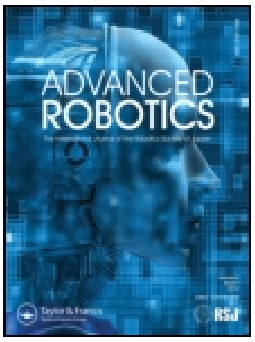

Taylor & Francis
Taylor & Francis Group

Advanced Robotics

ISSN: 0169-1864 (Print) 1568-5535 (Online) Journal homepage: www.tandfonline.com/journals/tadr20

Stable active running of a planar biped robot using Poincare map control

Behnam Dadashzadeh, M.J. Mahjoob, M. Nikkhah Bahrami & Chris Macnab

To cite this article: Behnam Dadashzadeh, M.J. Mahjoob, M. Nikkhah Bahrami & Chris Macnab (2014) Stable active running of a planar biped robot using Poincare map control, Advanced Robotics, 28:4, 231-244, DOI: 10.1080/01691864.2013.865299

To link to this article: https://doi.org/10.1080/01691864.2013.865299

Published online: 19 Dec 2013.

Advanced Robotics, 2014
Vol. 28, No. 4, 231–244, http://dx.doi.org/10.1080/01691864.2013.865299

Taylor & Francis
Taylor & Francis Group

FULL PAPER

Stable active running of a planar biped robot using Poincare map control

Behnam Dadashzadeh[a,b]*, M.J. Mahjoob[b], M. Nikkhah Bahrami[b] and Chris Macnab[a]

[a]*Department of Electrical and Computer Engineering, University of Calgary, Calgary, Canada;* [b]*School of Mechanical Engineering, University of Tehran, Tehran, Iran*

(*Received 18 June 2013; accepted 18 August 2013*)

This work formulates the active limit cycles of bipedal running gaits for a compliant leg structure as the fixed point of an active Poincare map. Two types of proposed controllers stabilize the Poincare map around its active fixed point. The first one is a discrete linear state feedback controller designed with appropriate pole placement. The discrete-time control first uses purely constant torques during stance and flight phase, then discretizes each phase into smaller constant-torque intervals. The other controller is an invariant manifold based chaos controller: a generalized Ott, Grebogi and Yorke controller having a linear form and a nonlinear form. Both controllers can stabilize active running gaits on either even or sloped terrains. The efficiency of these controllers for bipedal running applications are compared and discussed.

Keywords: underactuated bipedal running; Poincare map; fixed point; state feedback; chaos control

1. Introduction

Through three decades of research on legged locomotion, researchers continue to seek methods that would allow a biped robot able to run as efficiently and quickly as living creatures with the same amount of stability. The design problem subdivides into two main parts: the mechanical structure and the control system. A useful mechanism should be simple, robust, and energy efficient. The controller should be able to deal with the highly nonlinear dynamics, containing discontinuous events, using limited actuator power. This paper focuses on achieving steady and stable running with an underactuated planar biped robot, addressing both the mechanical design and control system.

Underactuated robots have fewer control inputs than number of degrees of freedom. The class of underactuated biped robots contains purely passive robots as well as those with some active and some passive joints. A robot with point feet provides an example of a common underactuated biped, since there is no ankle actuation. While flat-footed biped robots are often controlled by zero moment point stability criterion – showing unnatural and slow walking and running gaits [1] – underactuated robots demonstrate more natural dynamics and faster gaits. Many researchers have looked at gait generation and control of underactuated walking biped robots.[2] The control design for a running multi-body biped robot (rather than a spring loaded inverted pendulum model) encounters more difficulties. Running robots contain complicated hybrid dynamics, more degrees of underactuation in flight phase, and higher impact at touchdown. Some of the more notable experimental underactuated biped robots include Raibert's hopper [3], McGeer's passive walker,[4] RABBIT,[5] MABEL [6] and ATRIAS.[7] In the early 1980's, Raibert successfully demonstrated two-dimensional and three-dimensional one-legged hopping and running, by dividing the controller into three simple parts. In the 1990's, McGeer's pioneering work to design a passive biped walker inspired many researchers to investigate passive robots [8] and to design passivity-based controllers for active robots.[9] RABBIT, a planar underactuated robot that uses a time-invariant controller, demonstrated stable walking and also a few steps of running. MABEL, a planar robot, uses compliances in parallel to motors and can walk and run stably using a hybrid zero dynamics (HZD) controller.[10] The HZD method uses a set of virtual constraints and defines a low-dimensional sub-model to describe the complex dynamic system. This method has been used to build stable active running controllers for multi-body biped models. ATRIAS, a newly developed three-dimensional biped robot, has compliances in series to motors and aims to walk and run on uneven surfaces.

The role of passive compliant elements in achieving efficient bipedal running has been proven effective using a multi-body hopper model.[11] Also it has been shown that the SLIP model, the simplest compliant biped model consisting of a point mass and two massless springy legs,[12] describes dynamics of human bipedal walking and running.[13] Rummel et al. [14] investigated a minimal model of a planar one-legged robot (with two links,

*Corresponding author. Email: bdzadeh@ut.ac.ir a compliant knee, and a motor in the hip) in both simulation and experiment, showing it could generate a stable hopping movement using an open-loop sinusoidal motor input. Iida et al. [15] presented a simplified compliant leg structure (using linear springs as biarticular muscle tendons and hip-only actuation) which produced joint kinematics and ground reaction forces similar to human movement for both walking and (short flight phase) running. Elastic coupling of limbs may make gaits faster and more human-like.[16] In order to achieve human-muscle characteristics, some have proposed using artificial pneumatic muscles [17,18] and others series elastic actuators.[19,20] In this work, we investigate a multi-body biped model with torsional springs parallel to the motors in each joint. We show that the proposed multi-body biped model can generate running gaits with constant control commands in flight and stance. This finding may simplify bipedal running algorithms, and constitutes one of the main contributions of this paper.

A classic and powerful tool for analyzing the stability of periodic orbits of dynamic systems is the Poincare map. It converts the hybrid dynamic model of the robot to a discrete mapping. A fixed point of the map corresponds to a periodic walking or running gait of the robot. To have a stable gait, all the eigenvalues of the linearized Poincare map around its fixed point should be located inside the unit circle.[21] This method has been widely used for the stability analysis of passive gaits, for example by Kuo [22] and Piiroinen [23], and for the stability analysis of active gait controls, for example by Chevallereau [10], Hurmuzlu [24], and Cho [25]. Also, the linearized Poincare map serves as a linear discrete-time model, and has been used to design discrete linear quadratic regulator (LQR) controllers to stabilize passive running gaits. McGeer [26] proposed an LQR controller for passive running gaits using a biped model with telescoping springy legs and massless arced feet. While biped models with arced feet have been found to have some stable passive running gaits, no multi-body biped model with point feet has been found to have a stable passive running gait. Hu et al. [27] investigated passive running stabilization of a simple compliant biped model with massless point feet, using an event-based feedback controller to stabilize the passive limit cycle with an additive passivity-based control to enlarge the basin of attraction. These works were confined to biped models with massless springy legs that have passive running gaits on horizontal terrains, and could not be applied to real biped robot models that do not have a passive limit cycle. They used the Poincare map in the passive form and linearized it around its fixed point $\boldsymbol{x}^*$ and zero control values $\boldsymbol{u} = \boldsymbol{0}$. Instead, we propose using an active Poincare map in the form of $\boldsymbol{x}(k+1) = \boldsymbol{P}(\boldsymbol{x}(k), \boldsymbol{u}(k))$ and arrange all discretized control parameter values of one complete step in vector $\boldsymbol{u}$. Thus, the system can be linearized around the active fixed point $\boldsymbol{x}^*$, $\boldsymbol{u}^*$ and linear controllers will stabilize any planned active running (or walking) gaits (many research results are available for efficient bipedal gait planning [28,29]).

A major contribution of this paper is proposing an active (rather than just passive) Poincare map and fixed point, which has allowed us to extend the applications to different types of robots running on different terrains (horizontal and sloped surfaces). Another contribution of this paper is proposing a new compliant leg design that generates natural running gaits even with constant motor torques during running phases. The last novelty in this work is applying a generalized Ott, Grebogi and Yorke (OGY) invariant manifold-based controller [30] to stabilize bipedal running.

2. Dynamic modeling

Studies on the energetics and kinematics of spring-mass model show that compliant elements in legs of biped robots take an important role in both walking and running.[15] Although containing very complicated muscle-tendon neural control systems, human legs show simple spring-like behavior in running and during some walking speeds.[31] For example, a one-legged hopper with a springy passive knee achieved stable hopping motion using a harmonic input in the hip motor.[14] Using Hill-type muscles in a biped can model a human-like leg, but this may be unnecessarily complicated. In order to simulate muscle compliance with a minimalistic model, we propose a biped model in which each joint comes equipped with a rotational spring parallel to a torque motor.

