First Steps in Passive Dynamic Walking

Chapter · January 2005

DOI: 10.1007/3-540-29461-9_73

Arend L Schwab
Delft University of Technology

Pablo Gonzalez-de-Santos
Centre for Automation and Robotics

First steps in Passive Dynamic Walking

Martijn Wisse[1] and Arend L. Schwab[2]

[1] Delft Biorobotics Laboratory
Delft University of Technology
The Netherlands
m.wisse@wbmt.tudelft.nl
[2] Laboratory for Engineering Mechanics
Delft University of Technology
The Netherlands
a.l.schwab@wbmt.tudelft.nl

Summary. Passive dynamic walking is a promising concept for the design of efficient, natural two-legged walking robots. Research on this topic requires an initial point of departure; a stability analysis can be executed only after the first successful walking motion has been found. Experience indicates that it is difficult to find this first successful walking motion. Therefore, this paper provides the basic tools to simulate a simple, two-dimensional walking model, to find its natural cyclic motion, to analyze the stability, and to investigate the effect of parameter changes on the walking motion and the stability. Especially in conjunction with the accompanying MATLAB[3] files, this paper can serve as a quick and effective start with the concept of passive dynamic walking.

1 Introduction

This text is written for prospective researchers of 'Passive Dynamic Walking'. Passive Dynamic Walking is an approach to investigate bipedal (two-legged) walking systems, be it humans or other bipedal animals, or bipedal walking robots that you want to build or control. Passive Dynamic Walking is a way to look at bipedal walking. Instead of seeing it as a continuous struggle to keep balance, bipedal walking is much better understood when regarding it as a continuous passive fall, only intermittently interrupted by a change of foot contact. A steady succession of steps can then be analyzed as a cyclic motion.

The approach of Passive Dynamic Walking as originally proposed by McGeer [4] has led to various insights regarding human walking [3], and has produced a number of natural and efficient walking machines [1, 4, 6], see

[3] For availability of the accompanying files, please try http://dbl.tudelft.nl or contact the first author.

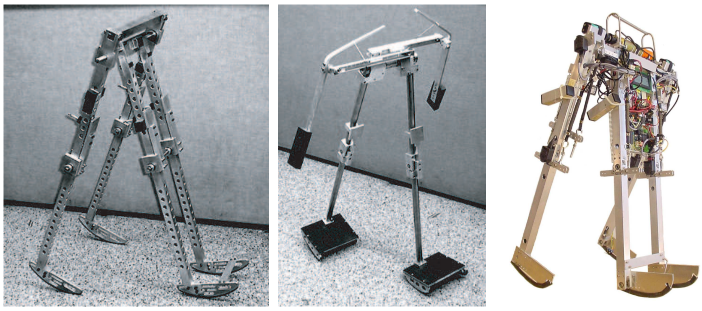

Fig. 1. Prototypes of passive dynamic walking bipeds that have been developed over the years. Left: Copy of Dynamite, McGeer [4], middle: 3D walker, Collins *et al.* [1], right: Mike, Wisse and Frankenhuyzen [6].

Figure 1. It is our opinion that huge progress can be made in both fields using the approach of Passive Dynamic Walking. However, experience indicates that it is rather difficult to get started with Passive Dynamic Walking, as one can start analysis only after at least one successful walking motion has been found. This text serves as a guide to that first start.

We will present the complete simulation procedure for a simple, two-dimensional passive dynamic walker. The model is realistic enough to enable the construction of a physical prototype with corresponding behavior. Section II describes the required algorithms for a computer simulation that will predict a walking motion after a proper launch of the biped. This section includes the model description, the derivation of the equations of motion, numerical integration, heel strike detection and the derivation of the impact equations. Section III focuses on the analysis of the step-to-step progression of disturbances on the walking motion, encompassing the selection of a Poincaré section and a linearized stability analysis.

The text is accompanied by a set of MATLAB (version 5.2 or higher) files that will provide an operational programming example for a quick start. The following sections will guide you through the functions and background of each of the files.

