The RunBot Architecture for Adaptive, Fast, Dynamic Walking

Poramate Manoonpong*, Tao Geng†, Bernd Porr‡ and Florentin Wörgötter*

*Bernstein Center for Computational Neuroscience
University of Göttingen, D-37073 Göttingen, Germany
Email: poramate@bccn-goettingen.de, worgott@bccn-goettingen.de

†Department of Computer Science
University of Essex, Wivenhoe Park, Colchester CO4 3SQ, U.K.
Email: tgeng@essex.ac.uk

‡Department of Electronics& Electrical Engineering
University of Glasgow, Glasgow, Scotland
Email: B.Porr@elec.gla.ac.uk

***Abstract*— In this paper we will present the architecture of the planar biped robot "RunBot". It has been developed on the basis of three hierarchical levels: $Biomechanical$, $Local$ and $Central$. The biomechanical level concerns an appropriate biomechanical design of RunBot which utilizes some principles of passive walkers to ensure stability. The local level is a low-level neuronal structure which generates dynamically stable gaits as well as fast motions with some degree of self-stabilization to guarantee basic robustness. In the central level, we simulate a mechanism for synaptic plasticity which allows RunBot to autonomously learn to adapt its locomotion to different terrains, e.g. level floor versus up or down a ramp. As a result, the structural coupling of all these levels generates adaptive, fast dynamic walking of RunBot.**

I. Introduction

Adaptive, fast dynamic walking of biped robots has been one of the most intense ambitions in robotic research because, by this, one hopes to achieve human-like performance and energy efficiency while walking. Most recent studies had focused on the mechanical design, especially on so-called passive dynamic walkers which are simple mechanical devices that can walk stably down a shallow slope [1]. Adding actuators to their joints may increase their capabilities to walk also on a level surface [1] but these systems *cannot* easily *adapt their gait* in different terrains and/or *change their speed*.

Traditionally, successful advanced biped robots, e.g. [2], [3], have been built based on closed-loop control of joint-angle positions [4]. By using such a method, it is difficult to relate these machines to human walking, because such closed-loop control requires highly precise actuators unlike muscles, tendons, and human joints, which do not operate with this precision. Moreover, such robot systems require much energy, which is in conflict with measured human power consumption during walking or running [5] and their control is non-neuronal.

From these points of view, we have developed our robot system "RunBot", in a stepwise manner during the last four years [6], [7], [8]. The system covers now the achievement (adaptive, fast dynamic walking) using only few components and reaching a speed of up to $\approx$ 3.5 leg-length/s [6], which has so far not been achieved with other walkers. Although it is still a planar robot it is nonetheless a dynamic walking machine, which does not use any explicit gait calculation or trajectory control, but instead fully relies on its two neuronal control levels. Furthermore, it even can learn to adapt its gait in accordance with terrain conditions. The following section describes the architecture of our robot system. Experiments and results are discussed in section 3. Conclusions are given in the last section.

II. The RunBot Architecture

The RunBot architecture has been designed in general following the classical subsumption architecture [9]. We divide our robot system into three levels (or layers) where they are organized as a hierarchical structure and coupled via the environment. Each level is described in the following sections.

A. Mechanical Setup of RunBot (Biomechanical Level)

RunBot is 23 cm high, foot to hip joint axis (see Fig. 1). Its legs have four actuated joints: left hip, right hip, left knee and right knee. Each joint is driven by a modified RC servo motor where the built-in Pulse Width Modulation (PWM) control circuit is disconnected while its built-in potentiometer is used to measure the joint angles. A mechanical stopper is implemented on each knee joint to prevent it from going into hyperextension. The motor of each hip joint weights 40 g and can produce a torque up to 5.5 kg·cm while the motor of each knee joint produces a smaller torque (3 kg·cm) but has fast rotating speed with 21 rad/s for foot clearance during swing phases. Approximately seventy percent of the robot's weight is concentrated on its trunk and the parts of the trunk are assembled in a way that its center of mass is located forward of the hip axis. RunBot has no actuated ankle joints resulting in very light feet and being efficient for fast walking. Its feet were designed having a small circular form (4.5 cm long). Each foot is equipped with a switch sensor to detect ground contact events. Hip and knee joints are driven by output signals of the leg controller (running on a Linux PC) through a DA/AD converter board (USB-DUX).