This model (Figure 1) consists of a point-mass hip with two kneed legs. The leg segments (thigh and shank) have both mass and moment of inertia. There are three motors parallel to rotational springs: one in the hip and two in the knees. The model's lengths and masses model a typical human (Table 1). Inspired by the fact that human muscles change their stiffness during running,[32] we investigated changing the knee stiffness between stance and flight (stiffer in stance) in order to produce more efficient running gaits. We define the free angles of the torsion springs in B and D as $q_2 = -\pi/4$, $q_4 = -\pi/4$ and the free angle of the hip torsion spring as $q_3 - q_1 = 0$. We assume the terrain slopes at constant angle φ and our Cartesian X, Y coordinates describe absolute horizontal and vertical directions (relative to the Earth). We look only at running gaits, consisting of a stance phase (with one stance leg touching the ground and one swing leg) and a flight phase.

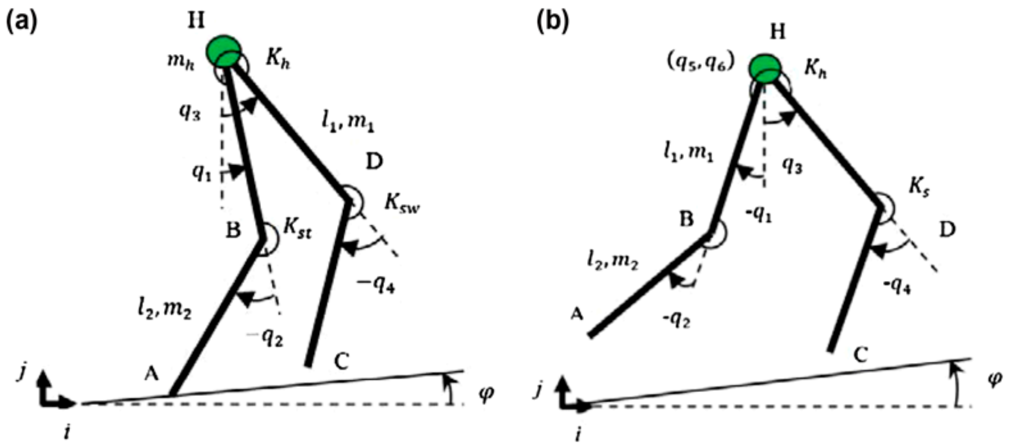

Figure 1. (a) Stance phase generalized coordinates and (b) flight phase generalized coordinates for the kneed biped.

Table 1. Nomenclature of the kneed biped robot.

Parameter	Value (SI units)	Description
$m_1, l_1, \bar{I}_1$	8, 0.45, 0.135	Mass, length, and moment of inertia of thigh BH, DH
$m_2, l_2, \bar{I}_2$	7, 0.5, 0.146	Mass, length, and moment of inertia of shank AB, CD
m_h	50	Point mass of hip at H
φ	Arbitrary	Terrain slope with respect to horizon
q_1	Variable	Angle of thigh BH with respect to vertical
q_2	Variable	Angle of shank AB with respect to thigh BH
q_3	Variable	Angle of thigh CH with respect to vertical
q_4	Variable	Angle of shank CD with respect to thigh CH
u_1	Variable	Torque of hip motor in the direction of $q_3 - q_1$
u_2	Variable	Torque of motor in knee B in the direction of q_2
u_3	Variable	Torque of motor in knee C in the direction of q_4
K_h	200	Torsion spring stiffness in hip
K_{st}	1000	Torsion spring stiffness in the knee of stance leg
K_{sw}	500	Torsion spring stiffness in the knee of swing leg

2.1. Stance phase

The stance phase generalized coordinates $[\boldsymbol{q}_s]_{4\times1}$ is comprised of angles of leg segments as shown in Figure 1(a). By utilizing the Lagrange equation,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \tag{1}$$

in which, Lagrangian function $L(\mathbf{q}, \dot{\mathbf{q}}) = T(\mathbf{q}, \dot{\mathbf{q}}) - V(\mathbf{q})$ is the subtraction of kinetic and potential energy and Q_i are the generalized forces, the stance phase dynamic equations become

$$[\mathbf{D}_s(q_s)]_{4\times4} \cdot [\ddot{\mathbf{q}}_s]_{4\times1} + [\mathbf{C}_s(\mathbf{q}_s, \dot{\mathbf{q}}_s)]_{4\times1} = [\mathbf{B}_s]_{4\times3} \cdot [\mathbf{u}_s]_{3\times1} \tag{2}$$

in which, $\mathbf{B}_s = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $\quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$.

Here, a symbolic manipulation software (Maple/Matlab toolbox) can be used to derive the analytical dynamic equations. First, x and y components of the joints positions are written as functions of generalized coordinates. Kinetic and potential energies are then found using these functions and their symbolic time derivatives and substituted in Equation (1) to derive Equation (2). By defining the stance phase state vector as $\boldsymbol{x}_s = [\mathbf{q}_s; \dot{\mathbf{q}}_s]$, the four second-order differential Equation (2) turn into eight first-order state equations. More details about the dynamic equations can be found in [33].

2.2. Take-off

A take-off consists of an instantaneous transition from stance to flight, occurring when the ground reaction force reaches zero. The generalized coordinates $\mathbf{q}_f$ in flight use the same q_1 to q_4 as in stance, but include q_5 and q_6 as the Cartesian X and Y coordinates of the robot's hip, respectively. In order to find the initial coordinates in the subsequent flight phase, use

$$\begin{gathered} q_{f1}^{+} = q_{s1}^{-}(t),\ q_{f2}^{+} = q_{s2}^{-}(t),\ q_{f3}^{+} = q_{s3}^{-}(t), \\ q_{f4}^{+} = q_{s4}^{-}(t) q_{f5}^{+} = x_H(\mathbf{q}_s^{-}) q_{f6}^{+} = y_H(\mathbf{q}_s^{-}) \end{gathered} \tag{3}$$

where subscripts s and f represent stance and flight, respectively, and superscripts $-$ and $+$ indicate time instants just before and after the event, respectively.

2.3. *Flight phase*

According to Figure 1(b), the flight phase generalized coordinates contain joints angles and components of hip position, $\mathbf{q}_f = [q_1, q_2, q_3, q_4, x_h, y_h]^T$. The dynamic equations of the flight phase thus become

$$[\mathbf{D}_f(\mathbf{q}_f)]_{6\times6}[\ddot{\mathbf{q}}_f]_{6\times1} + [\mathbf{C}_f(\mathbf{q}_f, \dot{\mathbf{q}}_f)]_{6\times1} = [\mathbf{B}_f]_{6\times3}.[\mathbf{u}_f]_{3\times1} \tag{4}$$

2.4. *Touchdown*

To detect a touchdown instant during numerical simulation, we have to find the first intersection of the trajectory of foot point D and the line $y = x\tan\varphi$. We use Lagrange's impact model to find the collision map at touchdown

$$\mathbf{D}_f(\mathbf{q}_f).(\dot{\mathbf{q}}_f^+ - \dot{\mathbf{q}}_f^-) = \hat{Q} \tag{5}$$

in which D is the inertia matrix from

$$T = \frac{1}{2}\dot{\mathbf{q}}_f^T\mathbf{D}_f(\mathbf{q}_f)\dot{\mathbf{q}}_f \tag{6}$$

The principle of virtual work provides generalized impact modeling. In the following equations, q_i denotes i-th component of $\mathbf{q}_f$. The position of the touchdown foot in terms of flight coordinates is:

$$x_C = q_5 + l_1\sin q_3 - l_2\sin(-q_3 - q_4) \tag{7}$$

$$y_C = q_6 - l_1\cos q_3 - l_2\cos(-q_3 - q_4) \tag{8}$$

and virtual work is denoted

$$\begin{aligned}\boldsymbol{\delta}W = \hat{F}_x.\boldsymbol{\delta}x_C + \hat{F}_y.\boldsymbol{\delta}y_C = \sum_{i=1}^{4}\left(\hat{F}_x\frac{\partial x_C}{\partial q_i} + \hat{F}_y\frac{\partial y_C}{\partial q_i}\right)\boldsymbol{\delta}q_i \\ \equiv \sum_{i=1}^{4}\hat{Q}_i\boldsymbol{\delta}q_i\end{aligned} \tag{9}$$

So, the generalized impacts become

$$\hat{Q}_i = \hat{F}_x\frac{\partial x_C}{\partial q_i} + \hat{F}_y\frac{\partial y_C}{\partial q_i} \tag{10}$$

At touchdown, the velocities change instantaneously because of an inelastic impact. The angles do not have any instantaneous change, but our coordinate notation does change due to the other leg becoming the stance leg. The stance coordinates just after contact become

$$\begin{bmatrix} q_{s1}^+ \\ q_{s2}^+ \\ q_{s3}^+ \\ q_{s4}^+ \end{bmatrix} = \begin{bmatrix} q_{f3}^- \\ q_{f4}^- \\ q_{f1}^- \\ q_{f2}^- \end{bmatrix} \tag{11}$$

During touchdown, the velocities change instantaneously, and afterwards we have

$$\begin{bmatrix} \dot{q}_{s1}^+ \\ \dot{q}_{s2}^+ \\ \dot{q}_{s3}^+ \\ \dot{q}_{s4}^+ \end{bmatrix} = \begin{bmatrix} \dot{q}_{f3}^+ \\ \dot{q}_{f4}^+ \\ \dot{q}_{f1}^+ \\ \dot{q}_{f2}^+ \end{bmatrix} \tag{12}$$

Assuming fully plastic contact, the post-contact position of the hip produced by differentiating post-contact stance coordinates is

$$\begin{bmatrix} \dot{q}_{f5}^+ \\ \dot{q}_{f6}^+ \end{bmatrix} = \begin{bmatrix} \dot{x}_h(\mathbf{q}_s^+, \dot{\mathbf{q}}_s^+) \\ \dot{y}_h(\mathbf{q}_s^+, \dot{\mathbf{q}}_s^+) \end{bmatrix} \tag{13}$$

Equations ((5), (11–13)) contain 16 equations with 16 unknowns $\dot{\mathbf{q}}_f^+, q_s^+, \dot{\mathbf{q}}_s^+, \hat{F}_x, \hat{F}_y$, which constitute the touchdown map. These equations, together with the stance and flight phase and take-off map, form the hybrid dynamic model of a running gait.

