JCAMECH
Vol. 46, No. 2, July 2015, pp 205- 220

Dynamics modeling and stable gait planning of a quadruped robot in walking over uneven terrains

Mahdi Khorram* and S. Ali A. Moosavian

Center of Excellence in Robotics and Control Advanced Robotics and Automated Systems Lab, Dept of Mech Eng, K. N. Toosi Univ of Tech, Tehran, Iran.

Received: 11 July 2015; Accepted: 16 August 2015

Abstract

Quadruped robots have unique capabilities for motion over uneven natural environments. This article presents a stable gait for a quadruped robot in such motions and discusses the inverse-dynamics control scheme to follow the planned gait. First, an explicit dynamics model will be developed using a novel constraint elimination method for an 18-DOF quadruped robot. Thereafter, an inverse-dynamics control will be introduced using this model. Next, a dynamically stable condition under sufficient friction assumption for the motion of the robot on uneven terrains will be obtained. Satisfaction of this condition assures that the robot does not tip over all the support polygon edges. Based on this stability condition, a constrained optimization problem is defined to compute a stable and smooth center of gravity (COG) path. The main feature of the COG path is that the height of the robot can be adjusted to follow the terrain. Then, a path generation algorithm for tip of the swing legs will be developed. This smooth path is planned so that any collision with the environment is avoided. Finally, the effectiveness of the proposed method will be verified.

Keywords: *constraint elimination method, dynamics modeling, dynamic stability, inverse-dynamics control, quadruped robot, uneven terrains.*

1. Introduction

Legged robots have attracted considerable attention in recent decades owing to their interesting potentials. One of the main advantages of these robots is the unique characteristics in traversing uneven environments and overcoming obstacles while wheeled mobile and tracked robots may not be able to move in such conditions. Thus, there are some complicated issues that are associated with the development of legged robots. The challenging problems to achieve a dexterous and versatile legged robot are preserving the robot balance, the high number of degrees of freedom, and the under-actuated nature.

One of the main issues in the field of legged robots is the assurance of stability (balance) for motion on even and uneven terrains. For instance, a quadruped robot can continue to move towards its target as long as the robot remains stable. Thus far, many quadruped robots have been built [1-3], while only a few

∗ Corresponding Author, Tel: + 98 21 8406 3238; Fax: + 98 21 8867 4748, Email: mahdi.khorram@gmail.com

of them have acceptable performance. However, their performance is still not comparable with their biological counterparts. The zero-moment point (ZMP) criterion is a well-known stability criterion that is used in many works to generate a stable motion [4, 5]. However, this criterion in its usual form is limited to the case where the robot walks on even terrains [6-8]. Many attempts have been made to propose a stability criterion for motion on uneven terrains [9-11]. However, most previous studies propose complicated algorithms to measure robot stability instead of proposing an algorithm in order to generate a stable center of gravity (COG) path trajectory.

The path planning problem of legged robots for motion on uneven terrains has been investigated by various researchers. In [12], the ZMP criterion was employed to plan a stable and optimal path for a quadruped robot to walk on uneven terrains. In this method, the ZMP dynamic equations have been used in the COG path generation algorithm. However, the Z-component of ZMP has been ignored, whereas the Z-component of this point is no longer zero in motion over uneven terrains. This assumption over moderately uneven terrains may lead to acceptable results, but this does not guarantee robot stability over severely rough terrains. Zheng et al proposed the contact wrench sum as a dynamic stability criterion on uneven terrains[13]. Computational complexity is the main drawback of this method. The maximum hoop stress (MHS) criterion has been used for stability investigation of wheeled mobile robotic manipulators [14]. This measure is based on stabilizing/ destabilizing moments exerted on the main platform which direct the rotational behavior, and has been effectively used for various wheeled robots. Therefore, one of the main concerns of this paper was to develop a stability condition for quadruped robots moving on uneven terrains.

The design of a model-based controller for a quadruped robot requires having an explicit dynamics model. However, deriving an explicit dynamics model for legged robots is not a simple task as serial robots. This is due to the fact that quadruped robots usually have much higher number of degrees of freedom. Furthermore, such robots are in contact with the environment, which introduces some restrictive constraints to the dynamics model. The conventional methods for deriving the dynamics model are Newton-Euler and Lagrange methods [15]. However, the equations of motion obtained by these methods suffer from high computational complexities and they are not suitable for real-time implementation. Thus, many attempts have been made to tackle this problem. Featherstone presented the spatial notation which reduces the computational complexities of the equations of motion [16]. Moosavian *et al*. proposed the explicit dynamics method based on Lagrangian formulation to obtain the dynamics model for a space robot [17]. In this paper, this method will be used to drive the dynamics equations of a quadruped robot.

Having a free-constraint dynamics model is required to solve the forward or inverse dynamics problems. Actually, contact forces are unknown and should be removed from the equations by considering the kinematics constraints. Consequently, to remove the contact forces from the equations of motion, the constraint elimination problem is solved. The main objective of the proposed methods was to derive two independent equations in terms of the contact forces and joint torques. Mistry et al. presented the orthogonal decomposition method to solve the inverse-dynamics problem without the need to compute the contact forces [18]. Aghili proposed a linear operator to map the dynamics model into a constraint-free space [19]. Righetti *et al*. considered the inverse dynamics problem and through the definition of an optimization algorithm proposed a solution for this problem [20]. In this article, a free-constraint space will be defined with its dimension being equal to the system degrees of freedom (DOFs) minus the number of kinematic constraints. The dynamics equations will be mapped into this space by an operator which is obtained by using the kinematic constraints.

In this paper, a dynamics model for a quadruped robot will first be derived. Then, by considering the kinematics constraints, the contact forces are eliminated from the dynamics equations. Then, an inverse-dynamics controller is introduced based on using the developed dynamics model. A stability condition will be developed to ensure the robot stability on uneven terrains. Then, a path planning algorithm for the center of gravity will be introduced on the basis of the defined condition by using a constrained optimization algorithm. Finally, a smooth path for the tip of the swing legs will be planned. The proposed algorithm will be implemented on a quadruped by simulation, and the obtained results will be discussed.

2. Dynamics modeling

In this section, the model of quadruped robot will first be introduced. Then, an explicit dynamics model will be derived. Finally, the constraint elimination method to obtain the free-constraint equations of motion will be introduced.

2.1. The model of quadruped robot

To model a quadruped robot, a rectangular base was used as the main body with four legs attached to its corners. This robot is shown in Figure 2. Each leg consists of two rigid links connected to each other with revolute joints. The main body has six passive degrees of freedom: three translational and three rotational. Three degrees of freedom were chosen for each leg, in order to increase the workspace of each leg and consequently the mobility and kinematic reachability of the robot. Thus, each leg has the ability to place its foot anywhere in 3D space. There are two revolute joints in the hip, one along the roll axis and the other along the pitch axis and one revolute joint in the knee along the pitch axis. Thus, the robot has nine rigid-body links with eighteen degrees of freedom. It was assumed that the contact between the stance legs and the ground occurs at a point. In addition, the friction between the stance legs and the ground is assumed to be large enough, prohibiting any slippage.

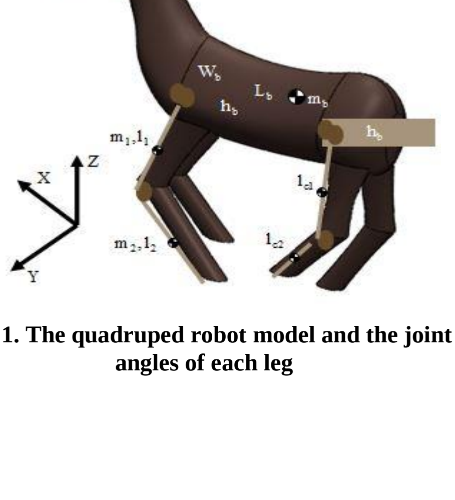

Fig. 1. The quadruped robot model and the joint angles of each leg

2.2. Dynamics model

To obtain full dynamics equations of the robot, the explicit dynamics method is employed. On the basis of the defined model, the robot configuration can be defined as:

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

where $\mathbf{q}_B \in \mathbb{R}^{6\times 1}$ is the vector of the position and orientation of the frame is attached to the main body with respect to the world frame. Also, $\mathbf{q}_L \in \mathbb{R}^n$ denotes the joint angles of all the legs. The dynamics equations of the robot can be written as:

$$M\left(q\right)\ddot{q} + V\left(q,\dot{q}\right) + G\left(q\right) = B\tau + J^T F_{Leg} \tag{2}$$

where $\mathbf{M(q)}$ denotes the mass matrix, $\mathbf{V(q,\dot{q})}$ is the Coriolis and centrifugal forces, $\mathbf{G(q)}$ represents the gravitational forces. In addition, $\mathbf{B}$ is the under-actuation matrix which its elements are one for the actuated joints, that is, the joints of all the legs and are zero for the virtual joints of the main body, $\boldsymbol{\tau}$ defines the joint torques, $\mathbf{J}$ represents the Jacobian matrix for the contact positions and $\mathbf{F_{Leg}}$ is the contact forces applied on the robot at the contact points. Due to the contact of the stance legs with the environment, the last term was added to the dynamics equations. All terms of the dynamics equations can easily be calculated using the formulations presented in [17], and are detailed in Appendix A. Since the kinetic energy of each rigid link is divided into three different terms and each term is differentiated separately and there after summed to compute the dynamics terms, the resultant dynamics equations are very computationally efficient.