To extend its walking capabilities for walking on different terrains, e.g. level floor versus up or down a ramp, one servo motor with a fixed mass, called the *upper body component* (UBC), is implemented on top. The UBC has a total weight

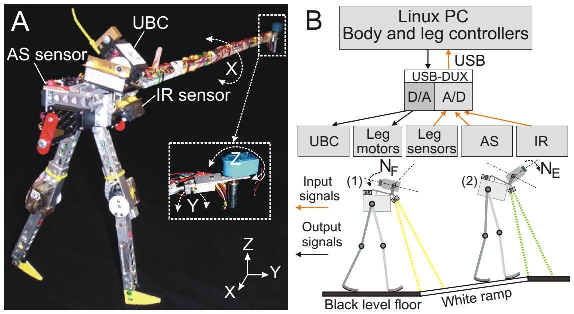

Fig. 1. (A) The planar robot $RunBot$ with its UBC. (B) Schematic set-up of the RunBot system. Leg sensors consist of joint angle and ground contact switch sensors, leg motors are the motors of the left and right hip and knee joints. The detection range of the IR sensor for slope sensing is shown in the lower figure where the yellow ray of the IR sensor (1) indicates that the sensor gives a high output signal while the green ray (2) means a low signal. Hence the sensor responds more strongly to the white ramp.

of 50 g. It leans backward (see Fig. 1B) during walking on a level floor and this position is also suitable for walking down a ramp [8] while it will lean forward (reflex action) when RunBot falls backwards or after it successfully learned to walk up a ramp (see Fig. 1B). The corresponding reflex is controlled by an accelerometer sensor (AS), see Fig. 1A. The AS is installed on top of the right hip joint. In addition, one infrared (IR) sensor is implemented at the front part of RunBot (see Fig. 1A) pointing downwards to detect a ramp. Here, the IR sensor serves as a simple vision system, which can distinguish between a level floor with black color and a painted ramp (white color). This sensory signal is used for adaptive control in the central level. The scheme of our set-up is shown in Fig. 1B. RunBot is constrained sagitally by a boom of one meter length. It is attached to the boom via a freely-rotating joint in the x axis while the boom is attached to the central column with freely-rotating joints in the y and z axes (see Fig. 1A). Thus, the motions of RunBot are only constrained on a circular path. This set-up has no influence on dynamics of RunBot in the sagittal plane. RunBot's design [6] has some special features, e.g. small curved feet and a properly positioned center of mass, which rely quite strongly on the concepts of self-stabilization of gaits in passive walkers [1]. These properties, depicted by the *loop* 1 (Biomechanics) in Fig. 2, allow the robot to perform passive dynamic walking during some stage of its gait cycles (see Fig. 4A (red areas)). Note that, the reason to develop the small-sized RunBot is that the cost of building a small robot is very low and a small design can be used in experiments to describe or test the performance of a design of big size as long as they are dynamically similar (i.e., they have the same Froude Number [6]). However, RunBot's design can be scaled up if we can find light and powerful motors to drive the big-sized design.

B. Reflexive Neuronal Controller (Local Level)

The reflexive neuronal controller consists of two networks: one, a $low-level\ reflexive\ neuronal\ network$, is for leg control and the other, a $long-loop\ reflexive\ neuronal\ network$, is for body (UBC) control. Both networks have a distributed implementation but they are indirectly coupled through the biomechanical (Bio.) level (see Fig. 2). Neurons are modelled as non-spiking neurons with the standard sigmoid transfer function. They are simulated on a Linux PC with an update frequency of 250 Hz (see Fig. 1B).

