Robotica (2017) volume 35, pp. 1670–1689. © Cambridge University Press 2016
doi:10.1017/S0263574716000394

Push recovery of a quadruped robot on challenging terrains

Mahdi Khorram* and S. Ali A. Moosavian

Center of Excellence in Robotics and Control, Advanced Robotics and Automated Systems Lab, Department of Mechanical Eng, K. N. Toosi Univ of Technology, Tehran, Iran
E-mail: moosavian@kntu.ac.ir

(Accepted May 28, 2016. First published online: June 30, 2016)

SUMMARY

Legged robots may become unstable when subjected to unexpected disturbances such as external pushes and environmental irregularities mostly while moving on natural terrains. To enhance the mobility performance, legged robots should be able to keep or restore their balanced configuration when a sudden disturbance is exerted. The aim of this article is to design a controller for a quadruped robot to restore its balanced configuration despite exerting external pushes. This is achieved based on developing a full-dynamics model of the robot moving over even and uneven terrains. The proposed controller is based on a PD module which calculates the required accelerations for restoring the robot equilibrium. However, these accelerations may make the robot unstable and also cause the slippage of stance feet. Therefore, an optimization algorithm is used to compute the maximum admissible accelerations. The constraints of the optimization problem are the conditions which guarantee the robot stability and the stance feet slippage avoidance. The optimization algorithm is transformed into a linear constrained least-squares problem to be solved in real-time. The main contributions of this article are the development of a push recovery algorithm for quadruped robots and also the introduction of an appropriate condition which guarantees the stability of the robot even on uneven terrains. This stability condition is developed based on a full-dynamics model of the robot. The proposed algorithm is applied on an 18-DOF quadruped robot when the robot is standing over both even and uneven terrains. The obtained results show that the robot can successfully restore its balanced configuration by precise adjustment of the position and orientation of its main body while a massive external disturbance is exerted.

KEYWORDS: Quadruped robots, Push recovery, Dynamic stability, Whole-body dynamics, Uneven terrains.

1. Introduction

Quadruped robots should be able to maintain and recover their stability against external pushes when these robots move over natural irregular environment. These disturbances may be imposed from an external push or environmental irregularities. The external push should be recognized immediately, and proper reactions to maintain the system stability should be performed. Quadruped Robots usually change their posture or take one or several steps to regain their balanced configuration. In general, there are three different strategies confronting to an unexpected disturbance. The ankle or hip strategy in the field of humanoid robots, restores the stability of the robot by adjusting the position and orientation of its main body.[1−3] This strategy is mostly effective when the robot is pushed by a small disturbance force. When the magnitude of the external push increases so that the robot cannot handle the disturbing effects by just using the balance controller, stepping strategy should be used.[4,5] In the case of large pushes, fall control approach should be used to reduce the damage effects due to falling.[6,7] Most of previous researches have focused on how to solve the balance recovery problem for humanoid robots and only few works were devoted to quadruped robots.[8,9]

* Corresponding author. E-mail: mahdi.khorram@gmail.com

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

There are two different approaches to study the push recovery problem for legged robots. In the first approach, to overcome the computational complexities of the push recovery problem for a real robot which has a large number of DOFs, the reduced models such as LIPM (linear-inverted pendulum model) are considered.[10,11] These models provide considerable insights into the push recovery problem. The ZMP compensation method by using an inverted pendulum,[12] and the MPC-based push recovery algorithm by employing the LIPM,[13,14] have been proposed to study the push recovery problem. In order to increase the capabilities of the model, a two link inverted pendulum was also presented.[15] A combination of biped models for a quadruped robot was introduced to study the balance recovery problem for such robots.[8] The Angular Momentum Pendulum Model (AMPM) was used to design a balance controller to confront rotational perturbations.[16,17] The main drawback of all these models is that they cannot include complete effects of the motion of main body and legs on recovering the robot balance, whereas biological counterparts exploit all motion capabilities in recovering their balance. In the second approach, a full-dynamics model of the robot is used for the push recovery problem to fully exploit the motion capabilities of the robot.[18–20] In this work, such a model will be derived to design the balance recovery algorithm.

In a few researches, the balance recovery algorithm has been proposed based on the regulation of contact forces by using a constrained optimization problem.[21] The optimization of contact forces combined with the Model Predictive Control approach is another solution of the balance recovery problem.[22] The momentum-based balance controller was introduced based on the regulation of angular and linear momentums, since one way to keep the robot balance is to control the momentum of the robot. Kajita *et al.* developed a method to produce the whole-body motion for given linear and angular momentums.[23] Through the definition of a constrained optimization problem, admissible joint accelerations were computed to obtain the desired angular and linear momentums.[24] In an attempt to improve previous results, a balance controller with considering the stability and friction conditions has been proposed through the definition of a constrained optimization problem.[25,26] The maximum admissible joint accelerations have been computed through a constrained optimization problem to recover the stability of a biped robot.[27] In this approach, the stability and friction conditions are formulated as few equality and inequality constraints of the optimization problem. Henceforth, in this article, the push recovery problem is solved by computing the main body admissible accelerations. Compared to previous studies, a stability condition will be proposed to assure the robot balance moving over even and uneven terrains. Besides, the stability condition will be expressed in terms of the optimization variables in appropriate form as linear constraints.

Few researches have addressed the push recovery problem for quadruped robots. The BigDog[9] and recently the Spot have shown excellent performance in response to external pushes, but there are mainly some video clips which report these performances rather than published articles. To study the push recovery for a quadruped robot, a reactive controller has been proposed.[28] A distributed control system, which is made of four independent leg controllers, was proposed to realize the stable dynamic walking of a quadruped robot and also is able to counteract to small disturbances.[29] A controller which uses an active compliance was introduced to achieve the disturbance rejection for a quadruped robot.[30]

The focus of this article is to design a balance controller for a quadruped robot under an unknown external push in the standing posture over even and uneven terrains. In order to compare this work to other recent researches, the work by Gehring *et al.* is chosen. They have investigated the push recovery for a quadruped robot based on the regulation and distribution of contact forces while the algorithm is applied to the robot only over even terrains.[31] However, in this article, the push recovery is performed by regulating the main body accelerations. Besides, using the proposed stability condition makes it possible to use the algorithm for recovering the robot balance in motion over uneven terrains. To this end, the explicit dynamics equations will be formulated by using a computationally efficient method. Then, free-constraint dynamics equations will be derived with a constraint elimination method. The balance controller will be defined based on a PD controller which computes the desired accelerations. Then, an optimization problem will be defined to compute the admissible accelerations. Next, the stability condition for tumbling prevention will be obtained and expressed as a linear function of the optimization variables. Besides, the friction and joint torques saturation limits will be defined and expressed as linear functions of the optimization variables. So, the resultant optimization problem for computing the admissible accelerations can be solved with a linear least-squares algorithm. To recover the robot balanced configuration, these accelerations are applied to the robot by using an

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

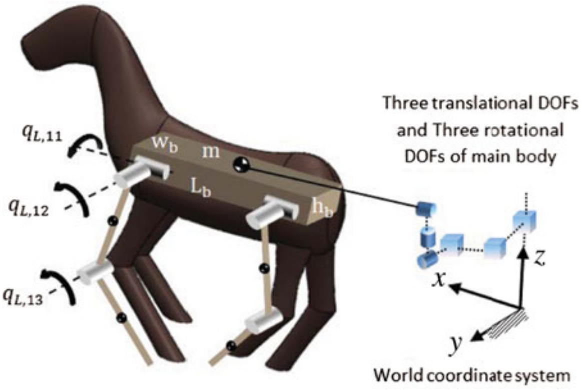

Fig. 1. The model of the quadruped robot.

inverse dynamics controller. Finally, the merits of the proposed algorithms will be confirmed through simulation studies on a quadruped robot standing over even and uneven terrains.

2. Whole-Body Dynamics Modelling

To obtain the robot dynamics equations, explicit dynamics derivation method is employed.[32] Considering the quadruped robot shown in Fig. 1, its configuration can be defined as

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_B^T & \mathbf{q}_L^T \end{bmatrix}^T, \tag{1}$$

where $\mathbf{q}_B \in \mathbf{R}^{6\times 1}$ represents the vector of main body positions and orientations. In addition, $\mathbf{q}_L \in \mathbf{R}^n$ denotes the vector of joint positions and n is the whole number of the joints of legs. The dynamics equations of the robot can be expressed as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \begin{bmatrix} \mathbf{O}_{1\times 6} & \tau^T \end{bmatrix}^T + \mathbf{J}^{\mathbf{T}}\mathbf{F}_{\mathbf{Leg}}, \tag{2}$$

where $\mathbf{M}(\mathbf{q}) \in \mathbf{R}^{(n+6)\times(n+6)}$ denotes the mass matrix, $\mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbf{R}^{(n+6)\times 1}$ is the vector of Coriolis and centrifugal forces, $\mathbf{G}(\mathbf{q}) \in \mathbf{R}^{(n+6)\times 1}$ represents the gravitational forces. In addition, $\tau \in \mathbf{R}^{n\times 1}$ defines the joint torques, $\mathbf{J}$ represents the Jacobian matrix associated with contact points and $\mathbf{F}_{\mathbf{Leg}}$ is the contact forces are applied on the tip of stance feet at contact points. The last term in above equation is added to the dynamics equations due to the contact of stance feet with environment. All terms of the dynamics equations can be calculated by the formulations presented in ref. [32].

2.1. Constraint elimination method

In order to compute the joint torques or the joint angles in inverse or forward dynamics problems, respectively, the contact forces should be exactly known. To tackle this problem, there are two different approaches: the direct measurement of the contact forces by means of force sensors and the elimination of the contact forces from the dynamics equations by using kinematic constraints. Since the output signals of the force sensors are very noisy, the measured contact forces are not accurate and thus the performance of the first method is poor. However, in the second method, the term associated with the contact forces are cancelled by using kinematic constraints.[33,34] In the following, the constraint elimination method will be introduced.

