

## Research Article

# Gait Tracking Control of Quadruped Robot Using Differential Evolution Based Structure Specified Mixed Sensitivity $H_\infty$ Robust Control

Petrus Sutyasadi and Manukid Parnichkun

Mechatronics, Asian Institute of Technology, P.O. Box 4, Klong Luang, Pathumthani 12120, Thailand

Correspondence should be addressed to Petrus Sutyasadi; st114299@ait.ac.th

Received 1 March 2016; Revised 14 July 2016; Accepted 17 July 2016

Academic Editor: Carlos-Andrés Garcia

Copyright © 2016 P. Sutyasadi and M. Parnichkun. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposed a control algorithm that guarantees gait tracking performance for quadruped robots. During dynamic gait motion, such as trotting, the quadruped robot is unstable. In addition to uncertainties of parameters and unmodeled dynamics, the quadruped robot always faces some disturbances. The uncertainties and disturbances contribute significant perturbation to the dynamic gait motion control of the quadruped robot. Failing to track the gait pattern properly propagates instability to the whole system and can cause the robot to fall. To overcome the uncertainties and disturbances, structured specified mixed sensitivity  $H_\infty$  robust controller was proposed to control the quadruped robot legs' joint angle positions. Before application to the real hardware, the proposed controller was tested on the quadruped robot's leg planar dynamic model using MATLAB. The proposed controller can control the robot's legs efficiently even under uncertainties from a set of model parameter variations. The robot was also able to maintain its stability even when it was tested under several terrain disturbances.

## 1. Introduction

Legged robot has the advantage over wheeled robot in motion when the surface of workplace is not well structured [1]. Furthermore, the leg locomotion has a unique capability of obstacles avoiding or surmounting compared to the wheeled locomotion [2]. Therefore, the research of legged robot still attracts interest of many researchers worldwide [3]. The movement of the legged robot relies on the gait pattern being implemented. Interlimb coordination manages the footfall pattern that forms the gait pattern [4, 5]. BigDog from Boston Dynamics applied robust trotting gait to manage the robot to walk or trot on any terrains [6]. Good gait tracking has significant contribution to maintaining the center of mass (CoM) of the trunk to stay inside the stability margin. In trotting gait, when the stance or swing diagonal legs do not follow the gait trajectory, the trunk becomes unstable. The trunk depends upon the robot's pitch and roll angles. Body pitch and roll angle significantly affect the motion

control and overall stability [7]. Golubovic and Hu improved quadruped robot's gait through an evolutionary algorithm [8]. Chung et al. proved that their gait design method enabled a steady hexapod walking even on a rugged terrain [9]. Ugurlu et al. [10] state that, according to some experiments with their HyQ quadruped robot, considering only sagittal motion is enough to achieve stable trot walk. Therefore it does not have to sway in lateral direction following Yoneda and Hirose method [11] to stabilize the trot. Since it only incorporates joints in sagittal planar motion then it is clear that the gait tracking performance is important. Some planar motion studies in sagittal plane to control quadruped robot have been conducted also in [12, 13], but no real hardware implementation has been reported.

This research proposes a robust gait tracking trajectory control that guarantees robust performance and robust stability under uncertainties, such as model dynamics or external disturbance. The algorithm is then implemented and verified on AIT quadruped robot.

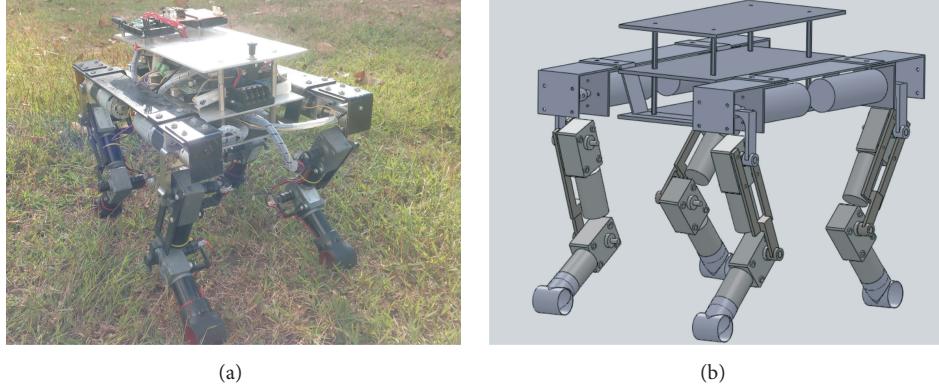


FIGURE 1: (a) AIT quadruped robot. (b) Mechanical frame design.

TABLE 1: Hardware specification of AIT quadruped robot.

Specification	Value
Length	400 mm
Width	281 mm
Height	285 mm
Weight	10,25 kg
Battery	12 V × 2
Degree of freedom	12 DoF
DC motor power	18 W × 8, 25 W × 4

## 2. Quadruped Robot Mechanical Structure

The quadruped robot is made of steel and aluminium frame. It has 12 degrees of freedom, where there is 3 degrees of freedom on each leg. The overall specification is given in Table 1.

AIT quadruped robot and its mechanical frame design are shown in Figure 1. The detail of the kinematic design and the D-H parameters of the leg is shown in Figure 2.

Based on the reference frames embedded at the joints of the robot, the D-H parameters are given in Table 2.

The homogenous transformation matrices of the quadruped robot are obtained:

$${}^0A_1 = \begin{pmatrix} \cos \theta_1 & 0 & \sin \theta_1 & a_1 \cos \theta_1 \\ \sin \theta_1 & 0 & -\cos \theta_1 & a_1 \sin \theta_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$${}^1A_2 = \begin{pmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & a_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (1)$$

$${}^2A_3 = \begin{pmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & a_3 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & a_3 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Therefore, the homogeneous transformation matrix is represented by

$$\begin{aligned} {}^0A_3 &= {}^0A_1 {}^1A_2 {}^2A_3 = \begin{pmatrix} R_{3 \times 3} & d_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{pmatrix} \\ &= \begin{pmatrix} \cos \theta_1 \cos (\theta_2 + \theta_3) & -\sin (\theta_2 + \theta_3) \cos \theta_1 & \sin \theta_1 & \cos \theta_1 (a_3 \cos (\theta_2 + \theta_3) + a_2 \cos \theta_2 + a_1) \\ \sin \theta_1 \cos (\theta_2 + \theta_3) & -\sin (\theta_2 + \theta_3) \cos \theta_1 & -\cos \theta_1 & \sin \theta_1 (a_3 \cos (\theta_2 + \theta_3) + a_2 \cos \theta_2 + a_1) \\ \sin (\theta_2 + \theta_3) & \cos (\theta_2 + \theta_3) & 0 & a_3 \sin (\theta_2 + \theta_3) + a_2 \sin \theta_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \end{aligned} \quad (2)$$

and  $d = (x, y, z)^T$  is the foot position vector, where

$$x = \cos \theta_1 (a_3 \cos (\theta_2 + \theta_3) + a_2 \cos \theta_2 + a_1), \quad (3)$$

$$y = \sin \theta_1 (a_3 \cos (\theta_2 + \theta_3) + a_2 \cos \theta_2 + a_1), \quad (4)$$

$$z = a_3 \sin (\theta_2 + \theta_3) + a_2 \sin \theta_2. \quad (5)$$

Inverse kinematics of the robot's leg are analyzed to determine joint angles based on foot position [2]. Multiply (3) with  $\sin (\theta_1)$ , multiply (4) with  $\cos (\theta_1)$ , and then subtract them:

$$x \sin (\theta_1) - y \cos (\theta_1) = 0,$$

$$\tan \theta_1 = \frac{\sin \theta_1}{\cos \theta_1} = \frac{y}{x}. \quad (6)$$

Thus,

$$\theta_1 = \tan^{-1} \left( \frac{y}{x} \right). \quad (7)$$

From (3), (4), and

$$\sin^2 \theta_j + \cos^2 \theta_j = 1, \quad (8)$$

hence

$$\cos(\theta_2 + \theta_3) = \frac{x \cos \theta_1 + y \sin \theta_1 - a_2 \cos \theta_2 - a_1}{a_3}. \quad (9)$$

From (5),

$$\sin(\theta_2 + \theta_3) = \frac{z - a_2 \sin \theta_2}{a_3}. \quad (10)$$

Substitute  $\cos(\theta_2 + \theta_3)$  and  $\sin(\theta_2 + \theta_3)$  from (9) and (10) into (8) to get

$$A \cdot \sin \theta_2 + B \cdot \cos \theta_2 = D, \quad (11)$$

where

$$A = -z,$$

$$B = a_1 - (x \cos \theta_1 + y \sin \theta_1),$$

$$D = \frac{2a_1(x \cos \theta_1 + y \sin \theta_1) + a_3^2 - a_2^2}{2a_2} \quad (12)$$

$$+ \frac{-a_1^2 - z^2 - (x \cos \theta_1 + y \sin \theta_1)^2}{2a_2}.$$

From (11) [14],

$$A = r \cos \varphi, \quad (13)$$

$$B = r \sin \varphi.$$

Then,

$$\varphi = \tan^{-1} \left( \frac{B}{A} \right), \quad (14)$$

$$r = \pm \sqrt{A^2 + B^2}.$$

Equation (11) can be written as

$$\cos \varphi \sin \theta_2 + \sin \varphi \cos \theta_2 = \frac{D}{r}, \quad (15)$$

$$\sin(\varphi + \theta_2) = \frac{D}{r}.$$

From (8),

$$\cos(\varphi + \theta_2) = \pm \sqrt{1 - \sin^2(\varphi + \theta_2)} = \frac{\pm \sqrt{r^2 - D^2}}{r}. \quad (16)$$

Thus,

$$\tan(\varphi + \theta_2) = \frac{D}{\pm \sqrt{r^2 - D^2}}, \quad (17)$$

$$\theta_2 = -\varphi + \tan^{-1} \left( \frac{D}{\pm \sqrt{r^2 - D^2}} \right).$$

TABLE 2: D-H parameters of AIT quadruped robot.

Parameter	Link 1	Link 2	Link 3
$a_i$	$a_1$	$a_2$	$a_3$
$\alpha_i$	$\pi/2$	0	0
$d_i$	0	0	0
$\theta_i$	$\theta_1$	$\theta_2$	$\theta_3$

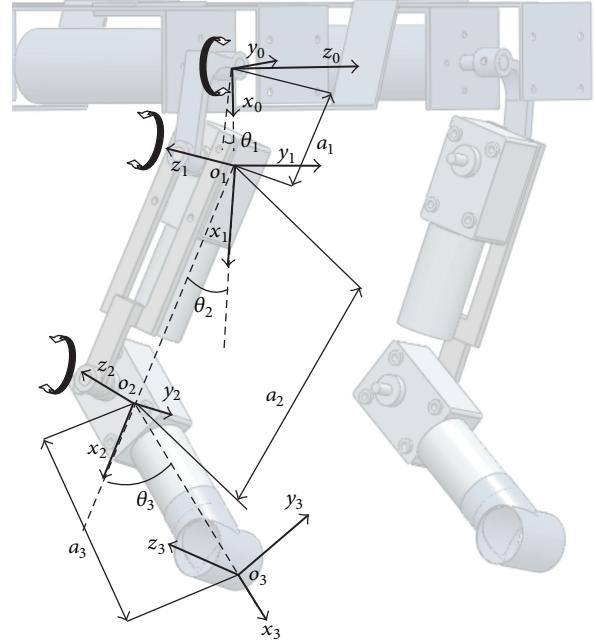


FIGURE 2: D-H parameters of AIT quadruped robot's leg.

