JCAMECH

Vol. 48, No. 1, June 2017, pp 27-38
DOI: 10.22059/jcamech.2017.232721.141

Stiffness control of a legged robot equipped with a serial manipulator in stance phase

Mohammad Reza Lavaei, Mohammad Mahjoob*, Amir Behjat

Center for Mechatronics and Intelligent Machines
School of Mechanical Engineering, College of Engineering, University of Tehran
Tehran, Iran

Received: 6 May. 2017, Accepted: 28 June. 2017

Abstract

The ability to perform different tasks by a serial manipulator mounted on legged robots, increases the capabilities of the robot. The position/force control problem of such a robot in the stance phase with point contacts on the ground is investigated here. A target plane with known stiffness is specified in the workspace. Active joints of the legs and serial manipulator are used to exert the desired normal force on the plane while tracking a desired trajectory on the plane. First, the equations of motion of the robot and contact forces of the feet on the ground are derived. A controller is then proposed which tracks the desired trajectory while keeping the feet contacts on the ground and prevent slipping. An optimization problem is solved in each control loop to minimize the actuation effort. This minimization is subject to position tracking for the end-effect or (using inverse dynamics controller), force requirements of the feet contacts with the ground, and actuators capabilities. Simulations are conducted for the simplified model of a quadruped robot with a 2-DOF serial manipulator. To test the controller, a 20 N normal force is applied onto the target plane while moving the tip of the end-effector. It is shown that the robot can perform the task effectively without losing the ground contact and slipping.

Keywords: Legged robots, Stiffness control, Minimum effort, Contact forces, Serial manipulator.

1. Introduction

The end-effector of a serial manipulator has force interactions with the environment while performing a position/force control task. In such a situation a method for accounting the interaction forces in the controller is required. Without a force control strategy, the robot or the environment may be damaged. Also, some tasks such as grinding and polishing depend on applying a desired force on a surface while tracking a desired trajectory along the surface. In stiffness control [1], [2], the desired force is exerted by commanding a desired trajectory inside the environment whose stiffness is known. The concept of hybrid position/force control proposed in [3] and extended in [4] uses the feedback data of force sensors to combine position and force controllers. Some variations of this method is presented in [5] and [6].

However, when a serial manipulator is mounted on a mobile robot, stability of the whole robot becomes an important matter. The mobile robot is not fixed on the ground, hence force slipping or losing the contact with

* Corresponding Author. Tel.: +982161119906
Email Address: mmahjoob@ut.ac.ir

the ground may happen. When the mobile robot is supposed to travel in an unstructured terrain, as in search-and-rescue operations, legged robots are better choices compared to wheeled robots [7]. Inverse dynamics controllers are implemented for position control of legged robots in locomotion [8]. Generating actuation torques or forces in legged robots that prevent slipping or losing contact of the legs which are in stance phase during the locomotion is also considered in [9]–[13]. The control method proposed in [9] computes the torques that minimizes tangential forces in contact points with the ground during locomotion. The advantage of using this method is that it gives an analytical solution to generate such input torques. But it does not check if the feet which are supposed to be in stance phase, actually remain in contact with the ground. Checking this condition becomes more important when a serial manipulator, which is operating a position/force control task, is mounted on the body of a legged robot.

Separate position control for the serial manipulator and the body is proposed and implemented on a quadruped in [14], but separate controls needs estimation or feedback data of the force interaction between the body and the serial manipulator. In a position/force control task, the requirements of the contact points on the ground may be violated more easily than in a position control task. Hence, to avoid delays and errors, we do not separate the control. In addition, separate controls makes it difficult to present a minimum-effort control law, which is desired in mobile robots.

We propose a general method for stiffness control of a serial manipulator mounted on a legged robot. The robot is assumed to be in stance phase and we utilize both the active joints of the legs and the serial manipulator to perform the task by a minimum effort control strategy. The actuation torques computed in the controller, will ensure that the feet remain in contact with the ground and do not slip. Presenting a unified minimum effort control method for the whole legged robot to perform stiffness control for the end-effector of the mounted serial manipulator while keeping the contacts of the feet on the ground and preventing the robot from slipping is not treated in the literature and is the subject of this paper. The modeling and the proposed control method is general and is not limited to a particular design of the robot.

The kinematics and dynamics of the robot is modeled in section 2. The controller is introduced in section 3 where an optimization problem is solved in each loop of the control to generate the required actuation efforts. Simulation is performed for a simplified model of a quadruped equipped with a 2-DOF serial manipulator in section 4. Results and discussion are presented in section 5 and the conclusion is presented in section 6.

2. Modeling

A general legged robot is shown in Fig. 1. All the legs are in contact with the ground and there is no slipping between the feet and the ground. The tip of the end-effector of the serial manipulator is in touch with a specified plane in the workspace, but there is no force interaction between them. We assume this configuration to be the initial configuration in our problem.

Reaching to the configuration in Fig. 1 from another arbitrary configuration can be done using a suitable method for the problem such as what discussed in [14]. Also, we can treat the serial manipulator as another leg and use the position control methods for legged robots to reach the configuration depicted in Fig. 1[8]. The contact points of the legs on the ground are fixed and we assume that we have their position vectors in an inertial coordinate system.

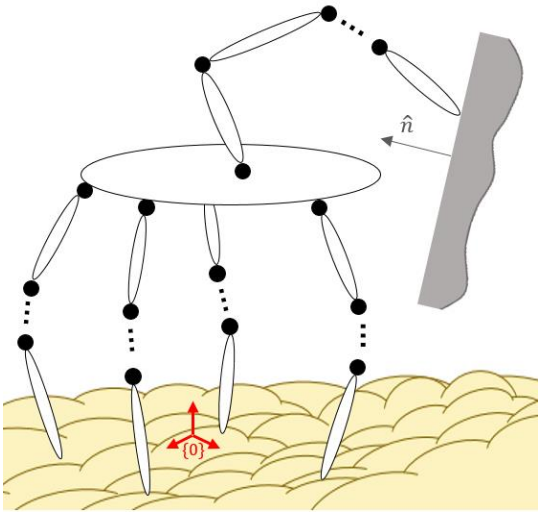

Fig. 1. A general legged robot in stance phase with a mounted serial manipulator. The position of the contact points are fixed and known in the inertial coordinate system {0}. The tip of the end-effector is in touch with a specified plane, but no force is exerted on the plane in this configuration.

2.1. *Kinematics*

We describe the position and orientation of the body of the robot in the inertial coordinate system by $\boldsymbol{q}_b \in \mathbb{R}^W$. We denote the joint configuration of the legs by $\boldsymbol{q}_l \in \mathbb{R}^j$ and the joint configuration of the serial manipulator

by $\boldsymbol{q}_a \in \mathbb{R}^r$. Thus, the set of generalized coordinates $\boldsymbol{q} \in \mathbb{R}^n$, $n = w + j + r$, is:

$$\boldsymbol{q} = [\boldsymbol{q}_b^T \quad \boldsymbol{q}_a^T \quad \boldsymbol{q}_l^T]^T \tag{1}$$

Now if we attach local coordinate systems to the links, the position and orientation of them can be obtained in terms of $\boldsymbol{q}$.