2.3. Constraint elimination method

When trying to control the robot or calculate the joint torques required to perform a specific maneuver of the main body, the exact values of contact forces should be known. There are two solutions for this problem: (1) measuring the contact forces directly and (2) elimination of the contact forces from the dynamics equations. The first method does not yield precise results, because the output of force sensors is a noisy signal which leads to inaccurate values. However, in the second method, the contact forces are removed from the dynamics equations using the kinematics constraints. The second method will be used to derive free-constraint dynamics equations subsequently.

Given that legged robots are in contact with the environment, some constraints are imposed on the dynamics equations. If it is assumed that the stance legs do not slip during the robot motion, this requires having sufficient friction between the stance legs and the ground, and then, the velocities of the tip of stance legs become zero. Thus, the kinematic constraints can be defined as:

$$V_{tip,i} = O_{3p\times 1} \quad for\ i = 1,...,\ if \quad i = s\tan ce\ leg \tag{3}$$

The above equation can be expressed in the Jacobian form as:

$$J\dot{q} = O_{3p\times 1} \tag{4}$$

where p denotes the number of stance legs.

In the following, these constraints will be exploited to obtain the free-constraint of dynamics equations from Equation 1. In other words, the contact forces from the dynamics equations will be eliminated. In doing this, a new space is defined, called the independent space. If k constraints are applied on the robot through the contact of the stance legs with the ground, the dimension of this space will be $n+6-k$. The variables of this space consist of the position and orientation of the main body and also the joint angles of the swing legs. The reason for the selection of such space is that the control of the variables of the independent space guarantees that the whole configurations of the robot will be controlled as long as the stance legs remain stationary during the robot motion. Let us define this space as:

$$\beta = \begin{bmatrix} q_{B}^{T} & q_{SL}^{T} \end{bmatrix}^{T} \tag{5}$$

where $\mathbf{q}_{SL}$ denotes the joint angles of the swing legs. At present, we want to rewrite the dynamics equations in terms of $\boldsymbol{\beta}$ such that the contact forces are removed from the dynamics equations. The relation between the whole configuration and the variables of the independent space can be expressed as:

$$\dot{q} = S\dot{\beta} \tag{6}$$

where $\mathbf{S}$ is a matrix which maps the robot configuration space onto the independent space. This matrix is defined based on the kinematic constraints. It can be stated as follow:

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

where $\mathbf{F_{1,i}}$ and $\mathbf{F_{2,i}}$ can be given as:

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

$\mathbf{J_{b,i}}$ and $\mathbf{J_{L,i}}$ are the Jacobian matrices of the main body and the i^{th} swing leg, respectively, which 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 i^{th} stance leg. Since the robot has the under-actuated structure due to its floating main body, the whole joints were partitioned into two components: the under-actuated and the actuated. This partition was performed to eliminate the contact forces from the dynamics equations. In order for this to be done, the whole joint velocities were divided based on being actuated or under-actuated. Thus, Equation 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 subscript "a" means the actuated joints and subscript "ua" represents the under-actuated joints. Time derivate of Equation 10 yields:

$$\ddot{\mathbf{q}}_{ua} = \mathbf{S}_{ua}\ddot{\boldsymbol{\beta}} + \dot{\mathbf{S}}_{ua}\dot{\boldsymbol{\beta}}, \ddot{\mathbf{q}}_{a} = \mathbf{S}_{a}\ddot{\boldsymbol{\beta}} + \dot{\mathbf{S}}_{a}\dot{\boldsymbol{\beta}} \tag{11}$$

In the following equations, the dynamics equations were partitioned into the actuated and the under-actuated components as follows:

$$\begin{gathered} \begin{bmatrix} \mathbf{M}_{1ua} & \mathbf{M}_{2ua} \\ \mathbf{M}_{2a} & \mathbf{M}_{2a} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_{ua} \\ \ddot{\mathbf{q}}_{a} \end{bmatrix} + \begin{bmatrix} \mathbf{V}_{ua} \\ \mathbf{V}_{a} \end{bmatrix} + \begin{bmatrix} \mathbf{G}_{ua} \\ \mathbf{G}_{a} \end{bmatrix} = \\ \begin{bmatrix} \mathbf{O}_{6\times1} \\ \boldsymbol{\tau}_{\mathbf{a}} \end{bmatrix} + \begin{bmatrix} \mathbf{J}_{ua}^{T} \\ \mathbf{J}_{a}^{T} \end{bmatrix} \mathbf{F_{Leg}} \end{gathered} \tag{12}$$

If the above equations were rewritten to obtain two independent equations, then we have:

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

By multiplying Equation 13 a by $\mathbf{S}_{ua}^{T}$ and Equation 13-b by $\mathbf{S}_{a}^{T}$ and the summation of the resultant equations, we have:

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

Based on the definition of the $\mathbf{S}_{a}$ and $\mathbf{S}_{ua}$, we can easily prove that $\mathbf{J}_{a}\mathbf{S}_{a} + \mathbf{J}_{ua}\mathbf{S}_{ua} = \mathbf{O}$. Thus, we have:

$$\begin{aligned} &\mathrm{M}_{ua}\ddot{\mathrm{q}}_{ua} + \mathrm{M}_{a}\ddot{\mathrm{q}}_{a} + \mathrm{S}_{ua}^{T}\mathrm{V}_{ua} + \mathrm{S}_{a}^{T}\mathrm{V}_{a} \\ &+\mathrm{S}_{ua}^{T}\mathrm{G}_{ua} + \mathrm{S}_{a}^{T}\mathrm{G}_{a} = \mathrm{S}_{a}^{T}\tau \end{aligned} \qquad (15)$$

As shown in the above equation, the contact forces are eliminated from the dynamics equations. Now, we want to map the above equation into the independent space. By substituting Equation 11 in Equation 15, we have:

$$\mathrm{M}_{\beta}\ddot{\beta} + \mathrm{V}_{\beta} + \mathrm{G}_{\beta} = \mathrm{S}_{a}^{T}\tau \qquad (16)$$

where

$$\begin{aligned} &\mathrm{M}_{\beta} = \mathrm{M}_{ua}\mathrm{S}_{ua} + \mathrm{M}_{a}\mathrm{S}_{a} \\ &\mathrm{V}_{\beta} = \mathrm{S}_{ua}^{T}\mathrm{V}_{ua} + \mathrm{S}_{a}^{T}\mathrm{V}_{a} + \mathrm{M}_{ua}\dot{\mathrm{S}}_{ua}\dot{\beta} + \mathrm{M}_{a}\dot{\mathrm{S}}_{a}\dot{\beta} \qquad (17) \\ &\mathrm{G}_{\beta} = \mathrm{S}_{ua}^{T}\mathrm{G}_{ua} + \mathrm{S}_{a}^{T}\mathrm{G}_{a} \end{aligned}$$

Equation 16 introduces an inverse dynamics controller for a constrained quadruped robot. Joint torques can be calculated by the Moore–Penrose pseudo inverse of $\mathbf{S}_{a}^{T}$, when the joint angles, velocities and accelerations of the independent variables are known This equation allows us to calculate the joint torques without the need to compute the contact forces and also the inversion of mass matrix. In addition, the contact forces can be calculated as follow:

$$\begin{aligned} &\mathrm{F}_{\mathrm{Leg}} = \left(\mathrm{J}_{ua}^{T}\right)^{\#} \\ &\left(\mathrm{M}_{1ua}\mathrm{S}_{ua}\ddot{\beta} + \mathrm{M}_{2ua}\mathrm{S}_{a}\ddot{\beta} + \mathrm{M}_{1ua}\dot{\mathrm{S}}_{ua}\dot{\beta} + \mathrm{M}_{2ua}\dot{\mathrm{S}}_{a}\dot{\beta} + \mathrm{V}_{ua} + \mathrm{G}_{ua}\right) \end{aligned} \qquad (18)$$

3. Stability condition of quadruped robots over uneven terrains

A quadruped robot should remain stable in motion over even and uneven terrains in order to avoid falling down. Thus, a particular path should be designed for the COG. In order to design a particular path for the COG, a proper condition should be developed to guarantee robot stability. Since the focus of the current paper is on the robot motion over uneven terrains, first, a stability condition to guarantee robot stability will be proposed. This condition yields a stable motion over uneven terrain under sufficient friction assumption between the stance legs and the environment. The main feature of this condition is that the robot can vary its height to follow the terrain. In other words, the height of main body does not remain constant when compared with conventional stability criteria. On the other hand, the motion of the main body along z-axis may help the robot to keep its balance, especially over uneven terrains.

