Tao Geng
Department of Psychology
University of Stirling
Stirling, UK
runbot05@gmail.com

Bernd Porr
Department of Electronics & Electrical Engineering
University of Glasgow, UK
B.Porr@elec.gla.ac.uk

Florentin Wörgötter
Department of Psychology
University of Stirling
Stirling, UK, and
Bernstein Center for Computational Neuroscience
University of Göttingen, Germany
worgott@chaos.gwdg.de

Fast Biped Walking with a Sensor-driven Neuronal Controller and Real-time Online Learning

Abstract

In this paper, we present our design and experiments on a planar biped robot under the control of a pure sensor-driven controller. This design has some special mechanical features, for example small curved feet allowing rolling action and a properly positioned center of mass, that facilitate fast walking through exploitation of the robot's natural dynamics. Our sensor-driven controller is built with biologically inspired sensor- and motor-neuron models, and does not employ any kind of position or trajectory tracking control algorithm. Instead, it allows our biped robot to exploit its own natural dynamics during critical stages of its walking gait cycle. Due to the interaction between the sensor-driven neuronal controller and the properly designed mechanics of the robot, the biped robot can realize stable dynamic walking gaits in a large domain of the neuronal parameters. In addition, this structure allows the use of a policy gradient reinforcement learning algorithm to tune the parameters of the sensor-driven controller in real-time, during walking. This way RunBot can reach a relative speed of 3.5 leg lengths per second after only a few minutes of online learning, which is faster than that of any other biped robot, and is also comparable to the fastest relative speed of human walking.

KEY WORDS—dynamic biped, reflex, neuronal controller, online learning, fast walking

The International Journal of Robotics Research
Vol. 25, No. 3, March 2006, pp. 243-259
DOI: 10.1177/0278364906063822

1. Introduction

Building and controlling fast biped robots demands a deeper understanding of biped walking than for slow robots (Pratt 2000). While slow robots may walk statically, fast biped walking has to be dynamically balanced and more robust as less time is available to recover from disturbances (Pratt 2000). Although many biped robots have been developed using various technologies in the past 20 years, their walking speeds are still not comparable to that of their counterpart in nature, humans. Some biped robots employ various types of model-based control of an inverted pendulum of the upper body (Kajita and Kobayashi 1987; Miyazaki and Arimoto 1987; Sano and Furusho 1990). Chevallereau et al. (2003) designed a trajectory tracking controller based on the zero dynamics of a planar biped robot with unactuated ankles, by which asymptotically stable walking gaits were realized. Most of the successful biped robots have commonly used the ZMP (Zero Moment Point, Vukobratovic et al. 1990) as the criterion for stability control and motion generation Miyakoshi and Cheng 2002; Hirai 1997; Inoue 2000; Kuroki et al. 2001; Yamaguchi et al. 1999; Nishiwaki 2000). The ZMP is the point on the ground where the total moment generated by gravity and inertia equals zero (Vukobratovic et al. 1990). This measure has two deficiencies for high-speed walking. First, the ZMP must always reside in the convex hull of the stance foot, and the stability margin is measured by the minimal distance between the ZMP and the edge of the foot. To ensure an appropriate stability margin, the foot has to be flat and large, which

will reduce the robot's performance and pose great difficulty during fast walking. This difficulty can be seen clearly when humans try to walk with skis or swimming fins. Second, the ZMP criterion does not permit rotation of the stance foot at the heel or the toe, which, however, can amount to up to 80% of a normal human walking gait (Hardt and von Stryk 2002), and is important and inevitable in fast biped walking.

On the other hand, sometimes dynamic biped walking can be achieved simply without explicitly considering any stability criterion. Specific trajectories and precise trajectory tracking are not indispensable for biped walking. For example, passive biped robots can walk stably down a shallow slope with no sensing or control. Usually equipped with point feet or curved feet, only one point of the foot touches the ground at any time, which would be unstable when applying the ZMP criterion. However, compared with powered bipeds, passive biped robots have obvious drawbacks: for example, their need for walking down a slope and their inability to control the speed (Pratt 2000). Some researchers have proposed approaches to equip a passive biped with actuators to improve its performance. Van der Linde (1998) made a biped robot walk on level ground by pumping energy into a passive biped at each step. Tedrake applied reinforcement learning on a 3D half-passive biped to get dynamic stable gaits (Collins et al. 2005). Nevertheless, no one has yet built a passive biped robot that can walk at a speed comparable to humans, though humans also exploit passive movements in some stages of their walking gaits.

Unlike the robots described above, in humans, stable and somewhat robust biped gaits can emerge from the global entrainment between the neuro-musculo-skeletal system and the environment (Taga 1995). In this study, we will realize fast planar biped walking with a simple neuro-mechanical system, in which a properly designed mechanical structure is directly driven by a neuronal controller. Moreover, the neuronal controller is built with a small number of sensor neurons and motor neurons. Rather than employing intensive feedback control or model-based control as other biped robots usually did, the motor neurons in the neuronal controller directly drive the motors of the joints. It will be shown in our experiments that fast and stable biped walking can emerge from the interaction between such a neuro-mechanical system and the ground.

This paper is organized as follows. First we describe the mechanical design of our biped robot named "RunBot". Next, we present the neural model of our sensor-driven networks for biped walking control. Then we demonstrate the result of biped walking experiments.

2. The Robot

RunBot is a mechanical redesign of our previous robot (Geng, Porr, and Wörgötter 2006) with a simplified controller and specific properties to allow for fast walking. RunBot (see Figure 1) is 23 cm high, foot to hip joint axis. It has four actuated joints: left hip, right hip, left knee, right knee. Each joint is driven by a modified RC servo-motor. A hard mechanical stop is installed on the knee joints, preventing it from going into hyperextension, similar to the function of knee caps in animals. The built-in pulse width modulation control circuits of the RC motors are disconnected while its built-in potentiometer is used to measure the joint angles. Each foot is equipped with a modified piezo transducer to sense ground contact events. We constrain the robot only in the sagittal plane by a boom of one meter length. The robot is attached to the boom via a freely rotating joint while the boom is attached to the central column with a universal joint (see Figure 1). Thus, RunBot's movements are constrained on the surface of a sphere. However, given that the length of the boom is more than 4 times RunBot's height, the influence of the boom on RunBot's dynamics in the sagittal plane is very small. The boom still allows RunBot to freely trip or fall forwards or backwards.

Passive biped robots are usually equipped with circular feet (see Figure 15), which increases the basin of attraction of stable walking gaits, and makes the motion of the stance leg look smoother. Instead, powered biped robots typically use flat feet so that their ankles can more effectively apply torque to propel the robot forward in the stance phase, and to facilitate its stability control. Although RunBot is a powered biped, it has no actuated ankle joints, rendering its stability control even more difficult than that of other powered bipeds, but, on the other hand, an unactuated foot can be very light, which is more efficient for fast walking. Since we intended to exploit RunBot's natural dynamics during some stages of its gait cycle; similar to passive bipeds; its foot bottom is also curved with a radius equal to half the leg length (if the radius is too large, the tip of the foot may strike the ground during its swing phase). During the stance phase of such a curved foot, always only one point touches the ground, thus allowing the robot to roll passively around the contact point, which is similar to the rolling action of human feet. Therefore, with curved feet the difficulties caused by flat feet in fast walking can be avoided. However, how long should such a foot be? In theory, larger curved feet bring more stability for passive biped walking. In practice however, large feet make foot clearance of the swing leg difficult, and tremendously limit the walking speed of the robot. In order to achieve a fast speed, RunBot is equipped with small feet (4.5 cm long) whose relative length, the ratio between the foot-length and the leg length, is 0.20, less than that of humans (about 0.30) and that of other biped robots (powered or passive).

The most important consideration in the mechanical design of our robot is the location of its center of mass. Its links are made of aluminum alloy, which is light but strong enough. The motor of each hip joint is a HS-475HB from Hitec. These weigh 40 g and can produce a torque of up to 5.5 kg cm. Due to the effect of the mechanical stop, the motor of the knee joint

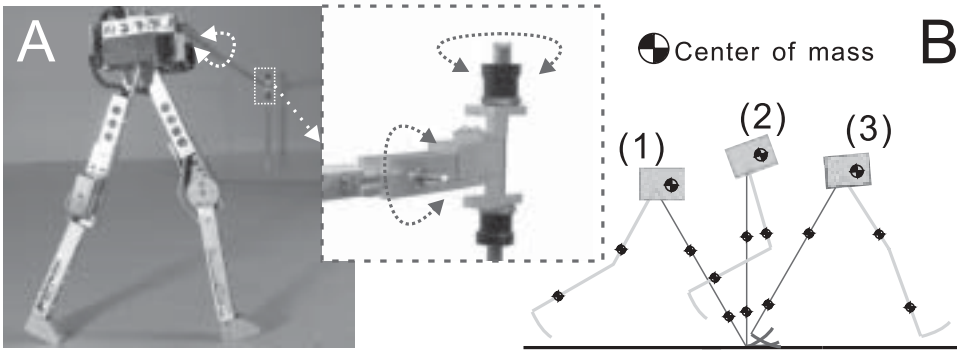


Fig. 1. (A). The robot, RunBot, and its boom structure. The three orthogonal axes of the boom indicated with curved arrows rotate freely. (B). Illustration of a walking step of RunBot.

bears a smaller torque than the hip joint in stance phases, but must rotate quickly during swing phases for foot clearance. We use a PARK HPXF from Supertec on the knee joints, which has a light weight (19 g), but is fast with 21 rad/s. Thus, about 70% of the robot's weight is concentrated on its trunk. The parts of the trunk are assembled in such a way that its center of mass is located in front of the hip axis (see Figure 1). The effect of this design is illustrated in Figure 1(B). As shown, one walking step includes two stages, the first from (1) to (2), the second from (2) to (3). During the first stage, the robot has to use its own momentum to rise up on the stance leg. When walking at a low speed, the robot may have not enough momentum to do this. So, the distance the center of mass has to cover in this stage should be as short as possible, which can be fulfilled by locating the center of mass of the trunk forward. In the second stage, the robot just falls forward naturally and catches itself on the next stance leg. Then the walking cycle is repeated. The figure also shows clearly the rolling movement of the curved foot of the stance leg. A stance phase begins with the heel touching ground, and terminates with the toe leaving ground. To evaluate the effect of the location of the mass center, we have done some simulation. The simulation results are described in Appendix B.