Substitute the value of  $\varphi$  and  $r$  in (14):

$$\theta_2 = -\tan^{-1} \left( \frac{B}{A} \right) + \tan^{-1} \left( \frac{D}{\pm \sqrt{r^2 - D^2}} \right). \quad (18)$$

There are two solutions. The valid solution depends on the mechanical constraints.

From (9) and (10),

$$\theta_3 = \tan^{-1} \left( \frac{z - a_2 \sin \theta_2}{x \cos \theta_1 + y \sin \theta_1 - a_2 \cos \theta_2 - a_1} \right) - \theta_2. \quad (19)$$

### 3. Quadruped Robot Control Architecture

The quadruped robot control architecture is divided into two levels: high level control and low level control. The high level control manages the behavior of the robot including the gait generation and balancing control. The low level control is dedicated to controlling the leg's joint position control. The low level controller receives commands from the high level controller to manage the legs positions. The low level controller is required to track the commanded gait trajectory. The proposed robust controller will be implemented in the low level control. The overall control strategy for the quadruped can be seen in Figure 3.

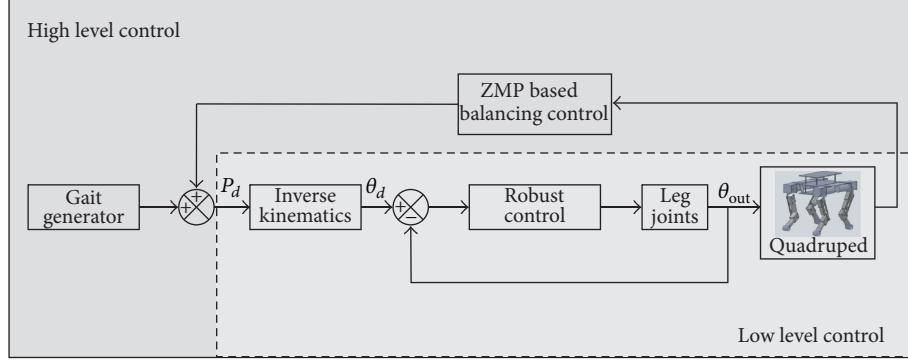


FIGURE 3: Control block diagram of the quadruped robot, where  $P_d$ ,  $\theta_d$ , and  $\theta_{out}$ , respectively, denote foot position reference, joint angle reference, and actual joint angles.

**3.1. Balancing Control.** Quadruped robot is supported by only two legs in some part or whole phase while walking in dynamic trot gait. The robot starts to fall sideways when the diagonally paired swinging legs rise. When the quadruped trots it can be modeled as biped [15]. Figure 4 shows in detail how it can be modeled as biped. As the diagonally paired leg always moves at the same time with the same trajectory then it acts as one of the biped's legs. The center of mass of the quadruped trunk becomes the point mass of the biped model.

Many researchers conduct balancing control of biped using linear inverted pendulum model (LIPM) [16–18]. In this research we also use LIPM to balance the quadruped after being simplified into biped model by employing ZMP concept.

Suppose Center of Gravity (CG) of a massless legs compass biped travels horizontally, and a force  $F$  and a moment  $M$  act on the CG. According to Yoneda and Hirose [11], if the friction between the leg and the ground is high, ZMP is the point on the ground at which the moment  $M$  and the force  $L$  acting on the CG can be resisted by applying a force only without moment. They also introduce ZMP balancing for quadruped robot by applying dynamic  $y$ -directional translation of the body [19]. They tried to keep the ZMP staying in the supporting line made by the diagonally paired legs.

ZMP general equations are

$$\begin{aligned} x_{ZMP} &= \frac{\sum m_i (\ddot{z}_i + g) x_i - \sum m_i \ddot{x}_i z_i}{\sum m_i (\ddot{z}_i + g)}, \\ y_{ZMP} &= \frac{\sum m_i (\ddot{z}_i + g) y_i - \sum m_i \ddot{y}_i z_i}{\sum m_i (\ddot{z}_i + g)}, \end{aligned} \quad (20)$$

where  $x_{ZMP}$  and  $y_{ZMP}$  are the coordinates for the ZMP and  $g, m_i, x_i, y_i$ , and  $z_i$  denote the gravity, mass, and the position of link, respectively. The equations of simplified model to single mass inverted pendulum are

$$\begin{aligned} x_{ZMP} &= x_{cg} - \frac{\ddot{x}_{cg} z_{cg}}{\ddot{z}_{cg} + g}, \\ y_{ZMP} &= y_{cg} - \frac{\ddot{y}_{cg} z_{cg}}{\ddot{z}_{cg} + g}, \end{aligned} \quad (21)$$

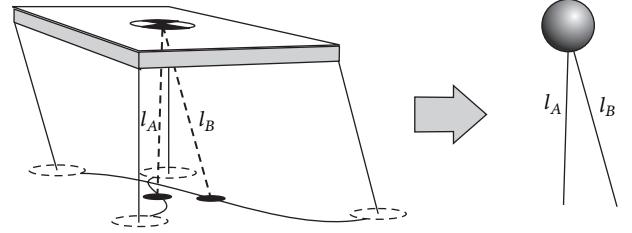


FIGURE 4: Simplified model of trotting quadruped into biped model.

where  $x_{cg}$  and  $z_{cg}$  denote the center of gravity position of the pendulum single mass. In our case the trunk of the robot is maintained with constant height all the time; then the equation becomes

$$\begin{aligned} x_{ZMP} &= x_{cg} - \frac{\ddot{x}_{cg} z_{cg}}{g}, \\ y_{ZMP} &= y_{cg} - \frac{\ddot{y}_{cg} z_{cg}}{g}. \end{aligned} \quad (22)$$

ZMP criterion is based on its position in a supporting polygon. In real application of quadruped trot gait, the supporting line from the diagonally paired legs is not a perfect thin line segment but a supporting stripe because the foot that touches the ground has area [20]. The geometry of the AIT quadruped foot is shown in Figure 5. The width of the shoe is 30 mm and the round surface allows the foot to make wider contact with the ground along its rotation while walking to about 8 mm.

From the geometry of the foot, dimension of the robot, and trot gait parameters, the supporting stripe of the paired diagonal legs and the ZMP reference trajectory can be designed. The detail of the ZMP reference trajectory and the supporting stripes can be shown in Figure 6.

With the foot stepping area of  $30 \text{ mm} \times 8 \text{ mm}$ , it forms a significant supporting stripe as the supporting polygon for the ZMP. With stride length of 30 mm we can design the ZMP trajectory to stay on a straight line along the  $x$ -axis, where  $x$  is the axis along the direction of moving and  $y$  is the axis parallel to the lateral motion.

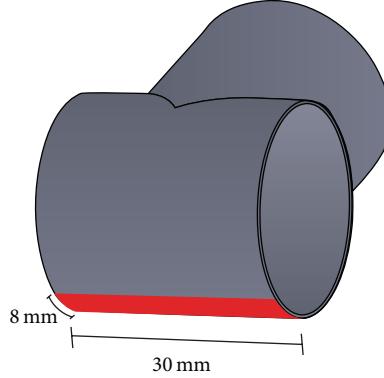
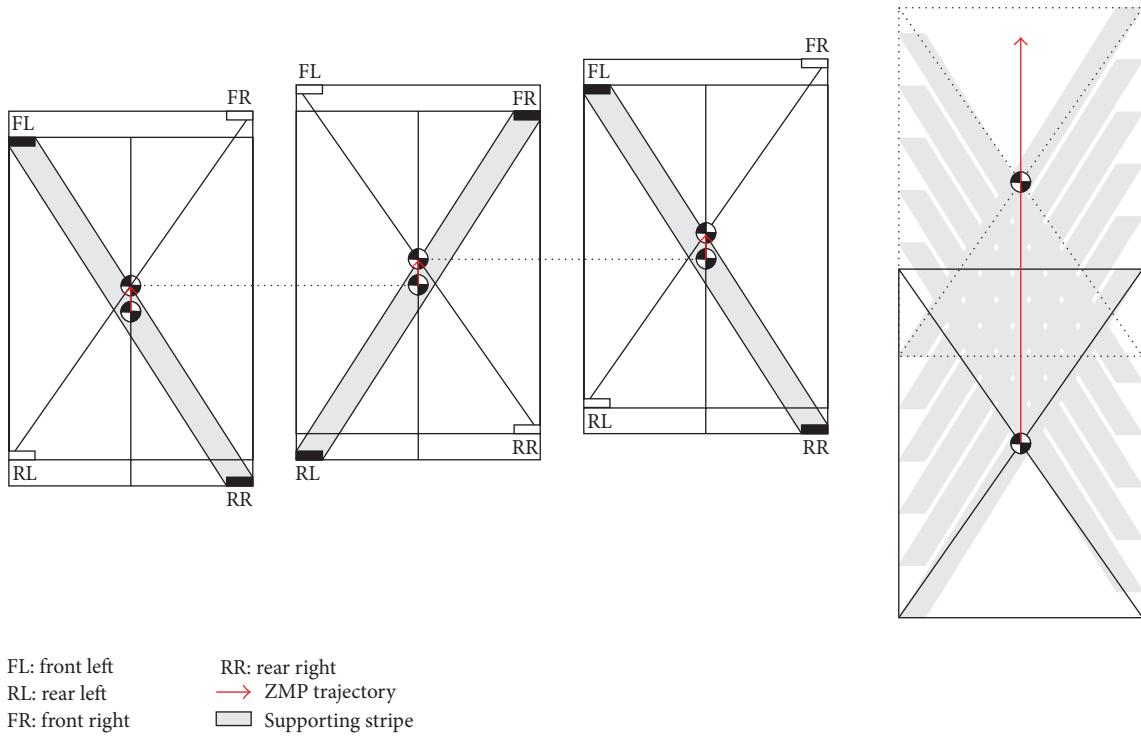


FIGURE 5: Foot geometry of AIT quadruped and area contact with the ground.

FIGURE 6: Supporting stripe as the supporting polygon for the ZMP. The ZMP trajectory can be made straight along the  $x$ -axis.

Overall balancing is decoupled into transverse and sagittal plane control. The CoM of the robot will be moved in  $x$ - $y$  plane of the robot to the predefined trajectory. For the lateral motion in the transverse plane the robot is modeled as compass biped model.

**3.2. Robust Control.** As previously stated, trotting control of a quadruped robot can be done by considering its sagittal motion only because the diagonally paired legs equally balance the trunk while walking [10]. Therefore, only joints on the sagittal plane, that is, hip and knee joints for flexion/extension, will be considered. Therefore, the robot trotting motion is done by controlling each foot to form a certain planar gait trajectory. Hip joint for abduction/adduction is not included because it is responsible for lateral motion

only, for example, while balancing and turning. Control of two-degrees-of-freedom leg with hip and knee joints to follow such planar gait trajectory is considered as a two-link manipulator control. Figure 7 shows two-link kinematic configuration and example trajectory of half-sinusoidal gait pattern.

