Robotica (2003) volume 21, pp. 45–57. © 2003 Cambridge University Press
DOI: 10.1017/S0263574702004551 Printed in the United Kingdom

Control of foot placement, forward velocity and body orientation of a one-legged hopping robot

J. Vermeulen,* D. Lefeber** & B. Verrelst†

Vrije Universiteit Brussel, Department of Mechanical Engineering, Pleinlaan 2, 1050 Brussels (Belgium).
E-mail: Jimmy.Vermeulen@vub.ac.be

(Received in Final Form: September 4, 2002)

SUMMARY
This paper intends to contribute to the study of dynamically balanced legged robots. A real-time applicable control algorithm for a planar one-legged robot is developed, which allows for locomotion on an irregular terrain. The simulated model consists of an articulated leg and a body, vertically placed upon the leg. During the stance phase the leg is supported by a massless foot. The algorithm is based on the choice of a number of objective locomotion parameters which can be changed from one hop to another. From a chosen initial configuration the robot is able to transfer to a chosen end configuration, while simultaneously controlling its forward velocity, its step length and its stepping height. The foot is thus being placed exactly on a chosen foothold. To reach this goal, the actuators track polynomial functions. The calculation of these functions is based on the objective parameters, and takes into account the constraints acting on the robot. These constraints result from the fact that during flight the center of gravity of the robot tracks a parabolic trajectory, and that the angular momentum with respect to the center of gravity is conserved. Writing the angular momentum constraint in a Caplygin form is the key to the algorithm. Promising simulation results for the algorithm are shown for two different experiments.

KEYWORDS: Running robots; Motion on irregular terrain; Objective locomotion parameters.

1. INTRODUCTION: LEGGED LOCOMOTION

1.1. Walking versus running

In the field of mobile robotics, various applications do not provide a continuous and smooth path of support for the robot. In these cases of locomotion on a so-called irregular terrain, legged robots demonstrate better mobility and versatility when compared to tracked or wheeled vehicles. However, when it comes to autonomous capability of a robot, mobility should still be increased. Extensive research is needed before legged robots can be widely and practically used. During the last decades, many legged machines have been built to study the principles of legged locomotion (see 1.2). These machines can be classified into two main categories, viz. statically balanced walking machines and dynamically balanced machines. Moreover, the dynamically balanced machines can be classified into dynamic walking robots and running robots. The main difference between walking and running originates from the fact that during walking at least one of the legs supporting the robot's body is in contact with the ground, whereas during running this condition must not be satisfied. These restrictions on the motion of body and legs of a walking machine imply that running machines can attain higher velocities and can take steps with a greater length and a greater height. Besides, because of the presence of ballistic flight phases, a running machine is able to jump over an obstacle, instead of trying to avoid the obstacle by making a turn. Until now, little research has been done on running robots, when compared to the research on walking robots. Most of the work is focused on steady-state locomotion on a flat terrain, although in the real world a terrain is, in general, of an irregular shape. Despite the great potential of running and jumping machines, their control is a difficult problem. They have highly nonlinear dynamics, and in the case of monopeds and bipeds, they are statically unstable. A statically stable walking robot can be controlled in a kinematic way by ensuring that the projection of the center of gravity lies in the polygon formed by the contour of the supporting feet. This is possible as long as the robot moves slowly and the inertial forces can be neglected. On the contrary, a running robot can only be balanced through its motion, by taking into account and manipulating in the right way the inertial forces. In this paper a control algorithm for a one-legged hopping robot is developed, which allows for locomotion on an irregular terrain. This is achieved by using a goal-directed inverse dynamical approach. A number of objective parameters, viz. step length, stepping height, and forward velocity during flight, are used as steering variables. These steering variables describe the motion from an external point of view rather than in terms of the internal joint angles. Expressing the locomotion in function of objective parameters is a strategy commonly used in the domain of animation.[1,2]

1.2. Chronological overview

The robots developed in the United States by M.H. Raibert and his team are probably the best known dynamically

* Corresponding author.
** E-mail: dlefeber@vub.ac.be
† E-mail: bverrels@vub.ac.be

balanced robots and the only really running ones, because of their hopping behavior. The basic control algorithm used in all his robots consists of three decoupled parts: energy stored in a hydraulic spring in the legs is modulated to manipulate hopping height, forward speed is controlled by positioning the legs during the flight phase; and body attitude is regulated during the stance phase. Based on this principle a one-legged robot hopping in 2D was constructed,[3] a one-legged robot hopping in 3D,[4] a running robot on four legs,[5,6], a bipedal robot running and being able to execute a forward flip in 2D,[7,8] as well as a bipedal robot running and executing a somersault in 3D.[9] Later, many researchers showed a renewed interest in Raibert's one-legged hopping robot, mostly focusing on its stability from the mathematical and system theory point of view. J.J. Helferty et al. studied the feasibility of using neural networks to control vertical hopping, resulting in a stable limit cycle in the state space of the robot.[10] Z. Li et al. also studied the robot's limit cycle, using the so called energy-balance method and compared this method to the discrete dynamical system theory.[11] They also showed that using the angular momentum constraint, it is possible to control the orientation of a one-legged hopping robot during its flight phase, an observation that will be used in this paper too.[12] Further studies on the limit cycle of a hopping robot have been made by D.E. Koditschek et al. They showed that the theory of uni-modal return maps can be applied to the dynamics of a one-legged robot. They modelled the robot as a point mass supported by a massless leg acting as a pneumatic spring and studied vertical hopping only.[13] R.T. M'Closkey et al. showed that the same model exhibits period doubling and chaotic behavior, both in 1-dimensional and 2-dimensional hopping.[14–16] A more generally valid feedback control algorithm based on return maps was developed by J.P. Ostrowski et al. and again applied to a hopping robot.[17] Gregorio et al. also studied Raibert's one-legged hopper, and actuated it with electrical DC-motors and a ball screw. Even though making use of low power electrical actuators, they managed to build a fast running robot, being more energy efficient than previously built hopping robots.[18,19] In the late nineties, more and more researchers started looking for control algorithms which were more applicable for locomotion on an irregular terrain than those compared to Raibert's algorithm. De Man et al. developed a control algorithm for a one-legged hopping robot with a telescopic leg, making it possible to change a number of objective locomotion parameters from one hop to another, and thus allowing for locomotion on an irregular terrain.[20] Later, the algorithm was also applied to a one-legged hopper with an articulated leg.[21] Both simulated models had an upper body with its center of mass located at the hip, resulting in a decoupled motion of body and leg. They differ from Raibert's models in that sense that the actuators track continuous polynomial functions, which guarantee that the desired values of the objective parameters are attained. The same principle has also been applied by Chevallereau et al.[22] Recently, a promising project started in Germany under the coordination of F. Pfeiffer. His team designed a bipedal robot who is able to walk and will soon be able to jog too.[23]

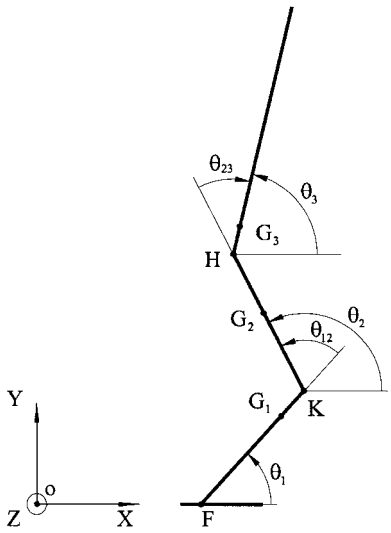

Fig. 1. Robot geometry.

2. THE DYNAMICAL MODEL

Figure 1 depicts the robot geometry, while the inertial parameters are given in Table I. To be able to study the conceptual features of a running machine, such as its under-actuated and non-holonomic nature, without unnecessarily increasing the complexity, only one leg is considered. For simplicity reasons the motion is restricted to the sagittal plane. Our model is a planar multibody system, consisting of an articulated leg, a body and a massless foot. The length of the i-th link is l_i, its mass is m_i and the moment of inertia around its center of mass G_i is I_i. The angles between the horizontal and the different links of the robot are θ_1, θ_2 and θ_3. The relative angle between lower and upper leg is θ_{12}, and the relative angle between upper leg and body is θ_{23}. It is clear that $\theta_{12} = \theta_2 - \theta_1$ and $\theta_{23} = \theta_3 - \theta_2$. Point F represents the connection between lower leg and foot. This point coincides with the ankle joint, but is called F since during flight the foot is considered as a point. Point K represents the robot's knee and point H represents its hip. The locations of the center of mass G_1 of the lower leg, G_2 of the upper leg and G_3 of the body are given by $FG_1 = \alpha l_1$, $KG_2 = \beta l_2$ and $HG_3 = \gamma l_3$, where $0 < \alpha, \beta, \gamma < 1$.

