2008 8th IEEE-RAS International Conference on Humanoid Robots
December 1 ~ 3, 2008 / Daejeon, Korea

Stabilization Control for Humanoid Robot to Walk on Inclined Plane

Yong-Duk Kim [1], In-Won Park [2], Jeong-Ki Yoo [2], and Jong-Hwan Kim [2]

[1]*Agency for Defense Development, Republic of Korea,* $yd.kim@add.re.kr$

[2]*Dept. of EECS, KAIST, Republic of Korea,* $\{iwpark, jkyoo, johkim\}@rit.kaist.ac.kr$

***Abstract*— In this paper, a novel stabilization control is proposed for humanoid robot to walk dynamically on an inclined plane. Online walking control is indispensable to obtain a stable dynamic walking, even if the walking pattern is provided based on a zero moment point and an angular momentum because modeling errors and external disturbances, which are not expected in the modeling stage, may exist in the actual condition. Considering these issues, this paper proposes a stabilization control, which consists of landing force controller, posture controller and walking pattern generator to walk on the inclined plane. The proposed control does not require a complex dynamic equation of robot and the adjustment of control parameters because it is based on time-domain passivity approach. Moreover, it can guarantee the stability of the controller without requiring any dynamic model information. The proposed control is verified through dynamic walking simulations using Webots simulator.**

I. INTRODUCTION

There has been a lot of research to develop a humanoid robot which is similar to human beings, both morphologically and functionally [1]–[3]. Walking trajectory generation [4]–[7] and online (real-time) balance control during walking [8]–[10] are still fundamental research issues for human-like walking. Research has been going on for the issue that humanoid robot would coexist with humans. In this situation, the ability to move around within various environments is essential. The various environments include inclined surface as well as flat surface. For the case of humanoid robot walking on the inclined surface, there should be more considerations compared to wheel-type mobile robots because it has to walk using two legs. It means that its posture should be controlled to be stable and also its walking pattern should be adjusted properly according to the inclined surface.

When human walks on an inclined environment, stable walking can be obtained by simply maintaining the body balance if the inclination of ground is small. As the uneven level of ground gradually increases, however, he or she starts to modify the walking pattern according to the level. The identical walking strategy can be applied to a humanoid robot. In order for a robot to walk on inclined plane, initial strategy to keep stable posture as well as walking pattern generation adjusted to this walking environment are required.

In this paper, a novel stabilization control is proposed for humanoid robot to walk dynamically on inclined plane. Online walking control is indispensable to obtain the stable dynamic walking, even if the walking pattern is provided based on zero moment point (ZMP) and angular momentum. This is because modeling errors and external disturbances, which are not expected in the modeling stage, may exist in the actual condition. Considering this issue, this paper proposes a stabilization control scheme, which consists of landing force controller, posture controller and walking pattern generator.

Although walking pattern is designed considering the ground reaction force, the problem is the force does not act on the robot with expected magnitude at exact time because of modeling errors and external disturbances. This force either accelerates or rotates robot's center of gravity (COG) such that it alters robot's ZMP and angular momentum, which may lead to instability. Thus, reducing landing force is one of the important factors for stable walking. For this purpose, landing force controller was proposed [11] based on time-domain passivity approach [12]–[14].

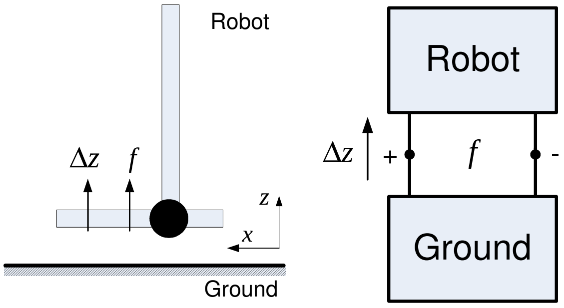
Posture control is also essential for the stabilization controller. It is required to compensate unexpected inclination of robot's posture caused by external disturbances. Posture control algorithm is designed by using time-domain passivity approach, which controls robot's posture according to resulted torque and surface inclination. Similarly as in the landing force controller, the posture controller uses the network modeled system that consists of robot and environment. Note that it calculates the energy of network system to compensate the surface inclination.

