Robotica (2011) volume 29, pp. 627–640. © Cambridge University Press 2010
doi:10.1017/S0263574710000378

Generating efficient rigid biped running gaits with calculated take-off velocities

Q. Guo, C. J. B. Macnab* and J. K. Pieper

Schulich School of Engineering, University of Calgary, 2500 University Dr. NW, Calgary, AB, Canada T2N 1N4

(Received in Final Form: July 16, 2010. First published online: August 23, 2010)

SUMMARY

In this paper, an offline algorithm generates gaits for robotic biped running motion. A minimization procedure on a novel objective function produces initial joint velocities for the flight phase given the initial joint angles. These computed initial velocities ensure a reasonable starting point for the constrained dynamic optimization procedure to try and formulate a gait that minimizes energy consumption. The simulated robot runs even on ground and up stairs. Results demonstrate the rigid robot solution has a intuitively satisfying human-like motion with an energy consumption that appears reasonable for a rigid robot.

KEYWORDS: Running robots; Biped running; Biped robots; Multibody dynamics; Dynamic optimization.

1. Introduction

This paper presents an offline generator of running gaits for rigid bipeds. Our work takes steps towards solving problems in the biped literature, including running on varied terrain and achieving energy efficiency. In order to find solutions for the initial velocities in the flight phase, at take-off, we minimize the difference in rotational kinetic energy between take-off and touch-down as well as the absolute rotational kinetic energies. This approach differs from that of ref. [1] which requires the initial velocities to be given. Looking for initial joint velocities through trial-and-error is a time consuming process that has no guarantee of success. Both approaches assume known boundary joint angles, but these are relatively easy to find by looking at examples of human gaits. Ultimately, after a dynamic optimization procedure, the efficient generated gaits serve to validate our approach. Moreover, the same strategy generates running gaits where the even ground condition does no longer holds, e.g. running up stairs.

Section 2 reviews the relevant literature in walking, hopping and running robots. Section 3 describes the modelling of the biped robot and generating the gaits (including the two running phases, the foot/ground collision and all relevant constraints). The main original contribution of this paper appears in Section 3.2, where we propose two novel objective functions that can be used to find appropriate boundary velocities. Section 4 illuminates some key points in implementing the dynamic optimization in software. Simulated running for two cases, on even ground and up stairs, appears in Section 5. Conclusions on practicality follow, including required torques and energy consumption.

* Corresponding author. E-mail: cmacnab@ucalgary.ca

2. Background

A common aim of many robotics researchers is to achieve human-like motions with a biped robot for the purposes of achieving energy efficiency and/or creating human-like android robots. Our work does not consider unusual gaits like hopping, skipping, sneaking, limping, statically-stable walking or 'Groucho running'. Humans spend most of their time either running or walking in the standard sense[2] to reduce energy consumption, although standard gaits do change to deal with environmental factors like uneven terrain, obstacles and stairs. We assume that the stance phase is symmetric.[3]

To formally distinguish our definitions of running and walking, consider the following comparisons:

(1) Running contains a flight phase, where both feet are off the ground and the Centre of Mass (CoM) executes a ballistic motion. In walking at least one foot touches the ground at all times, and includes a double support phase where both feet stand on the ground simultaneously.
(2) Running can achieve a faster speed and a larger range of speed. The maximum speed of walking is theoretically bounded by $\sqrt{gl_0}$, with g being the gravitational acceleration constant and l_0 being the leg length.[4] Running speed is only constrained by practical constraints including actuator saturation, slip and ground reaction forces.
(3) In human running, the vertical ground reaction force (GRF) appears to be bell-shaped with the opening downwards. In contrast, the vertical GRF of walking is typically M-shaped, with two peaks and a valley in between.[5]
(4) In running, the CoM reaches the lowest position at the mid-stance when the hip of the stance leg passes over the ankle, whereas the CoM climbs to the highest position at the mid-stance in walking.[6]

The robotics literature contains many biped walking solutions, including experimentally validated energy-efficient gaits.[7] For running, typical designs incorporate compliant elements which can produce large instantaneous torques/forces at take-off.[8–12] Running with rigid bipeds presents difficulties, mostly due to a tendency to saturate

https://doi.org/10.1017/S0263574710000378 Published online by Cambridge University Press

the actuators at take-off. Thus, it is important to make the gait as energy efficient as possible.

While many researchers concentrate their attention on use of compliant elements, some investigators are trying to implement running function on the rigid bipeds that were originally designed for walking. Kajita *et al.* proposed a 'resolved momentum control' algorithm to generate running gaits for the bipedal Humanoid Robotics Platform (HRP-2L) using a simple inverted pendulum model.[13,14] The famous HONDA Asimo also announced running videos on its official website, but the technical details were not disclosed yet. Fujimoto applied variational calculus to generate low-energy running gaits.[1] Morris *et al.* reported an achievement of bipedal running with the RABBIT, based on hybrid system control theory.[15] Offline optimization computes the desired joint states (angles and angular velocities), but the initial joint states have to be picked manually, as in Fujimoto's method.

Computing rather than choosing joint velocities is the main contribution of our paper. The improvement in the proposed method comes from an easier search for initial joint velocities and reduced energy consumption. If one chooses joint velocities which are unreasonable then the legs can oscillate during flight, wasting energy. On the other hand, searching for reasonable velocities using intuition or trial and error is very difficult. In our own efforts, we were not able to produce satisfying gaits by guessing the initial velocities. Moreover, we do not expect treating initial angles and velocities as uncoupled quantities would lead to a minimum energy solution. Evidence for this point of view stems from observing human motion, where a particular human always uses the same take-off angles and velocities for a given average forward speed. Besides the practical improvement of our method, our energy analysis may offer insight into biped locomotion.

3. Modelling and Gait Generation

The planar biped is a multibody dynamic chain consisting of seven links and six actuated frictionless pin joints. The rigid links include an upper body, two thighs, two calves and two feet. Movement comes from two joints at the hip, two knee joints and two ankle joints. By assumption, the mass, inertia and height of the two foot links are insignificant. The mathematical model consists of five links and five link angles (Fig. 1). The five links have lengths l_i and masses M_i for $i = 1, \ldots, 5$. The total mass is $M_{\text{tot}} = \sum_{i=1}^{5} M_i$. Each moment of inertia I_i is taken about the CoM of the i^{th} link. Each r_i is a ratio of the CoM location of the massive link to the link lengths. The joint position vector includes the five angles $\Theta = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)^{\text{T}}$, which, by assumption, are measured.

Note that in our coordinate system there are two hip joints, one for each leg. Therefore there are six joints in total, but only five (significant) links. Ensuring a safe landing, the actuators in the ankle joints keep the feet parallel to the ground in flight. By assumption, in the stance phase the support foot grips the ground firmly, without bounce, slipping, or rotation.

A complete running cycle consists of two half-cycles, where each half-cycle contains a flight phase and a stance phase. The support leg leaving the ground, the *take-off* event, indicates the beginning of the flight phase. The foot on the other leg eventually collides with the ground, a *touch-down* event. The front leg touch-down indicates the end of the flight phase and the start of the stance phase. The flight phase take-off ends the stance phase.

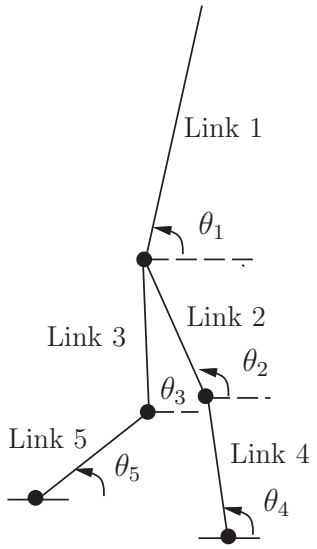


Fig. 1. The model of the planar biped.

The position of the CoM

$$P_{\text{c}} = \begin{bmatrix} X_{\text{c}} \\ Y_{\text{c}} \end{bmatrix} \tag{1}$$

follows from a Cartesian position of the first ankle $P_{\text{a1}} = [X_{\text{a1}} \quad Y_{\text{a1}}]^{\text{T}}$ and a function of joint angles $f_1(\Theta)$ using

$$P_{\text{c}} = P_{\text{a1}} + f_1(\Theta), \tag{2}$$

Alternatively, the second ankle can provide the base

$$P_{\text{c}} = P_{\text{a2}} + f_2(\Theta). \tag{3}$$

Differentiating Eqs. (2) and (3), respectively, once gives

$$\dot{P}_{\text{c}} = \dot{P}_{\text{ai}} + \frac{\partial f_i}{\partial \Theta}\dot{\Theta} \qquad \text{for} \quad i = 1, 2 \tag{4}$$

and twice gives

$$\ddot{P}_{\text{c}} = \ddot{P}_{\text{ai}} + \frac{\text{d}}{\text{d}t}\left(\frac{\partial f_i}{\partial \Theta}\right)\dot{\Theta} + \frac{\partial f_i}{\partial \Theta}\ddot{\Theta}. \tag{5}$$

In the stance phase, gravity and GRFs act on the robot. In contrast, during the flight phase gravity is the only external force. When completing the flight phase, the robot foot collides with the ground with a certain velocity. This collision causes an abrupt velocity jump at each joint. The discussion starts with the flight phase, followed by a solution to the velocity jump due to the collision between the support foot and the ground, and finally goes on to explain the support phase. Symbol definitions appear in Tables I and II.

https://doi.org/10.1017/S0263574710000378 Published online by Cambridge University Press

Table I. Selected symbols.

P	$(X, Y)^T$, Cartesian position	m
V	$(\dot{X}, \dot{Y})^T$, Cartesian velocity	m/s
Θ	Absolute link angles	rad
D	Inertia matrix	–
H	Centrifugal/Coriolis terms	–
G	Gravity vector	–
τ	Joint torque	Nm
B	Coefficient matrix of τ	–
g	Gravity acceleration	m/s^2
L_s	Stride length	m
h_s	Stride height	m
Γ	Angular momentum	kg m^2/s
I	Inertia about link CoM	kg m^2
J	Jacobian matrix	–

Table II. Subscripts and superscripts.