We have assumed that the robot is in stance phase. Therefore, if it has s legs, there are s independent vector loops that define the kinematics of the body and legs. Writing these vector loops in a Cartesian reference frame, we have $3s$ equations, $\boldsymbol{c}_1 \in \mathbb{R}^{3s}$:

$$\boldsymbol{c}_1(\boldsymbol{q}_b, \boldsymbol{q}_l) = \boldsymbol{0} \tag{2}$$

These are in fact, constraint equations that must be satisfied in the kinematic model of the robot. First and second time-derivatives of (2) are:

$$A\dot{\boldsymbol{q}} = \boldsymbol{0} \tag{3}$$

$$A\ddot{\boldsymbol{q}} + \dot{A}\dot{\boldsymbol{q}} = \boldsymbol{0}, \tag{4}$$

where, $A(\boldsymbol{q}) \in \mathbb{R}^{m \times n}$, $m = 3s$ is:

$$A = \frac{\partial \boldsymbol{c}_1}{\partial \boldsymbol{q}} \tag{5}$$

If we denote the tip of the end-effector of the serial manipulator by D, the position of this point in the inertial frame is a function of $\boldsymbol{q}_b$ and $\boldsymbol{q}_a$:

$$\boldsymbol{c}_2(\boldsymbol{q}_b, \boldsymbol{q}_a) = \boldsymbol{r}_D \tag{6}$$

Also, the orientation of the end-effector in the reference frame is related to $\boldsymbol{q}_b$ and $\boldsymbol{q}_a$. For example, if we let ψ, ϕ and θ be the xyz Euler angles of the end-effector, the orientation of the end-effector is:

$$\boldsymbol{c}_3(\boldsymbol{q}_b, \boldsymbol{q}_a) = [\psi \quad \phi \quad \theta]^T \tag{7}$$

2.2. *Dynamics*

We assume that the stiffness of the ground is high enough so that it can be considered as a rigid material. Because the end-effector is in contact with another environment, the forces between the two must be accounted in the dynamics. It is desired to apply a normal force by the end-effector's tip to a plane specified in the workspace which is the external surface of an environment. The unit normal vector of this plane is $\hat{\boldsymbol{n}} \in \mathbb{R}^3$. We assume that the stiffness of the environment is uniform in the $\hat{\boldsymbol{n}}$ direction and denote it by k. We neglect the friction force between the surface and the tip of the end-effector. In the initial configuration, the tip of the end-effector is just in touch with the plane but does not exert any force on it. We denote the tip of the end-effector by D and its initial position in the inertial frame by $r_D^\star$.

There is no force interaction between the end-effector and the environment if point D is outside the environment, i.e. when $(\boldsymbol{r}_D - \boldsymbol{r}_D^\star).\hat{\boldsymbol{n}} \geq 0$ in Fig. 2. When point D goes inside the environment, $(\boldsymbol{r}_D - \boldsymbol{r}_D^\star).\hat{\boldsymbol{n}}$ becomes negative in Fig. 2, and the end-effector is exerting force on the environment.

In this case, the work done by the tip of the end-effector on the environment is:

$$W_f = \frac{1}{2}k\left((\boldsymbol{r}_D - \boldsymbol{r}_D^\star).\hat{\boldsymbol{n}}\right)^2 \tag{8}$$

In matrix notation, the above equation is:

$$W_f = \frac{1}{2}k\left(\hat{\boldsymbol{n}}^T(\boldsymbol{r}_D - \boldsymbol{r}_D^\star)\right)^2 \tag{9}$$

Therefore:

$$\delta W_f = k\,\hat{\boldsymbol{n}}^T(\boldsymbol{r}_D - \boldsymbol{r}_D^\star)\hat{\boldsymbol{n}}^T\frac{\partial \boldsymbol{r}_D}{\partial \boldsymbol{q}}\delta\boldsymbol{q} \tag{10}$$

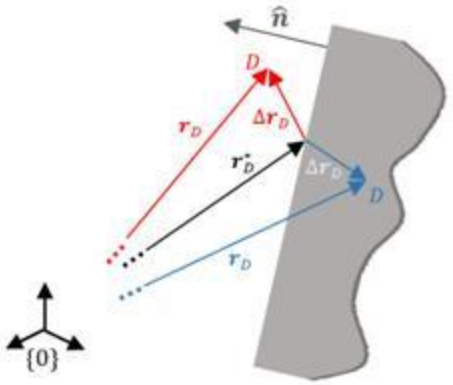

Fig. 2. The difference between the current and the initial position vector of point D, the tip of the end-effector, is denoted by $\boldsymbol{\Delta r_D}$. If point D is outside the environment (red vectors): $\boldsymbol{\Delta r_D}.\hat{\boldsymbol{n}} \geq \boldsymbol{0}$. If point D is inside the environment (blue vectors): $\boldsymbol{\Delta r_D}.\hat{\boldsymbol{n}} < \boldsymbol{0}$

If all the legs remain in contact with the ground and do not slip, the dynamics of the robot which has p actuators, using Lagrange's method, can be written in the form of:

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}}) = B\boldsymbol{\tau} + A^T\boldsymbol{\lambda} - k\left(\frac{\partial \boldsymbol{r}_D}{\partial \boldsymbol{q}}\right)^T \hat{\boldsymbol{n}}(\boldsymbol{r}_D - \boldsymbol{r}_D^\star)^T\hat{\boldsymbol{n}} \qquad (11)$$
$$A\ddot{\boldsymbol{q}} + \dot{A}\dot{\boldsymbol{q}} = \boldsymbol{0},$$

where $M(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the positive-definite inertia matrix, $\boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^n$ is the vector of gravitational, centrifugal and Coriolis forces, $B \in \mathbb{R}^{n\times p}$ is the actuation selector matrix, $\boldsymbol{\tau} \in \mathbb{R}^p$ is the actuation vector, $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the vector of Lagrange's multipliers, and the last term is the contribution of the normal interaction force to the dynamics of the robot. This term is the transpose of what we obtained in (10). The second equation in (11) is the constraint equations at acceleration level in (4). In (11), it is obvious that as long as $\boldsymbol{r}_D$ equals $\boldsymbol{r}_D^\star$, there will be no exerted force on the plane. This is especially true when the joint configuration of the robot changes but $\boldsymbol{r}_D$ remains equal to its initial value: $\boldsymbol{r}_D^\star$.

The main simplification that we had made to include the external force in (11), is that we have assumed that the normal vector of the target plane and its stiffness is known. Therefore, tracking the desired trajectories will ensure the tracking of the desired force. However, if the normal vector and the stiffness are not known, force sensor data are needed to model the dynamics; in which case, the control procedure must have a force control loop to track desired forces [15].