When it walks on a slope, landing force controller and posture controller should be operated at the same time. In addition, walking pattern should be modified according to the inclination of the plane in order to walk properly with stability. In this paper, the robot is modeled as an inverted pendulum of a point mass.

Proposed stabilization control does not require complex dynamic equations of robot and adjustment of control parameters. Moreover, it can guarantee the stability of the controller without needing any dynamic model information because it is based on passivity property. The proposed control scheme is verified through dynamic walking simulations using Webots simulator.

The remainder of this paper is organized as follows. In Section II, time-domain passivity and landing force controller are briefly reviewed. In Section III, the posture controller is proposed using time-domain passivity approach. Section IV presents the overall control system and its effectiveness is

978-1-4244-2822-9/08/$25.00 ©2008 IEEE



(a) (b)

Fig. 1. Robot's foot model. (a) Robot's foot and ground surface. (b) One-port network model of the robot's foot and ground surface.

demonstrated through computer simulations and experiments of walking on inclined plane in Section V. Finally, conclusions follow in Section VI.

II. PRELIMINARIES

This section briefly summarizes time-domain passivity and a landing force controller [11], which is to be integrated into a stabilization control in Section IV.

A. Time-Domain Passivity

Robot's foot and ground can be modeled as a one-port network system, where both input and output are connected to each other as shown in Fig. 1(a). The impedance of ground is zero when the foot is in swing mode, whereas it has a certain value when the foot is in contact with ground. The sign conventions of force and velocity are positive in the upper direction such that the energy is positive when a power enters into the system port of robot's foot as shown in Fig. 1(b). The passivity of robot's foot system can be defined as follows [11]:

$$\begin{aligned} E(k) &= \sum_{j=0}^{k} F(j)(z(j) - z(j-1)) + E(0) \\ &= \sum_{j=0}^{k} F(j)\Delta z(j) + E(0) \quad \geq 0 \end{aligned} \tag{1}$$

where j is the time index, z is the height of foot position, and Δz is the difference between two consecutive sampled data of z. The system is assumed to have an initial stored energy of $E(0)$.

When the robot's foot physically absorbs contact force, a connected system between robot's foot and ground is passive. However, the connected system becomes active when the robot's foot kicks off the surface due to big landing force. To alleviate this landing force, a landing force controller is proposed in the previous research. Once the passivity is provided, the stability of system can be also guaranteed because the passivity is a sufficient condition of stability [12].

B. Landing Force Controller

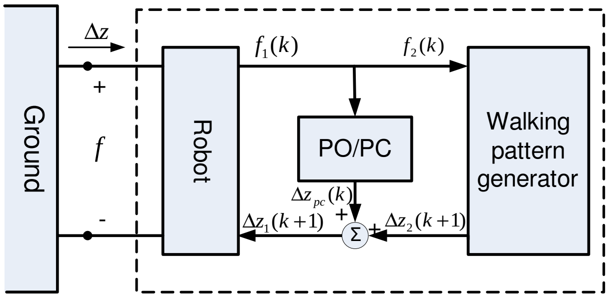


Fig. 2. One-port network of robot's landing force controller.

Fig. 2 shows the one-port network of robot's foot system, which is composed of a foot position controller and a walking pattern generator. The proposed landing force controller based on time-domain passivity approach consists of passivity observer (PO) and passivity controller (PC), which respectively monitors and controls the input/output energy flow between the robot's foot and the ground. Using this concept, the PO/PC algorithm is developed for the one-port robot's foot [11].

