International Journal of Control, Automation, and Systems (2014) 12(2):372-382 ISSN:1598-6446 eISSN:2005-4092
DOI 10.1007/s12555-013-0053-5 http://www.springer.com/12555

Motion Control of a Quadruped Robot in Unknown Rough Terrain using 3D Spring Damper Leg Model

Duc Trong Tran, Ig Mo Koo, Yoon Haeng Lee, Hyungpil Moon, Jachoon Koo, Sangdeok Park, and Hyouk Ryeol Choi*

Abstract: This paper presents a simple method to control the motion of a quadruped robot in unknown rough terrain using a full dynamic model. First, using the stiffness control method, the behavior of the four legs is approximated using four 3D spring damper systems. In this way, the dynamic model can be derived in Cartesian space. Based on this model, a control strategy is proposed to preserve the asymptotical stability of the system. In addition, a reflex motion control is introduced to cope with the rotational disturbance of the robot body. Finally, dynamic simulations and experiments of a quadruped walking robot were performed on unknown rough terrain to verify the proposed method.

Keywords: 3D spring damper system, dynamic legged locomotion, quadruped locomotion.

1. INTRODUCTION

Recently, quadruped robots have gained attention among robotics researchers [1-4]. A number of control strategies have been proposed to control quadruped robots [1-3]. Some of these are based on biological insight [1] while others are based on heuristic reflexive motion control [2,3].

For fast running or dynamic walking, the idea of approximating the motion of the robots as a spring damper inverted pendulum has been proposed by several researchers [4-6]. Previous works related to this model began with an inverted point mass-spring pendulum in a plane [4] and then extended to more complicated cases by adding more degrees of freedom such as pitch angle [4,5], 3D translational motion [6], etc. However, a real robot is not a simple inverted pendulum or a system of springs attached to a rigid body but a system of four legs attached to a rigid body. Previous approaches addressed how to model the motion of the robot body, yet how to control the four legs of the robot to realize the analyzed motion was not discussed.

In contrast to typical manipulators for which the stability is observed simply by the error dynamics of the system, the stability of quadruped robots relies on the proposed stability criteria because the full dynamics of the quadruped robots is difficult to handle using closed formed equations, especially on unknown rough terrain. A summary of these criteria can be found in [7]. Altendorfer *et al*. performed an analysis on the stability of the motion of the quadruped robot based on return factor maps [8]. Similar work studying the limit cycle stability was also carried out by Ahmadi *et al*. [9]. However, these works only used a simplified spring-damper inverted pendulum model or a single leg model but not all leg dynamics were used. Meek analyzed the behaviors of passive legged robots using a simple under-actuation system [10] and a similar adaptive control using compliant legs was proposed by Buchli *et al*. [11]. On the other hand, Berkemeier *et al*. and Zou *et al*. analyzed the stability of the robot by deriving a full dynamic model under a number of constraints [12,13]. However, they only considered several special simplified cases in which the robot motion was only considered in a plane. Gait planning using landing accordance ratio was introduced in [17]. This approach was done in subcontroller level only without discussing whole dynamics of the system.

In this work, we aim to control the motion of a robot using an approximated dynamic model. The core idea is to approximate the four legs with virtual spring damper systems such that the full dynamic equations of the translational motion of the robot are easily derived in Cartesian space. The difference between this and previous works is that we introduce these virtual spring-damper systems in a 3D space that includes three independent dimensions, not in a single dimension as typical models in previous works. Furthermore, we show how to perform the approximation in detail by using the stiffness control method for the sub-controller of the robot legs, which was briefly discussed in previous works. In addition, using the obtained dynamic equations, a control strategy is proposed such that the robot can

Manuscript received February 1, 2013; revised July 28, 2013 and September 27, 2013; accepted November 5, 2013. Recommended by Associate Editor Pinhas Ben-Tzvi under the direction of Editor Myotaeg Lim.

This work was supported by the Ministry of Knowledge Economy under the Human Resources Development Program for Convergence Robot Specialists (NIPA-2012-H1502-13-1001).

Duc Trong Tran, Ig Mo Koo, Yoon Haeng Lee, Hyungpil Moon, Ja Choon Koo, and Hyouk Ryeol Choi are with the School of Mechanical Engineering, Sungkyunkwan University, Suwon, Korea (e-mails: {jamestran, kooigmo, yhlee, hyungpil, jckoo, hrchoi}@me.skku.ac.kr).

Sangdeok Park is with the Division of Applied Robot Technology, Korea Institute of Industrial Technology, Ansan, Korea (e-mail: sdpark@kitech.re.kr).

* Corresponding author.

follow the predesigned control motion. The error dynamics of the system are also verified to be asymptotically stable by using the proposed control method. Finally, dynamic simulations and outdoor experiments of a quadruped robot performing dynamic walking on several unknown rough terrains are presented to validate the proposed control method.

This paper is organized as follows. Section 2 discusses the 3D spring damper leg model and the basic assumptions used in this paper. Sections 3 and 4 present the core focuses including the equation of motion and control strategy. Section 5 shows the simulation results and experiments are introduced in section 6. Finally, conclusions are given in section 7.

2. 3D SPRING DAMPER LEG MODEL

2.1. Coordinate definitions and basic assumptions

In this research, we define two reference coordinate frames for observing the motion of the robot. The global coordinate frame Σ_G is attached to the ground and the local coordinate frame Σ_L is attached to the center point of the robot body and rotates with it as shown in Fig. 1(a). For simplicity and consistency, the variable definition is introduced in Fig. 1(a).

In this paper, the normal letter (e.g., '$\mathbf{r}$') is used for vectors calculated in the global frame Σ_G and super-scripted letters (e.g., '$\mathbf{r}^L$') are used for vectors calculated in the local frame Σ_L. In our work, two basic assumptions are made: (1) slippage does not occur at the feet and (2) the mass of the legs is small in comparison with the total mass of the robot and the moments of inertia of the legs are negligible. These assumptions are typically used in previous works and are used here to considerably reduce the complexity of our problem [1,4-6,10-12].

2.2. Spring damper approximation

In this section, we introduce the analysis in which the leg behaviors are approximated as a 3D spring-damper system.

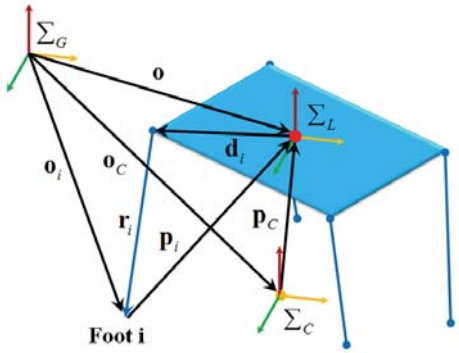
If we consider leg i as an independent manipulator, the force applied at the hip and the supporting force at the foot can be treated as the base applied force and end-effector acting force, as in the case of regular manipulators (Fig. 1(b)). Based on the static analysis of a regular industrial manipulator as in the textbook [15], it is well-known that the relation of the applied torque $\boldsymbol{\tau}_i$ on the joints and the supporting force at the foot is given by

$$\boldsymbol{\tau}_i = -\mathbf{J}_i^T \mathbf{f}_i^L, \tag{1}$$

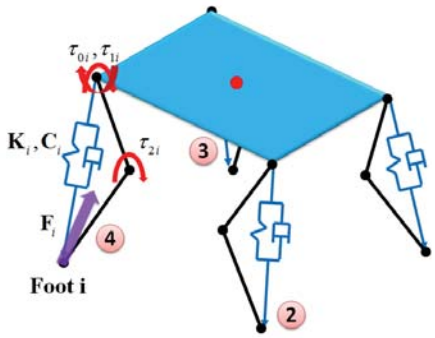
where $\mathbf{J}_i$ is the Jacobian matrix of the i th leg. The minus sign is omitted in (1) because the supporting force $\mathbf{f}_i^L$ is equal to but has an opposite direction with the end-effector acting force at the foot.