To simplify the notation, according to what we discussed about Fig. 2, we take:

$$\boldsymbol{h}_k = \begin{cases} \boldsymbol{h} + k\left(\frac{\partial \boldsymbol{r}_E}{\partial \boldsymbol{q}}\right)^T \hat{\boldsymbol{n}}(\boldsymbol{r}_D - \boldsymbol{r}_D^\star)^T\hat{\boldsymbol{n}}(\boldsymbol{r}_D - \boldsymbol{r}_D^\star)^T\hat{\boldsymbol{n}} < 0 \\ \boldsymbol{h}(\boldsymbol{r}_D - \boldsymbol{r}_D^\star)^T\hat{\boldsymbol{n}} \geq 0 \end{cases} \qquad (12)$$

and rewrite (11):

$$M\ddot{\boldsymbol{q}} + \boldsymbol{h}_k = B\boldsymbol{\tau} + A^T\boldsymbol{\lambda}$$
$$A\ddot{\boldsymbol{q}} + \dot{A}\dot{\boldsymbol{q}} = \boldsymbol{0}, \qquad (13)$$

In the inverse dynamics problem, (13) is a set of $n+m$ equations for $n+m$ unknowns, namely $\boldsymbol{q}$ and $\boldsymbol{\lambda}$.

The Lagrange's multipliers can be obtained from (13). We obtain $\ddot{\boldsymbol{q}}$ from the first equation in (13) and then replace it in the second equation[16]. The result is:

$$\boldsymbol{\lambda} = B_1(\boldsymbol{q})\boldsymbol{\tau} + \boldsymbol{B}_2(\boldsymbol{q},\dot{\boldsymbol{q}}), \qquad (14)$$

where B_1 and $\boldsymbol{B}_2$ are:

$$B_1 = -(AM^{-1}A^T)^{-1}AM^{-1}B$$
$$\boldsymbol{B}_2 = (AM^{-1}A^T)^{-1}\left(AM^{-1}\boldsymbol{h}_k - \dot{A}\dot{\boldsymbol{q}}\right) \qquad (15)$$

3. Control method

Here, the tasks of the control system are to: 1) apply a desired amount of normal force by the tip of the end-effector on a defined plane in the workspace, 2) tracking a desired trajectory for the end-effector's tip on the plane and tracking the desired orientation of the end-effector, and 3) ensuring that the feet remain in contact with the ground and do not slip. The computed τ in the controller to satisfy the above requirements will be based on (13). Note that in this equations, we have in fact modeled the force interaction of the robot's end-effector and the specified plane by its equivalent effect on the generalized coordinates using the stiffness of the environment. Now, tracking the desired trajectories of the generalized coordinates will accomplish both tasks of position and force control of the end-effector, i.e. tasks 1 and 2.

3.1. *Controller*

In order to obtain a convenient form of the equations of motion for inverse dynamics control law, Lagrange's multipliers must be eliminated from (13). We first obtain the orthogonal complement of matrix A, which is $C \in \mathbb{R}^{(n-m)\times n}$ and then pre-multiply the first equation in (13) by C^T:

$$C^TM\ddot{\boldsymbol{q}} + C^T\boldsymbol{h}_k = C^TB\boldsymbol{\tau} + (AC)^T\boldsymbol{\lambda} \qquad (16)$$

Since $AC = 0_{m\times n}$, Lagrange multipliers are eliminated and from the equations of motion. Denote C^TM in (16) by $\hat{M}$, $C^T\boldsymbol{h}_k$ by $\hat{\boldsymbol{h}}_k$, and C^TB by $\hat{B}$, and rewrite (16):

$$\hat{M}\ddot{\boldsymbol{q}} + \hat{\boldsymbol{h}}_k = \hat{B}\boldsymbol{\tau} \qquad (17)$$

where $\hat{M} \in \mathbb{R}^{(n-m)\times n}$, $\hat{\boldsymbol{h}}_k \in \mathbb{R}^{n-m}$, and $\hat{B} \in \mathbb{R}^{(n-m)\times p}$.

Thus, after the elimination of the Lagrange's multipliers, (13) can be written in the form of:

$$\hat{M}\ddot{\boldsymbol{q}} + \hat{\boldsymbol{h}}_k = \hat{B}\boldsymbol{\tau}$$
$$A\ddot{\boldsymbol{q}} + \dot{A}\dot{\boldsymbol{q}} = \boldsymbol{0}, \qquad (18)$$

Different methods such as SVD [17] or QR decomposition [18] can be used to compute C.The above equation can be written in the form of:

$$\tilde{M}\ddot{\boldsymbol{q}} + \tilde{\boldsymbol{h}}_k = \tilde{B}\boldsymbol{\tau}, \qquad (19)$$

where $\tilde{M} \in \mathbb{R}^{n\times n}$, $\tilde{\boldsymbol{h}}_k \in \mathbb{R}^n$ and $\tilde{B} \in \mathbb{R}^{n\times p}$ are:

$$\tilde{M} = \begin{bmatrix} \hat{M} \\ A \end{bmatrix}, \tilde{\boldsymbol{h}}_k = \begin{bmatrix} \hat{\boldsymbol{h}} \\ \dot{A}\dot{\boldsymbol{q}} \end{bmatrix}, \tilde{B} = \begin{bmatrix} \hat{B} \\ 0 \end{bmatrix} \qquad (20)$$

For now, we assume that the trajectories of the generalized coordinates are already planned. These trajectories, if properly tracked, will ensure that the end-effector will exert the desired force on the plane, follows the desired trajectory on the plane, and that the dynamic stability of the robot is not violated. Also, we assume that the feedback data of $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ are available in each control loop.Thus, in each loop in the controller, having the values of $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$, we use an inverse dynamics control law to track the desired trajectories. The desired values of the generalized coordinates and their first and second time-derivatives are denoted by subscript d. We compute vector $\boldsymbol{u} \in \mathbb{R}^n$:

$$\boldsymbol{u} = \ddot{\boldsymbol{q}} = \ddot{\boldsymbol{q}}_d + K_D(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + K_P(\boldsymbol{q}_d - \boldsymbol{q}), \quad (21)$$

where, $K_D \in \mathbb{R}^{n\times n}$ and $K_P \in \mathbb{R}^{n\times n}$ are the positive-definite diagonal matrices of D and P coefficients in the PD controller. The above equation will result in a stable error dynamics, $\boldsymbol{e} = \boldsymbol{q} - \boldsymbol{q}_d$, $\boldsymbol{e} \in \mathbb{R}^n$:

$$\ddot{\boldsymbol{e}} + K_D\dot{\boldsymbol{e}} + K_P\boldsymbol{e} = \boldsymbol{0} \quad (22)$$

Note that $\boldsymbol{u} = \ddot{\boldsymbol{q}}$ computed in (21) may violate the constraint equations at acceleration level, i.e.(4). Therefore, we will compute $\boldsymbol{u} = \ddot{\boldsymbol{q}}$ in a slightly different way.