In summary, our mechanical design of RunBot has the following special features that distinguish it from other powered biped robots and facilitate high-speed walking and exploitation of natural dynamics:

(a) small curved feet allowing for rolling action;

(b) unactuated, hence light, ankles;

(c) lightweight structure;

(d) light and fast motors;

(e) proper mass distribution of the limbs;

(f) properly positioned mass center of the trunk.

3. The Neural Structure of our Sensor-Driven Controller

The sensor-driven walking controller of RunBot is a simplified version of our former design (Geng, Porr, and Wörgötter 2006). It follows a hierarchical structure (see Figure 2). The bottom level represents the neuron modules local to the joints, including motor neurons and angle sensor neurons. The top level is a distributed neural network consisting of hip stretch receptors and ground contact sensor neurons, which modulate the motor neurons of the bottom level. Neurons are modeled as non-spiking neurons simulated on a Linux PC, and communicated to the robot via a DA/AD board. Though somewhat simplified, they still retain some of the prominent neuronal characteristics.

The directions of the extensor (flexor) movements and the thresholds of the sensor neurons are illustrated in Figure 3. At the bottom level, the function of the thresholds of the sensor neurons ($\Theta_{ES,h}$, $\Theta_{FS,h}$, $\Theta_{ES,k}$, $\Theta_{FS,k}$, see Figures 2 and 3) in each neuron module is to roughly limit the extensor and flexor movements of the joint. At the top level, the functions of the AEA signal and the ground contact signal are shown in Figure 4.

3.1. Model Neuron Circuit of the Top Level

The joint coordination mechanism in the top level is implemented with the neuron circuit illustrated in Figure 2. The ground contact sensor neuron of each leg has excitatory connections to the motor neurons of the hip flexor and knee extensor of the same leg as well as to the hip extensor and knee flexor of the other leg. The stretch receptor of each hip has excitatory (inhibitory) connections to motor neuron of the knee extensor (flexor) in the same leg. Detailed models of the stretch receptor, and ground contact sensor neuron are described in the following sections.

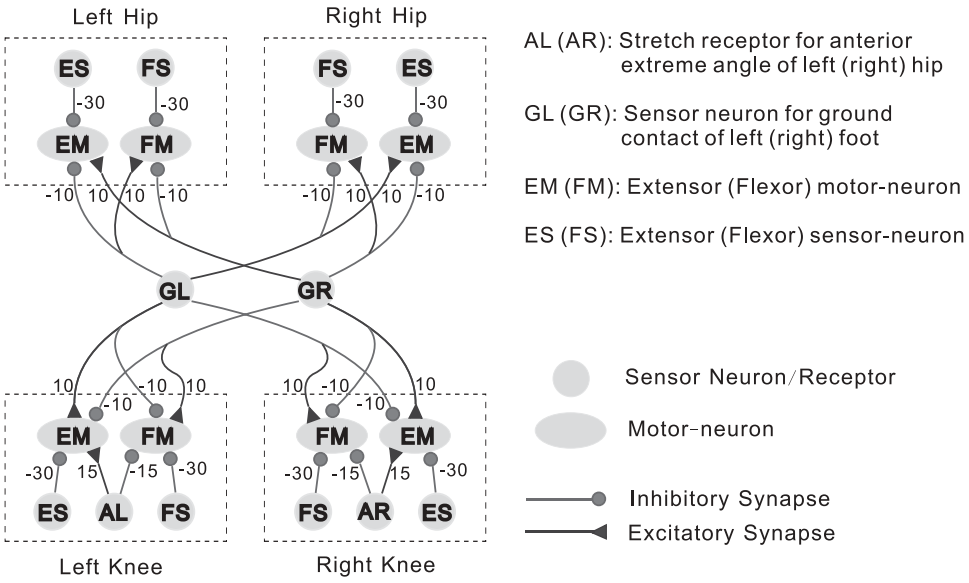


Fig. 2. The neuron model of the sensor-driven controller on RunBot. The small numbers give the values of the connection weights.

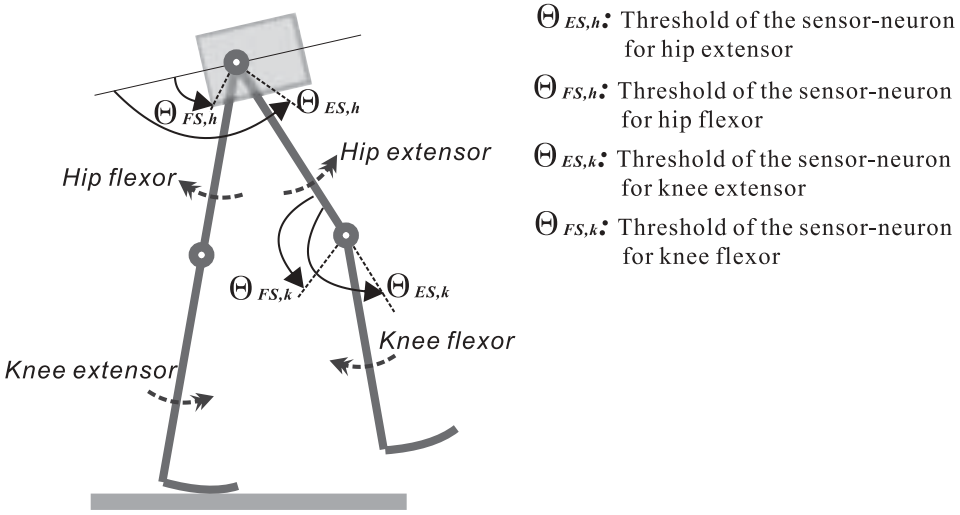


Fig. 3. Control parameters for the joint angles.

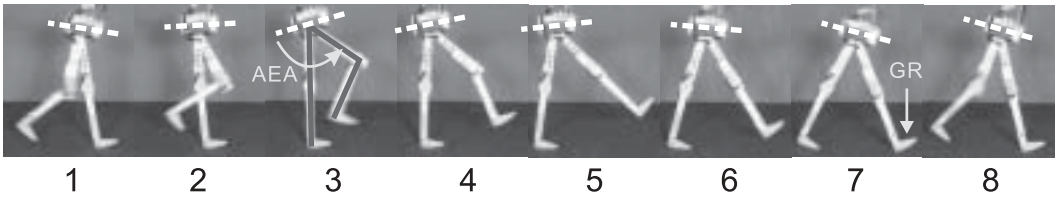


Fig. 4. Series of frames of one walking step. At the time of frame (3), The stretch receptor (AEA signal) of the swing leg is activated, which triggers the extensor of the knee joint in this leg. At the time of frame (7), the swing leg begin to touch the ground. This ground contact signal triggers the hip extensor and knee flexor of the stance leg, as well as the hip flexor and knee extensor of the swing leg. The swing leg and the stance leg swap their roles thereafter.

3.1.1. Stretch Receptors

Stretch receptors play a crucial role in animal locomotion control. When the limb of an animal reaches an extreme position, its stretch receptor sends a signal to the controller, resetting the phase of the limbs. There is also evidence that phasic feedback from stretch receptors is essential for maintaining the frequency and duration of normal locomotive movements in some insects (Chiel and Beer 1997).

While other biologically inspired locomotive models and robots use two stretch receptors on each leg to signal attainment of the leg's AEP (Anterior Extreme Position) and PEP (Posterior Extreme Position) respectively, our robot has only one stretch receptor on each leg to signal the AEA (Anterior Extreme Angle) of its hip joint. Furthermore, the function of the stretch receptor on our robot is only to trigger the extensor motor neuron on the knee joint of the same leg, rather than to implicitly reset the phase relations between different legs as in the case of Cruse's model.