In Fig. 2, a landing force, f, is measured by force sensors on the sole of foot. The modified foot position, z_1, is calculated from the originally planned height trajectory of foot position, z_2, without considering the landing force from ground and the output of the PO/PC, Δz_{pc}. PO computes the amount of energy flow using the landing force and foot position as follows:

$$W(k) = W(k-1) + f_1(k)(z_1(k) - z_1(k-1)) \tag{2}$$

$$W_o(k+1) = W(k) + f_1(k)(z_2(k+1) - z_1(k)) \tag{3}$$

where $W(k)$ is the total output energy from 0 to k, and $W_o(k+1)$ is the prediction of total output energy in one-step-ahead. The last term of (3) is the estimation of one-step-ahead output energy, which is the output energy from k to $k+1$. Note that the planned position $z_2(k+1)$ is available at each step k.

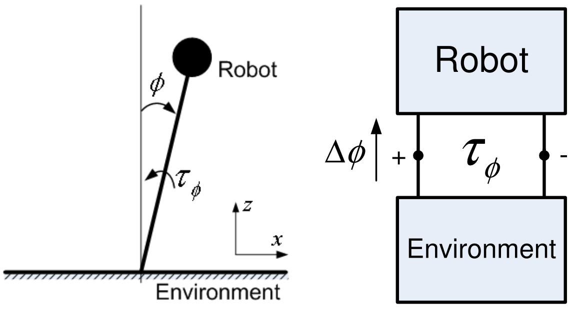
If PO can predict whether the system at the next step will be passive or not at the current step k, then PC can modify the desired position at the next step $(k+1)$ to make the system passive. In other words, PC absorbs the exact amount of net energy output, if any, measured by the PO at each time sample. Similarly, the stability of landing force controller can be guaranteed by the passivity property.

III. POSTURE CONTROLLER

This section proposes a posture controller to compensate robot's posture by using time-domain passivity approach. Similar to the landing force controller, the posture controller uses a networked model system that consists of robot and environment.

A. Robot's Posture Model

To control robot's posture, torque and tilted angular velocity are modeled as the parameters of a network system. Fig. 3(a) and Fig. 3(b) show the inverted pendulum model of robot and its corresponding one-port network system, respectively. The robot's tilted angular value, ϕ, is measured by using a posture sensor, where $\Delta\phi$ is the difference between two consecutive



(a) (b)

Fig. 3. Robot's posture model along pitching axis. (a) Inverted pendulum model with the sign convention of torque and angular velocity. (b) One-port network model of the robot's tilted posture and environment.

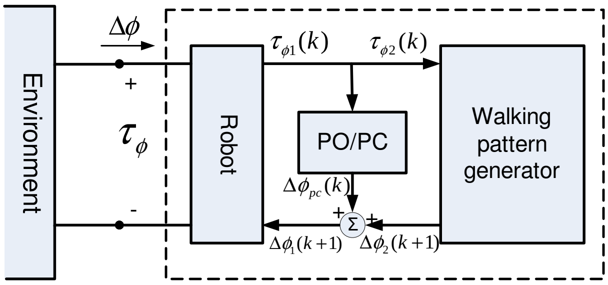


Fig. 4. One-port network of robot's posture controller.

sampled data of ϕ. The sign conventions of torque and angular velocity are defined as in Fig. 3(a) such that an energy is zero when the robot stands straight and becomes negative when the robot is tilted. An angle follows the right-hand rule, where the torque is defined as the opposite to tilting direction in order to make the energy negative when the robot is tilted. This rule can be similarly defined and utilized in rolling axis as well.

In the model, robot's COG is represented based on a supporting leg. If the supporting leg is fixed as the starting point, the position and mass of robot's each joint are defined as p_i and m_i, respectively. Then, the position of robot's COG, p_{cog}, can be calculated as follows:

$$p_{cog} = \frac{\sum_i m_i p_i}{M} \tag{4}$$

where

$$M = \sum_i m_i. \tag{5}$$

From (4), the torque generated on the center of supporting leg due to the inclination of robot's body can be calculated as follows:

$$\tau_\phi = -Mg|p_{cog}|\sin\phi \tag{6}$$

where $|p_{cog}|$ is the length from the supporting point to the COG. Note that the negative sign is used to represent the opposite to tilting direction so that the energy becomes negative as the rate of change in robot's tilting angle increases. Similarly as in Section II, the passivity of robot's tilted posture system can be defined as follows:

$$\begin{aligned} E(k) &= \sum_{j=0}^{k} \tau_\phi(j)(\phi(j) - \phi(j-1)) + E(0) \\ &= \sum_{j=0}^{k} \tau_\phi(j)\Delta\phi(j) + E(0) \quad \geq 0. \end{aligned} \tag{7}$$