If we denote the active generalized coordinates by $\overline{\boldsymbol{q}} \in \mathbb{R}^p$, we first compute $\overline{\boldsymbol{u}} \in \mathbb{R}^p$:

$$\overline{\boldsymbol{u}} = \ddot{\overline{\boldsymbol{q}}} = \ddot{\overline{\boldsymbol{q}}}_d + K_D\left(\dot{\overline{\boldsymbol{q}}}_d - \dot{\overline{\boldsymbol{q}}}\right) + K_P(\boldsymbol{q}_d - \boldsymbol{q}) \quad (23)$$

Where $K_V \in \mathbb{R}^{p\times p}$ and $K_P \in \mathbb{R}^{p\times p}$. Then, denoting the passive generalized coordinates by $\boldsymbol{q}^\circ \in \mathbb{R}^{n-p}$, we can obtain $\boldsymbol{u}^\circ = \ddot{\boldsymbol{q}}^\circ$ from (4) (maybe by using a generalized inverse). Then we arrange the elements of $\overline{\boldsymbol{u}}$ and $\boldsymbol{u}^\circ$ to obtain the vector $\boldsymbol{u}$.

Now, computing$\widehat{M}$, $\widehat{\boldsymbol{h}}_k$ and $\hat{B}$ in (18) by feedback data of $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$, we can find the required values of $\boldsymbol{\tau}$ to track the desired trajectories:

$$\hat{B}\boldsymbol{\tau} = \widehat{M}\boldsymbol{u} + \widehat{\boldsymbol{h}}_k, \quad (24)$$

where $\hat{B}$ is a $n-m$ by p matrix. The usual case for legged robots in stance phase is that $n-m > p$. Now in presence of a mounted serial manipulator, this inequality becomes stronger. Therefore, (24)is a set of equations with more unknowns than the number of equations. Hence, Moore-Penrose generalized inverse gives the answer for $\boldsymbol{\tau}$ while minimizing norm-2 of the $\boldsymbol{\tau}$.

Applying the computed actuation vector in (24) by the actuators, ensures the tracking objective of the desired force on the plane and the desired position/orientation of the end-effector. But note that tracking the desired trajectories of the generalized coordinates is not sufficient in this problem. Note that (13) which our controller is based upon, was derived by the assumption that all feet remain in contact with the ground and do not slip. When the computed $\boldsymbol{\tau}$ in (24) is applied to the robot, it may result in lifting the feet from the ground or tangential contact forces between some of the feet and the ground which cause slipping.

Lagrange's multipliers in (13) are related to contact (or constraint) forces between the feet and the ground. The relation can be obtained if the normal vector of the surface under each foot is known. We have derived Lagrange's multipliers in (14) in terms of$\boldsymbol{\tau}$, $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$. Let $\boldsymbol{f}_i$ and N_i represent the normal and tangential constraint forces exerted to the i-th feet in contact with the ground. According to what we discussed, we can express $\boldsymbol{f}_i$ and N_i in terms of the $\boldsymbol{\tau}$, $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ using(14), if the normal vectors of all surfaces under the feet are known:

$$i = 1, 2, \dots, s \begin{cases} \boldsymbol{f}_i = C_{1_i}(\boldsymbol{q})\boldsymbol{\tau} + \boldsymbol{C}_{2_i}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ N_i = C_{3_i}(\boldsymbol{q})\boldsymbol{\tau} + \boldsymbol{C}_{4_i}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \end{cases} \quad (25)$$

These equations can be used in the controller to prevent slipping and loss of contact with the ground. The requirements on $\boldsymbol{f}_i$ and N_i are inequality constraints, because N_i must be positive and $|\boldsymbol{f}_i| < \mu_{s_i}N_i$, where μ_{s_i} is the static coefficient of friction corresponding to the i-th foot on the ground. Therefore,(25) can be considered as inequality constraints for an optimization problem.

In mobile robots, it is desired to seek minimum-effort control laws, because these robots have to carry their own power supplies. In the controller we can solve an optimization problem in each loop, to find the minimum of the norm-2 of $\boldsymbol{\tau}$, subjected to equality constraint (24) and inequality constraints(25). The problem must also be subjected to actuator's limits to prevent saturation.

To sum up, having the feedback values of $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ in each loop, we first construct $\boldsymbol{u}$ and compute $\widehat{M}$, $\hat{B}$, $\widehat{\boldsymbol{h}}_k$ in (24) and C_1 to $\boldsymbol{C}_4$ in (25) for all the legs. Then we solve the following optimization problem:

$$\min \|\boldsymbol{\tau}\|_2$$
$$\text{subject to: } \begin{cases} \hat{B}\boldsymbol{\tau} = \widehat{M}\boldsymbol{u} + \widehat{\boldsymbol{h}}_k \\ i = 1, 2, \dots, s: \begin{cases} N_i > N_L \\ |\boldsymbol{f}_i| < \zeta N_i \end{cases} \\ j = 1,2, \dots, p : \tau_{\min} < \tau_j < \tau_{\max} \end{cases}, \quad (26)$$

where we have used the limit $N_L > 0$ to prevent normal force to become too much close to zero. Also instead of μ_s we used $0 < \zeta < \mu_k < \mu_s$, where μ_k is the kinetic coefficient of friction. Now the controller stops the slipping if it occurs between two consecutive computation of $\boldsymbol{\tau}$. Note that N_L, ζ and $\tau_{\min}$,$\tau_{\max}$ can be different for different feet and actuators.

3.2. *Trajectory planning and dynamic stability*

So far, we have assumed that the desired trajectories of all the generalized coordinates were planned. In practice, a desired final position and orientation for the end-effector is given and the required joint configuration must be determined. Therefore, we first need to determine the final configuration of the robot, and then plan the trajectories of all generalized coordinates from the initial to final configuration while maintaining the dynamic stability of the robot. We assume that the robot is statically stable in its initial configuration.