Inspired by the stiffness control method, we design the applied torque for the joints of the i th leg as

$$\boldsymbol{\tau}_i = \mathbf{J}_i^T \left[\mathbf{K}_i(\mathbf{r}_{i0}^L - \mathbf{r}_i^L) - \mathbf{C}_i \dot{\mathbf{r}}_i^L \right] + \mathbf{G}_i(\boldsymbol{\theta}_i), \tag{2}$$



(a)



(b)

Fig. 1. Reference coordinate frames and defined position vectors.

where $\mathbf{K}_i$ and $\mathbf{C}_i$ are the stiffness matrix and damping matrix; and $\mathbf{r}_{i0}^L$ is the desired trajectory of the i th foot with respect to the hip joint in the local frame Σ_L. $\mathbf{r}_i^L$ and $\dot{\mathbf{r}}_i^L$ are the real position vectors of the i th foot with respect to the hip joint i in the local frame Σ_L and its time derivative.

The last term in (2) is the gravity compensation which is calculated as

$$\mathbf{G}_i(\boldsymbol{\theta}_i) = \frac{\partial V_i}{\partial \boldsymbol{\theta}_i},$$

where V_i is the potential energy and $\boldsymbol{\theta}_i$ is the joint angle vector of leg i.

In this work, we assume the Jacobian matrix is fully ranked in $\mathbb{R}^{3x3}$, which means each leg has 3 degrees of freedom (DOF). For cases in which the legs have more than 3 DOF, a higher dimensional spring-damper system model should be used. Equating (1) and (2) and then rearranging them, we have the final form of the supporting force in the local frame such as

$$\mathbf{f}_i^L = \mathbf{K}_i(\mathbf{r}_i^L - \mathbf{r}_{i0}^L) + \mathbf{C}_i \dot{\mathbf{r}}_i^L. \tag{3}$$

This equation has the form of a spring damper system with the stiffness matrix $\mathbf{K}_i$ and damping matrix $\mathbf{C}_i$. It is not a single dimensional system but a 3D system in which the independency of the motion along the three local axes relies on the form of matrices, $\mathbf{K}_i$ and $\mathbf{C}_i$. From this result, we approximate the behavior of the four legs as four virtual 3D spring damper systems (Fig. 1(b)) which have the force-displacement relation of (3). As

usual, to modulate the behavior of these virtual 3D spring damper legs, we control the desired trajectory $\mathbf{r}_{i0}^{L}$ which is similar to the "natural length" of a normal spring. The generated force is then changed according to the change of $\mathbf{r}_{i0}^{L}$. Considering the symmetry of the robot structure, we choose the stiffness matrix and damping matrix to be the same for all four legs and named them as **K** and **C**. This maintains the freedom to choose different stiffness and damping values along the three axes.

In our work, we tried to simplify the problem by ignoring the friction in the system. In fact, modeling the friction is a difficult practical issue in the robot control field. Concerning quadruped robots, there are two majored forms of friction: joint friction and foot-ground contact friction. In common manipulators which are similar to quadruped legs, the joint friction is usually modeled to be inverse-signed and proportional to joint velocities. This term, however, provides similar effects as those of the virtual damping in our model. Therefore, we believe that modulating the virtual damping term in our 3D spring damper model can somehow simulate the effect of joint friction simultaneously. On the other hand, our controller is designed to deal with unknown environments in which information concerning the friction, height map, and profile of the terrains is assumed to be unknown. Hence, it is not possible to model the friction at each foot contact. Furthermore, this type of friction affects only the slippage at the feet. With our assumption that slippage does not occur at any of the feet, this friction does not affect the energy consumption of the robot and thus is not considered in the dynamic equation.

3. EQUATIONS OF MOTION

To control the robot, we first need to derive the full dynamic equations of translational motion. Using the proposed 3D spring damper leg model, the Newton 2nd equation for the robot body is written as

$$\begin{aligned} m\ddot{\mathbf{o}} &= \mathbf{R}\sum_{i\in\mathbf{I}_{SP}}\mathbf{f}_i^{L} + m\mathbf{g} \\ &= \mathbf{R}\sum_{i\in\mathbf{I}_{SP}}\left[\mathbf{K}(\mathbf{r}_i^{L}-\mathbf{r}_{i0}^{L})+\mathbf{C}\dot{\mathbf{r}}_i^{L}\right]+m\mathbf{g}, \end{aligned} \tag{4}$$

where **o** denotes the position vector of COM with respect to the global frame and m is the total mass of the robot. **R** is the rotational matrix of the robot body and $\mathbf{I}_{SP}$ represents the index set of all supporting legs. The first term of the right hand side (RHS) is all the elastic forces from the spring damper supporting the legs and the last term is gravitational force.

For consistency, we changed all the vectors in the RHS of (4) into the global frame by using the following relations

$$\begin{aligned} \mathbf{r}_i^{L} &= \mathbf{R}^{T}\mathbf{r}_i, \\ \dot{\mathbf{r}}_i^{L} &= \mathbf{R}^{T}(\dot{\mathbf{r}}_i-\boldsymbol{\omega}\times\mathbf{r}_i), \end{aligned}$$

where $\boldsymbol{\omega}$ is the rotational velocity of the robot body. Substituting those relations into (4), we have

$$m\ddot{\mathbf{o}} = \sum_{i\in\mathbf{I}_{SP}}\left\{\mathbf{K}_R(\mathbf{r}_i-\mathbf{r}_{i0})+\mathbf{C}_R(\dot{\mathbf{r}}_i-\boldsymbol{\omega}\times\mathbf{r}_i)\right\}+m\mathbf{g}, \tag{5}$$

where $\mathbf{K}_R=\mathbf{R}\mathbf{K}\mathbf{R}^{T}$ and $\mathbf{C}_R=\mathbf{R}\mathbf{C}\mathbf{R}^{T}$.

For testing the stability and robustness of the proposed model, we use an additional frame Σ_C which is parallel to the global frame and located at a specific point C defined by the following position vector

$$\mathbf{o}_C = \frac{1}{n_{SP}}\sum_{i\in\mathbf{I}_{SP}}\mathbf{o}_i, \tag{6}$$

where n_{SP} is the number of supporting legs.

The point C is actually the centroid of the supporting geometry. It is a fixed point with respect to the global frame due to the assumption that no slippage occurs at the supporting feet. Consequently, the time derivative $\dot{\mathbf{o}}_C$ is zero. Thus, without lost of generality, the observed dynamics of the robot motion with respect to the global frame and this frame Σ_C is the same. In fact, we assume that the robot is not equipped with any sensor that measures its location with respect to the global frame. Therefore, the defined frame Σ_C is required to locally observe the global dynamics of the robot motion.

The translational motion of the COM with respect to the frame Σ_C is monitored by the vector

$$\mathbf{p}_C = \mathbf{o}-\frac{1}{n_{SP}}\sum_{i\in\mathbf{I}_{SP}}\mathbf{o}_i = \frac{1}{n_{SP}}\sum_{i\in\mathbf{I}_{SP}}\mathbf{p}_i. \tag{7}$$

Then, by changing all the variables in (5) into p_C, we have

$$\begin{aligned} m\ddot{\mathbf{p}}_C &= n_{SP}\mathbf{K}_R(\mathbf{p}_{C0}-\mathbf{p}_C)-n_{SP}\mathbf{C}_R\dot{\mathbf{p}}_C+m\mathbf{g} \\ &\quad+\mathbf{K}_R\sum_{i\in\mathbf{I}_{SP}}(\mathbf{d}_{i0}-\mathbf{d}_i)+n_{SP}\mathbf{C}_R\boldsymbol{\omega}\times\mathbf{p}_C, \end{aligned} \tag{8}$$

where $\mathbf{p}_{C0}$ is the control variable of the robot COM with respect to the frame Σ_C.