Conventional stability conditions yields acceptable results in cases when the robot moves on an even terrain or the height of the main body during its motion is constant. For instance, the COG path can be generated for motion on a horizontal surface based on the ZMP [21], because this point is defined on the horizontal surface. Whereas, for uneven terrains, the ZMP should stay inside the support polygon which is no longer coincident with the horizontal ground [9]. Therefore, a stability condition needs to be introduced for uneven terrains. Here, the reasons for robot instability are divided in general into the rotation about the edges of support polygon and the slippage of the stance legs. Moments around the edges of support polygon is taken as the main cause of instability, since it is assumed that there is sufficient friction between the stance legs and the ground. Thus, the robot should move such that the tumbling moments produced by the external forces, about all edges of the support polygon, would hold the contact between the stance legs and the ground. A point-mass model is taken into consideration to reduce computational complexities in the derivation procedure of obtaining a stability condition. In other words, it is assumed that the masses of all legs are concentrated in a point called COG. Since the masses of all the legs are ignored, the gravitational and inertial forces of the main body are the main forces which influence robot stability.

The edges of support polygon should be formulated in the first step, because the edges of support polygon are the key variables to determine robot stability. If the positions of stance legs are defined as $\mathbf{P}_{STL1}, \mathbf{P}_{STL2}, ..., \mathbf{P}_{STLn}$, the unit vectors of the edges of support polygon can be defined as:

$$\kappa_i = \frac{\mathrm{P}^{STL}_{i+1} - \mathrm{P}^{STL}_{i}}{\left\| \mathrm{P}^{STL}_{i+1} - \mathrm{P}^{STL}_{i} \right\|} \quad i = 1 : n - 1 \ \ clockwise$$
$$\kappa_n = \frac{\mathrm{P}^{STL}_{1} - \mathrm{P}^{STL}_{n}}{\left\| \mathrm{P}^{STL}_{1} - \mathrm{P}^{STL}_{n} \right\|} \qquad (19)$$

The sequence of two contact points for calculating the unit vector should be chosen such that this vector goes around the support polygon in a clockwise direction. A quadruped robot in motion on uneven terrains remains stable if the tumbling moments about all edges of the support polygon do not tend to separate the contact between the stance legs and the ground. In other words, the following condition must be satisfied for $t > 0$

$$\kappa_{\mathrm{i}}^{\mathrm{T}} \mathrm{M}_{\mathrm{i}} > \varepsilon \quad \text{for all i and } \varepsilon > 0 \qquad (20)$$

where $\mathbf{M_i}$ denotes the moment of external forces (that is, inertial and gravitational forces) about the i^{th} edge and ε stands for the stability margin. The above inequality equation can be expressed in terms of the COG accelerations and the variables of the edges of support polygon for the i^{th} edge of support polygon as follow:

$$\kappa_i^T \mathrm{M}_i = \begin{vmatrix} \kappa_i^x & \kappa_i^y & \kappa_i^z \\ P_G^x - p_i^x & P_G^y - p_i^y & P_G^z - p_i^z \\ -m\ddot{P}_G^x & -m\ddot{P}_G^y & -m(g + \ddot{P}_G^z) \end{vmatrix} > \varepsilon \qquad (21)$$

where p_i^x , p_i^y and p_i^z indicate the position of an arbitrary point on the i^{th} edge of support polygon. In addition, P_G^x, P_G^y and P_G^z indicate the position of the COG expressed in the world frame. As such, it is assumed that the gravitational acceleration only points along negative Z-axis. To guarantee robot stability, the COG path must be planned so that the moments about all edges of support polygon have positive values during the robot motion. Since there is no assumption about the type of the terrain, this stability condition can be used to determine robot stability over uneven terrains. Since all tumbling moments should be positive to maintain robot stability, the stability index is defined as the minimum value of tumbling moments about all edges of the support polygon. For instance, when the leg 4 is in the swing phase, it can be expressed as:

$$M_{stab} = \min\left\{ M^{l_{12}}_{stab}, M^{l_{23}}_{stab}, M^{l_{32}}_{stab} \right\} \qquad (22)$$

where, for instance, $M^{l_{12}}_{stab}$ is given as:

$$M^{l_{12}}_{stab} = \kappa_{12}^T \mathrm{M}_{12}$$
$$\kappa_{12} = \frac{\mathrm{P}^{STL}_{1} - \mathrm{P}^{STL}_{2}}{\left\| \mathrm{P}^{STL}_{2} - \mathrm{P}^{STL}_{1} \right\|} \qquad (23)$$

To guarantee robot stability, the stability index should be positive during the robot motion. The stability condition, that is, Equation 21, which is expressed as an inequality equation, only describes robot stability at certain instant of the time when all motion parameters of the robot motion are known. However, the main objective is to generate a stable COG path by using this equation. Thus, a procedure should be introduced to plan the COG path, so that the stability condition is satisfied. The procedure for obtaining the stable COG path based on this stability condition will be discussed in the following.

4. Stable COG path planning on uneven terrains

Here, an algorithm for generating a stable COG path based on the stability condition will be proposed. In the real-time path planning, the COG path is generated for each step of a single gait before the robot starts the step. An optimization problem is defined to calculate the COG path since the stability condition is expressed as an inequality equation and the stability condition is considered as a nonlinear and inequality constraint of the optimization. In addition, the COG path should satisfy smoothness conditions especially at the instant of switching swing leg. Furthermore, the COG path should cross through the predefined points at the start and the end of each step within a single gait. To guarantee smoothness conditions of the COG path within a gait, the robot goes to a four-leg support phase at the start, middle and the end of a cycle. Thus, the middle of a walking cycle means that the instant in which all legs of one side performed their motions, the first leg of the other side should start its motion.

Using a four-leg support phase is a common method to guarantee robot stability by enlarging the support polygon used particularly at the instant in which the transferring leg is the front leg and the next transferring leg is the rear leg [12, 22]. This is due to the fact that the triangles of the stability (which its vertexes are the tip of stance legs), change drastically at the instant, and the use of this phase helps the robot to change the COG path smoothly to start the next step stability by increasing the support polygon. In this paper, the robot uses the wave gait for walking over uneven terrains. This gait is an optimal pattern of leg lifting in terms of stability and velocity [23, 24], which is motivated by the locomotion of animals. However, the four-leg support phase may not been seen in mammal locomotion. This difference is due to the simplifications which are made to model a quadruped mammal.

To obtain the stable COG path, an eighth-order polynomial function of the time is chosen for each step of a gait and along each direction. The trajectory equation for the j^{th} step of a single gait is defined as:

$$\begin{aligned}&\mathrm{P}_j^G(t) = \mathrm{a}_j^1 t^8 + \mathrm{a}_j^2 t^7 + \mathrm{a}_j^3 t^6 + \mathrm{a}_j^4 t^5 + \mathrm{a}_j^5 t^4\\&+\mathrm{a}_j^6 t^3 + \mathrm{a}_j^7 t^2 + \mathrm{a}_j^8 t + \mathrm{a}_j^9 \quad t_j^s \le t \le t_j^f\end{aligned} \tag{24}$$

where t_j^s and t_j^f mean the start and the end times of the j^{th} step, respectively. The coefficients of the polynomials will be computed based on the smoothness and stability conditions.

The COG path should pass through the prescribed points determined by the footstep planning algorithm. Since these points are only computed based on the reachability and terrain adaptability conditions, they may be inappropriate in terms of robot stability. Therefore, we attempt to obtain the closest possible points to the desired ones. For this purpose, the errors between the desired and resultant COG positions at the start and end of each step within a single gait are chosen as the cost function of the optimization. Since the path should be continuous at the instants of switching swing leg at the levels of position, velocity and acceleration, these conditions impose some constraints on the optimization problem. Furthermore, the stability condition for each step of a single gait is taken as a nonlinear and inequality constraint of the optimization problem. The optimization problem for computing the COG path is defined as:

$$\begin{aligned}&\min_{\mathbf{a}_j} \quad \sum_{j=1}^{6}\left(\mathbf{P}_j^G(t_{j+1}) - \mathbf{P}_{des}^G(t_{j+1})\right)^T \mathbf{W}\left(\mathbf{P}_j^G(t_{j+1}) - \mathbf{P}_{des}^G(t_{j+1})\right)\\&\text{subject to } \boldsymbol{\kappa}_i^T \mathbf{M}_i > 0 \text{ about all edges of the support polygon}\\&\text{and for all steps of a gait}\\&\mathbf{P}_j^G(t_{j+1}) = \mathbf{P}_{j+1}^G(t_{i+1}), \dot{\mathbf{P}}_j^G(t_{j+1}) = \dot{\mathbf{P}}_{j+1}^G(t_{i+1}), \dot{\mathbf{P}}_j^G(t_{j+1}) = \dot{\mathbf{P}}_{j+1}^G(t_{i+1}) \text{ j=1,..,7}\\&\mathbf{P}_0^G(t_1) = \mathbf{P}_0^G, \dot{\mathbf{P}}_0^G(t_1) = \dot{\mathbf{P}}_0^G, \ddot{\mathbf{P}}_0^G(t_1) = \ddot{\mathbf{P}}_0^G\\&\mathbf{P}_7^G(t_8) = \mathbf{P}_f^G, \dot{\mathbf{P}}_7^G(t_8) = \dot{\mathbf{P}}_f^G, \ddot{\mathbf{P}}_7^G(t_8) = \ddot{\mathbf{P}}_j^G\end{aligned} \tag{25}$$

The solution of the above optimization problem yields the coefficients of the defined polynomials. After that, the smooth and stable COG path can be calculated using Equation 24. Here, a single gait is divided into seven segments. Therefore, six intermediate points are chosen to define the cost function. The algorithm for the calculation of the COG path is summarized in Figure 2.