Subscripts	
a1	Right ankle
a2	Left ankle
to	Take-off
td	Touch-down
c	Centre of mass (CoM)
tot	Total
x	Horizontal direction
y	Vertical direction
rot	Rotational
KE	Kinetic energy
PE	Potential energy
Superscripts	
fl	Flight phase
st	Stance phase

3.1. *The flight phase*

3.1.1. Equations of motion. During flight the Cartesian and rotational dynamics are decoupled. The equations of motion in Cartesian space are

$$D_1^{\mathrm{fl}} \ddot{P}_c + G_1^{\mathrm{fl}} = 0, \tag{6}$$

where 'fl' stand for *flight* and

$$D_1^{\mathrm{fl}} = M_{\mathrm{tot}} \mathrm{I}_{2x2}, \qquad G_1^{\mathrm{fl}} = [0 \quad M_{\mathrm{tot}} g]^T. \tag{7}$$

Equation (6) simplifies to

$$\ddot{X}_{\mathrm{c}} = 0, \qquad \ddot{Y}_{\mathrm{c}} = -g, \tag{8}$$

implying a parabolic trajectory.

Formulating the rotational equations of motion using Euler–Lagrange method gives

$$D_2^{\mathrm{fl}}(\Theta)\ddot{\Theta} + H_2^{\mathrm{fl}}(\Theta, \dot{\Theta}) = B_2^{\mathrm{fl}} \tau, \tag{9}$$

where

- $D_2^{\mathrm{fl}} \in \Re^{5\times5}$ is the inertia matrix,
- $H_2^{\mathrm{fl}} \in \Re^{5\times5}$ has centrifugal and Coriolis forces,
- $\tau^{\mathrm{fl}} = (\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6)^{\mathrm{T}}$ contains the torques of the hips, the knees and the ankles,
- the term $B_2^{\mathrm{fl}} \in \Re^{5\times6}$ is the constant coefficient matrix of τ

Choosing generalized coordinates in flight as

$$q^{\mathrm{fl}} = \begin{bmatrix} q_1^{\mathrm{fl}} \\ q_2^{\mathrm{fl}} \end{bmatrix} = \begin{bmatrix} P_{\mathrm{c}} \\ \Theta \end{bmatrix} \tag{10}$$

leads to the compact expression for the dynamics

$$D^{\mathrm{fl}}(q)\ddot{q}^{\mathrm{fl}} + H^{\mathrm{fl}}(q^{\mathrm{fl}}, \dot{q}^{\mathrm{fl}}) + G^{\mathrm{fl}} = B^{\mathrm{fl}} \tau, \tag{11}$$

where $D = \mathrm{diag}(\mathrm{D}_1, \mathrm{D}_2)$,$H = \mathrm{diag}(0, \mathrm{H}_2)$, $G = [G_1^T, 0^{\mathrm{T}}]^{\mathrm{T}}$ and $B = [0^{\mathrm{T}}, \mathrm{B}_2^{\mathrm{T}}]^{\mathrm{T}}$ (we drop the superscript 'fl' here and for the rest of this subsection to improve readability).

3.1.2. The stride. Defining the right ankle take-off position as $P_{\mathrm{a1,to}} = (0, 0)^T$ means the left ankle touch-down position will be $P_{\mathrm{a2,td}} = (L_{\mathrm{s}}, h_{\mathrm{s}})^{\mathrm{T}}$, where L_{s} denotes stride length and h_{s} stride height. In our algorithms Θ_{to} and Θ_{td} are picked in advance and then positions $P_{\mathrm{c,to}}$ and $P_{\mathrm{c,td}}$ follow from Eqs. (2) and (3), respectively.

We command the robot to run at a constant average forward speed V_{x} i.e. we want

$$\dot{X}_{\mathrm{c,to}} = \dot{X}_{\mathrm{c,td}} = V_{\mathrm{x}}. \tag{12}$$

The flight lasts for the duration

$$T^{\mathrm{fl}} = (X_{\mathrm{c,td}} - X_{\mathrm{c,to}})/V_{\mathrm{x}}. \tag{13}$$

The initial vertical velocity of the flight phase is

$$\dot{Y}_{\mathrm{c,to}} = \left(Y_{\mathrm{c,td}} - Y_{\mathrm{c,to}} + \frac{1}{2} g (T^{\mathrm{fl}})^2 \right) / T^{\mathrm{fl}}, \tag{14}$$

and the final velocity is

$$\dot{Y}_{\mathrm{c,td}} = \dot{Y}_{\mathrm{c,to}} - g T^{\mathrm{fl}}. \tag{15}$$

3.1.3. Conservation of angular momentum. Since gravity is the only force on the CoM during flight, the angular momentum of the robot about the CoM, Γ_{c} cannot change:

$$\Gamma_{\mathrm{c,to}}(\Theta_{\mathrm{to}}, \dot{\Theta}_{\mathrm{to}}) = \Gamma_{\mathrm{c,td}}(\Theta_{\mathrm{td}}, \dot{\Theta}_{\mathrm{td}}), \tag{16}$$

where

$$\Gamma_{\mathrm{c}} = \sum_{i=1}^{5} s_i \dot{\theta}_i, \tag{17}$$

with $s_i = \sum_{j=3}^{7} d_{i+2,j}$ for $i = 1, \ldots, 5$, with each $d_{i,j}$ being an element of $D(q)$ indexed by i, j.

https://doi.org/10.1017/S0263574710000378 Published online by Cambridge University Press

3.1.4. Energy analysis. The total energy is a sum of kinetic Φ_{KE} and potential Φ_{PE} energies

$$\begin{aligned}\Phi_{\mathrm{tot}} &= \Phi_{\mathrm{KE}} + \Phi_{\mathrm{PE}},\\ &= \frac{1}{2}\dot{q}_1^{\mathrm{T}} D_1 \dot{q}_1 + \frac{1}{2}\dot{q}_2^{\mathrm{T}} D_2(q_2)\dot{q}_2 + \Phi_{\mathrm{PE}}, \qquad (18)\\ &= \frac{1}{2}\dot{q}_1^{\mathrm{T}} D_1 \dot{q}_1 + \Phi_{\mathrm{rot}} + \Phi_{\mathrm{PE}}, \qquad (19)\end{aligned}$$

where Φ_{rot} is the rotational kinetic energy of the links. Expanding the Cartesian terms gives

$$\Phi_{\mathrm{tot}} = \frac{1}{2}M_{\mathrm{tot}}\dot{X}_{\mathrm{c}}^2 + \frac{1}{2}M_{\mathrm{tot}}\dot{Y}_{\mathrm{c}}^2 + \Phi_{\mathrm{rot}} + \Phi_{\mathrm{PE}}. \qquad (20)$$

Differentiating energies produces

$$\dot{\Phi}_{\mathrm{PE}} = G^T \dot{q} = M_{\mathrm{tot}} g \dot{Y}_{\mathrm{c}} \qquad (21)$$

and

$$\begin{aligned}\dot{\Phi}_{\mathrm{rot}} &= \dot{q}_2^{\mathrm{T}} D_2(q_2)\ddot{q}_2 + \frac{1}{2}\dot{q}_2^{\mathrm{T}} \dot{D}_2(q_2)\dot{q}_2,\\ &= \dot{q}_2^{\mathrm{T}}\left[B_2\tau - H_2(q_2,\dot{q}_2)\dot{q}_2 + \frac{1}{2}\dot{D}_2(q_2)\dot{q}_2\right]. \qquad (22)\end{aligned}$$

Choosing a skew-symmetric form for $\dot{D}_2 - 2H_2$ makes $\dot{q}_2^{\mathrm{T}}(\frac{1}{2}\dot{D}_2 - H_2)\dot{q}_2 = 0$. Thus

$$\dot{\Phi}_{\mathrm{rot}} = \dot{q}_2^{\mathrm{T}} B_2 \tau. \qquad (23)$$

Differentiating Eq. (20), using Eqs. (8), (21) and (23) yields

$$\dot{\Phi}_{\mathrm{tot}} = \dot{q}_2^{\mathrm{T}} B_2 \tau = \dot{\Phi}_{\mathrm{rot}}. \qquad (24)$$

Equation (24) means that the sum of the translational kinetic energy and the potential energy in (20) is a constant, since the rate of change of this sum is 0. When applying control effort, τ is not zero in general and thus $\frac{\mathrm{d}}{\mathrm{d}t}(\Phi_{\mathrm{rot}}) \neq 0$. Therefore, the rotational kinetic energy is not conserved.

3.2. Choosing boundary states

This section includes the main contributions of this work: creating novel objective functions for producing initial joint velocities using static optimization.

3.2.1. Minimizing rotational energy difference. Integrating both sides of (23) over the whole flight phase gives us the rotational energy difference

$$\Delta\Phi_{\mathrm{rot}} = \Phi_{\mathrm{rot,td}} - \Phi_{\mathrm{rot,to}} = \int_{t_{\mathrm{to}}}^{t_{\mathrm{td}}} \dot{q}_2^{\mathrm{T}} B_2 \tau \,\mathrm{d}t, \qquad (25)$$

The total amount of work done by the actuators during flight is

$$W^{\mathrm{fl}} = \int_{t_{\mathrm{to}}}^{t_{\mathrm{td}}} \sum_{j=1}^{5} |\dot{q}_{2,j}||(B_2\tau)_j| \mathrm{d}t. \qquad (26)$$

and the work must be greater

$$|\Delta\Phi_{\mathrm{rot}}| \leq W^{\mathrm{fl}}. \qquad (27)$$

Ultimately, we would like the dynamic optimization to minimize the amount of work done during the flight phase. Recent results from the biological sciences directly inspire our approach. Measured human torque curves at the ankle, the knee and the hip show relatively small torques during the flight phase compared to the stance phase.[16] Synthesized running gaits also display this characteristic.[1,13] Thus, it seems reasonable to assume that minimizing flight-energy will result in an energy-efficient gait. At this point, however, we are just trying to generate boundary velocities. We argue that boundary velocities that minimize $\Delta\Phi_{\mathrm{rot}}$ give us the best chance of minimizing W^{fl} in the subsequent dynamic optimization procedure. The results of energy consumption in our simulations in Section 5 will serve to validate our assumptions, for a particular biped configuration.