The leg-motor control circuit (see Fig. 2) simulated as mono-synaptic connections contains motor neurons (N) [10], which are linear and can send their signals unaltered to the motors M. Furthermore, there are several local sensor neurons, which by their conjoint reflex-like actions trigger the different walking gaits. These local sensor neurons can be distinguished into three local loops: joint control ($Local_1$), intra-joint control ($Local_2$) and leg control ($Local_3$). Joint control arises from sensors S at each joint, which measure the joint angle and influence only their corresponding motor neurons. Intra-joint control is achieved from sensors A, which measure the anterior extreme angle at the hip and trigger an extensor reflex at the corresponding knee. Leg control comes from ground contact sensors G, which influence the motor neurons of all joints.

The body-motor control circuit (see Fig. 2) represents a long-loop reflex ($Central_1$) which is directly modulated by its sensor AS. However, this sensor is also involved in controlling synaptic plasticity within the whole network for adaptive walking. Here we first present its pure reflex function prior to learning. The UBC is controlled by its flexor and extensor motor neurons N_F, N_E driven by the signal of one accelerometer sensor neuron AS. On flat terrain, AS is inactive and the flexor motor neuron N_F is activated to lean the body backward while the extensor motor neuron N_E is inhibited. This situation is reverted when a strong signal from the accelerometer sensor exists, which happens only when RunBot falls backwards, e.g. RunBot tries to walk up a ramp. This will trigger a leaning reflex of the UBC. More detailed descriptions of all neuron models together with the neuronal network structures and the discussion of their parameters can be found in [6], [7], [8].

Many walking robots have used attitude control as the sensors of those robots could provide enough information required by attitude control (e.g., speed and acceleration signals at each joint). By contrast, RunBot's controller is a minimal one, using very limited sensor signals which are not eligible for explicit attitude control. RunBot's stability comes from the coupling of its biomechanical structure with the reflexive neuronal controller. Consequently, it can perform dynamic walking with self-adapting to minor disturbances and even reacting in a robust way to abruptly induced gait changes [6], [7] (see Fig. 4B).

C. Adaptive Neuronal Controller (Central Level)

In the local level, we have implemented a long-loop body reflex at the UBC, triggered by a strong backward lean. This reflex behavior can be changed by learning. The learning goal in this study is to finally avoid the reflex and thereby learn to also change gait parameters in an appropriate way to prevent RunBot from falling. RunBot's task was to learn walking up a ramp and then continue again on a level floor. This requires an

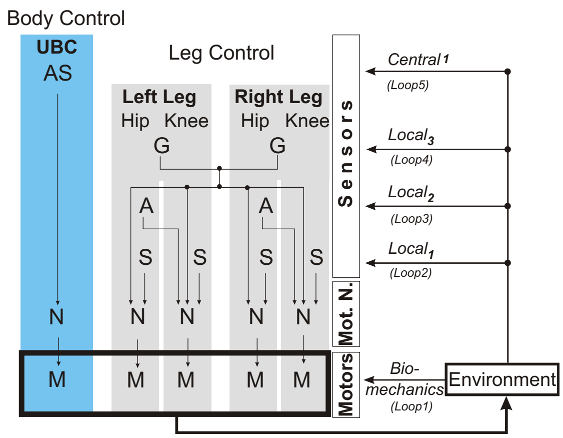

Fig. 2. The different reflexive control levels of RunBot (solid lines). The black box at the bottom represents RunBot's physical embodiment, colored boxes its neuronal control and sensor networks. Walking control arises from the interplay of the different sensori-motor loops ($Local$, $Central$) implemented in RunBot together with its passive dynamic walking properties ($Biomechanics$). Abbreviations are: $Mot.N.$ = motor neurons, AS = accelerometer sensor neuron, G = ground contact sensor neuron, A = stretch receptor neuron for anterior extreme angle of the hips, S = local angle sensor neuron of hips and knees, N = motor neuron, M = motor.

adaptive network of six more learner neurons ($L_{1,2,...,6}$) (see Fig. 3) which converge onto target neurons at the reflexive networks in the local level effectively changing their activation parameters.

