s}$ (c) $t = 1.0\,\mathrm{s}$ (d) $t = 1.5\,\mathrm{s}$ (f) $t = 2.0\,\mathrm{s}$ (f) $t = 2.5\,\mathrm{s}$ (g) $t = 3.0\,\mathrm{s}$

Fig. 8. The biped robot walked with joint torque control. The board tilted while the robot was moving its left leg. The robot avoided falling down.

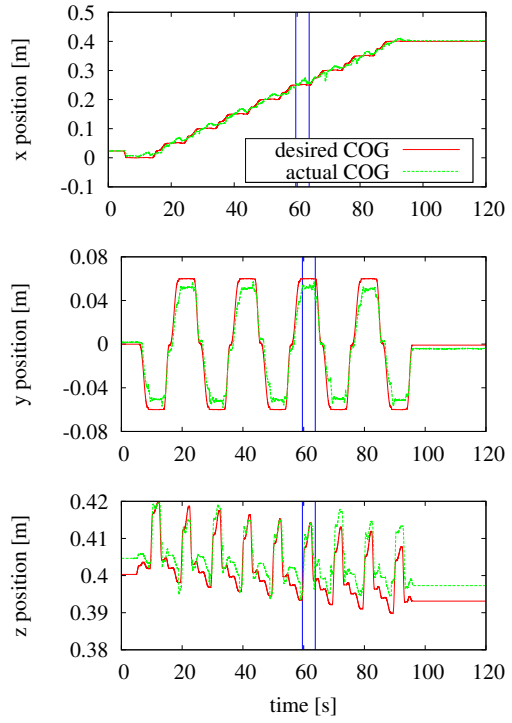


Fig. 10. Desired COG trajectory and actual COG trajectory in the walking experiment: the board started tilting at 59.8s and finished tilting at 63.8s (blue line).

friction, and calculates the desired motor torque. We implement this control in ankle joints and Joint Compliance Control in the other joints.

We evaluated the proposed controller through balancing and walking experiments on a tilting board. In these experiments, the robot with proposed controller was able to avoid falling down and keep its position and posture regardless of whether it was stopping or moving.

In the future work, we desire to expand this controller for more fast motions like dynamic walking or running.

REFERENCES

[1] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi. Biped walking stabilization based on linear inverted pendulum tracking. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4489–4496, Oct 2010.

[2] M. Zucker, S .Joo, M. X. Grey, C. Rasmussen, E. Huang, M. Stilman, and A. Bobick. A General-purpose System for Teleoperation of the DRC-HUBO Humanoid Robot. *Journal of Field Robotics*, Vol. 32, No. 3, pp. 336–351, 2015.

[3] G. Cheng, S. H. Hyon, J. Morimoto, A. Ude, J. G. Hale, G. Colvin, W. Scroggin, and S. C. Jacobsen. CB: a humanoid research platform for exploring neuroscience. *Advanced Robotics*, Vol. 21, No. 10, pp. 1097–1114, 2007.

[4] M. A. Hopkins, R. J. Griffin, A. Leonessa, B. Y. Lattimer, and T. Furukawa. Design of a compliant bipedal walking controller for the DARPA Robotics Challenge. In *Proceedings of the 2015 IEEE-RAS International Conference on Humanoid Robots*, pp. 831–837, Nov 2015.

[5] S. H. Hyon, D. Suewaka, Y. Torii, and N. Oku. Design and Experimental Evaluation of a Fast Torque-Controlled Hydraulic Humanoid Robot. *IEEE/ASME Transactions on Mechatronics*, Vol. 22, No. 2, pp. 623–634, April 2017.

[6] N. G. Tsagarakis, D. G. Caldwell, F. Negrello, W. Choi, L. Baccelliere, V.G. Loc, J. Noorden, L. Muratore, A. Margan, A. Cardellino, L. Natale, E. Mingo Hoffman, H. Dallali, N. Kashiri, J. Malzahn, J. Lee, P. Kryczka, D. Kanoulas, M. Garabini, M. Catalano, M. Ferrati, V. Varricchio, L. Pallottino, C. Pavan, A. Bicchi, A. Settimi, A. Rocchi, and A. Ajoudani. WALK-MAN: A High-Performance Humanoid Platform for Realistic Environments. *Journal of Field Robotics*, 2017.

[7] J. Englsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid, and A. Albu-Schäffer. Overview of the torque-controlled humanoid robot TORO. In *Proceedings of the 2014 IEEE-RAS International Conference on Humanoid Robots*, pp. 916–923, Nov 2014.

[8] K. Nagasaka, Y. Kawanami, S. Shimizu, T. Kito, T. Tsuboi, A. Miyamoto, T. Fukushima, and H. Shimomura. Whole-body cooperative force control for a two-armed and two-wheeled mobile robot using Generalized Inverse Dynamics and Idealized Joint Units. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pp. 3377–3383, May 2010.

[9] Y. Nagamatsu, T. Shirai, H. Suzuki, Y. Kakiuchi, K. Okada, and M. Inaba. Distributed Torque Estimation toward Low-Latency Variable Stiffness Control for Gear-Driven Torque Sensorless Humanoid. In *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5239–5244, Sep 2017.

[10] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1620–1626, Sept 2003.

[11] S. H. Hyon and G. Cheng. Passivity-based full-body force control for humanoids and application to dynamic balancing and locomotion. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4915–4922, Oct 2006.

[12] L. Righetti, J. Buchli, M. Mistry, and S. Schaal. Inverse dynamics control of floating-base robots with external constraints: A unified view. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, pp. 1085–1090, May 2011.

[13] S. Feng, X. Xinjilefu, W. Huang, and C. G. Atkeson. 3D walking based on online optimization. In *Proceedings of the 2013 IEEE-RAS International Conference on Humanoid Robots*, pp. 21–27, Oct 2013.

[14] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, and S. Schaal. Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics. In *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 981–988, Sept 2014.

[15] C. Ott, M. A. Roa, and G. Hirzinger. Posture and balance control for biped robots based on contact force optimization. In *Proceedings of the 2011 IEEE-RAS International Conference on Humanoid Robots*, pp. 26–33, Oct 2011.

[16] F. Caccavale, C. Natale, B. Siciliano, and L. Villani. Six-DOF impedance control based on angle/axis representations. *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 2, pp. 289–300, Apr 1999.

[17] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, Vol. 6, No. 4, pp. 327–363, 2014.

[18] T. Yoshikawa and O. Khatib. Compliant humanoid robot control by the torque transformer. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3011–3018, Oct 2009.

[19] S. Katsura, Y. Matsumoto, and K. Ohnishi. Modeling of force sensing and validation of disturbance observer for force control. *IEEE Transactions on Industrial Electronics*, Vol. 54, No. 1, pp. 530–538, Feb 2007.

[20] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson. Optimization based full body control for the atlas robot. In *Proceedings of the 2014 IEEE-RAS International Conference on Humanoid Robots*, pp. 120–127, Nov 2014.

[21] H. Hirukawa, S. Hattori, S. Kajita, K. Harada, K. Kaneko, F. Kanehiro, M. Morisawa, and S. Nakaoka. A Pattern Generator of Humanoid Robots Walking on a Rough Terrain. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pp. 2181–2187, April 2007.