This is the final form of the dynamics equation of motion of the robot COM. The first two terms on the RHS of (8) are the total spring-damping forces from all supporting legs. The third term is the gravity force and the last two terms are the spring-damping forces due to the rotational motion of the robot body.

4. CONTROL STRATEGY

4.1. Global control strategy

Inspired from the computed torque method in the manipulator control field, we aim to generate stable error dynamics for the system by designing the control law as follows.

$$\begin{aligned} \mathbf{p}_{C0} &= \mathbf{p}_{Cd}+\mathbf{K}_R^{-1}\mathbf{C}_R(\dot{\mathbf{p}}_{Cd}-\boldsymbol{\omega}\times\mathbf{p}_C) \\ &\quad-\frac{1}{n_{SP}}\sum_{i\in\mathbf{I}_{SP}}(\mathbf{d}_{i0}-\mathbf{d}_i)+\frac{m}{n_{SP}}\mathbf{K}_R^{-1}\mathbf{p}_{Cd}, \end{aligned} \tag{9}$$

where $\mathbf{p}_{Cd}$ is the desired trajectory of the COM with respect to frame Σ_C. This is usually a predesigned

function of time based on the motion that needs to be performed by the robot.

Substituting (9) into (8), we obtain

$$\ddot{\mathbf{e}}_C + \frac{n_{SP}\mathbf{C}_R}{m}\dot{\mathbf{e}}_C + \frac{n_{SP}\mathbf{K}_R}{m}\mathbf{e}_C = \mathbf{g}\left(\mathbf{e}_C = \mathbf{p}_{Cd} - \mathbf{p}_C\right), \quad (10)$$

where $\mathbf{e}_C$ is the error vector of the COM trajectory with respect to frame Σ_C.

The error dynamics in (10) is obviously a second order linear system which has the homogenous eigenvalues as

$$\lambda_{2i-1,2i} = \frac{1}{2m}\left(-n_{SP}c_i \pm \sqrt{n_{SP}^2 c_i^2 - 4mn_{SP}k_i}\right), \quad (11)$$

where i =1..3; and k_i and c_i are the stiffness and damping coefficients along the three axes which are defined on the diagonal lines of matrices $\mathbf{K}$ and $\mathbf{C}$, respectively.

Since all the parameters n_{SP}, c_i, and m are strictly positive real numbers, the eigenvalues in (11) always have a negative real part, which confirms the exponential stable behavior of the system at the equilibrium point

$$\mathbf{e}_0 = \frac{m\mathbf{K}_R^{-1}\mathbf{g}}{n_{SP}}. \quad (12)$$

Instead of obtaining zero error dynamics, the proposed control system exhibits a stable equilibrium point at some steady error which represents the level of elastic energy stored in all the standing legs to support the body weight. This means that, except for the height, the COM can follow the desired trajectory appropriately and stably.

4.2. Supporting leg control

In the previous subsection, (9) only shows the suitable control function to obtain the stable equilibrium for the COM. The two remaining issues are how to control the four legs to create this stabilizing function and how to maintain the robustness of the control signals of all individual legs.

First, it is necessary to determine how to calculate the desired motion represented by the vector $\mathbf{p}_{Cd}$ and its derivatives. In this work, we assume the system does not have sensors that can measure the global position and motion of the robot. Therefore, the idea for tracking the exact motion and location of the robot in the global frame is inappropriate. In addition, our work deals with unknown rough terrains. Thus, an exact pre-designed supporting geometry (or the desired location of point C), and the desired motion of the robot related to this point C are impossible tasks.

In this approach, the desired motion of the robot on a simple flat terrain in the ideal case is first pre-analyzed to design the desired positions $\mathbf{p}_{id}$ of all four legs of the robot along with this motion as functions of time. In this way, the desired trajectory of the COM is calculated by

$$\mathbf{p}_{Cd} = \frac{1}{n_{SP}}\sum_{i\in\mathbf{I}_{SP}}\mathbf{p}_{id} = -\frac{1}{n_{SP}}\sum_{i\in\mathbf{I}_{SP}}\left(\mathbf{r}_{id} + \mathbf{d}_{id}\right), \quad (13)$$

where $\mathbf{r}_{id}$ is the desired foot position vector with respect to the hip of the leg i and $\mathbf{d}_{id}$ is the desired motion of the hip i with respect to the global frame. In a real system, tracking in acceleration is very difficult and inappropriate. Therefore, we assume that the desired acceleration $\ddot{\mathbf{p}}_{id}$ is zero.

The control function $\mathbf{p}_{C0}$ in (9) includes n_{SP} individual control signal $\mathbf{p}_{i0}$ for all supporting legs. The only constraint for calculating this function is the RHS of (9). Therefore, if $n_{SP} > 1$, it becomes a redundant constraint for choosing n_{SP} vector functions $\mathbf{p}_{i0}$. In this case, the form of the constraint would become

$$\sum_{i\in\mathbf{I}_{SP}}\left\{\mathbf{r}_{i0} - \mathbf{r}_{iT} - \mathbf{r}_{iR}\right\} = \mathbf{0}, \quad (14)$$

where

$$\mathbf{r}_{iT} = \mathbf{r}_{id} + \mathbf{K}_R^{-1}\mathbf{C}_R\left(\dot{\mathbf{r}}_{id} - \boldsymbol{\omega}\times\mathbf{r}_i\right),$$

$$\mathbf{r}_{iR} = \mathbf{d}_{id} - \mathbf{d}_i + \mathbf{K}_R^{-1}\mathbf{C}_R[\boldsymbol{\omega}_d\times\mathbf{d}_{id} - \boldsymbol{\omega}\times\mathbf{d}_i],$$

and $\boldsymbol{\omega}_d$ denotes the desired rotational velocity of the robot body.

While many solutions for solving these constraints may be possible in order to choose the control functions $\mathbf{r}_{i0}$, we choose the solutions for $\mathbf{r}_{i0}$ that maximize the independency between the four leg control signals. This helps to not only simplify the problem but also to prevent the complicated disturbance that might occur due to the unexpected landing or swinging of the legs due to unknown terrain geometry.

The analytic forms of the chosen solutions are

$$\mathbf{r}_{i0} = \mathbf{r}_{iT} + \mathbf{r}_{iR}, \quad (15)$$

or

$$\mathbf{r}_{i0}^L = \mathbf{R}^T\mathbf{r}_{i0} = \mathbf{R}^T(\mathbf{r}_{iT} + \mathbf{r}_{iR}). \quad (16)$$

The first term in the RHS of (15) is used to control the translational motion of the body, and the second term is used to balance the rotation motion. Instead of deriving complicated rotational equations of motion, we choose to balance the rotation using an undirected method of controlling the vector $\mathbf{d}_{id}$ [1,10]. Even though rotational dynamics are not computed in detail, this method proved to be useful as shown in the results of Sections 5 and 6.

4.3. Swinging leg relocating control

The robot body motion can only be controlled by the supporting legs. The swinging legs, on the other hand, are used to compensate and modify the motion in the next phase of the motion cycle. The control function given in (15) is only counted for the supporting state of all the legs. Therefore, modifications are needed for controlling the swinging legs, as is discussed in this subsection.

As mentioned in the previous sections, explicit optimal relocating control for swinging legs is impossible for walking on rough terrains. Instead, in our work, the controller simply tries to relocate the swinging legs to the desired swinging motion.

The motion of the swinging legs is governed by the following equation:

$$\mathbf{K}_R\left(\mathbf{r}_j-\mathbf{r}_{j0}\right)+\mathbf{C}_R\left(\dot{\mathbf{r}}_j-\boldsymbol{\omega}\times\mathbf{r}_j\right)=\mathbf{0}\quad\left(j\in\mathbf{I}_{SW}\right),\qquad(17)$$

where $\mathbf{I}_{SW}$ is the index set of all swinging legs.