The robot has three actuators, which are placed at respectively, the hip, the knee and the ankle joint. During the ballistic flight phase, it possesses 5 DOF and only 2 useful actuators, i.e. at hip and knee. This means our model is under-actuated during that phase. It is, however, subjected to a number of constraints, viz. two holonomic constraints resulting from the fact that the global center of gravity (COG) tracks a parabolic trajectory, and one non-holonomic constraint resulting from the angular momentum conservation. During the stance phase the robot has 3 DOF. This assumes a non-slippery, rigid ground. During this phase, the

Table I. Inertial parameters of the robot.

i	$l_i(m)$	$m_i(kg)$	$I_i(kgm^2/s)$
1	0.342	1.781	0.0138
2	0.308	1.373	0.0218
3	0.666	8.507	0.7979

https://doi.org/10.1017/S0263574702004551 Published online by Cambridge University Press

robot is fully actuated since the ankle actuator can also be used. Thus, the model is characterized by different mathematical descriptions in non-overlapping regions of the state space. The transaction between the flight phase and the stance phase, which is referred to as the impact phase, is modelled as an inelastic impulsive impact.[24,25] Five generalized coordinates define the machine's configuration space, viz. the absolute angle of the body with respect to the horizontal axis θ_3, the relative angles at hip and knee, θ_{23} and θ_{12} respectively, and the coordinates of the global COG, X_G and Y_G. The reduced configuration space, obtained by reduction to the COG, is formed by $\{\theta_{12}, \theta_{23}, \theta_3\}$. During the stance phase the shape space is three-dimensional, whereas during flight it is two-dimensional. Despite its underactuated nature during the flight phase, we encounter no problems concerning accessibility and controllability. In the case of a zero angular momentum, the model is proven to be fully controllable, since the Lie algebra formulated in the reduced configuration space has a full rank.[12,26,27] With a non-zero angular momentum, a constant drift term is introduced in the control system. In that case the system is still accessible and also Small Time Locally Controllable (STLC) if the controls are sufficiently large.[28,29] Since our algorithm is based (indirectly) on the choice of an appropriate value of the angular momentum with respect to G, the drift term always helps the controls.

3. THE CONTROL ALGORITHM

3.1. General description

The main objective of this work is the development of an algorithm that allows a one-legged robot to simultaneously control its forward velocity during flight and the placement of its foot on desired footholds, which means controlling the step length and stepping height. The goal is to make the robot move from a given initial condition (take-off) to a given end-condition (touch-down) while attaining the desired values of the objective locomotion parameters. Since these values are completely determined by the take-off conditions, the crucial task of the control algorithm is to determine these conditions and to choose the control functions for the actuators in such a way as to guarantee the desired behavior of the jumping mechanism during flight. Moreover, any drift in the upper body motion should be avoided. Although the robot is fully actuated during the stance phase, the torques exerted by the actuators are limited. Special attention has to be given to the torque exerted at the ankle joint, since foot rotation should be avoided.[30] Therefore, by choosing (indirectly) an appropriate value for the angular momentum about the COG during flight, the rotation of the upper body is kept within certain limits, such that the induced rotation can be compensated during the stance phase. To reach these goals, the dynamics of the robot are controlled in a kinematic way, resulting in a dynamically stable behavior. During flight, two polynomial trajectories are generated to be tracked by the actuators at hip and knee, respectively. During the stance one more polynomial function is added, to be tracked by the actuator at the ankle joint.

3.2. Flight phase

3.2.1. Kinematics. During flight the position of the global COG is given by:

$$X_G = X_F + a\cos(\theta_3 - \theta_{12} - \theta_{23}) + b\cos(\theta_3 - \theta_{23}) + c\cos\theta_3 \quad (1)$$

$$Y_G = Y_F + a\sin(\theta_3 - \theta_{12} - \theta_{23}) + b\sin(\theta_3 - \theta_{23}) + c\sin\theta_3 \quad (2)$$

with:

$$a = (\alpha\eta_1 + \eta_2 + \eta_3)l_1$$
$$b = (\beta\eta_2 + \eta_3)l_2$$
$$c = \gamma\eta_3 l_3$$

and:

$$\eta_i = \frac{m_i}{\sum_{i=1}^{3} m_i}$$

The angular momentum with respect to G can be written as:

$$\mu_G = A_3\dot{\theta}_3 + A_{23}\dot{\theta}_{23} + A_{12}\dot{\theta}_{12} \quad (3)$$

with:

$$A_3 = d_1 + d_2 + d_3 + 2e_{12}\cos\theta_{12} + 2e_{23}\cos\theta_{23} + 2e_{13}\cos(\theta_{12} + \theta_{23})$$
$$A_{23} = d_1 + d_2 + 2e_{12}\cos\theta_{12} + e_{23}\cos\theta_{23} + e_{13}\cos(\theta_{12} + \theta_{23})$$
$$A_{12} = d_1 + e_{12}\cos\theta_{12} + e_{13}\cos(\theta_{12} + \theta_{23})$$

and:

$$d_1 = I_1 + \frac{l_1^2}{M} m_1(m_2 + m_3)(1 - \alpha)^2$$

$$d_2 = I_2 + \frac{l_2^2}{M}\{\beta^2 m_1 m_2 + [m_1 + (1 - \beta)^2 m_2]m_3\}$$

$$d_3 = I_3 + \frac{l_3^2}{M}\gamma^2 m_3(m_1 + m_2)$$

$$e_{12} = \frac{l_1 l_2}{M} m_1(1 - \alpha)(\beta m_2 + m_3)$$

$$e_{13} = \frac{l_1 l_3}{M}\gamma m_1 m_3(1 - \alpha)$$

$$e_{23} = \frac{l_2 l_3}{M}\gamma m_3[m_1 + (1 - \beta)m_2]$$

$$M = m_1 + m_2 + m_3$$

Expression (3) represents a system with the structure of a three-dimensional Lagrangian with a cyclic coordinate, being θ_3. The coefficients A_{12}, A_{23} and A_3 are highly nonlinear (trigonometric). The expression can, however, be written in a Caplygin form, with $\dot{\theta}_3$ on the left-hand side and the right-hand side being independent of θ_3.

3.2.2. Dynamics. The dynamical behavior of the system is subjected to two time dependent holonomic constraints due

https://doi.org/10.1017/S0263574702004551 Published online by Cambridge University Press

to the parabolic trajectory of the COG, and one non-holonomic constraint due to the conservation of the angular momentum.

These constraints are written as:

$$\ddot{X}_G = 0 \tag{4}$$

$$\ddot{Y}_G = -g \tag{5}$$

$$\dot{\mu}_G = 0 \tag{6}$$

After integration over time this becomes:

$$\dot{X}_G = \dot{X}_G^{to} \tag{7}$$

$$\dot{Y}_G = -gt + \dot{Y}_G^{to} \tag{8}$$

$$\mu_G = \mu_G^{to} \tag{9}$$

and integrating (7) and (8) over time leads to:

$$X_G = \dot{X}_G^{to} t + X_G^{to} \tag{10}$$

$$Y_G = -g\frac{t^2}{2} + \dot{Y}_G^{to} t + Y_G^{to} \tag{11}$$

3.2.3. Control strategy

Choices. Suppose that the configuration of the robot at take-off and touch-down is chosen. This determines the values of:

- take-off: θ_{12}^{to}, θ_{23}^{to}, θ_3^{to}
- touch-down: θ_{12}^{td}, θ_{23}^{td}, θ_3^{td}

Three objective locomotion parameters are introduced, i.e.:

- Forward velocity: ν
- Step length: λ
- Stepping height: δ

When we suppose the foot is positioned in the origin of the coordinate system at the moment of take-off, we can write:

- $X_F^{to} = 0$
- $Y_F^{to} = 0$
- $X_F^{td} = \lambda$
- $Y_F^{td} = \delta$

Considering the foot doesn't slip at take-off, one obtains:

- $\dot{X}_F^{to} = 0$
- $\dot{Y}_F^{to} = 0$
- $\ddot{X}_F^{to} = 0$
- $\ddot{Y}_F^{to} = 0$

The velocity of the foot at touch-down determines the amount of kinetic energy lost during impact.[24,25,31] Performing touch-down, e.g. with the leg acting as one rigid body, introduces high energy losses during impact. Therefore, a kick action by the foot is introduced. This can be done by choosing an appropriate value of the velocity of the foot at touch-down. Consider that we choose the motion of the foot relative to motion of the COG:

- $\dot{X}_F^{td} = k_1 \dot{X}_G^{td}$
- $\dot{Y}_F^{td} = k_2 \dot{Y}_G^{td}$

where k_1 and k_2 can be freely chosen. Choosing k_1 and k_2 both equal to zero would cause the percussions to be zero and thus there would be no shock and no energy loss. However, it might be interesting to choose a negative value for $\dot{X}_F^{td}$, since this gives the COG a higher forward velocity after impact, which can reduce the ankle torque at the beginning of the stance phase. A shock in the vertical direction is, however, of no use, and thus we will choose k_2 equal to zero.[31]

The acceleration of the foot at touch-down determines the value of the ground reaction force immediately after impact. Let's choose the acceleration of the foot relative to the acceleration of the COG:

- $\ddot{X}_F^{td} = k_3 \ddot{X}_G^{td}$
- $\ddot{Y}_F^{td} = k_4 \ddot{Y}_G^{td}$

where k_3 and k_4 can be freely chosen. It is important that the vertical component of the resultant ground reaction force R_y after impact is large enough. The length of the foot limits the torque which can be exerted at the ankle joint (see 3.4). When R_y is higher, the torque which can be exerted without causing foot rotation is also higher. The vertical acceleration of the COG is, of course, $-g$. Choosing $k_4 > 1$ results in a vertical acceleration of the foot which is higher (in absolute value) than the acceleration of the COG. This introduces a pushing effect of the foot on the ground, which, in turn introduces a higher value of R_y. The influence of the horizontal acceleration of the foot is rather limited and since the robot is in free fall before impact, $k_3 = 0$ is used.

Further, note:

- take-off: $t^{to} = 0$
- touch-down: $t^{td} = T^{fl}$

Constructing the polynomial functions: Since the position of the foot and the configuration of the robot are known at take-off as well as at touch-down, equations (1) and (2) determine the values of X_G^{to}, Y_G^{to} and X_G^{td}, Y_G^{td}. Rewriting equation (10) by taking into account the choices we made above allows us to determine the value of the flight-time T^{fl}:

$$T^{fl} = \frac{X_G^{td} - X_G^{to}}{\nu} \tag{12}$$

Note that since $\nu \neq 0$, hopping in place cannot be achieved with this strategy. Hopping in place is, however, a type of motion for which easier strategies can be used.

Rewriting equation (11) at touch-down allows us to compute the necessary vertical velocity of G at take-off $\dot{Y}_G^{to}$:

$$\dot{Y}_G^{to} = \frac{Y_G^{td} - Y_G^{to}}{T^{fl}} + \frac{gT^{fl}}{2} \tag{13}$$

The hopping height is defined as the difference between the maximum height of G during flight and its height at take-off:

$$\Delta h = \frac{(\dot{Y}_G^{to})^2}{2g} \tag{14}$$

During the flight phase, the leg will swing forward, in order to position the foot on a chosen foothold. Because of the

https://doi.org/10.1017/S0263574702004551 Published online by Cambridge University Press

angular momentum constraint the body will rotate too. At touch-down the body will have orientation θ_3^{td} and an angular velocity $\dot{\theta}_3^{td}$. The take-off conditions will have to be chosen in such a way that the orientation and velocity of the body at touch-down equalize these desired values.

To reduce computation time, we use an approximation of the angular velocity $\dot{\theta}_3^{td}$ of the upper body at the instance of touch-down. We suppose that the body rotates with a constant angular velocity from θ_3^{to} to θ_3^{td}. Then we use the following approximation for $\dot{\theta}_3^{td}$:

$$\dot{\theta}_3^{td} \approx \frac{\Delta\theta_3^{fl}}{\Delta t^{fl}} = \frac{\theta_3^{td} - \theta_3^{to}}{T^{fl}} \tag{15}$$

Now, when we evaluate expressions (7), (8) and (9) at touch-down, these equations are solved for $\dot{\theta}_{12}^{td}$, $\dot{\theta}_{23}^{td}$ and μ_G^{to}.

Further, when evaluating expressions (7), (8) and (9) at take-off, this set can be solved for $\dot{\theta}_{12}^{to}$, $\dot{\theta}_{23}^{to}$ and $\dot{\theta}_3^{to}$.

Next, when evaluating (4), (5) and (6) at take-off, this set can be solved for $\ddot{\theta}_{12}^{to}$, $\ddot{\theta}_{23}^{to}$ and $\ddot{\theta}_3^{to}$. Finally, when evaluating (4), (5) and (6) at touch-down, this set can be solved for $\ddot{\theta}_{12}^{td}$, $\ddot{\theta}_{23}^{td}$ and $\ddot{\theta}_3^{td}$.

As a result of the preceding computations, the values for θ_{12} and θ_{23} at take-off and touch-down as well as their first and second derivatives are found, and are used to establish two fifth order polynomial functions for $\theta_{12}^{fl}(t)$ and $\theta_{23}^{fl}(t)$. These polynomial functions are the trajectories to be tracked during flight by the controllers at knee and hip, respectively. Further, after rewriting (3):

$$\dot{\theta}_3 = \frac{\mu_G^{to} - A_{23}\dot{\theta}_{23}^{fl} - A_{12}\dot{\theta}_{12}^{fl}}{A_3} \tag{16}$$

the real angle $\theta_{3,real}^{td}$ results from integrating (16) over time during flight:

$$\theta_{3,real}^{td} = \theta_3^{to} + \int_0^{T^{fl}} \left(\frac{\mu_G^{to} - A_{23}\dot{\theta}_{23}^{fl} - A_{12}\dot{\theta}_{12}^{fl}}{A_3} \right) dt \tag{17}$$

This angle will, in general, differ from the desired value of θ_3^{td}, since an approximated expression for $\dot{\theta}_3^{td}$ was used. If necessary, $\dot{\theta}_3^{td}$ can be adjusted iteratively:

$$\dot{\theta}_3^{td,n+1} = \dot{\theta}_3^{td,n} + \frac{1}{T^{fl}}(\theta_3^{td} - \theta_{3,real}^{td})$$

where θ_3^{td} is the desired value of the absolute angle of the body with respect to the horizontal axis at touch-down, and we repeat the above calculations. In fact, we do not have to reach exactly the desired value of θ_3^{td}. This angle is not an objective parameter and should not be considered as a goal of the algorithm. The angle at touch-down has to be acceptable, which means that we should be able to compensate for an eventual error during the stance phase. Simulations showed that in all cases a second iteration was sufficient. The error made by the approximation of $\dot{\theta}_3^{td}$ seemed to be rather small. The fast convergence of the iterative procedure makes the algorithm applicable for real-time control.

Note that expression (16) results from writing the angular momentum constraint in a Caplygin form. Writing (16) in this form assures that the right hand side is independent from θ_3. In this way, $\dot{\theta}_3$ can be isolated at the left hand side and the expression can be integrated for calculating θ_3^{td}. When writing the angular momentum constraint, e.g. in terms of absolute angles only, this integration would not be possible.

Pulling up the foot: The shape of the polynomial functions is completely determined by the beginning and end points only. We have no control on what happens in between. Hitting the ground by the foot should by all means be avoided. Therefore a correction on the polynomial functions is introduced. An intermediate point is added to the polynomial functions to make sure the foot reaches a certain maximum height at $t=t^*$, where t^* is the time step where G reaches its maximum height:

$$\dot{Y}_G(t^*) = 0 \tag{18}$$

From equation (8) it can be found that:

$$t^* = \frac{\dot{Y}_G^{to}}{g} \tag{19}$$

Two correction functions $C_{12}(t)$ and $C_{23}(t)$ will be added, which don't influence the boundary conditions at the beginning and end points of the polynomial functions, since the same values for all the objective locomotion parameters have to be reached:

$$\zeta_{12}(t) = \theta_{12}^{fl}(t) + C_{12}(t) = \theta_{12}^{fl}(t) + K_{12}f(t) \tag{20}$$

$$\zeta_{23}(t) = \theta_{23}^{fl}(t) + C_{23}(t) = \theta_{23}^{fl}(t) + K_{23}f(t) \tag{21}$$

with

$$f(t) = \left[1 - 3(t - t^*)\left(\frac{1}{t^*} - \frac{1}{T^{fl} - t^*} \right) \right] \frac{t^3(T^{fl} - t)^3}{t^{*3}(T^{fl} - t^*)^3}$$

and K_{12} and K_{23} being constants which have to be determined depending on the height the foot has to be pulled up to.