We denote the desired final position of the end-effector by r_{D_f} and its desired final orientation by $[\alpha_f \quad \beta_f \quad \gamma_f]^T$. Since the inverse kinematics problem for legged robots in stance phase, usually has infinite solutions, one can find the final joint configuration which minimizes a cost function. We denote the final values of the generalized coordinates and the actuation vector by $\boldsymbol{q}_f$ and $\boldsymbol{\tau}_f$, respectively. We assume that the robot is going to be at rest in the final configuration, therefore: $\dot{\boldsymbol{q}}_f$and $\ddot{\boldsymbol{q}}_f$ are both zero. Thus, the cost function can be a general function of $\boldsymbol{q}_f$ and $\boldsymbol{\tau}_f$. We will ensure that a static stability criterion is satisfied in the final configuration. The static stability criterions are functions of the configuration of the robot and the external forces/torques [19][20]; therefore one can formulate them as a constraint for the optimization problem in the final configuration. Moreover, all the feet must be in contact with the ground and must not slip in the final configuration. Also, joint limits must be accounted. Therefore, according to (2), (6), and (7), we solve an offline optimization problem to find a suitable joint configuration to realize the mentioned requirements:

$$\min g(\boldsymbol{q}_f, \boldsymbol{\tau}_f)$$
$$\text{s.t.:} \begin{cases} \hat{B}\boldsymbol{\tau}_f = \hat{\boldsymbol{h}}_k \\ i = 1, 2, \dots, s: \begin{cases} N_i > N_L \\ |\boldsymbol{f}_i| < \zeta N_i \end{cases} \\ j = 1, 2, \dots, p: \tau_{\min} < \tau_f(j) < \tau_{\max} \\ v = 1, \dots, n: q_f(v)_{\min} < q_f(v) < q_f(v)_{\max} \\ \boldsymbol{c}_1(\boldsymbol{q}_f) = \boldsymbol{0} \\ \boldsymbol{c}_2(\boldsymbol{q}_f) = \boldsymbol{r}_{D_f} \\ \boldsymbol{c}_3(\boldsymbol{q}_f) = [\alpha_f \quad \beta_f \quad \gamma_f]^T \\ \text{static stability criterion} \end{cases} \tag{27}$$

The result is the final joint configuration and the actuation vector in this configuration.

Now that we have the initial and final values of $\boldsymbol{q}$, the desired trajectories of all the generalized coordinates, $\boldsymbol{q}_d$, can be planned to be tracked in the controller. Note that in planning the trajectories from the initial to the final point, the dynamic stability of the robot must be considered using a convenient method. Some of these methods are discussed in [21]–[25].

However, one might plan the trajectories without paying attention to the dynamic stability, using polynomial functions or other methods such as potential fields [26]–[28]. In this case, there might exist regions between the initial and final configuration in which the dynamic stability of the robot is violated. Note that the robot is statically stable in the initial and final configurations. The controller in (26) computes a vector $\boldsymbol{\tau}$ in each loop that ensures that the magnitude of the normal force for all the feet is at least equal to N_L and that the feet do not slip. Thus, when the robot is passing through a dynamically unstable region, between each two consecutive computation of $\boldsymbol{\tau}$, the magnitude of the normal force under unstable feet becomes less than N_L. If the next $\boldsymbol{\tau}$ is computed and applied to the robot before the magnitude of the normal force reaches to zero, the contact will be maintained. However, sudden changes in the magnitude of the normal forces is not desired. It is clear that the success in passing through the unstable regions with less undesirable effects depends on the fast computation of $\boldsymbol{\tau}$ in each loop in(26).

A flowchart representation of the control procedure is shown in Fig. 3.

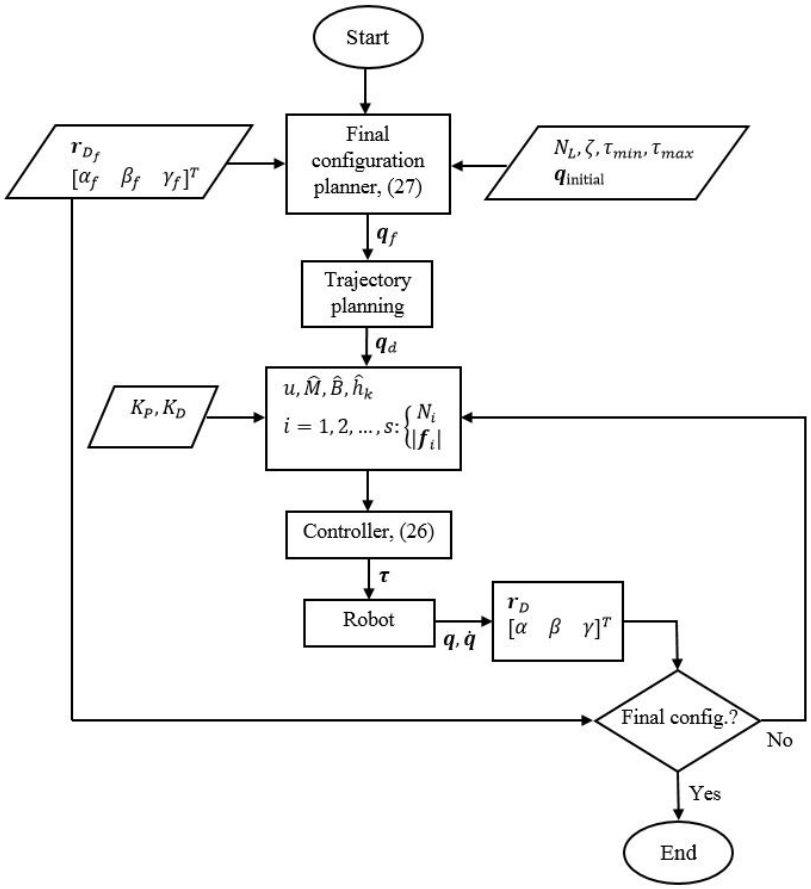

Fig. 3. Flowchart of the control system.

4. Simulation

In this section we implement the discussed method for the model of a quadruped robot that has a mounted 2-DOF serial manipulator on its body. The position and force control of the end-effector is to be done in the 2D plane. To keep the simulation as simple as possible, we assume that the motion of the front legs is generated synchronically. If we assume the same for rear legs, we can model the robot in the 2D plane. The schematic of the robot in an arbitrary configuration is depicted inFig. 4, where the coordinate systems and the generalized coordinates are also shown.The unit normal vector of the plane is:

$$[n_x \quad n_y]^T = [-\cos \pi/6 \quad -\sin \pi/6]^T \qquad (28)$$

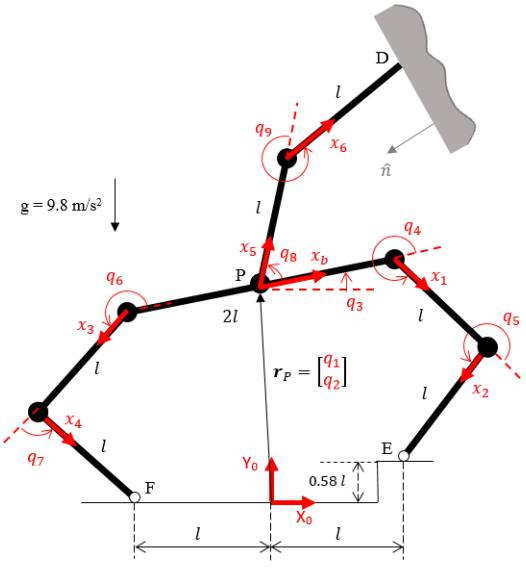

Fig. 4. The 2-D model of a quadruped robot with a mounted 2-DOF serial manipulator. Point ***D*** is in contact with the

plane whose unit normal vector is $\hat{\boldsymbol{n}}$. The inertial coordinate system and attached local coordinate systems to the links are depicted. The definition of generalized coordinates are also shown.