Lagrange equation of the two-link manipulator is expressed in (23) with  $M$ ,  $C$ , and  $G$  as the mass, Coriolis, and gravity matrix, respectively, and  $\tau$  is the torque:

$$M(\theta) \cdot \ddot{\theta} + C(\theta, \dot{\theta}) \cdot \dot{\theta} + G(\theta) = \tau, \quad (23)$$

where

$M$

$$= \begin{bmatrix} (m_1 + m_2)l_1^2 + m_2l_2^2 + 2m_2l_1l_2 \cos \theta_2 & m_2l_2^2 + m_2l_1l_2 \cos \theta_2 \\ m_2l_2^2 + m_2l_1l_2 \cos \theta_2 & m_2l_2^2 \end{bmatrix},$$

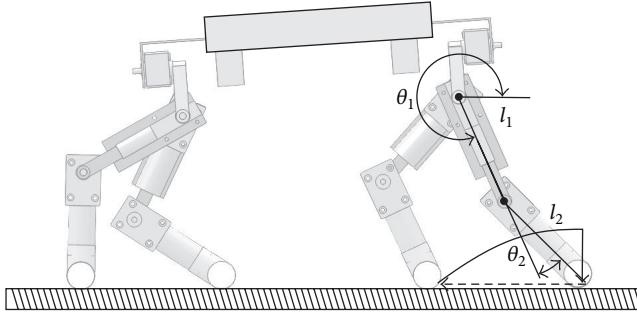


FIGURE 7: Half-sinusoidal gait pattern and kinematic design of two-link planar manipulator representing each leg of the robot.

$$\begin{aligned} C &= \begin{bmatrix} -m_2 l_1 l_2 (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) \sin \theta_2 \\ m_2 l_1 l_2 \dot{\theta}_1^2 \sin \theta_2 \end{bmatrix}, \\ G &= \begin{bmatrix} (m_1 + m_2) g l_1 \cos \theta_1 + m_2 g l_2 \cos(\theta_1 + \theta_2) \\ m_2 g l_2 \cos(\theta_1 + \theta_2) \end{bmatrix}, \end{aligned} \quad (24)$$

with  $m_j$ ,  $l_j$ , and  $\theta_j$  are the mass, length, and angular position of link  $j$ th.

The two-link manipulator that represents the planar leg in sagittal plane consists of two DC motors with potentiometers as the position sensors. The motors are controlled using Arduino UNO microcontroller and VNH30SP DC motor driver Arduino shield circuit. The detailed diagram of the system is shown in Figure 8(a). Overall electronic circuit diagram of the quadruped robot is shown in Figure 8(b).

The system is a nonlinear and coupled dynamic system. It needs to be linearized in order that linear  $H_\infty$  robust controller can be applied. Feedback linearization using Computed Torque Control method is applied. Computed Torque Control is a traditional technique for manipulator control [14, 21–23]. This method transforms a multivariable nonlinear plant into a set of decoupled linear systems and provides good control performance when there are no uncertainties from external disturbance and unmodeled dynamics [24]. Block diagram of the feedback linearization of the quadruped robot is shown in Figure 9. The linear robust controller is in the outer loop, while the coupled nonlinear system is in the inner loop [23].

$H_\infty$  is an optimal controller that works based on the optimal condition among the bonded uncertainties that may happen. In an ideal condition, a simple PD controller may have better performance than the robust controller. However, in reality the robot does not always work in ideal condition. In other words, the robust controller does not provide the best performance of the system but provides the optimal performance in the range of uncertainties [25].

#### 4. Structure Specified Mixed Sensitivity $H_\infty$ Robust Control

Feedback linearization is applied to decouple the system. The leg in planar motion acts as a two-link manipulator system.

Each link is actuated by a DC motor. Total inertia of the DC motor system is the combination of inertia from the rotor and load attached to the motor shaft. Likewise, model uncertainties of the hip motor and the knee motor are also different. The hip joint has more load as it carries the whole leg and foot as its load. Therefore, the two joints have different transfer functions. Figure 10 shows the equivalent electrical and mechanical model of a DC motor.

The motor torque is proportional to the armature current:

$$T = Ki. \quad (25)$$

The back emf is proportional to the angular velocity of the rotor:

$$e = K\dot{\theta}. \quad (26)$$

From Figure 10 and by considering (25) and (26), transfer function of the DC motor can be derived as

$$\begin{aligned} J\ddot{\theta} + b\dot{\theta} &= Ki, \\ L\frac{di}{dt} + Ri &= V - K\dot{\theta}, \end{aligned} \quad (27)$$

where  $T$  is motor torque,  $K$  is motor constant,  $i$  is motor current,  $e$  is back emf constant,  $R$  is armature resistance,  $L$  is armature inductance,  $J$  is rotor inertia,  $b$  is viscous friction constant,  $\theta$  is rotor angular position,  $\dot{\theta}$  is rotor angular velocity, and  $V$  is input voltage.

The transfers function of the system with input voltage  $V(s)$  and output angular position  $\Theta(s)$  becomes

$$\frac{\Theta(s)}{V(s)} = \frac{K}{(s((J+b)(L+R)+K^2))}. \quad (28)$$

After substitution of hip motor parameters into (28), the nominal transfer function of the hip motor attached with thigh structure, shank structure, and knee motor as the load is obtained as

$$\begin{aligned} \frac{\Theta(s)}{V(s)} &= \frac{0.0197}{1.935 \exp -11s^3 + 1.547 \exp -05s^2 + 0.0003938s}. \end{aligned} \quad (29)$$

After substitution of knee motor parameters into (28), the nominal transfer function of the knee motor attached with the shank structure as the load is obtained as

$$\begin{aligned} \frac{\Theta(s)}{V(s)} &= \frac{0.0197}{9.687 \exp -12s^3 + 7.743 \exp -06s^2 + 0.0003938s}. \end{aligned} \quad (30)$$

**4.1. Model Uncertainty.** Figure 11 shows multiplicative uncertainty affecting a single-input single-output system.  $G_n(s)$  is the nominal system,  $\Delta G(s)$  is the system perturbation,  $K(s)$  is the controller,  $r(t)$  is the reference input,  $e(t)$  is the tracking

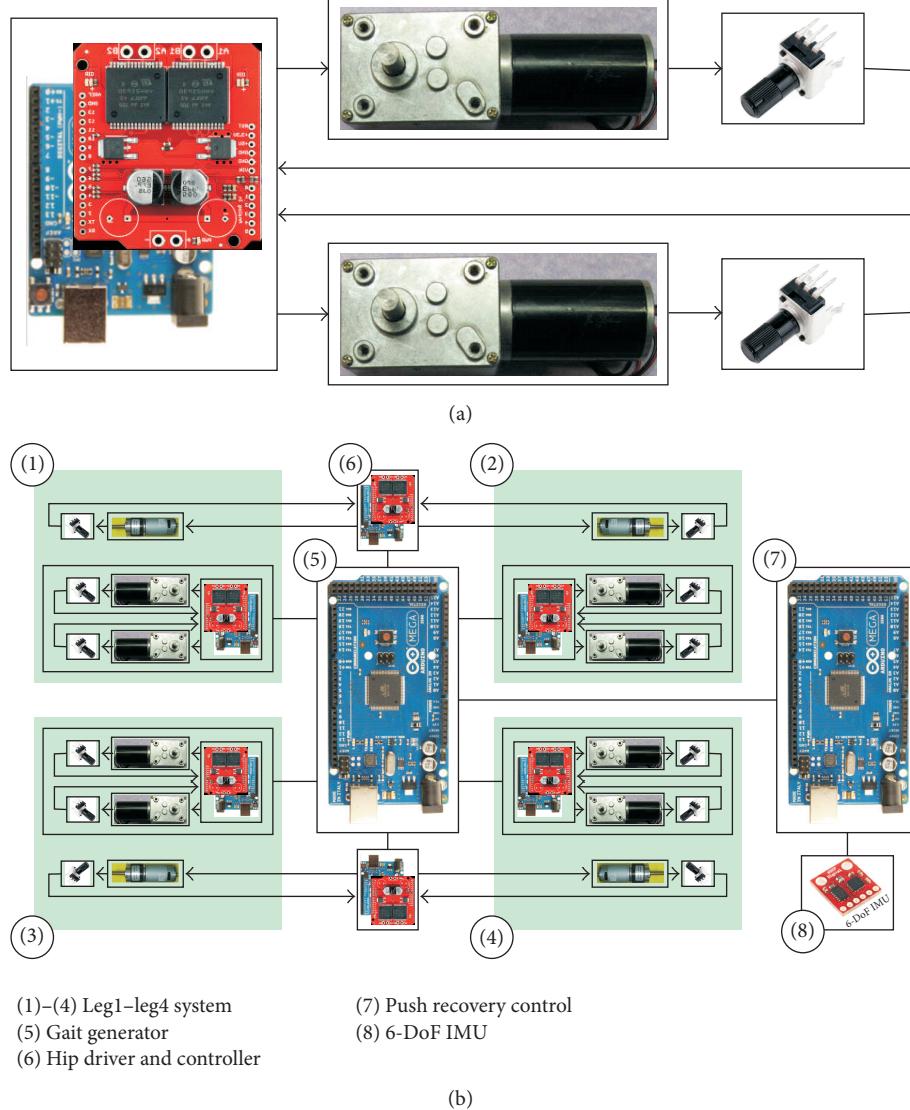


FIGURE 8: Controller architecture of the quadruped robot. (a) Hip and knee controller block diagram. (b) Overall quadruped robot controller block diagram.

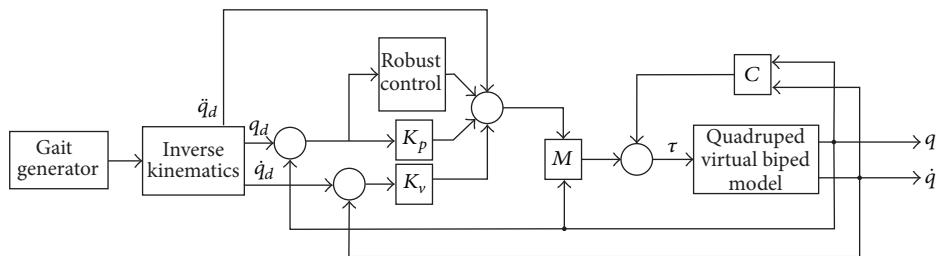


FIGURE 9: Feedback linearization block diagram of the gait trajectory control.

error,  $d(t)$  is the external disturbance, and  $y(t)$  is the output of the system.

The perturbed system is expressed by

$$\tilde{G}(s) = G_n(s)(1 + \Delta(s)). \quad (31)$$

Thus, the multiplicative system perturbation is determined from

$$\Delta(s) = \left( \frac{\tilde{G}(s)}{G_n(s)} - 1 \right). \quad (32)$$

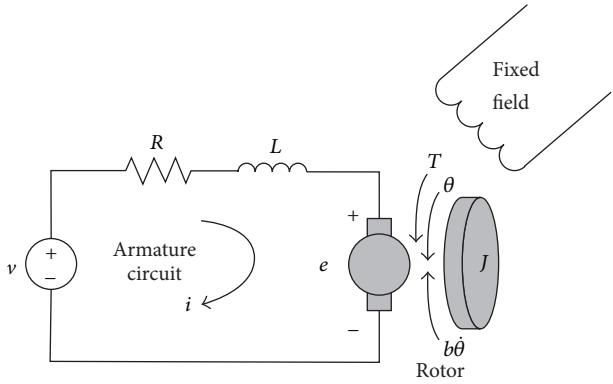


FIGURE 10: Equivalent electrical circuit and mechanical diagram of a DC motor.