As the hip joint approaches the AEA, the output of the stretch receptors for the left (AL) and the right hip (AR) is increased as

$$\rho_{AL} = \left(1 + e^{\alpha_{AL}(\Theta_{AL} - \phi)}\right)^{-1} \quad (1)$$

$$\rho_{AR} = \left(1 + e^{\alpha_{AR}(\Theta_{AR} - \phi)}\right)^{-1}. \quad (2)$$

Here ϕ is the real-time angular position of the hip joint, Θ_{AL} and Θ_{AR} are the hip anterior extreme angles whose values are tuned by hand, α_{AL} and α_{AR} are positive constants. This model is inspired by a sensor neuron model presented in Wadden and Ekeberg (1998) that is thought capable of emulating the response characteristics of populations of sensor neurons in animals.

3.1.2. Ground Contact Sensor Neurons

Another kind of sensor neuron incorporated in the top level is the ground contact sensor neuron, which is active when the foot is in contact with the ground. Its output, similar to that of the stretch receptors, changes according to

$$\rho_{GL} = (1 + e^{\alpha_{GL}(\Theta_{GL} - V_L + V_R)})^{-1} \quad (3)$$

$$\rho_{GR} = (1 + e^{\alpha_{GR}(\Theta_{GR} - V_R + V_L)})^{-1}. \quad (4)$$

Here V_L and V_R are the output voltage signals from piezo sensors of the left foot and right foot respectively. Both of them are used as inputs of each ground contact sensor neuron to prevent these two neurons from being activated at the same time. Θ_{GL} and Θ_{GR} work as thresholds, α_{GL} and α_{GR} are positive constants.

While AEP and PEP signals account for switching between stance phase and swing phase in other walking control structures, ground contact signals play a crucial role in phase transition control of our sensor-driven controller. In PEP/AEP-models, the movement pattern of a leg will break down as soon as the AEP or PEP cannot be reached, which may happen as a consequence of an unexpected disturbance from the environment or due to intrinsic failure. This can be catastrophic for a biped, though tolerable for a hexapod due to its high degree of redundancy.

3.2. Neural Circuit of the Bottom Level

The neuron module for each joint is composed of two angle sensor neurons and the motor neurons they contacts (see Figure 2). Whenever its threshold is exceeded, the angle sensor neuron directly inhibits the corresponding motor neuron (see Figure 2). This direct connection between angle sensor neurons and motor neurons is inspired by a motor neuron described in cockroach locomotion (Beer et al. 1997). In addition, each motor neuron also receives an excitatory synapse and an inhibitory synapse from the neurons of the top level, by which the top level can modulate the neuron module of the bottom level.

The model of angle sensor neurons is similar to that of the stretch receptors described above. The extensor angle sensor neuron changes its output according to

$$\rho_{ES} = \left(1 + e^{\alpha_{ES}(\Theta_{ES} - \phi)}\right)^{-1} \quad (5)$$

where ϕ is the real-time angular position obtained from the potentiometer of the joint (see Figure 3). Θ_{ES} is the threshold of the extensor motor neuron (see Figure 3) and α_{ES} a positive constant.

Likewise, the output of the flexor sensor neuron is modeled as

$$\rho_{FS} = \left(1 + \mathrm{e}^{\alpha_{FS}(\phi - \Theta_{FS})}\right)^{-1} \tag{6}$$

where Θ_{FS} and α_{FS} similar as above.

It should be particularly noted that the thresholds of the sensor neurons in the motor neuron modules do not work as desired positions for joint control, because our sensor-driven controller does not involve any exact position control algorithms that would ensure that the joint positions converge to a desired value. In fact, as will be shown in the walking experiments, the hip joints often pass these thresholds in swing- and stance phase, and move continuously until the friction of the joint gears stops it. However, during fast walking, the knee joints usually cannot attain the thresholds of their flexor-motor-neuron sensor neurons (see Figure 7(B)) because the phase-switching is so quick.

The definition and direction of the joint angles is illustrated in Figure 3. The direction of extensor on both hip and knee joints is forward while that of flexors is backward.

The motor neuron model is adapted from one used in the neural controller of a hexapod simulating insect locomotion (Beer and Chiel 1992). The state and output of each extensor motor neuron is governed by eqs. (7) and (8) (Gallagher et al. 1996). Those of flexor motor neurons are similar:

$$\tau \frac{\mathrm{d}y}{\mathrm{d}t} = -y + \sum \omega_X \rho_X \tag{7}$$

$$u_{EM} = \left(1 + \mathrm{e}^{\Theta_M - y}\right)^{-1} \tag{8}$$

Here y represents the mean membrane potential of the neuron. Equation (8) is a sigmoidal function that can be interpreted as the neuron's short-term average firing frequency, Θ_M is a bias constant that controls the firing threshold. τ is a time constant associated with the passive properties of the cell membrane (Gallagher et al. 1996), ω_X represents the connection strength from the sensor neurons and stretch receptors to the motor neuron (Figure 2). ρ_X represents the output of the sensor neurons and stretch receptors that contact this motor neuron (e.g., ρ_{ES}, ρ_{AL}, ρ_{GL}, etc.)

Note that, in RunBot, the output value of the motor neurons, after multiplication by a gain coefficient, is sent to the servo amplifier to directly drive the joint motors.

The voltage of the motor in each joint is determined by

$$\text{Motor Voltage} = M_{AMP} G_M (s_{EM} u_{EM} + s_{FM} u_{FM}), \tag{9}$$

where M_{AMP} represents the magnitude of the servo amplifier, which is 3 on RunBot. G_M stands for output gain of the motor neurons in the joint. s_{EM} and s_{FM} are signs for the motor voltage of flexor and extensor in the joint, being $+1$ or -1, depending on the polarity of the motors. u_{EM} and u_{FM} are the outputs of the motor neurons (see Figure 2).

3.3. Tuning the Neuron Parameters

Most of the values for the neuron parameters are chosen intuitively. In this section, we address the tuning of the various neuron parameters except two parameters at the hip joints, $\Theta_{ES,h}$ (see Figure 3) and $G_{M,h}$ (the gain of the motor neurons in hip joints), which will be tuned in the experiments below.

The positive constants of the sensor neurons and the stretch receptors (α_{ES}, α_{FS}, α_{AL}, α_{AR}, α_{GL}, α_{GR}) affect their response speed. We set these constants to 2, ensuring a quick response of these neurons. Our experiments have shown that values bigger than 2 do not make any evident difference in RunBot's gaits.

The threshold of the sensor neurons for the extensor (flexor) in the neuron module roughly limits the movement range of the joint. The thresholds of these sensor neurons in the neuron modules of the knee joints are chosen as: $\Theta_{FS,k} = 110^\circ$, $\Theta_{ES,k} = 175^\circ$ (see Figure 3), which is in accordance with the observation of humans' normal gaits. The movements of the knee joints are needed mainly for timely ground clearance without making big contributions to the walking speed. After some trials, we set the gain of the motor neurons in knee joints to be $G_{M,k} = 0.9 G_{M,h}$.

The threshold of the stretch receptors is simply chosen to be the same as that of the sensor neurons for the hip extensor, $\Theta_{AL(AR)} = \Theta_{ES,h}$.

The threshold of the ground contact sensor neurons is chosen to be 2 volts according to test results on the piezo sensors. In a certain range, the output voltage of the piezo sensor is roughly proportional to the pressure acted on the foot bottom when it is touching the ground.

The time constant of the motor neurons, τ (see eq. 8), is chosen as 10 ms, which is in the normal range of data in biology.

To simplify the problem, we also fix the threshold of the flexor sensor neurons of the hips ($\Theta_{FS,h}$) to 85°.

There are three kinds of synapses in the neuronal controller (see Figure 2). Here we use following symbols to represent the absolute value of the weights of these synapses:

- W_{GM}: Weights of the synapses between the ground contact sensor neurons and the motor neurons.
- W_{AM}: Weights of the synapses between the stretch receptors and the motor neurons.
- W_{SM}: Weights of the synapses between the angle sensor neurons and the motor neurons in the neuron modules of the joints.

The threshold of the motor neurons, Θ_M (see eq. 8), can be any positive value as long as following conditions are satisfied:

$W_{GM} \geq \Theta_M + 4$
$W_{AM} - W_{GM} \geq \Theta_M + 4$
$W_{SM} - W_{AM} - W_{GM} \geq \Theta_M + 4.$

The function of these rules is to make sure that, among all the neurons which contact the motor neurons, the angle sensor neurons in the neuron modules of each joint have the first priority while the stretch receptors have second priority and the ground contact sensor neurons have the lowest priority. So, we simply choose them as $\Theta_M = 1$, $W_{GM} = 10$, $W_{AM} = 15$, $W_{SM} = 30$ (see Figure 2).

Obviously, the function of this neuronal controller can also be realized with a simple mode-switching controller. We prefer using model neurons for the following reasons.

(a) The passive properties of the cell membrane (see eq. 8) can naturally make the output of the neuronal controller much smoother (see Figure 6), thus reducing the jerk in the joint movement.

(b) Our long-term aim is to investigate the effect of neuronal plasticity on the walking behavior with a biped robot. Neuronal plasticity will be embodied by a high-level neural structure, which can then be seamlessly connected with this neuronal controller.

4. Robot Walking Experiments with the Sensor-Driven Controller

In the experiments described below, we only need to tune the two parameters of the hip joints: the threshold of the extensor sensor neurons ($\Theta_{ES,h}$) and the gain of the motor neurons ($G_{M,h}$). They work together to determine the walking speed and the gait properties of RunBot.