5. Tip of swing leg path planning

In this section, an appropriate path for the tip of swing legs will be planned. In the path generation of the tip of swing legs, obstacle avoidance and smoothness conditions will be taken into account. In other words, the main objective of tip of swing leg path generation algorithm is to obtain a smooth path without any collision with the environment. In order for this to be done, the path of the tip of each swing leg is divided into *N-1* segments and a third-order polynomial function of time is selected for each segment. It is assumed that time interval of $\left[t_s^i, t_f^i\right]$, which is the i^{th} leg is in the swing phase, and is divided into $N-1$ segments. Through the selection of a third-order polynomial function of the time for each segment, the trajectory equation of the j^{th} segment of the i^{th} leg, for instance, is stated as:

$$\mathrm{P}^{i}_{swl,j}(t) = {}^{i}\mathrm{a}^{3}_{swl,j}(t-t^{i}_{s,j})^{3} + {}^{i}\mathrm{a}^{2}_{swl,j}(t-t^{i}_{s,j})^{2} + {}^{i}\mathrm{a}^{1}_{swl,j}(t-t^{i}_{s,j}) + {}^{i}\mathrm{a}^{0}_{swl,j} \qquad (26)$$

$$if \;\; t^{i}_{s,j} < t < t^{i}_{f,j} \quad for \;\; j = 1,\dots,N-1$$

where

$${}^{i}\mathbf{a}^{m}_{swl,j} = \left[\,{}^{i}a^{m,x}_{swl,j} \quad {}^{i}a^{m,y}_{swl,j} \quad {}^{i}a^{m,z}_{swl,j}\,\right]^{T} \quad m = 0,\dots,3 \qquad (27)$$

The trajectory equation for all segments of i^{th} swing leg during a single step can be defined as:

$$\mathrm{P}^{i}_{swl}(t) = \begin{cases} \mathrm{P}^{i}_{swl,1}(t) & t^{i}_{s,1} < t < t^{i}_{f,1} \\ \vdots & \\ \mathrm{P}^{i}_{swl,N-1}(t) & t^{i}_{s,N-1} < t < t^{i}_{f,N-1} \end{cases} \qquad (28)$$

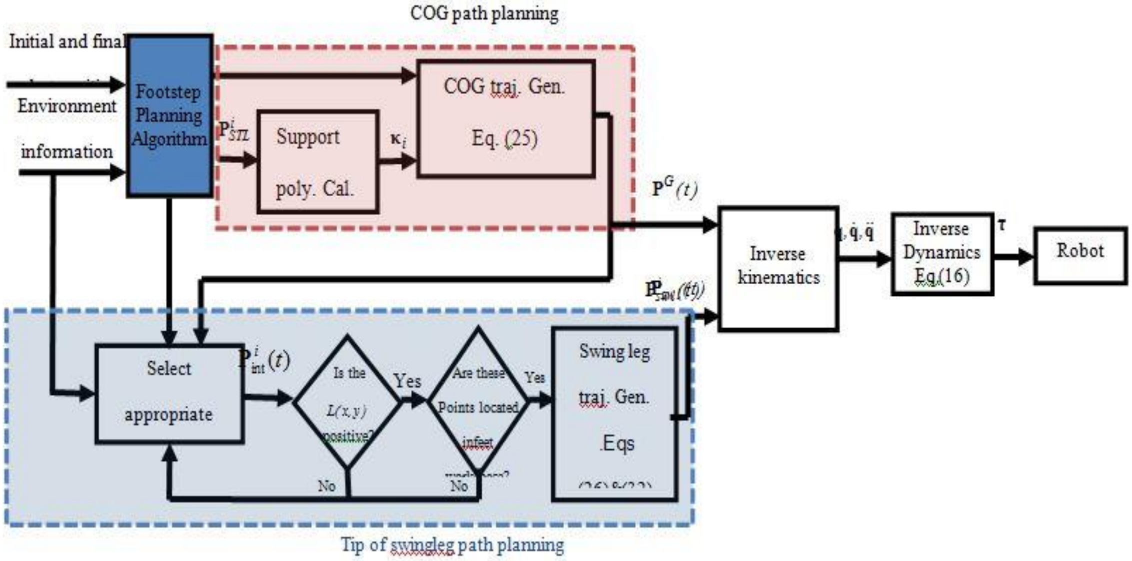

Fig. 2. Stable COG and tip of swing leg path planning algorithms for walking on uneven terrains

To calculate the coefficients of each swing leg trajectory equation, appropriate conditions should be defined. As the first stage, the proper intermediate points are selected between the lifting and landing positions. These points are selected based on avoiding any collision with the environment. The workspace for each swing leg can be calculated easily, because the COG path has been calculated in the previous section. The intermediate points for each swing leg will be chosen from an area, which is the intersection of the workspace of that leg and the corresponding free-collision area was calculated through the geometry equation of the environment which is assumed to be known. These points can be defined as:

$$\mathrm{P}^{i}_{\mathrm{int}}(t) = \left\{ \mathrm{X} \middle| H(\mathrm{X}) > 0, \mathrm{X} \in FW^{i} \right\} \quad j = 1:N-2 \qquad (29)$$

where FW^{i} denotes the workspace of the i^{th} swing leg and $H(\mathbf{X})$ is the terrain geometry equation. Since *N-1* polynomial functions are selected for the path of tip of each swing leg, some additional restrictions should be added into the problem to obtain a smooth path. One of these restrictions is the smoothness condition. Since the path of the tip of each swing leg is a piecewise continuous function of the time, in order to obtain a smooth path, it should be continuous in the levels of position, velocity and acceleration at the instant of switching swing leg. These conditions for the i^{th} swing leg can be defined as:

$$\begin{aligned} \mathrm{P}^{i}_{swl,j}(t^{i}_{f,j}) &= \mathrm{P}^{i}_{swl,j+1}(t^{i}_{s,j+1}) \\ \dot{\mathrm{P}}^{i}_{swl,j}(t^{i}_{f,j}) &= \dot{\mathrm{P}}^{i}_{swl,j+1}(t^{i}_{s,j+1}) \qquad j = 1,\dots,N-1 \\ \ddot{\mathrm{P}}^{i}_{swl,j}(t^{i}_{f,j}) &= \ddot{\mathrm{P}}^{i}_{swl,j+1}(t^{i}_{s,j+1}) \end{aligned} \qquad (30)$$

Furthermore, the path must be planned such that each swing leg begins its motion from the current location and reaches to the target within a single gait and also it should pass through the intermediate points. These conditions are defined as:

$$\begin{aligned} &\mathrm{P}^{i}_{swl,1}(t^{i}_{s,1}) = \mathrm{P}^{i}_{swl,0}, \mathrm{P}^{i}_{swl,(N-1)}(t^{i}_{f,N-1}) = \mathrm{P}^{i}_{swl,f} \\ &\dot{\mathrm{P}}^{i}_{swl,1}(t^{i}_{s,1}) = \dot{\mathrm{P}}^{i}_{SWL}(t_{0}), \dot{\mathrm{P}}^{i}_{swl,(N-1)}(t^{i}_{f,N-1}) = \dot{\mathrm{P}}^{i}_{swl,f} \\ &\mathrm{P}^{i}_{swl,j}(t^{i}_{f,j}) = \mathrm{P}^{i}_{\mathrm{int},j}, \mathrm{P}^{i}_{swl,j+1}(t^{i}_{s,j+1}) = \mathrm{P}^{i}_{\mathrm{int},j} \quad j = 1:\dots:N-2 \end{aligned} \qquad (31)$$