The correction functions have the following characteristics:

$$C_{ij}(0) = 0$$
$$\dot{C}_{ij}(0) = 0$$
$$\ddot{C}_{ij}(0) = 0$$
$$C_{ij}(T^{fl}) = 0$$
$$\dot{C}_{ij}(T^{fl}) = 0$$
$$\ddot{C}_{ij}(T^{fl}) = 0$$
$$C_{ij}(t^*) = K_{ij}$$
$$\dot{C}_{ij}(t^*) = 0$$

Identifying (2) with (11) and evaluating this expression at t^* gives the following:

$$Y_F^{des}{}^* + a\sin(\zeta_3^* - \zeta_{12}^* - \zeta_{23}^*) + b\sin(\zeta_3^* - \zeta_{23}^*)$$

$$+ c\sin\zeta_3^* = -\frac{g}{2}t^{*2} + \dot{Y}_G^{to}t^* + Y_G^{to} \tag{22}$$

https://doi.org/10.1017/S0263574702004551 Published online by Cambridge University Press

with $Y_F^{des}*$ being the desired height of the foot at t^*. The * indicates the evaluation of the functions ζ_{12} and ζ_{23} at time step t^*, and ζ_3^* represents the new value of the absolute angle of the body with respect to the horizontal axis at that time step.

After introducing (20) and (21) in (22) we obtain:

$$\begin{aligned} Y_F^{des}* &+ a\sin(\zeta_3^* - \theta_{12}^* - \theta_{23}^* - K_{12} - K_{23}) \\ &+ b\sin(\zeta_3^* - \theta_{23}^* - K_{23}) + c\sin\zeta_3^* \\ &= -\frac{g}{2}t^{*2} + \dot{Y}_G^{to}t^* + Y_G^{to} \end{aligned} \tag{23}$$

Further, identifying the first derivative of (2) with (8) and evaluating this expression at t^* leads to:

$$\begin{aligned} \dot{Y}_F^{des}* &+ a\cos(\zeta_3^* - \zeta_{12}^* - \zeta_{23}^*)(\dot{\zeta}_3^* - \dot{\zeta}_{12}^* - \dot{\zeta}_{23}^*) \\ &+ b\cos(\zeta_3^* - \zeta_{23}^*)(\dot{\zeta}_3^* - \dot{\zeta}_{23}^*) + c\cos\zeta_3^*\dot{\zeta}_3^* \\ &= -gt^* + \dot{Y}_G^{to} \end{aligned} \tag{24}$$

Taking into account (8) and (18) it is seen that the right hand side of (24) is equal to zero:

$$-gt^* + \dot{Y}_G^{to} = \dot{Y}_G(t^*) = 0$$

After introducing (20) and (21), equation (24) becomes:

$$\begin{aligned} &a\cos(\zeta_3^* - \theta_{12}^* - \theta_{23}^* - K_{12} - K_{23})(\dot{\zeta}_3^* - \dot{\theta}_{12}^* - \dot{\zeta}_{23}^*) \\ &+ b\cos(\zeta_3^* - \theta_{23}^* - K_{23})(\dot{\zeta}_3^* - \dot{\theta}_{23}^*) \\ &+ c\cos\zeta_3^*\dot{\zeta}_3^* + \dot{Y}_F^{des}* = 0 \end{aligned} \tag{25}$$

Equations (23) and (25) have to be solved for K_{12} and K_{23}. The problem is that the values of ζ_3^* and $\dot{\zeta}_3^*$ are unknown. They depend on the trajectories of ζ_{12} and ζ_{23}, which at this point are unknown. Therefore we assume that they don't change significantly because of the corrections on θ_{12} and θ_{23}:

$$\zeta_3^* \approx \theta_3^*$$
$$\dot{\zeta}_3^* \approx \dot{\theta}_3^*$$

Since we want to reach a maximum vertical height of the foot at t^*, we choose:

$$\dot{Y}_F^{des}* = 0$$

Finally, since c is small and since ζ_3^* fluctuates around $\frac{\pi}{2}$ we assume:

$$c\cos\zeta_3^*\dot{\zeta}_3^* \approx 0$$

Now the set of equations (23) and (25) has the following form:

$$A_1\sin\alpha_1 + A_2\sin\alpha_2 = C_1 \tag{26}$$
$$B_1\cos\alpha_1 + B_2\cos\alpha_2 = 0 \tag{27}$$

with:

$$A_1 = a$$
$$A_2 = b$$
$$C_1 = -\frac{g}{2}t^{*2} + \dot{Y}_G^{to}t^* + Y_G^{to} - c\sin\theta_3^* - Y_F^{des}$$
$$B_1 = a(\dot{\theta}_3^* - \dot{\theta}_{12}^* - \dot{\theta}_{23}^*)$$
$$B_2 = b(\dot{\theta}_3^* - \dot{\theta}_{23}^*)$$
$$\alpha_1 = \theta_3^* - \theta_{12}^* - \theta_{23}^* - K_{12} - K_{23}$$
$$\alpha_2 = \theta_3^* - \theta_{23}^* - K_{23}$$

This set of (26) and (27) can easily be solved by applying the following substitutions:

$$X_1 = \sin\alpha_1$$
$$X_2 = \sin\alpha_2$$
$$Y_1 = \cos\alpha_1$$
$$Y_2 = \cos\alpha_2$$

which leads to the following set:

$$X_1^2 + Y_1^2 = 1$$
$$X_2^2 + Y_2^2 = 1$$
$$A_1X_1 + A_2X_2 = C_1$$
$$B_1Y_1 + B_2Y_2 = 0$$

We can, for example, solve this set for X_2 by eliminating X_1, Y_1 and Y_2:

$$X_2^2(A_1^2B_2^2 - B_1^2A_2^2) + 2X_2A_2B_1^2C_1 + B_1^2A_1^2 - B_1^2C_1^2 - A_1^2B_2^2 = 0$$

which is a quadratic equation in X_2.

The correction functions have to be added before equation (16) is integrated. They do have an influence on the angle θ_3^{td}.

3.3. Impact phase

At the landing (touch-down), the foot of the hopper hits the ground. Let's model the foot as a point located at the ankle joint. This is a valid assumption, since the foot is considered massless and inertialess. We also assume the foot doesn't bounce back and doesn't slip, which means that it stays in contact with the ground. These are the assumptions corresponding to an inelastic impulsive impact.[24,25] Because of this impact phase, discontinuities in the velocity state variables will occur, caused by so called percussions. The configuration of the robot is assumed to stay unchanged. Let's denote the time step right before landing as t^- and indicate all variables at that time step with a $-$. Analogously, we denote the time step after landing as t^+ and indicate all variables at that time step with a $+$. In fact, t^- is the same as t^{td}, being the end of the flight phase, and t^+ is the beginning of the stance phase. On the configuration level, we have the following:

$$\theta_{12}^+ = \theta_{12}^-$$
$$\theta_{23}^+ = \theta_{23}^-$$
$$\theta_3^+ = \theta_3^-$$
$$X_F^+ = X_F^-$$
$$Y_F^+ = Y_F^-$$

https://doi.org/10.1017/S0263574702004551 Published online by Cambridge University Press

On the velocity level, we have the following 6 vector expressions, allowing us to solve for all the percussions and the values of $\dot{\theta}_{12}^{+}$, $\dot{\theta}_{23}^{+}$, $\dot{\theta}_{3}^{+}$:

$$m_1(\bar{V}_{G_1}^{+} - \bar{V}_{G_1}^{-}) = \bar{P}_F - \bar{P}_K$$
$$m_2(\bar{V}_{G_2}^{+} - \bar{V}_{G_2}^{-}) = \bar{P}_K - \bar{P}_H$$
$$m_3(\bar{V}_{G_3}^{+} - \bar{V}_{G_3}^{-}) = \bar{P}_H$$
$$\bar{\mu}_{G_1}^{+} - \bar{\mu}_{G_1}^{-} = G_1 - F \times \bar{P}_F - G_1 - K \times \bar{P}_K$$
$$\bar{\mu}_{G_2}^{+} - \bar{\mu}_{G_2}^{-} = G_2 - K \times \bar{P}_K - G_1 - H \times \bar{P}_H$$
$$\bar{\mu}_{G_3}^{+} - \bar{\mu}_{G_3}^{-} = G_3 - H \times \bar{P}_H$$

with $\bar{V}_{G_i}$ being the velocity of the center of gravity of link i, $\bar{\mu}_{G_i}$ being the angular momentum of link i with respect to the center of gravity of link i, and $\bar{P}_F$, $\bar{P}_K$ and $\bar{P}_H$ being the percussions at foot, knee and hip, respectively. It is clear that at this level the vertical component of $\bar{P}_F$ should always point upwards, which means $P_F^y > 0$.