3.2.2. Minimizing absolute rotational energy. Minimizing only $\Delta\Phi_{\mathrm{rot}}$ would be too short-sighted. Taking into account the entire running cycle, as much applied control energy as possible should be turned into translational kinetic energy (as little as possible into rotational kinetic energy). Thus, it makes sense to also penalize $\Phi_{\mathrm{rot,to}}$ or $\Phi_{\mathrm{rot,td}}$ in an optimization procedure. Under some circumstances minimizing these absolute rotational energies may also minimize $\Delta\Phi_{\mathrm{rot}}$ but, in general, the constraints will force trade-offs. This is especially true for unusual gaits like climbing stairs or jumping over obstacles.

3.2.3. Novel objective functions. One possible static optimization procedure could take objective function

$$\mathcal{O}_1 = \gamma\Phi_{\mathrm{rot,to}} + (1-\gamma)(\Delta\Phi)_{\mathrm{rot}}^2, \qquad (28)$$

where the constant $\gamma \in [0,1]$ is a weighting factor. This would then seek a trade-off between minimizing rotational energy at take-off and the difference between rotational energies.

However, this objective may be unnecessarily complicated. As long as the objective function includes both $\Phi_{\mathrm{rot,to}}$ and $\Phi_{\mathrm{rot,td}}$ in the minimization (and successfully force them to be small) then $\Delta\Phi_{\mathrm{rot}}$ will also be small. Thus, $\Delta\Phi_{\mathrm{rot}}$ does not necessarily have to appear explicitly. This leads us to propose a more concise objective function

$$\mathcal{O}_2 = \gamma\Phi_{\mathrm{rot,to}} + (1-\gamma)\Phi_{\mathrm{rot,td}}. \qquad (29)$$

Minimization of $\mathcal{O}_2$ forces $\Phi_{\mathrm{rot,to}}$ and $\Phi_{\mathrm{rot,td}}$ to be small (implying small $\Delta\Phi_{\mathrm{rot}}$ as well) and still allows trade-offs in the gait generation.

We find that in practice (29) outperforms (28), resulting in more efficient gaits and allowing a larger range for γ.[17] In the sequel, only (29) is used.

3.2.4. Boundary joint velocities. The boundary joint angles are picked manually, but the boundary joint velocities have to be determined. Specifically, we need to solve for $\dot{\Theta}_{\mathrm{to}}$ and $\dot{\Theta}_{\mathrm{td}}$ such that the objective function (29) is minimized, subject to constraints. The constraints come in two forms: equality and

https://doi.org/10.1017/S0263574710000378 Published online by Cambridge University Press

inequality. The equality constraints limit the boundary states of a motion phase, and our choice is meant to be simple:

$$\dot{P}_{\mathrm{c,to}} = [V_{\mathrm{x}}, \dot{Y}_{\mathrm{c,to}}]^T,$$
$$\dot{P}_{\mathrm{c,td}} = [V_{\mathrm{x}}, \dot{Y}_{\mathrm{c,td}}]^T,$$
$$\Gamma_{\mathrm{c,to}} = \Gamma_{\mathrm{c,td}}.$$

Inequality constraints limit the motion. We bound the angular velocity of the joints by

$$\omega_{i,\min} \leq \omega_i \leq \omega_{i,\max}, \tag{30}$$

where ω_i for $i = 1, \ldots, 5$ is the angular velocity at the i^{th} joint, $\omega_{i,\min}$ and $\omega_{i,\max}$ are the lower and upper bounds of the corresponding joint velocity.

A static optimization procedure, with constraints, then finds $\dot{\Theta}_{\mathrm{to}}$ and $\dot{\Theta}_{\mathrm{td}}$. We then use $\dot{\Theta}_{\mathrm{to}}$ as the initial value of flight phase but discard the result for $\dot{\Theta}_{\mathrm{td}}$. It will be the dynamic optimization procedure that will produce the real value of $\dot{\Theta}_{\mathrm{td}}$.

3.2.5. Foot velocity regulation. Energy loss at touch-down is proportional to the foot velocity immediately before the collision.[11] Although we could demand a foot velocity of 0 at touch-down, this would require a lot of extra control effort. Thus we choose

$$\dot{P}_{\mathrm{a2,td}} = \begin{bmatrix} k_{\mathrm{x}} & 0 \\ 0 & k_{\mathrm{y}} \end{bmatrix} \dot{P}_{\mathrm{c,td}}, \tag{31}$$

where we use constants $k_{\mathrm{x}} \in [0, 1]$ and $k_{\mathrm{y}} \in [0, 1]$ to trade off collision energy loss and control effort. Choosing k_{x} smaller increases acceleration and *vice versa*. Although large k_{x} and k_{y} may result in significant energy loss at touch-down, it may be worth the decreased control effort overall.

3.3. Collision between the foot and the ground

By assumption, the foot collides with the ground instantaneously and inelastically at touch-down. The joint angles change continuously but the joint velocities jump discontinuously. Using superscripts '$-$' and '$+$' for states immediately before and after collision, respectively, the collision model in ref. [1] predicts discrete velocity changes

$$\dot{q}^+ = \left[\mathrm{I}_{7\times 7} - [D^{\mathrm{fl}}(q^-)]^{-1} J^{\mathrm{T}}(q^-) \cdot \left(J(q^-)[D^{\mathrm{fl}}(q^-)]^{-1} J^{\mathrm{T}}(q^-)\right)^{-1} J(q^-)\right]\dot{q}^-, \tag{32}$$

where J is the Jacobian matrix defined from differentiating (3)

$$P_{\mathrm{a2}} = P_{\mathrm{c}} - f_2(\Theta) \tag{33}$$
$$\dot{P}_{\mathrm{a2}} = \frac{\partial}{\partial q^{\mathrm{fl}}}(P_{\mathrm{c}} - f_2(\Theta))\dot{q}^{\mathrm{fl}} \tag{34}$$
$$= J\dot{q}^{\mathrm{fl}} \tag{35}$$

The velocity of the CoM immediately after the collision, $\dot{P}^+_{\mathrm{c,td}}$, follows directly from (4).

3.4. The stance phase

3.4.1. The stance dynamics. Assuming the foot remains firmly in contact with the ground during the stance phase, the robot moves with five DOFs. Defining the stance-phase generalized coordinates as $q^{\mathrm{st}} = \Theta$, results in dynamic model

$$D^{\mathrm{st}}(q^{\mathrm{st}})\ddot{q}^{\mathrm{st}} + H^{\mathrm{st}}(q^{\mathrm{st}}, \dot{q}^{\mathrm{st}})\dot{q}^{\mathrm{st}} + G^{\mathrm{st}}(q^{\mathrm{st}}) = B^{\mathrm{st}}\tau^{\mathrm{st}}, \tag{36}$$

where $D^{\mathrm{st}} \in \Re^{5\times 5}$ is the inertia matrix, $H^{\mathrm{st}} \in \Re^{5\times 5}$ contains centrifugal and Coriolis terms and $G^{\mathrm{st}} \in \Re^5$ is the gravitational torque vector. The matrix $B^{\mathrm{st}} \in \Re^{5\times 6}$ is the coefficient matrix and depends on the choice of the generalized coordinates (actuation occurs with six motors but there are five absolute link angles in our coordinate system).

3.4.2. Constraints. In the stance phase, the time duration, boundary joint velocities, GRFs, and the zero-moment-point (ZMP) have to be determined or bounded. The GRFs include the normal support force of the ground, and the tangential friction force between the support foot and the ground.

Assuming the robot moves at commanded velocity V_{x}, the stance phase lasts for time

$$T^{\mathrm{st}} = (X_{\mathrm{c,to}} - X_{\mathrm{c,td}})/V_{\mathrm{x}}. \tag{37}$$

Since the boundary states are known in advance, the stance phase is a typical two point boundary value problem (TPBVP). Recall the boundary angles are chosen manually and the velocities at take-off follow from the static optimization of Section 3.2. Thus, in the stance phase, the initial velocity follows from (32) and the final velocity is the velocity at take-off.

The support force from the ground is

$$F_{\mathrm{y}} = M_{\mathrm{tot}}\left(g + \ddot{Y}^{\mathrm{st}}_{\mathrm{c}}\right) \tag{38}$$

and must point upward, implying

$$\ddot{Y}^{\mathrm{st}}_{\mathrm{c}} \geq -g. \tag{39}$$

Assuming the touch-down collision is instantaneous implies the resultant normal touch-down GRF may be extremely large. In practice robots use specially designed feet to prevent damage (see ref. [18, 19]). We can also mitigate this problem by requiring a limit on this force in the gait-generation

$$F_{\mathrm{y}} \leq F_{\mathrm{y,max}}. \tag{40}$$

At take-off the normal GRF vanishes, implying $F^{\mathrm{st}}_{\mathrm{y,to}} = 0$ and

$$\ddot{Y}^{\mathrm{st}}_{\mathrm{c,to}} = -g. \tag{41}$$

Assuming the robot doesn't slip, the horizontal inertia force of the robot is less than the static friction at the foot,

$$\left|\ddot{X}^{\mathrm{st}}_{\mathrm{c}}\right| \leq \mu F_{\mathrm{y}}/M_{\mathrm{tot}}, \tag{42}$$

where μ is the friction coefficient between the support foot and the ground.

https://doi.org/10.1017/S0263574710000378 Published online by Cambridge University Press

3.4.3. The zero moment point. Assuming the support foot does not rotate during the stance phase, the ZMP should be located within the range of the support foot. The ZMP for the planar biped follows from standard arguments[20]

$$X_{\text{zmp}} = \frac{\sum_{i=1}^{5} M_i \left[X_{\text{c,i}}^{\text{st}} \left(\ddot{Y}_{\text{c,i}}^{\text{st}} + g\right) - Y_{\text{c,i}}^{\text{st}} \ddot{X}_{\text{c,i}}^{\text{st}}\right] + \sum_{i=1}^{5} I_i \ddot{\theta}_i}{M_{\text{tot}} \left(\ddot{Y}_{\text{c}}^{\text{st}} + g\right)}. \tag{43}$$

with the restriction that

$$-l_{\text{f2}} \leq X_{\text{zmp}} \leq l_{\text{f1}}. \tag{44}$$

where l_{f1} and l_{f2} are the front and rear lengths of the support foot, as measured from the ankle pivot point.