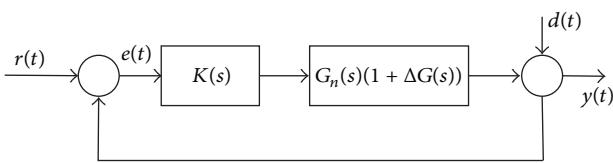


FIGURE 11: Single-input single-output controlled system with perturbation.

Model uncertainty exists due to inaccuracy and variation of inertial and noninertial matrices of the manipulator model during system identification process [24]. The controller synthesis includes  $\pm 30\%$  of load inertia and motor armature resistance variation as the model uncertainties in the form of multiplicative perturbation. Singular plot of model uncertainty  $\Delta(s)$  for the hip joint control is depicted in Figure 12, while singular plot of model uncertainty  $\Delta(s)$  for the knee joint control is shown in Figure 13.

**4.2. Mixed Sensitivity  $H_\infty$  Control.** If a controller,  $K(s)$ , is designed so that the nominal closed loop system ( $\Delta(s) = 0$  and  $d(t) = 0$ ) is asymptotically stable, the robust stability performance against external disturbance satisfies the following inequality [25]:

$$J_{\infty,a} = \|W_s(s)S(s)\|_\infty < 1. \quad (33)$$

A known stable function  $W_s(s)$  is used to attenuate external disturbance. The multiplicative perturbation is upper bounded by a known stable function  $W_t(s)$ :

$$\Delta(s)_\infty \leq W_t(s)_\infty. \quad (34)$$

The robust stability against system perturbation satisfies the following inequality:

$$J_{\infty,b} = \|W_t(s)T(s)\|_\infty < 1, \quad (35)$$

where  $W_s(s)$  is sensitivity weight,  $W_t(s)$  is transmissibility or the complementary sensitivity weight,  $S(s)$  is the sensitivity function, and  $T(s)$  is the complementary sensitivity function. Structure of the controlled system is shown in Figure 14.

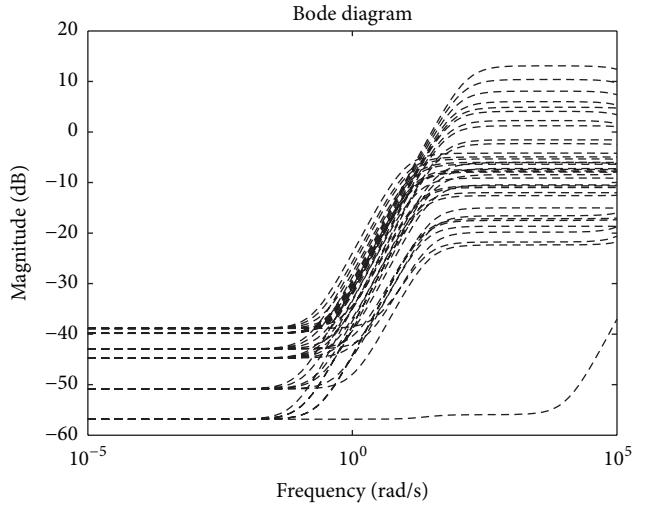


FIGURE 12: Model uncertainty of hip motor and its load singular plots.

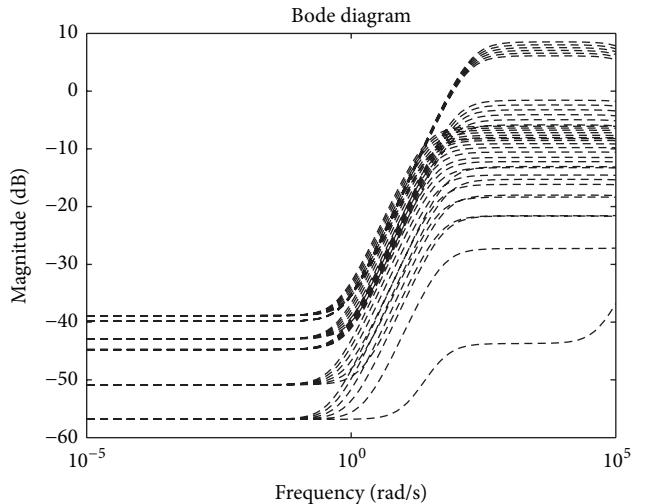


FIGURE 13: Model uncertainty of knee motor and its load singular plots.

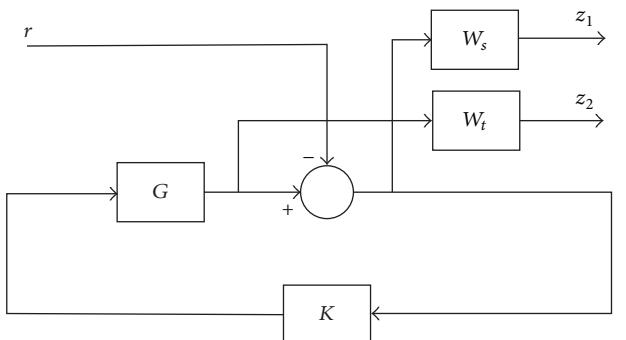
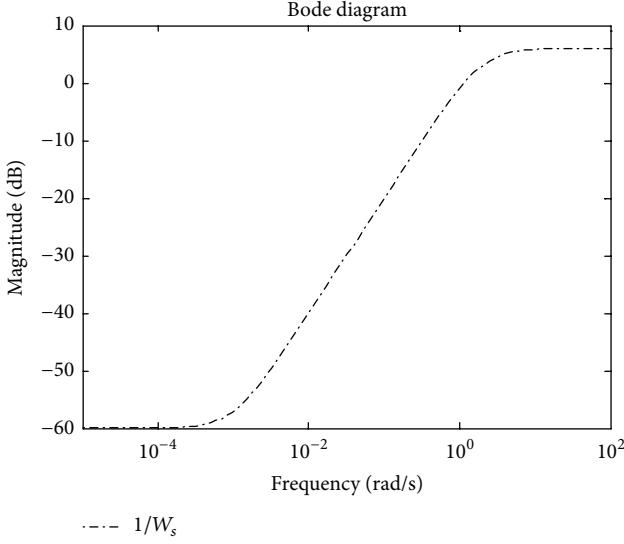
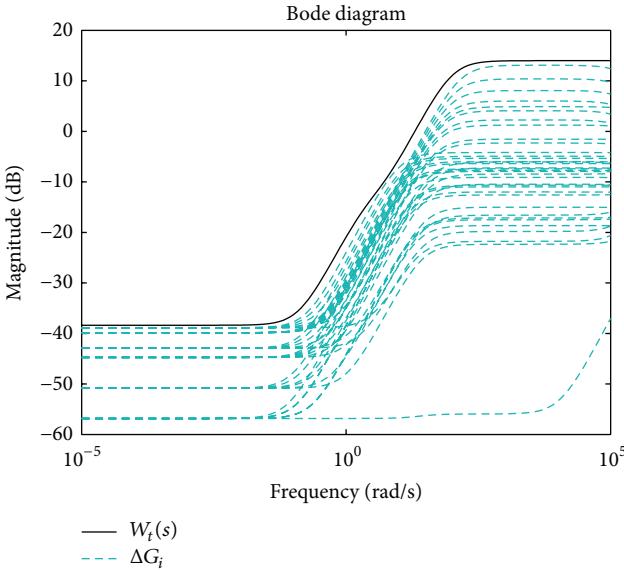


FIGURE 14: Mixed sensitivity control block diagram.

FIGURE 15: Inverse of sensitivity weight  $W_s(s)$ .FIGURE 16: Complementary sensitivity weight  $W_t(s)$  and the system uncertainty of the hip joint control.

Following Skogestad's method [25], after parameters substitution, the sensitivity weights ( $W_s(s)$ ) for the hip and knee motors are

$$W_s(s) = \frac{0.5s + 1}{s + 0.001}. \quad (36)$$

Singular value plot of inverse of the sensitivity weight is shown in Figure 15.

The plot of the complementary sensitivity weight and the system uncertainty singular value is shown in Figure 16. The complementary sensitivity weight  $W_t(s)$  of the hip control system is obtained as

$$W_t(s) = \frac{5.006s^2 + 24.14s + 3.02}{s^2 + 102.4s + 250.5}. \quad (37)$$

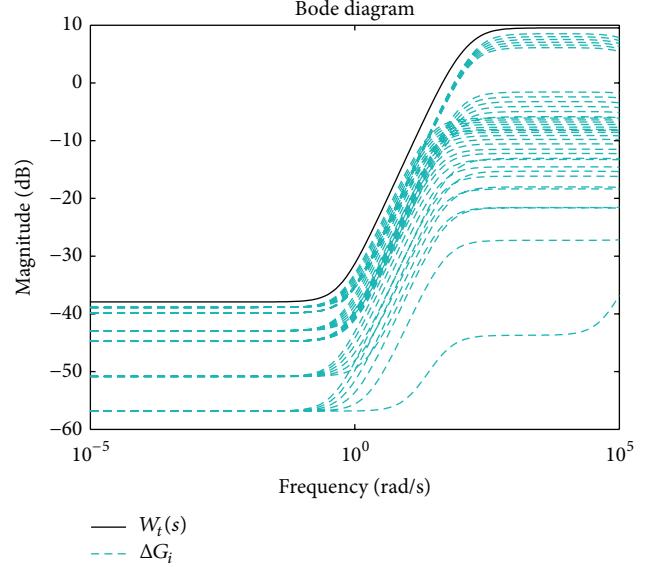
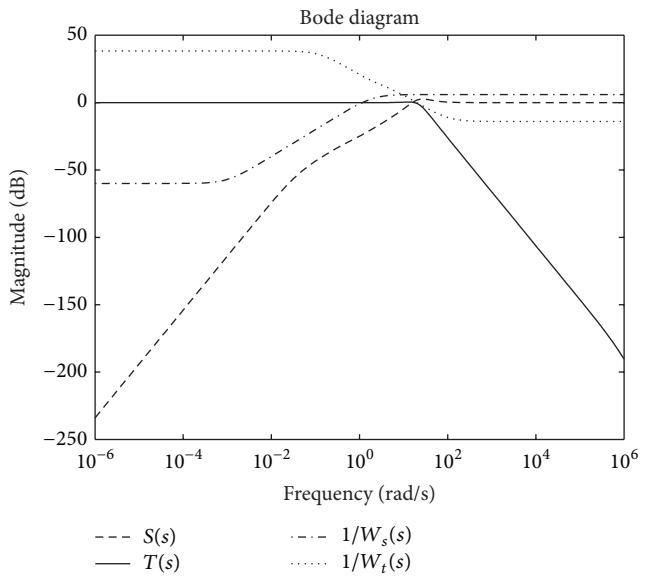
FIGURE 17: Complementary sensitivity weight  $W_t(s)$  and the system uncertainty of the knee joint control.

FIGURE 18: The sensitivity, complementary sensitivity, and their inverse of the weight singular values of the hip joint.

Meanwhile complementary sensitivity weight  $W_t(s)$  of the knee control system is obtained as

$$W_t(s) = \frac{2.996s^2 + 363.3s + 190.8}{s^2 + 259.9s + 15040}. \quad (38)$$

The plot of the complementary sensitivity weight and the system uncertainty singular value is shown in Figure 17.