Since legged robots are in contact with environment, some constraints are imposed on the dynamics equations. Assuming that the stance legs remain stationary during motion due to existing sufficient friction between these legs and the ground, the linear velocities of the tip of these legs will be zero. However, these legs can rotate about all directions at the contact points. Therefore, the kinematic constraints can be written as

$$\mathbf{X}_{\mathbf{tip,i}} = \mathbf{C}_i \quad \text{or} \quad \mathbf{V}_{\mathbf{tip},i} = \mathbf{O}_{3\times 1} \quad \text{for} \quad \text{i} = 1,\ldots,4 \text{ when i}^{th} \text{ leg is in the stance phase}, \tag{3}$$

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

where $\mathbf{X_{tip,i}}$ and $\mathbf{V}_{\mathbf{tip},i}$ denote the vectors of position and velocity of the tip of ith stance leg, respectively. Also, $\mathbf{C}_i$ is a constant parameter which represents the position of the ith stance leg with respect to the world frame. The above equation can be stated in the Jacobian form as follows:

$$\mathbf{J}\dot{\mathbf{q}} = \mathbf{O}_{3p\times 1}, \tag{4}$$

where P denotes the number of the stance legs. The goal of the constraint elimination method is that to transform Eq. (2) into free-constraint equations by using kinematic constraints. In other words, the contact forces will be dropped from the dynamics equations. To this end, a new space called as independent joint space is defined. Another characteristic of this space is that the control of the variables of this space is sufficient for tracking the desired path as long as the stance legs remain stationary on the ground and also the robot motion are consistent with the constraints. Assuming that K constraints are applied on the robot due to the contact of the stance legs with the ground, the dimension of this space will be $n + 6$–k. As defined earlier, n is the number of the joints of all legs. Also, it is assumed that the degree of freedom of main body is six. The variables of this space are the position and orientation of main body as well as the joint positions of swing legs. For instance, in walking gait, which at least three legs are in contact with the ground, for a quadruped robot with twelve joints (i.e., each leg with three DOFs), the dimension of this space is nine. The independent joint space can be defined as follows:

$$\boldsymbol{\beta} = \begin{bmatrix} \mathbf{q_B^T} & \mathbf{q_{SL}^T} \end{bmatrix}^T, \tag{5}$$

where $\mathbf{q_{SL}}$ denotes the joint angles of swing legs. In the following, the dynamics equations will be rewritten in terms of $\boldsymbol{\beta}$ such that the term related to the contact forces is cancelled from the dynamics equations. The relationship between the whole-body configuration and the variables of the independent joint space in the velocity level can be written as follows:

$$\dot{\mathbf{q}} = \mathbf{S}\dot{\boldsymbol{\beta}}, \tag{6}$$

where $\mathbf{S}$ is a mapping matrix and can be obtained from the kinematic constraints as follows:

$$\mathbf{S} = \begin{bmatrix} \mathbf{I}_{6\times6} & \mathbf{O}_{6\times3} & & \cdots & & \mathbf{O}_{6\times3} \\ \mathbf{K}_{\mathbf{1},1} & \mathbf{K}_{\mathbf{2},1} & \mathbf{O}_{3\times3} & & \cdots & \mathbf{O}_{3\times3} \\ \mathbf{K}_{1,2} & \mathbf{O}_{3\times3} & \mathbf{K}_{\mathbf{2},2} & & & \mathbf{O}_{3\times3} \\ & & & \vdots & & \\ \mathbf{K}_{\mathbf{1},4} & \mathbf{O}_{3\times3} & & \cdots & \mathbf{O}_{3\times3} & \mathbf{K}_{\mathbf{2},4} \end{bmatrix}, \tag{7}$$

where $\mathbf{K_{1,i}}$ and $\mathbf{K}_{2,\mathbf{i}}$ are given by

$$\mathbf{K_{1,i}} = \begin{cases} -\mathbf{J_{L,i}^{-1}J_{b,i}} & \text{when i}^{th} \text{ leg is in the stance phase} \\ \mathbf{O}_{3\times6} & \text{when i}^{th} \text{ leg is in the swing phase} \end{cases}, \quad \mathbf{K_{2,i}} = \begin{cases} \mathbf{O}_{3\times3} & \text{when i}^{th} \text{ leg is in the stance phase} \\ \mathbf{I_{3\times3}} & \text{when i}^{th} \text{ leg is in the swing phase} \end{cases}, \tag{8}$$

where $\mathbf{J_{b,i}}$ and $\mathbf{J_{L,i}}$ are the Jacobian matrices of the main body and the ith stance leg, respectively. They can be computed as

$$\mathbf{J_{b,i}} = \frac{\partial \mathbf{X}_{L,i}}{\partial \mathbf{q}_b}, \mathbf{J}_{L,\mathbf{i}} = \frac{\partial \mathbf{X}_{L,i}}{\partial \mathbf{q}_{L,i}} \tag{9}$$

where $\mathbf{X}_{L,i}$ represents the position of the tip of the ith stance leg. Since legged robots are underactuated intrinsically due to its floating main body, joint angles are split into underactuated and actuated parts. This division is motivated by the fact that there are no control efforts on the

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

underactuated joints i.e., the position and orientation of main body. For this purpose, Eq. (6) can be rewritten as

$$\begin{bmatrix} \dot{\mathbf{q}}_{ua} \\ \dot{\mathbf{q}}_{a} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{ua} \\ \mathbf{S}_{a} \end{bmatrix} \dot{\boldsymbol{\beta}}, \tag{10}$$

where the subscript "a" stands for the actuated joints and the subscript "ua" represents the underactuated joints. Time differentiation of Eq. (10) yields

$$\begin{bmatrix} \ddot{\mathbf{q}}_{ua} \\ \ddot{\mathbf{q}}_{a} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{S}}_{ua} \\ \dot{\mathbf{S}}_{a} \end{bmatrix} \dot{\boldsymbol{\beta}} + \begin{bmatrix} \mathbf{S}_{ua} \\ \mathbf{S}_{a} \end{bmatrix} \ddot{\boldsymbol{\beta}}. \tag{11}$$

The dynamics equations can be decomposed into actuated and underactuated parts as follows:

$$\begin{bmatrix} \mathbf{M}_{1ua} & \mathbf{M}_{2ua} \\ \mathbf{M}_{1a} & \mathbf{M}_{2a} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_{ua} \\ \ddot{\mathbf{q}}_{a} \end{bmatrix} + \begin{bmatrix} \mathbf{V}_{ua} \\ \mathbf{V}_{a} \end{bmatrix} + \begin{bmatrix} \mathbf{G}_{ua} \\ \mathbf{G}_{a} \end{bmatrix} = \begin{bmatrix} \mathbf{O}_{6\times 1} \\ \tau_{\mathbf{a}} \end{bmatrix} + \begin{bmatrix} \mathbf{J}_{ua}^{T} \\ \mathbf{J}_{a}^{T} \end{bmatrix} \mathbf{F}_{\mathbf{Leg}}. \tag{12}$$

Above equation can be rewritten into two independent equations as follows:

$$\begin{aligned} &\mathbf{M}_{1ua}\ddot{\mathbf{q}}_{ua} + \mathbf{M}_{2ua}\ddot{\mathbf{q}}_{a} + \mathbf{V}_{ua} + \mathbf{G}_{ua} = \mathbf{J}_{ua}^{T}\mathbf{F}_{\mathbf{Leg}} \qquad \text{(a)} \\ &\mathbf{M}_{1a}\ddot{\mathbf{q}}_{ua} + \mathbf{M}_{2a}\ddot{\mathbf{q}}_{a} + \mathbf{V}_{a} + \mathbf{G}_{a} = \tau + \mathbf{J}_{a}^{T}\mathbf{F}_{\mathbf{Leg}} \qquad \text{(b).} \end{aligned} \tag{13}$$

By left-multiplying Eq. (13)-a by the term of $\mathbf{S}_{ua}^{T}$ and Eq. (13)-b by the term of $\mathbf{S}_{a}^{T}$ and finally the sum of resultant equations, we have

$$\begin{aligned} &\mathbf{M}_{ua}\ddot{\mathbf{q}}_{ua} + \mathbf{M}_{a}\ddot{\mathbf{q}}_{a} + \mathbf{S}_{ua}^{T}\mathbf{V}_{ua} + \mathbf{S}_{a}^{T}\mathbf{V}_{a} + \mathbf{S}_{ua}^{T}\mathbf{G}_{ua} + \mathbf{S}_{a}^{T}\mathbf{G}_{a} = \mathbf{S}_{a}^{T}\tau + \left(\mathbf{J}_{a}\mathbf{S}_{a} + \mathbf{J}_{ua}\mathbf{S}_{ua}\right)^{T}\mathbf{F}_{\mathbf{Leg}} \\ &\mathbf{M}_{ua} = \mathbf{S}_{ua}^{T}\mathbf{M}_{1ua} + \mathbf{S}_{a}^{T}\mathbf{M}_{1a}; \mathbf{M}_{a} = \mathbf{S}_{ua}^{T}\mathbf{M}_{2ua} + \mathbf{S}_{a}^{T}\mathbf{M}_{2a}. \end{aligned} \tag{14}$$

From the definition of $\mathbf{S}_{a}$ and $\mathbf{S}_{ua}$, we can easily prove that

$$\mathbf{J}_{a}\mathbf{S}_{a} + \mathbf{J}_{ua}\mathbf{S}_{ua} = \mathbf{O}. \tag{15}$$

In the following, this claim will be proved in standing posture which all legs are in contact with the ground. Similar procedure can be used to prove this claim for other gaits such as walking and trotting. When all legs are in contact with the ground, the Jacobian matrix, **J**, can be defined as follows:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{\mathrm{B},1} & \mathbf{J}_{\mathrm{L},1} & \cdots & \mathbf{O}_{3\times 9} \\ \mathbf{J}_{\mathrm{B},2} & \mathbf{O}_{3\times 3} & \mathbf{J}_{\mathrm{L},2} & \mathbf{O}_{3\times 6} \\ \mathbf{J}_{\mathrm{B},3} & \mathbf{O}_{3\times 6} & \mathbf{J}_{\mathrm{L},3} & \mathbf{O}_{3\times 3} \\ \mathbf{J}_{\mathrm{B},4} & \mathbf{O}_{3\times 9} & \cdots & \mathbf{J}_{\mathrm{L},4} \end{bmatrix}. \tag{16}$$

Based on Eqs. (10) and (16), $\mathbf{J}_{a}$ and $\mathbf{J}_{ua}$ are defined as

$$\mathbf{J}_{ua} = \begin{bmatrix} \mathbf{J}_{\mathrm{B},1} \\ \mathbf{J}_{\mathrm{B},2} \\ \mathbf{J}_{\mathrm{B},3} \\ \mathbf{J}_{\mathrm{B},4} \end{bmatrix}, \mathbf{J}_{a} = \begin{bmatrix} \mathbf{J}_{\mathrm{L},1} & \mathbf{O}_{3\times 9} & \\ \mathbf{O}_{3\times 3} & \mathbf{J}_{\mathrm{L},2} & \mathbf{O}_{3\times 6} \\ \mathbf{O}_{3\times 6} & \mathbf{J}_{\mathrm{L},3} & \mathbf{O}_{3\times 3} \\ \mathbf{O}_{3\times 9} & \mathbf{J}_{\mathrm{L},4} & \end{bmatrix}. \tag{17}$$

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

On the other hand, based on the definition of the matrix of **S** i.e., Eqs. (7) and (8), $\mathbf{S}_a$ and $\mathbf{S}_{ua}$ can be defined as follows:

$$\mathbf{S}_{ua} = \mathbf{I}_{6\times 6},\, \mathbf{S}_a = \begin{bmatrix} -\mathbf{J}_{L,1}^{-1}\mathbf{J}_{B,1} \\ -\mathbf{J}_{L,2}^{-1}\mathbf{J}_{B,2} \\ -\mathbf{J}_{L,3}^{-1}\mathbf{J}_{B,3} \\ -\mathbf{J}_{L,4}^{-1}\mathbf{J}_{B,4} \end{bmatrix}. \tag{18}$$