In the initial configuration the tip of the end-effector is in touch with the plane, but no force is applied to it.The values of the generalized coordinates in the initial configuration is given inTable 1. It is desired to apply a 10 N normal force on the plane in the final configuration, while moving the tip of the end-effector 0.05 m upward along the plane. The stiffness of the environment is 10^3 N/m. Therefore, the tip of the end-effector must have a 0.01 m normal displacement into the environment. The initial position of the tip of the end-effector is $[0.073 \quad 0.556]^T$ m and its desired final position is $[0.057 \quad 0.604]^T$ m. To decide on final values of $\boldsymbol{q}$, we set$\boldsymbol{h} = \boldsymbol{h}_k$ in (27) and solved the offline optimization. Norm-2 of $\boldsymbol{\tau}$was used as the cost function. The resulted final configuration is given in Table 1. The values of N_L and ζ for all legs were set to 10 N and 0.2, respectively. The kinematic constraint equations (2) for this robot consist of two vector loops:

$$\begin{aligned}\begin{bmatrix}x_E\\ y_E\end{bmatrix} = \begin{bmatrix}l\\ 0\end{bmatrix} = \begin{bmatrix}q_1\\ q_2\end{bmatrix} + {}^0_BR\begin{bmatrix}1\\ 0\end{bmatrix}l + {}^0_1R\begin{bmatrix}1\\ 0\end{bmatrix}l \\ + {}^0_2R\begin{bmatrix}1\\ 0\end{bmatrix}l \\ \begin{bmatrix}x_F\\ y_F\end{bmatrix} = \begin{bmatrix}-l\\ 0\end{bmatrix} = \begin{bmatrix}q_1\\ q_2\end{bmatrix} - {}^0_BR\begin{bmatrix}1\\ 0\end{bmatrix}l + {}^0_3R\begin{bmatrix}1\\ 0\end{bmatrix}l \\ + {}^0_4R\begin{bmatrix}1\\ 0\end{bmatrix}l\end{aligned} \tag{29}$$

In these relations, 2-by-2 rotation matrices are used to write the unit vectors in the reference frame. For example ${}^0_bR = R_z(q_3)$ and ${}^0_4R = {}^0_bR\,{}^b_3R\,{}^3_4R$, where ${}^3_4R = R_z(q_7)$ and ${}^b_3R = R_z(q_6)$.For this robot, the formulation of (6) is:

$$\begin{bmatrix}q_1\\ q_2\end{bmatrix} + l_5\,{}^0_5R + l_6\,{}^0_6R = \boldsymbol{r}_D \tag{30}$$

The desired orientation of the end-effector is not specified in the final configuration, therefore there is no related constraint in the form of (7) in (26). All of the revolute joints are active in which $\tau_{\min}$ and $\tau_{\max}$ are set to -10 Nm and 10 Nm, respectively. The body is a uniform slender bar of 2 kg mass. All other links are also uniform slender bars and each of them is 1 kg. Length l inFig. 4is 0.2 m. Our static stability criterion was to locate the projection of the center of the mass on the ground to be in the range of $-l/3$ and $l/3$. The obtained final configuration of the robot is given in Table 1, the time derivative of the generalized coordinates in the initial and final configuration are zero. The kinetic and potential energies of all the links can be written in terms of the generalized coordinates. Then the equations of the motion for the robot in (11) can be obtained, where $\partial \boldsymbol{r}_D/\partial \boldsymbol{q}$ is obtained from (30). In the controller, the P and D coefficients were set to 300 and 120 for all active generalized coordinates. These values of P and D coefficients were obtained by trial-and-error to achieve a fast response with minimum overshoot and steady-state error. The trajectories of the generalized coordinates from their initial to final values were obtained using fourth order polynomials. The first and second time derivatives of all generalized coordinates were assumed to be zero at $t = 0.60$ s in planning the desired trajectories. After $t = 0.6\ s$, all the generalized coordinates equal to their desired final values.

Table 1. Generalized coordinates values in the initial and final configurations. The first two elements are in meter and others are in radian.

	$\boldsymbol{q}$
Initial config.	[0.000 0.283 0.00 5.67 4.36 3.93 1.57 2.09 4.71]^T
Final config.	[-0.003 0.400 0.01 5.23 5.25 4.80 -0.18 2.29 4.27]^T

5. Results and Discussion

The problem was simulated for 2.00 s where the control signal was updated at the rate of 200 Hz. The same rate was used to feed the values of the desired interpolated trajectories to the controller. All of the optimizations were solved using "fmincon" function with interior-point algorithm in MATLAB®. In these optimizations, the lower bound on the size of the step was equal to 10^{-10} and the lower bound on the change in the value of the cost function during a step was equal to 10^{-6}. For the offline optimization in (27), the optimization was stopped after 32 iterations because the change in the value of the cost function during the last step was less than 10^{-6}. The value of the cost function in each step is depicted in Fig. 5.

The differential equations in (19) were solved using fourth order Runge-Kutta method. Reaching the desired position of the point D ensures that it has moved 0.05 cm along the plane and applied a 10 N normal force to it in the final configuration. The desired and actual trajectories of x and y components of point D is shown in Fig. 6 and Fig. 7, respectively. The desired and actual trajectory of point D in the xy plane is depicted in Fig. 8.The actuation effort for all of actuators is depicted in Fig. 9. Their values are within the range of -10 to 10 Nm, so none of them were saturated. Note that in Fig. 9, $\tau(i)$ acts on q_{i+3} for $i = 1, 2, \dots, 6$ depicted in Fig. 4.

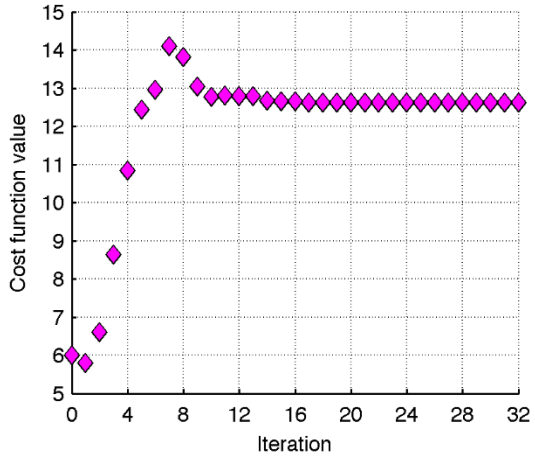

Fig. 5. The values of cost function, $|\boldsymbol{\tau_f}|$, in each iteration in the offline optimization (). During the last step, the change in the value is less than 10^{-6}.