Since the environment can be considered as an intrinsically passive system, the connected network system (between robot's posture and environment) can be passive if the one-port network of robot's tilted posture is passive. In the proposed model, the system will release the energy to the environment when the robot's posture is tilted and resulted in falling down. As the robot stands straight, the system absorbs the energy from the environment and the energy gradually increases from the negative value. The value becomes zero once the robot stands straight.

B. Controlling Robot's Posture using Time-domain Passivity

Fig. 4 shows the one-port network, which compensates a tilted posture of robot in pitching axis. τ_ϕ $(= \tau_{\phi 1} = \tau_{\phi 2})$ represents the exerted torque on robot. When the robot walks or stands on an inclined surface without knowing the slope, it cannot walk properly with the originally planned walking trajectory (ϕ_2). Thus, the posture controller calculates $\Delta\phi_{pc}$ in order to compensate the inclination caused by the slope. The controller then determines a modified trajectory, $\Delta\phi_1$, by adding $\Delta\phi_{pc}$ to the originally planned one.

PO calculates an energy by using the exerted torque on robot and the tilted angle as follows:

$$W_\phi(k) = W_\phi(k-1) + \tau_{\phi 1}(k)(\phi_1(k) - \phi_1(k-1)) \tag{8}$$

$$W_{o\phi}(k+1) = W_\phi(k) + \tau_{\phi 1}(k)(\phi_2(k+1) - \phi_1(k)) \tag{9}$$

where $W_\phi(k)$ is the total output energy from 0 to k, and $W_{o\phi}(k+1)$ is the predicted output energy in one-step-ahead. The last term of (9) is the estimated output energy in one-step-ahead, which is the output energy from k to $k+1$. Similarly as in the landing force controller, PO calculates the energy by using the exerted torque on robot and the tilted angle, and then, PC exactly dissipates a generated active energy, if any. The proposed posture control algorithm in pitching axis is presented as follows:

1) $\tau_{\phi 1}(k) = (\tau_{\phi 2}(k))$ is the input.
2) $\Delta\phi_1(k) = \phi_1(k) - \phi_1(k-1)$,
   $\Delta\phi_2(k+1) = \phi_2(k+1) - \phi_1(k)$.
3) $\Delta\phi_2(k)$ is the output of one-port network.
4) $\Delta\phi_{pc} = 0$ and ends PC, if $|\phi| < \phi_{MARGIN}$.
5) $W_\phi(k) = W_\phi(k-1) + \tau_{\phi 1}(k)\Delta\phi_1(k)$ is the output energy at step k.
6) $W_{o\phi}(k+1) = W_\phi(k) + \tau_{\phi 1}(k)\Delta\phi_2(k+1)$ is the predicted energy at step $k+1$.

7) PC output for making the system passive is calculated as follows:

$$\Delta\phi_{pc} = \begin{cases} \frac{-W_{o\phi}(k+1)}{\tau_1(k)}, & \text{if } W_{o\phi}(k+1) < 0 \\ 0, & \text{if } W_{o\phi}(k+1) \geq 0 \end{cases}.$$

8) The modified angle ($\phi_1(k+1)$) is calculated from $\Delta\phi_1(k+1) = \Delta\phi_2(k+1) + \Delta\phi_{pc}(k)$.

Updated ϕ represents the tilted angle of robot's COG, which is used to calculate the angle of each link through inverse kinematics. In the proposed PC, the stable boundary ϕ_{MARGIN} is used to define a region to stop PC. In the actual implementation, however, a robot's control value, ϕ, may oscillate when the energy oscillates around 0 due to the error caused by sensor and model. To solve this oscillation problem in the proposed PC, the stable boundary, ϕ_{MARGIN} is employed.