Similarly, the control function generating the zero error dynamics for the swinging legs is calculated as

$$\mathbf{r}_{j0}^{L}=\mathbf{R}^{T}\left[\mathbf{r}_{jd}+\mathbf{K}_R^{-1}\mathbf{C}_R\left(\dot{\mathbf{r}}_{jd}-\boldsymbol{\omega}\times\mathbf{r}_j\right)\right]\quad\left(j\in\mathbf{I}_{SW}\right).\ (18)$$

The swinging control function in (18) is exactly the same as the first component $\mathbf{r}_{iT}$ in (15). Therefore, the second component $\mathbf{r}_{iR}$ will cause discontinuity in the control signal when the support-swing changing phase occurs, which is a considerable burden for the system.

To overcome this, we modify the control strategy for the swinging phase as follows:

$$\mathbf{r}_{j0}^{L}=\mathbf{R}^{T}\mathbf{r}_{jT}+\mathbf{r}_{j\_ori}\quad\left(j\in\mathbf{I}_{SW}\right),\qquad(19)$$

where $\mathbf{r}_{iT}$ is defined as given in (15) and

$$\mathbf{r}_{i\_ori}^{L}=\begin{cases}-b_0\mathbf{R}^{T}\mathbf{r}_{iR} & \text{if leg i is supporting}\\ \mathbf{f}(v_0) & \text{if leg i is swinging.}\end{cases}\qquad(20)$$

The scalar b_0 is a scaling gain, $\mathbf{r}_{iR}$ is computed as given in (15), and $\mathbf{f}(v_0)$ is a robust decaying function. We have $\mathbf{y}(t)=\mathbf{f}(v_0)$ if and only if

$$\dot{\mathbf{y}}(t)=\begin{cases}-sign[\mathbf{y}(t)]v_0 & \|\mathbf{y}(t)\|>0\\ 0 & \|\mathbf{y}(t)\|=0.\end{cases}\qquad(21)$$

Using this definition, in the swinging state, the discontinuous term will decrease smoothly with a constant velocity v_0 until it vanishes.

It is noted that the behavior of the robot with respect to frame C and the global frame is the same only outside of the swing-supporting instances. At those instances, there is a switching peak in the position of point C. The error dynamics of the system can be considered as a piecewise continuous equation. However, point C was only used to observe the stability of the trajectory tracking in our system, not for planning or control. As described in (15) and (19), the control signal does not rely on the location of point C. In addition, the discontinuity only occurred due to the difference between the control signal of the supporting leg (in (15)) and the control signal of the swinging leg (in (19)), which has already been discussed and processed using (20). In addition, for the hybrid nature of error dynamics, the equilibrium point (12) is the same for all symmetric switching in the supporting-swinging phase (FL-HR and FR-HL switching). In addition, the error dynamics exhibit exponential stability; therefore, even for switching that occurs near the equilibrium or elsewhere, the error also keeps reaching equilibrium. A severe case might occur in which the four legs might not be able to always perform symmetric switching in the supporting-swinging phase on unknown extreme rough terrain. For example, in the trot gait, a situation might occur in which three legs are supporting and one leg is swinging due to the unexpected change in the terrain height map. For these cases, the equilibrium point of the error dynamics changes according to the switching among the 2 nearby points. In this way, the tracking error would tend to go back and forth among these points in a single step. Up to the point at which this non-symmetric switching stops, the error will return to the original equilibrium point. Fortunately, even though this case might occur and an additional equilibrium might appear in the error dynamics, this would not have an adverse effect on the system because the additional equilibrium point is located at the lower level of errors (due to the fact that n_{SP} is larger at this equilibrium point).

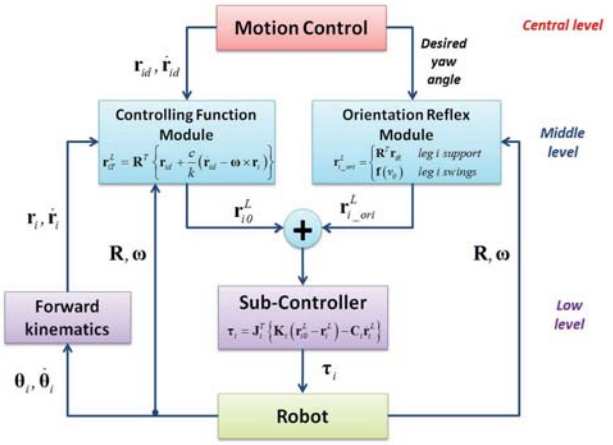


Fig. 2. Control diagram.

4.4. Controller diagram

As a brief summary, the control diagram of the entire system is depicted in Fig. 2. Three levels are shown in this control diagram. The highest level is the central level where we directly give the required tasks usually related to robot motion control such as operating velocity, heading direction, etc. The second level is the middle level where the controller itself calculates the necessary control signals to fulfill the commanding tasks using the designed functions as given in (15) and (19). The final level is the lowest level where the necessary torque is calculated.

In each level, sensory information is required for the calculating processes. In our work, we use basic sensors such as joint sensors to obtain the joint angle and angular velocity, and an IMU sensor to obtain the roll pitch yaw angles of the robot and its time derivatives.

Our control diagram can easily be extended to have performances that are more complex by simply adding new reflex modules in the middle level. The reflex modules are used to deal with the situation or the tasks that are designed using other additional possible sensors. For example, using an Inertia Measurement Unit (IMU) sensor, we can obtain the robot translational acceleration. From this feedback, external impacts applied to the robot body can be sensed and the counterbalancing signal in the additional reflex module is then calculated. Other similar problems can be addressed using the same method.

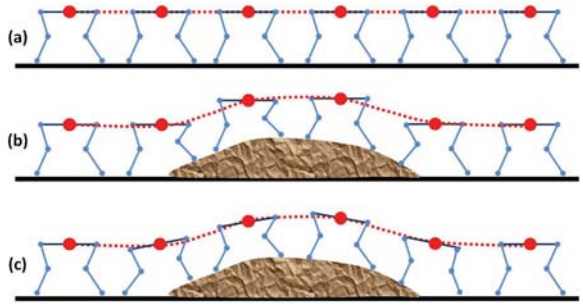


Fig. 3. Example of robot behavior on rough terrain.

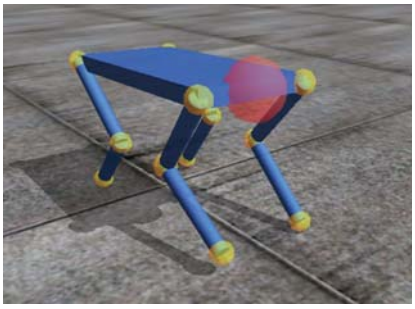


Fig. 4. The robot model in ODE simulation.

Table 1. Properties of simulation model.

	Body	Thigh	Tibia
Mass [kg]	20.014	0.221	0.221
Moment of Inertia $\times 10^{-3}$ [kg m^2]	I_x = 33.36 I_y = 150.1 I_z = 183.0	I_x = 0.32 I_y = 0.32 I_z = 0.01	I_x = 0.32 I_y = 0.32 I_z = 0.01
Length [cm]	30	13	13
Width [cm]	14	2	2
COM	c.o.l*	c.o.l*	c.o.l*

c.o.l* = center of link

4.5. Robot behavior analysis

In this work, we assume that the robot does not have any information about the environment and simply understands the environment as a simple flat terrain.

For example, we assume that the robot initially walks on a flat terrain and suddenly steps on an unknown rock as depicted in Fig. 3. Before stepping on the rock, the robot understands the ground as its walking plane and the desired motion is tracked with respect to this plane so that it simply moves forward in a motion parallel to the ground as shown in Fig. 3(a). However, when the robot steps on the rock, it will understand the inclined plane formed by all the supporting legs as its working plane. Therefore, the robot body will no longer move parallel to the ground. Instead, it will move along the plane which is parallel to the inclined plane and which follows the desired direction (Fig. 3(b)). This phenomenon occurs because the desired motion of the robot COM is designed to be parallel to the working plane.