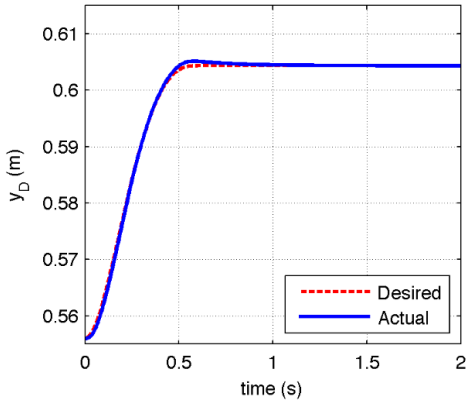

Fig. 7. The desired and actual trajectories of $\boldsymbol{y}$ component of point $\boldsymbol{D}$.

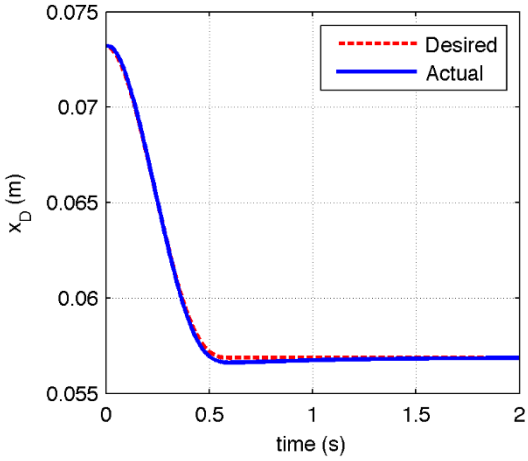

Fig. 6. The desired and actual trajectories of $\boldsymbol{x}$ component of point $\boldsymbol{D}$.

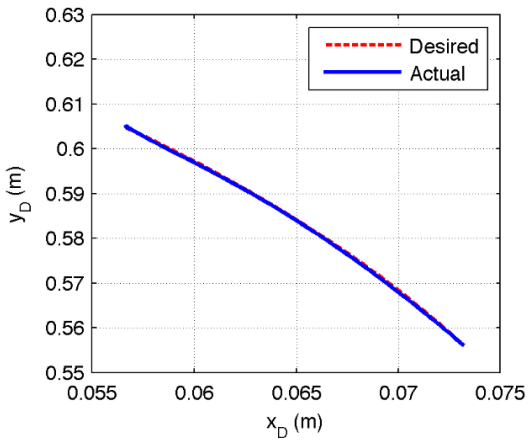

Fig. 8. The desired and actual trajectories of point $\boldsymbol{D}$ in the $\boldsymbol{xy}$ plane.

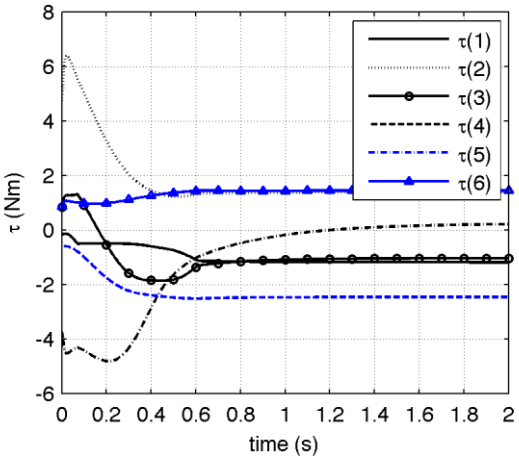

Fig. 9. Trajectories of computed $\boldsymbol{\tau}$ in the controller. For $\boldsymbol{i} = \boldsymbol{1}, \boldsymbol{2}, \ldots, \boldsymbol{6}$, $\boldsymbol{\tau(i)}$ acts on $\boldsymbol{q_{i+3}}$ in Fig. 4.

However, tracking the desired position without paying attention to the force requirements of the feet on the ground was meaningless. An important task of the controller was to keep all the feet in contact with the ground. The normal force at points E and F during the simulation are depicted in Fig. 10. Their values are greater than $N_L = 10$ N. The other crucial task was to prevent slipping. Applying the computed torques in the controller must result in the friction coefficients at points E and F, i.e. f_E/N_E and f_F/N_F, to be in the range of -0.2 to 0.2 for all instants. The trajectories of these coefficients are depicted in Fig. 11. In the initial configuration there is no slipping in points E and F. It is clear from Fig. 11that if position tracking was not subjected to no-slipping conditions in the controller, point F would have slipped immediately. The controller has produced $\boldsymbol{\tau}$ in each loop that keeps f/N in both points, E and F, between -0.2 and0.2.

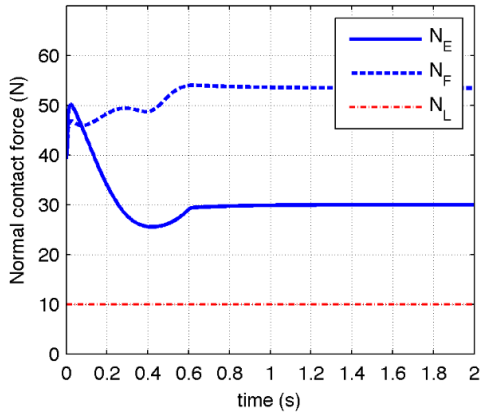

Fig. 10. Normal reaction forces in points E and F. Their values are greater than $\boldsymbol{N_L = 10}$ N that was set in the controller.

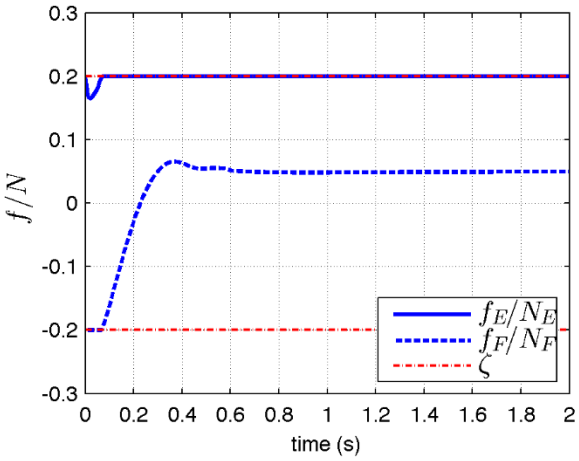

Fig. 11. The fraction $\boldsymbol{f/N}$ for points E and F during the simulation. This fraction must is kept between $\boldsymbol{\zeta = -0.2}$ and $\boldsymbol{\zeta = 0.2}$ to prevent or stop slipping if it occurs.