Given that the effect of the impact between the swing legs and the ground is not considered on robot stability in the current study, the initial and final velocities of the tip of each swing leg was set equal to zero. To obtain the coefficients of polynomials and consequently compute the path of the tip of the i^{th} swing leg, Equation 30 and Equation 31 must be solved. Along each direction, the above conditions offer a matrix equation to obtain the unknown coefficients. For instance, z-directional equation for obtaining the coefficients of the i^{th} swing leg trajectory can be defined as:

$${}^{i}\mathrm{H}^{i}\,\mathrm{Z} = {}^{i}\;\mathrm{N} \tag{32}$$

where ${}^{i}\mathbf{X}$ and ${}^{\mathrm{i}}\mathbf{N}$ are given by:

$$\begin{gathered}
{}^{i}\mathbf{Z} = \begin{bmatrix} {}^{i}a^{0,z}_{swl,1} & {}^{i}a^{1,z}_{swl,1} & {}^{i}a^{2,z}_{swl,1} & {}^{i}a^{3,z}_{swl,1} & \cdots & {}^{i}a^{0,z}_{swl,N-1} & {}^{i}a^{1,z}_{swl,N-1} & {}^{i}a^{2,z}_{swl,N-1} & {}^{i}a^{3,z}_{swl,N-1} \end{bmatrix}^{T} \\
{}^{i}\mathbf{N} = \begin{bmatrix} \underset{3(N-2)}{0\cdots0}\; \underbrace{P^{i,z}_{\mathrm{int},1}\cdots P^{i,z}_{\mathrm{int},N-1}}_{N-2}\, \underbrace{P^{i,z}_{swl,0}\; P^{i,z}_{swl,f}\; \dot{P}^{i,z}_{swl,0}\; \dot{P}^{i,z}_{swl,f}}_{4} \end{bmatrix}^{T}
\end{gathered} \tag{33}$$

The matrix of ${}^{i}\mathbf{H}$ is calculated as follow:

$$\begin{gathered}
{}^{i}\mathbf{H} = \begin{bmatrix} \mathbf{A}_{C1}{}^{T} & \mathbf{A}_{C2}{}^{T} & \mathbf{A}_{C3}{}^{T} & \mathbf{A}_{C4}{}^{T} & \mathbf{A}_{C5}{}^{T} \end{bmatrix}^{T}_{4(N-1)\times4(N-1)} \\
\mathbf{A}_{C1} = \begin{bmatrix} \mathbf{F}pt_1 & 0 & & \cdots & & \mathbf{O}_{1\times3} \\ \mathbf{O}_{1\times4} & \mathbf{F}pt_2 & 0 & & \cdots & \mathbf{O}_{1\times3} \\ \mathbf{O}_{1\times4} & \mathbf{O}_{1\times4} & \mathbf{F}pt_3 & 0 & \cdots & \mathbf{O}_{1\times3} \\ \vdots & & & \ddots & & \vdots \\ \mathbf{O}_{1\times4} & & \cdots & 0 & \mathbf{F}pt_{N-1} & \mathbf{O}_{1\times3} \end{bmatrix}_{(N-2)\times4(N-1)}, \mathbf{A}_{C2} = \begin{bmatrix} \mathbf{F}vt_1 & 0 & & \cdots & & \mathbf{O}_{1\times2} \\ \mathbf{O}_{1\times4} & \mathbf{F}vt_2 & 0 & & \cdots & \mathbf{O}_{1\times2} \\ \mathbf{O}_{1\times4} & \mathbf{O}_{1\times4} & \mathbf{F}vt_3 & 0 & \cdots & \mathbf{O}_{1\times2} \\ \vdots & & & \ddots & & \vdots \\ \mathbf{O}_{1\times4} & & \cdots & 0 & \mathbf{F}vt_{N-1} & \mathbf{O}_{1\times2} \end{bmatrix}_{(N-2)\times4(N-1)} \\
\mathbf{A}_{C3} = \begin{bmatrix} \mathbf{F}at_1 & 0 & & \cdots & & 0 \\ \mathbf{O}_{1\times4} & \mathbf{F}at_2 & 0 & & \cdots & 0 \\ \mathbf{O}_{1\times4} & \mathbf{O}_{1\times4} & \mathbf{F}at_3 & 0 & \cdots & 0 \\ \vdots & & & \ddots & & \vdots \\ \mathbf{O}_{1\times4} & & \cdots & 0 & \mathbf{F}at_{N-1} & 0 \end{bmatrix}_{(N-2)\times4(N-1)}, \mathbf{A}_{C4} = \begin{bmatrix} \mathbf{O}_{1\times4} & \mathbf{F}pct_1 & 0 & \cdots & & \mathbf{O}_{1\times4} \\ \mathbf{O}_{1\times4} & \mathbf{O}_{1\times4} & \mathbf{F}pct_2 & 0 & \cdots & 0 \\ \mathbf{O}_{1\times4} & & & \ddots & \cdots & \vdots \\ \vdots & & & & & \mathbf{O}_{1\times4} \\ \mathbf{O}_{1\times4} & & \cdots & 0 & & \mathbf{F}pct_{N-1} \end{bmatrix}_{(N-2)\times4(N-1)} \\
\mathbf{A}_{C5} = \begin{bmatrix} 1 & 0 & & \cdots & & & 0 \\ 0 & \cdots & 0 & 1 & (t_N - t_{N-1}) & (t_N - t_{N-1})^2 & (t_N - t_{N-1})^3 \\ 0 & 1 & 0 & & & \cdots & 0 \\ 0 & \cdots & 0 & 1 & & 2(t_N - t_{N-1}) & 3(t_N - t_{N-1})^2 \end{bmatrix}_{4\times4(N-1)} \\
\mathbf{F}pt_i = \begin{bmatrix} 1 & (t_i - t_{i-1}) & (t_i - t_{i-1})^2 & (t_i - t_{i-1})^3 & -1 \end{bmatrix}, \mathbf{F}vt_i = \begin{bmatrix} 0 & 1 & 2(t_i - t_{i-1}) & 3(t_i - t_{i-1})^2 & 0 & -1 \end{bmatrix} \\
\mathbf{F}at_i = \begin{bmatrix} 0 & 0 & 2 & 6(t_i - t_{i-1}) & 0 & 0 & -1 \end{bmatrix}, \mathbf{F}pct_i = [1 \;\; 0 \;\; 0 \;\; 0]
\end{gathered} \tag{34}$$

The tip of swing leg path planning algorithm is shown in Figure 2. First, proper intermediate points are chosen for each swing leg. If these points are placed inside the workspace of that leg and also not located inside the environment, they are selected as the intermediate points. Then, the trajectory equation for tip of each swing leg can be calculated by Equation 26 and Equation 32.

6. Obtained results

In order to evaluate the proposed COG and tip of swing leg path planning algorithms, a quadruped robot in the simulation in walking on an uneven terrain was used for the test. The simulation was performed with the assumption of sufficient friction between the stance legs and the ground. As mentioned earlier, a walk gait was used to move on uneven terrain. The sequence of the leg lifting is the right hind, right front, left hind and left front. The robot starts its motion with a four-leg support phase to be prepared for lifting its leg. Then, the rear and front legs of one side go to the swing phase according to the leg lifting sequence. A four-leg support phase also is chosen at the middle of a cycle. Then, the legs of the other side go to the swing phase. The robot finishes its motion at the end of a cycle with a four-leg support phase.

Since one of the main contributions of this paper is to walk on uneven natural terrains, the uneven terrain must be modeled in the first step. The uneven terrain, which is considered in this paper, is shown in Figure 3. The color circles represent the footprints of all legs when the robot motion form was in their initial location to their final location. The green, red, blue and yellow circles show the footholds of the left hind, right hind, left front and right front legs, respectively. Here, it is assumed that an expert user decides about the footprints on the terrain by considering the geometry of the terrain and also the physical properties of the robot. As seen, when the robot walks on this terrain, footprints are located on non-coplanar surfaces. In other words, the height of the footprints is not the same. Thus, conventional stability conditions do not lead to accurate results in this case. To simplify the problem, the y-component of the footprints is chosen to be the same. To plan a path for the robot, the specifications of the robot and some essential COG path parameters are gathered as show in Table . The geometric and mass properties of the robot are similar to the Starl *ETH* [25].

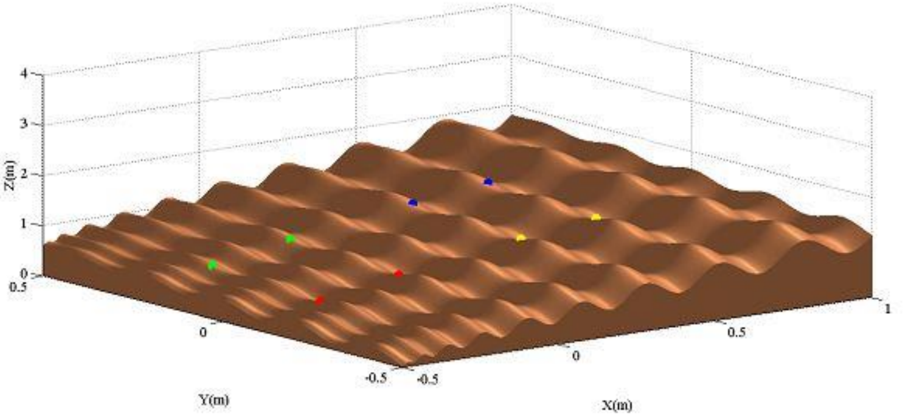

Fig. 3. An uneven terrain on which the robot should walk. The color circles represent the footholds of all legs when the robot in motion form move from the initial location to the target. Green, red, blue and yellow circles show the footholds of left hind, right hind, left front and right front legs, respectively.