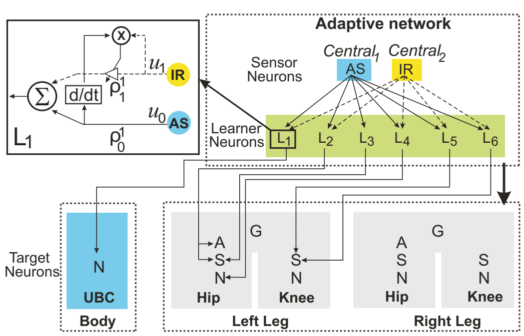

Fig. 3. The complete neuronal control structure where the adaptive network (central level, see text for details) is implemented on top as a high level control to modulate the reflexive networks (local level) through learner neurons. Connections between learner neurons and target neurons of the right leg, which are identical to those of the left leg, are not shown. Learning mechanism (L_1, see text for details) is shown in a solid frame. Note that, all learner neurons have the same learning mechanism.

We know from previous experiments [8] that a stable gait for upslope walking can be obtained by controlling the posture of the UBC as well as changing RunBot's gait. Here, the leaning of the UBC is controlled by exciting or inhibiting $N_{E,F}$ while changing the gait is achieved by adjusting the following local neuronal parameters. At the knee joints, the firing threshold of neurons $S_{E,F}$ has to be decreased. While, at the hip joints, the firing threshold of neurons $S_{E,F}$, which also effects the stretch receptor neurons A, has to be increased but the gain g of motor neurons $N_{E,F}$ has to be decreased [8]. This leads to smaller steps also observed in humans when climbing.

In our learning algorithm [11] (described below), the modification of all those parameters will be controlled by two kinds of input signals: one is an early input (called predictive signal) and the other is a later input (called reflex signal). Here, we use the IR signal as a predictive signal while the AS signal serves as a reflex signal. Both sensory signals are provided to the learner neurons as shown in Fig. 3. At the beginning, the connections between the predictive signal and learner neurons converge with zero strengths (dashed arrows in Fig. 3). In this situation, parameters of the target neurons will be altered only by the reflex signal; i.e. the leaning reflex of the UBC together with the gait adaptation will be triggered by the AS signal (solid arrows between the reflex signal and learner neurons in Fig. 3) as soon as RunBot falls. Hence, RunBot will begin to walk up the ramp with a wrong set of gait parameters and an inappropriate posture of the UBC. Thus, it will eventually fall leading to a signal at the AS, which will change RunBot's parameters but too late (when it already lies on the ground). Due to learning the modifiable synapses, which connect the predictive IR-signal with the learner neurons, will grow (see Fig. 5c). Consequently, after 3-5 falls during the learning phase, gait adaptation together with posture control of the UBC will finally be driven by the predictive IR-signal instead. Correspondingly, RunBot will adapt its gait together with leaning the UBC in time. The used learning algorithm has the property that learning will stop when the reflex signal is zero [11]; i.e. when RunBot does not fall anymore. On returning to flat terrain, the IR output will get small again and RunBot will change its locomotion back to normal for walking on a level floor. Note that the same circuitry and mechanisms can be used to learn different gaits for other given tasks, e.g. walking down a ramp.