3. Gait generation

3.1. *Cost of transport*

The cost of transport (COT) defines an energy expenditure index for evaluating and comparing the generated running gaits. Assuming perfect efficiency of the motors, the energy expenditure in one step is calculated as

$$W = \sum_{i=1}^{3}\int_0^{t_{\text{step}}} |\dot{\theta}_i u_i|\,\mathrm{d}t \tag{14}$$

in which u_i denotes the motor torque of each joint, $\dot{\theta}_i$ is angular velocity of the corresponding joint in the direction of $u_{i,}$ and t_{step} defines the time interval of one step of the gait.

The COT is the consumed energy per total weight per distance traveled

$$\text{COT} = \frac{W}{m_{\text{tot}}g\sqrt{L_s^2 + h_s^2}} \tag{15}$$

in which L_s and h_s are the horizontal and vertical components of the stride, respectively.[34]

3.2. Active running gaits with constant torques during each phase

Gait generation will consist of finding a set of initial conditions and control commands that can produce a periodic orbit for running. A Poincare map of one complete running step will serve as a convenient method for describing a periodic orbit. We choose the post-contact state vector as the Poincare section, which is also the initial condition of the stance phase. Thus, the state vector at the beginning of a stance phase gives the Poincare map its input. The Poincare map outputs the next state vector at the beginning of the next stance phase (next step). Since $\boldsymbol{x}_s$ has eight components, our Poincare map state vector $\boldsymbol{x}$ is an eight-dimensional vector.

For compliant leg biped models with massless feet a *passive* solution can be found, where the torque at the hip remains zero at all times.[27] The Poincare map for a passive gait is

$$\boldsymbol{x}(k+1) = \mathbf{P}(\boldsymbol{x}(k)). \tag{16}$$

in which k is the step number. However, in robots whose feet are not massless, there is inevitable energy loss at touchdown and so the Poincare map does not have a passive fixed point. We define the active Poincare map as

$$\boldsymbol{x}(k+1) = \mathbf{P}(\boldsymbol{x}(k), \boldsymbol{u}(k)). \tag{17}$$

Vector $\boldsymbol{u}(k)$ contains the data of control effort of one complete step. Since we wish to use discrete-time control methods, we require controls to be constant over discrete-time intervals. We use one constant motor torque during each phase i.e. a constant value in the stance and a different constant value in the flight. Three motors provide torque, one in the hip and two in the knees; therefore, vector $\boldsymbol{u}$ is a six-dimensional vector, three components for the stance phase and three for the flight.

$$\boldsymbol{u} = [\boldsymbol{u}_s;\, \boldsymbol{u}_f]. \tag{18}$$

Any fixed point of the Poincare map indicates a periodic orbit of the overall dynamic model and provides a valid initial condition for an active periodic running gait. To find the fixed point of the Poincare map, find the zero of the vector function:

$$Er = \boldsymbol{x}(k+1) - \boldsymbol{x}(k) \tag{19}$$

At least one fixed point exists for any given running speed. We measure the running speed as the horizontal velocity of the center of mass at the touchdown instance, and then assume its difference from the desired velocity as a component of error function in the optimization routine. An active fixed point

$$[\boldsymbol{x}^*]_{8\times 1} = \mathbf{P}([\boldsymbol{x}^*]_{8\times 1}, [\boldsymbol{u}^*]_{6\times 1}) \tag{20}$$

of the Poincare map will help generate gaits on horizontal or sloped terrains. For a robot with multiple motors, countless fixed points exist for each running speed. We find a fixed point that minimizes energy expenditure subject to the constraint of the swing leg remaining clear of the ground. The energy expenditure index will be introduced in Section 3.3.

The nonlinear optimization needed to find zeros or minimize (19) presents some practical difficulties. Since this complicated function contains the hybrid dynamic model with continuous and discontinuous time phases, in practice the optimization algorithm will often settle in a local minimum. To solve this problem, after finding each solution we re-initialize the optimization algorithm by rounding the last result, and repeat this procedure until reaching the desired tolerance. The presence of events in the dynamic model causes challenges in choosing an integration step size; we use a relatively large time step, with maximum step size of 10 ms, for the continuous-time phases but a much smaller time step around the events, with the maximum step size in the order of 0.1 ms.

To generate an efficient running gait, springs rates should be adjusted for that speed. We found a set of values by trial and error that can produce a natural looking gait with the speed of 10 m/s: compare to the fastest human running speed of 12.4 m/s.[35] This gait has the Froude number of 13.2 which is defined as

$$Fr = \frac{V^2}{gL_{\text{leg}}} \tag{21}$$

in which, V is the running speed and L_{leg} is the leg length from hip.

As illustrations, two gaits are generated with the velocity of 10 m/s on horizontal and 5° sloped terrain. Stick diagrams of the generated gaits appear in Figure 2. Red curves indicate the trajectory of the center of mass during flight, demonstrating a considerable flight. Although unstable, these gaits can continue open-loop running several steps before falling down. The more precise the fixed point, the more open-loop running steps can occur before falling. Figure 3 shows phase diagrams of the leg BH for the gaits that diverge from their limit cycle. The needed control effort for one step of running, using constant torques in stance and flight, is shown in Figure 4. It shows the maximum torque on sloped terrain is 35% greater than on horizontal terrain. The generated running gait on level ground results in a cost of transport of 1.98 and increases with the slope of terrain (Figure 5).

Our kneed model presents a more realistic model, similar to a human leg structure. (In the next section we will show that the cost of transport can be reduced with

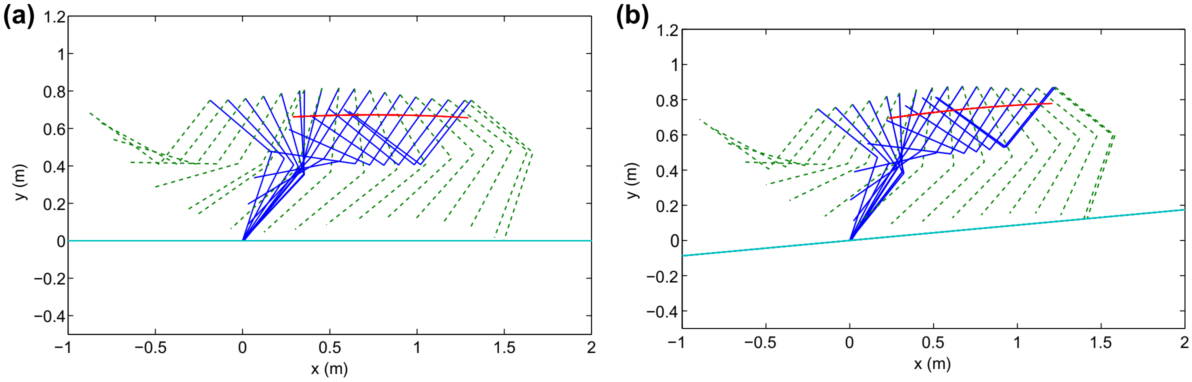

Figure 2. Stick diagram of one step of running with the velocity of 10 and 10 ms time intervals between snapshots, (a) on horizontal terrain (b) on 5^o sloped terrain.

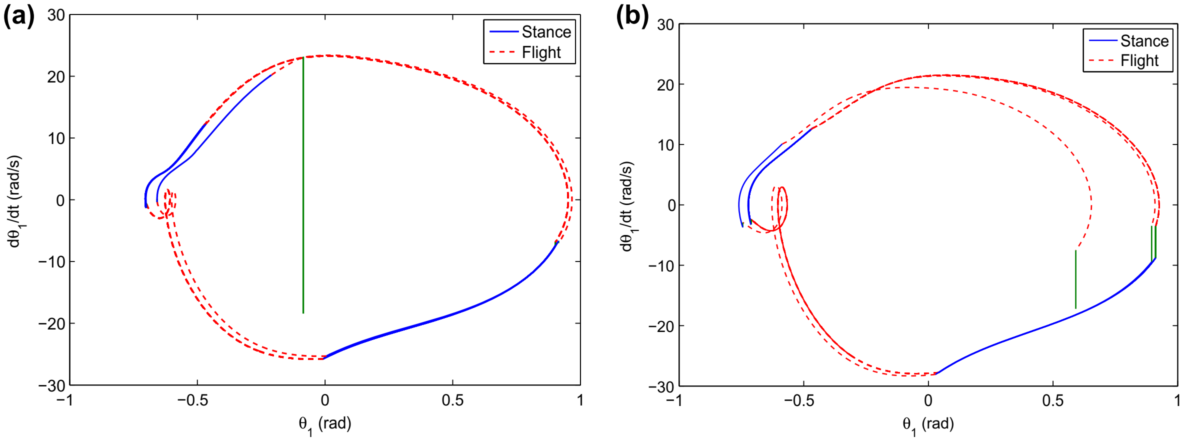

Figure 3. Phase diagram of leg BH when open-loop running starts from the fixed point (a) on horizontal terrain, (b) on 5^o sloped terrain.