4. Implementation of the Gait Generator

Dynamic optimization can produce the running gaits, in the form of a torque profile for each joint. Since the equations of motion (11) and (36) are highly nonlinear, a solution requires numerical methods. Our work uses Matlab's $fmincon(\cdot)$, available with the optimization toolbox.

The two running phases can be optimized separately, or together as a half-running-cycle. A five step optimization procedure is used in this work:

(1) Pick Θ_{to} and Θ_{td}.
(2) Search for initial joint velocities of the flight phase $\dot{\Theta}_{\text{to}}$, given L_{s}, h_{s}, V_x, $\mathcal{O}_2$ and γ.
(3) Optimize the flight phase.
(4) Optimize the stance phase.
(5) Re-optimize the generated flight phase and stance phase together as a half-running-cycle.

4.1. Single-phase optimization

Single phase optimization proceeds on the two running phases separately. In general, a nonlinear optimization solver requires a decision vector, boundary values of the decision vector, a cost function and some constraints (see ref. [21] for more details.) The decision vector, denoted by U, contains the discretized joint torques τ_i for $i = 1, \ldots, 6$. The modelling method, the decision vector and the cost function are almost identical for the two phases. However, the constraints in different phases may be very different.

4.1.1. Implementation. The computer model of second-order dynamic models (11) and (36) consists of Matlab Simulink blocks. As a matter of practical implementation, a saturation block limits the output of term $H(q, \dot{q})\dot{q}$ to ensure the integrator does not fail during intermediate simulations required in the optimization. The final optimized solution, however, does not saturate.

The definition of the decision vector to be minimized is

$$U = \left[\tau(1)^T, \tau(2)^T, \ldots, \tau(k)^T, \ldots, \tau(N)^{\text{T}}\right]^T, \tag{45}$$

where N is the number of discretized time intervals, k indicates the kth discretized time interval and $\tau(k) = [\tau_1(k), \tau_2(k), \tau_3(k), \tau_4(k), \tau_5(k), \tau_6(k)]^T$. The initial estimate of U is U_0, a zero vector with the same dimension as U. The torque limits of the joints form the bounds of the decision vector,

$$\tau_{i,\min} \leq \tau_i \leq \tau_{i,\max}. \tag{46}$$

Correspondingly, the decision vector is bounded by $U_{\min}$ and $U_{\max}$ which are appropriate formations of $\tau_{i,\min}$ and $\tau_{i,\max}$.

The cost function in each running phase has the same form

$$\mathcal{C} = \frac{1}{2} U^{\text{T}} U \Delta t, \tag{47}$$

where Δt is the time interval (assuming the phase duration, T^{fl} or T^{st}, is equally discretized,) and it may be different in the two phases. This cost function implies the *least control effort*.

4.1.2. Constraints in the flight phase. In the flight phase, the boundary joint angles, the initial joint velocities and Eq. (31) are the equality constraints. The linear inequality constraints include all physical joint limits. Besides (30), the following constraints must be satisfied:

$$\Theta_{i,\min}^{\text{fl}} \leq \Theta_i^{\text{fl}} \leq \Theta_{i,\max}^{\text{fl}},$$
$$\alpha_{\min}^{\text{fl}} \leq \alpha^{\text{fl}} \leq \alpha_{\max}^{\text{fl}},$$

where α is the relative angle of a knee.

4.1.3. Constraints in the stance phase. In the stance phase, the boundary joint angles, the boundary joint velocities and Eq. (41) are the equality constraints. The linear inequality constraints could be written in the same way as those in the flight phase, with the superscript 'fl' being replaced by 'st'. In general, the values of the bounds in the stance phase may not be equal to their counterparts in the flight phase but in our implementation they are identical.

Constraints defined by Eqs. (39), (40), (42) and (44) impose the most important nonlinear inequality constraints.

During the stance phase, the swing foot may scuff the ground if the foot position $Y_{\text{a,sw}}$ is too low, where subscript 'sw' indicates swing. Recall the assumption that the foot remains parallel the ground and consider

$$Y_{\text{a,sw}} \geq Y_{\text{a,sw,min}}, \tag{48}$$

as an additional nonlinear inequality constraint.

4.2. Optimization of an entire half-running-cycle

The half-running-cycle optimization synthesizes the two running phases together. The cost function evaluating the control effort of the half-cycle is

$$\mathcal{C}_1 = \frac{1}{2}\left[\sum_{i=1}^{N^{\text{fl}}} (U^{\text{fl}})^{\text{T}} (U^{\text{fl}}) \Delta t^{\text{fl}} + \sum_{i=N^{\text{fl}}+1}^{N^{\text{fl}}+N^{\text{st}}} (U^{\text{st}})^{\text{T}} (U^{\text{st}}) \Delta t^{\text{st}}\right], \tag{49}$$

where N is the number of discretized time intervals. The flight and stance constraints remain the same, but are now included as one constraint function. The decision vector for the half-running-cycle optimization is $U = [(U^{\text{fl}})^{\text{T}}, (U^{\text{st}})^{\text{T}}]^{\text{T}}$.

https://doi.org/10.1017/S0263574710000378 Published online by Cambridge University Press

Table III. Typical transportation costs of running.

Species	J/Nm	Data sources
Humans	0.280	[22]
Monopod I	0.70	[9]
Monopod II	0.22	[9]
HRP-2L	3.57	Estimated from[13]

Table IV. The constants in the simulations.

Constants	Values	Constants	Values
M_1	8 kg	I_1	0.4 kg-m^2
$M_{2/3}$	1 kg	$I_{2/3}$	0.02 kg-m^2
$M_{4/5}$	0.8 kg	$I_{4/5}$	0.01 kg-m^2
l_1	0.6 m	r_1	0.4
$l_{2/3}$	0.35 m	$r_{2/3}$	0.6
$l_{4/5}$	0.4 m	$r_{4/5}$	0.6
$l_{\rm f1}$	0.15 m	$l_{\rm f2}$	0.05 m
g	9.81 m/s^2	$Y_{\rm a,sw,min}$	0.05 m
$\theta_{1,\min}$	45°	$\theta_{1,\max}$	90°
$\theta_{2/3,\min}$	60 deg	$\theta_{2/3,\max}$	225°
$\theta_{4/5,\min}$	30°	$\theta_{4/5,\max}$	120°
$\alpha_{\min}$	0°	$\alpha_{\max}$	150°
$\dot\theta_{1,\min}$	−4 rad/s	$\dot\theta_{1,\max}$	4 rad/s
$\dot\theta_{2/3,\min}$	−9 rad/s	$\dot\theta_{2/3,\max}$	9 rad/s
$\dot\theta_{4/5,\min}$	−6 rad/s	$\dot\theta_{4/5,\max}$	6 rad/s
$\tau_{1/2,\min}$	−80 Nm	$\tau_{1/2,\max}$	80 Nm
$\tau_{3/4,\min}$	−40 Nm	$\tau_{3/4,\max}$	40 Nm
$\tau_{5/6,\min}$	−40 Nm	$\tau_{5/6,\max}$	40 Nm
μ	0.6	$F_{\rm y,max}$	$2M_{\rm tot}g$ N

4.3. A performance index

In the flight phase, the actuators do work

$$W^{\rm fl} = \sum_{i=0}^{N^{\rm fl}-1} \int_{t_{\rm to}+i\,\Delta t^{\rm fl}}^{t_{\rm to}+(i+1)\Delta t^{\rm fl}} \sum_{j=1}^{5} \left|\dot q_{2,j}^{\rm fl}\right| \left|\left(B_2^{\rm fl}\tau^{\rm fl}\right)_j\right| dt. \quad (50)$$

Note that each joint torque $\tau_j^{\rm fl}$ is a piecewise constant signal. The work in the stance phase is similar, but with 'fl' replaced by 'st'. The total work in a half-running-cycle is

$$W = W^{\rm fl} + W^{\rm st}. \quad (51)$$

Enabling us to compare the energy efficiency of the robot, a performance index, *the cost of transport*, is defined as in ref. [7, 22] for level ground

$$\epsilon = \frac{W}{M_{\rm tot} g L_{\rm s}}, \quad (52)$$

and for stair-running

$$\epsilon = \frac{W}{M_{\rm tot} g \sqrt{L_2^2 + h_s^2}}. \quad (53)$$

A smaller cost of transport implies higher energy efficiency. For comparison, the costs of running for humans, Monopod I, Monopod II and the HRP-2L are listed in Table III. The cost of transport of the HRP-2L was not explicitly given in ref. [13] so our value is estimated by using the provided specifications (but it may be high because of unmodelled inefficiencies in motors and gears). Note that efficiency of the Monopod II is higher than that of humans, showing the springs do indeed help to reduce energy consumption.

Table V. The parameters for running on even ground.

Parameters	Values
$L_{\rm s}$	0.6 m
$h_{\rm s}$	0 m
$V_{\rm x}$	1.6 m/s
$\Theta_{\rm to}$	$(80, 100, 140, 70, 60)^{\rm T}$ deg
$\Theta_{\rm td}$	$(80, 100, 115, 45, 100)^{\rm T}$ deg
γ	0.3
$k_{\rm x}$	0.5
$k_{\rm y}$	0.5

5. Simulation Results and Discussions

The simulated robot runs on even ground and up stairs. The simulation produces two complete running cycles for the different gaits. The constants listed in Table IV apply in all simulations. The robot achieves a desired (input) step length $L_{\rm s}$, landing height $h_{\rm s}$ and average forward speed $V_{\rm x}$.