Now, by substituting of Eqs. (17) and (18) in Eq. (15), the proof will be completed

$$\mathbf{J}_a\mathbf{S}_a + \mathbf{J}_{ua}\mathbf{S}_{ua} = \begin{bmatrix} \mathbf{J}_{\mathrm{B},1} \\ \mathbf{J}_{\mathrm{B},2} \\ \mathbf{J}_{\mathrm{B},3} \\ \mathbf{J}_{\mathrm{B},4} \end{bmatrix} \times \mathbf{I}_{6\times 6} + \begin{bmatrix} \mathbf{J}_{\mathrm{L},1} & \mathbf{O}_{3\times 9} & \\ \mathbf{O}_{3\times 3} & \mathbf{J}_{\mathrm{L},2} & \mathbf{O}_{3\times 6} \\ \mathbf{O}_{3\times 6} & \mathbf{J}_{\mathrm{L},3} & \mathbf{O}_{3\times 3} \\ & \mathbf{O}_{3\times 9} & \mathbf{J}_{\mathrm{L},4} \end{bmatrix} \times \begin{bmatrix} -\mathbf{J}_{L,1}^{-1}\mathbf{J}_{B,1} \\ -\mathbf{J}_{L,2}^{-1}\mathbf{J}_{B,2} \\ -\mathbf{J}_{L,3}^{-1}\mathbf{J}_{B,3} \\ -\mathbf{J}_{L,4}^{-1}\mathbf{J}_{B,4} \end{bmatrix} = \mathbf{O}. \tag{19}$$

By using Eq. (15), the contact forces can be cancelled from Eq. (14) as follows:

$$\mathbf{M}_{ua}\ddot{\mathbf{q}}_{ua} + \mathbf{M}_a\ddot{\mathbf{q}}_a + \mathbf{S}_{ua}^T\mathbf{V}_{ua} + \mathbf{S}_a^T\mathbf{V}_a + \mathbf{S}_{ua}^T\mathbf{G}_{ua} + \mathbf{S}_a^T\mathbf{G}_a = \mathbf{S}_a^T\tau. \tag{20}$$

To complete the procedure of the derivation of the free-constraint dynamic equations, above equation should be expressed in terms of the variables of the independent joint space. By substituting Eq. (11) in Eq. (20) we can get

$$\mathbf{M}_\beta\ddot{\boldsymbol{\beta}} + \mathbf{V}_\beta + \mathbf{G}_\beta = \mathbf{S}_a^T\tau, \tag{21}$$

where

$$\begin{aligned} &\mathbf{M}_\beta = \mathbf{M}_{ua}\mathbf{S}_{ua} + \mathbf{M}_a\mathbf{S}_a \\ &\mathbf{V}_\beta = \mathbf{S}_{ua}^T\mathbf{V}_{ua} + \mathbf{S}_a^T\mathbf{V}_a + \mathbf{M}_{ua}\dot{\mathbf{S}}_{ua}\dot{\beta} + \mathbf{M}_a\dot{\mathbf{S}}_a\dot{\boldsymbol{\beta}} \\ &\mathbf{G}_\beta = \mathbf{S}_{ua}^T\mathbf{G}_{ua} + \mathbf{S}_a^T\mathbf{G}_a. \end{aligned} \tag{22}$$

The dynamics equation has some unique characteristics. First, the term associated with the contact forces are cancelled from the dynamics equations. Second, the dynamics equations are expressed in terms of the variables of the independent joint space.

In the following, the contact force will be computed. On the basis of Eq. (13a), there are nine unknown contact forces and only six equations, for instance, in the walking gait. Therefore, there are more constraint forces than the number of equations and the contact forces cannot be computed uniquely. To calculate the contact forces, the moore–penrose pseudoinverse will be used. Using this inverse will calculate the minimum norm constraint forces vector that will achieve the desired motion. Now, the contact forces can be calculated from Eq. (13a) and Eq. (11) as follows:

$$\mathbf{F}_{\mathbf{Leg}} = \left(\mathbf{J}_{ua}^T\right)^{\#}\left(\mathbf{M}_{1ua}\mathbf{S}_{ua}\ddot{\boldsymbol{\beta}} + \mathbf{M}_{2ua}\mathbf{S}_a\ddot{\boldsymbol{\beta}} + \mathbf{M}_{1ua}\dot{\mathbf{S}}_{ua}\dot{\boldsymbol{\beta}} + \mathbf{M}_{2ua}\dot{\mathbf{S}}_a\dot{\boldsymbol{\beta}} + \mathbf{V}_{ua} + \mathbf{G}_{ua}\right). \tag{23}$$

In above equation, $(.)^{\#}$represents moore–penrose pseudoinverse.

3. Push Recovery Controller

The aim of this section is the design of an appropriate controller in order to recover the desired posture of a quadruped robot when it is pushed. To do so, some assumptions are made. First, the direction and the magnitude of external push are assumed to be unknown. Second, it is assumed that, without loss of generality, all legs are in contact with the ground. However, the algorithm can be used to recover the balanced configuration with some modifications even when the robot uses other gait such as walking and trotting. Since all legs are in the stance phase, the elements of the independent joint space are

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

the position and the orientation of the main body. When an external disturbance is applied on the robot, the robot accelerates and it causes the robot deviates from the desired posture. Therefore, the position and velocity errors increase. To counteract this push, the appropriate joint torques should be exerted on the robot. However, the main question is that how much torques should be provided by actuators. To calculate the appropriate torques, we first compute proper accelerations. To oppose to the push, the main body accelerations in the opposite direction of the applied push should be acted on the robot. Higher main body accelerations are required for greater position error and rate of errors. Therefore, a proportional-derivative (PD) controller is used to calculate the required accelerations. The connection between the external disturbance and the main body accelerations is the main body position and velocity errors because the effect of the external push appears in the position errors and the rate of errors. In other words, larger external pushes produce higher errors and thus higher main body accelerations are needed to be applied on the robot. In brief, in the proposed controller, the main body accelerations are adjusted preciously in order to handle external disturbances. The desired accelerations can be computed as follows:

$$\ddot{\boldsymbol{\beta}}_d = \mathbf{K}_P \left(\boldsymbol{\beta}_d - \boldsymbol{\beta}\right) + \mathbf{K}_v \left(\dot{\boldsymbol{\beta}}_d - \dot{\boldsymbol{\beta}}\right). \tag{24}$$

In fact, a virtual set of spring-damper elements between the desired and actual values is used to return the robot to its desired posture in the presence of an external push. If the desired accelerations are applied on the robot, the robot can restore its balance in confronting external disturbances. These accelerations are applied on the robot through the joint torques calculated by Eq. (21). However, there is a major problem in applying these torques. Since these robots are in contact with environment, the high accelerations cause the slippage of the stance feet or losing the robot stability due to tumbling. In other words, the applied accelerations are restricted by the robot stability and the friction constraints. Therefore, a certain amount of external pushes can be compensated by using the balance controller. In order to improve the performance of the push recovery algorithm and also speed up recovering the robot balance, the admissible accelerations should be used. They are the maximum accelerations which hold the limitations. Therefore, we use an optimization problem to calculate the appropriate accelerations under the stability and friction constraints. In previous works, the ZMP or COP criteria are used to maintain the stability. In these criteria, the point-mass model is used to develop a stability constraint. However, in this article, we will introduce a stability condition for the full-dynamics model of the robot which takes the masses of legs into consideration and it can guarantee the robot stability in motion on uneven terrains.

3.1. The stability condition

In the following, an appropriate stability condition will be presented and it will be formulated such that a linear constraint in terms of the main body accelerations is achieved. To keep the robot stability in motion on even or uneven terrains, the tumbling moments about all support edges must be positive. More precisely, the moments of inertial and gravitational forces about the support edges must reinforce the contact between the stance legs and the ground. The model of the robot with all inertial and gravitational forces exerted on the COG of the main body and also the COG of all legs is shown in Fig. 2. When all legs are in the support phase, based on this model, the stability condition is defined as

$$\begin{aligned} &M_{stab} > 0 \\ &M_{stab} = \min\left\{ M_{41} \quad M_{34} \quad M_{23} \quad M_{12} \right\}, \end{aligned} \tag{25}$$

where, for instance, M_{41} is the moment of external forces about the support edge which is composed of the tips of the leg 1 and leg 4. The direction of each support edge is determined so that the support polygon goes around in a clockwise sense. In other words, the direction of the resultant moment and the unit vector of the support edge must be the same. This is due to the fact that the moment along this direction enforces the contact between the stance legs and the ground. For instance, to calculate M_{41}, the direction of the tumbling moment should be along the line connecting the tip of leg 1 to the

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

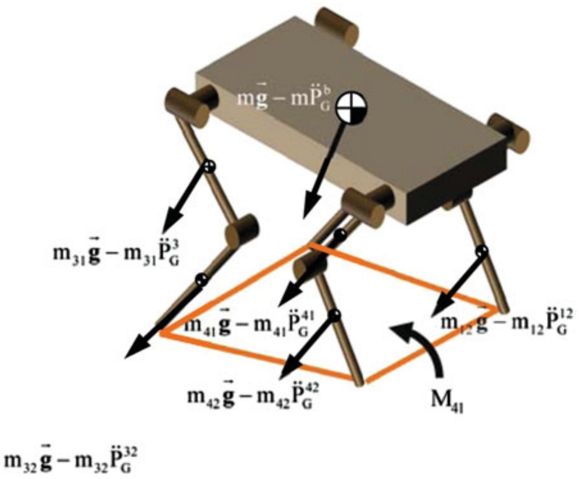
$m\vec{g} - m\ddot{P}_G^b$
$m_{31}\vec{g} - m_{31}\ddot{P}_G^3$
$m_{41}\vec{g} - m_{41}\ddot{P}_G^{41}$
$m_{12}\vec{g} - m_{12}\ddot{P}_G^{12}$
$m_{42}\vec{g} - m_{42}\ddot{P}_G^{42}$
M_{41}
$m_{32}\vec{g} - m_{32}\ddot{P}_G^{32}$

Fig. 2. The support polygon of the stance legs (red lines) as well as the inertial and gravitational forces exerted on the COG of the main body and the COGs of legs.

tip of leg 4. The moment about the support edges, for instance, M_{41} can be computed as

$$M_{41} = \mathbf{e}_{41}^T\left((\mathbf{P}_G - \mathbf{P}_i) \times \left(-m\ddot{\mathbf{P}}_G + m\mathbf{g}\right) + \sum_{i=1}^{m}\sum_{j=1}^{n}\left(\left(\mathbf{P}_G^{ij} - \mathbf{P}_i\right) \times \left(-m_{ij}\ddot{\mathbf{P}}_G^{ij} + m_{ij}\mathbf{g}\right)\right)\right), \quad (26)$$

where $\mathbf{P}_i$ is an arbitrary point on the support edge which is composed of supporting leg 1 and leg 4. In addition, $\mathbf{P}_G$ and $\ddot{\mathbf{P}}_G$ are the vector of the position and the linear acceleration of the main body, respectively and $\mathbf{P}_G^{ij}$ and $\ddot{\mathbf{P}}_G^{ij}$ are the position and the linear acceleration of the jth segment of the ith leg, respectively. In addition, $\mathbf{e}_{41}$ is the unit vector of the support edge composed of supporting leg 1 and leg 4 which is calculated as

$$\mathbf{e}_{41} = \frac{\mathbf{P}_4 - \mathbf{P}_1}{\|\mathbf{P}_4 - \mathbf{P}_1\|}. \quad (27)$$

It is assumed that the robot has m legs and each leg is composed of n segments. For our quadruped robot, we have $m = 4, n = 2$. It is noted that the angular accelerations of main body and legs is completely ignored in the derivation of the stability condition.