Learning Algorithm: In general, each learner neuron L_n requires two input signals (u_0, u_1) with synaptic weights (ρ_0, ρ_1) (see a solid frame (L_1) in Fig. 3). Here, we use the AS and the IR signals as u_0 and u_1, respectively. Only ρ_1 (dashed arrows in Fig. 3) is allowed to change through plasticity while ρ_0 (solid arrows in Fig. 3) is set to a positive value. The output activity v of L_n and the learning rule for the weight change ρ_1^n are given by:

$$v(L_n) = \rho_0^n u_0 + \rho_1^n u_1, \quad n = 1, \ldots, 6, \tag{1}$$

$$\frac{d\rho_1^n}{dt} = \mu_n u_1 \frac{du_0}{dt}, \quad n = 1, \ldots, 6; \tag{2}$$

where we use only input signals to correlate with each other [11]. μ_n is the learning rate. It is independently set for each learner neuron which will define the desired equilibrium point and how fast the system can learn. Here, we set μ_1 = 10, μ_2 = 7.0, μ_3 = 10.5, μ_4 = 0.14, μ_5 = 3.0, μ_6 = 10.0. In neurons with multiple inputs such a mechanism can be used to modify the synaptic strengths according to the order of the arriving inputs. As a consequence, the predictive input will get strengthened if the predictive signal u_1 is followed by the reflex input u_0, where the reflex drives the neuron into firing. This rule will lead to weight stabilization as soon as $u_0 = 0$ [11], hence, when the reflex has successfully been avoided. As a result we obtain behavioral and synaptic stability at the same time without any additional weight-control mechanisms.

III. Robot Walking Experiments

In this section, we demonstrate the performance of the RunBot system. Fig. 4A presents the passive properties of RunBot reflected from the biomechanical design ($Bio.\ level$). While Fig. 4B shows that the intrinsic robustness of the RunBot system makes neuronal parameter fine-tuning unnecessary ($Local\ level$); i.e. it is possible to immediately switch manually from a slower walking speed of 39 cm/s to a faster one of 73 cm/s. Fig. 5 illustrates that the adaptive control with the learning technique ($Central\ level$) enables RunBot to learn to adapt its gait in different terrains. As a result, RunBot can manage to walk on an eight degree ramp after 3 falls which is approximately 14 s of learning time and average walking speed was about 50 cm/s. The video clips of all experiments can be seen at www.chaos.gwdg.de/∼poramate/Runbot.html.

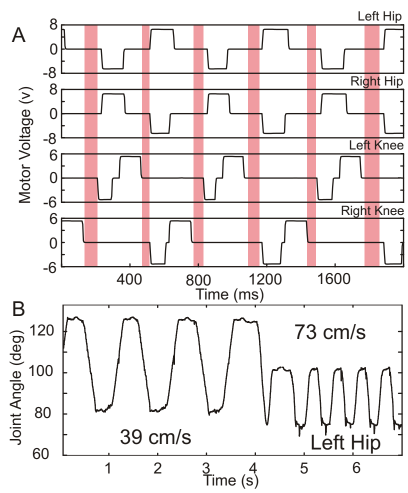

Fig. 4. (A) Motor voltages directly sent from the leg motor neurons to the servo amplifiers while the robot is walking. Red areas indicate when all four motor voltages remain zero during some stage of every gait cycle where the robot walks passively. (B) Real-time data of the left hip joint angle recorded during walking and changing speed on the fly. Parameters are changed greatly and abruptly for all extensor sensor thresholds of the hip joint from 120.0 deg to 93.0 deg and for all motor neuron gain values of the hip joint from 1.55 to 3.0. This way speed changed from 39 cm/s to 73 cm/s.

IV. Conclusions

RunBot's architecture is constructed with a set of nested loops (see Fig. 2). In this way there exists tight coupling of the different levels of physical and neuronal control via feedback from the environment. Such an architecture allowed us to additionally implement an adaptive control driven by peripheral sensors but it influences all levels of control; explicitly at the local level by modifying neuronal parameters and implicitly at the biomechanical level by the resulting new stable gait for walking on different terrains. As a result, through the design and implementation of the presented architecture, we show that adaptive, fast dynamic walking can be achieved in such simple dynamically stable bipeds, like RunBot.