2 Forward dynamic simulation

Model

The simplest system that can perform a Passive Dynamic Walking motion consists of two rigid legs interconnected through a passive hinge. We will study a two-dimensional model for the sake of simplicity. A real-world prototype can be made to behave (more-or-less) two-dimensional through the construction of two symmetric pairs of legs, see Figure 2. The corresponding dynamic model is shown in Figure 2.

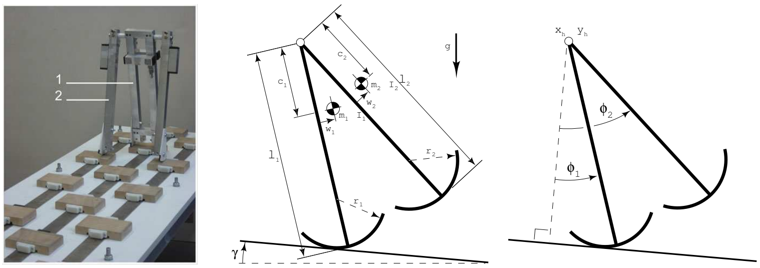

Fig. 2. Left: Prototype Passive Dynamic Walking robot with four legs (two-dimensional walking behavior), walking on a checkerboard surface to prevent foot-scuffing at mid-stance. Middle: Parameters of the simulation model. Right: four degrees of freedom of the simulation model; the position of the hip and the absolute leg angles.

We will make a number of assumptions to keep the simulation manageable. First, we assume that the legs suffer no flexible deformation and that the hip joint is free of damping or friction. Second, we idealize the contact between the foot and the floor, assuming perfectly circular feet that do not deform or slip, while the heel strike impact is modeled as an instantaneous, fully inelastic impact where no slip and no bounce occurs. Finally, the floor is assumed to be a rigid and flat slope with a small downhill angle.

There is one problem due to oversimplification of the model. Contrary to humans who have knees, the legs of the model cannot extend or retract, which inevitably leads to foot-scuffing at mid-stance. In a real-world prototype this problem is solved by covering the floor with a checkerboard pattern of tiles that provide foot clearance for the swing foot, see Figure 2. In the computer simulation, we will simply assume that there is no interference between the floor and the swing foot under certain conditions, as described in the section 'Heel strike detection'.

Based on these assumptions, the model is defined with 14 parameters, which is done in the file `wse_par.m`. The world is parameterized with gravity g and slope angle γ. A leg must be parameterized as a single rigid body with a mass m, a moment of inertia I, the coordinates for its center of mass with respect to the hip in vertical direction c and in horizontal direction w, the leg length l and the foot radius r. An idealized model consists of two completely equal legs. However, we have noticed that a small difference in parameter values between the legs can strongly influence the walking behavior, so the model will be prepared for legs with different parameter values. All parameters are summarized in Table 1 in which we have also provided a set of default parameter values that should lead to a successfully walking model or prototype.

The number of degrees of freedom of this model requires some attention; although the two legs each have two position and one orientation coordinate in a two-dimensional world resulting in a total of six degrees of freedom (twelve

World		
gravity	g	9.81 m/s^2
slope angle	γ	0.01 rad
Leg 1 and 2		
length	l	0.4 m
foot radius	r	0.1 m
CoM location	c	0.1 m
	w	0 m
mass	m	1 kg
mom. of inertia	I	0.01 kgm^2

Table 1. Parameters for a simple passive dynamic walking model corresponding to Figure 2. The given default parameter values were chosen to 1) comply with a realistic prototype, and 2) provide stable simulation results.