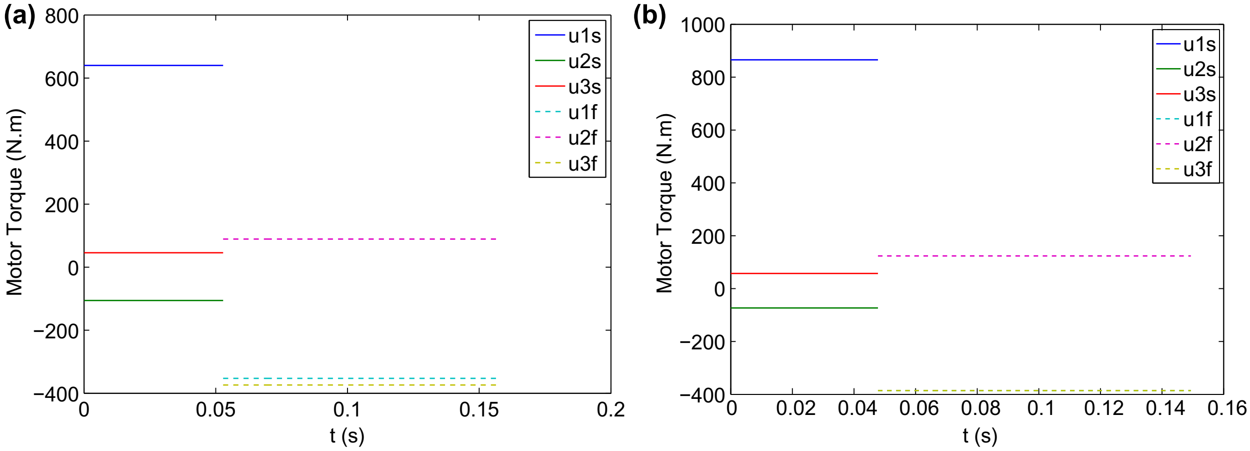

Figure 4. Control effort for one step of running with the velocity of 10 m/s (a) on horizontal terrain, (b) on 5^o sloped terrain.

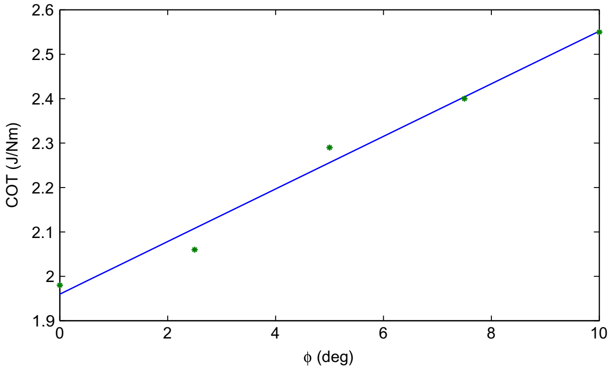

Figure 5. Cost of transport vs. terrain slope for the kneed leg robot.

variable control torques during phases.) Also note that the obtained COT points do not lie on a straight line in Figure 5; it is because the kneed biped with three motors does not have a unique gait for each condition (we find the one with minimum energy consumption).

3.3. *Active running gaits with variable torques during each phase*

The gaits investigated in the previous section had constant torques during each stance and each flight phase, reducing the volume of calculations needed yet restricting the energy efficiency of the gait. Here, the motor torques are discretized into smaller time steps, but with torques still constant during each step. Due to the variation of motor torques, the stance and flight times will vary, requiring us to choose a number of steps that covers more time than the entire phase. Looking at the stance time interval 0.056 s and the flight time interval 0.104 s from the previous section, we choose stance time step sizes of 0.02 s and flight time step size of 0.04 s, and four time steps for each phase. The control vector is defined as

$$\boldsymbol{u} = \left[u_{s1,1}, u_{s2,1}, u_{s3,1}, \ldots, u_{s1,n}, u_{s2,n}, u_{s3,n}, u_{f1,1}, u_{f2,1}, u_{f3,1}, \ldots, u_{f1,m}, u_{f2,m}, u_{f3,m}\right] \tag{22}$$

which contains all the motor torque values for both stance and flight phase of one step. In this formula, s and f indicate stance and flight phases, 3 is the number of motors of the robot, and n and m are the number of discretizations of stance and flight phase, respectively. So, the active fixed point is written as

$$[\boldsymbol{x}^*]_{8\times1} = \mathbf{P}([\boldsymbol{x}^*]_{8\times1}, [\boldsymbol{u}^*]_{3(m+n)\times1}) \tag{23}$$

To find the fixed point, an optimization problem with 8 + 3(m + n) parameters produces a minimum COT, constrained by $\boldsymbol{x}(k+1) = \boldsymbol{x}(k)$ and clearance of the swing leg. In this manner, any active biped running gait can be formulated and stabilized using this control strategy. The generated gait results in a cost of transport of 1.47 J/Nm. We repeat the optimization procedure with a stance time step of 0.01 s and a flight time step of 0.02 s, for which the COT is 1.31 J/Nm. The motor torques for one step is shown in Figure 6(b) and (c). Comparing to the results for constant torque gait (Figure 6(a)), using smaller time steps has reduced the maximum torque required. Also, COT decreases with decreasing step size in an almost linear manner (Figure 6(d)).

The generated energy efficient running gait with variable motor torques with stance and flight time step size of 0.01 and 0.02 s is shown in Figure 9(a). This gait has stance phase time interval of 0.057 and flight 0.121 s. The COT of different biped models and gaits are summarized in Table 2. Guo et al. [28] generated a biped running gait with COT of 1.01 J/Nm using a rigid model with feet, knees and torso and six motors in the ankles, knees, and hip joints. That more efficient gait may be due to more degrees of actuation or better performance of their optimization procedure. We use Matlab fmincon tool to find the fixed point and minimize the energy consumption. This optimization problem with 56 parameters requires extensive calculations and is likely to settle in a local minimum. Although it does not guarantee a global minimum, we did show our controller's applicability to stabilize other biped running gaits that have been generated by a global optimization, for example in [29].

4. Stabilizing controlers

4.1. *Linear state feedback controller*

The active open-loop gaits found in the previous section are unstable; even tiny disturbances, like those due to truncated numerical calculations in simulations, will cause the joint trajectories to drift and result in the robot falling after a few steps. To investigate gait stability, consider the linearized Poincare map around the fixed point (20) or (23) as

$$(\boldsymbol{x}(k+1) - \boldsymbol{x}^*) = \mathbf{A} \cdot (\boldsymbol{x}(k) - \boldsymbol{x}^*) + \mathbf{B} \cdot (\boldsymbol{u}(k) - \boldsymbol{u}^*) \tag{24}$$

in which $[\mathbf{A}]_{8\times8}$ and $[\mathbf{B}]_{8\times3(m+n)}$ are coefficient matrices obtained by linearization. We rewrite (24) as

$$\boldsymbol{\delta x}(k+1) = \boldsymbol{A} \cdot \boldsymbol{\delta x}(\boldsymbol{k}) - \boldsymbol{B} \cdot \boldsymbol{\delta u}(\boldsymbol{k}). \tag{25}$$

Now, instead of a hybrid nonlinear dynamic system (2–5, 11–13), we have a linear digital system (25) with an unstable equilibrium point at the origin. State feedback control