Since the optimization variables are the accelerations of main body and also the stability condition should be added to the optimization problem as its constraint, the stability condition should be expressed in terms of the main body accelerations. The accelerations of the COGs of main body and all legs can be expressed in terms of the main body accelerations as follows:

$$\ddot{\mathbf{P}}_G = \begin{bmatrix} I_{3\times3} & O_{3\times3} \end{bmatrix}\ddot{\boldsymbol{\beta}}, \ddot{\mathbf{P}}_G^{ij} = \mathbf{J}_B^{ij}\ddot{\boldsymbol{\beta}} + \mathbf{J}_L^{ij}\ddot{\mathbf{q}}_L^i + \dot{\mathbf{J}}_B^{ij}\dot{\boldsymbol{\beta}} + \dot{\mathbf{J}}_L^{ij}\dot{\mathbf{q}}_L^i. \quad (28)$$

By the substitution of Eq. (28) in Eq. (26), we have

$$\begin{aligned} M_{41} = \mathbf{e}_{41}^T\Bigg((\mathbf{P}_G - \mathbf{P}_i) \times \left(-m\begin{bmatrix} I_{3\times3} & O_{3\times3} \end{bmatrix}\ddot{\boldsymbol{\beta}} + m\mathbf{g}\right) + \sum_{i=1}^{m}\sum_{j=1}^{n}\Big(\left(\mathbf{P}_G^{ij} - \mathbf{P}_i\right) \\ \times \left(-m_{ij}\left(\mathbf{J}_B^{ij}\ddot{\boldsymbol{\beta}} + \mathbf{J}_L^{ij}\ddot{\mathbf{q}}_L^i + \dot{\mathbf{J}}_B^{ij}\dot{\boldsymbol{\beta}} + \dot{\mathbf{J}}_L^{ij}\dot{\mathbf{q}}_L^i\right) + m_{ij}\mathbf{g}\right)\Big)\Bigg) > 0, \quad (29)\end{aligned}$$

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

The above stability condition can be rearranged and stated as a linear function of the variables of optimization problem i.e.,$\ddot{\boldsymbol{\beta}}$ as follows:

$$\begin{aligned}
M_{41} &= \mathbf{L}_{41}^{1}\ddot{\boldsymbol{\beta}} + L_{41}^{2} \\
\mathbf{L}_{\mathbf{41}}^{\mathbf{1}} &= \mathbf{e}_{41}^{T}\left((\mathbf{P}_G - \mathbf{P}_i) \times -m\begin{bmatrix} I_{3\times 3} & O_{3\times 3}\end{bmatrix} + \sum_{i=1}^{m}\sum_{j=1}^{n}\left(\left(\mathbf{P}_G^{ij} - \mathbf{P}_i\right) \times -m_{ij}\mathbf{J}_{BL}^{iji}\right)\right) \\
L_{41}^{2} &= \mathbf{e}_{41}^{T}\left((\mathbf{P}_G - \mathbf{P}_i) \times m\mathbf{g} + \sum_{i=1}^{m}\sum_{j=1}^{n}\left(\left(\mathbf{P}_G^{ij} - \mathbf{P}_i\right) \times \left(-m_{ij}\mathbf{J}_B^{ij}\ddot{\mathbf{q}}_L^{i} - \dot{\mathbf{J}}_B^{ij}\dot{\boldsymbol{\beta}} - \dot{\mathbf{J}}_L^{ij}\dot{\mathbf{q}}_L^{i} + m_{ij}\mathbf{g}\right)\right)\right).
\end{aligned} \tag{30}$$

The same procedure can be used to obtain the similar equations for other support edges. Therefore, the stability conditions for all support edges can be defined as

$$\begin{aligned}
&\mathbf{L}_T^{1}\ddot{\boldsymbol{\beta}} + \mathbf{L}_T^{2} > \mathbf{O} \\
&\mathbf{L}_T^{\mathbf{1}} = \begin{bmatrix} \mathbf{L}_{41}^{1\,T} & \mathbf{L}_{34}^{1\,T} & \mathbf{L}_{23}^{1\,T} & \mathbf{L}_{12}^{1\,T} \end{bmatrix}^{T} \\
&\mathbf{L}_T^{2} = \begin{bmatrix} L_{41}^{2} & L_{34}^{2} & L_{23}^{2} & L_{12}^{2} \end{bmatrix}^{T}.
\end{aligned} \tag{31}$$

The above equation introduces the inequality linear constraints in terms of the position and orientation of main body which should be added to the optimization problem as the stability condition.

3.2. The friction constraint

The robot can restore its balance as long as the stance feet remain stationary on the ground. In the following, friction condition for the stance legs will be proposed and expressed as a linear function of the main body accelerations. When large pushes are applied on the robot, the contact forces increase very much and the stance legs may slip. Therefore, a constraint should be defined in terms of the accelerations of main body to restrict these accelerations. To describe the friction properties at the contact points, the coulomb's friction model is used. This model defines a condition on the contact forces, $f_t \leq \mu f_n$, to avoid the slip occurring at the stance feet. f_t and f_n are the magnitude of the tangential and the normal contact forces, respectively and μ is the friction coefficient. In order to avoid the slip and maintain the contact, the admissible contact forces should satisfy below conditions

$$\sqrt{f_{tx}^{2} + f_{t,y}^{2}} \leq \mu f_n,\ f_n \geq 0. \tag{32}$$

In addition, to avoid the separation between the stance legs and the ground, the normal contact forces at the contact points must be positive. In other words, the normal contact forces should be along the outward normal unit vector at the contact points. This condition is defined due to the fact that each leg can only push the ground and cannot pull it. The friction conditions impose some restrictions on the contact forces. However, these conditions are very non-linear and if they are used in this form, the complexity of the problem increases. To simplify the friction condition, we approximate the friction cone with an inscribed pyramid. In this case, the friction condition is expressed as

$$|f_{tx}| \leq \mu_n f_n,\ \left|f_{ty}\right| \leq \mu_n f_n,\ f_n \geq 0, \tag{33}$$

where μ_n is the approximated friction coefficient and its value is $\mu_n = \frac{\mu}{\sqrt{2}}$. The linear function of the friction constraint in terms of the contact forces can be expressed as

$$\begin{aligned}
&\mathbf{C}_{AS}^{1}\mathbf{f}_i \leq \mathbf{C}_{AS}^{2} \\
&\mathbf{f}_i = \begin{bmatrix} f_{tx}^{i} & f_{ty}^{i} & f_n^{i} \end{bmatrix}^{T}, \mathbf{C}_{AS}^{2} = \begin{bmatrix} 1 & 0 & -\mu_n \\ -1 & 0 & -\mu_n \\ 0 & 1 & -\mu_n \\ 0 & -1 & -\mu_n \\ 0 & 0 & -1 \end{bmatrix}, \mathbf{C}_{AS}^{2} = O_{5\times 1}.
\end{aligned} \tag{34}$$

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

In above equation, $\mathbf{f}_i$ is the contact force of the ith stance leg which is defined in the local coordinate system located at the contact point. When the robot is located on an even terrain, the directions of local contact forces are exactly along the unit vectors of the world frame. However, in motion over uneven terrains, this relationship is not valid. Therefore, the mapping from the local frame to the world frame is defined as

$$\mathbf{F}_i = \mathbf{R}_i^W \mathbf{f}_i, \tag{35}$$

where $\mathbf{R}_i^W$ is the rotation matrix between the local frame attached to the contact point and the world frame and $\mathbf{F}_i$ is the contact force of ith leg expressed in the world frame. For four-leg support phase, the friction condition is given by

$$\mathbf{C}^1_{AS,T}\mathbf{F}_{Leg} \leq \mathbf{C}^2_{AS,T}$$

$$\mathbf{C}^1_{AS,T} = \begin{bmatrix} \left(\mathbf{C}^1_{AS,1}\mathbf{R}_1^W\right) & \mathbf{O}_{5\times 3} & \mathbf{O}_{5\times 3} & \mathbf{O}_{5\times 3} \\ \mathbf{O}_{5\times 3} & \left(\mathbf{C}^1_{AS,2}\mathbf{R}_2^W\right) & \mathbf{O}_{5\times 3} & \mathbf{O}_{5\times 3} \\ \mathbf{O}_{5\times 3} & \mathbf{O}_{5\times 3} & \left(\mathbf{C}^1_{AS,3}\mathbf{R}_3^W\right) & \mathbf{O}_{5\times 3} \\ \mathbf{O}_{5\times 3} & \mathbf{O}_{5\times 3} & \mathbf{O}_{5\times 3} & \left(\mathbf{C}^1_{AS,4}\mathbf{R}_4^W\right) \end{bmatrix},$$

$$\mathbf{F}_{Leg} = \begin{bmatrix} \mathbf{f}_1 & \mathbf{f}_2 & \mathbf{f}_3 & \mathbf{f}_4 \end{bmatrix}^T, \mathbf{C}^2_{AS,T} = \mathrm{O}_{20\times 1}. \tag{36}$$

By substituting of Eq. (23) in Eq. (36), the friction constraint will be expressed in terms of the main body accelerations as follows:

$$\mathbf{C}^1_{AS}\ddot{\boldsymbol{\beta}} \leq \mathbf{C}^2_{AS}, \tag{37}$$

where

$$\begin{aligned} \mathbf{C}^1_{AS} &= \mathbf{C}^1_{AS,T}\left(\mathbf{J}_{ua}^T\right)^{\#}\left(\mathbf{M}_{1ua}\mathbf{S}_{ua} + \mathbf{M}_{2ua}\mathbf{S}_a\right) \\ \mathbf{C}^2_{AS} &= \mathbf{C}^2_{AS,T} - \left(\mathbf{J}_{ua}^T\right)^{\#}\left(\mathbf{M}_{1ua}\dot{\mathbf{S}}_{ua}\dot{\boldsymbol{\beta}} + \mathbf{M}_{2ua}\dot{\mathbf{S}}_a\dot{\boldsymbol{\beta}} + \mathbf{V}_{ua} + \mathbf{G}_{ua}\right). \end{aligned} \tag{38}$$

As seen, the friction constraint is also a linear function of main body accelerations. This makes the problem very simple and consequently reduces the required time for finding the solution of the optimization problem. Therefore, the algorithm can be used in real-time implementation.