After solving for the angular velocities, the calculation of the values of the angular accelerations is done by using the equations of motion for the stance phase. The configuration, as well as the angular velocities, are known. The torques are considered as constants, i.e. their values at touch-down; thus the accelerations are easily found.

3.4. Stance phase

Constructing the polynomial functions: During the stance phase the robot is fully actuated since an extra actuator is considered, located at the ankle joint. For the robot to be able to perform the control of the flight phase described above, there is a certain control needed during the stance phase of the preceding hop, yielding the desired initial conditions at take-off. Using the results of the impact phase and the results of the algorithm developed for controlling the flight phase, three polynomial functions are constructed which have to be tracked during stance. With respect to the desired objectives, viz. forward velocity during flight, step length and stepping height, both a hopping pattern consisting of different successive hops as well as a steady-state hopping pattern, can be realized. Steady-state means that the leg as well as the body act in the same way every hop, both in the stance phases and the flight phases.

Two polynomial functions will be constructed to steer the internal angles at hip and knee. The third function will be used to steer the absolute angle of the lower leg with respect to the ground. Since the beginning of the stance phase is the end of the impact phase, the initial conditions for the stance phase are calculated as in the above section. In the case of steady-state hopping, the same values for θ_{12}, θ_{23}, and θ_3 and their first and second derivatives, as calculated for the control of the preceding flight phase can be used for the construction of the polynomial functions of the stance phase. In fact, the ankle steer function is θ_1, but the boundary values for this function are calculated as:

$$\theta_1^{+} = \theta_3^{+} - \theta_{23}^{+} - \theta_{12}^{+}$$
$$\theta_1^{to} = \theta_3^{to} - \theta_{23}^{to} - \theta_{12}^{to}$$

The first and second derivatives are similarly derived.

In case a hopping pattern with different consecutive hops is needed, when, for example hopping on an irregular terrain, or when accelerating or decelerating, the polynomial functions can be constructed in the same way, but now using different initial and final values for each stance phase. The final values are determined by the beginning of the next flight phase, thus depending on the desired objectives of that next phase.

Compared to the desired functions for the flight phase, there is an additional degree of freedom in the construction of the desired functions for the stance phase. The end of the flight phase is determined by the flight time, which on its term is determined by expression (12), whereas the end of the stance phase, being the moment of take-off, can be freely chosen. This means that we can choose the stance time T^{st}. Suppose that during the stance phase the forward velocity of the COG has to change from $\dot{X}_G^{td} = \dot{X}_G^{to,old}$ at touch-down to $\dot{X}_G^{to,new}$ at take-off. We want the forward velocity of COG to vary as uniformly as possible during the stance phase, since decelerating and re-accelerating G in the horizontal direction would be a waste of energy. If we assume that the COG moves with a constant mean horizontal speed $\dot{X}_G^{st,mean}$ during the stance phase, then we have:

$$\dot{X}_G^{st,mean} = \frac{\Delta X_G^{st}}{\Delta t^{st}} = \frac{X_G^{to} - X_G^{td}}{T^{st}}$$

with X_G^{to} being determined by the take-off configuration of the next flight phase and X_G^{td} is known from the preceding flight phase.

If we approximate the mean forward velocity as follows:

$$\dot{X}_G^{st,mean} \approx \frac{\dot{X}_G^{to,old} + \dot{X}_G^{to,new}}{2}$$

then the following value for the stance time T^{st} is chosen:

$$T^{st} = \frac{X_G^{to} - X_G^{td}}{\frac{\dot{X}_G^{to,old} + \dot{X}_G^{to,new}}{2}} \tag{28}$$

As T^{st} is known, the polynomial functions $\theta_{12}^{st}(t)$, $\theta_{23}^{st}(t)$, $\theta_3^{st}(t)$ can be established. When the controllers for both the flight phase and the stance phase are able to track the prescribed functions, the robot is able to perform any desired hopping pattern, with every hop satisfying its desired objective locomotion parameters.

Torque limitations: In terms of geometrical and actuator constraints, the hopping pattern has, of course, to be physically realizable. In that context, special attention has to be given to the actuator at the ankle joint. Because of the limited length of the foot, the torque T_F which can be exerted at the ankle joint is limited. If this torque exceeds a certain value, the foot will start rotating,[30] or in other words the robot will start tipping over. In order to avoid this, the torque will be truncated. The following constraints have to be satisfied during the stance phase:

$$T_F > -M(\ddot{Y}_G + g)l_{F1} \tag{29}$$
$$T_F < -M(\ddot{Y}_G + g)l_{F2} \tag{30}$$

where

https://doi.org/10.1017/S0263574702004551 Published online by Cambridge University Press

$$M(\ddot{Y}_G+g)=R_y$$

Condition (29) has to be satisfied when $T_F<0$, and condition (30) in case of $T_F>0$. The length l_{F1} is the length of the foot before the ankle point, length l_{F2} is the length of the foot behind the ankle point (for the simulations reported in this paper we assumed that $l_{F1}=0.15$ m and $l_{F2}=0.05$ m). The constraints (29) and (30) result from the fact that the Zero Moment Point[30] should remain within the surface of the foot. As we can see, the magnitude of reaction force R_y has great influence on the torque which can be exerted. The larger R_y, the further the Zero Moment Point stays away from the endpoints of the foot. This is especially of great importance immediately after impact. At that stage, the torque exerted on the foot has, in general, a positive value, since the COG is still behind the foot point and the actuators have to move this COG forward. A positive torque is especially dangerous since the part of the foot behind the ankle point is smaller than the part before the ankle point. Therefore we make the foot generate an extra push at touch-down (see 3.2.3).

Another limitation which is applicable to the exerted torques at hip, knee and ankle joints is due to the limited friction between the foot and the ground. When applying Coulomb's friction law, the following constraint has to be satisfied during the stance phase in order to avoid slipping of the foot:

$$|R_x| \leq f|R_y|$$

with f being the friction coëfficient.

A last limitation is due to the performance of the different actuators. Every actuator has a limitation on its maximum power as well as on the maximum torque it can exert.

When the control algorithm generates the reference trajectories, it is possible to take all these constraints into account and calculate the most ideal trajectories depending on the criterion that is used.[22] This method can, however, not be used in real-time because it requires an extensive amount of computation. We suggest to use simulations to determine which take-off and touch-down configurations in combination with which sets of the objective locomotion parameters deliver trajectories which satisfy all the constraints, and store this information in a lookup table. The controller of a real robot could then use this information to make sure the reference trajectories are calculated so that the maximum torques are not exceeded in some area around the trajectory.

4. SIMULATIONS

In this section, the results of 2 different simulations are presented. These simulations are performed using the Multibody Code Mechanica Motion. An important remark has to be made concerning the ground model. As mentioned before, the control algorithm uses an inelastic impulsive impact phase to estimate the angular velocities after the shock. This model is used because, in reality, it is difficult to measure these velocities. Not only will the sensors on the robot be disturbed because of the shock, but it is also difficult to measure the exact duration of the impact phase. Therefore we use this model, which predicts in fact what the values of the angular velocities will be. This means that on a real robot, there will always be an error on the first point of the polynomial functions for the stance phase. This first point is, however, not crucial, since the robot is fully actuated and only the end-point determines the values of the objective parameters. In order to simulate the difference between the ground model used for the calculations and the real ground on which the real robot would have to move, the simulations are performed with another ground model. A parallel spring and damper system is used to model the real ground.