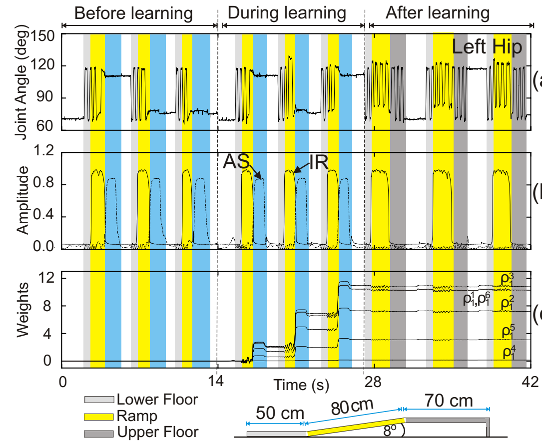

Fig. 5. The real-time data of left hip angle (a), reflexive AS and predictive IR signals (b) and all plastic synapses ρ_1 (c) in three situations where there was no learning for walking up a slope at the beginning. The data was recorded while RunBot was initially walking from a lower floor (light gray areas) to an upper floor (dark gray areas) through a ramp (yellow areas). Blue areas depict the situation where RunBot falls backwards and white areas where RunBot was manually returned to the initial position.

V. Acknowledgment

Discussion and implementation of the learning technique were supported by Christoph Kolodziejski.

References

[1] S. H. Collins, A. Ruina, R. Tedrake and M. Wisse, "Efficient Bipedal Robots Based on Passive Dynamic Walkers," *Science*, vol. 307, pp. 1082–1085, 2005.

[2] K. Okada, T. Ogura, A. Haneda, D. Kousaka, H. Nakai, M. Inaba and H. Inoue, "Integrated system software for HRP2 humanoid," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 3207–3212, 2004.

[3] Y. Ogura, H. Kondo, A. Morishima, Hun-ok Lim and A. Takanishi, "Development of a new humanoid robot WABIAN-2," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 76–81, 2006.

[4] M. Vukobratovic, B. Borovac, D. Surla and D. Stokic, *Biped locomotion: dynamics, stability, control and application*, Springer-Verlag, 1990.

[5] A. A. Biewener, C. T. Farley, T. J. Roberts and M. Temaner, "Muscle mechanical advantage of human walking and running: implications for energy cost," *J. Appl. Physiol.*, vol. 97, pp. 2266–2274, 2004.

[6] T. Geng, B. Porr and F. Wörgötter, "Fast Biped Walking with A Sensor-driven Neuronal Controller and Real-time Online Learning," *Int. J. Robot. Res.*, Sage press, vol. 25, no. 3, pp. 243–259, 2006.

[7] T. Geng, B. Porr and F. Wörgötter, "A Reflexive Neural Network for Dynamic Biped Walking Control," *Neural Computation*, MIT press, vol. 18, no. 5, pp. 1156–1196, 2006.

[8] P. Manoonpong, T. Geng and F. Wörgötter, "Exploring the dynamic walking range of the biped robot "RunBot" with an active upper-body component," *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, on CD, 2006.

[9] R. A. Brooks, "How to build complete creatures rather than isolated cognitive simulators," *Architectures for Intelligence*, pp. 225–239, 1991.

[10] Indexing of variables in this article organized as a hierarchical structure: Body-level (UBC = B, left-leg = L, right-leg = R). Leg level (hip = H, knee = K). Joint level (flexor = F, extensor = E). In general indices are omitted below the last relevant level, e.g. $S_{L,H,E}$ applies to the extensor of the hip of the left leg whereas S would apply to flexor *and* extensor of the hip and knee of the left and right legs.

[11] B. Porr and F. Wörgötter, "Strongly Improved Stability and Faster Convergence of Temporal Sequence Learning by Using Input Correlations Only," *Neural Computation*, vol. 18, no. 6, pp. 1380–1412, 2006.