In experiments with walking on a flat floor, surprisingly, we have found that stable gaits can appear in quite a large range of the parameters $\Theta_{ES,h}$ and $G_{M,h}$ (see Figure 5).

Figure 6 shows the motor voltages of the four joints while RunBot is walking at medium speed. During more than half of every cycle of each joint, its motor voltage remains zero, allowing unactuated movements of the joints.

As shown in Figure 6, during some period of every step (e.g., gray area in Figure 6), the motor voltages of the motor neurons on all the four joints remain zero, so RunBot's movement is unactuated until the swing leg touches the ground (see Figure 13(A)). During this time, which is roughly one-third of a step (see Figure 6 and Figure 13(A)), the movement of the whole robot is exclusively following its natural dynamics that is dominated by gravity, the inertia of the links, and the properties of the motors and gears; no feedback based active control acts on it. This demonstrates very clearly how the sensor-driven controller and the mechanical properties work together to generate the whole gait trajectory. It is also similar to what happens in animal locomotion. Muscle control of animals usually exploits the natural dynamics of their limbs. For instance, during the swing phase of the human walking gait, the leg muscles first experience a power spike to begin leg swing and then remain limp throughout the rest of the swing phase (Pratt 2000), similar to what is shown in Figure 13.

4.1. Changing Speed on the Fly

RunBot's walking speed can be changed on the fly without problems by tuning $\Theta_{ES,h}$ and $G_{M,h}$ as long as they still remain in the stable area shown in Figure 5. Figure 7 shows the gait when the parameters are changed greatly and abruptly from point S to F (see Figure 5) at a time instant t (indicated with a line in Figure 7). The walking speed is immediately changed from slow (0.38 m/s) to fast (0.70 m/s). By exploiting the natural dynamics, the sensor-driven controller is robust to such drastic parameter variations as shown in Figure 5. The video clip (extension 1) of this experiment can be seen at, http://www.cn.stir.ac.uk/~tgeng/runbot/speedchange.mpg

4.2. Walking on Irregular Terrain

With parameters in the central area in Figure 5, the walking gait shows more robustness. As shown in Figure 8, RunBot can walk over a low obstacle with a height of 0.9 cm. Figure 9 shows a stick diagram of RunBot's gait walking down a shallow slope of 5°. Note that RunBot can neither detect the disturbance nor adjust any parameters of its controller to address it. Nonetheless, after the disturbance, the walking gait soon returns to its normal orbit, demonstrating that the walking gait is to some degree robust against disturbances.

5. Fast Walking with Online Policy Searching

Because there is no position or trajectory tracking control in RunBot, it is impossible to control its walking speed directly. Moreover, the sensor-driven neuronal controller does not employ any form of dynamics model of the robot, ruling out the possibility of analytically or explicitly describing the relationship between neuronal parameters and the walking speed.

However, knowing that RunBot's walking gait is determined almost exclusively by two parameters, $\Theta_{ES,h}$ and $G_{M,h}$ (Figure 11), we formulate RunBot's fast walking control as a policy gradient reinforcement learning problem by considering each point in the parameter space (Figure 11) as an open-loop policy that can be executed by RunBot in real time.

Our approach is similar to that of Kohl and Stone (2004), except for the algorithms for adaptive step size and for local optimum avoiding, which are designed by us particularly for the biped walking in RunBot. Learning starts from an initial parameter vector $\pi_0 = (\theta_1, \theta_2)$ (here θ_1 and θ_2 represent $G_{M,h}$ and $\Theta_{ES,h}$, respectively) and proceeds to evaluate the following five polices at or near π:

$R_1 = (\theta_1, \theta_2)$
$R_2 = (\theta_1, \theta_2 - \epsilon_2)$

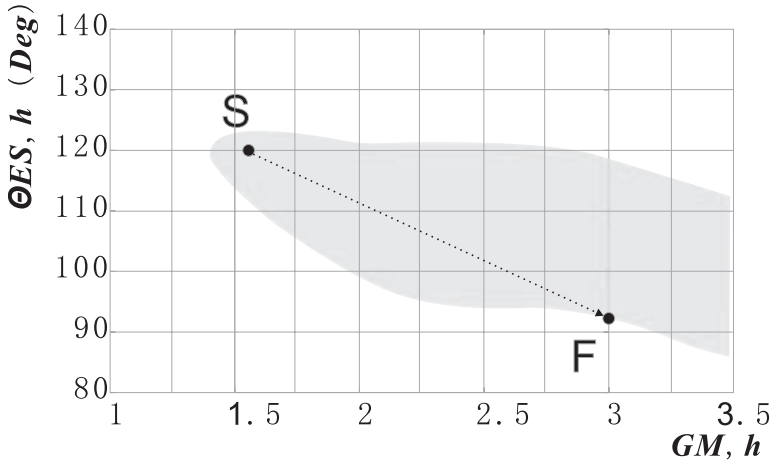


Fig. 5. The shaded areas are the range of the two parameters, in which stable gaits appear. The maximum permitted value of $G_{M,h}$ is 3.45 (higher values will destroy the motor of the hip joint).

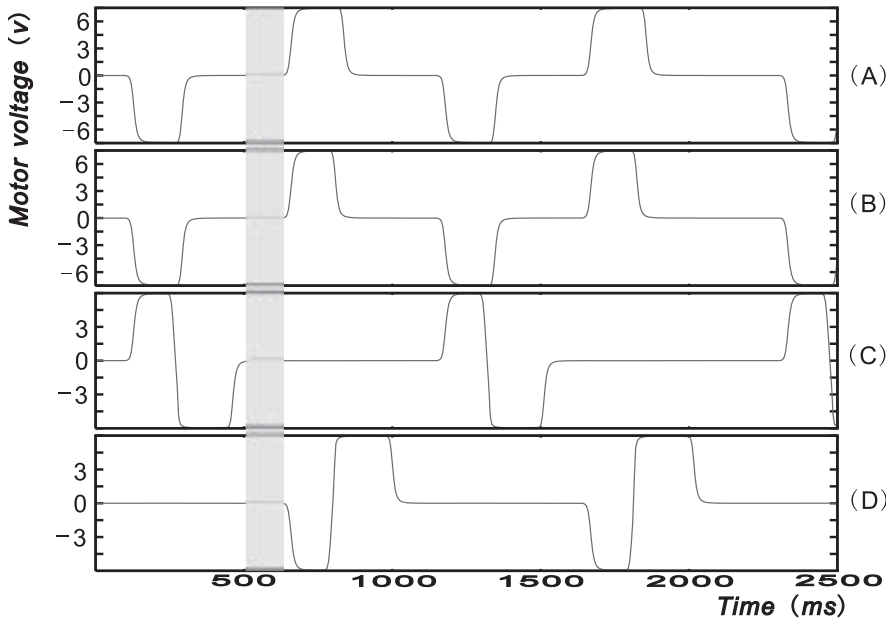


Fig. 6. Motor voltages sent to the servo amplifiers directly from the motor neurons while the robot is walking. (A) left hip; (B) right hip; (C) left knee; (D) right knee. Note that during some period of every gait cycle (gray area), all four motor voltages remain zero and the whole robot moves without actuation. (see Figure 13).

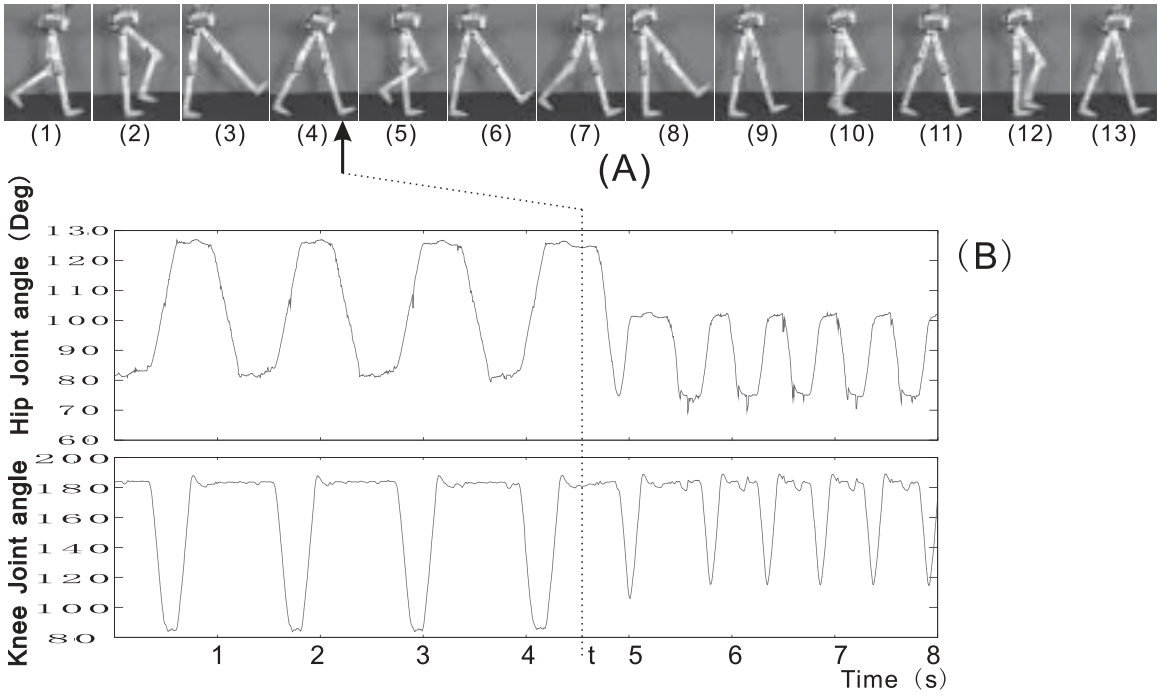


Fig. 7. (A) Series of sequential frames of the walking gait. The neuron parameter is changed at the time of frame (4). The interval between two adjacent frames is 133 ms. (B) Real-time data of the angular position (in trunk coordinates as illustrated in Figure 3) of hip joint and knee joint of one leg, indicated with an arrow in frame (4) of (A), while the walking speed is changed at time t.