Figures 18 and 19 show that the singular values of the inverse of the weight functions on the hip and the knee joints are larger than the sensitivity and the complementary sensitivity singular values which confirm  $|W_s| < 1$  and  $|W_t| < 1$ .

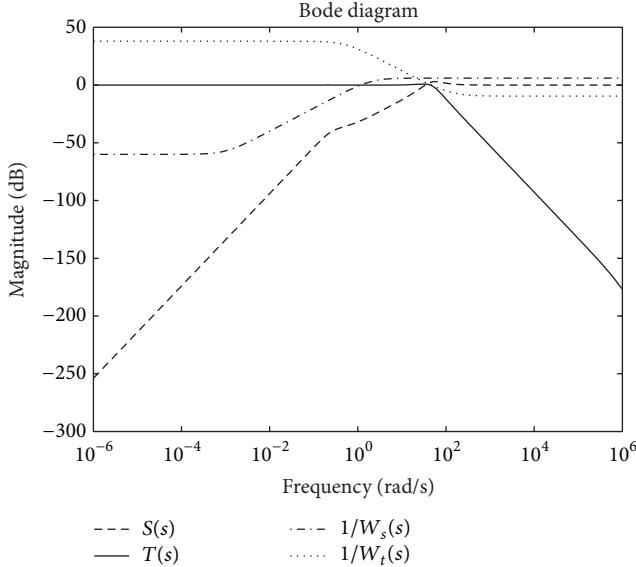


FIGURE 19: The sensitivity, complementary sensitivity, and their inverse of the weight singular values of the knee joint.

In this research, a PID controller structure is selected as the structure of the proposed controller:

$$K(s) = K_p + \frac{K_i}{s} + K_d \cdot s, \quad (39)$$

where  $K_p$  is proportional constant,  $K_i$  is integral constant, and  $K_d$  is derivative constant.

**4.3. Differential Evolution.** Differential Evolution (DE) is an evolutionary based optimization algorithm [26]. The optimal controller gains for the structure specified  $H_\infty$  controller are obtained by using DE [27]. Parameters of DE are set as follows: number of population = 20, differential weight = 0.8, and crossover probability = 0.7. Figures 20 and 21 show the evolution of the hip and knee controller parameters during the optimization of  $J_{\infty,a}$  and  $J_{\infty,b}$  by using DE.

Derived from the result, equation of controller for the hip motor is expressed by

$$K(s) = 0.3424 + \frac{0.0246}{s} + 0.0133s. \quad (40)$$

Derived from the result, equation of controller for the knee motor is expressed by

$$K(s) = 0.9986 + \frac{0.029}{s} + 0.0309s. \quad (41)$$

## 5. Simulation and Experimental Results

Figure 22 shows the simulation result of the hip joint controlled by the proposed controller with a step input under uncertainty. The controller provided efficient control of the system. Settling times of the system response under uncertainty were still admissible even though almost half of the responses have overshoot. The maximum overshoots are

around 10%. The magnitude of the overshoots and the settling times are small compared to the gait cycle period.

As previously explained, without uncertainty, even a PD controller can be optimally set to obtain better response than any robust controllers. Figure 23 shows the response of the hip nominal model without uncertainty controlled by a well-tuned PID controller. It shows that the system is able to follow the step input much better than the robust controller. However, Figure 24 shows various responses of the system with a well-tuned PID controller under uncertainty. Some graphs show better responses than the proposed controller. However, in some conditions, the system has significant steady-state error. For the worst condition, the well-tuned PID controller cannot handle the uncertainty.

Figure 25 shows the simulation result of the proposed controller in controlling knee joint. The result shows better performance in terms of settling time and overshoot compared to the hip joint because of lower value of the shank load compared to the hip load. However, similar to the hip system, Figure 26 shows that the well-tuned PID controller obtains better performance than the proposed controller on the nominal model. However, Figure 27 shows that, under the same range of uncertainty, the proposed controller performs better than the well-tuned PID. Similar to the hip joint, it has steady-state error under some model uncertainty.

To verify the simulation result, we conducted three experiments:

- (1) In the first experiment, individual leg was tested using both the proposed controller and well-tuned PID controller following a step trajectory input. To observe robustness to model uncertainty, inertia of the leg was varied.
- (2) In the second experiment, hip and knee joint angles were measured while trotting forward to see the performance of the proposed controller tracking the gait trajectory pattern. To observe robustness to the environment uncertainty, walking surface for the quadruped was varied.
- (3) To see the contribution of the gait improvement control to the overall robot stability, the third experiment was conducted by measuring the trunk behavior when the quadruped trotted using the proposed controller combined with the balancing algorithm. The measurements were done using Inertial Measurement Unit (IMU) and visual inspection using camera recorder. The quadruped robot walked in three different surfaces during the test.

Inertia parameter as one of the system model parameters was varied into three values: nominal, minimum, and maximum value. For the maximum value, the tip of the shank or the shoe was added with a point mass so the inertia will increase around 30% similar to the value on the simulation test. The additional inertia on the tip is considered as a point mass inertia with amount of  $mr^2$ . The minimum inertia cannot be set similar to the simulation which is reduction of 30% inertia. Minimum inertia was set by replacing the metal shoe with light nonmetal shoe to get the mass reduction.

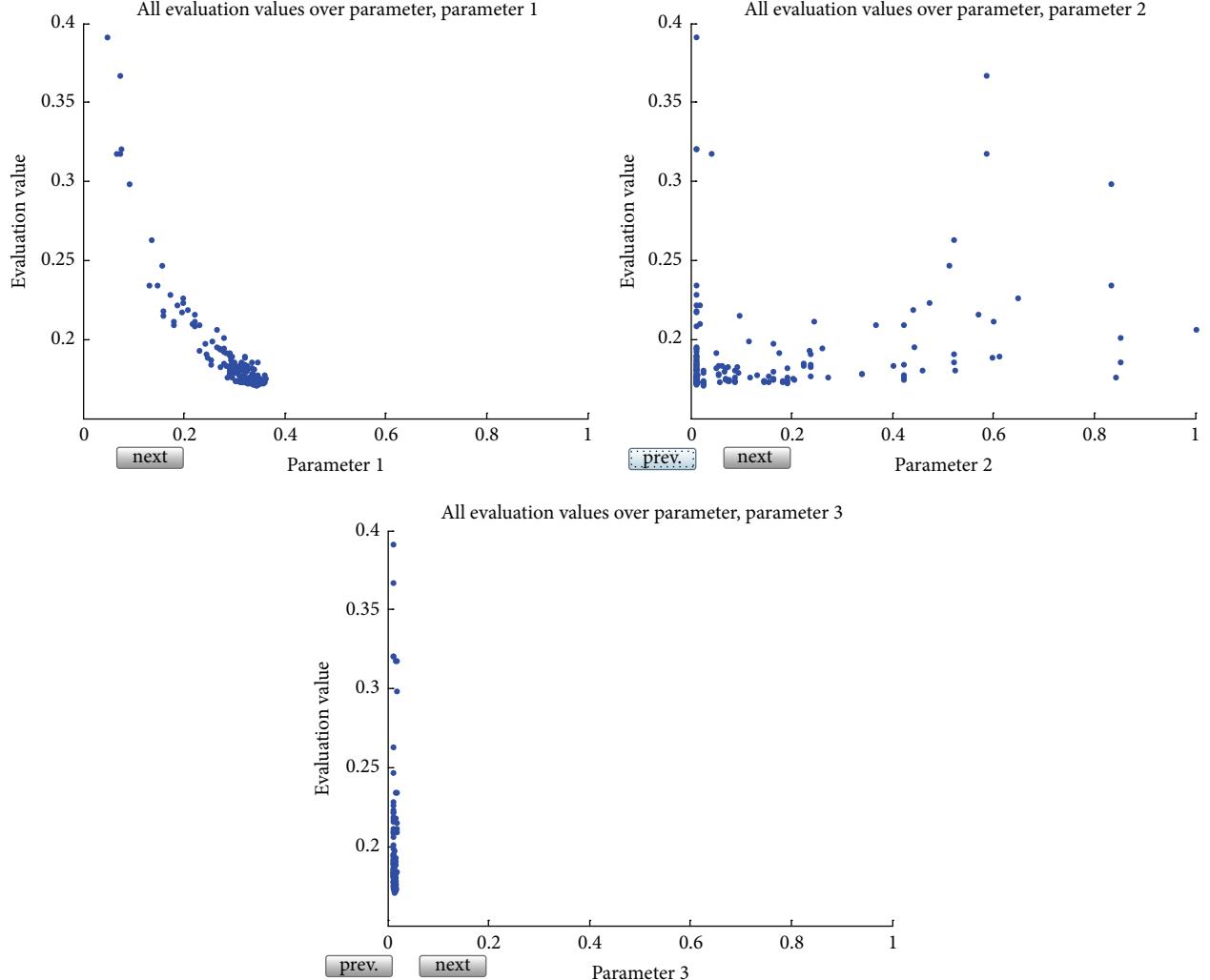


FIGURE 20: Evolution of parameters during optimization.

By assuming inertia reduction of a point mass on the tip of the shank, 10% reduction was set on the leg. Nominal inertia means that there is no addition or reduction of mass to the leg. To limit the consideration just to the mass addition and reduction uncertainty only, the individual leg tests were conducted on the leg maximum stretch position. This fixed orientation was chosen because, during the trot, different configuration of thigh and shank along the trajectory will generate inertia variation to the hip.

Figures 28 and 29 show the comparison of the proposed controller and well-tuned PID controller performance controlling hip and knee joint position, respectively. Reference signal was set as a step function with the amount of 30 degrees' angle. The experiment was conducted under three different values of leg inertia. Both Figures 28 and 29 consist of three subparts. Part (a), part (b), and part (c) represent the experiment test on the nominal leg inertia value, minimum leg inertia value, and maximum leg inertia value, respectively.

Figure 28 shows system responses of hip joint control using the proposed controller and well-tuned PID controller.

Figure 28(a) shows that, on the nominal model, well-tuned PID controller provides a good response as it has no overshoot, short settling time, and no steady-state error. It even has better response than the proposed controller. In this case, the well-tuned PID controller works better because the controller constants have been properly tuned with the nominal model (without uncertainties). Figure 28(b) shows that the well-tuned PID controller has higher overshoot but better settling time than the proposed controller. It might be because the 10% of mass reducing did not significantly disturb the system as it happened on the simulation where there was 30% of mass reducing. However, Figure 28(c) shows that addition of 30% of inertia made the well-tuned PID have lower settling time than the proposed controller. We can see from the three graphs that there are no big variances, for example, on the rise time and overshoot in the proposed controller output response. The settling time became longer when the leg inertia was in the maximum value. However, there were no steady state errors in all of the three inertia variation experiments.