In addition, the orientation of the robot body is controlled by the signal $\mathbf{r}_{iR}$ presented in section 3. By manipulating this signal, we can control the robot orientation by keeping it parallel to the ground as shown in Fig. 3(b) or tracking the orientation of the terrain which is actually unknown to the robot itself as depicted in Fig. 3(c). This is an interesting result of our control method since it is difficult to obtain the terrain orientation or roughness using current non-visual sensors. Therefore, it is difficult to adapt to the terrain roughness.

It is helpful to recall that quadrupedal animals change their pitch and roll angles to adapt to the ground when they walk on rough terrain. In this work, we decided to only control the yaw angle of robot body, which means controlling the heading direction of the robot. The other roll and pitch angles are set to move freely, allowing the motion of the robot on rough terrain to be more flexible and more stable since it can adapt to the change of the environment roughness.

5. SIMULATIONS

5.1. Simulation model

To verify the proposed controller, we performed dynamic simulations with the model shown in Fig. 4 by using an Open Dynamic Engine [16].

The robot has a total of 12 joints, comprising 3 joints for each leg. The total weight of the robot is about 22 kg and each leg weighs 0.442 kg (about 2% of the total weight). The size of the robot body is 30 x 14 x 4 cm and the full stretched leg length is 26 cm. The structures of all four legs are the same and the details of the robot specification are given in Table 1. In all the simulations, we control the robot to perform the typical trot gait. The hard contact model with Coulomb friction is used in the simulation. The friction coefficient is set to 0.7.

5.2. Selection of simulation parameters

The first important parameter to choose is the stiffness. According to the report of the previous biological study, the reference leg compression is around 10% of the maximum leg length in dynamic locomotion [14]. In this work, we choose the steady state error in (12) to be similar to the reference leg compression. Hence, the desired displacement of the spring to support the robot body in the trot gait is about 3cm (about 11% of leg length). Therefore, the required stiffness which is chosen to be the same for all three axes should be

$$k = \frac{mg}{ne_{C0}} = \frac{22 \times 9.81}{2 \times 0.026} = 3597 \text{ Nm}^{-1}. \quad (22)$$

Based on our experiences, the damping coefficient is not as important as the spring stiffness since it is the energy dissipating term. However, there exists a limit, whereby the robot becomes very sensitive to velocity changes and becomes unstable if the damping coefficient is larger than this limit. This is due to the level of accuracy of the numerical integration of the simulation. As the given timestep decreases, the accuracy of the numerical computation increases. In this work, we choose the damping coefficient c to be about 3% of the spring stiffness value which is 105Nsm^{-1}.

In all the simulations, the desired walking height h of the robot is chosen to be 80% of the leg length. In addition, we use the Froude number, which represents

the relation between the moving speed and the height of the robot at a certain gait, to determine the suitable required velocity for the robot [10]. The Froude number Fr is chosen to be 0.4 for which the robot performs the trot walking gait. In this case, the velocity is

$$v = \sqrt{ghFr} = 0.903 \text{ ms}^{-1}. \quad (23)$$

The decaying velocity v_0 in (21) is set to 0.2 ms^{-1}.

5.3. Simulation scenarios

Two environments were prepared for the robot. The first terrain is extremely rough, similar to a complex rocky field (Fig. 5) and the second terrain is slightly rough, similar to a hill (Fig. 8). The sizes of these two terrains are 3 x 1 m and 10 x 5 m, respectively. The robot performed two tasks on these terrains. The detailed motions are shown in Figs. 5 and 8.

For the first terrain, the robot task was simply to move forward and successfully overcome the terrain without falling over. As shown in Fig. 5, the robot successfully overcame the first extremely rough terrain in a straightforward manner. On some positions of the terrain, the robot legs actually slipped due to the unknown roughness of the terrain. Fortunately, even though the controller did not count these effects, the robot could still successfully finish the task. The error dynamics are shown in Fig. 6. Starting to move at t = 0 s, the robot stepped on the rough terrain after 2.5 s. While moving on the first terrain, the unexpected swing-support phase change usually occurred such that the total stiffness and damping of the robot changed accordingly. Besides, these sudden changes or disturbances generated impact forces on the robot due to the fact that the robot legs are not perfectly without mass in the simulation model. However, due to the confirmed asymptotical stability,

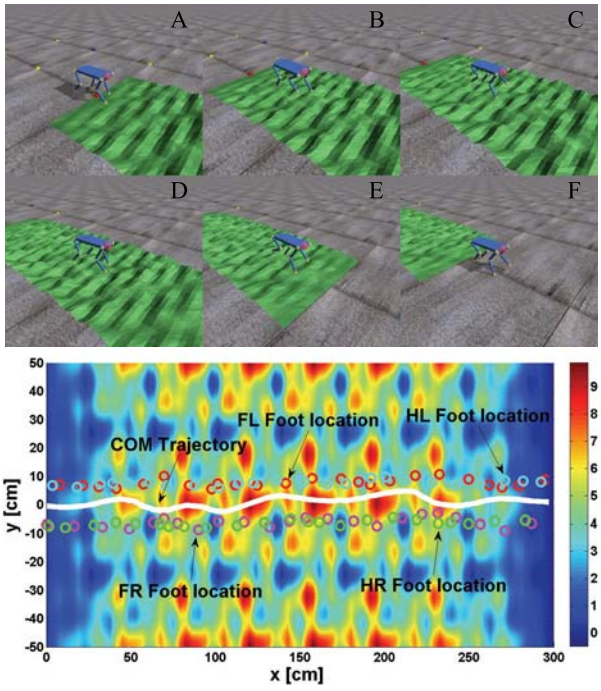


Fig. 5. The robot trotting motion on a rough terrain.

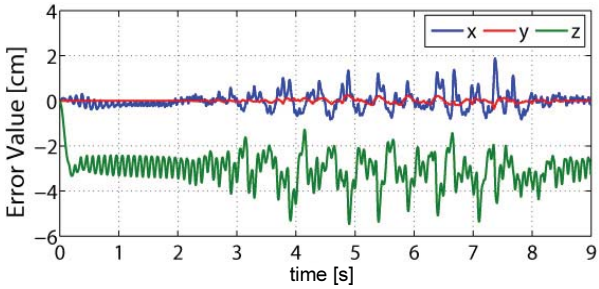


Fig. 6. Error vector of the COM trajectory on the first rough terrain.

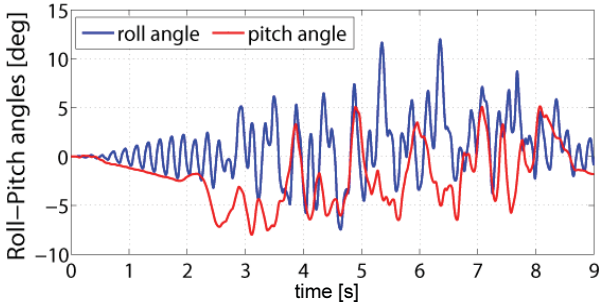


Fig. 7. Roll and pitch angles of robot body during the motion on the rough terrain.

regardless of the number of supporting legs, the error of the robot motion merely oscillated around the expected equilibrium point as shown in Fig. 6. The adaptation of the robot body to the environment as discussed in section 4.5 can be seen in Fig. 5 and is also confirmed by the data of the roll and pitch angle in Fig. 7. During the time on the terrain, the orientation of the robot changed accordingly to adapt to the terrain roughness and then it returned to the initial normal pose.