6. Conclusion

The position/force control for the end-effector of a general serial manipulator mounted on a legged robot was discussed using stiffness control approach. Active joints in both the manipulator and the legs were used in the control. The task was to apply a desired normal force to a specified plane, while moving the end-effector tip over some desired trajectories on the plane. Assuming stiffness of the target plane is known, tracking the desired trajectories meant satisfying both force and position requirements. The equations of motion of the robot were obtained using Lagrange's method. In the equations it was assumed that the ground was rigid and the contact points of the feet on the ground were fixed. We eliminated the Lagrange's multipliers to obtain a suitable form for position tracking purpose. Because the Lagrange's multipliers were functions of the feet contact forces, we recovered them in terms of the generalized coordinates, their first time-derivatives and the actuation effort. The resulted contact force functions in the controller were then used to compute an actuation vector which kept the contacts of the feet with the ground and prevented them from slipping. An optimization problem was solved in each loop of the control to minimize the actuation effort. The optimization was subject to position tracking of the end-effector, force requirements of the feet and actuators capabilities. Simulations were conducted for a simplified model of a quadruped robot equipped with a 2-DOF serial manipulator. It was desired to apply a 10 N normal force to a plane while moving the end-effector's tip 5 cm upward on the plane. It was shown that the reaction of the normal force, exerted to the

robot, was enough to make it slip on the ground. Therefore, tracking the desired trajectories for the end-effector was impossible without preventing the feet from slipping. The position/force control task was accomplished successfully because the generated actuation torques were computed regarding both requirements of the end-effector and the feet on the ground.

7. References

[1] J. Salisbury, "Active stiffness control of a manipulator in cartesian coordinates," in *Proc. IEEE Conf. on Decision and Control (CDC)*, 1980, vol. 19, pp. 95–100.

[2] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot Manipulator Control: Theory and Practice*. Taylor & Francis, 2003.

[3] M. H. Raibert and J. J. Craig, "Hybrid Position/Force Control of Manipulators," *J. Dyn. Syst. Meas. Control*, vol. 103, no. 2, p. 126, 1981.

[4] T. Yoshikawa, "Dynamic hybrid position/force control of robot manipulators--Description of hand constraints and calculation of joint driving force," *IEEE Journal on Robotics and Automation*, vol. 3, no. 5. pp. 386–392, 1987.

[5] Q. Huang and R. Enomoto, "Hybrid position, posture, force and moment control of robot manipulators," *2008 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2008*, no. 1, pp. 1444–1450, 2008.

[6] A. C. Leite, F. Lizarralde, and Liu Hsu, "A cascaded-based hybrid position-force control for robot manipulators with nonnegligible dynamics," *Proc. 2010 Am. Control Conf.*, pp. 5260–5265, 2010.

[7] M. H. Raibert, *Legged Robots that Balance*. MIT Press, 1986.

[8] M. Mistry, J. Buchli, and S. Schaal, "Inverse dynamics control of floating base systems using orthogonal decomposition," *2010 IEEE Int. Conf. Robot. Autom.*, no. 3, pp. 3406–3412, 2010.

[9] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan, and S. Schaal, "Optimal distribution of contact forces with inverse-dynamics control," *Int. J. Rob. Res.*, vol. 32, no. 3, pp. 280–298, 2013.

[10] L. Righetti, M. Mistry, j. Buchli, and S. Schaal, "Inverse Dynamics Control of Floating-Base Robots With External Contraints: an Unified View," *2011 Ieee Int. Conf. Robot. Autom.*, pp. 1085–1090, 2011.

[11] Z. Li, S. S. Ge, and S. Liu, "Contact-force distribution optimization and control for quadruped robots using both gradient and adaptive neural networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 8, pp. 1460–1473, 2014.

[12] F. T. Cheng and D. E. Orin, "Optimal Force Distribution in Multiple-Chain Robotic Systems," *IEEE Trans. Syst. Man Cybern.*, vol. 21, no. 1, pp. 13–24, 1991.

[13] H. Takemura, M. Deguchi, J. Ueda, Y. Matsumoto, and T. Ogasawara, "Slip-adaptive walk of quadruped robot," *Rob. Auton. Syst.*, vol. 53, no. 2, pp. 124–141, 2005.

[14] B. U. Rehman, M. Focchi, J. Lee, H. Dallali, D. G. Caldwell, and C. Semini, "Towards a multi-legged mobile manipulator," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2016–June, pp. 3618–3624, 2016.

[15] L. Righetti, M. Kalakrishnan, P. Pastor, J. Binney, J. Kelly, R. C. Voorhies, G. S. Sukhatme, and S. Schaal, "An autonomous manipulation system based on force control and optimization," *Auton. Robots*, vol. 36, no. 1–2, pp. 11–30, 2014.

[16] H. Hemami and F. C. Weimer, "Modeling of Nonholonomic Dynamic Systems With Applications," *J. Appl. Mech.*, vol. 48, no. 1, pp. 177–182, Mar. 1981.

[17] R. P. Singh, "Singular Value Decomposition for Constrained Dynamical Systems," *J. Appl. Mech.*, vol. 52, no. 4, p. 943, 1985.

[18] S. S. Kim and M. J. Vanderploeg, "QR Decomposition for State Space Representation of Constrained Mechanical Dynamic Systems," *J. Mech. Transm. Autom. Des.*, vol. 108, no. 2, pp. 183–188, Jun. 1986.

[19] P. V Nagy, S. Desa, and W. L. Whittaker, "Energy-based stability measures for reliable locomotion of statically stable walkers: Theory and application," *Int. J. Rob. Res.*, vol. 13, no. 3, pp. 272–287, 1994.

[20] S. Hirose, H. Tsukagoshi, and K. Yoneda, "Normalized energy stability margin and its contour of walking vehicles on rough terrain," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 2001, vol. 1, pp. 181–186.

[21] S. Zhang, J. Gao, X. Duan, H. Li, Z. Yu, X. Chen, J. Li, H. Liu, X. Li, Y. Liu, and Z. Xu, "Trot pattern generation for quadruped robot based on the ZMP stability margin," *2013 ICME Int. Conf. Complex Med. Eng. C. 2013*, pp. 608–613, 2013.

[22] B.-S. Lin and S.-M. Song, "Dynamic modeling, stability and energy efficiency of a quadruped walking machine," *IEEE Int. Conf. Robot. Autom.*, pp. 367–373, 1993.

[23] K. Yoneda and S. Hirose, "Tumble stability criterion of integrated locomotion and manipulation," in *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, 1996, vol. 2, pp. 870–

876 vol.2.

[24] E. Garcia and P. G. de Santos, "An improved energy stability margin for walking machines subject to dynamic effects," *Robotica*, vol. 23, no. 1, pp. 13–20, 2005.

[25] D. Zhou, K. H. Low, and T. Zielinska, "A stability analysis of walking robots based on leg-end supporting moments," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 2000, vol. 3, pp. 2834–2839 vol.3.

[26] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Trans. Syst. Man. Cybern.*, vol. 22, no. 2, pp. 224–241, 1992.

[27] O. Montiel, U. Orozco-Rosas, and R. Sepúlveda, "Path planning for mobile robots using Bacterial Potential Field for avoiding static and dynamic obstacles," *Expert Syst. Appl.*, vol. 42, no. 12, pp. 5177–5191, 2015.

[28] J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization.," in *Robotics: science and systems*, 2013, vol. 9, no. 1, pp. 1–10.