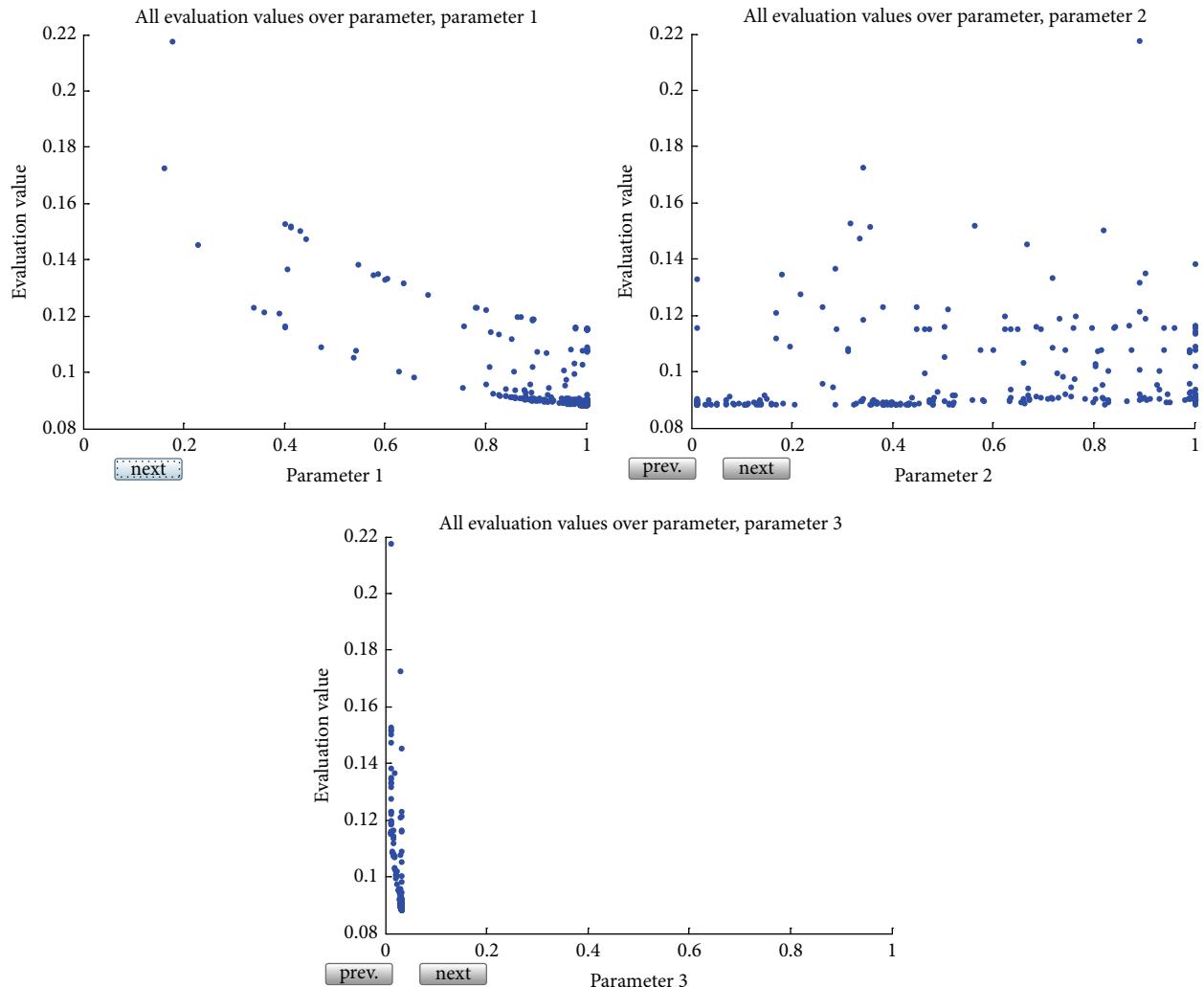


FIGURE 21: Evolution of parameters during optimization.

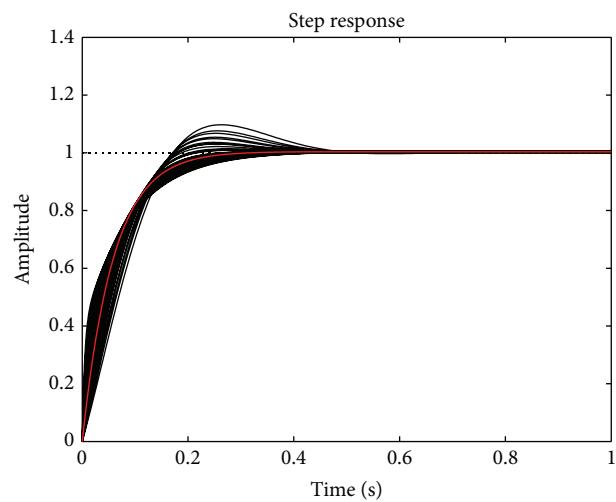


FIGURE 22: Simulation result of the controller on the hip.

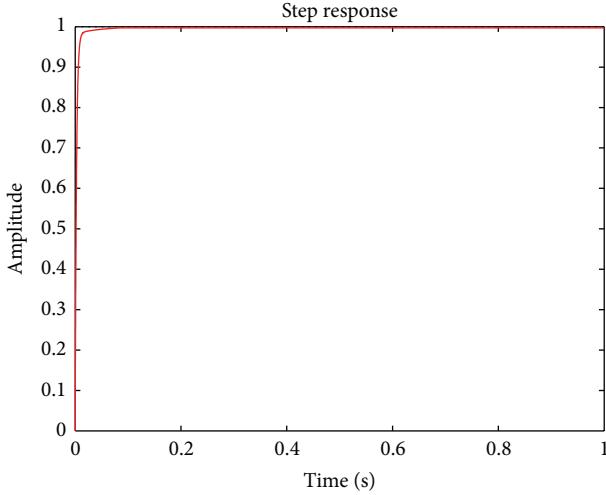


FIGURE 23: Simulation result of a well-tuned PID controller on hip nominal model.

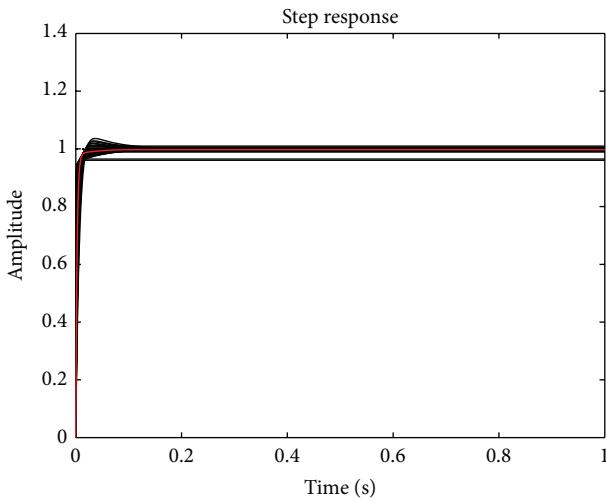


FIGURE 24: Simulation result of a well-tuned PID controller on hip model under uncertainty.

Figure 29 shows system responses of knee joint control using the proposed controller and well-tuned PID controller. Figure 29(a) shows that, on the nominal model, well-tuned PID controller and the proposed algorithm both perform well and only the well-tuned PID slightly has smaller settling time. Figure 29(b) shows that in the second case, where the system has the minimum value of inertia, well-tuned PID and the proposed controller have almost the same settling time. However, the proposed controller has smaller overshoot. Figure 29(c) shows that the knee joint control system of the well-tuned PID has steady-state error when the inertia is in the maximum value. The proposed controller gives a satisfactory result where there is no steady-state error, short rise time without overshoot, and short settling time.

Overall experiment shows that the proposed controller has less response fluctuation compared to the well-tuned PID

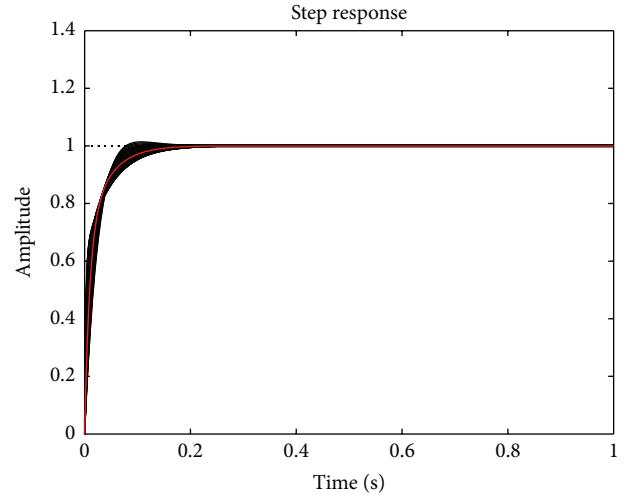


FIGURE 25: Simulation result of the proposed controller on the knee joint under uncertainty.

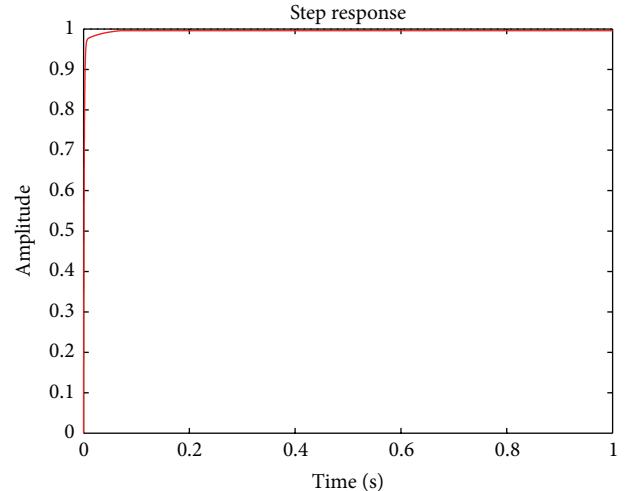


FIGURE 26: Simulation result of a well-tuned PID controller on the nominal model of the knee joint.

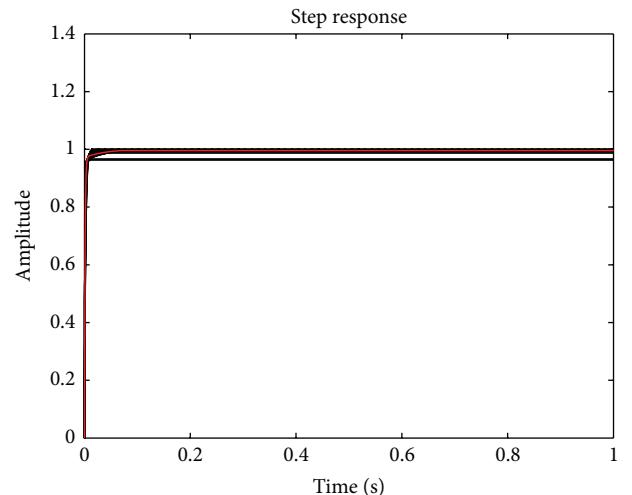


FIGURE 27: Simulation result of a well-tuned PID controller on the knee joint under uncertainty.