The actuators are simulated by PD-controlled torques, which have to track the polynomial steer functions calculated by the algorithm. This way, the algorithm is tested with non-ideal controllers.

4.1. Steady-state hopping pattern

To test the algorithm, first a hopping pattern of a number of identical consecutive hops has been simulated. Consequently, the desired values of the objective parameters are the same for every hop, as well as the behavior of the upper body. The chosen parameters are the following:

- $\nu=1\ m/s$, $\lambda=0.4\ m$, $\delta=0\ m$
- $\theta_{12}^{to}=42°$, $\theta_{23}^{to}=-9.5°$, $\theta_3^{to}=90°$
- $\theta_{12}^{td}=44°$, $\theta_{23}^{td}=-42.5°$, $\theta_3^{td}=80°$
- $\dot{X}_F^{td}=-1\ m/s$, $\dot{Y}_F^{td}=0\ m/s$
- $\ddot{X}_F^{td}=0\ m/s^2$, $\ddot{Y}_F^{td}=-2g\ m/s^2$
- $Y_F^{des}*=0.05\ m$, $t*=\frac{T^{fl}}{2}$

This results in the following:

- $T^{fl}=0.2\ s$
- $\dot{\theta}_3^{td}=-0.74\ rad/s$ $(n=2)$
- $\mu_G^{to}=-0.255\ kgm^2/s$
- $T^{st}=0.2\ s$

Figure 2 shows the forward velocity $\dot{X}_G$ of the global COG versus time. The horizontal parts of the graph represent the velocity during the flight phases, equal to the desired value of 1 m/s. The first hop the robot reaches exactly the desired value, because the initial conditions were set manually to start the simulation. The second hop is the first hop which results from a physical stance phase. The small deviation is due to the non-ideal controllers. The deviation stays however constant during the successive hops.

Figure 3 gives the horizontal position X_F of the foot versus time. The horizontal parts of the graph represent the position of the foot during the successive stance phases. The difference between the position during two successive stance phases equals the desired step length of 0.4 m.

The vertical position of the foot Y_F is shown in Figure 4. During the stance phases the position is equal to 0 since this is the desired stepping height. During flight the foot reaches its highest vertical position at $t=\frac{T^{fl}}{2}$ as was chosen. The maximum value differs slightly from the chosen value (about 5 mm) because of the approximations we made. The graphs shows also that the robot bounces back slightly after

https://doi.org/10.1017/S0263574702004551 Published online by Cambridge University Press

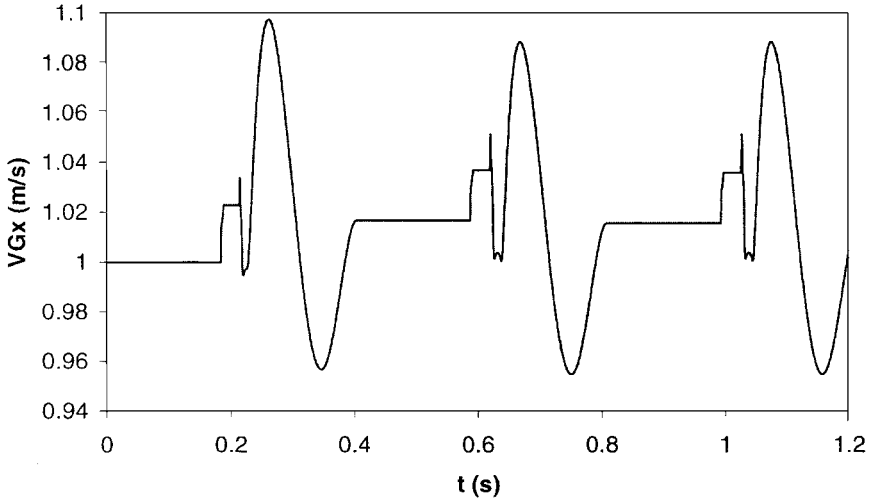

Fig. 2. Forward velocity COG.

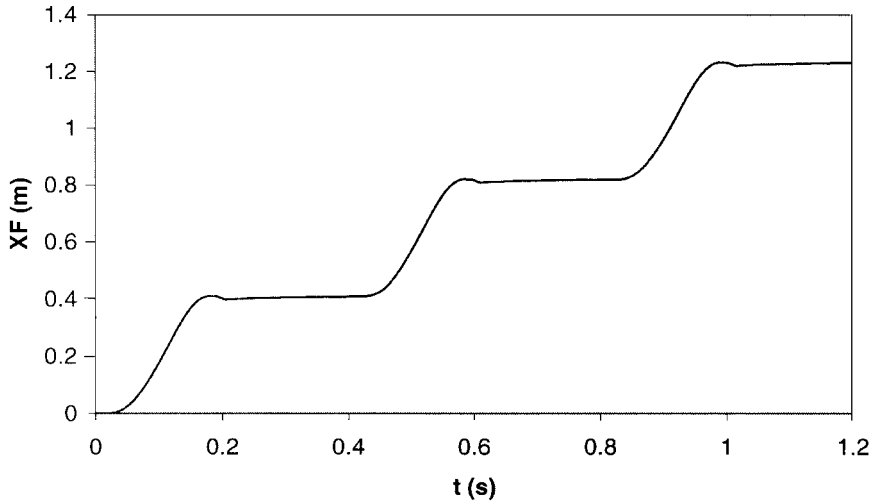

Fig. 3. Horizontal position foot.

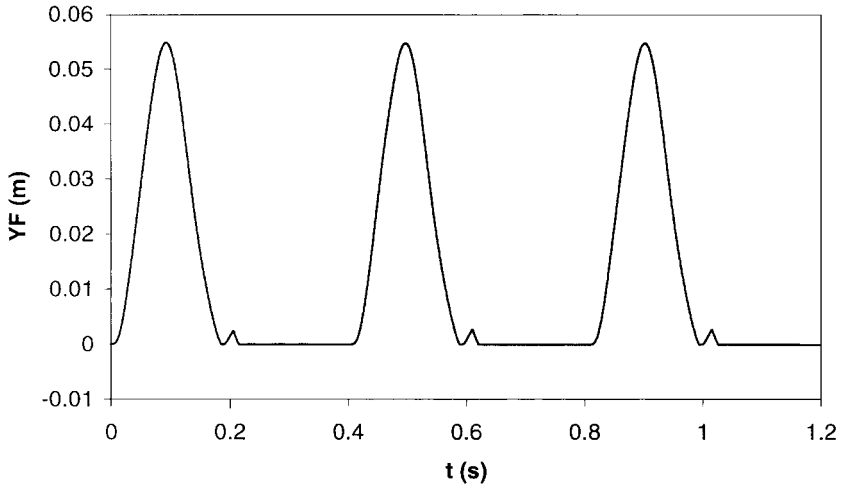

Fig. 4. Vertical position foot.

https://doi.org/10.1017/S0263574702004551 Published online by Cambridge University Press

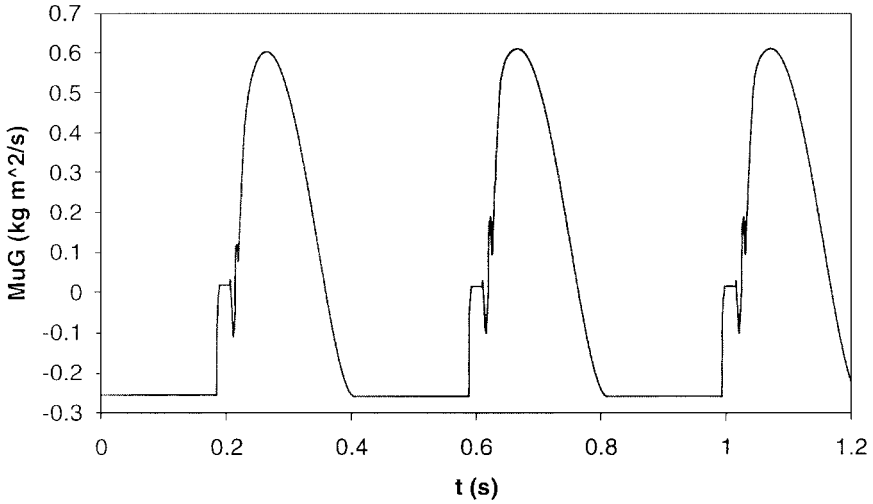

Fig. 5. Angular momentum about COG.