3.3. The joint torques limitations

Since each actuator cannot supply an unlimited amount of torque to each joint of the robot, saturation constraints should be taken into account. Therefore, a constraint should be defined to impose these limitations on the joints of the robot in the push recovery algorithm. This is due to the fact that higher torques should be applied on joints to produce higher joint accelerations. Additionally, this condition affects the push recovery performance and may restrict the maximum push which may handle with the balance controller. The saturation constraint on the joint torque is stated as

$$\tau_{\min} \leq \tau \leq \tau_{\max}, \tag{39}$$

where $\tau_{\min}$ and $\tau_{\max}$ are the minimum and maximum available torques. Since above condition will be considered as a constraint of the optimization problem, it should be expressed in terms of optimization variables i.e., the accelerations of main body. The substitution of Eq. (21) in Eq. (39) yields

$$\tau_{\min} \leq \left(\mathbf{S}_a^T\right)^{\#}\left(\mathbf{M}_\beta\ddot{\boldsymbol{\beta}} + \mathbf{V}_\beta + \mathbf{G}_\beta\right) \leq \tau_{\max}. \tag{40}$$

In above equation, $(.)^{\#}$ denotes the Moore–Penrose pseudoinverse. As seen in Eq. (40), this equation leads to the linear constraints which are the functions of main body accelerations. Therefore, the torque

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

saturation constraints can be given by

$$\begin{aligned}&\mathbf{A}_\tau \ddot{\boldsymbol{\beta}} \le \mathbf{B}_\tau\\ &\mathbf{A}_\tau = \begin{bmatrix}\mathbf{A}_{\tau_{\max}}\\ \mathbf{A}_{\tau_{\min}}\end{bmatrix}, \mathbf{B}_\tau = \begin{bmatrix}\mathbf{B}_{\tau_{\max}}\\ \mathbf{B}_{\tau_{\min}}\end{bmatrix}\\ &\mathbf{A}_{\tau_{\max}} = \left(\mathbf{S}_a^T\right)^{\#}\mathbf{M}_\beta, \mathbf{A}_{\tau_{\min}} = -\left(\mathbf{S}_a^T\right)^{\#}\mathbf{M}_\beta\,.\\ &\mathbf{B}_{\tau_{\max}} = \tau_{\max} - \left(\mathbf{S}_a^T\right)^{\#}\left(\mathbf{V}_\beta + \mathbf{G}_\beta\right)\\ &\mathbf{B}_{\tau_{\min}} = \left(\mathbf{S}_a^T\right)^{\#}\left(\mathbf{V}_\beta + \mathbf{G}_\beta\right) - \tau_{\min}\end{aligned} \tag{41}$$

Three main constraints were defined and also expressed as the linear functions of variables of the optimization problem. Now, we can introduce the push recovery strategy. It is based on an optimization problem which computes the admissible accelerations meeting those constraints. The admissible accelerations are the output of the optimization problem. In the case where the desired accelerations hold the constraints, the admissible and desired accelerations are the same. However, when the desired accelerations violate the constraints, the admissible accelerations are computed by using the optimization problem.

4. The Push Recovery Strategy

As explained earlier, the main objective of the push recovery algorithm is to determine the maximum admissible main body accelerations which recover the robot balance after an unexpected push. However, the maximum accelerations of the main body are restricted by the constraints introduced in previous sections. To resolve this problem, we define an optimization problem. The cost function of the optimization problem is given by

$$J = \left(\ddot{\boldsymbol{\beta}} - \ddot{\boldsymbol{\beta}}_d\right)^T W \left(\ddot{\boldsymbol{\beta}} - \ddot{\boldsymbol{\beta}}_d\right), \tag{42}$$

where W is a diagonal weight matrix. Also, $\ddot{\boldsymbol{\beta}}$ and $\ddot{\boldsymbol{\beta}}_d$ are the admissible and desired accelerations, respectively. The weighted errors of the main body acceleration are selected as the cost function due to the fact that the ideal case is to use the desired acceleration for the push recovery. To obtain the main body accelerations i.e., $\ddot{\boldsymbol{\beta}}$, above cost function is minimized such that Eq. (31), Eq. (37), and Eq. (41) are held. Therefore, the optimization problem is summarized as follows:

$$\begin{aligned}&\min_{\ddot{\boldsymbol{\beta}}} J\\ &s.t.\ \mathbf{C}_{AS}^1 \ddot{\boldsymbol{\beta}} \le \mathbf{C}_{AS}^2, \mathbf{L}_T^1 \ddot{\boldsymbol{\beta}} + \mathbf{L}_T^2 > \mathbf{O}, \mathbf{A}_\tau \ddot{\boldsymbol{\beta}} \le \mathbf{B}_\tau.\end{aligned} \tag{43}$$

The optimization problem is easily transformed into a linear Least Squares problem. This is due to the fact that the constraints are formulated as the linear functions of the optimization variables. The main advantage of the defined optimization problem is its computational efficiency. The push recovery algorithm is shown in Fig. 3. The desired position and velocity of the main body and also the position of the stance legs are considered as the inputs of the algorithm. The desired accelerations of main body are computed by using the PD controller. Next, the optimization algorithm calculates the admissible accelerations of the main body. Finally, the admissible accelerations are applied to the robot through the joint torques obtained by the inverse dynamics controller.

5. Obtained Results

To show the merits of the proposed balance recovery algorithm, it is examined on a quadruped robot, as depicted in Fig. 4, in the standing posture on even and uneven terrains through performing several simulations and the obtained results are discussed. The robot is composed of four legs. Each leg has three degrees of freedom to be able the robot to place its tip of leg freely anywhere in the 3D space. The joints of each leg consist of two hip joints about pitch and roll axes and a knee joint about pitch

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

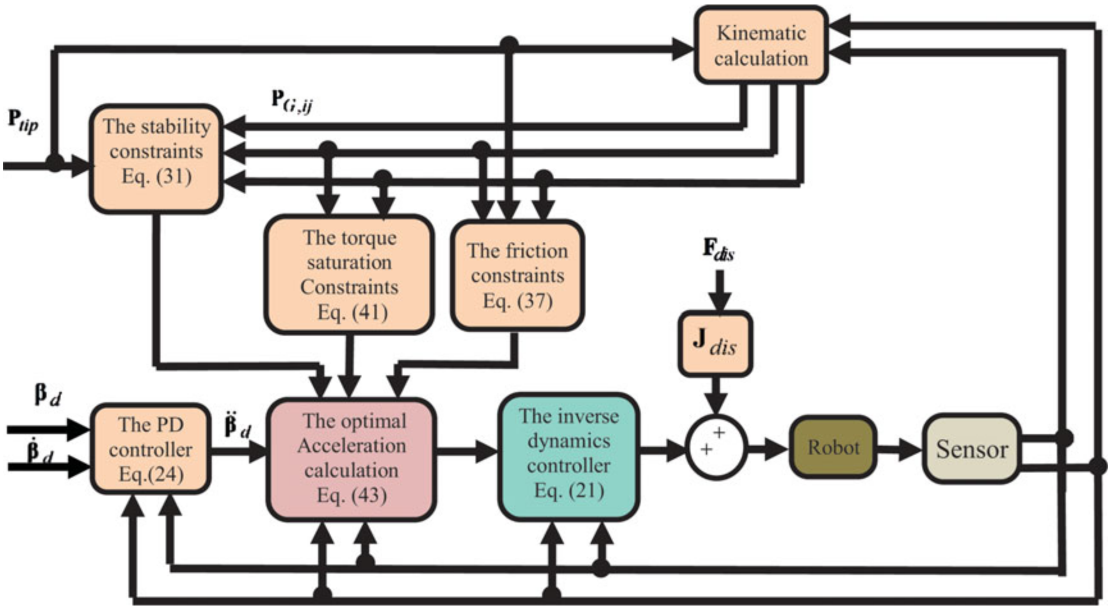
Kinematic calculation
P_{tip}
$P_{G,ij}$
The stability constraints Eq. (31)
The torque saturation Constraints Eq. (41)
The friction constraints Eq. (37)
F_{dis}
J_{dis}
β_d
$\dot{\beta}_d$
$\ddot{\beta}_d$
The PD controller Eq.(24)
The optimal Acceleration calculation Eq. (43)
The inverse dynamics controller Eq. (21)
+
+
Robot
Sensor

Fig. 3. The push recovery block diagram.

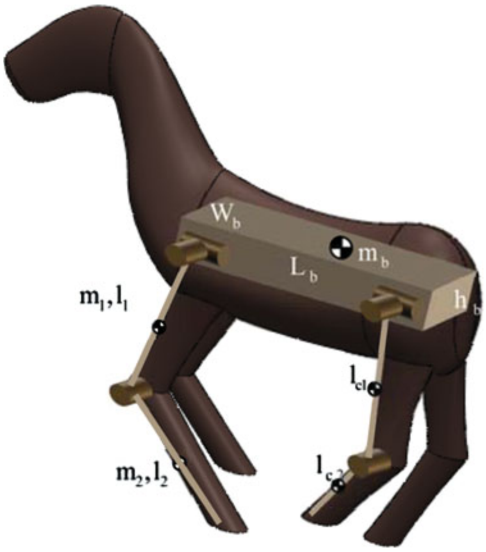

Fig. 4. The model of the Quadruped robot.

axis. In addition, the main body of the robot is free to move along all axes of the coordinate system and also rotate about all three axes. Therefore, the overall degree of freedom of the robot is 18 where the six DOFs of main body are passive. The specifications of the robot and the controller parameters are summarized in Table I. The link lengths and also the mass properties of the robot are similar to the Starl*ETH* quadruped robot.[35]

The validity of the proposed algorithm will be examined in three different case studies i.e., in standing over an even terrain, an uneven terrain, and a soft terrain.

5.1. Case A. the push recovery on an even terrain

To show the effectiveness of the proposed algorithm, this algorithm first will be tested on the quadruped robot in the standing posture over an even terrain. It is assumed that the robot is standing at rest on a flat terrain and a push of the magnitude of 250 N at 0.5 s is exerted on the robot from the middle of the back in the forward direction i.e., along x-axis. The applied push lasts for 0.1 s.

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

Table I. The mass properties and link lengths and the controller parameters.

Symbol	Value (unit)	Definition
m_b	40 (kg)	Robot's body Mass
l_1	0.2 (m)	length of thigh
l_2	0.22 (m)	length of shank
m_1	2 (kg)	Mass of thigh
m_2	0.5 (kg)	Mass of shank
L_b	0.5 (m)	Length of body
W_b	0.37 (m)	width of body
h_b	0.1 (m)	height of body
l_{c1}	0.02 (m)	Distance between the hip joint and the COG of thigh
l_{c2}	0.08 (m)	Distance between the knee joint and the COG of shank
μ	0.25	friction coefficient
$\mathbf{K}_p$	$diag([10\ 1\ 10\ 1\ 10\ 1])$	proportional gain matrix
$\mathbf{K}_v$	$diag([1\ 0.1\ 1\ 0.1\ 1\ 0.1])$	damping gain matrix
$[\tau_{\min}\ \tau_{\max}]$	$[-30\ 30](N.m)$	Minimum and Maximum available torques

Table II. The properties of the position tracking errors.

	RMS	Maximum values
The position error of main body along x-axis(m)	0.0943	0.1943
The position error of main body along z-axis(m)	0.0041	0.0094
The orientation error of main body about Pitch axis (rad)	0.0094	0.0248

When the robot is pushed along x-axis, we expect that the robot moves forward and upward and also rotates about the pitch axis. Therefore, the gains for those axes are selected more than the gains of the rest of the joints of main body. Additionally, the selection of the higher gains is equivalent to employ the stiffer compliance elements. In this case, the movement of the robot is restricted but higher torques are required and actuators may go beyond the saturation limits. Here, to better show the movement of the robot during the push recovery, low gains are selected. These gains are shown in Table I. As seen, the gains along x- and z-axes and also about y-axis are higher than the gains along y-axis and also about x- and z-axes. The effect of the gains on the performance of algorithm will be considered at the end of this section.