For the second terrain, the robot was required to not only overcome the terrain successfully but also track the given moving direction. The height map of this second terrain is similar to a hill; therefore, if the robot automatically moves forward, it might fail due to the extremely high inclined surfaces. For this reason, we manually controlled the direction of the robot to help it find a possible path to move. However, the turning and direction tracking task is performed automatically by the robot once the command is given. This may appear to be simple for the robot but it is actually still a difficult task because the robot did not have any information of the terrain and merely followed the desired direction. This simulation proved the capability of the proposed controller to manually control the robot in real time. For technical details, Fig. 8 shows an outline of the motion of the robot on the hill along with its COM trajectory. In addition, the associate yaw angle of the robot body during this task is depicted in Fig. 9. In this case, the robot started to move at $t = 2$ s and it stepped on the hill terrain at $t = 3$ s. As shown in Fig. 9, the robot followed the turning-left command at $t = 6.4$ s and then turned right at $t = 9$ s and so on. This given data clearly shows that the robot could track the commanded moving direction quite well while trotting stably.

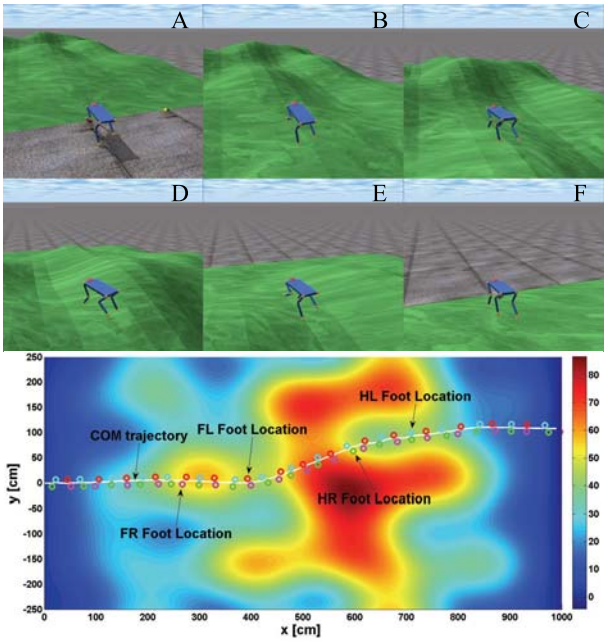


Fig. 8. The robot turning motion on a hill-like terrain.

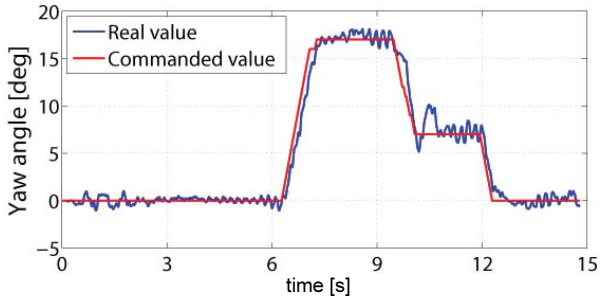


Fig. 9. Yaw angle of the robot body while turning on hill-like terrain.

5.4. Instability from the accuracy of the approximation

In our approach, the assumption of a "light mass" leg is a disadvantage since it is usually violated in case of real robots. All real robots have considerable leg mass of which the inertia effect can become a control difficulty. To test the effects of the leg mass on our proposed controller, we performed several simulations with different leg mass values.

In these simulations, the robot was commanded to trot forward on a flat terrain at the velocity of 0.903 ms^{-1} for 10 s. The stiffness and damping coefficient were chosen as for the previous simulations. The only different parameter among these simulations is the leg mass. We serially performed three tests with the leg mass of 0.4, 2, and 6 kg for the body mass of 2 %, 10 %, and 30 %, respectively. With a leg mass of 0.4 kg, the robot walked quite stably and the leg motion closely tracked the desired patterns. However, when the leg mass increased, the motion became more unstable. Due to the heaviness of the leg, the tracking was not sufficiently fast and accurate, resulting in disturbance within the motion. The rolling motion became greater in the case of 4 kg leg mass and almost became uncontrollable in the case of 6 kg. In addition, the tracking on the x axis was weaker

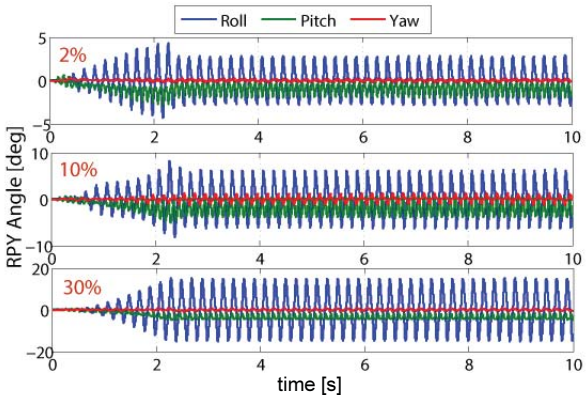


Fig. 10. RPY angles of the robot body in three tests of the effect of different leg mass.

in the heavy leg so that the motion of the robot could not reach the desired velocity and some juggling motion occurred. The results of roll-pitch-yaw angles in these tests are plotted in Fig. 10.

6. EXPERIMENTS

6.1. Outlines of the robot

For testing the proposed controller, we designed a robot named AiDIN III as depicted in Fig. 11. The robot has a total of 16 joints, and each leg has 3 actuated rotary joints and 1 passive linear spring-damper prismatic joint as shown in Fig. 11. Each rotary joint is controlled by a DC motor with an encoder to measure the joint angle. In addition, an IMU sensor is attached to the robot body to measure the orientation (roll-pitch-yaw angles), body linear acceleration, and rotational velocity. Each leg is also equipped with a loadcell to measure the linear force acting on the linear passive joint and the contact ON-OFF state of the leg. The details of the robot's dimensions and mass properties are summarized in Fig.

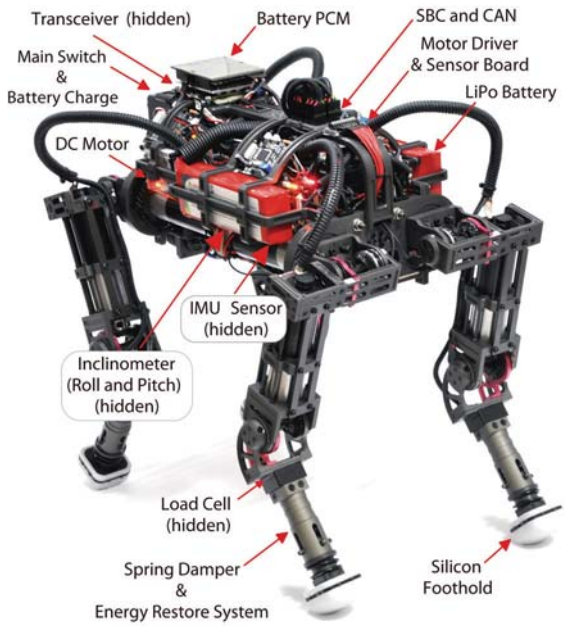


Fig. 11. The quadruped robot AiDIN III.

Table 2. Mass Properties of AiDIN III.

	Mass [kg]	Moment of Inertia × 10^{-3} [kg.m^2]
Body	8.195	I = [14.126, 73.262, 75.241]
Hip	0.812	I = [8.424, 8.498, 0.225]
Thigh	0.743	I = [6.163, 6.193, 0.168]
Tibia	0.165	I = [1.583, 1.597, 0.056]

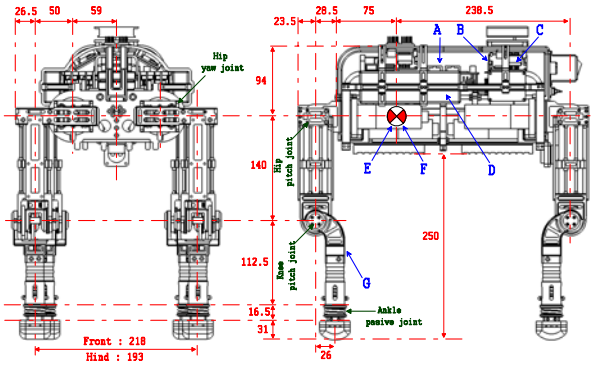


Fig. 12. Dimension details of AiDIN III.