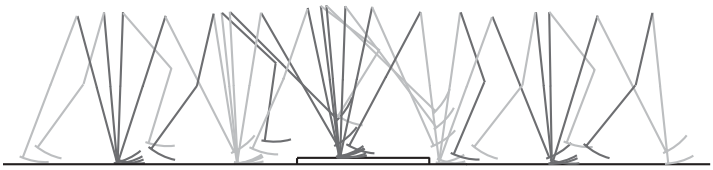


Fig. 8. Stick diagram of RunBot walking over a low obstacle (9mm high, higher ones cannot be tackled). The interval between any two consecutive snapshots is 100 ms.

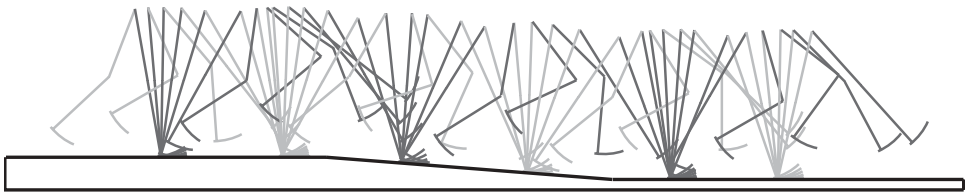


Fig. 9. Stick diagram of RunBot walking down a shallow slope of 5°. The interval between any two consecutive snapshots is 67 ms.

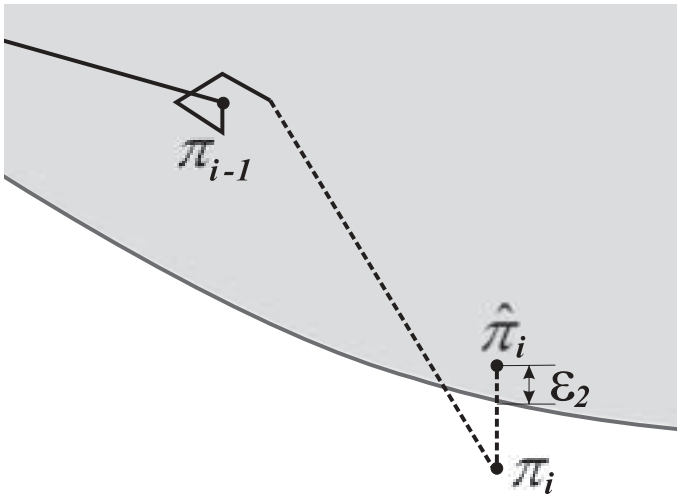


Fig. 10. If the parameter vector π_i is not appropriate, it will be "pushed" back into the stable area. See text for more information.

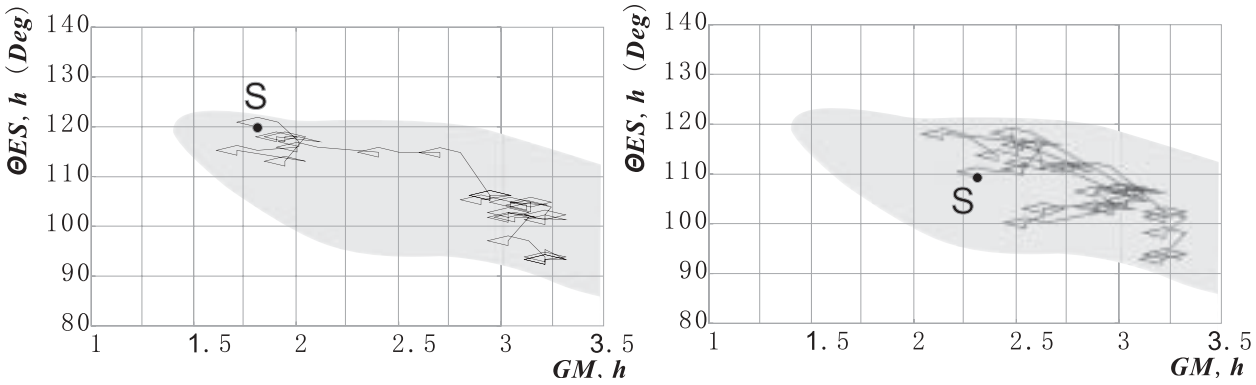


Fig. 11. Changing of the controller parameters, $G_{M,h}$ and $\Theta_{ES,h}$, during two experiments of online learning. See text for more information.

$$R_3 = (\theta_1 - \epsilon_1, \theta_2)$$
$$R_4 = (\theta_1, \theta_2 + \epsilon_2)$$
$$R_5 = (\theta_1 + \epsilon_1, \theta_2)$$

where each ϵ_j is a fixed value that is small relative to θ_j. The evaluation of each policy generates a score, S_{Ri}, that is a measure of the speed of the gait described by that policy (R_i). We use these scores to construct an adjustment vector A (Kohl and Stone 2004):

$$A_1 = 0 \quad \text{if} \quad S_{R1} > S_{R3} \quad \text{and} \quad S_{R1} > S_{R5}$$
$$A_1 = S_{R5} - S_{R3} \quad \text{otherwise.}$$

Similarly,

$$A_2 = 0 \quad \text{if} \quad S_{R1} > S_{R2} \quad \text{and} \quad S_{R1} > S_{R4}$$
$$A_2 = S_{R4} - S_{R2} \quad \text{otherwise.}$$

If $A = 0$, this means a possible local optimum is encountered. In this case, we replace A with a stochastically generated vector. Although this is a very simple strategy, our experiments show that it can effectively prevent the real-time learning from getting trapped locally.

Then A is normalized and multiplied by an adaptive step-size:

$$\eta = \eta_0(v_{max} - s_{max})/v_{max} \tag{10}$$

where v_{max} stands for the maximum speed RunBot has ever attained during the time before. s_{max} is the maximum value of S_{Ri} of this current iteration. η_0 is a constant. If $\eta < \eta_{min}$ (or $\eta > \eta_{max}$), it is set to be η_{min} (or η_{max}). η_{min} and η_{max} are predefined lower and upper limits for η.

We use a sensor at the central axis of the boom to measure the angular speed of the boom when RunBot is walking, from which the walking speed can be calculated. To get an accurate walking speed, each policy is executed for N_{cyc} gait cycles (one gait cycle includes two steps). Because the speed of the first gait cycle of each policy is still influenced by the last policy, it is neglected and the average speed of these $N_{cyc} - 1$ cycles is regarded as the speed of the gait corresponding to this policy. At the beginning of the learning process, N_{cyc} is set to be 2. Then N_{cyc} is recalculated at the end of each iteration according to the following rule:

$$N_{cyc} = (\text{int})((v_{max} - v_{min})/3)$$
$$\text{if} \quad N_{cyc} < 2, N_{cyc} = 2$$
$$\text{if} \quad N_{cyc} > 6, N_{cyc} = 6$$

where v_{min} stands for the minimum speed RunBot has ever attained during gait cycles before.

Finally, A is added to π_0, obtaining a new parameter vector, π_1, and the next iteration begins.

Results are shown in Figure 11 (left) and Figure 12. RunBot starts walking with parameters at point S in Figure 11 (left) corresponding to a speed of 41 cm/s (see Figure 12(C)). After 240 s of continuous walking with the learning algorithm and no human intervention, RunBot attains a walking speed of about 80 cm/s (see Figure 12(C), which is equivalent to 3.5 leg lengths per second. Figure 13 shows video frames of walking gaits at a fast and a medium speed, respectively, in which we can clearly see the change of gaits during the process of the learning.

In another experiment, RunBot starts walking with different parameters corresponding to point S in Figure 11 (right). The data of this experiment are shown in Figure 14. In 280 s, the robot also attains a speed of around 80 cm/s (see Figure 14).

In the two experiments of online learning reported above, the learning started from policies located in the upper or middle part of the stable area (see Figure 11). In this case, the subsequent policies usually do not exceed the boundaries of the stable area. But, in other experiments, if learning starts from a policy near the lower boundary of the stable area, subsequent policies can indeed sometimes leave the stable area. To prevent this, we use the following strategy: At the beginning of the ith iteration, if any of the five policies that will be evaluated at or near π_i is located outside the stable area, the vector π_i is replaced with another vector in the stable area, $\hat{\pi}_i$, which is nearest to π_i on the coordinate of θ_1 (θ_2), and has a distance of ϵ_1 (ϵ_2) to the boundary of the stable area (see Figure 10).