A series of snapshots representing the robot when an external push is applied on the robot is shown in Fig. 5. The response of the robot to the push is shown in Fig. 6(a)–(c). As seen in the figure, the robot exploits the motion along z-axis and also the rotational motion about the pitch axis to restore its balance. The properties of the position tracking errors are shown in Table II. As seen, the error along x-axis is higher than the errors along other axes because the robot is pushed along x-axis. Besides, the robot uses the rotation about the pitch axis in recovering its balance because the rotation of the robot is more effective in the balance recovery than its motion along z-axis. This means that the robot uses the regulation of its angular momentum to recover its balance instead of regulating its linear momentum. Also, the linear momentum of main body along x-axis helps the robot to regain its balance. Since the applied push is exerted only along x-axis, other DOFs of main body remain unchanged and thus they are not shown in the figure. To prove that the robot remains stable during the push recovery, the minimum value of the tumbling moments about all support edges is computed and shown in Fig. 6(f). The robot maintains its stability because the minimum value of the moments about all support edges is always greater than zero. After that the robot is pushed, the tumbling moment begins to decrease due to increasing the accelerations of the main body. However, the tumbling moment never reaches below zero due to using the optimization algorithm. The normal components of the contact forces for the front and hind legs are shown in Fig. 6(d). It can be observed in this figure that the stance legs do not lose their contact with the ground. However, when an external push is exerted, the contact forces of the rear legs decrease whereas the contact forces of front legs increases. This is due to the fact that the robot moves forward. Therefore, the hind are more likely to be separated from the ground. To consider whether the slip in the stance legs occurs or not during the push recovery, the tangential

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

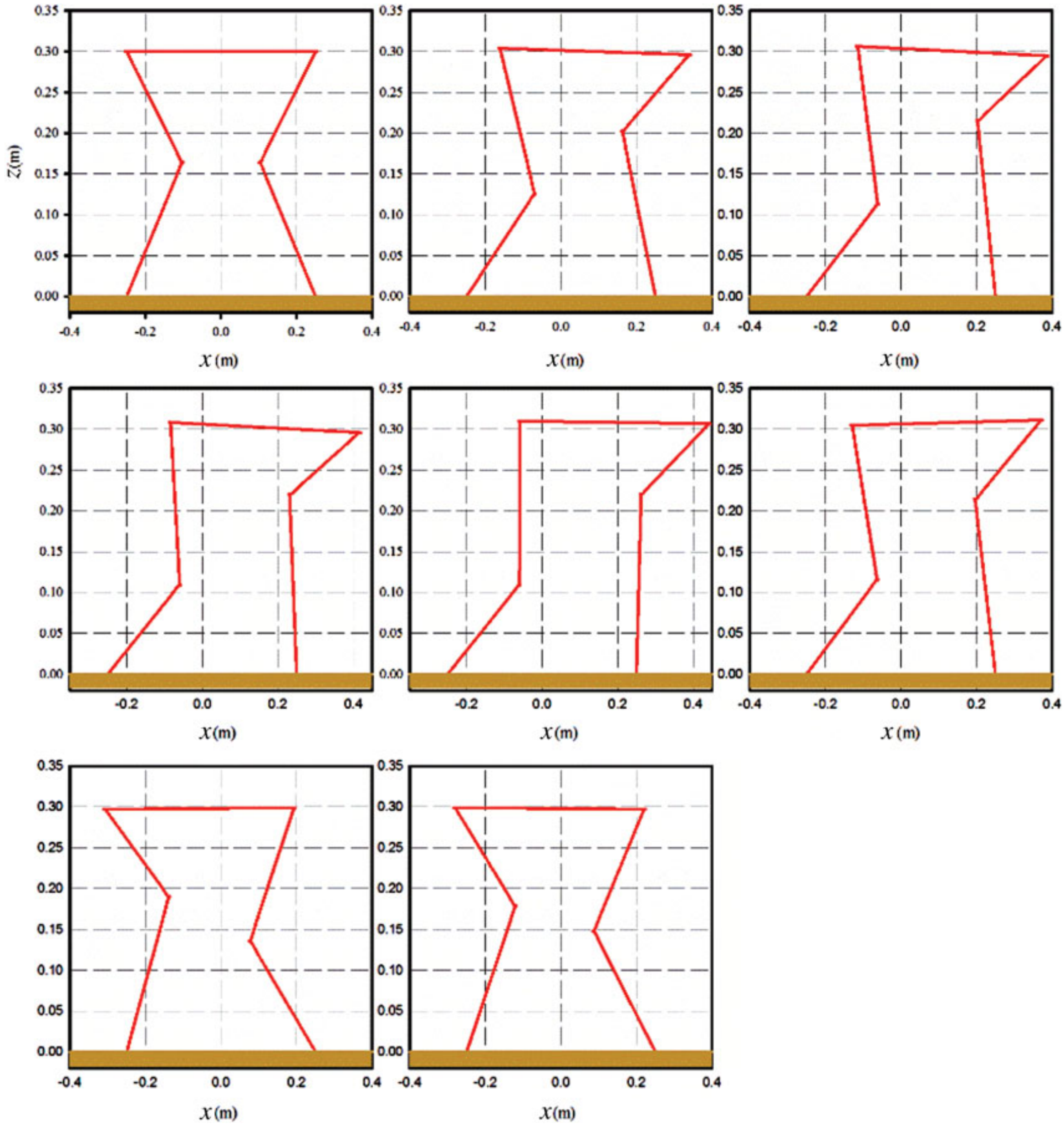

Fig. 5. The snapshots of the robot when an external push of 250 N is applied on the robot over an even terrain.

forces and the limits of the slip of the front and hind legs are shown in Fig. 6(h) and (L). As seen in these figures, the tangential components of the contact forces are kept within the safe region and it guarantees a slip-free motion. Since the push is applied along x-axis, the tangential forces along y-axis are negligible and only x-directional tangential contact forces are shown in the figure. Since the normal component of contact force of this leg decrease, the bounds of the slippage decrease. Meanwhile, the tangential component of contact force of this leg grows due to increasing the main body accelerations. Consequently, the hind leg i.e., the leg 2 reaches its slip limits after the push as depicted in the figure. The desired accelerations and their admissible values are shown in Fig. 6(d) and (e). As seen, the maximum accelerations are not achievable at some instants of time due to the constraints. The joint torques and the saturation limits are shown in Fig. 6(m) and (n). The saturation of joint torques does not happen during the push recovery because the joint torques always lie within the saturation limits. However, the joint torques raise after exerting the push due to increasing the main boy accelerations. Therefore, the joint torques reach close to the saturation limits. It is noted that in this case the applied energy to the robot due to the push will be absorbed by the energy consumed by the actuators. The effect of the selection of the proportional and damping gains on the performance of push recovery has been considered and shown in Fig. 7. Two different sets of gains

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

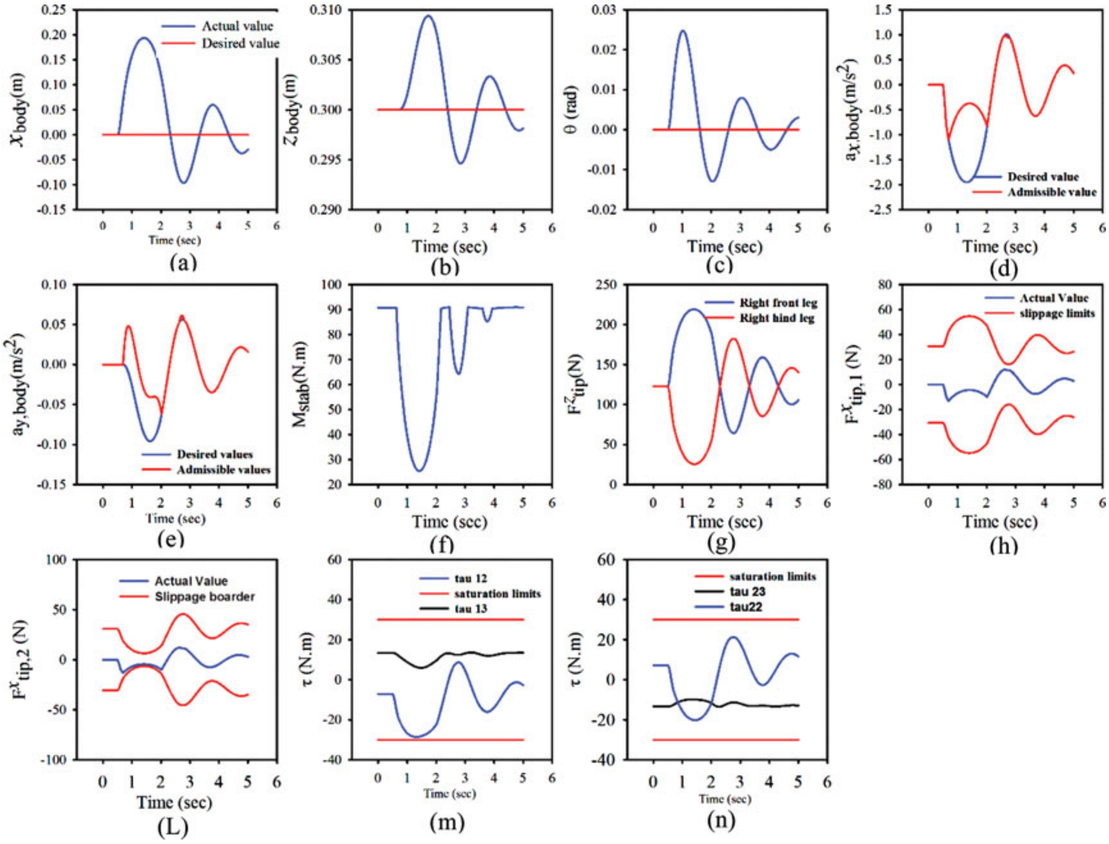

Fig. 6. The trajectory of important properties of the simulation when a push with the magnitude of 200 N is applied on the robot while standing over an even terrain.