Table 1. The physical specifications of the robot and the required data for the COG path generation

Parameter	Values (unit)	Description
m	40(kg)	Mass of main body
$\mathbf{I}_B$	$\begin{bmatrix} 0.4897 & 0 & 0 \\ 0 & 0.8667 & 0 \\ 0 & 0 & 1.2879 \end{bmatrix} (kg.m^2)$	Rotational of inertia of main body
$\mathbf{P}_G(t_0)$	$[0.01 \quad 0 \quad 0.544]^{\mathrm{T}}$ (m)	Initial COG position
$\mathbf{P}_G(t_f)$	$[0.21 \quad 0 \quad 0.5743]^{\mathrm{T}}$ **(**m)	final COG position
l_1	0.2 (m)	The length of the thigh
l_2	0.22 (m)	The length of the shank
m_1	2 (kg)	Mass of the thigh
$\mathbf{I}_1$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.0067 & 0 \\ 0 & 0 & 0.0067 \end{bmatrix} (kg.m^2)$	Rotational of inertia of the thigh
m_2	0.5 (kg)	Mass of the shank
$\mathbf{I}_2$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.002 & 0 \\ 0 & 0 & 0.002 \end{bmatrix} (kg.m^2)$	Rotational of inertia of the 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)	The distance between the COG of the thigh and hip joint
l_{c2}	0.08 (m)	The distance between the COG of the shank and the knee joint
ε	0.1	Stability margin

The path of tip of each swing leg within a single gait is shown in Figure 4. As seen, each swing leg leaves the ground and reaches to the prescribed target without any collision with the environment. In the design of the path of the main body, it is assumed that the orientation of main body remains fixed when the robot is in motion and their values are zero. The position, velocity and acceleration of the COG during a single gait are represented in Figure 5. The duration of each step of a gait and also the duration of the initial and final four leg-support phases is selected as *1 sec*. However, to increase the velocity of the robot, the duration of the four-leg support phase at the middle of the gait is chosen as a small value, *0.1 sec*. As expected, the COG path has the desired characteristics. In other words, the COG path is a continuous function in the levels of position, velocity and acceleration, particularly, at the instant of switching swing leg. In addition, the stability of the robot is guaranteed because the stability index when the robot is in motion is positive. The stability index is shown in Figure 6. When compared with the previous methods [12], this algorithm has some merits. First, in this algorithm, moments about the support edges are taken as the robot stability criterion, thus the designed path is more reliable for the real robot. Second, the design of the path along the Z-axis may help the robot to keep its balance over uneven terrains. This is due to the fact that acceleration along Z-axis may increase the moments about all edges of support polygon in cases when the accelerations of main body along x-and y-axes are limited due to following their desired paths towards the target and the robot becomes unstable by using these accelerations. In other words, since the stability index is a function of the motion variables along Z-axis, the motion along that axis may increase this index to improve robot stability through increasing or decreasing the height of the main body. On the other hand, the height of the main body should change accordingly to the variation of the terrain geometry to adapt to it. Here, the z-directional COG path is designed based on the robot stability whereas in conventional methods, there is no specific method for the design of the COG path along the z-axis.

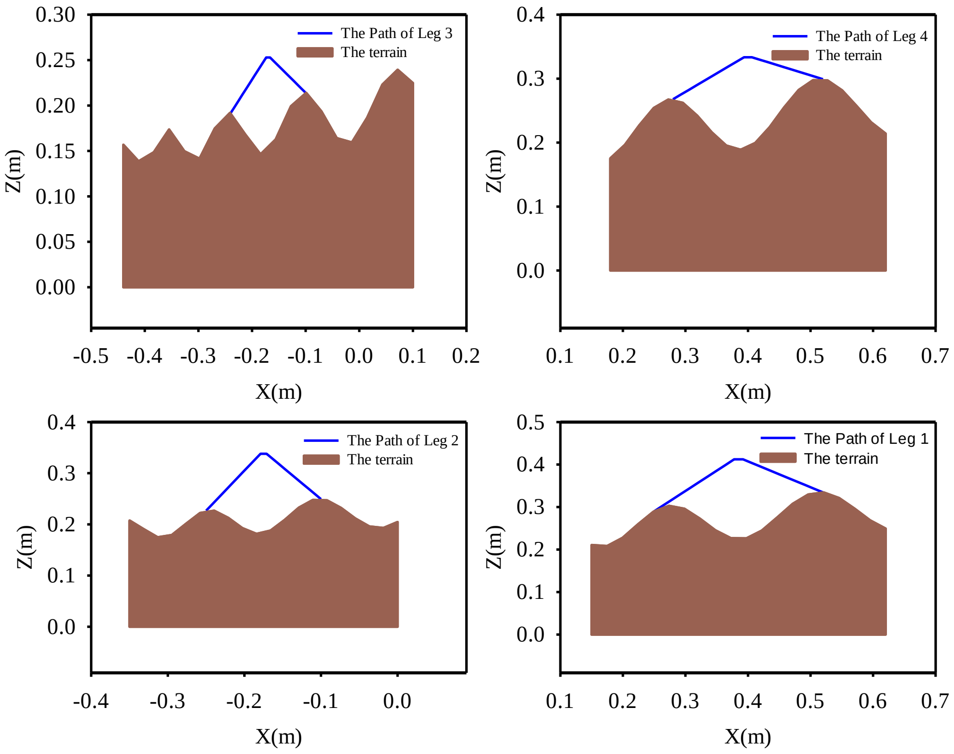

Fig. 4. The path of tip of all swing legs and the terrains which each leg should cross over it without any collision

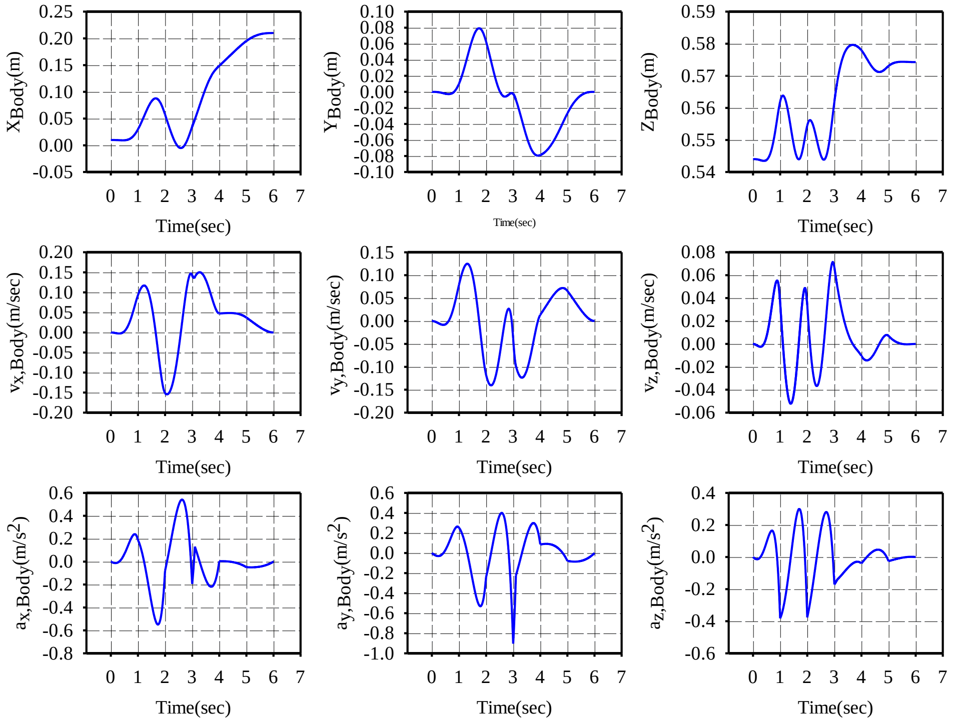

Fig. 5. The position, velocity and acceleration of the COG in motion on the terrain along x-, y- and z-axes for a single gait

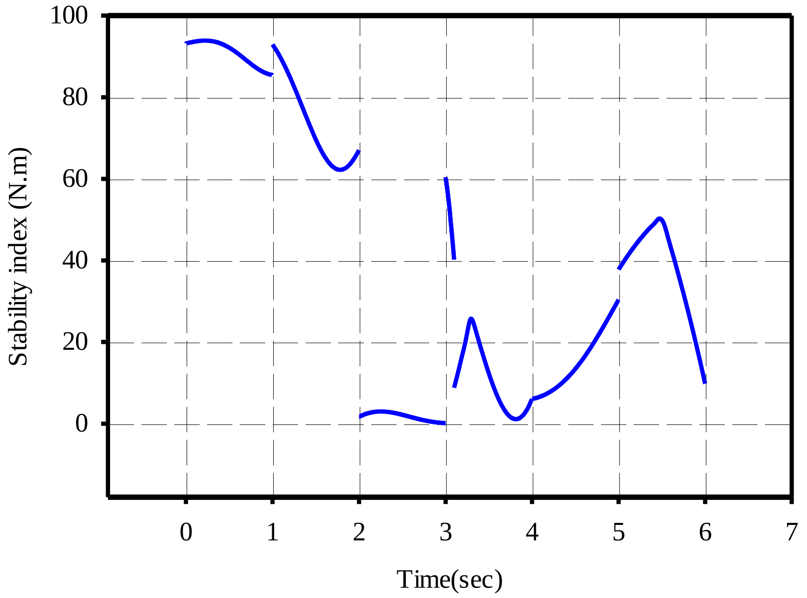

Fig. 6. The stability index in the motion over an uneven terrain during a single gait

The explicit dynamics model of the robot was obtained by the proposed method. The validity of the dynamics equations was confirmed with a simulated model. The joint torques for generating the designed path calculated using the inverse dynamics controller, are shown in Figure 7.