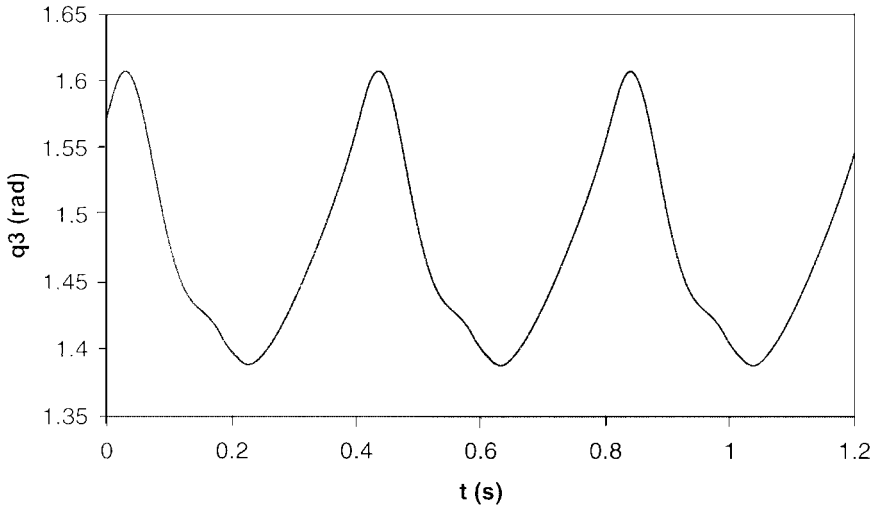

Fig. 6. Absolute angle upper body.

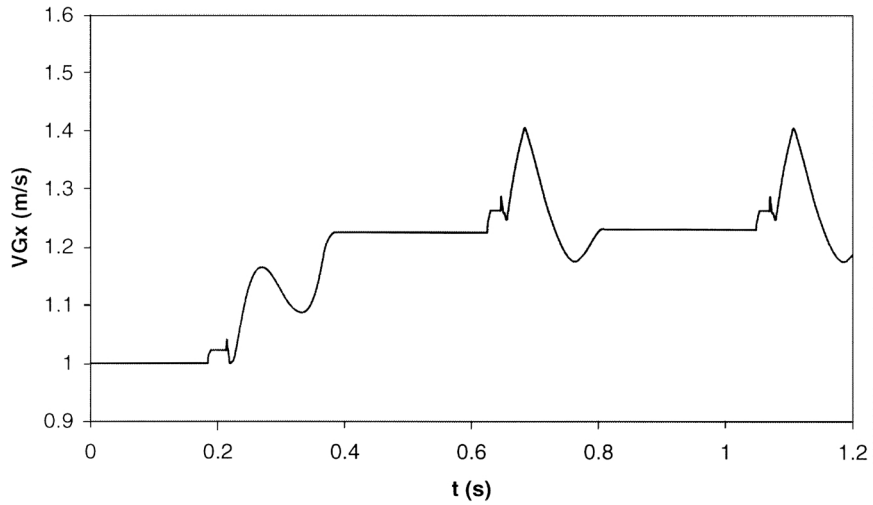

Fig. 7. Forward velocity COG.

https://doi.org/10.1017/S0263574702004551 Published online by Cambridge University Press

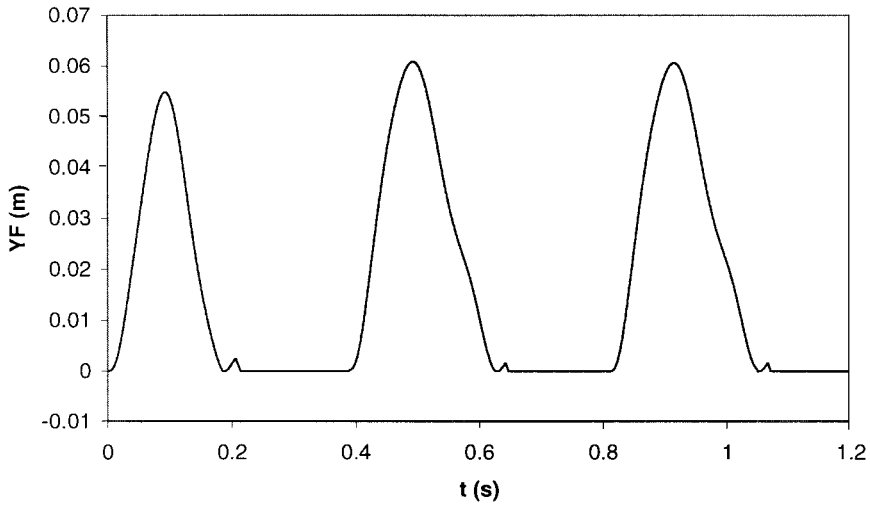

Fig. 8. Vertical position foot.

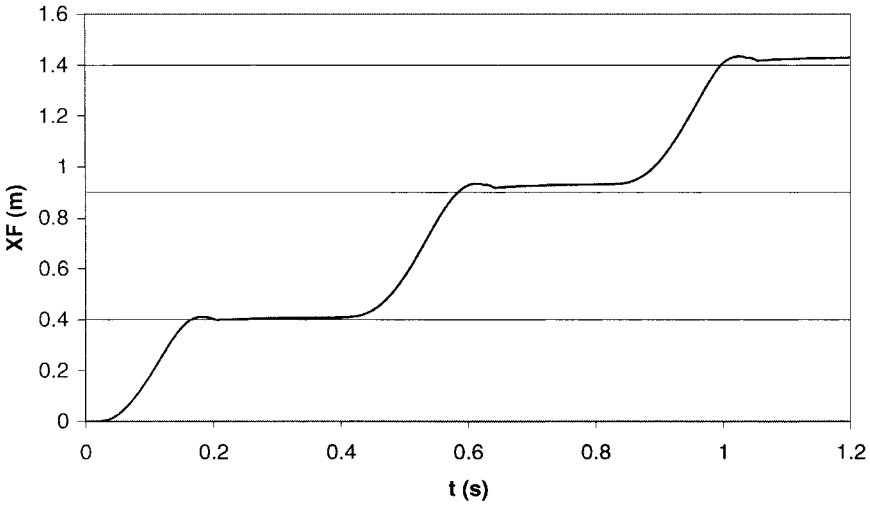

Fig. 9. Horizontal position foot.

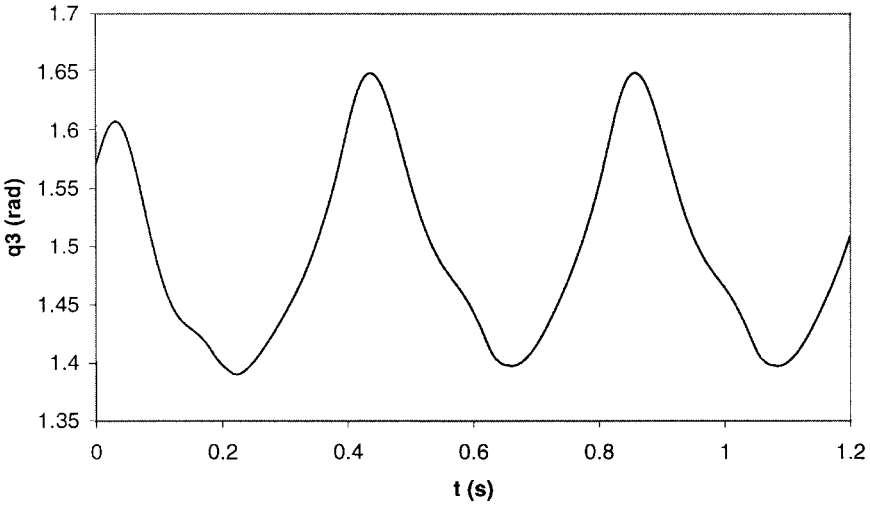

Fig. 10. Absolute angle upper body.

https://doi.org/10.1017/S0263574702004551 Published online by Cambridge University Press

impact. Contrary to what the control algorithm assumes, there is not a perfect inelastic collision. This does, however, not disturb the motion, since the robot still reaches the desired value for its objective locomotion parameters.