states when including the velocities), only three states are independent at the start of a step. We get from twelve to three by successively considering the hip joint constraint, the foot contact, and the Poincaré section. First, the hip joint constrains two degrees of freedom (four states) so that the model has only four independent generalized coordinates, x_h, y_h, ϕ_1 and ϕ_2, see Figure 2. Second, the foot contact constrains two more degrees of freedom (again four states), leaving only ϕ_1 and ϕ_2 as independent coordinates. The hip coordinates depend alternatingly on the one or the other foot contact, which is calculated in the file wse_dep.m. Third, we take a Poincaré Section of the cyclic walking motion. This means that we will focus our attention on the start of each step defined as the instant just after heel strike when both feet are in contact with the floor, which makes one more state dependent; only one leg angle is independent, the other is the same but opposite in sign. Together with the two independent velocities, there are three independent initial conditions that completely define the state at the start of a step, see Table 2. The definition of the initial conditions takes place in the file wse_ic.m. The values in Table 2 together with the default parameter values in Table 1 will result in a cyclic and stable walking motion.

Next to defining initial conditions for the model coordinates, we also need to define the foot contact coordinates. The actual foot contact point travels forward as the model 'rolls' forward over the sole of its circular feet. Therefore we appoint a single, fixed location as foot contact coordinate for the entire duration of a step. This location is defined as the actual point of contact if the leg angle is zero. The piecewise non-holonomic nature of walking systems requires that the foot contact coordinates are re-evaluated after each step. The initial values for the foot contact locations are set rather arbitrarily to zero in Table 2.

Independent initial conditions		
Stance leg (leg 1) angle	ϕ_1	0.2015 rad
Stance leg (leg 1) angular velocity	$\dot{\phi}_1$	-1.4052 rad/s
Swing leg (leg 2) angular velocity	$\dot{\phi}_2$	-1.1205 rad/s
Dependent initial conditions (`wse_dep.m`)		
Swing leg (leg 2) angle	ϕ_2	-0.2015 rad
Hip horizontal displacement	x_h	0.0802 m
Hip vertical displacement	y_h	0.3939 m
Hip horizontal velocity	$\dot{x}_h$	0.5535 m/s
Hip vertical velocity	$\dot{y}_h$	0.0844 m/s
Initial foot contact coordinates		
Foothold location stance leg (leg 1)	xf_1	0 m
Foothold location swing leg (leg 2)	xf_2	0 m

Table 2. Initial conditions for a simple passive dynamic walking model corresponding to Figure 2. Leg 1 is chosen as the initial stance leg. The given default values will, in combination with the default parameter values in Table 1, result in a stable cyclic walking motion.

Derivation of equations of motion

The equations of motion are the heart of the computer simulation. For our model we will first derive the generalized equations of motion for the two legs plus hip joint, then derive the algebraic equations that describe the alternating foot contact, and finally put these together in a system of DAE's - Differential Algebraic Equations. The equations in this section correspond to the file wse_eom.m.

Let's first consider the system of legs and hip without foot contact. As explained above, that system has four independent *generalized* coordinates $\mathbf{q}$. Their accelerations are calculated with the set of equations

$$\overline{\mathbf{M}}\ddot{\mathbf{q}} = \overline{\mathbf{f}} \tag{1}$$

with the generalized mass matrix $\overline{\mathbf{M}}$ and the generalized force vector $\overline{\mathbf{f}}$. They are constructed with the principle of virtual power and d'Alembert inertia forces (the so-called TMT-method) resulting in

$$\overline{\mathbf{M}} = \mathbf{T}^T\mathbf{M}\mathbf{T}, \qquad \overline{\mathbf{f}} = \mathbf{T}^T[\mathbf{f_g} - \mathbf{M}\mathbf{h}]. \tag{2}$$

In this $\mathbf{M}$ and $\mathbf{f_g}$ are the terms from 'normal' Newton-Euler equations of motion, i.e. without hip joint constraints and thus for six coordinates. The matrix T transfers the independent generalized coordinates $\dot{\mathbf{q}}$ into the velocities of the center of mass of the bodies $\dot{\mathbf{x}}$. The vector $\mathbf{h}$ holds the convective accelerations. $\mathbf{T}$ and $\mathbf{h}$ are generated by running wse_sde.m once, which creates the file wse_mat.m containing all necessary matrices.