By using the identical method, the tilted angle of robot's posture in rolling axis, θ, can be controlled as well. The overall posture controller operates the two controllers in parallel to control both ϕ and θ so that it is able to maintain the posture properly in all directions.

IV. STABILIZATION CONTROL

Fig. 5 shows a stabilization control, which is mainly composed of landing force controller, posture controller, and walking pattern generator. If a landing moment is detected from force sensing resistor (FSR), the robot's leg has come to a contact phase, and then, the landing force controller immediately compensates for this landing force. The controller based on PO/PC in Section II is used to absorb the contact force. A low-pass filter is used to get a reliable force value because the measured values from FSR have high frequency noises.

Both position and angle of robot's COG are controlled through the posture controller. This paper assumes that robot's tilted posture depends on the surface inclination. According to the surface inclination, the robot's posture along with robot's foot should be adjusted by the posture controller, which is described in Section III. In other means, the adjustment of robot's foot angle is attained through the posture controller.

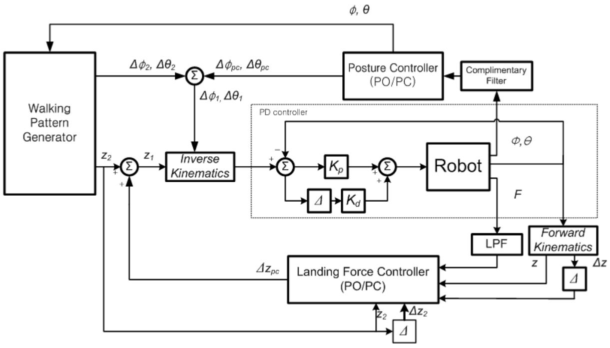


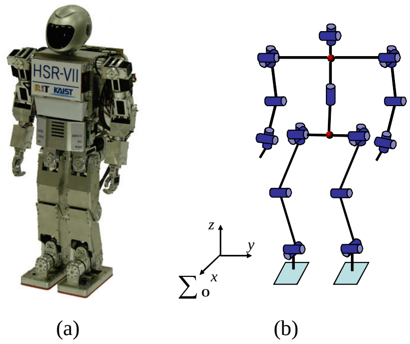
Fig. 5. Block diagram of stabilization control.

Robot's inclination is measured by using a posture sensor, which is located on the center of robot's pelvis link.

The walking pattern generator modifies a walking pattern according to the inclination of surface. During the swing and support phases, the robot legs are controlled by PD controller to follow the desired trajectory, which is provided in the walking pattern generator. In Fig. 5, K_p and K_d are the coefficients of PD controller, and Δ represents the difference operator between two consecutive sampled data.

V. SIMULATION RESULTS

For a simulation, Webots simulator was employed, which is a 3D mobile robotics simulation software. Both size and physical quantities of humanoid robot in the simulation were almost identical to a real small-sized humanoid robot, HanSaRam-VII, developed in RIT Lab, KAIST (Fig. 6).



(a) (b)

Fig. 6. Experiment platform. (a) HanSaRam-VII. (b) Configuration.

To measure force and posture of the robot, 4 FSRs were equipped on the sole of each foot and posture sensor, which is composed of accelerometer, gyro and complementary filter, was installed on the robot's pelvis link. The stable boundary of posture controller in pitching axis, ϕ_{MARGIN}, was set to 0.2 degrees. The coefficients of PD controller in the stabilization control, K_p and K_d, were set to 50 Nm/rad and 5000 Nm/(rad/sec), respectively.

The effectiveness of the posture controller was verified through posture balancing simulation on inclined plane. In the simulations, the surface was flat and the slope was varied from -15 degrees to 15 degrees by a sine function of $15\sin\left(\frac{2\pi}{10}t\right)$. All the results were measured and plotted for 10 seconds.