5.1. Results from choosing joint velocities

The even ground simulations use the parameters listed in Table V and generate a complete running cycle using the 5-step optimization procedure. To demonstrate the usefulness of our proposed method of calculating initial joint velocities, we first show results using chosen velocities. The algorithm for choosing joint velocities follows:

(1) Try to choose $\dot\theta_1$ close to zero, because the pitch angle of the upper body, θ_1, changes in a small range due to the much larger mass compared to other links.
(2) Try to choose $\dot\theta_4$ and $\dot\theta_5$ close to zero, since the ankle actuators simply ensure the feet are parallel with the ground in the flight phase and the resulting ankles velocities are usually small.
(3) Calculate $\dot\theta_2$ and $\dot\theta_3$ using the initial velocity constraints $\dot P_{g,x} = V_x$ and $\dot P_{g,y} = V_{y,0}$.
(4) Are $\dot\theta_2$ and $\dot\theta_3$ within physical limits? If yes, keep them; if no return to step 1 using a different guess.

Using $\dot\Theta_{\rm to} = (1, -6.4893, 7.8105, 2, -2)^T$ results in a solution that satisfies the constraints (Fig. 2). The flight phase consumes energy 14 J and the stance phase consumes 390 J. The cost of transport is 5.9 J/N-m.

5.2. Running on even ground

Using the proposed method of calculating initial joint velocities results in an elegant-looking gait (Fig. 4). All joint torques fall within the corresponding ranges (Fig. 5). The largest torque applied in the stance phase is only 72 Nm, compared to 135 Nm when the using the trial-and-error initial velocity method (Fig. 3). In each half-running-cycle, the energies consumed in the flight phase and the stance phase are 12 J and 58 J, respectively. The stance phase,

https://doi.org/10.1017/S0263574710000378 Published online by Cambridge University Press

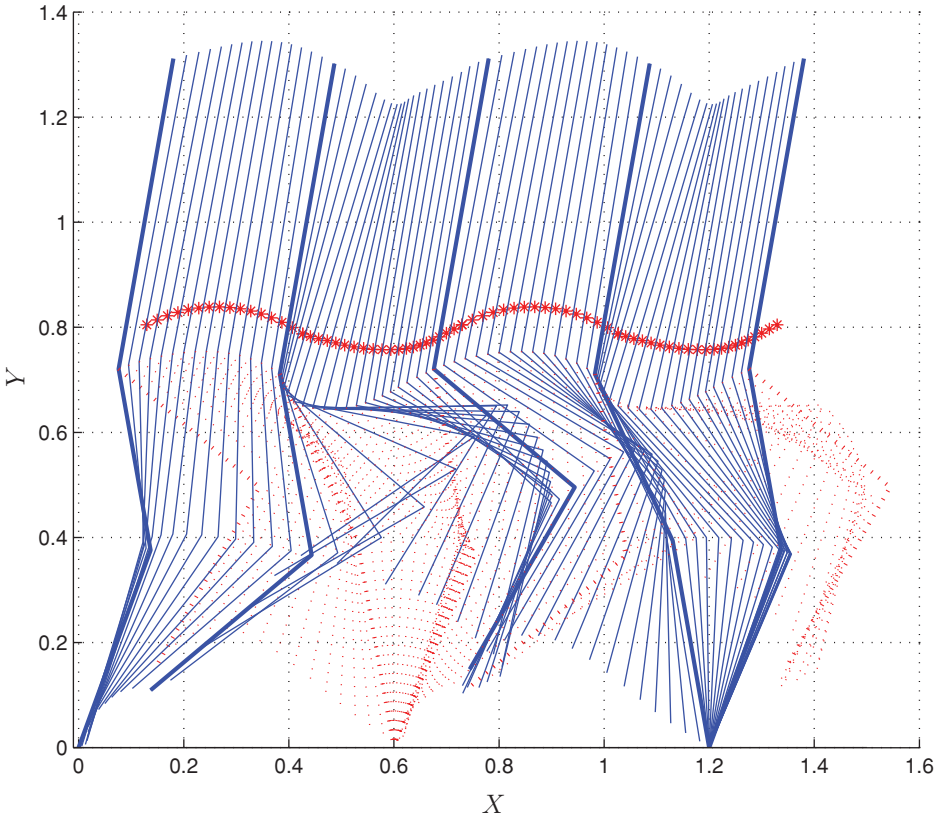


Fig. 2. Example: a gait resulting from choosing boundary velocities. Right leg and upper body ' − ', left leg ' · · · ', CoM trajectory ' ∗ '.

then, consumes only 15% of the energy of the solution found when choosing initial velocities by trial and error. The flight phase consumes 17% of the total energy of the half-running-cycle. Calculating the cost of transport of the simulated robot produces 1.02 J/N-m. Assuming that the transmission efficiency of the robot is about 50%, we estimate that the true cost of transport runs about 2 J/Nm. The cost appears to be higher than humans and the spring-loaded Monopods I

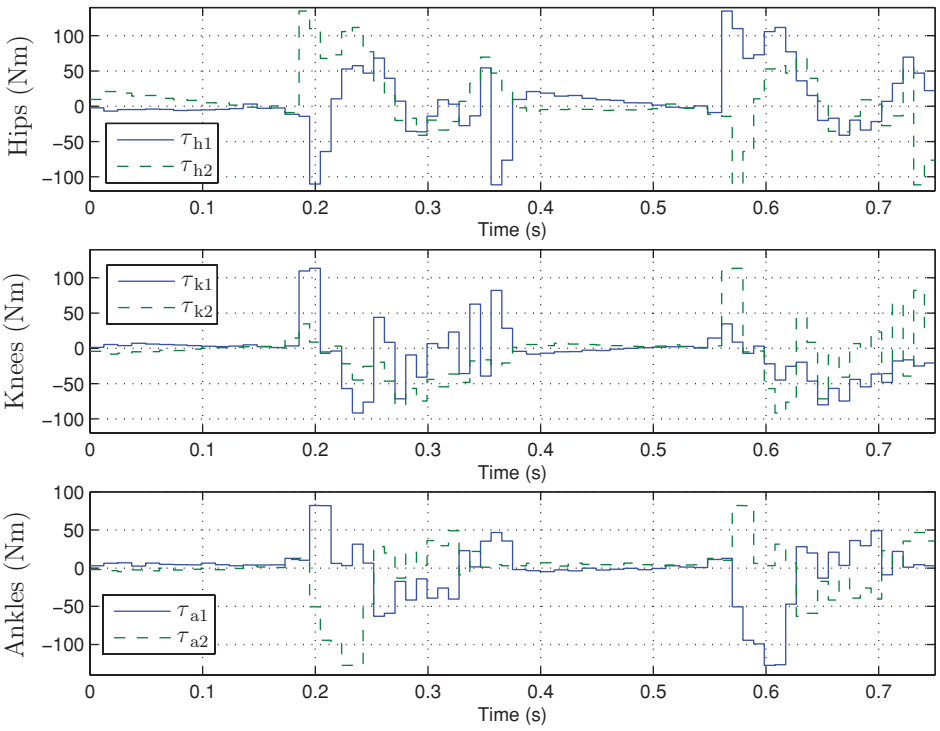


Fig. 3. Example of choosing velocities: control torques.

https://doi.org/10.1017/S0263574710000378 Published online by Cambridge University Press

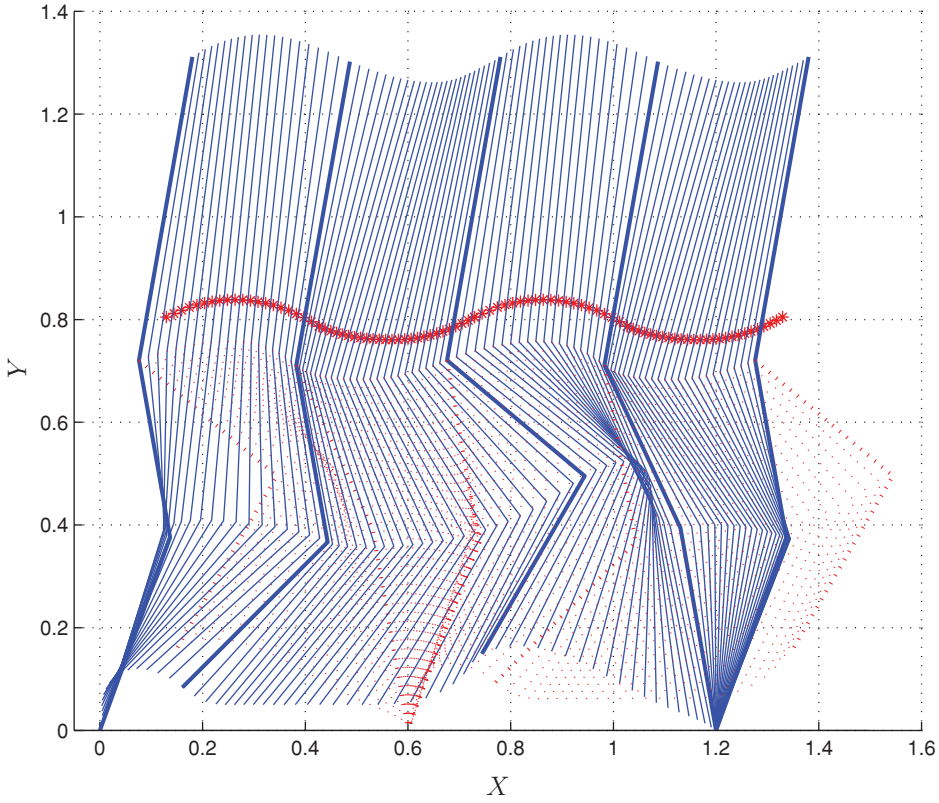


Fig. 4. Running on even ground. Right leg and upper body ' − ', left leg ' ⋯ ', CoM trajectory ' ∗ '.

and II, but possibly lower than the rigid-joint HRP-2L (Table III).