With equation (2) we can calculate the accelerations for the two legs while ensuring that they remain connected at the hip. However, the system is in free fall like this as we have not yet incorporated the contact between the feet and the floor. This contact is described with two equations per leg. First, the foot should be at floor level. Since we apply circular feet, the vertical constraint equations becomes

$$g_y = y_h - (l - r) * cos(\phi) - r \tag{3}$$

where g_y must be zero to fulfill the constraint. Second, the horizontal displacement of the foot must be related to the leg angle plus some initial offset (x_f) depending on where the foot has landed,

$$g_x = x_h + (l - r) * sin(\phi) + r * \phi - x_f \tag{4}$$

where g_x must be zero to prescribe pure rolling without slip.

To construct the complete set of DAE's we must first determine which foot is in contact, as only one set of foot contact constraints is active at a time. We will need the second derivative of these constraint equations (in the form of **D** and **D2**) to allow a combination with the equations of motion in the total system of equations

$$\begin{bmatrix} \overline{M} & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ F_c \end{bmatrix} = \begin{bmatrix} \overline{F} \\ D2^T \dot{q}^2 \end{bmatrix} \tag{5}$$

Solving these equations at any instant will provide the generalized accelerations $\ddot{\mathbf{q}}$ and the foot contact forces $\mathbf{f_c}$ at that instant.

Numerical integration

The next step is to go from accelerations at any instant to a continuous motion. To obtain that motion numerical integration is needed, which is done in the file wse_rk4.m. We use the classical Runge-Kutta 4 method, which calculates in four intermediate steps the positions and velocities at time $t + \Delta t$.

One of the problems of numerical integration is the accumulation of numerical errors. The overall error can be checked by inspection of the energy content of the system, as the sum of kinetic and potential energy should be constant for a passive walker. An example of such energy checks is given in the file wse_ech.m. Otherwise, one could check stride characteristics such as stride time or stride length, and investigate how much these change by halving the integration step Δt.

Next to the overall error, there is the problem of non-satisfied constraint conditions. The accumulating numerical errors easily lead to a foot that sinks into the ground or flies away. The source of this type of errors is the fact that in equation (5) there are only second derivatives of the constraint equations, which only impose that the *acceleration* of the foot is zero. A small round-off error leads to huge position displacements after a while. Therefore, the file wse_rk4.m frequently calls the file wse_dep.m which recalculates the hip coordinates and velocities as a function of the independent leg angles and angular velocities so that the foot constraints are met.

Heel strike detection

During normal walking some events take place every step, whereas in the case of a fall a few other events could take place. To start with the latter, falling forward, falling backward, and losing ground contact (too high velocity) are three possible events. At every step, the file wse_evd.m checks for each of these terminal events and reacts by stopping the simulation.

During continuous locomotion, at every step a heel strike impact occurs, followed by a change of stance foot. This event is detected by monitoring the clearance of the swing foot (equation 3). Zero clearance means that either a genuine heel strike has occurred or that the swing leg has briefly reached floor level during mid-stance. To distinguish between the two, the file wse_evd.m contains a four-level decision tree;

`IF`

- the vertical distance between the swing foot and the floor has changed sign, `AND`
- the stance leg has passed mid-stance (i.e. its direction of motion is away from the vertical position), `AND`
- the swing foot is currently below floor level, `AND`
- the legs are not parallel but in a spread configuration

`THEN` there must have been a valid heel-strike somewhere between the previous and the current integration time step.

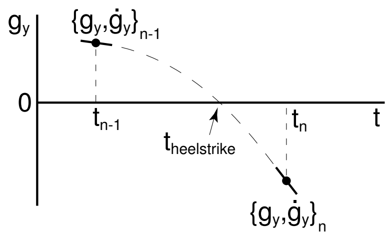

Fig. 3. Interpolation with third order polynomial to find the instant of heel strike between to integration steps. The clearance function g_y is given by equation 3.