A. *Results without using posture controller*

Fig. 7 shows the trajectories of surface inclination and robot's posture angle in pitching axis when the posture controller was not applied. Initially, the robot was able to maintain its balance at a small angle due to the size of sole of foot and the frictional effects between foot and surface. As the slope increases, however, the robot could not maintain its balance and eventually fell down.

B. *Results with using posture controller*

Fig. 8 shows the simulation result on the surface with varying slope along pitching axis when the proposed posture

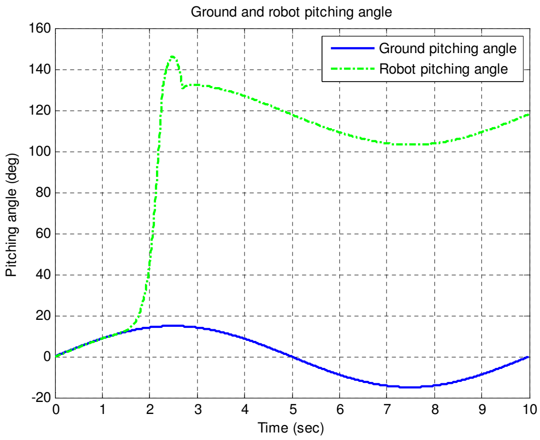
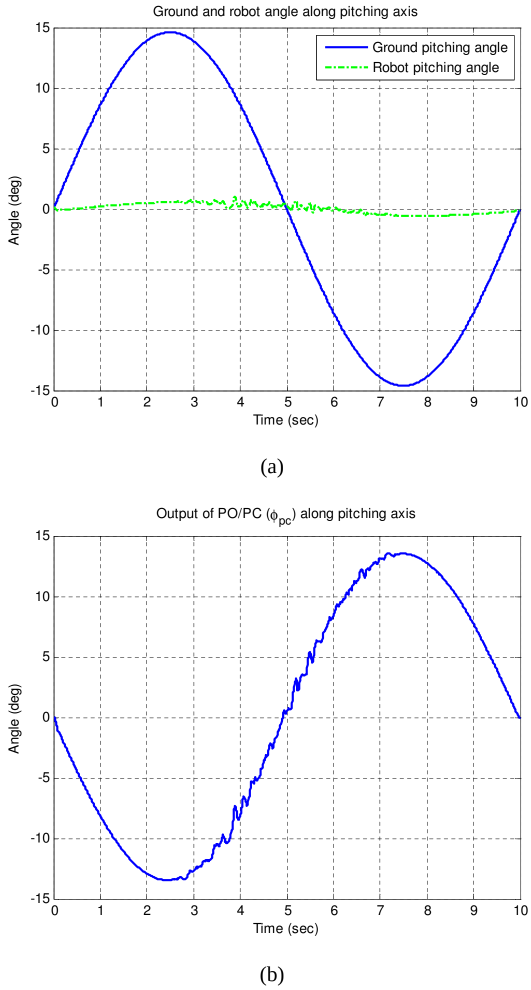


Fig. 7. Results without using posture controller by varying slope in pitching axis.



(a)

(b)

Fig. 8. Results with using posture controller by varying slope in pitching axis. (a) Ground and robot angles in pitching axis. (b) Output of PO/PC (ϕ_{pc}).

controller was applied. As shown in Fig. 8(a), robot's posture was stably maintained during the slope variation because the output of proposed posture controller completely compensated the surface inclination, which is plotted in Fig. 8(b).

Due to the presence of proposed posture controller, the robot never collapsed and always maintained stable posture even when the surface slope was changed with respect to time. When the posture controller was applied, the robot was impossible to maintain its balance with the stability on a higher slope of 18 degrees or a lower slope of -18 degrees. Similar to the results in pitching axis, the robot could keep its posture against the change of surface inclination in the case of rolling axis.

C. *Results with using stabilization control*

The effectiveness of stabilization control was verified through walking simulation on inclined plane. In the simulation, the biped robot walked with a speed of 5.3 cm/s and a step length of 4.0 cm. Double and single support phases of each step were 0.15 s and 0.60 s, respectively.