The angular momentum with respect to the global COG μ_G is given in Figure 5. It shows clearly that a steady state behavior is reached and that during every flight phase the angular momentum is equal to the value predicted by the control algorithm.

Finally, Figure 6 shows the absolute angle of the upper body with respect to the horizontal axis θ_3. There is no drift in the upper body motion. The rotation which occures during flight is fully compensated during the next stance phase.

4.2. Non steady-state hopping pattern

The following experiment makes the robot change its objective parameters from one hop to another, which simulates the motion on an irregular terrain. A first hop is performed with the set of objective parameters given in 4.1, and a second hop is performed by increasing both the step length and the forward velocity with 20%. This is a rather heavy transition, which is performed in one step. After that, the robot is back in steady-state with the new set of objective parameters. The transition is made without changing the take-off and touch-down configuration. Figures 7, 8, 9 and 10 show that the same conclusions can be drawn as in section 4.2.

5. CONCLUSIONS

A real-time applicable control algorithm for a one-legged hopping robot is presented. The robot is able to hop on an irregular terrain, since it is possible to change its objective locomotion parameters from one hop to another. Each hop, the velocity, as well as the step length and the stepping height, can be altered. Drift on the upper body motion is avoided, since the rotation induced during the flight phase is fully compensated during the next stance phase. During the flight phase the robot is under-actuated. It can, however, move from an initial chosen configuration to a chosen end configuration. To make this possible, the angular momentum constraint is written in a Caplygin form, and an adequate estimation of the touch-down angular velocity of the body is made. Using an iterative procedure the angular velocity is adapted until the desired end configuration is reached. Simulations show that very few iterations are needed, and that all the values of the objective parameters as well as the requested end configuration are attained. Future work will be concerned with the expansion of the algorithm for use by a running and walking biped.

References

1. R. Boulic, N. Magnenat-Thalmann and D. Thalmann, "A global human walking model with real-time kinematic personification," *The Visual Computer* **6**, 344–358 (1990).
2. A. Bruderlin and T. Calvert, "Goal-directed animation of human walking," *Computer Graphics* 23(3), 233–242 (1989).
3. M. Raibert and H. B. Brown (Jr.), "Experiments in balance with a 2d one-legged hopping machine," *Journal of Dynamic Systems, Measurement and Control* **106**, 75–81 (1984).
4. M. Raibert, H. B. Brown (Jr.) and M. Chepponis, "Experiments in balance with a 3d one-legged hopping machine," *Int. J. Robotics Research* **3**(2), 75–92 (1984).
5. M. Raibert, M. Chepponis and H. B. Brown (Jr.), "Running on four legs as though they were one," *IEEE Transactions on Robotics and Automation* **2**(2), 70–82 (1986).
6. M. Raibert, "Trotting, pacing and bounding by a quadruped robot," *Journal of Biomechanics* **23**(1), 79–98 (1990).
7. J. Hodgins, J. Koechling and M. Raibert, "Running experiments with a planar biped," *Proceedings 3rd International Symposium on Robotics Research* (Giralt and Ghallab, eds.), MIT Press (1986) pp. 349–355.
8. J. Hodgins and M. Raibert, "Biped gymnastics," *Int. J. Robotics Research (Special Issue on Legged Locomotion)* **9**(2), 115–132 (1990).
9. R. Playter and M. Raibert, "Control of a biped somersault in 3d," *Proceedings 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems* Raleigh, NC, USA (1992) pp. 582–589.
10. J. Helferty, J. Collins and M. Kam, "A neural network learning strategy for the control of a one-legged hopping machine," *Proceedings IEEE International Conference on Robotics and Automation*, Scottsdale, USA (1989) pp. 1604–1609.
11. Z. Li and J. He, "An energy perturbation approach to limit cycle analysis in legged locomotion systems," *Proceedings 29th Conference on Decision and Control*, Honolulu, Hawaii (1990) pp. 1989–1994.
12. Z. Li and R. Montgomery, "Dynamics and optimal control of a legged robot in flight phase," *Proceedings IEEE International Conference on Robotics and Automation*, Cincinnati, USA (1990) pp. 1816–1820.
13. D. Koditschek and M. Bühler, "Analysis of a simplified hopping robot," *Int. J. Robotics Research* **10**(6), 587–605 (1991).
14. R. M'Closkey and J. Burdick, "An analytical study of simple hopping robots with vertical and forward motion," *Proceedings 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, USA (1991) pp. 1392–1397.
15. A. Vakakis, J. Burdick and T. Caughey, "An 'interesting' strange attractor in the dynamics of a hopping robot," *Int. J. Robotics Research* **10**(6), 606–618 (1991).
16. R. M'Closkey and J. Burdick, "Periodic motions of a hopping robot with vertical and forward motion," *Int. J. Robotics Research* **12**(3), 197–218 (1993).
17. J. Ostrowski and J. Burdick, "Designing feedback algorithms for controlling the periodic motions of legged robots," *Proceedings 1993 IEEE International Conference on Robotics and Automation* (1993) pp. 260–266.
18. P. Gregorio, M. Ahmadi and M. Buehler, "Experiments with an electrically actuated planar hopping robot," *Experimental Robotics* **III**, 269–281 (1994).
19. P. Gregorio, M. Ahmadi and M. Buehler, "Design, control, and energetics of an electrically actuated legged robot," *IEEE Transactions on Systems, Man and Cybernetics* **27B**(4), 626–634 (1997).
20. H. D. Man, D. Lefeber, F. Daerden and E. Faignet, "Simulation of a new control algorithm for a one-legged hopping robot (using the multibody code mechanica motion)," *Proceedings International Workshop on Advanced Robotics and Intelligent Machines*, Manchester, UK (1996) pp. 1–13 (paper No. 8).
21. H. D. Man, D. Lefeber and J. Vermeulen, "Design and control of a one-legged robot hopping on irregular terrain," *Proceedings Euromech 375: Biology and Technology of Walking*, Munich, Germany (1998) pp. 173–180.
22. C. Chevallereau and Y. Aoustin, "Optimal running trajectories for a biped," *Proceedings 2nd International Conference on*

https://doi.org/10.1017/S0263574702004551 Published online by Cambridge University Press

Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR '99) (G. Virk, M. Randall and D. Howard, eds.), Professional Engineering Publishing, Portsmouth, UK (1999) pp. 559–570.
23. K. Löffler, M. Gienger and F. Pfeiffer, "Simulation and control of a biped jogging robot," *Proceedings on the 4th International Conference on Climbing and Walking Robots: From Biology to Industrial Applications (CLAWAR 2001)* (K. Berns and R. Dillmann, eds.), Professional Engineering Publishing (2001), pp. 867–874.
24. P. Janssens, *Mécanique Rationelle*, **Vol. 2**, chapter 13 (Les Chocs Dans les Systèmes de Solides), 2nd edition (1983) pp. 298–321.
25. C. François and C. Samson, "A new approach to the control of the planar one-legged hopper," *Int. J. Robotics Research* **17**(11), 1150–1166 (1998).
26. A. De Luca and G. Oriolo, *Modeling and Control of Nonholonomic Mechanical Systems*, Springer Verlag, Wien (1995) chapter 7, pp. 277–342.
27. A. Bloch, M. Reyhanoglu and N. McClamroch, "Control and stabilization of nonholonomic dynamic systems," *IEEE Transactions on Automatic Control* **37**(11), 1746–1757 (1992).
28. J. M. Godhavn, A. Balluchi, L. Crawford and S. S. Sastry, "Control of nonholonomic systems with drift terms," *Tech. Rep. UCB/ERL M97/1*, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA-94720, California, USA (1997).
29. J.-M. Godhavn, A. Balluchi, L. S. Crawford and S. S. Sastry, "Path planning for nonholo-nomic systems with drift," *Proc. 1997 IEEE American Control Conference*, Albuquerque, New Mexico, USA (1997) **Vol. 1**, pp. 532–536,.
30. A. Goswami, "Postural stability of biped robots and the foot-rotation indicator (FRI) point," *Int. J. Robotics Research* **18**(6), 523–533 (1999).
31. A. Daberkow, J. Gao and W. Schiehlen, "Walking without impacts," *Proceedings 8th Symposium on Theory and Practice of Robots and Manipulators*, Cracow, Poland (1990) pp. 339–347.

https://doi.org/10.1017/S0263574702004551 Published online by Cambridge University Press