If this is detected, an interpolation is necessary to determine the exact instant of heel contact. We approximate the motion between the timesteps t_{n-1} and t_n with a third order polynomial and determine when this polynomial passes through zero, see Figure 3. This is done in wse_int.m. After this operation we know the precise instant of heel contact and the state of the model at that instant.

Derivation of impact equations

We assume that heel strike is a fully inelastic impact between the forward foot and the floor. During the heel-strike impact there are very high forces for a very short time. This process can be interpreted as an impulsive motion, an instantaneous event in which the velocities change but not the positions of the model elements. To calculate this we can use the same equations of motion as eq. (5) if we apply an integration over the impact duration and take the limit of this duration to zero. The result is a system of impact equations with much resemblance to eq. (5):

$$\begin{bmatrix} \overline{M} & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ f_c \end{bmatrix} = \begin{bmatrix} \overline{M}\dot{q}^- \\ 0 \end{bmatrix} \tag{6}$$

The D matrix again represents the foot constraints, and is equal to the D used in equation 5. We must carefully decide which foot constraints are active during heel strike and which are not. At heel strike, both feet are at floor level, so both could possibly participate in the impact. However, the contacts are unilateral which means that only compressive forces can occur. We should only incorporate those constraint equations that result in a compressive impulse. For our model during normal walking, it turns out that only the forward foot does participate whereas the hind foot does not. Presumably the hind foot will obtain an upward velocity as a result from the impact calculation. If it doesn't, the assumption was wrong and we should have incorporated *both* feet in equation 6, which would have resulted in a full stop. The file wse_evd.m checks for this.

Walking cycle

Now we have sufficient tools and algorithms to simulate a continuous walking motion. Let's give the model some initial conditions and see how many steps it will take or how it might fall. The file wse_scw.m ties all previously mentioned files together. Use it by first setting the appropriate parameter values and initial conditions in wse_par.m and wse_ic.m and then running wse_scw.m. The simulation results are then stored in the large matrices `t_t`, `q_t`, `qd_t`, `f_t`, and `g_t`, accessible from MATLAB's base workspace as global variables. To visualize the results, one can use and modify wse_fig.m which plots some basic graphs (Figure 4), or wse_ani.m which displays a simple animation of the resulting motion, see Figure 5.

3 Step-to-step stability analysis

Stability

The most important characteristic of a walking machine is its stability; it should not fall down. According to the classical interpretation, this requires

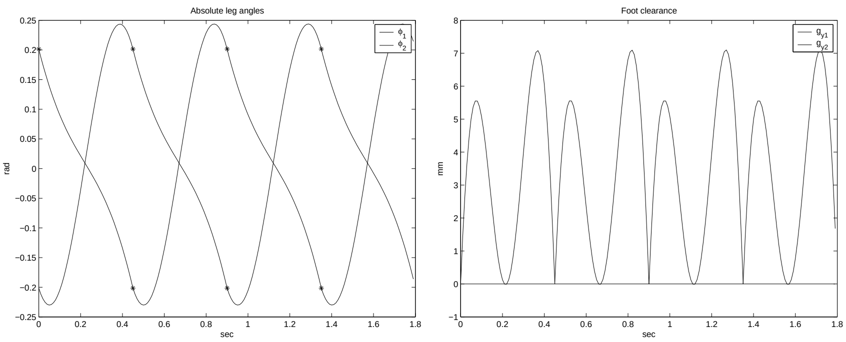

Fig. 4. Result figures produced by `wse_fig.m`.