12 and Table 2.

For communication, we set up a Xenomai-based Single Board Computer (SBC) on the robot. The motion command is performed in a separate computer that connects to the SBC via a wireless communication. Each motor is controlled by a dsPIC driver. These drivers communicate with the SBC through the CAN protocol. The calculation process in the SBC and the SBC-dsPIC CAN communication speed are performed every 4 ms (250Hz). The dsPIC runs at 1kHz frequency to control the motor via PWM signals.

It is important to note that AiDIN III has considerably heavy legs. The weight of each leg is about 21% of the body weight. This might cause disturbance in the frontal plane in which the robot rolling motion might be sustained at some level as discussed in the previous section. The robot COM is located not at the symmetrical point of the body but at its front part. In addition, the mechanical system has a backlash at a certain level. Hence, in the following experiments, the proposed controller has to face more severe realistic conditions in comparison with the proposed model in the presented simulations.

6.2. Rough terrain tests

In the experiments, three tests were performed with the proposed controller. The first test involves trotting on a simple flat terrain. The second test involves trotting down a terrain covered with small rocks and the final test involves trotting on a complex rough terrain.

For the first test, the robot was controlled to simply trot on a flat surface inside a corridor. The desired velocity is set to be 0.3ms^{-1}. This test is performed to show the normal state of the robot controller in which the terrain difficulty is minimized. The only task the controller must finish is to trot forward at the desired velocity. With a proper setting, the robot successfully completed the task. The commanded and real positions obtained in the controller are shown in Fig. 13. This

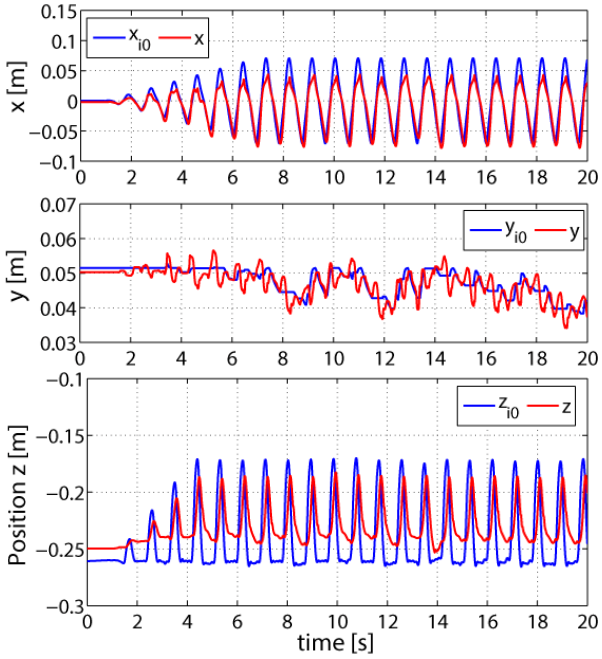


Fig. 13. The position tracking of the robot FL leg performing trotting on a flat terrain.

graph confirmed that the robot tracked the desired motion quite well despite the considerably large differences between AiDIN III and the simulation model as discussed in the previous section.

The second terrain is chosen to test the balancing ability of AiDIN III using the proposed controller. This terrain is an outdoor field covered with small stones of a similar size to the robot feet. In addition, the information of this terrain is not given to the robot controller. For this terrain, the supporting feet of the robot can easily slide over or dig under the stones such that the robot motion is considerably disturbed. The normal desired leg patterns might not be able to reject these disturbances and help to balance the robot. Fortunately, with the designed control method, AiDIN III successfully performed trotting on this terrain at the velocity of 0.12ms^{-1}. The outlines of this motion are shown in Fig. 14. In fact, the interaction between the feet and the stones caused many disturbances for the robot that required continuous actions to balance it. Using the proposed method, the commanded leg position continuously adapted to the terrain along with the detected changes of roll and pitch angles as shown in Fig. 14. Instead of being constant, as for the normal trotting on flat terrain shown in Fig. 13, the controlled height of the supporting feet is flexibly modulated to adapt to the environment.

In the final test, the robot was required to trot over a natural terrain that included a lot of grasses, various slopes, several holes, rocks, slippery sandy spots, etc. Similarly, this terrain condition is unknown to the robot, which caused severe disturbance to the robot motion. The robot needed to overcome not only standing on slippery stones or sand but also needed to adapt to the inclined shape of the terrain. With the proposed controller, the robot successfully trotted over this terrain

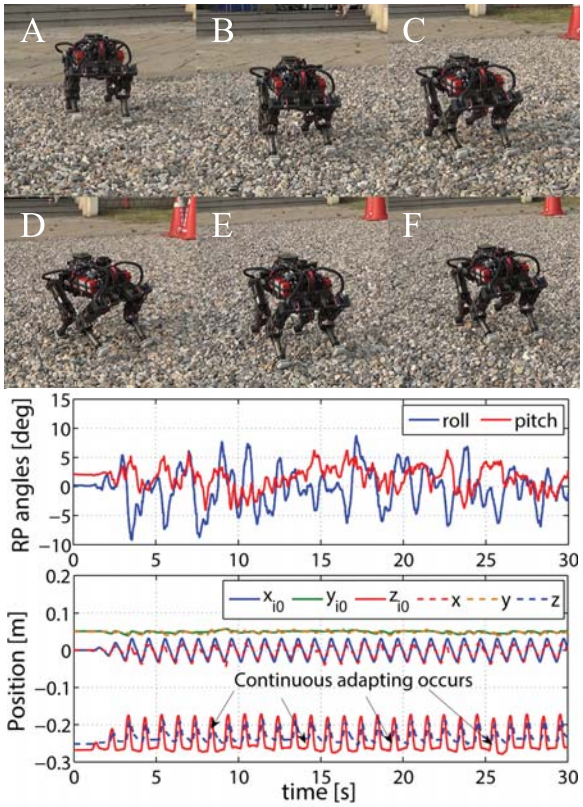


Fig. 14. This graph shows the roll-pitch angles and the adaptation of FL leg position of the robot performing trotting on a rocky terrain.

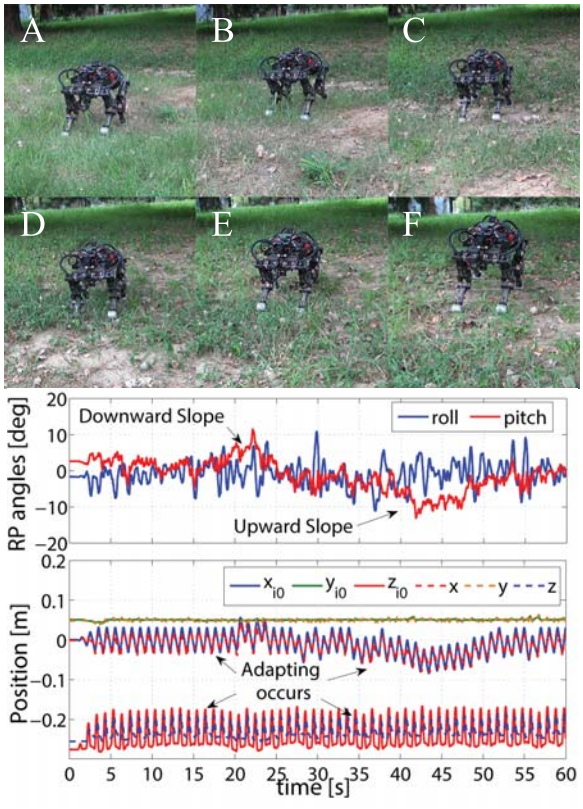


Fig. 15. This graph shows the roll-pitch angles and the adaptation of FL leg position of the robot performing trotting on a complex rough terrain.

at the velocity of $0.12ms^{-1}$. The outlines of the motion are shown in Fig. 15. In these experiments, the robot adaptation occurred not only in the z axis to balance the disturbance in the rolling plane but also in the x axis to help the robot adapt to the terrain's inclined surface. The details of these actions are shown in Fig. 15.