In the stance phase, the trajectory of the CoM behaves like an elastic inverted pendulum, consistent with the SLIP model. At mid-stance the CoM arrives at its minimum height. The joint angles are continuous (Fig. 6) whereas the joint velocities jump due to the collision between the foot and the ground (Fig. 7). In reality, the collisions

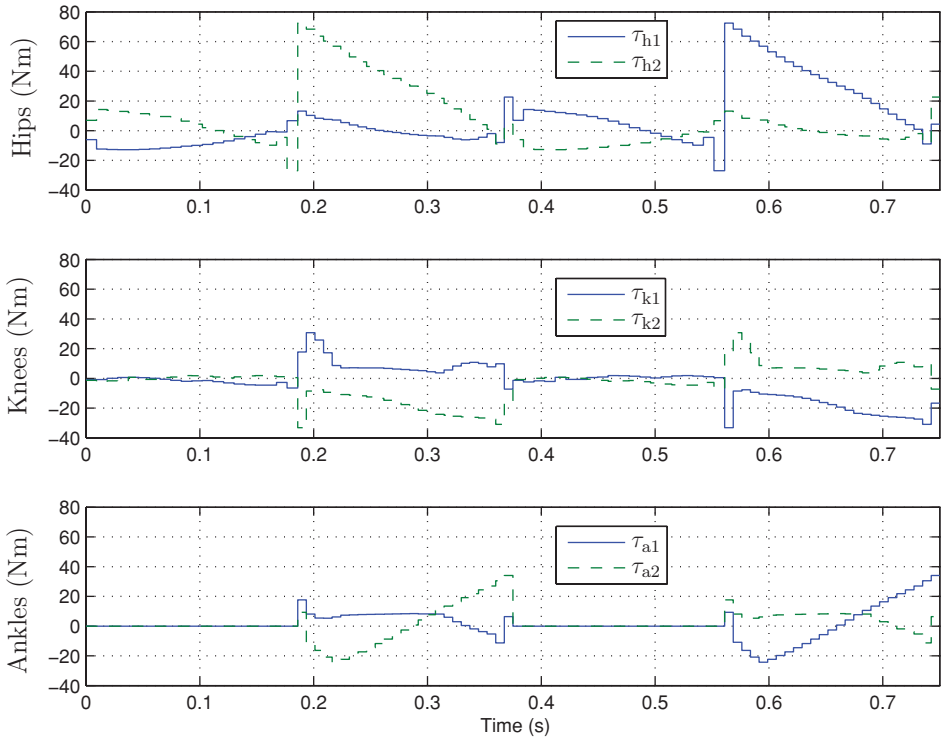


Fig. 5. Control torques for running on even ground.

https://doi.org/10.1017/S0263574710000378 Published online by Cambridge University Press

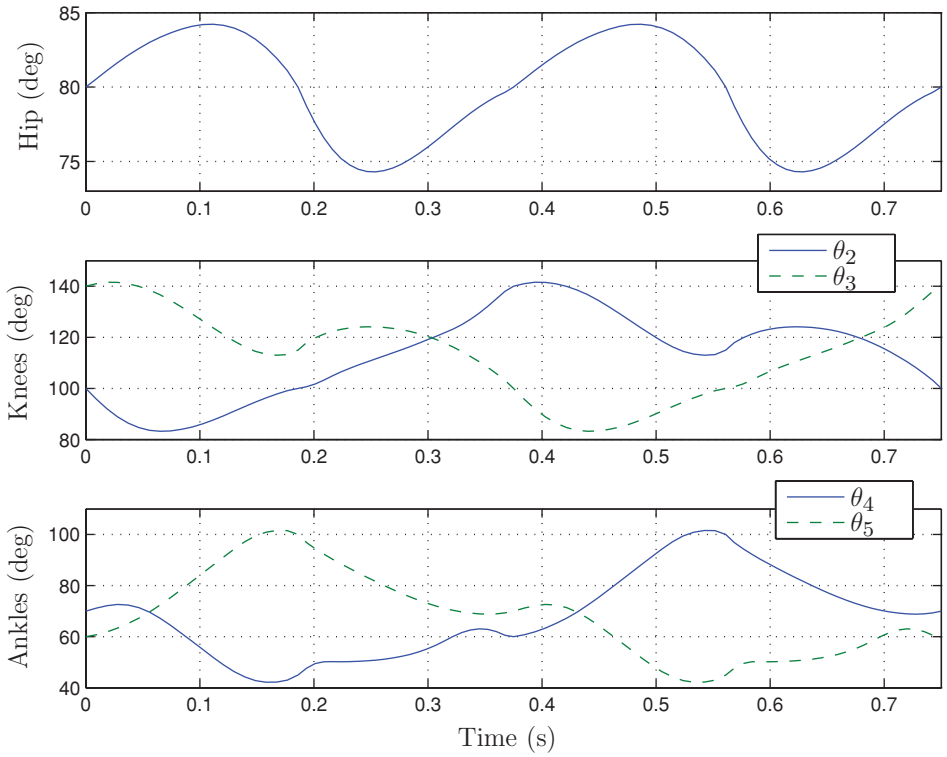


Fig. 6. Absolute joint angles for running on even ground.

should exhibits some elasticity resulting in smaller velocity jumps.

The neutral position of the ZMP, where $X_{\text{zmp}} = 0$, coincides with the ankle positions in the inertial coordinate (Fig. 8 top). Note that the ZMP is not defined in the flight phase. Since the ZMP lies within the support foot, the robot will not tip over. The vertical acceleration of the CoM grows from 5.5 m/s^2 to g, and then decreases to $-g$ (Fig. 8b). Thus, the normal GRF increases from 178 N to $F_{\text{y,max}}$, then decreases dramatically before finally vanishing at take-off (Fig. 8 bottom). Interestingly, the normal GRF appears in an M-shape which is rare in human running. Due to Eq. (42),

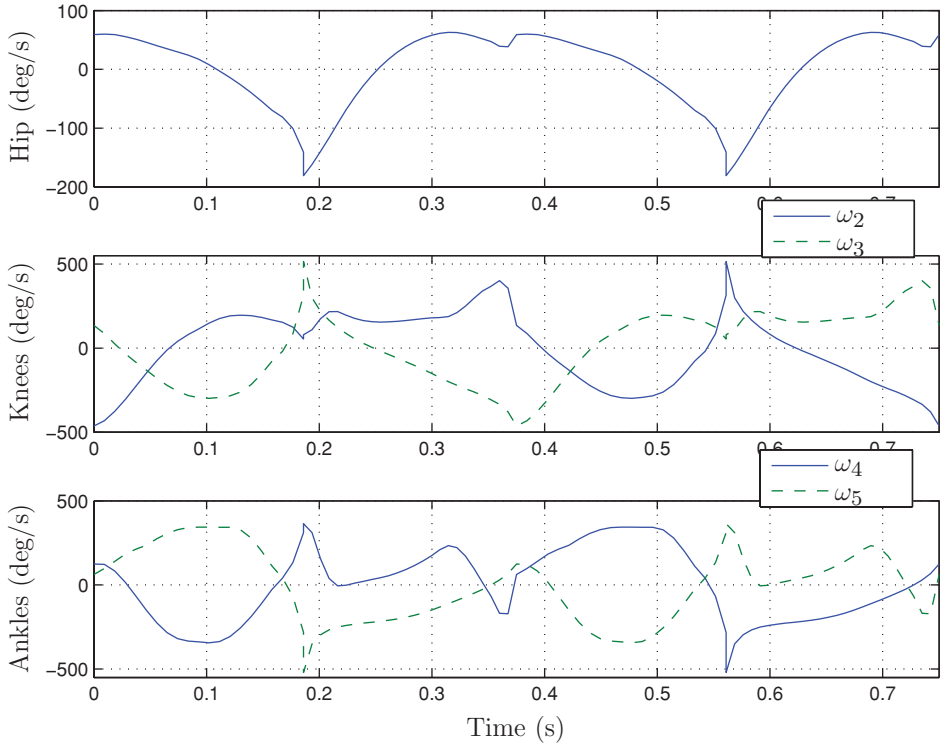


Fig. 7. Joint velocities for running on even ground.

https://doi.org/10.1017/S0263574710000378 Published online by Cambridge University Press

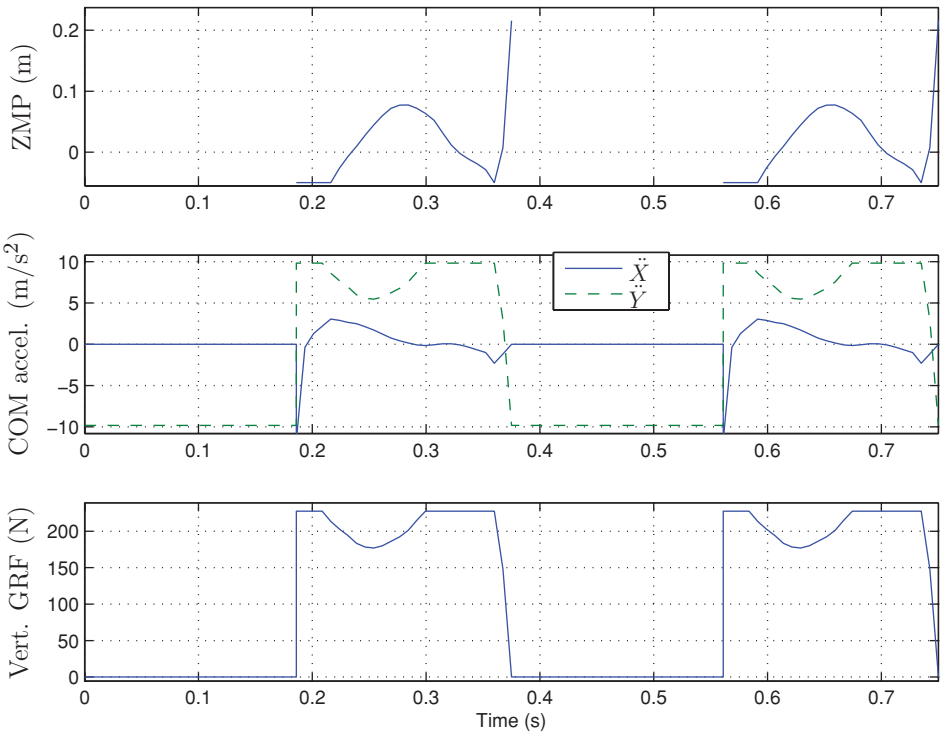


Fig. 8. The ZMP trajectory (a), acceleration components of the CoM (b) and the vertical GRF (c) for running on even ground.