The contact forces exerted on the tip of the stance legs are depicted in Figure 8. These forces become zero when the leg goes to the swing phase. As shown, the forces along the z-axis during all the steps are positive. This means that the stance legs remain stationary on the ground for the period of all steps. The main feature of the inverse dynamics control is its computational efficiency and the joint torques when a single gait is computed in a small computation time. This is a good characteristic which makes the algorithm to be used in real-time. The stick-animation of the quadruped robot to perform its motion during a single gait is shown in Figure 9.

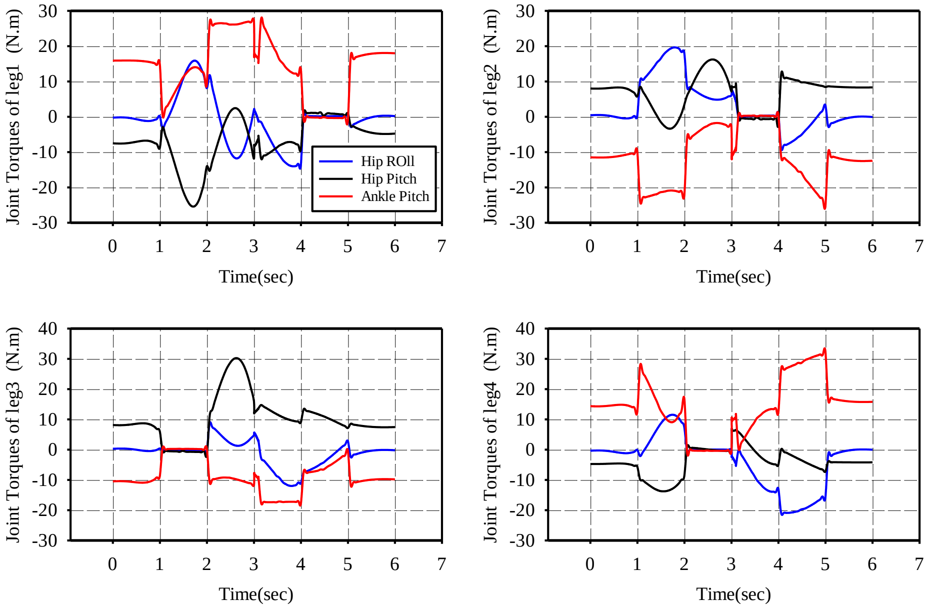

Fig. 7. The joint torques applied to generate the defined path

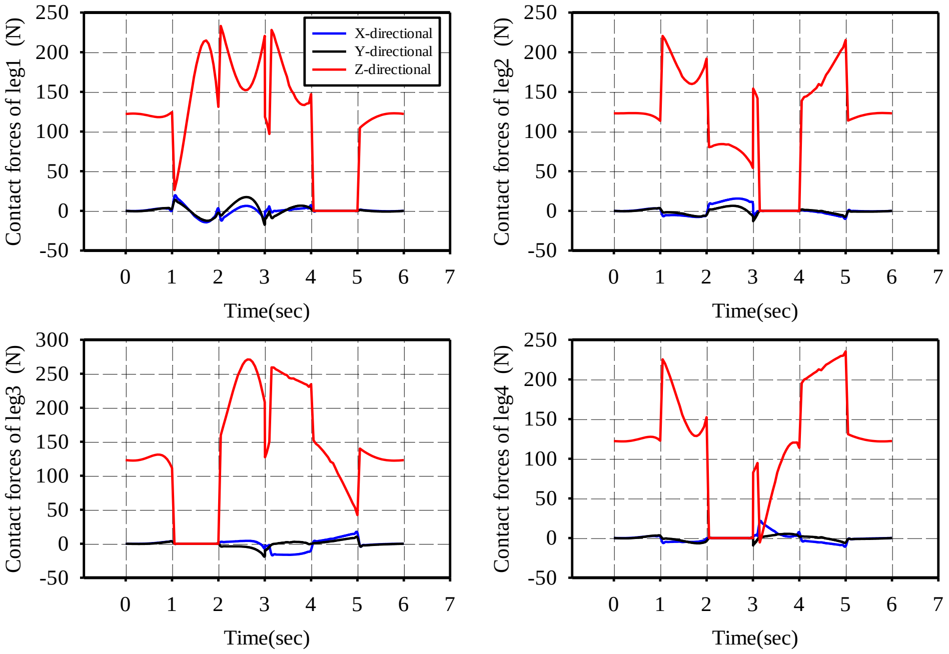

Fig. 8. The constraint forces exerted on the tip of the stance legs. When each leg goes to the swing phase, these forces become zero

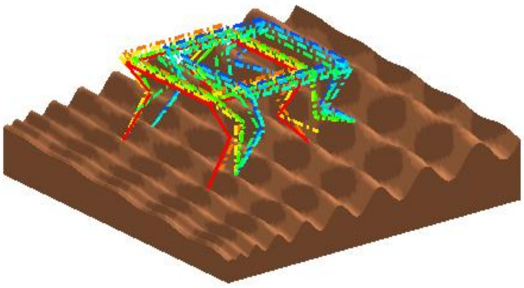

Fig. 9. The stick-animation of the quadruped robot in the motion on the uneven terrain

7. Conclusion

In this study, the dynamics modeling of a quadruped robot was investigated. In addition, a stable and smooth path for the COG in motion over an uneven terrain was generated. The explicit dynamics equations with good computational characteristic were derived by utilizing the explicit dynamics algorithm. A constraint elimination method was proposed to obtain free-constraint dynamics equations. Based on this model, an inverse dynamics controller was introduced to compute the joint torques. A stability condition was developed and a stable and smooth path was calculated through the definition of an optimization problem to obtain a stable path for the main body of the quadruped robot in motion on uneven terrain. Finally, an algorithm was proposed to compute a smooth and free-collision path for tip of swing legs. The proposed algorithm was tested on a quadruped robot in the simulation. The obtained results proved the merits of the proposed algorithm. A stable path was designed for robot in motion over an uneven terrain and this path was generated by using an efficient inverse-dynamics controller.

Appendix A

Through kinematic calculations, the linear and angular velocities of each rigid-body link are known. At this time, the terms of dynamics equations can easily be computed. The mass matrix can be calculated as [17].

$$\begin{aligned}
M_{ij} &= M_b \frac{\partial \overrightarrow{R_b}}{\partial q_i} . \frac{\partial \overrightarrow{R_b}}{\partial q_j} + \frac{\partial \overrightarrow{\boldsymbol{\omega}_b}}{\partial \dot{q}_i} .I_b . \frac{\partial \overrightarrow{\boldsymbol{\omega}_b}}{\partial \dot{q}_j} \\
&+ \sum_{m=1}^{4} \sum_{k=1}^{2} \left(m_K^{(m)} \frac{\partial \overrightarrow{R_{c_K}}^{(m)}}{\partial q_i} . \frac{\partial \overrightarrow{R_{c_K}}^{(m)}}{\partial q_j} + \frac{\partial \overrightarrow{\boldsymbol{\omega}_k}^{(m)}}{\partial \dot{q}_i} .I_k^{(m)} . \frac{\partial \overrightarrow{\boldsymbol{\omega}_k}^{(m)}}{\partial \dot{q}_j} \right) \\
&+ \sum_{m=1}^{4} \sum_{k=1}^{2} \left(m_K^{(m)} \frac{\partial \overrightarrow{R_{c_K}}^{(m)}}{\partial q_i} \right) . \frac{\partial \overrightarrow{R_b}}{\partial q_j} + \sum_{m=1}^{4} \sum_{k=1}^{2} \left(m_K^{(m)} \frac{\partial \overrightarrow{R_{c_K}}^{(m)}}{\partial q_j} \right) . \frac{\partial \overrightarrow{R_b}}{\partial q_i}
\end{aligned} \tag{35}$$