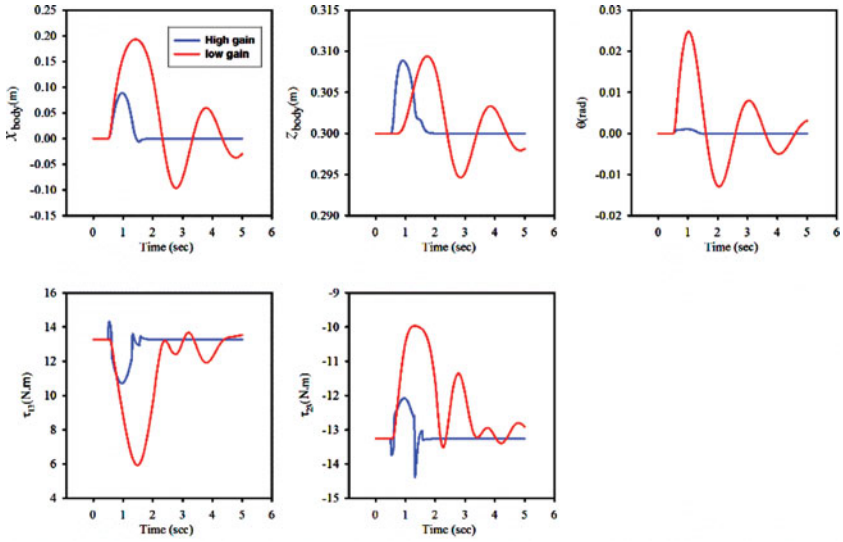

Fig. 7. The effect of the selection of the controller gains on the performance of the proposed algorithm.

i.e., low gains and high gains were chosen and the simulations were performed with those gains. The low and high gains are selected as $\mathbf{K}_p = diag([10\ 1\ 10\ 1\ 10\ 1])$ and $\mathbf{K}_v = diag([1\ 0.1\ 1\ 0.1\ 1\ 0.1])$, $\mathbf{K}_p = diag([100\ 10\ 100\ 10\ 100\ 10])$ and $\mathbf{K}_v = diag([10\ 1\ 10\ 1\ 10\ 1])$, respectively. As seen in the figure, when the gains are high, the motion of the robot is restricted. This is due to the fact that the robot should overcome a stiff set of virtual spring-damper elements to be able to move. However, as

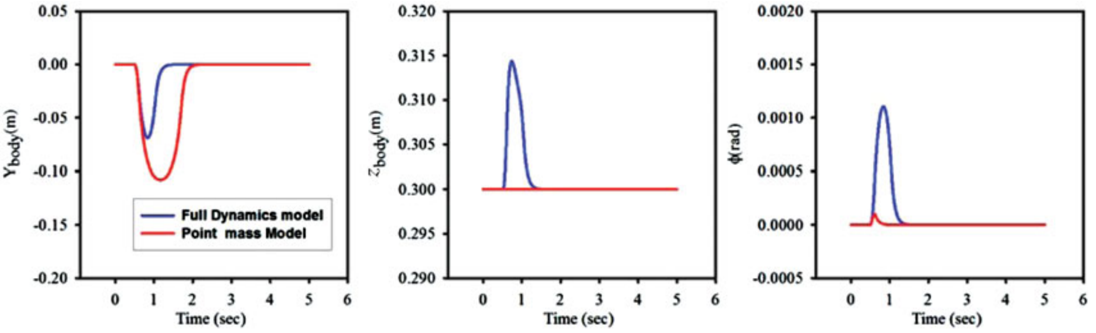

Fig. 8. The comparison between the point mass model and the full-dynamics model in the push recovery of the robot.

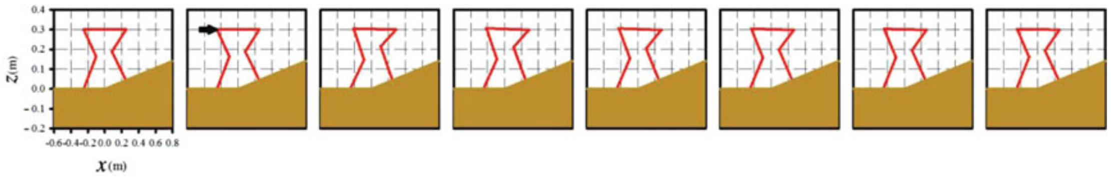

Fig. 9. The snapshots of the robot when an external push of 200 N is applied on the robot over an inclined terrain.

shown in the figure, higher torques are required for restoring the robot balance in the case of high gains.

The obtained results in this paper were compared to the algorithm which the ZMP criterion based on a point-mass model was used as its stability condition. In the simulation, the friction coefficient is assumed to be high so that the slip of the stance legs is prevented and also the robot is pushed along y-axis. The results for two models are shown in Fig. 8. As seen, using the full-dynamics model gives better performance and the robot fully exploits all their motion capabilities to recover its balance. Compared to the point mass model, the robot uses the motion along z-axis and also the rotation about the pitch axis to recover its balance whereas the motion along x-axis decreases.

5.2. *Case B. The push recovery on uneven terrains*

As mentioned earlier, the proposed balance controller can be used to recover the robot balance in standing on uneven terrains. Thus, the robot is assumed to be located on an inclined terrain with the slope of 10° and a push of 200 N at 0.5 s is exerted on the back of the robot in the forward direction. The applied push continues till 0.6 s. The proportional and damping gain matrices are selected as $\mathbf{K}_p = diag([20\ 20\ 20\ 20\ 20\ 20])$ and $\mathbf{K}_v = diag([15\ 15\ 15\ 15\ 15\ 15])$, respectively. The snapshots depicting the robot motion during the balance recovery are shown in Fig. 9. To consider the performance of the algorithm on this terrain, the simulation results of the motion of the robot over this terrain and also other important properties of the balance recovery are depicted in Fig. 10. The robot comes back to the desired posture quickly because the gains are higher than the gains for previous case. Due to the usage of the optimization algorithm, the maximum accelerations are not available during the balance recovery. In the standing over uneven terrain, important matter is the robot stability. As seen, the balance of the robot also is kept because the tumbling moments about all support edges are positive. The slippage of stance foot and also the saturation of actuators do not occur during the balance recovery.

5.3. *Case C. The push recovery on a soft terrain*

Finally, the effectiveness of the balance controller is tested on the robot while standing over a soft terrain. To model the soft terrain, spring-damper elements are used under the tips of the stance feet. It is assumed that these elements act only along z-axis. The stiffness and damping coefficients of the terrain are selected as $\mathbf{K} = diag([\ 10^5\ \ 10^5\ \ 10^5\ \ 10^5\ ])$ and $C = diag([\ 50\ 50\ 50\ 50\ ])$, respectively. The proportional and damping gain matrices are similar to the gains which were selected for the uneven

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

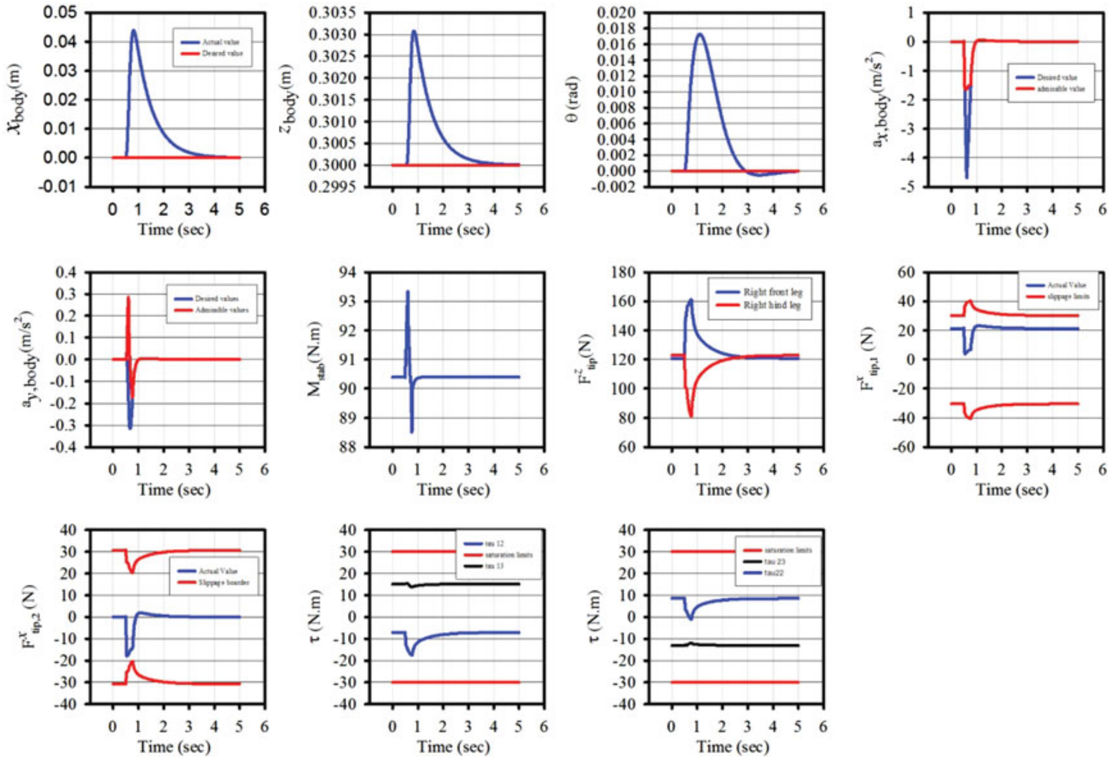

Fig. 10. The trajectory of the important properties of the simulation when a push with the magnitude of 200 N is applied on the robot while standing on an inclined terrain with the slope of 10°.

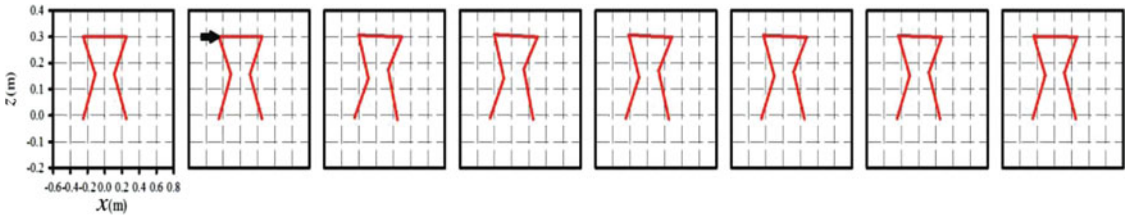

Fig. 11. The snapshots of the robot motion when an external push of 200 N is applied on the robot while standing over a soft terrain.

terrain. Here, it is assumed that a push of 200N is applied on the robot at 0.5 s. The push duration is 0.1 s. The snapshots of the robot after exerting an external push are shown in Fig. 11. As expected, when the push is exerted on the robot, since the contact forces of front legs increase, the front legs move down and the hind legs move up. The variation of the position and orientation of main body and also the important properties of the push recovery are shown in Fig. 12. As seen in the figure, the robot moves forward and increase its height in confronting a push. At the same time, the main body of the robot rotates about the pitch axis. Finally, the robot returns to the desired posture which here is equal to the initial configuration. Using the optimization algorithm restricts the main body accelerations as seen in Fig. 12. Additionally, the stability of the robot is kept. This issue can easily be proven form the minimum tumbling moment which remains positive during the balance recovery. The stance legs do not slip and also do not leave the ground. Consequently, the obtained results prove the effectiveness of proposed algorithm in order to recover the robot balance on the soft terrain.