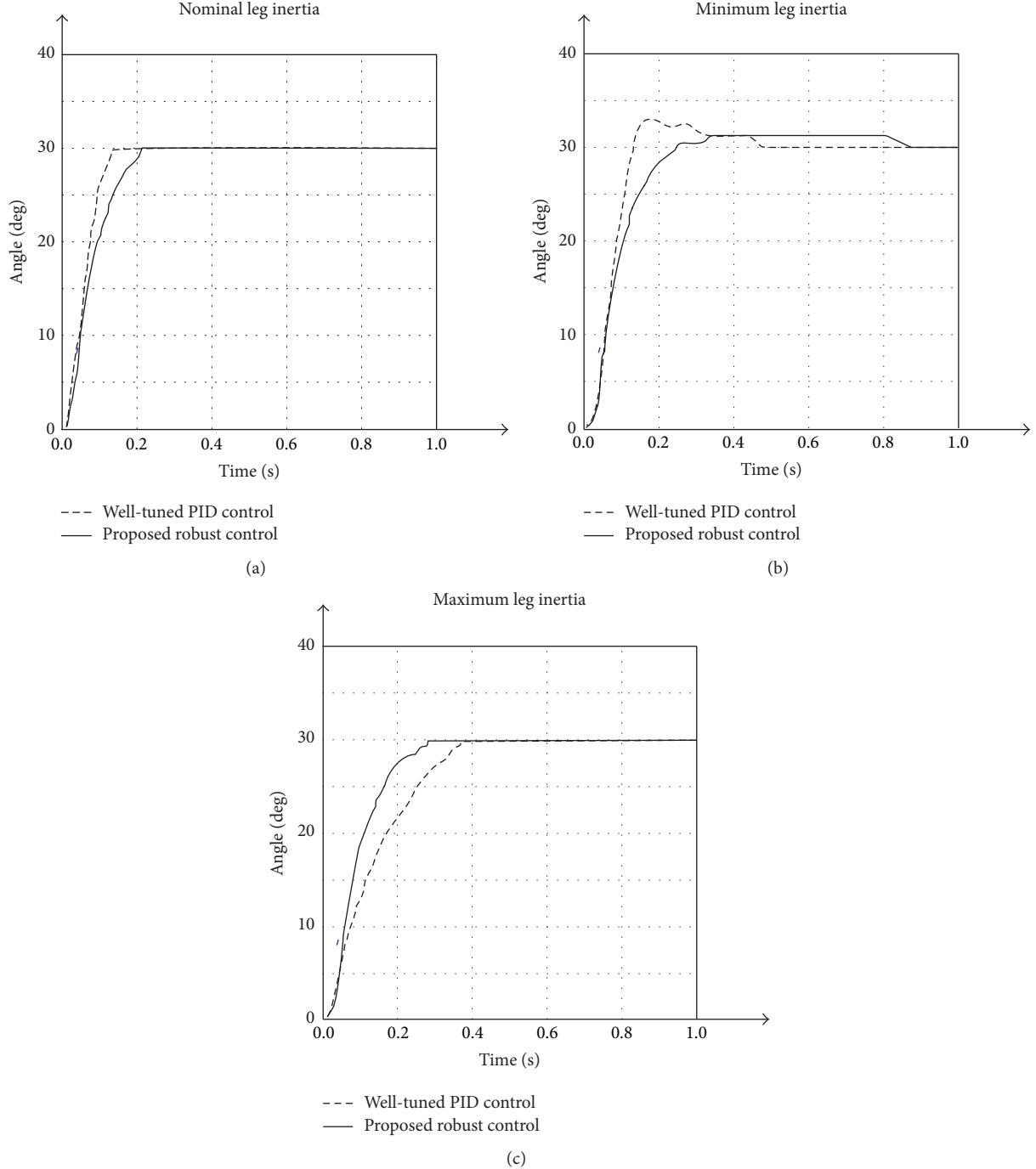


FIGURE 28: Comparison of hip joint position control performance using the proposed controller and well-tuned PID controller. (a) Comparison on nominal inertia value. (b) Comparison on minimum inertia value. (c) Comparison on maximum inertia value.

controller and always gives satisfactory result in all conditions of model parameter change.

Hip and knee joint angles as the inverse kinematic result of half-sinusoidal foot trajectory pattern were sent to the controller to achieve trotting gait of the quadruped. Trot gait parameters were set as follows: duty factor  $\beta$  set at 50% with stride length of 30 mm, CG height constant set at 220 mm, swing height 30 mm, and trot frequency at 2 Hz. Besides inertia uncertainty from the dynamics of the body, external

disturbance also arises from three different walking surfaces which are hard tiles floor, medium soft carpet, and grass. Hard tile floor provides hard landing impact on the legs. Grass floor provides more damping to the legs while trotting but more friction while swinging. Carpet floor is softer and has more friction compared to the tile floor.

Figure 30 shows performance of the proposed controller to control hip and knee joint angles in the presence of those uncertainties. Set points are set of angles generated from the

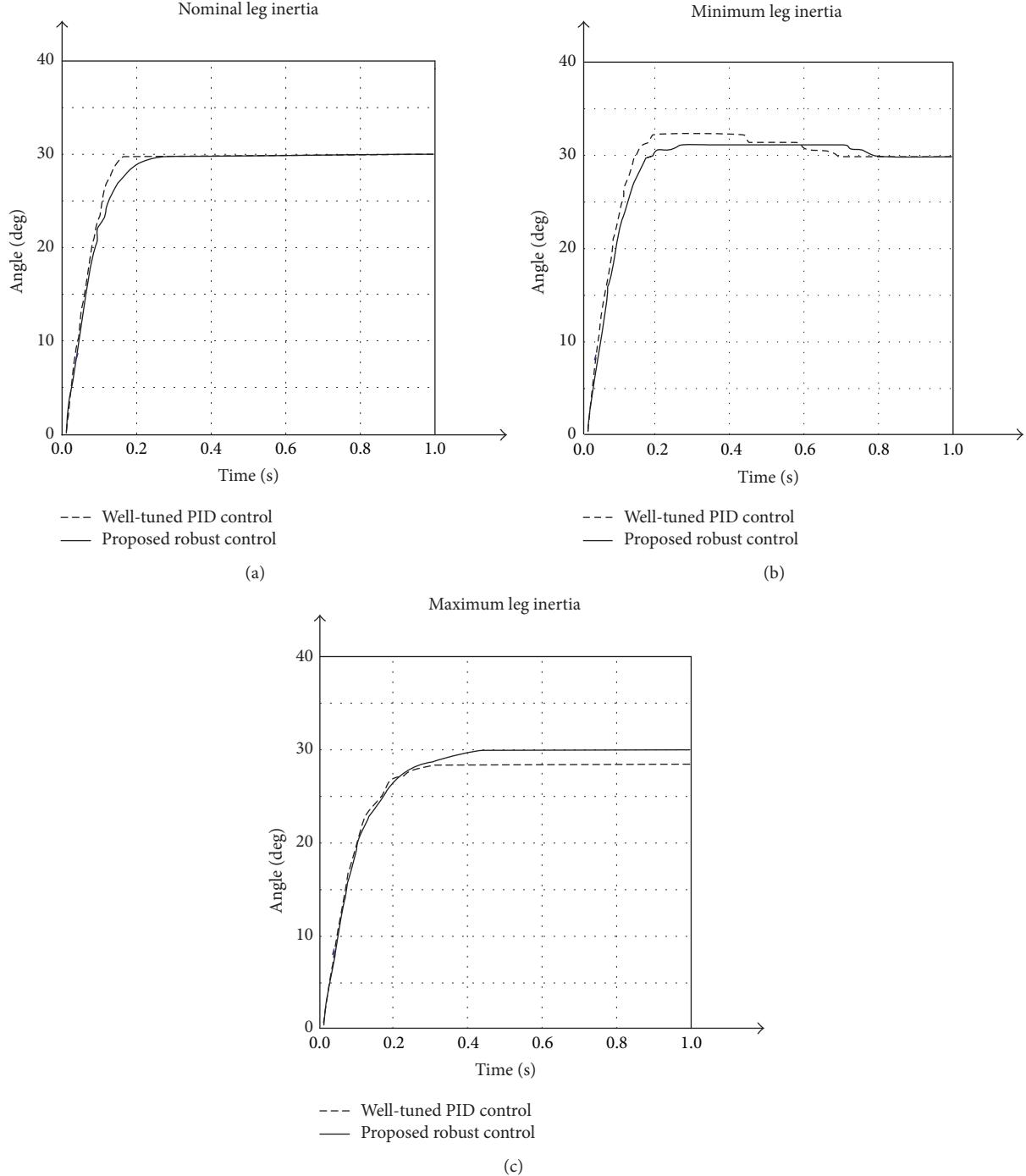


FIGURE 29: Comparison of knee joint position control performance using the proposed controller and well-tuned PID controller. (a) Comparison on nominal inertia value. (b) Comparison on minimum inertia value. (c) Comparison on maximum inertia value.

gait generator block that the joints should follow. Hip joint control has better performance in terms of response time in tracking the trajectory compared to knee joint control. It is because they have different set point pattern. Hip has fewer sudden changes of the angle command unlike on the knee set point. Yet, the knee joint settling time actually is still considerably small because the gait frequency is 2 Hz. It means that completing one single gait needs 0.5 seconds; then

the longest settling time is around 0.2 seconds. However, the knee joint control responses have better tracking result as the graphs stay close to each other along the trajectory. Overall, the output angle patterns are consistent along the repetition; it means that the gait pattern tracking is also consistent. The graphs also show that the three outputs from three different walking surface tests have almost the same trajectories. It indicates that the gait trajectory tracking is well maintained

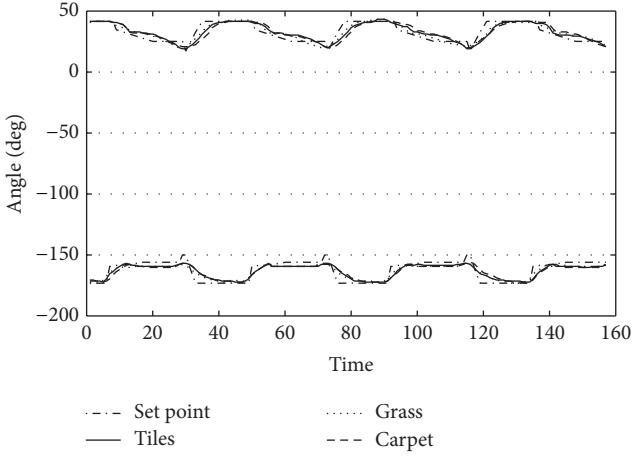


FIGURE 30: Gait trajectory tracking performance when trotting on three different surfaces: hard tiles, grass, and carpet.

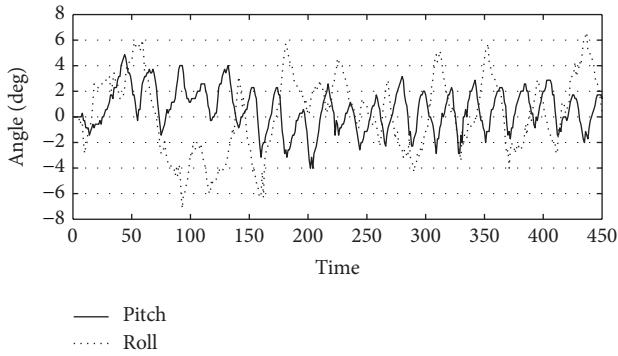


FIGURE 31: Trunk attitudes when quadruped robot trots on hard tile floor.

by the proposed controller even in the presence of previously mentioned uncertainties.