7. CONCLUSIONS

In this paper, we introduced a method to approximate the legs of a quadruped robot with 3D spring damper systems. In this way, the detailed dynamic equations of motion of the entire robot were easily derived in Cartesian space. In addition, a control strategy for the robot was proposed in which the error dynamics of the system exhibited asymptotic stability. For this control, minimum sensors were required. Finally, the efficiency of the proposed control model was validated by the dynamic simulations of a quadruped walking robot performing trotting and turning motions on several unknown rough terrains. In addition, real experiments with AiDIN III were performed on several rough terrains.

REFERENCES

[1] Y. Fukuoka, H. Kimura, and A. H. Cohen, "Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts," *Int. J. Robotics Research*, vol. 22, no. 3-4, pp. 287-202, 2003.

[2] L. R. Palmer III and D. E. Orin, "Quadrupedal running at high speed over uneven terrain," *Proc. of IEEE Int. Conf. Intelligent Robots and Systems*, San Diego, USA, pp. 303-308, 2007.

[3] D. T. Tran, I. M. Koo, G. L. Vo, S. Roh, S. Park, H. Moon, and H. R. Choi, "A new method in modeling central pattern generators to control quadruped walking robots," *Proc. of IEEE Int. Conf. Intelligent Robots and Systems*, St. Louis, USA, pp. 129-134, 2009.

[4] P. Holmes, "The dynamics of legged locomotion: Models, Analyzes, and Challenges," *SIAM Reviews*, vol. 48, no. 2, pp. 207-304, 2006.

[5] J. E. Seipel and P. Holmes, "Running in three dimensions: analysis of a point-mass sprung-leg model," *Int. J. Robotics Research*, vol. 24, no. 8, pp. 657-674, 2005.

[6] J. E. Seipel and P. Holmes, "Three-dimensional translational dynamics and stability of multilegged runners," *Int. J. Robotics Research*, vol. 25, no. 9, pp. 889-902, 2006.

[7] P. Gonzalez de Santos, E. Garcia, and J. Estremera, *Quadrupedal Locomotion an Introduction to the Control of Four-legged Robots*, Springer, 2006.

[8] R. Altendorfer, D. E. Koditschek, and P. Holmes, "Stability analysis of legged locomotion models by symmetry- factored return maps," *Int. J. Robotics Research*, vol. 23, no. 10-11, pp. 979-999, 2004.

[9] M. Ahmadi, H. Michalska, and M. Buehler, "Control and stability analysis of limit cycles in a hopping robot," *IEEE Trans. on Robotics*, vol. 23, no.

3, pp. 553-563, 2007.
[10] S. Meek, J. Kim, and M. Anderson, "Stability of a trotting quadruped robot with passive, underactuated legs," *Proc. of IEEE Int. Conf. Intelligent Robotics and Automation*, Pasadena, USA, pp. 347-352, 2008.
[11] J. Buchli and A. J. Ijspeert, "Self-organized adaptive legged locomotion in a compliant quadruped robot," *Autonomous Robot*, vol. 25, pp. 331-347, 2008.
[12] M. D. Berkemeier and P. Sukthankar, "Self- organizing running in a quadruped robot model," *Proc. of IEEE Int. Conf. Intelligent Robotics and Automation*, Barcelona, Spain, pp. 4108-4113, 2005.
[13] H. Zou and J. Schmiedeler, "The effect of asymmetrical body-mass distribution on the stability and dynamics of quadruped bounding," *IEEE Trans. on Robotics*, vol. 22, no. 4, pp. 711-723, 2006.
[14] J. Rummel and A. Seyfarth, "Stable running with segmented legs," *Int. J. Robotics Research*, vol. 27, no. 8, pp. 919-934, 2008.
[15] J. J. Craig, *An Introduction to Robotics: Mechanics and Control*, 3rd edition, Pearson, Prentice Hall, 2005.
[16] Open Dynamic Engine, www.ode.org.
[17] M. Won, T. H. Kang, and W. K. Chung, "Gait planning for quadruped robot based on dynamic stability: landing accordance ratio," *Intel. Serv. Robotics*, 2, pp. 105-112, 2009.

Duc Trong Tran received his B.S. degree in Mechatronics from Ho Chi Minh City University of Technology in Vietnam in 2005. He also obtained his M.S. and Ph.D. degrees in Mechanical Engineering from the Sungkyunkwan University in Korea in 2008 and 2013, respectively. His research interests include biological inspired control and adaptive control of quadruped walking robot.

Ig Mo Koo received his B.S. degree in Mechanical Engineering from Myongji University, Yongin, Korea. He also obtained his M.S. and Ph.D. degrees in Mechanical Engineering from the Sungkyunkwan University, Suwon, Korea. His research interests include artificial muscle actuator, haptics, tactile display, walking and climbing robots, biomimetics.

Yoon Haeng Lee received his B.S. degree in Mechanical Engineering from Won-kwang University, Iksan, Korea, in 2012, and his M.S degree in Mechanical Engineering from the Sungkyunkwan University, Suwon, Korea, in 2014, where he is currently working toward a Ph.D. degree in Mechanical Engineering from Sungkyunkwan University, Intelligent Robotics and Mechatronics System Laboratory. His research interests include quadruped walking robot system.

Hyungpil Moon received his B.S. and M.S. degrees in Mechanical Engineering from Pohang University of Science and Technology (POSTECH), Pohang, Korea, in 1996 and 1998, respectively, and his Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, in 2005. He was a postdoctoral fellow at Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. He is currently an assistant professor in the school of mechanical engineering, Sungkyunkwan University, Suwon, Korea. He is also an Associate Editor of Journal of Mechanical Science and Technology (JMST). His research interests include distributed manipulation, localization and navigation of multi-agent systems, and biomimetic robotics.

Jachoon Koo received his B.S. degree from Hanyang University, Seoul, in 1989, and his M.S. and Ph.D. degrees from the University of Texas at Austin, in 1997 and 1992, respectively. He is currently a Professor in the School of Mechanical Engineering, Sungkyunkwan University, Suwon, Korea. He was an Advisory Engineer at IBM, San Jose, CA, and a Staff Engineer at Samsung Information Systems America, San Jose. His current research interests include modeling of dynamics systems, sensors, and actuators.

Sangdeok Park is currently with the Division of Applied Robot Technology, Korea Institute of Industrial Technology, Ansan, Korea.

Hyouk Ryeol Choi received his B.S. degree from Seoul National University, Seoul, Korea, in 1984, an M.S. degree from the Korea Advanced Technology of Science and Technology (KAIST), Taejon, Korea, in 1986, and a Ph.D. degree from the Pohang University of Science and Technology (POSTECH), Pohang, Korea, in 1994. Since 1995, he has been with Sungkyunkwan University, Suwon, Korea, where he is currently a Professor in the School of Mechanical Engineering. He was an Associate Engineer with LG Electronics Central Research Laboratory, Seoul, Korea, from 1986 to 1989. From 1993 to 1995, he was with Kyoto University, Kyoto, Japan, as a grantee of scholarship funds from the Japanese Educational ministry. He visited the Advanced Institute of Industrial Science Technology (AIST), Tsukuba, Japan, as a JSPS Fellow from 1999 to 2000. He was a visiting scholar in Biorobotics Laboratory in the University of Washington from 2008 to 2009. He served as an Associate Editor in IEEE Transactions on Robotics. He is now an Editor of International Journal of Control, Automation and Systems (IJCAS), and Associate Editor of Journal of Intelligent Service Robotics and General Chair of IEEE CASE. His interests include dexterous mechanisms, field application of robots, and artificial muscle actuators.