In the experiments with RunBot, self-stabilizing properties as a result of increasing speed, such as those suggested by Seyfarth and Blickhan (2002) and Poulakakis and Buehler (2003) in monopod and quadruped, only seem to happen to a limited degree when starting the learning from a policy near the upper boundary or middle of the stable area. It is usually a puzzling problem how to quantitatively measure the stability of the walking robots (like RunBot) that do not use any kind of dynamics model. The eigenvalues of the linearized Poincaré map are often used for stability analysis of the walking robots (Garcia 1999). In simulations, the eigenvalues of the linearized Poincaré map can be calculated by minutely perturbing the robot from the fixed point in each dimension. In real robots, however, the lack of sufficient and accurate sensor signals make this kind of idealized analysis very difficult (if not impossible). To build the Poincaré map of the RunBot's gait, we need both the position and the speed data of the four actuated joints and the unactuated stance ankle joint. But in RunBot, only the position data of the four actuated joints are available. Even on these four joints, due to the noise and inaccuracy of the potentiometers, measuring tiny perturbations is almost impossible.

To compare the walking speed of various biped robots whose sizes are quite different from each other, we use the relative speed, speed divided by the leg length. Maximum relative speeds of RunBot and some other typical planar biped robots (passive or powered) are listed in Figure 15. We know of no other biped robot attaining such a fast relative speed. The world record for human walking is equivalent to about 4.0–4.5 leg lengths per second. So, RunBot's highest walking speed is comparable to that of

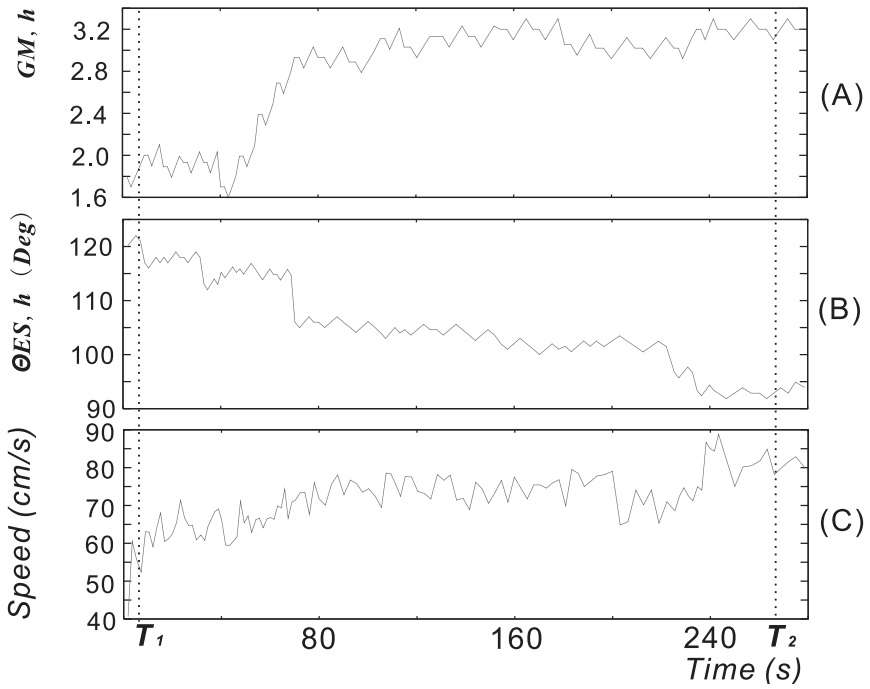


Fig. 12. Real-time data of one experiment. Changes of the controller parameters (A) and (B) and the walking speed (C) during the entire process of learning.

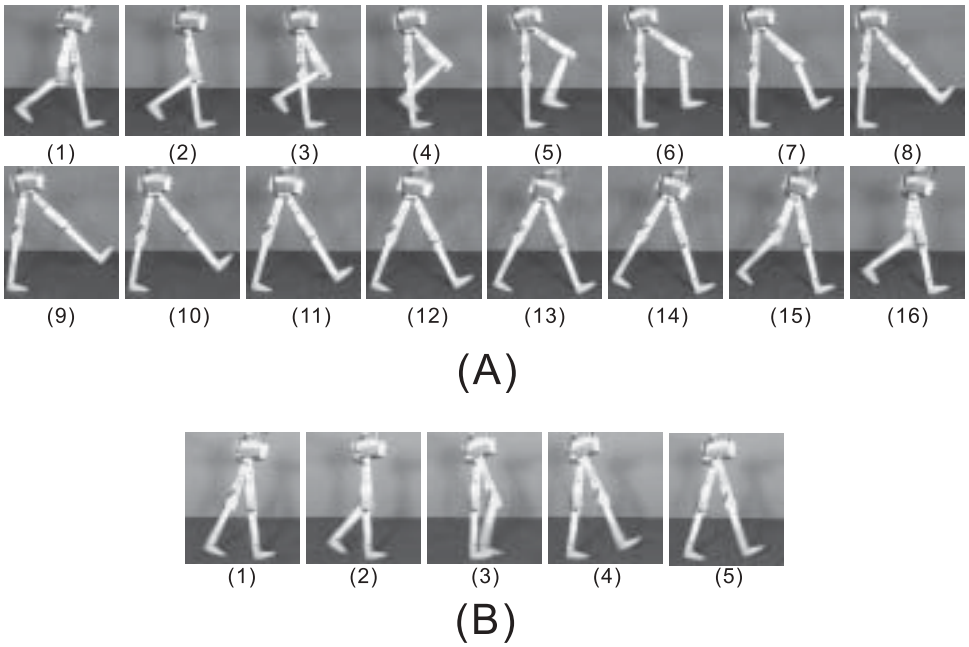


Fig. 13. Series of sequential frames of two walking gaits. The interval between two adjacent frames is 33 ms. (A) Gait of a medium speed (53 cm/s), the parameter values of which are indicated as T_1 in Figure 12. Note that during the time between frame (8) and frame (13), which is nearly one-third of the duration of a step (corresponding to the gray area in Figure 6), the whole robot is moving without actuation. At the time of frame (13), the swing leg touches the floor and a next step begins. (B) Gait of a fast speed (80 cm/s), the parameter values of which are indicated as T_2 in Figure 12.

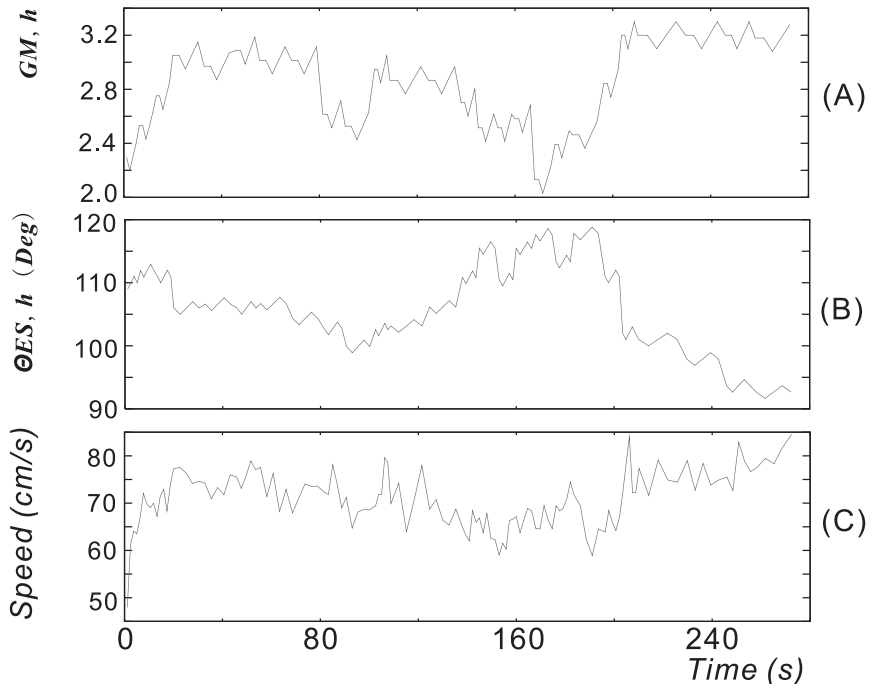
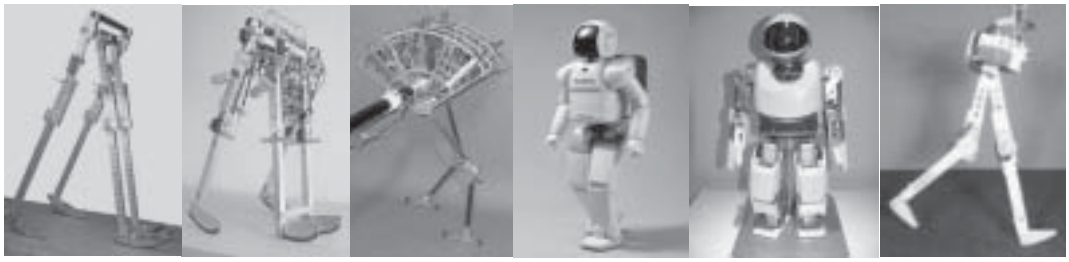


Fig. 14. Real-time data of another experiment. Changes of the controller parameters (A) and (B) and the walking speed (C) during the entire process of learning.