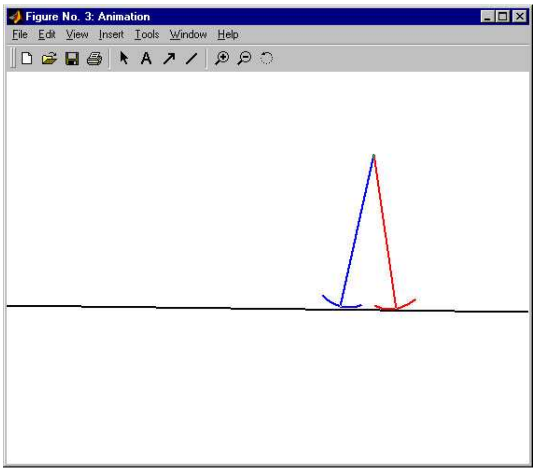

Fig. 5. Animation screen produced by `wse_ani.m`.

postural control at every instant of the motion, aimed to keep the center of gravity above the support polygon. We believe that this static approach (and related approaches such as 'ZMP' control [5]) are suitable for *standing* but not for *walking*. As said before, walking should be regarded as a continuous passive fall with intermittent changes of foot contact. Instead of analyzing the balance at every instant we should analyze the stability of the entire cyclic motion in a step-to-step analysis.

A step-to-step analysis allows us to concentrate on the initial conditions only; the rest of the step is then a predictable passive motion. We can present the initial conditions in a phase-space graph (a plot of ϕ versus $\dot{\phi}$), where any point in the graph represents one specific combination of values for the three independent initial conditions. All points lead to a subsequent motion, but only some of them are successful steps. The end of a successful step is the start of a new one, and so some points in the graph map to some others. This is called 'Poincaré mapping'.

With a little bit of luck (depending on the model parameter values) there are one or two points in the graph that map onto themselves. These are called *fixed points*. They represent a continuous walking motion with all identical steps, which is called a *limit cycle*. With some more luck, one of the fixed points is stable; if the initial conditions are a small deviation away from the fixed point, this deviation disappears over a number of steps until the walker is back in its limit cycle. This stability for small errors is analyzed in the next section. A question that remains is 'how small is small'; for what initial conditions will we still find convergence to the limit cycle? That question is answered with an analysis of the basin of attraction, but that is outside the scope of this paper.

Linearized stability

We need to *find* a fixed point and to *analyze its linear (small-error) stability*. This is easiest understood in reverse order, so for now let's assume that we already know a fixed point. Actually we do, see Table 2. The three independent initial conditions are represented with $\mathbf{v} = [\phi_1, \dot{\phi}_1, \dot{\phi}_2]^T$, whereas we'll call the fixed point $\mathbf{v}_{fp}$. The Poincaré mapping is represented with the nonlinear function S, so that

$$\mathbf{v}_{n+1} = \mathbf{S}(\mathbf{v}_n) \tag{7}$$

where S is a short notation for the complete simulation of one walking step including the heel strike impact and a mirroring of the legs to compare $\mathbf{v}_{n+1}$ with $\mathbf{v}_n$. All initial conditions can be written as a sum of the fixed point plus some deviation:

$$\mathbf{v}_n = \mathbf{v}_{fp} + \Delta\mathbf{v}_n \tag{8}$$

Although S is highly nonlinear, for small deviations from $\mathbf{v}_{fp}$ we can approximate the mapping with a linearization according to

$$\begin{gathered}\mathbf{v}_{fp} + \Delta\mathbf{v}_{n+1} = \mathbf{S}(\mathbf{v}_{fp} + \Delta\mathbf{v}_n) \approx \mathbf{S}(\mathbf{v}_{fp}) + \mathbf{J}\Delta\mathbf{v}_n \\ with \quad \mathbf{J} = \frac{\partial \mathbf{S}}{\partial \mathbf{v}}\end{gathered} \tag{9}$$

This equation simplifies to $\Delta\mathbf{v}_{n+1} = \mathbf{J}\Delta\mathbf{v}_n$. The Jacobian (matrix of partial derivatives) $\mathbf{J}$ here is the key to our linearized stability analysis. Basically it multiplies the errors at step n to produce those at step $n+1$. If the multiplication factor is between -1 and 1, errors decrease step after step and the walker is stable. The multiplication factors are found in the eigenvalues of $\mathbf{J}$ that should all three have a modulus smaller than 1. In the case of the parameter values of Table 2 the eigenvalues are 0.65, 0.22 + 0.30i, and 0.22 - 0.30i, so the model is linearly stable.