The last experiment is to test the gait tracking control combined with the balancing algorithm when the quadruped also trots in different types of terrain including hard tile floor, grass, and carpet floor. Figures 31–33 show the experimental results of trunk attitude when the quadruped robot walks on different types of terrain. The result shows that the trunk attitude is not affected much by the variation of the terrains. Figure 31 shows that the fluctuation of the pitch and roll angle is around  $\pm 5$  and  $\pm 6.5$  degrees, respectively. But, on the grass, the robot has smaller fluctuation both of pitch and of roll angle of the trunk. Figure 32 shows around  $\pm 4$  and  $\pm 5$  degrees' angle fluctuation of pitch and roll angle, respectively, when the robot trots on the grass. Figure 33 shows different case when the quadruped trots on carpet surface. It has  $\pm 4.5$  degrees of pitch angle fluctuation and around  $\pm 5.5$  degrees of roll angle fluctuation. When the surface is hard such as tiles, the leg landing impact force is propagated through the entire structure that caused bigger disturbance to the robot. Grass damped the vibration from the leg landing impact and stance legs transition during the trot. In between those characteristics is the carpet, which is soft but the floor underneath is also

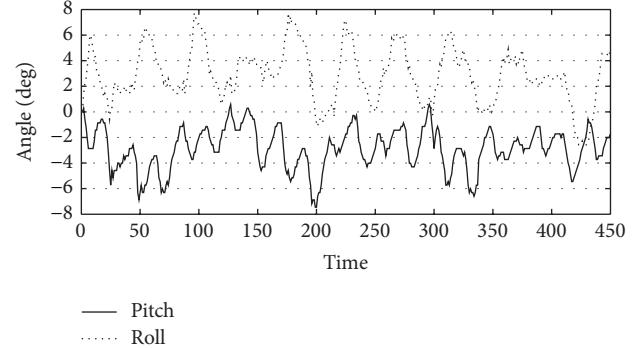


FIGURE 32: Trunk attitudes when quadruped robot trots on grass.

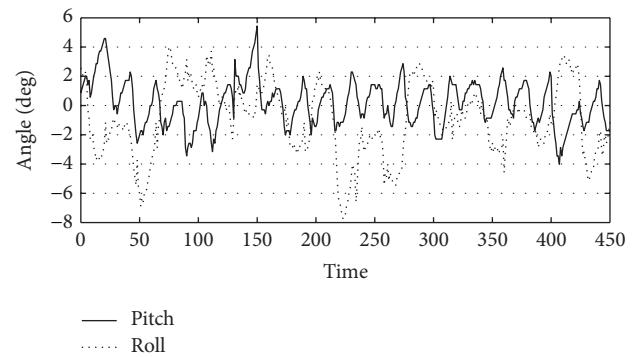


FIGURE 33: Trunk attitudes when quadruped robot trots on carpet floor.

hard. Considering those surface characteristics and resulting behavior, the quadruped is considered to be able to walk stably using the proposed robust gait controller combined with the balancing algorithm.

Figure 34 shows snapshot images of the trunk attitude when the robot was trotting on the hard tile floor. A cross mark was placed on the trunk. The robot was recorded by a camera to get a fixed height of the reference cross mark as indicated by a solid line in the images. The dashed line shows the trunk attitude in vertical direction monitored from the cross mark position. Here we did not have magnitude of the tracking error. However, it was shown visually that there is no significant oscillation on the trunk during the trot.

## 6. Conclusion

The proposed structure specified mixed sensitivity  $H_{\infty}$  controller was able to track the gait trajectory of the quadruped robot with unmodeled dynamics, uncertainty, and disturbance. MATLAB simulation showed that the system controlled by the proposed controller could be tracked following the step input without steady-state error with satisfactory rise time and settling time. Most of the responses had less than 0.5 s settling time and less than 10% overshoot. The settling times and the overshoot periods were considerably short compared to the gait cycle period.

Individual leg test with inertia variation of 10% reduction (minimum inertia), nominal value, and 30% addition

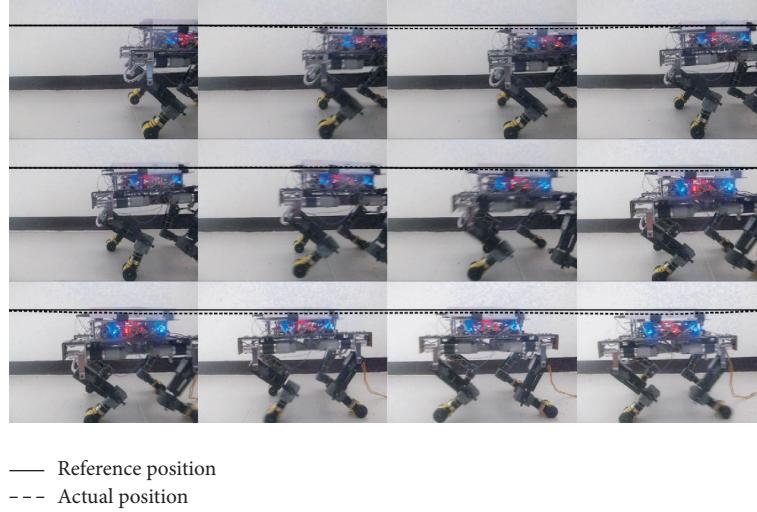


FIGURE 34: Snapshot images of the trunk attitude.

(maximum inertia) showed that, on the nominal cases, both controllers provided satisfactory performances. When the inertia was set to minimum, the well-tuned PID controller had longer settling time compared to the proposed controller, while, in the maximum inertia experiment, the well-tuned PID had 5% of steady-state error while the proposed controller performed satisfactorily with small rise time but no overshoot and short settling time without steady-state error.

Furthermore, the proposed controller also worked well to control the legs when the quadruped robot trotted on different terrains, such as hard tile floor, grass, and carpet. The proposed controller was able to track the gait trajectory over the three types of those walking surfaces. The trunk attitude was maintained in small range of variation. With roll angle variation in average around  $\pm 5$  degrees, the quadruped robot still could perform a good shape of trotting and was far from the critical angle that made the robot tip over. The variation of the quadruped robot's trunk pitch angle was less than that in the trunk roll angle.

It was proved that the structure specified  $H_\infty$  controller is a good controller especially for industrial usage because it simplifies the high order controller into familiar controller structure, such as PID, but still provides its robust characteristic.

## Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

The authors would like to acknowledge the support from Ministry of Research, Technology, and Higher Education of the Republic of Indonesia (RISTEKDIKTI).

## References

- [1] M. H. Raibert, *Legged Robots That Balanced*, MIT Press, Cambridge, Mass, USA, 1986.
- [2] P. G. Santos, E. Garcia, and J. Esteveira, *Quadrupedal Locomotion*, Springer, New York, NY, USA, 2006.
- [3] A. G. Gonzalez-Rodriguez, A. Gonzalez-Rodriguez, and F. C. Castillo-Garcia, "Improving the energy efficiency and speed of walking robots," *Mechatronics*, vol. 24, no. 5, pp. 476–488, 2014.
- [4] E. Muybridge, *Animal in Motion*, Chapman & Hall, London, UK, 1899.
- [5] M. Hildebrand, "The quadrupedal gaits of vertebrates," *Bio-Science*, vol. 39, no. 11, pp. 766–775, 1989.
- [6] H. M. Raibert, K. Blankenspoor, G. Nelson, and R. Playter, "BigDog, the rough-terrain quadruped robot," in *Proceedings of the in 17th IFAC World Congress*, Seoul, Republic of Korea, July 2008.
- [7] L. R. Palmer III and D. E. Orin, "3D control of a high-speed quadruped trot," *Industrial Robot*, vol. 33, no. 4, pp. 298–302, 2006.
- [8] D. Golubovic and H. Hu, "Evolving locomotion gaits for quadruped walking robots," *Industrial Robot*, vol. 32, no. 3, pp. 259–267, 2005.
- [9] H.-Y. Chung, C.-C. Hou, and S.-Y. Hsu, "Hexapod moving in complex terrains via a new adaptive CPG gait design," *Industrial Robot*, vol. 42, no. 2, pp. 129–141, 2015.
- [10] B. Ugurlu, I. Havoutis, C. Semini, and D. G. Caldwell, "Dynamic trot-walking with the hydraulic quadruped robot—HyQ: analytical trajectory generation and active compliance control," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '13)*, pp. 6044–6051, IEEE, Tokyo, Japan, November 2013.
- [11] K. Yoneda and S. Hirose, "Dynamic and static fusion gait of a quadruped walking vehicle on a winding path," *Advanced Robotics*, vol. 9, no. 2, pp. 125–136, 1995.
- [12] M. H. Yamchi, M. Shafiee, and H. A. Talebi, "Modeling and robust iterative learning control of a quadruped robot," in

- Proceedings of the IFAC World Congress*, pp. 9373–9378, Milan, Italy, 2011.
- [13] X. Rong, R. Song, and B. Li, “Simulation for sagittal plane motions of a quadruped robot using MATLAB and Simulink,” *Journal of Information and Computational Science*, vol. 9, no. 8, pp. 2165–2173, 2012.
  - [14] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, Addison-Wesley, Boston, Mass, USA, 1986.
  - [15] P. Sutayasadi and M. Parnichkun, “Trotting control of a quadruped robot using PID-ILC,” in *Proceedings of the in 41st Annual Conference of IEEE (IECON '15)*, pp. 4400–4005, IEEE, Yokohama, Japan, 2015.
  - [16] S. Kajita and K. Tani, “Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1405–1411, Sacramento, Calif, USA, April 1991.
  - [17] S. Kajita and K. Tani, “Experimental study of biped dynamic walking in the linear inverted pendulum mode,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2885–2891, IEEE, Nagoya, Japan, May 1995.
  - [18] J. H. Park and K. D. Kim, “Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control,” in *Proceedings of the in IEEE International Conference on Robotics and Automation (ICRA '98)*, Lueven, Belgium, May 1998.
  - [19] S. Hirose, K. Yoneda, R. Furuya, and T. Takagi, “Dynamic and static fusion control of quadruped walking vehicle,” in *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS '89)*, pp. 199–204, Tsukuba, Japan, 1989.
  - [20] T. Akbas, S. E. Eskimez, S. Ozel, O. K. Adak, K. C. Fidan, and K. Erbatur, “Zero Moment Point based pace reference generation for quadruped robots via preview control,” in *Proceedings of the 12th IEEE International Workshop on Advanced Motion Control (AMC '12)*, pp. 1–7, IEEE, Sarajevo, Bosnia and Herzegovina, March 2012.
  - [21] E. G. Gilbert and I. J. Ha, “An approach to non linear feedback control with application to robotics,” *IEEE Transactions on Automatic Control*, vol. 14, no. 6, pp. 879–884, 1984.
  - [22] L. R. Hunt, R. J. Sun, and G. Meyer, “Global transformations of nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 28, no. 1, pp. 24–31, 1983.
  - [23] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Robot Manipulator Control: Theory and Practice*, Marcel Dekker, New York, NY, USA, 2004.
  - [24] A. A. G. Siqueira, M. H. Terra, and M. Bergerman, *Robust Control of Robots*, Springer, New York, NY, USA, 2011.
  - [25] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control*, John Wiley & Sons, New York, NY, USA, 2001.
  - [26] D. Ruijun, “Differential evolution versus particle swarm optimization for PID controller design,” in *Proceedings of the 5th International Conference on Natural Computation*, pp. 236–240, 2009.
  - [27] B. Markus, “MATLAB Central File Exchange,” 2014, [http://www.mathworks.com/matlabcentral/fileexchange/18593-differential-evolution/all\\_files](http://www.mathworks.com/matlabcentral/fileexchange/18593-differential-evolution/all_files).