Leg length	0. 8m	0. 75m	0. 9m	0. 67m	0.28m	0. 23m
Max. speed	0. 6m/s	0. 40m/s	1. 25m/s	0. 69m/s	0.25m/s	0. 80m/s
Max relative Speed	0. 75	0. 53	1. 39	1.03	0.89	3.48
Froude Number	0.046	0.021	0.177	0.073	0.023	0.28
	(a)	(b)	(c)	(d)	(e)	(f)

Fig. 15. Relative leg length and maximum relative speed of various biped robots. (a) A copy of McGeer's planar passive biped robot walking down a slope Wisse 2003 **(NOT IN REFS.)**. (b) "Mike", similar to McGeer's robot, but equipped with pneumatic actuators at its hip joints. Thus it can walk half-passively on level ground Wisse 2003 **(NOT IN REFS.)**. (c) "Spring Flamingo", a powered planar biped robot with actuated ankle joints (Pratt 2000). (d) Asimo-3, Honda's humanoid robot (www.asimi.honda.com). (e) SDR-3X, Sony's entertainment biped robot (www.sony.net). (f) RunBot.

humans. To get a feeling of how fast RunBot can walk, we strongly encourage readers to watch a video clip (extension 2, http://www.cn.stir.ac.uk/~tgeng/runbot/learning.mpg), which recorded the final 80 s of RunBot's walking during an experiment of online learning.

Biped robots can help us to better understand the biomechanics of humans' walking if their gaits are dynamically similar. The Froude number, Fr, has been used to describe the dynamical similarity of legged locomotion over a wide range of animal sizes and speeds on earth (Alexander and Jayes 1983):

$$Fr = v^2/gl. \tag{11}$$

Here v is the walking speed, g gravity and l leg length. The Froude number of some typical biped robots are listed in Figure 15, most of which are far below the normal value of the adult human's Froude Number of 0.20 (Vaughan and O'Malley 2005), indicating that they are indeed not dynamically similar to adult humans, though some of them have been designed to mimic human walking (Vaughan and O'Malley 2005). However, 0.20 is in the attainable range of RunBot's Froude number (see Figure 15), implying that RunBot's walking gait, when at an appropriate speed (0.67 m/s), could with some confidence be regarded as dynamically similar to that of an adult human.

6. Discussion

Here, we will briefly discuss some remaining issues of RunBot, because most of the relevant discussion points have been treated in the above sections.

Our sensor-driven controller has some evident differences from Cruse's model. Cruse's model depends on PEP, AEP and GC (Ground Contact) signals to generate the movement pattern of the individual legs, whereas our sensor-driven controller uses only GC and AEA signals to coordinate the movements of the joints. Moreover, the AEA signal of one hip in RunBot only acts on the knee joint belonging to the same leg, not functioning on the leg-level as the AEP and PEP did in Cruse's model. The use of fewer phasic feedback signals has further simplified the controller structure in RunBot.

In order to achieve real-time walking gait in a real world, biologically inspired robots often have to depend on some kinds of position or trajectory tracking control on their joints (Beer et al. 1997; Fukuoka, Kimura, and Cohen 2003; Lewis 2001) However, in RunBot, there is no position or velocity control implemented. The neural structure of our sensor-driven controller does not depend on, or ensure the tracking of, any desired position. Indeed, it is this approximate nature of our sensor-driven controller that allows the physical properties of the robot itself (see the experiments), to contribute implicitly to generation of overall gait trajectories, and ensures its stability and robustness to some extent.

7. Conclusion

In this study, we have shown that, with a properly designed mechanical structure, a simple neuronal sensor-driven controller, and an online policy gradient reinforcement learning algorithm, our biped robot can attain a fast relative walking speed of 3.5 leg lengths per second, which is not only faster than any other biped walking robot we know of, but also comparable to a human's fastest walking speed.

This paper is concentrated on the robot experiments of online policy searching. An important issue remaining to be investigated is to explicitly analyze the attraction domain of its stable gaits and its relationship to the mechanical and controller parameters, which will have to be done next.

Appendix A: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org.

Table of Multimedia Extensions

Extension	Type	Description
1	Video clip	Changing speed on the fly
2	Video clip	Online learning experiment of RunBot

Appendix B: Simulations of the Influence of the Center of Mass of the Trunk

The dynamics of our robot are modeled as shown in Figure 16. With the Lagrange method, we can get the equations that gov-

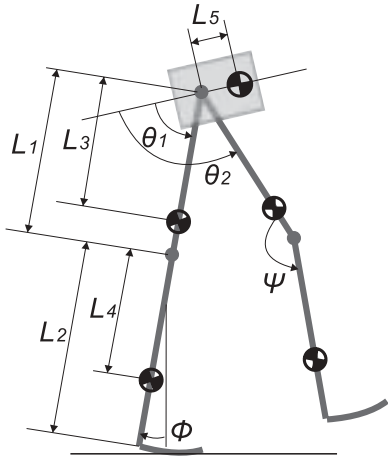


Fig. 16. Model of the dynamics of our robot. Sizes and masses are the same as those of the real robot.

ern the motion of the robot, which can be written in the form

$$D(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau \tag{B1}$$

where $q = [\phi, \theta_1, \theta_2, \psi]^{\mathrm{T}}$ is a vector describing the configuration of the robot (for definition of $\phi, \theta_1, \theta_2, \psi$, see Figure 16). $D(q)$ is the 4×4 inertia matrix, $C(q, \dot{q})$ is the 4×1 vector of centripetal and coriolis forces, $G(q)$ is the 4×1 vector representing gravity forces. $\tau = [0, \tau_1, \tau_2, \tau_3]^{\mathrm{T}}$, τ_1, τ_2, τ_3 are the torques applied on the stance hip (the hip joint of the stance leg in Figure 16), the swing hip, and the swing knee joints, respectively.

Considering that the electrical time constant of the motor is much smaller than the mechanical time constant of the robot, we neglect the dynamics of the electrical circuits of the motor. Thus the dynamics of the DC motor (including gears) of each joint can be described with the following equation (here, the hip of the stance leg is taken as an example; the models of other joints are likewise):

$$\tau_1 = -I_1\ddot{\theta}_1 - k_a\dot{\theta}_1 + k_b V_1 \tag{B2}$$

where V_1 is the applied armature voltage of the stance hip motor, I_1 is the combined moment of inertial of the stance-hip motor and gear train referred to the gear output shaft, and k_a and k_b are coefficients determined by the properties of the motor and gear. Details of eqs. (B1) and (B2) can be found in our previous paper (Geng, Porr, and Wörgötter 2006).

Combining eqs. (B1) and (B2), we can get the dynamical model of the robot with the applied motor voltages as its control input, while the motor voltages are directly calculated from the outputs of the motor neurons of the controller.

The heel strike at the end of swing phases and the knee strike at the end of knee extensor reflex are assumed to be inelastic impacts. This assumption implies the conservation of angular momentum of the robot just before and after the strikes, with which the value of $\dot{q}$ just after the strikes can be computed using its value just before the strikes. Because the transient double support phase is very short in RunBot's walking, it is neglected in our simulation as often done in the analysis of other passive bipeds (Garcia 1999).

The method of Poincaré maps is usually employed for stability analysis of cyclic movements of non-linear dynamic systems such as passive bipeds (Garcia 1999). We choose the Poincaré section (Garcia 1999) to be right after the heel strike of the swing leg. Each cyclic walking gait is a limit cycle in the state space, corresponding to a fixed point on the Poincaré section. Fixed points can be found by solving the roots of the mapping equation

$$P(x^n) - x^n = 0 \tag{B3}$$

where $x^n = [q, \dot{q}]^{\mathrm{T}} = [\phi, \theta_1, \theta_2, \psi, \dot{\phi}, \dot{\theta}_1, \dot{\theta}_2, \dot{\psi}]^{\mathrm{T}}$ is a state vector on the Poincaré section at the beginning of the n_{th} gait cycle. $P(x^n)$ is a map function mapping x^n to x^{n+1}, which is built numerically by combining the neuronal controller and the robot dynamics model described above.

Near a fixed point, x^*, the map function $P(x^*)$ can be linearized as (Garcia 1999)

$$P(x^* + \hat{x}) \approx P(x^*) + J\hat{x} \tag{B4}$$

where J is the 8×8 Jacobian matrix of partial derivatives of P.

With any fixed point, J can be obtained by numerically evaluating P eight times in a small neighborhood of the fixed point. If all eigenvalues of J lie within the unit cycle, the gait is asymptotically stable (Garcia 1999).

The values of the neuron parameters in the simulation are chosen the same as those in the real robot. Moreover, to simplify the problem, we also fix the gain of the motor neurons of the hip joints, i.e., $G_{M,h} = 2.5$ (at the middle of the stable area in Figure 5). Thus, we only need to adjust the value of $\Theta_{ES,h}$ to change the properties of the gaits.

To see how the location of the mass center (L_5 in Figure 16) of the trunk affect the stability and the speed of the gaits, we also change the value of L_5 in the simulation. With each set of L_5 and $\Theta_{ES,h}$, we use a multi-dimensional Newton–Raphson method solving eq. (B3) to find the fixed point (Garcia 1999). Then we compute the Jacobian matrix J of the fixed point using the approach described in Garcia (1999), and evaluate the stability of the fixed point according to its eigenvalues. The simulation results are shown in Figure 17.