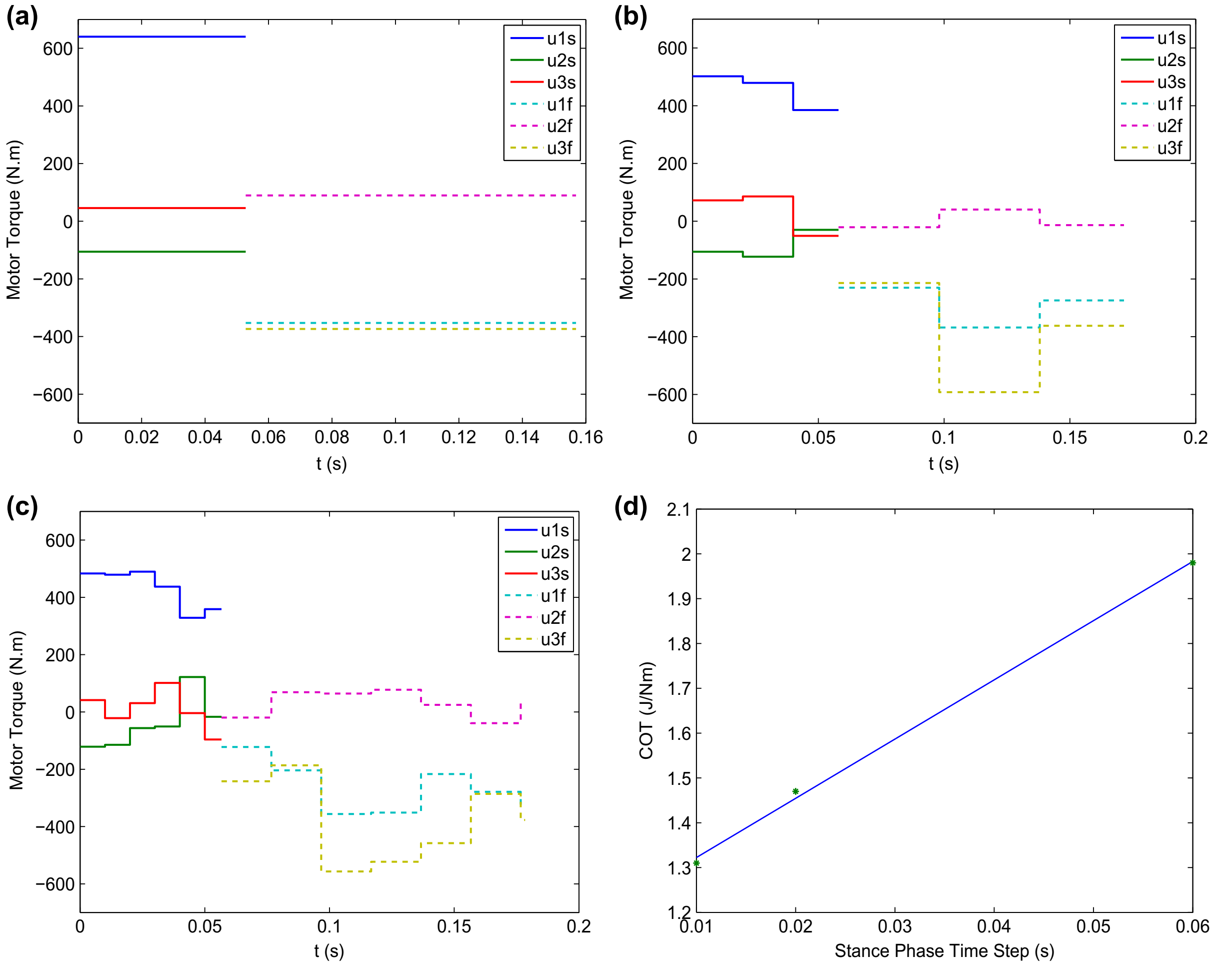

Figure 6. (a) Control effort of one step of running on horizontal terrain with constant torques, COT = 1.98, (b) variable torques with 0.02 s and 0.04 s time steps for stance and flight, COT = 1.47, and (c) variable torques with 0.01 s and 0.02 s time steps for stance and flight, COT = 1.31, (d) Cost of transport vs. time step size in variable torque gait.

$$\boldsymbol{\delta u}(k) = -\boldsymbol{K} \cdot \boldsymbol{\delta x}(k). \tag{26}$$

stabilizes system (31) if all of the eigenvalues of matrix *A-BK* are located inside the unit circle. Linear control techniques can produce an appropriate control gain matrix ***K***. In pole placement, the poles of closed-loop system ***A-BK*** are simply placed inside the unit circle. In the discrete linear quadratic regulator (DLQR) method, a matrix ***K*** is found that minimizes the cost function

$$J = \sum_k (\boldsymbol{\delta x}(k)^T \boldsymbol{Q}\boldsymbol{\delta x}(k) + \boldsymbol{\delta u}(k)^T \boldsymbol{R}\boldsymbol{\delta u}(k)). \tag{27}$$

In the DLQR method, matrices Q and R are chosen arbitrarily and they indirectly affect the closed-loop poles and controller performance. To be able to compare the state feedback controller (26) with the chaos controller introduced in the next section, the pole placement method is used to calculate matrix K in this section. The linearized Poincare map (25) for the gait on horizontal terrain has a maximum eigenvalue of 2.92 and for sloped terrain 3.63, quite suitable for application of linear state feedback controller. In the case of running on horizontal terrain, matrices $[K]_{6\times 8}$ for the constant torque and $[K]_{48\times 8}$ for the variable torque controller (with 0.01 s stance time step size) are designed to place the poles of the closed-loop system in the interval [0,0.2]. These controllers stabilize the nonlinear hybrid system and generate stable periodic running motions. When the initial condition is coincident with the fixed point of the Poincare map, the phase diagram of the closed-loop system remains on the limit cycle. Perturbing each element of the starting state by 5% from the fixed point, we observe initial deviations of the phase from the limit cycle but ultimate convergence within a few steps. Figures 7(a) and 9(b) indicate phase diagram of thigh BH for the constant and variable torque cases, respectively. In these figures, the point P is the start point just after touchdown, curve PQ corresponds to stance leg, Q to take-off,

Table 2. Cost of transport of different running biped models.

Biped robot	Horizontal terrain	5° Sloped terrain
Kneed Biped with constant torques	1.98	2.29
Kneed Biped with variable torques	1.31	–
Kneed fully actuated biped with variable torques [21]	1.01	–

QR to flight, R to touchdown of the other leg, RS to swing leg, S to take-off of the other leg, and SP to flight. The vertical green line P in one step corresponds to touchdown with instantaneous velocity changes. Figure 7(a) shows a good basin of attraction for this linear controller, since the phase diagram deviates considerably yet still converges to the limit cycle. For one complete step, the variable torque controller has 48 control parameters, instead of six parameters for the constant torque case, and so the former converges to the limit cycle much faster. It is noticeable in Figures 9(b) and 7(a) that starting from the same initial disturbances, phase diagrams converge within two and five steps, respectively. Motor torques of the two controllers for 15 and 10 steps of running are shown in Figures 7(b) and 10, labeled in accordance with Table 1. Due to the disturbance in the initial condition, the control torques have some fluctuations and converge to $\boldsymbol{u}^*$ after a few steps. We applied this controller also to stabilize a running gait with constant torques on a 5° sloped terrain. The graphs for this case (Figure 8) show more velocity discontinuities at touchdown and also greater motor torques than for horizontal terrain.

The variable torque gait has advantages of lower COT and better controllability at a cost of more calculations to find the fixed point – and more difficulty in implementation on a real robot.

4.2. *The generalized OGY chaos controller*

To evaluate the linear state feedback controller introduced in the previous section for biped running, another strategy is used to control the nonlinear hybrid system (17) with an unstable fixed point (23). The OGY control method has been proposed for chaos control of this type of system [36], but it is effective only in lower order systems with two-dimensional Poincare sections having saddle type unstable fixed points. We use the extended OGY chaos control method proposed in [30] to control chaos in higher order systems, which is based on the invariant manifold theory and sliding mode control concept.

To demonstrate the method, consider the eight-dimensional nonlinear map (17) with an unstable fixed point $\boldsymbol{x}^*$, $\boldsymbol{u}^*$, and its linearized system (24) around the fixed point. First, an invariant manifold is chosen which is independent of the system Jacobian,

$$h(k) = C \cdot \boldsymbol{\delta x}(k) = 0 \in \mathbb{R}^m \tag{28}$$

in which C is a $m \times n$ matrix, where n and m are the dimensions of the state vector $\boldsymbol{x}$ and control vector $\boldsymbol{u}$, respectively. The vector $\delta \boldsymbol{x}(k)$ lies on the intersection of hyperplanes perpendicular to each line of C. Then, the controller is designed to force the next step of the system state to lie on the selected manifold,

$$\begin{aligned} h(k+1) &= C \cdot \boldsymbol{\delta x}(k+1) \\ &= CA \cdot \boldsymbol{\delta x}(k) + CB \cdot \boldsymbol{\delta u}(k) \\ &= 0 \end{aligned} \tag{29}$$

which leads to the control law

$$\boldsymbol{\delta u}(k) = -(CB)^{-1} CA \cdot \boldsymbol{\delta x}(k). \tag{30}$$

If we consider the modeling error $\rho(k)$ due to system linearization as

$$\boldsymbol{\delta x}(k+1) = A \cdot \boldsymbol{\delta x}(k) + B \cdot \boldsymbol{\delta u}(k) + \rho(k) \tag{31}$$

then the vector

$$\begin{aligned} h(k+1) &= C \cdot \boldsymbol{\delta x}(k+1) \\ &= CA \cdot \boldsymbol{\delta x}(k) + CB \cdot \boldsymbol{\delta u}(k) + C\rho(k) \\ &= C\rho(k) \end{aligned} \tag{32}$$

is not zero using control law (30). To keep the orbit as close as possible to the invariant manifold (28), a nonlinear control law can be chosen as

$$\boldsymbol{\delta u}(k) = -(CB)^{-1} CA \cdot \boldsymbol{\delta x}(k) - (CB)^{-1} c\rho(k-1). \tag{33}$$