6. Conclusions

The design of a balance controller for a quadruped robot under external pushes was investigated in this article. The explicit dynamics equations were formulated by using a computationally efficient method. Then, the free-constraint dynamics equations were developed with the proposed constraint elimination method. The stability condition to avoid tumbling was introduced and expressed as a linear function of

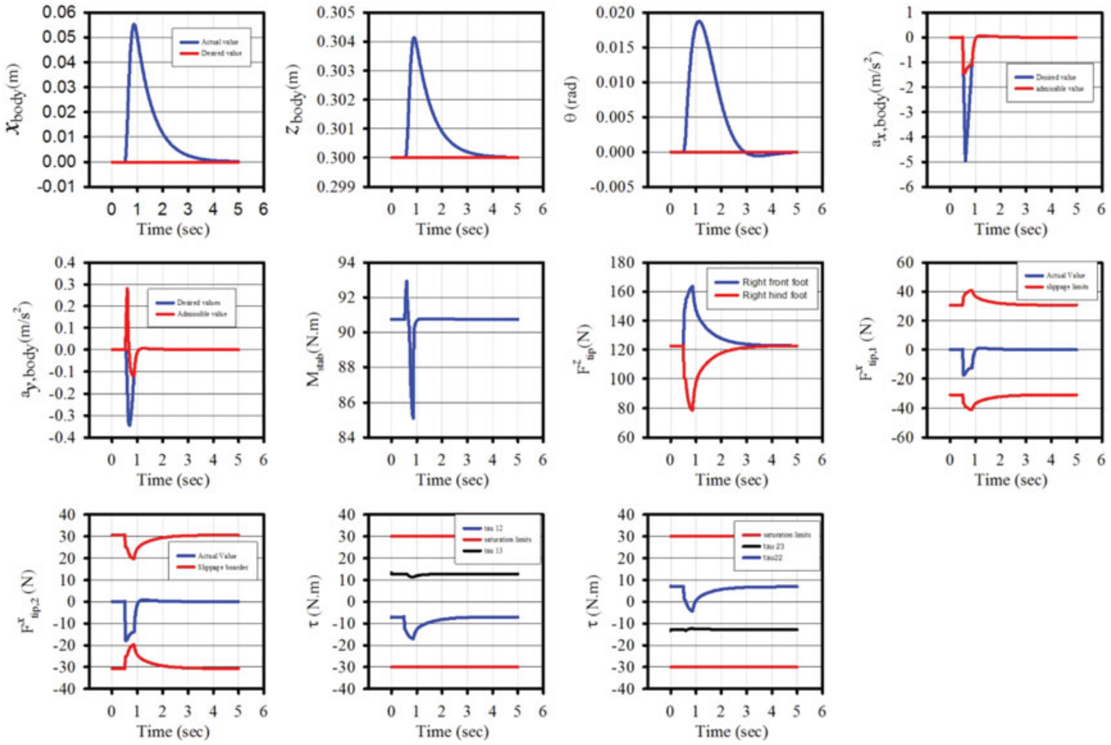

Fig. 12. The important properties of the simulation of the robot when a push with the magnitude of 200 N is exerted on the robot while standing on a soft terrain.

the optimization variables. A PD controller was used to compute the desired accelerations to recover the robot balance. Next, an optimization problem was defined to calculate the admissible accelerations so that the robot remains stable and the stance feet do not slip. The constrained optimization was solved with a linear constrained least-squares algorithm and the admissible accelerations which do not violate the friction, stability, and torque saturation conditions were obtained. To examine the proposed algorithm, the simulation studies were conducted in three different cases. First, this algorithm was tested on a quadruped robot while standing over an even terrain. The obtained results revealed that the robot can come back to the desired posture with the adjustment of the position and orientation of its main body. In the second case study, this algorithm was implemented on the robot located on an inclined terrain. The obtained results verified the merits of the applied algorithm for recovering the robot balance over such terrain. Finally, the balance controller was examined on the robot in the standing on a soft terrain. The results showed that the effectiveness of the proposed algorithm to restore the robot balance on this terrain.

References

1. W. Yu, G. Bao and Z. Wang, "Balance Recovery for Humanoid Robot in the Presence of Unknown External Push," *Proceedings of the International Conference on Mechatronics and Automation*, ICMA, Changchun, China (2009) pp. 1928–1933.
2. Y. Yoshida, K. Takeuchi, Y. Miyamoto, D. Sato and D. Nenchev, "Postural balance strategies in response to disturbances in the frontal plane and their implementation with a humanoid robot systems," *IEEE Trans. Man Cybern.: Syst.* **44**(6), 692–704 (2014).
3. D. N. Nenchev and A. Nishio, "Ankle and hip strategies for balance recovery of a biped subjected to an impact," *Robotica* **26**(05), 643–653 (2008).
4. J. Pratt, J. Carff, S. Drakunov and A. Goswami, "Capture Point: A Step Toward Humanoid Push Recovery," *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, Genova, Italy (2006) pp. 200–207.
5. B. J. Stephens and C. G. Atkeson, "Push Recovery by Stepping for Humanoid Robots with Force Controlled Joints," *10th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Nashville, TN (2010) pp. 52–59.
6. A. Goswami, S.-k. Yun, U. Nagarajan, S.-H. Lee, K. Yin and S. Kalyanakrishnan, "Direction-changing fall control of humanoid robots: Theory and experiments," *Auton. Robots* **36**(3), 199–223 (2014).

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press

7. S.-k. Yun and A. Goswami, "Tripod Fall: Concept and Experiments of a Novel Approach to Humanoid Robot Fall Damage Reduction," *Proceedings of IEEE International Conference on Robotics and Automation*, Hong Kong (2014) pp. 2799–2805.
8. J.-W. Chung, I.-H. Lee, B.-K. Cho and J.-H. Oh, "Posture stabilization strategy for a trotting point-foot quadruped robot," *J. Intell. Robot. Syst.* **72**(3-4), 325–341 (2013).
9. M. Raibert, "BigDog, the Rough-Terrain Quadruped Robot," In M. Chung (ed.), *Proceedings of the 17th IFAC World Congress*, COEX, Korea, South (2008) pp. 10822–10825.
10. B. Stephens, "Humanoid Push Recovery," *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, Pittsburgh, PA (2007) pp. 589–595.
11. J. Urata, K. Nshiwaki, Y. Nakanishi, K. Okada, S. Kagami and M. Inaba, "Online Walking Pattern Generation for Push Recovery and Minimum Delay to Commanded Change of Direction and Speed," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (2012), pp. 3411–3416.
12. V. Prahlad, G. Dip and C. Meng-Hwee, "Disturbance rejection by online ZMP compensation," *Robotica* **26**(01), 9–17 (2008).
13. P.-B. Wieber, "Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations," *IEEE-RAS International Conference on Humanoid Robots*, IEEE, Genova, Italy (2006) pp. 137–142.
14. Z. Aftab, T. Robert and P.-B. Wieber, "Ankle, Hip and Stepping Strategies for Humanoid Balance Recovery with a Single Model Predictive Control Scheme," *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, Osaka, Japan (2012) pp. 159–164.
15. C. G. Atkeson and B. Stephens, "Multiple Balance Strategies from One Optimization Criterion," *Proceedings of 7th IEEE-RAS International Conference on Humanoid Robots*, IEEE (2007) pp. 57–64.
16. T. Komura, H. Leung, S. Kudoh and J. Kuffner, "A Feedback Controller for Biped Humanoids that Can Counteract Large Perturbations During Gait," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, ICRA 2005. (2005) pp. 1989–1995.
17. Y. Wang, R. Xiong, Q. Zhu and J. Chu, "Compliance Control for Standing Maintenance of Humanoid Robots Under Unknown External Disturbances," *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong (2014) pp. 2297–2304.
18. S. Hyon, J. G. Hale and G. Cheng, "Full-body compliant human–humanoid interaction: Balancing in the presence of unknown external forces," *IEEE Trans. Robot.* **23**(5), 884–898 (2007).
19. B. J. Stephens and C. G. Atkeson, "Dynamic Balance Force Control for Compliant Humanoid Robots," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei (2010) pp. 1248–1255.
20. Y.-J. Kim, J.-Y. Lee and J.-J. Lee, "A force-resisting balance control strategy for a walking biped robot under an unknown, continuous force," *Robotica*, 1–22 (2014).
21. C. Ott, M. A. Roa and G. Hirzinger, "Posture and Balance Control for Biped Robots Based on Contact Force Optimization," *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, Bled, Slovenia (2011). pp. 26–33.
22. B. Henze, C. Ott and M. A. Roa, "Posture and Balance Control for Humanoid Robots in Multi-contact Scenarios Based on Model Predictive Control," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL (2014) pp. 3253–3258.
23. S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa, "Resolved Momentum Control: Humanoid Motion Planning Based on the Linear and Angular Momentum," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1642 (2003) pp. 1644–1650.
24. A. Macchietto, V. Zordan and C. R. Shelton, "Momentum control for balance," *ACM Trans. Graph.* **28**(3), 80–90 (2009).
25. S.-H. Lee and A. Goswami, "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground," *Auton. Robot.* **33**(4), 399–414 (2012).
26. A. Herzog, L. Righetti, F. Grimminger, P. Pastor and S. Schaal, "Momentum-based balance control for torque-controlled humanoids," technical report, Computing Research Repository, arXiv preprint abs/1305.2042 (2013) pp. 1–7.
27. X. Chen, Q. Huang, Z. Yu and Y. Lu, "Robust push recovery by whole-body dynamics control with extremal accelerations," *Robotica* 32(03), 467–476 (2014).
28. V. Barasuol, J. Buchli, C. Semini, M. Frigerio, E. R. De Pieri and D. G. Caldwell, "A Reactive Controller Framework for Quadrupedal Locomotion on Challenging Terrain," *IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany (2013) pp. 2554–2561.
29. C. Maufroy, H. Kimura and K. Takase, "Stable Dynamic Walking of a Quadruped via Phase Modulations Against Small Disturbances," *IEEE International Conference on Robotics and Automation*, ICRA'09. (2009) pp. 4201–4206.
30. I. Havoutis, C. Semini, J. Buchli and D. G. Caldwell, "Quadrupedal Trotting with Active Compliance," *Proceedings of IEEE International Conference on Mechatronics*, IEEE, Vicenza, Italy (2013) pp. 610–616.
31. C. Gehring, S. Coros, M. Hutter, M. Bloesch, M. A. Hoepflinger and R. Siegwart, "Control of Dynamic Gaits for a Quadrupedal Robot," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Karlsruhe, Germany (2013) pp. 3287–3292.

32. S. A. A. Moosavian and E. Papadopoulos, “Explicit dynamics of space free-flyers with multiple manipulators via SPACEMAPLE,” *Adv. Robot*. **18**(2), 223–244 (2004).
33. M. Mistry, J. Buchli and S. Schaal, “Inverse Dynamics Control of Floating Base Systems Using Orthogonal Decomposition,” *Proceedings of IEEE International Conference on Robotics and Automation*, (2010) pp. 3406–3412.
34. F. Aghili, “A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: Applications to control and simulation,” *IEEE Trans. Robot*. **21**(5), 834–849 (2005).
35. M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, C. D. Remy, and R. Siegwart, “StarlETH: A Compliant Quadrupedal Robot for Fast, Efficient, and Versatile Locomotion,” **In**: *15th International Conference on Climbing and Walking Robot*, USA (2012).

https://doi.org/10.1017/S0263574716000394 Published online by Cambridge University Press