where $\overrightarrow{R_b}$ and $\overrightarrow{\boldsymbol{\omega}_b}$ are the position and angular velocity of the COG of the main body, respectively. In addition, $\overrightarrow{R_{c_K}}^{(m)}$ and $\overrightarrow{\boldsymbol{\omega}_k}^{(m)}$ are the position and angular velocity of the COG of the k^{th} rigid body link of the m^{th} leg, respectively. The vector of the nonlinear terms can be defined as:

$$V = C_1 + C_2 \dot{\mathrm{q}} \tag{36}$$

where

$$\begin{aligned}
C_{1ij} &= M_T \frac{\partial \overrightarrow{R_b}}{\partial q_i} . \left(\sum_{s=1}^{18} \frac{\partial^2 \overrightarrow{R_b}}{\partial q_s \partial q_j} \dot{q}_s \right) + \frac{\partial \overrightarrow{\boldsymbol{\omega}_b}}{\partial \dot{q}_i} .I_b . \frac{\partial \overrightarrow{\boldsymbol{\omega}_b}}{\partial q_j} + \overrightarrow{\boldsymbol{\omega}_b} .I_b . \frac{\partial^2 \overrightarrow{\boldsymbol{\omega}_b}}{\partial \dot{q}_i \partial q_j} \\
&+ \frac{\partial \overrightarrow{R_b}}{\partial q_i} . \sum_{m=1}^{4} \sum_{k=1}^{2} \left(m_K^{(m)} \sum_{s=1}^{18} \frac{\partial^2 \overrightarrow{R_{c_k}}^{(m)}}{\partial q_s \partial q_j} \dot{q}_s \right) + \left(\sum_{s=1}^{18} \frac{\partial^2 \overrightarrow{R_{c_k}}^{(m)}}{\partial q_s \partial q_j} \dot{q}_s \right) \sum_{m=1}^{4} \sum_{k=1}^{2} \left(m_K^{(m)} \frac{\partial \overrightarrow{R_{c_K}}^{(m)}}{\partial q_j} \right) \\
&\sum_{m=1}^{4} \sum_{k=1}^{2} \left(m_K^{(m)} \frac{\partial \overrightarrow{R_{c_K}}^{(m)}}{\partial q_i} \left(\sum_{s=1}^{18} \frac{\partial^2 \overrightarrow{R_{c_k}}^{(m)}}{\partial q_s \partial q_j} \dot{q}_s \right) + \frac{\partial \overrightarrow{\boldsymbol{\omega}_k}^{(m)}}{\partial \dot{q}_i} .I_k^{(m)} . \frac{\partial \overrightarrow{\boldsymbol{\omega}_k}^{(m)}}{\partial q_j} + \overrightarrow{\boldsymbol{\omega}_k}^{(m)} .I_k^{(m)} . \frac{\partial^2 \overrightarrow{\boldsymbol{\omega}_k}^{(m)}}{\partial \dot{q}_i \partial q_j} \right)
\end{aligned} \tag{37}$$

and C_{2i} is given by:

$$C_{2i} = -\left(\overrightarrow{\boldsymbol{\omega}_b} . I_b . \frac{\partial \overrightarrow{\boldsymbol{\omega}_b}}{\partial q_i} + \sum_{m=1}^{4} \sum_{k=1}^{2} \left(\overrightarrow{\boldsymbol{\omega}_k}^{(m)} . I_k^{(m)} . \frac{\partial \overrightarrow{\boldsymbol{\omega}_k}^{(m)}}{\partial q_i} \right) \right) \qquad (38)$$

The gravitational forces, G, can be computed as:

$$G_i = M_b \vec{g} . \frac{\partial \overrightarrow{R_b}}{\partial q_i} + \vec{g} . \sum_{m=1}^{4} \sum_{k=1}^{2} \left(m_K^{(m)} \frac{\partial \overrightarrow{R_{c_K}}^{(m)}}{\partial q_i} \right) \qquad (39)$$

References

[1]. Raibert, M. 2008, "BigDog, the Rough-Terrain Quadruped Robot," in *Proceedings of the 17th IFAC World Congress*, COEX, Korea, South.

[2]. Semini, C. 2010, "HyQ - Design and Development of a Hydraulically Actuated Quadruped Robot," Doctor of Philosophy (Ph.D.), University of Genoa, Italy.

[3]. Hutter, M., Gehring, M., Bloesch, M., Mark, A.H., Remy, C.D. and Siegwart, R.Y. 2013, *StarlETH: A compliant quadrupedal robot for fast, efficient, and versatile locomotion*: Autonomous Systems Lab, ETH Zurich.

[4]. Moosavian, S.A.A., Alghooneh, M. and Takhmar, A. 2009, "Cartesian approach for gait planning and control of biped robots on irregular surfaces," *International Journal of Humanoid Robotics*, vol. 6, pp. 675-697.

[5]. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K. and Hirukawa, H. 2003, "Biped walking pattern generation by using preview control of zero-moment point," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, Vol. 2, pp. 1620-1626.

[6]. Sardain P. and Bessonnet, G. 2004, "Forces acting on a biped robot. Center of pressure-zero moment point," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, Vol. 34, pp. 630-637.

[7]. Takao, S., Yokokohji, Y. and Yoshikawa, T. 2003, "FSW (feasible solution of wrench) for multi-legged robots," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, pp. 3815-3820.

[8]. Ajallooeian, M., Gay, S., Tuleu, A., Sprowitz, A. and Ijspeert, A.J. 2013, "Modular control of limit cycle locomotion over unperceived rough terrain," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 3390-3397.

[9]. Hirukawa, H., Hattori, S., Harada, K., Kajita, S., Kaneko, Kanehiro, K.F., Fujiwara, K. and Morisawa, M. 2006, "A universal stability criterion of the foot contact of legged robots-adios ZMP," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 1976-1983.

[10]. Papadopoulos E. and Rey, D.A. 1996, "A new measure of tipover stability margin for mobile manipulators," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, pp. 3111-3116.

[11]. Sugihara, T., Nakamura, Y. and Inoue, H. 2002, "Real-time humanoid motion generation through ZMP manipulation based on inverted pendulum control," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, pp. 1404-1409.

[12]. Kalakrishnan, M., Buchli, J., Pastor, P., Mistry, M. and Schaal, S. 2011, "Learning, planning, and control for quadruped locomotion over challenging terrain," *The International Journal of Robotics Research,* Vol. 30, pp. 236-258.

[13]. Zheng, Y., Lin, M.C., Manocha, D., Adiwahono, A.H. and Chew, C.M. 2010, "A walking pattern generator for biped robots on uneven terrains," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 4483-4488.

[14]. Alipour K. and Moosavian, S.A.A. 2012, "Effect of terrain traction, suspension stiffness and grasp posture on the tip-over stability of wheeled robots with multiple arms," *Advanced Robotics*, Vol. 26, pp. 817-842.

[15]. Craig, J.J. 2005, *Introduction to robotics : mechanics and control*, 3rd ed. Upper Saddle River, N.J.: Pearson/Prentice Hall.

[16]. Featherstone, R. 2008, *Rigid body dynamics algorithms*, Vol. 49: Springer New York.

[17]. Moosavian S.A.A. and Papadopoulos, E. 2004, "Explicit dynamics of space free-flyers with multiple manipulators via SPACEMAPLE," *Advanced Robotics*, Vol. 18, pp. 223-244.

[18]. Mistry, M., Buchli, J. and Schaal, S. 2010, "Inverse dynamics control of floating base systems using orthogonal decomposition," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 3406-3412.

[19]. Aghili, F. 2005, "A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: applications to control and simulation," *Robotics, IEEE Transactions on,* Vol. 21, pp. 834-849.

[20]. Righetti, L., Buchli, J., Mistry, M. and Schaal, S. 2011, "Inverse dynamics control of floating-base robots with external constraints: A unified view," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 1085-1090.

[21]. Kurazume, R., Hirose, S. and Yoneda, K. 2001, "Feedforward and feedback dynamic trot gait control for a quadruped walking vehicle," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, pp. 3172-3180, Vol.3.

[22]. Yoneda, K., Iiyama, H. and Hirose, H. 1996, "Intermittent trot gait of a quadruped walking machine dynamic stability control of an omnidirectional walk," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, pp. 3002-3007.

[23].Song S.M. and Waldron, K.J. 1989, *Machines that walk: the adaptive suspension vehicle*: MIT press.

[24]. McGhee R.B. and Frank, A.A. 1968, "On the stability properties of quadruped creeping gaits," *Mathematical Biosciences,* vol. 3, pp. 331-351.

[25]. Hutter, M. 2013, "StarlETH & Co-design and control of legged robots with compliant actuation," Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 21073,.