This leads to

$$\begin{aligned} \|h(k+1)\| &= \|CA \cdot \boldsymbol{\delta x}(k) + CB \cdot \boldsymbol{\delta u}(k) + C\rho(k)\| \\ &= \|C\rho(k) - C\rho(k-1)\| \end{aligned} \tag{34}$$

and the control law (33) is effective only if

$$\|\rho(k+1) - \rho(k)\| \leq \gamma \|\rho(k)\| \,, \quad 0 < \gamma < 1. \tag{35}$$

Now, we have two generalized OGY control laws: the linear controller (30) and the nonlinear controller (33). The problem here is how to define matrix C. Rows of this matrix define the normal vectors of hyperplanes that specify the invariant manifold around the fixed point. For the Lorenz system discussed in [30], C is a 2×1 vector that defines direction of a line passing through $\boldsymbol{x}^*$ and the controller aims to confine $\delta \boldsymbol{x}(k)$ vectors to lie on that direction. In this simple system,

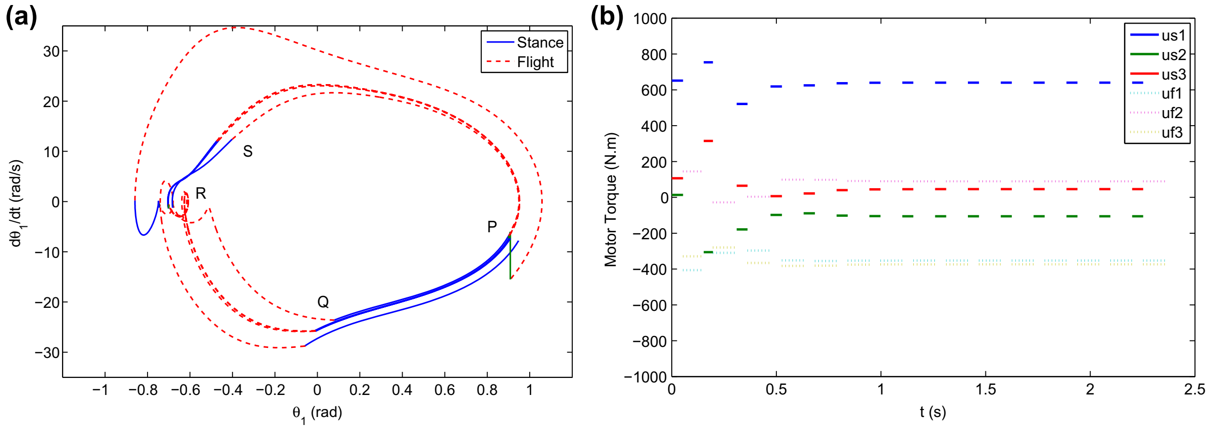

Figure 7. Results of the linear state feedback controller with constant motor torques gait on horizontal terrain (a) Phase diagram of leg BH, starting from an initial state 5% deviated from the fixed point and (b) Control effort for 15 steps of closed-loop running.

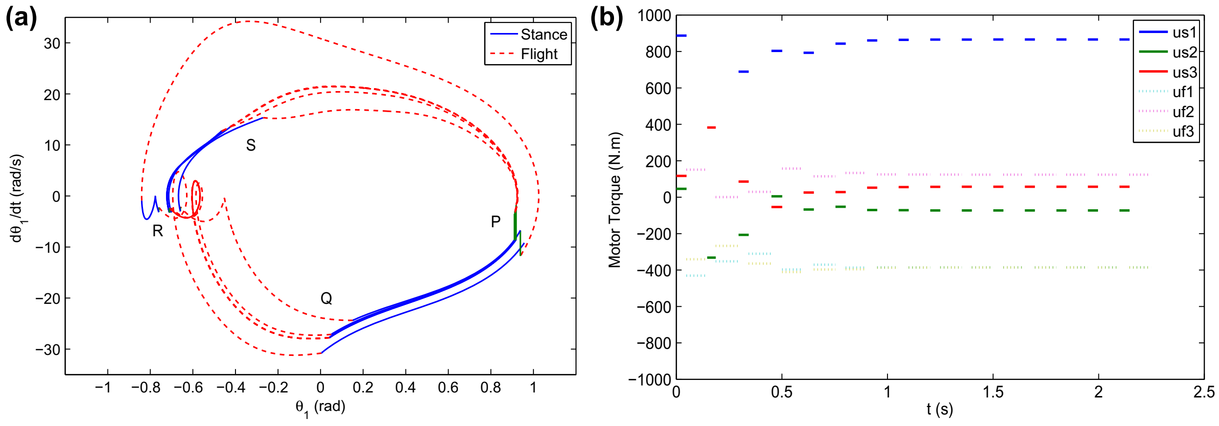

Figure 8. Results of the linear state feedback controller with constant motor torques gait on 5° sloped terrain (a) Phase diagram of leg BH, starting from an initial state 5% deviated from the fixed point and (b) Control effort for 15 steps of closed-loop running.

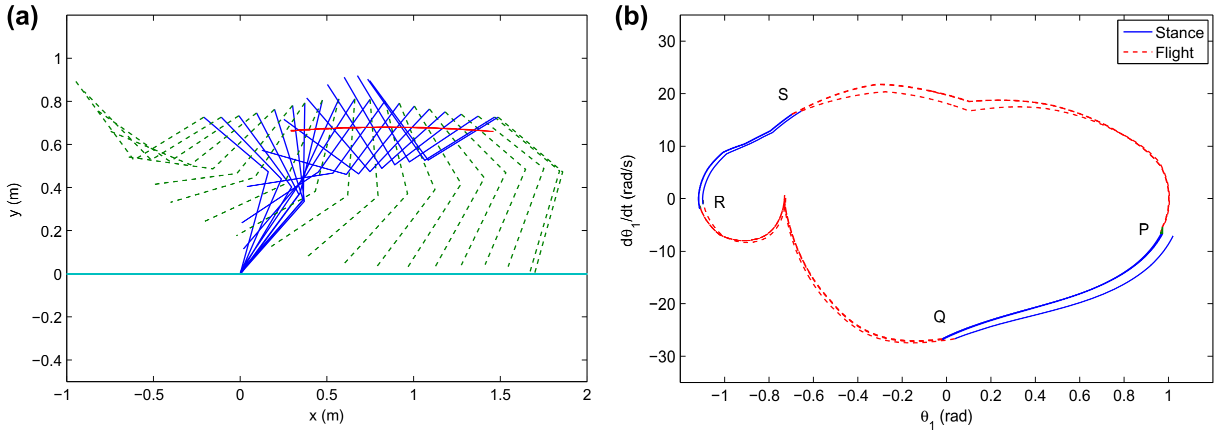

Figure 9. Results of the linear state feedback controller with variable motor torques gait on horizontal terrain (a) Stick diagram of one step of running, (b) Phase diagram of leg BH for 10 steps of closed-loop stable running, starting from a point deviated %5 from the fixed point.

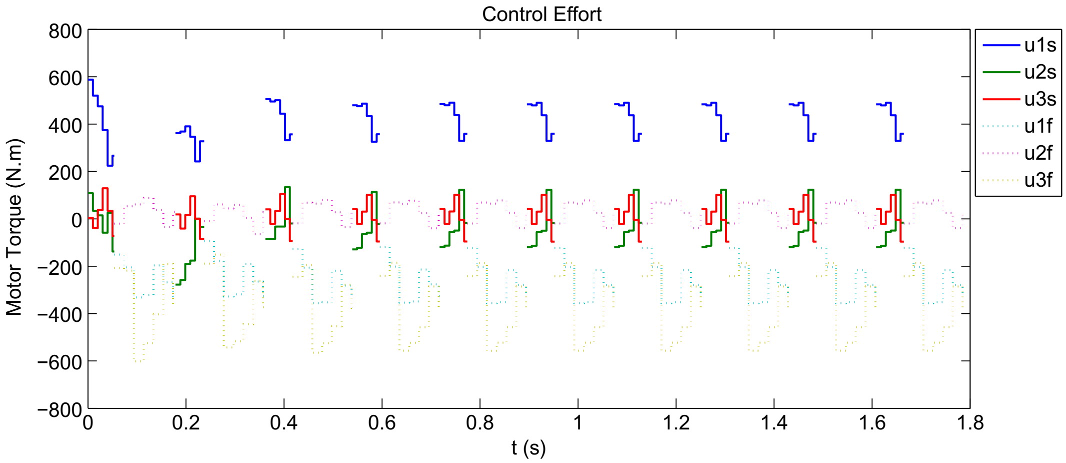

Figure 10. Control effort for 10 steps of closed-loop running on horizontal terrain.

choosing any arbitrary value for matrix C leads to a convergent closed-loop system. But for our application of the biped robot running with constant torques, C is a 6×8 matrix and an arbitrary C matrix can lead to a stable or unstable controller (for most of the cases, unstable). Using an optimization algorithm, we find a C matrix to confine the closed-loop poles of linear feedback system (30) in a circle with radius $r < 1$. Then, we use the resultant C matrix in linear controller (30) and the nonlinear controller (33). By choosing $r = 0.2$ we apply these controllers to stabilize the running gait with constant torques. Both of the controllers stabilize the system and generate stable steady running motions. The phase diagram of one leg and the control effort for these controllers, starting from an initial state 5% deviated from the fixed point, are shown in Figures 11 and 12. It is observed that the closed-loop system with the linear controller (30) settles down to its limit cycle faster than the nonlinear controller (33). By checking the necessary condition (35) of the nonlinear controller in this application we see that it is not satisfied, and so control law (35) is not more effective than (33).