Unfortunately, the Jacobian $\mathbf{J}$ is not readily available. It must be obtained by numeric differentiation by means of four full-step simulations; once for the initial conditions of the fixed point and three times to monitor the effect of a small perturbation on each of the initial conditions. This is done in the file wse_lca.m. The resulting eigenvalues of $\mathbf{J}$ tell us whether a fixed point is stable or not. However, more than a simple 'yes' or 'no' cannot be expected, as the actual eigenvalues and eigenvectors do not provide much more insight. To determine which model is 'more stable', one should investigate the maximally allowable disturbance size, which can be found by analysis of the basin of attraction (not in this paper).

Finding a fixed point

Now we know how to analyze a fixed point, but how do we *find* it? The approximation of equation (9) can also be applied to a set of initial conditions *close to* the fixed point (which we need to guess). This will provide an estimate for $\mathbf{J}$. With that estimate and with the difference between the beginning ($\mathbf{v}$) and the end ($\mathbf{S}(\mathbf{v})$) of a step, a Newton-Raphson iteration can be performed that will quickly converge to the fixed point. The iteration procedure as used in wse_lca.m is as follows:

$$\begin{array}{l} \texttt{repeat} \\ \quad \Delta\mathbf{v} = [\mathbf{I} - \mathbf{J}]^{-1}(\mathbf{S}(\mathbf{v}) - \mathbf{v}) \\ \quad \mathbf{v} = \mathbf{v} + \Delta\mathbf{v} \\ \texttt{until } |\Delta\mathbf{v}| < \epsilon \end{array} \tag{10}$$

If this procedure is started for example with $\{\phi_1, \dot{\phi}_1, \dot{\phi}_2\} = \{0.15, -1, -1\}$, it takes 7 iteration steps (± 20 seconds on a 2GHz PC) to arrive at the fixed point with $\epsilon < 10^{-12}$.

Note that the file wse_lca.m always simulates only a single step. In order to compare the end state with the begin state, the end state must be mirrored. This is the standard procedure used in most passive dynamic walking researches, although it is not entirely realistic. In case the model has two legs with different mass properties or in some other special situations [2], the limit cycle analysis should be performed on two subsequent steps which eliminates the necessity for mirroring. The drawback of this is that there is more chance of a fall and thus more difficulty in finding (unstable) cycles with a bad initial guess for the initial conditions.

4 Conclusion

This paper provides the basic tools to simulate a simple, two-dimensional walking model, to find its natural cyclic motion, to analyze the stability, and to investigate the effect of parameter changes on the walking motion and the stability. Especially in conjunction with the accompanying MATLAB files, this paper can serve as a quick and effective start with the concept of passive dynamic walking.

References

1. S. H. Collins, M. Wisse, and A. Ruina. A two legged kneed passive dynamic walking robot. *International Journal of Robotics Research*, 20(7):607–615, July 2001.
2. M. Garcia, A. Chatterjee, A. Ruina, and M. J. Coleman. The simplest walking model: Stability, complexity, and scaling. *ASME J. Biomech. Eng.*, 120(2):281–288, April 1998.
3. A. D. Kuo. Energetics of actively powered locomotion using the simplest walking model. *Journal of Biomechanical Engineering*, 124:113–120, February 2002.
4. T. McGeer. Passive dynamic walking. *Intern. J. Robot. Res.*, 9(2):62–82, April 1990.
5. M. Vukobratovic, B. Borovic, D. Surla, and D. Stokic. *Biped Locomotion: Dynamics, Stability, Control and Applications*. Springer–Verlag, Berlin, 1990.
6. M. Wisse and J. van Frankenhuyzen. Design and construction of mike; a 2d autonomous biped based on passive dynamic walking. In *Proc., Conference on Adaptive Motion of Animals and Machines, AMAM*, Kyoto, Japan, 2003. Paper number WeP-I-1 (8 pages).