Fig. 9 and Fig. 10 show the snapshots of robot walking up and down the slope of 10 degrees, respectively. The robot initially stood up straight and bent its knee to lower the COG. Once the slope of plane changed, the posture controller was activated to maintain its balance by using a posture sensor located on the center of pelvis. As the robot walked up the slope, the landing force controller was also activated to compensate landing force measured from the FSRs. Note that the maximum slope that the robot can either walk up or down was 12 degrees in the simulator.

When the stabilization control was not used, the robot could not balance its posture and decrease the impulsive landing force, which result impossible to walk on the inclined plane. In addition, it was difficult to apply the proposed stabilization control to walk on the surface, which was tilted along rolling axis due to the dynamics difference between two feet.

VI. CONCLUSION

This paper proposed a novel stabilization control for humanoid robot to walk stably and dynamically on an inclined plane. The proposed controller guaranteed the stability without requiring any dynamic model information because it was based on the time-domain passivity control approach. Moreover, the proposed controller did not require complex dynamic equations and the parameter adjustment for control. Since the passivity controller injected an adaptive damping whenever a net energy was produced, it was also able to work properly in a wide variety of dynamic operating conditions.

The proposed stabilization control was verified through dynamic walking simulations using the simulation program, Webots. Humanoid robot could stably walk even on inclined plane along pitching axis using the proposed posture control method. Since the current posture controller can only be applied in pitching axis, a future research will investigate walking on the surface, which is inclined along rolling axis. In addition, generating the modified walking pattern according

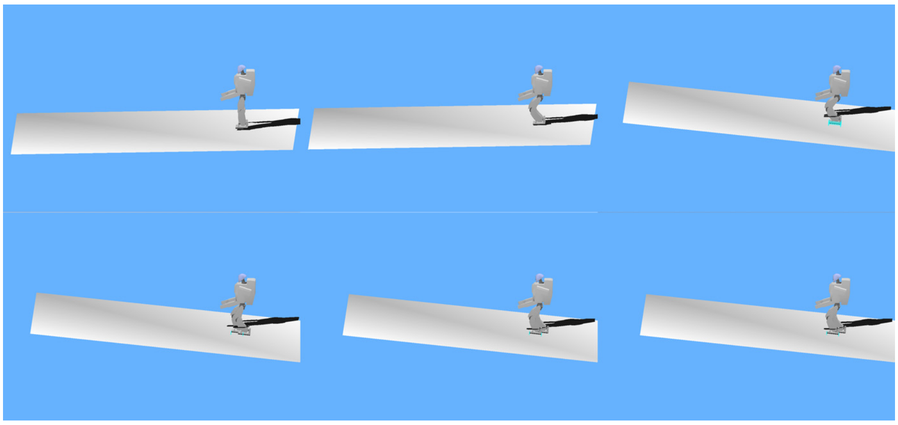


Fig. 9. Snapshots of biped robot walking up the slope of 10 degrees.

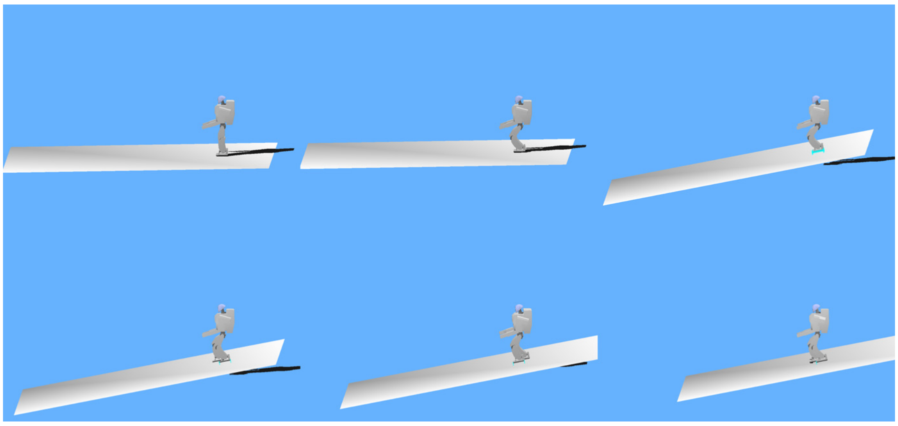


Fig. 10. Snapshots of biped robot walking down the slope of 10 degrees.

to real-time measured values from the posture sensor will be further explored.