To make a statistical comparison between the linear state feedback controller (26), the linear invariant manifold controller (30), and the nonlinear invariant manifold controller (33), the closed-loop poles of (26) and (30) are confined to a circle with radius $r < 1$, and it is repeated with various values of r from 0.1 to 0.9. We distribute the closed-loop poles of controller (26) evenly in the interval $[0, r]$, but for the controller (30) we confine just the radius of max pole and the poles are distributed automatically. Then, these three controllers are used to control the running gait with constant torques, and the number of running steps before settling to the limit cycle is counted if it is convergent. The settling criterion is chosen as

$$\|\boldsymbol{\delta x}(k)\| = \|\boldsymbol{x}(k) - \boldsymbol{x}^*\| \leq 0.01 \tag{36}$$

The simulations are done once with 5% deviation of the initial condition from the fixed point and once with 10%. The results are summarized in Table 3. The cases that are divergent are outside of the basin of attraction. It is seen that the linear invariant manifold controller has the best performance. It converges faster to the limit cycle and also it has larger basin of attraction. The second best controller is the state feedback controller. The nonlinear invariant manifold controller has the worst performance which is due to violation of the condition (35) in this application.

In another variation of the OGY control method, one can look at [37] which uses the influence of small changes of each control parameter on the chosen fixed point of the Poincare map, thus providing the Jacobian matrix $\partial u/\partial x^*$ numerically. However, attempts to utilize this control strategy for bipedal running resulted in instability. Likely this is due to our Poincare map having a non-unique fixed point, in its eight-dimensional state vector, for a defined running speed – making the Jacobian matrix unsuitable for control.

The OGY method and its extensions are kind of state feedback control method in which all states of the system are observed and used in the controller. Pyragas proposed time-delayed feedback control (TDFC) for stabilization of periodic orbits in continuous time systems by using output signal.[38] This control method uses the difference between the current output signal and the τ-time delayed output signal, in which τ is a period of the stabilized periodic orbit. This controller is a simple and powerful method and does not need an exact model of the system, but its stability analysis is very difficult and the feedback gain is usually determined using experimental adjustments. Ushio investigated TDFC in discrete-time systems (25) with an unstable fixed point and proved a limitation

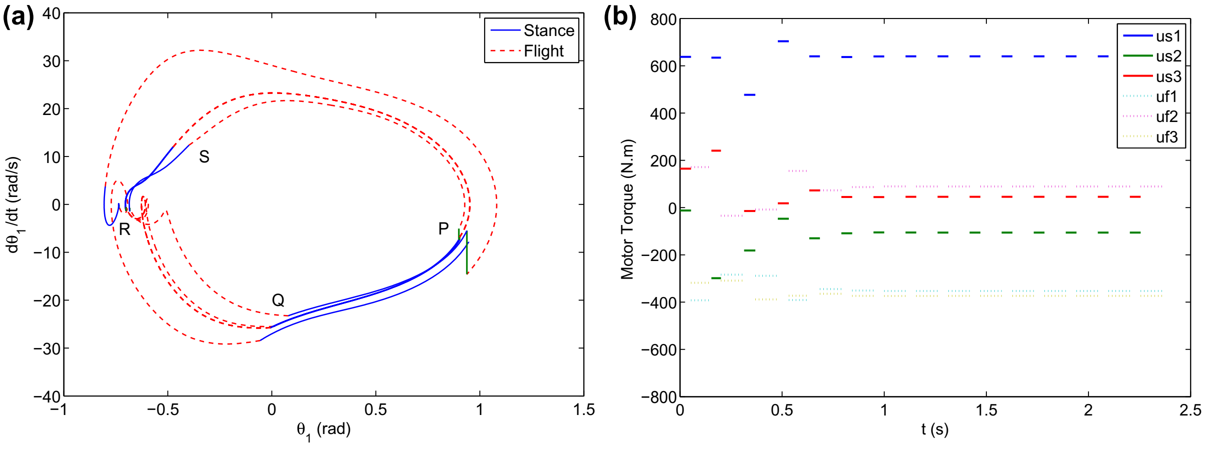

Figure 11. Results of the linear invariant manifold controller (a) Phase diagram of leg BH, starting from an initial state 5% deviated from the fixed point and (b) Control effort for 15 steps of closed-loop running on horizontal terrain.

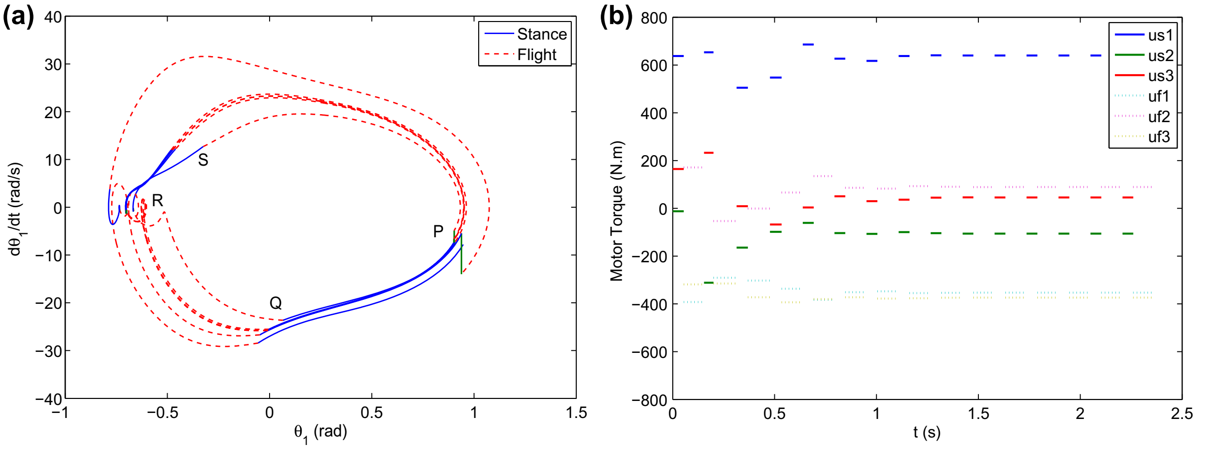

Figure 12. Results of the nonlinear invariant manifold controller (a) Phase diagram of leg BH, starting from an initial state 5% deviated from the fixed point and (b) Control effort for 15 steps of closed-loop running on horizontal terrain.

Table 3. Number of running steps before settling to the limit cycle, using the three controllers.

Maximum closed-loop pole	Linear state feedback controller with 5% deviation of initial state	Linear chaos controller with 5% deviation of initial state	Nonlinear chaos controller with 5% deviation of initial state	Linear state feedback controller with 10% deviation of initial state	Linear chaos controller with 10% deviation of initial state	Nonlinear chaos controller with 10% deviation of initial state
0	7	5	10	Divergent	Divergent	Divergent
0.1	6	6	12	Divergent	Divergent	Divergent
0.2	6	6	Divergent	8	6	Divergent
0.3	7	6	10	7	8	Divergent
0.4	7	6	9	8	7	Divergent
0.5	7	6	9	11	7	15
0.6	9	6	10	11	8	Divergent
0.7	11	11	12	Divergent	12	Divergent
0.8	Divergent	23	18	Divergent	Divergent	Divergent
0.9	Divergent	Divergent	37	Divergent	Divergent	Divergent

for this controller. He showed that a fixed point cannot be stabilized by TDFC if the linearized system around the fixed point has an odd number of real eigenvalues greater than one.[39] We checked the eigenvalues of our linearized systems for the generated running gaits and observed that some of them have an even number and some an odd number of real eigenvalues greater than one. So, this method cannot be used to stabilize all of these gaits. Furthermore, designing the feedback gain matrix for this controller is a challenge.

5. Conclusion

The dynamic equations of biped robot running were formulated and a numerical framework proposed to derive and solve the equations for a biped model. It was shown that the proposed compliant kneed multi-body biped model can produce active running limit cycles with either constant or variable motor torques during each phase. This model has a variable stiffness – changing between flight and stance – for greater energy efficiency and natural running gaits. A fixed point of a single-step Poincare map provided the basis to generate periodic running gaits. After generating running gaits with constant motor torques during each (stance and flight) phase, the motor torques were discretized using smaller intervals to show generality of the control strategy. Using more discretizations requires significantly more calculations to find the fixed point, but the motors consume less energy.