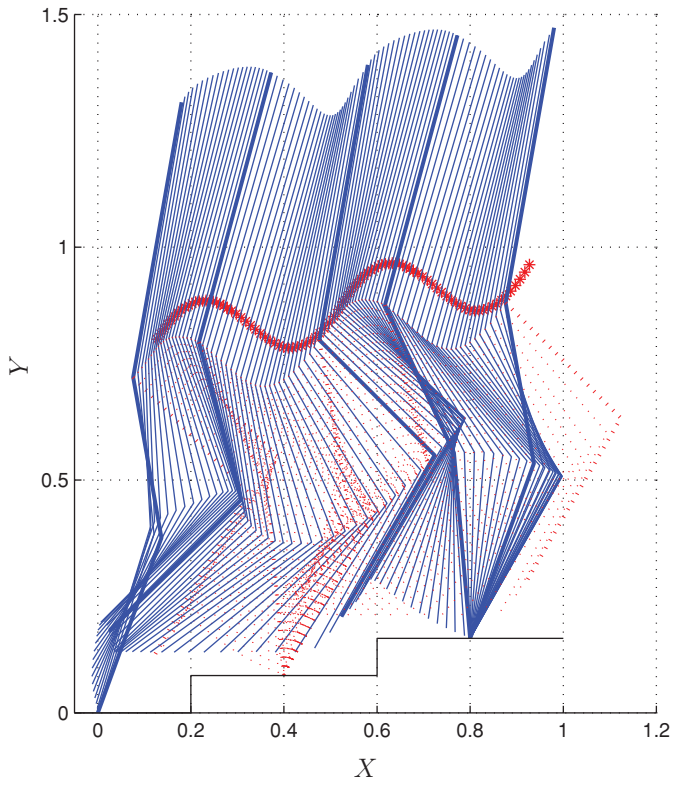


Fig. 9. A gait cycle for running up stairs. Right leg and upper body ' − ', left leg ' ⋯ ', CoM trajectory ' ∗ '.

the horizontal acceleration of the CoM becomes zero at the end of the stance phase (Fig. 8b). Therefore, the horizontal acceleration of the CoM remains continuous over the whole running cycle.

Table VI. The parameters for running up stairs.

Parameters	Values	Units
L_s	0.2	m
h_s	0.08	m/s
V_x	0.8	m/s
Θ_{to}	$(80, 100, 135, 70, 60)^T$	deg
Θ_{td}	$(75, 105, 115, 45, 95)^T$	deg
γ	0.3	–
k_x	0.5	–
k_y	0.8	–

5.3. Running up stairs

Using the parameters listed in Table VI results in the robot running up stairs (Fig. 9). The joint angles (Fig. 10), the joint velocities (Fig. 11), the joint torques (Fig. 12) differ in a reasonable manner from the even-ground results. The ZMP trajectories, the horizontal and vertical accelerations of the CoM and the normal GRF appear in Fig. 13.

The flight phase and the stance phase use 11 and 65 J of energy, respectively. The flight phase consumes 16% of the total energy of the half-running-cycle. The calculated cost of transport of the robot is about 1.7 J/Nm which is probably 3.4 J/Nm accounting for transmission efficiency. As expected, the robot loses efficiency when running up stairs compared to even ground (even though the average forward speed is reduced from 1.6 m/s to 0.8 m/s).

5.4. Some remarks

The flight phases consume relatively little energy compared to the stance phases, which provides evidence that our

https://doi.org/10.1017/S0263574710000378 Published online by Cambridge University Press

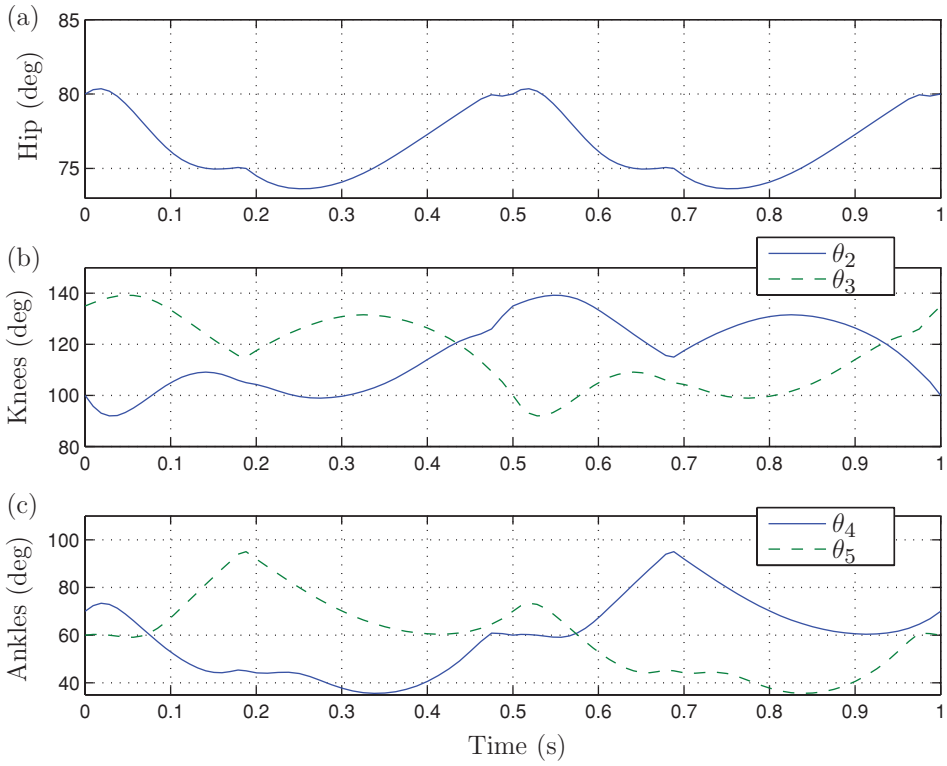


Fig. 10. Absolute joint angles for running up stairs.

method for calculating take-off velocities is a valid approach. This applies for running on even ground and up stairs.

We chose to set the maximum normal GRF, $F_{y,max}$, to be twice the robot's weight. Although a reasonable value for humans and animals, this value may be too large for a real running robot. A physical robot may require active force control or passive compliant feet in order to partly damp the large vertical GRF.[18,19] The static friction coefficient $\mu = 0.6$ represents a typical real-world value. However, if smaller $F_{y,max}$ and μ did occur, the robot could reduce average forward speed V_x, the landing height h_s, or the stride length L_s to achieve safe running.

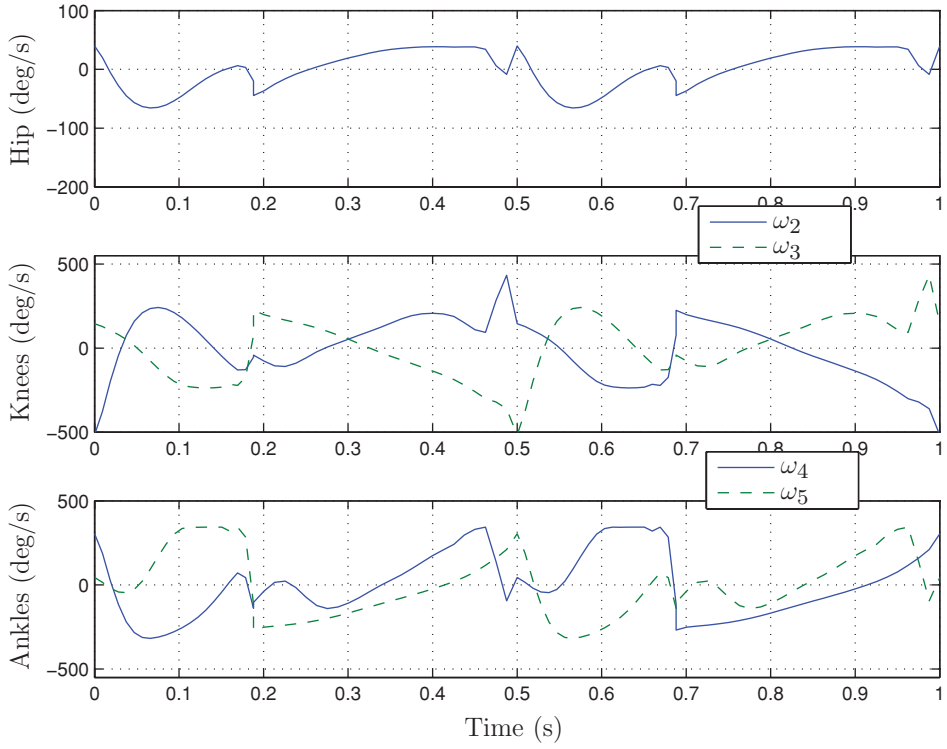


Fig. 11. Joint velocities for running up stairs.

https://doi.org/10.1017/S0263574710000378 Published online by Cambridge University Press

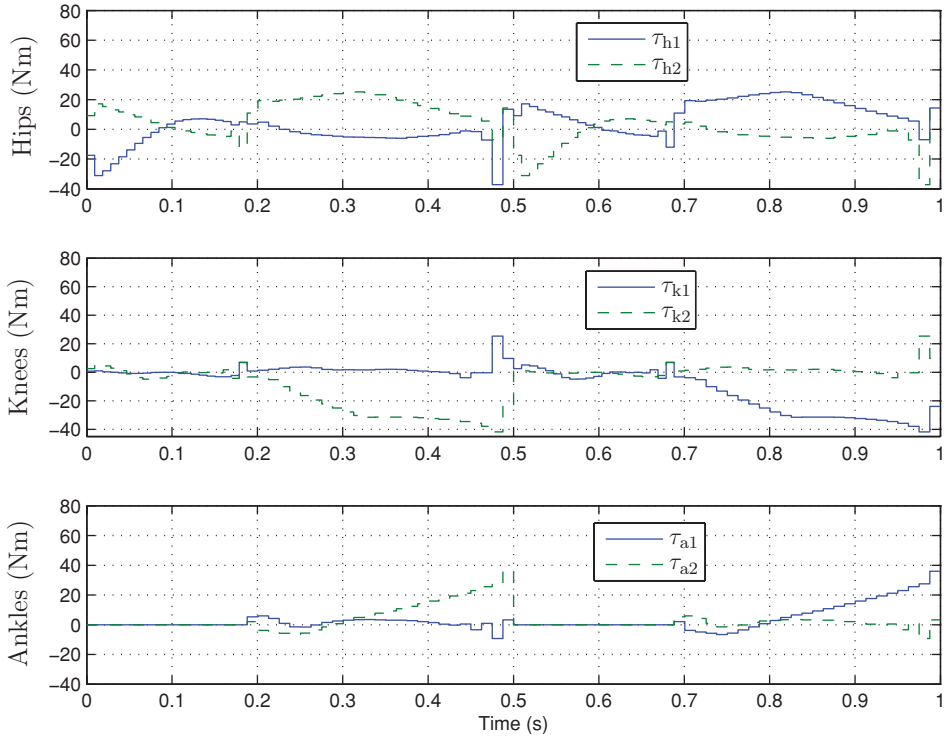


Fig. 12. Control signals for running up stairs.