REFERENCES

[1] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, “The development of honda humanoid robot,” in *Proc. of IEEE Int. Conf. on Robotics and Automations*, Leuven, Belgium, May 1998, pp. 1321–1326.

[2] S. Kagami, K. Nishiwaki, J. J. Kuffner., Y. Kuniyoshi, M. Inaba, and H. Inoue, “Online 3d vision, motion planning and bipedal locomotion control coupling system of humanoid robot : H7,” in *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 3, Lausanne, Switzerland, Oct. 2002, pp. 2557–2562.

[3] J.-H. Kim, D.-H. Kim, Y.-J. Kim, K.-H. Park, J.-H. Park, C.-K. Moon, K. T. Seow, and K.-C. Koh, “Humanoid robot HanSaRam: Recent progress and development,” *J. of Advanced Computational Intelligence & Intelligent Informatics*, vol. 8, no. 1, pp. 45–55, Jan. 2004.

[4] S. Kajita, A. Kobayashi, and T. Yamamura, “Dynamic walking control of a biped robot along a potential energy conserving orbit,” *IEEE Trans. on Robotics and Automation*, vol. 8, pp. 431–438, Aug. 1992.

[5] Q. Huang, K. Yokoi, S. Kajit, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie, “Planning walking patterns for a biped robot,” *IEEE Trans. on Robotics and Automation*, vol. 17, no. 3, pp. 280–289, June 2001.

[6] K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Ibana, and H. Inoue, “Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired ZMP,” in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 3, Lausanne, Switzerland, Sept. 2002, pp. 2684–2689.

[7] B.-J. Lee, D. Stonier, Y.-D. Kim, J.-K. Yoo, and J.-H. Kim, “Modifiable walking pattern of a humanoid robot by using allowable zmp variation,” *IEEE Trans. on Robotics*, vol. 24, no. 4, pp. 917–923, Aug. 2008.

[8] J. Yamaguchi, E. Soga, S. Inoue, , and A. Takanishi, “Development of a bipedal humanoid robot control method of whole body cooperative dynamic biped walking,” in *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 1, Detroit, MI, May 1999, pp. 368–374.

[9] M. Popovic, A. Hofmann, and H. Herr, “Angular momentum regulation during human walking: biomechanics and control,” in *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 3, New Orleans, LA, Apr. 2004, pp. 2405–2411.

[10] B.-J. Lee, Y.-D. Kim, and J.-H. Kim, “Balance control of humanoid robot for Hurosot,” in *Proc. of IFAC World Congress*, Prague, Czech, July 2005.

[11] Y.-D. Kim, B.-J. Lee, J.-H. Ryu, and J.-H. Kim, “Landing force control for humanoid robot by time-domain passivity approach,” *IEEE Trans. on Robotics*, vol. 23, no. 6, pp. 1294–1301, Dec. 2007.

[12] B. Hannaford and J.-H. Ryu, “Time-domain passivity control of haptic interfaces,” *IEEE Trans. on Robotics and Automation*, vol. 18, pp. 1–10, Feb. 2002.

[13] J.-H. Ryu, D.-S. Kwon, and B. Hannaford, “Stability guaranteed control: time domain passivity approach,” *IEEE Trans. on Control Systems Technology*, vol. 12, pp. 860–868, Nov. 2004.

[14] J.-H. Ryu, Y.-S. Kim, and B. Hannaford, “Sampled- and continuous-time passivity and stability of virtual environments,” *IEEE Trans. on Robotics*, vol. 20, no. 4, pp. 772–776, Aug. 2004.