A discrete-time linear state feedback and a generalized OGY controller stabilized the linearized Poincare map around the fixed point and also the main system around its limit cycle. Simulations demonstrated the controllers can stabilize running on both horizontal and inclined surfaces. Our controllers are not limited to passive gaits like in [26] and [27] and for active gaits our controller is much simpler than HZD method [10]. The fixed point formulation in Section 3.3 and its controllers in Section 4 should be able to stabilize any active biped running gait with variable discretized motor torques. The generalized OGY controller has two forms: linear and nonlinear. The three controllers were designed with various values of maximum closed-loop pole and were applied to control biped running gait with constant torques. The initial condition was deviated both 5% and 10% from the fixed point, and the number of running steps before converging to the fixed point was counted. The results summarized in Table 3 show that the linear generalized OGY controller had the best performance in terms of the basin of attraction and settling time and the linear state feedback controller was second.

Notes on contributors

Behnam Dadashzadeh received his BSc from the University of Tabriz in 2005 and his MSc and PhD from the University of Tehran in 2007 and 2013, all in Mechanical Engineering. He spent 7 months at the University of Calgary and 6 months at Dynamic Robotic Laboratory, Oregon State University working on running robots as a visiting researcher during 2012/2013. His research interests include bipedal running control, mechatronics and mobile robots.

M. J. Mahjoob (Mohammad Mahjoob Jahromi) received his BSc/MSc from University of Tehran in 1988 and the PhD from University of Waterloo Canada in 1995, all in Mechanical Engineering. Since 1995, he has been with the School of Mechanical Engineering, University of Tehran, where he is now associate professor and director of Center for Mechatronics and Intelligent Machines. His field of interest includes mechatronics, systems dynamics and control.

M. Nikkhah Bahrami received his BSc and MSc in Civil Engineering from the University of Texas in 1963 and 1965. He received his PhD in Engineering Mechanics and Computational Mechanics from the University of Texas. He is a professor of Mechanical Engineering at the University of Tehran where he has been a faculty member since 1969. He has published more than ten books in Dynamics, Vibration, Numerical Computing and Computer Programming. His research interests include Multibody Dynamics and Computational Mechanics, Wavelet Analysis and its Applications, Numerical Methods in Engineering, Mechanical Vibration and Modal Analysis.

Chris Macnab received his BEng in Engineering Physics from the Royal Military College of Canada in 1993. He received a PhD from the University of Toronto in 1999, where he attended the Institute for Aerospace Studies and investigated stable neural-adaptive control of flexible-joint robots. He worked at Dynacon Systems and at CRS Robotics (now Thermo CRS Ltd.) in Toronto. He is an Associate Professor at the Department of Electrical and Computer Engineering, University of Calgary where his current research interests include adaptive, fuzzy, and neural-network control applied to flexible-joint robots, helicopters, haptic teleoperation, and biped running robots.

References

[1] Vukobratović M, Borovac B, Surla D, Stokić D. Biped locomotion: dynamics, stability, control and application. New York (NY): Springer-Verlag; 1990.

[2] You Z, Zhang Z. An overview of the underactuated biped robots. Proceeding of the IEEE International Conference on Information and Automation Shenzhen; China; 2011 Jun.

[3] Raibert M-H. Legged robots that balance. Cambridge, MA: Massachusetts Institute of Technology; 1986.

[4] McGeer T. Dynamics and control of bipedal locomotion. J. Theor. Biol. 1993 Aug;163:277–314.

[5] Chevallereau C, Abba G, Aoustin Y, Plestan F, Westervelt ER, Canudas-de-Wit C, Grizzle JW. RABBIT: a test bed for advanced control theory. IEEE Control Syst. Mag. 2003;8:1–51.

[6] Park H-W, Sreenath K, Hurst JW, Grizzle JW. Identification of a bipedal robot with a compliant drivetrain: parameter estimation for control design. Control Syst. Mag. 2011 Apr;31:63–88.

[7] Grimes JA, Hurst JW. The design of ATRIAS 1.0 a unique monopod, hopping robot. In: International Conference on Climbing and Walking Robots (CLAWAR); 2012 Jul; USA.

[8] Goswami A, Thuilot B, Espiau B. A study of the passive gait of a compass-like biped robot: symmetry and chaos. Int. J. Rob. Res. 1998;17:1282–1301.

[9] Spong MW, Holm JK, Lee D. Passivity-based control of bipedal locomotion. IEEE Rob. Autom Mag. 2007;14:30–40.

[10] Chevallereau C, Westervelt ER, Grizzle JW. Asymptotically stable running for a five-link, four-actuator, planar bipedal robot. Int. J. Rob. Res. 2005;24:431–464.

[11] Ahmadi M, Buehler M. Stable control of a simulated one-legged running robot with hip and leg compliance. IEEE Trans. Rob. Autom. 1997 Feb;13:96–104.

[12] Blickhan R. The spring-mass model for running and hopping. J. Biomech. 1989;22:1217–1227.

[13] Geyer H, Seyfarth A, Blickhan R. Compliant leg behaviour explains basic dynamics of walking and running. Proceedings Proc. R. Soc. London, Ser. B; 2006;273:2861–2867.

[14] Rummel J, Iida F, Seyfarth A. One-legged locomotion with a compliant passive joint. In: Arai T, editor. Proceedings of the 9th International Conference on Intelligent Autonomous Systems; Tokyo: IOS Press; 2006, 566–573.

[15] Iida F, Rummel J, Seyfarth A. Bipedal walking and running with spring-like biarticular muscles. J Biomech. 2008 Jan;41:656–667.

[16] Dean JC, Kuo D. Elastic coupling of limb joints enables faster bipedal walking. J. R. Soc. 2009;6:561–573.

[17] Verrelst B, Vanderborght B, Vermeulen J, Ham R, Naudet J, Lefeber D. Control architecture for the pneumatically actuated dynamic walking biped “Lucy”. Mechatronics. 2005 Jul;15:703–729.

[18] Hosoda K, Takuma T. Biped robot design powered by antagonistic pneumatic actuators for multi-modal locomotion. Rob. Autom. Sys. 2008;56:46–53.

[19] Pratt J, Krupp B. Design of a bipedal walking robot. Proceedings of SPIE, Vol. 6962; 2008. DOI: 10.1117/12.777973.

[20] Radkhah K, Lens T, Seyfarth A, Stryk O. On the influence of elastic actuation and monoarticular structures in biologically inspired bipedal robots. In: 3rd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics; 2010 Sep; Japan.

[21] Hurmuzlu Y, Génot F, Brogliato B. Modeling, stability and control of biped robots – a general framework. Automatica. 2004;40:1647–1664.

[22] Kuo AD. Stabilization of lateral motion in passive dynamic walking. Int. J. Rob. Res. 1999;18:917–930.

[23] Piiroinen P, Dankowicz H. Low-cost control of repetitive gait in passive bipedal walkers. Int. J. Bifurcation Chaos. 2005;15:1959–1973.

[24] Hurmuzlu Y. Dynamics of bipedal gait: part I – objective functions and the contact event of a planar five-link biped. J. Appl. Mech. 1993;60:331–344.

[25] Cho B-K, Oh J-H. Running pattern generation with a fixed point in a 2D planar biped. Int. J. Humanoid Rob. 2009;6:241–264.

[26] McGeer T. Passive Bipedal running. Proc. R. Soc. London, Ser. B. 1990;240:107–134.

[27] Hu Y, Yan G, Lin Z. Stable running of a planar underactuated biped robot. Robotica. 2011;29:657–665.

[28] Guo Q, Macnab CJB, Pieper JK. Generating efficient rigid biped running gaits with calculated take-off velocities. Robotica. 2010;29:627–640.

[29] Hasaneini SJ, Macnab CJB, Bertram JEA, Leung H. The dynamic optimization approach to locomotion dynamics: human-like gaits from a minimally-constrained biped model. Adv. Rob. 2013;27:845–859.

[30] Xinghuo Yu, Chen G, Xia Yang, Song Yanxing, Cao Zhenwei. An invariant-manifold-based method for chaos control. IEEE Trans. Circuits Syst. – I: Fundam. Theory and Appl. 2001 Aug;48:930–937.

[31] Lipfert SW. Kinematic and dynamic similarities between walking and running. Hamburg: Verlag Dr. Kovac. ISBN-10: 3830050305; 2010.

[32] Niqq BM, Liu W. The effect of muscle stiffness and damping on simulated impact force peaks during running. J. Biomech. 1999 Aug;32:849–856.

[33] Westervelt ER, Grizzle JW, Chevallereau C, Choi JH, Morris B. Feedback control of dynamic bipedal robot locomotion. Boca Raton: CRC Press; 2007.

[34] Collins S, Ruina A, Tedrake R, Wisse M. Efficient bipedal robots based on passive-dynamic walkers. Science. 2005;307:1082–1085.

[35] http://joggersunited.wordpress.com/2012/02/23/40-random-facts-about-running-and-runners/

[36] Ott E, Grebogi C, Yorke JA. Controlling chaos. Phys. Rev. Lett. 1990;64:1196–1199.

[37] Boukabou A, Mansouri N. Controlling chaos in higher order dynamical systems. Int. J. Bifurcation Chaos. 2004;14:4019–4025.

[38] Pyragas K. Continuous control of chaos by self-controlling feedback. Phys. Lett. A. 1992;170:421–428.

[39] Ushio T. Limitation of delayed feedback control in nonlinear discrete-time systems. IEEE Trans. Circuits Syst. – I Fundam. Theory Appl.1996 Sep:43:815–816.