The use of the ZMP criterion in synthesis of running-like gaits is controversial[23,24] and it may be too conservative for high-speed motions. We justify our use of ZMP as nonlinear inequality constraint here precisely because it is a conservative approach. Elimination of the ZMP constraint would significantly reduce the number of search iterations, and thus appears to be a logical approach for future improvement.

Our definition of the joint angles requires measurement from the horizontal, rather than the relative angles between two neighboring links. This definition greatly simplifies model of the robot and speeds up computations. Also,

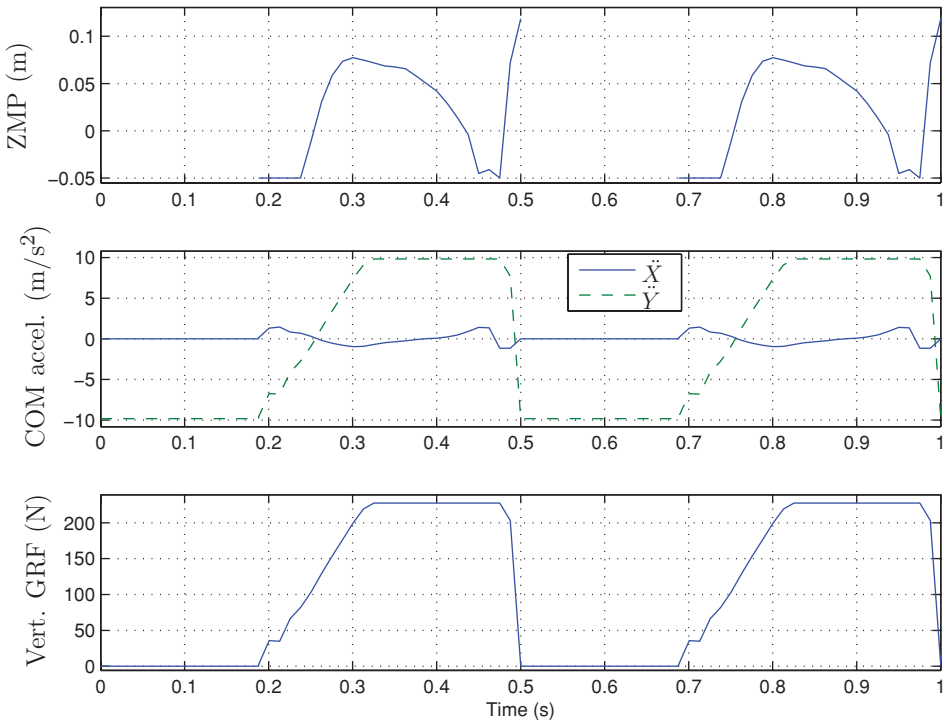


Fig. 13. The ZMP trajectory, acceleration components of the CoM and the vertical GRF for running up stairs.

https://doi.org/10.1017/S0263574710000378 Published online by Cambridge University Press

designing constraints for the dynamic optimization that prevent overturning in the flight phase becomes much easier. However, a designer might find it difficult to provide measurement of these absolute angles. Gyroscopes or inclinometers on the upper body of the robot might provide a sufficiently accurate estimate of the horizontal, since the upper body moves very little compared to the legs. Online observers might provide another solution to this problem, making use of known relative angle measurements.

The rigid-joint design appears to achieve better energy efficiency than other rigid-joint approaches. The rigid-joint biped clearly does not achieve the energy-efficiency of designs that incorporate compliant elements. However, we feel that this approach lays a useful mathematical framework for future research into bipedal running, including compliant-joint designs.

6. Conclusions

This paper presents a novel approach for generating bipedal running gaits. Analyzing the energy expenditure of the robot allows us to calculate suitable take-off velocities using static optimization. Minimization of a novel objective function produces the initial joint velocities of the flight phase. A dynamic optimization procedure then provides the two running phases, with a simple collision model predicting the abrupt joint velocity changes caused by the foot/ground collision. Ground reaction forces, the Zero-Moment Point and foot clearance of the swing leg serve as nonlinear constraints. Simulation results show that the generated running gaits, both on even ground and up stairs, resemble human running. The energy consumption in the two cases appears to be acceptable, for a rigid robot. Improving energy-efficiency further will probably require adding springs into the design.

Acknowledgments

This work was supported in part by grants from the Alberta Ingenuity Fund and the Natural Sciences and Engineering Research Council of Canada.

References

1. Y. Fujimoto, 'Trajectory Generation of Biped Running Robot with Minimum Energy Consumption,' *Proceedings of IEEE International Conferance on Robotics and Automatation*, New Orleans, LA (2004) pp. 3803–3808.
2. M. Srinivasan and A. Ruina, 'Computer optimization of a minimal biped model discovers walking and running,' *Nature* (439), 72–75 (2006).
3. M. Raibert, 'Running with symmetry,' *Int. J. Robot. Res.* **5**(4), 3–19 (1986).
4. R. M. Alexander, 'Walking and running,' *Math. Gaz.* **80**(488), 262–266 (1996).
5. H. Geyer, A. Seyfarth and R. Blickhan, 'Compliant leg behaviour explains basic dynamics of walking and running,' *Proc. R. Soc. Lond. Ser.* B*: Biol. Sci.* **273**(1603), 2861–2867 (2006).
6. G. Cavagna, H. Thys and A. Zamboni, 'The sources of external work in level walking and running,' *J. Physiol.* 262(3), 639–657 (1976).
7. S. Collins, A. Ruina, R. Tedrake and M. Wisse, 'Efficient bipedal robots based on passive-dynamic walkers,' *Science* **307**(5712), 1082–1085 (2005).
8. P. Gregorio, M. Ahmadi and M. Buehler, 'Design, control and energetics of an electrically actuated legged robot,' **27**, 626–634 (1997).
9. M. Ahmadi and M. Buehler, 'The ARL Monopod II Running Robot: Control and Energetics,' *IEEE International Conferance on Robotics and Automatation*, Detroit, MI (1999) pp. 1689–1694.
10. S. Hyon, T. Emura and T. Mita, 'Dynamics-based control of a one-legged hopping robot,' *J. Syst. Control Eng.* **217**(2), 83–98 (2003).
11. J. Vermeulen, Trajectory Generation for Planar Hopping and Walking Robots: An Objective Parameter and Angular Momentum Approach *Ph.D. Dissertation* (Brussels, Belgium: Vrije Universiteit, 2004).
12. F. Iida, J. Rummel and A. Seyfarth, 'Bipedal Walking and Running with Compliant Legs,' *Proceedings of IEEE Internatioanl Conferance Robotics and Automattion*, Roma, Italy (2007) pp. 3971–3975.
13. T. Nagasaki, S. Kajita, K. Yokoi, K. Kaneko and K. Tanie, 'Running Pattern Generation and its Evaluation Using a Realistic Humanoid Model,' *Proceedings of IEEE International Conferance on Robotics and Automation*, Taipei, Taiwan (2003) pp. 1336–1342.
14. S. Kajita, K. Kaneko, M. Morisawa, S. Nakaoka and H. Hirukawa, 'Running Pattern Generation and its Evaluation Using a Realistic Humanoid model,' *Proceedings of IEEE International Conferance on Robotics and Automation*, Roma, Italy (2007) pp. 3963–3969.
15. B. Morris, E. Westervelt, C. Chevallereau, G. Buche and J. Grizzle, 'Passive bipedal walking with phasic muscle contraction,' *Biol. Cybern.* 81(3), 227–237 (1999).
16. A. Meghdari and M. Aryanpour, 'Dynamic modeling and analysis of the human jumping process,' *J. Intell. Robot. Syst.* **37**(1), 97–115 (2003).
17. Q. Guo, C. Macnab and J.Pieper, 'Hopping on even ground and up stairs with a single articulated leg,' *J. Intell. Robot. Syst.* **53**(4), 331–358 (2008).
18. K. Hirai, M. Hirose, Y. Haikawa and T. Takenaka, 'The Development of Honda Humanoid Robot,' *Proceedings of IEEE International Conferance on Robotics and Automation,* Leuven, Belgium (1998), 1321–1326.
19. S. Collins, M. Wisse and A. Ruina, 'A three-dimensional passive-dynamic walking robot with two legs and knees,' *Int. J. Robot. Res.* **20**(2), 607–615 (2001).
20. M. Popovic, A. Goswami and H. Herr, 'Ground reference points in legged locomotion: Definitions, biological trajectories and control implications,' *Int. J. Robot. Res.* **24**(12), 1013–1032 (2005).
21. V. Becerra, 'Solving optimal control problems with state constraints using nonlinear programming and simulation tools,' *IEEE Trans. Educ.* **4**(3), 377–384 (2004).
22. T. Roberts, R. Kram, P. Weyand and R. Taylo, 'Energetics of bipedal running, Part I. Metabolic cost of generating force,' *J. Exp. Bio.* 201, 2745–2751 (1998).
23. K. Kondak and G. Hommel, 'Control of Online Computation of Stable Movement for Biped Robots,' *Proceedings of IEEE/RSJ Interantional Conferance on Intelligent Robots and Systems*, Las Vegas, NV (2003) 874–879.
24. E. Westervelt, 'Toward a Coherent Framework for the Control of Planar Biped Locomotion *Ph.D. Dissertation* (Ann Arbor, MI: University of Michigan, 2003).

https://doi.org/10.1017/S0263574710000378 Published online by Cambridge University Press