Because some details of the robot dynamics such as uncertainties of the ground contact, nonlinear frictions in the joints and the inevitable noise and lag of the sensors cannot be modeled precisely, the results of the simulation suggest a larger stable range than the real experiments. For example, in the real robot, the mass center of the trunk is located about 3 cm forward. With $G_{M,h} = 2.5$, stable gaits can appear when $\Theta_{ES,h}$ is in the range of 95–122° (see Figure 5). But in the simulation, the stable range of $\Theta_{ES,h}$ is somewhat bigger, 90–136° (see the curve indicated with $L_5 = 3$ cm in Figure 17). However, the simulation results have shown that the location of the mass center of the trunk does have a drastic influence on the stability and the speed of the gaits:

(a) a small value of L_5 (see Figure 16 and 17) is helpful to the stability of the gaits at slow walking speeds;

(b) if the mass center of the trunk is located appropriately forward (e.g., $L_5 = 3, 5$ cm in Figure 17), stable range and walking speed can both be improved;

(c) but, if the mass center is located too far forward (e.g., $L_5 = 7$ cm in Figure 17), the stable range for the neuron parameters will become quite small, though the walking speed can be very high.

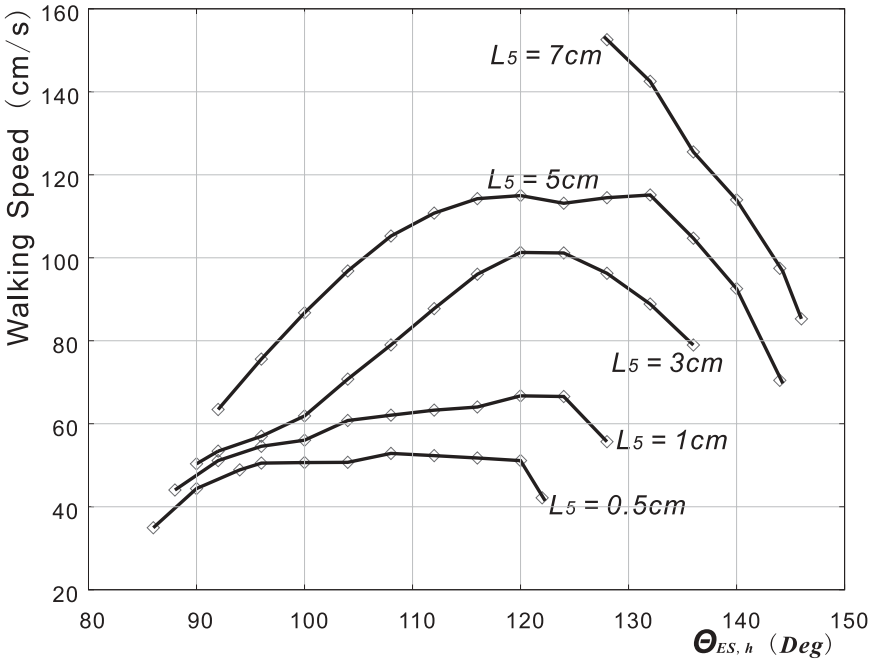


Fig. 17. Change of walking speed while $G_{M,h}$ is fixed at 2.5 and $\Theta_{ES,h}$ is changed in its stable range. Each curve corresponds to a different location of the mass center of the trunk (see Figure 16): i.e. $L_5 = 0.5$, 1, 3, 5, and 7 cm.

Acknowledgments

This work was supported by SHEFC grant INCITE to Professor F. Wörgötter. We thank Dr Peter Hancock for correction of the text.

References

Alexander, R. and Jayes, A. 1983. A dynamic similarity hypothesis for the gaits of quadrupedal mammals. *Journal of Zoology* 201:135–152.

Beer, R. and Chiel, H. 1992. A distributed neural network for hexapod robot locomotion. *Neural Computation* 4:356–365.

Beer, R., Quinn, R., Chiel, H., and Ritzmann, R. 1997. Biologically inspired approaches to robotics. *Communications of the ACM* 40(3):30–38.

Chevallereau, C., Abba, G., Aoustin, Y., Plestan, F., Westervelt, E., Canudas-de Wit, C., and Grizzle, J. 2003. Rabbit: A testbed for advanced control theory. *IEEE Control Systems* 23:57–79.

Chiel, H. and Beer, R. 1997. The brain has a body: adaptive behavior emerges from interactions of nervous system, body, and environment. *Trends in Neuroscience* 20:553–557.

Collins, S., Ruina, A., Tedrake, R., and Wisse, M. 2005. Efficient bipedal robots based on passive-dynamic walkers. *Science* 37:1082–1085.

Fukuoka, Y., Kimura, H., and Cohen, A. 2003. Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts. *International Journal of Robotics Research* 22:187–202.

Gallagher, J., Beer, R., Espenschied, K., and Quinn, R. 1996. Application of evolved locomotion controllers to a hexapod robot. *Robotics and Autonomous Systems* 19:95–103.

Garcia, M. 1999. *Stability, scaling, and chaos in passive-dynamic gait models*. Ph.D. thesis, Cornell University.

Geng, T., Porr, B., and Wörgötter, F. 2006. A reflexive neural network for dynamic biped walking control. *Neural Computation*. Accepted for publication.

Hardt, M. and von Stryk, O. 2002. The role of motion dynamics in the design, control and stability of bipedal and quadrupedal robots. In *LNAI Proceedings of RoboCup*, pp. 206–223.

Hirai, K. 1997. Current and future perspective of honda humanoid robot. *Proceedings of the International Conference on Intelligent Robots and Systems* pages 500–509. IEEE.

Inoue, H. and Tachi, S. 2000. HRP: Humanoid robotics project of MITI. In *Proceedings of the First International Conference on Humanoid Robots*. IEEE.

Kajita, S. and Kobayashi, A. 1987. Dynamic walk control of a biped robot with potential energy conserving orbit. *Journal of SICE* 23:281–287.

Kohl, N. and Stone, P. 2004. Policy gradient reinforcement learning for fast quadrupedal locomotion. *Proceedings of*

the IEEE International Conference on Robotics and Automation, Vol. 3, pp. 2619–2624.

Kuroki, Y., Ishida, T., Yamaguchi, J., Fujita, M., and Doi, T. 2001. A small biped entertainment robot. *Proceedings of International Conference on Humanoid Robots*, Piscataway, NJ, IEEE, pp. 181–186.

Lewis, M. 2001. Certain principles of biomorphic robots. *Autonomous Robots* 11:221–226.

Miyakoshi, S. and Cheng, G. 2002. Ballistic walking by compass-like biped walker—exploiting physical dynamics in achieving human-like walking. *Proceedings of 5th International Conference on Climbing and Walking Robots*, pp. 445–452.

Miyazaki, F. and Arimoto, S. 1987. A control theoretic study on dynamical biped locomotion. *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control* 102:233–239.

Nishiwaki, K., Sugihara, T., Kagami, S., Kanehiro, F., Inaba, M., and Inoue, H. 2000. Design and development of research platform for perception-action integration in humanoid robot. *Proceedings of the International Conference on Intelligent Robots and Systems*, Piscataway, NJ, IEEE, pp. 1559–1564.

Poulakakis, I., P. E. and Buehler, M. 2003. On the stable passive dynamics of quadrupedal running. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1368–1373.

Pratt, J. 2000. *Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots*. Ph.D. thesis, Massachusetts Institute of Technology.

Sano, A. and Furusho, J. 1990. 3D dynamic walking of biped robot by controlling the angular momentum. *Journal of SICE* 26:459–466.

Seyfarth, A., G. H. G. M. and Blickhan, R. 2002. A movement criterion for running. *Journal of Biomechanics* 35:649–655.

Taga, G. 1995. A model of the neuro-musculo-skeletal systems for human locomotion. *Biological Cybernetics* 73:97–111.

Van der Linde, R. Q. V. 1998. Active leg compliance for passive walking. *Proceedings of IEEE International Conference on Robotics and Automation*, Orlando, FL.

Vaughan, C. L. and O'Malley, M. J. 2005. Froude and the contribution of naval architecture to our understanding of bipedal locomotion. *Gait and Posture* 21:350–362.

Vukobratovic, M., Borovac, B., Surla, D., and Stokic, D. 1990. *Biped locomotion: dynamics, stability, control and application*. Berlin, Springer.

Wadden, T. and Ekeberg, O. 1998. A neuro-mechanical model of legged locomotion: Single leg control. *Biological Cybernetics* 79:161–173.

Wisse, M., Frankenhuyzen, J.V. 2003. Design and construction of Mike; a 2D autonomous biped based on passive dynamic walking. In *Proceedings 2nd International Symposium on Adaptive Motion of Animals and Machines*, Kyoto, Japan, Springer, pp. 235–240.

Yamaguchi, J., Soga, E., Inoue, S., and Takanishi, A. 1999. Development of a bipedal humanoid robot control method of whole body cooperative dynamic biped walking. *Proceedings of the International Conference on Robotics and Automation*, Piscataway, NJ, IEEE, pp. 368–374.